ETSI GR MEC 036 V4.1.1 (2025-08)

GROUP REPORT

Multi-access Edge Computing (MEC); MEC in resource constrained terminals, fixed or mobile

Disclaimer

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

Reference

DGR/MEC-0036v411ConstrDevice

Keywords

edge, MEC, terminal

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

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 IPR online database.

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™**, **LTE™** and **5G™** logo 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 Report (GR) has been produced by ETSI Industry Specification Group (ISG) Multi-access Edge Computing (MEC).

Modal verbs terminology

In the present document "**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 studies how terminal units, mobile hosts and personal devices can be used to support cloud computing at the edge.

The study focuses on these aspects:

- Limited availability of compute resources for running MEC applications and it is impact on life cycle management of VMs, Containers or other form of virtual instances.
- Mobility of constrained terminals impacting reachability of MEC applications, maintenance of reasonable connectivity, device availability and discovery of appropriate services.
- Impact of unavailability of reliable high bandwidth backhaul connectivity (e.g. wired or wireless).
- Security and authorization to use a constrained terminal, privacy of user data.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

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 may be useful in implementing an ETSI deliverable or add to the reader's understanding, but are not required for conformance to the present document.

[i.1] ETSI GS MEC 003: "Multi-access Edge Computing (MEC); Framework and Reference Architecture".

[i.2] ETSI TS 123 501: "5G; System architecture for the 5G System (5GS) (3GPP TS 23.501)".

[i.3] ETSI GR MEC 031: "Multi-access Edge Computing (MEC); MEC 5G Integration".

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

[i.5] ETSI GR MEC 001: "Multi-access Edge Computing (MEC); Terminology".

[i.6] ETSI GS MEC 002: "Multi-access Edge Computing (MEC); Use Cases and Requirements".

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

[i.8] ETSI GR MEC 035: "Multi-access Edge Computing (MEC); Study on Inter-MEC systems and MEC-Cloud systems coordination".

[i.9] Christensen, K., Mertz, C., Pillai, P., Hebert, M., & Satyanarayanan, M. (February, 2019). Towards a distraction-free waze. In Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications (pp. 15-20).

[i.10] IETF RFC 7228: "Terminology for Constrained-Node Networks".

[i.11] ETSI GS MEC 021: "Multi-access Edge Computing (MEC); Application Mobility Service API".

[i.12] ETSI GR MEC 044: "Multi-access Edge Computing (MEC); Study on MEC Application Slices".

[i.13] ETSI GS NFV-IFA 011: "Network Functions Virtualisation (NFV); Management and Orchestration; VNF Descriptor and Packaging Specification".

[i.14] ETSI GS MEC 016: "Multi-access Edge Computing (MEC); Device application interface".

[i.15] ETSI GS MEC 013: "Multi-access Edge Computing (MEC); Location API".

[i.16] Satyanarayanan, M. (2017): "The emergence of edge computing", Computer, 50(1), 30-39.

[i.17] 3GPP TR 22.842 (V17.2.0) (2019-12): "Study on Network Controlled Interactive Services (Release 17)".

[i.18] Jerdan, S., Grindle, M., Van Woerden, H., & Kamel Boulos, M. (2018, July 6). Head-Mounted Virtual Reality and Mental Health: Critical Review of Current Research. Retrieved August 22, 2020.

[i.19] "Mitsubishi's New Electric Car Shows Off AR Dashboard in Concept Video".

[i.20] "Augmented and Virtual Reality: the First Wave of 5G Killer Apps". .

[i.21] Wang, Shuo, et al.: "A survey on mobile edge networks: convergence of computing, caching and communications", IEEE™ Access 5 (2017): 6757-6779.

[i.22] ETSI GS MEC 10-1: "Mobile Edge Computing (MEC); Mobile Edge Management; Part 1: System, host and platform management.

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the following terms apply:

constrained device: devices with limited compute and storage capability, requires minimum power, can be mobile

constrained MEC host: MEC host on a constrained device. Its capability depends on how restricted it is with compute, memory, and storage

MEC application group: set of MEC applications that can be instantiated on one or more Constrained MEC hosts and/or MEC hosts to realize specific functionality (e.g. a XR MEC application group is a set of MEC applications needed for an XR game)

terminal device: device that connects via an access network, fixed or mobile

3.2 Symbols

Void.

3.3 Abbreviations

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

5GC	5G Core network
5GS	5G System
CPE	Customer Premise Equipment
CPN	Customer Premise Network
DN	Data Network

FQDN Fully Qualified Domain Name
LADN Local Area Data Network
NF Network Function
PLMN Public Land Mobile Network
UE User Equipment

4 Overview

4.1 Introduction

The present document studies how terminal units, mobile hosts and personal devices can be used to support cloud computing at the edge.

Clause 5 documents the use cases that require cloud computing on terminal devices, identifies requirements, analyse applicability of MEC system and identify any gaps through Key Issues (KI).

Figure 1 illustrates the overall framework of the use cases considered. The framework consists of three logical layers, namely the network layer, computing layer, and application layer.

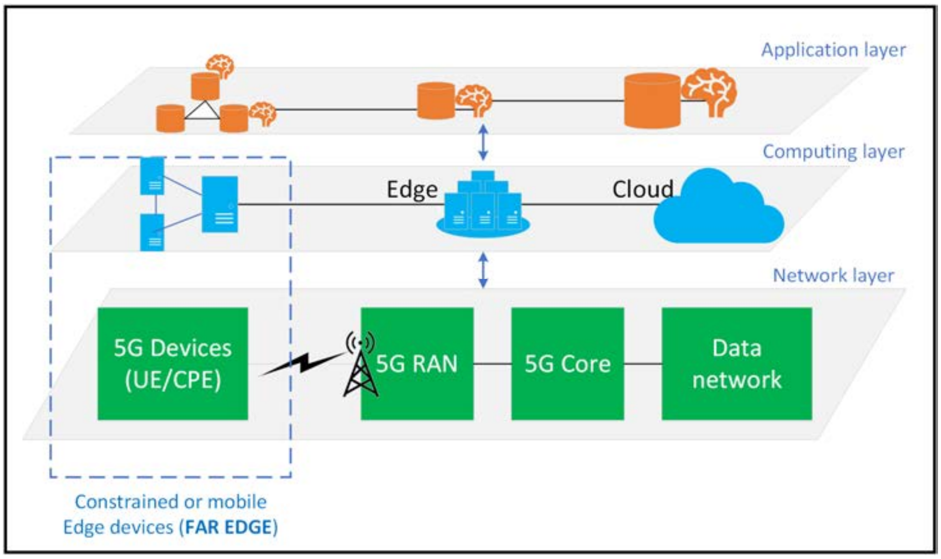

Figure 4.1-1: High Level Framework

The network layer is depicted using an end-to-end 5G network. The computing layer is composed of different computing tiers, namely, the central cloud, the edge cloud (e.g. Telco Edge) connected to network edge and far edge capabilities associated with the constrained devices (e.g. UEs or CPEs). Far edge capabilities may be embedded in the constrained terminal devices or dynamically provisioned. Constrained devices may be battery-powered, mobile, volatile, with limited compute and connectivity as compared to the traditional edge clouds. The constrained devices may collaborate and exchange information among themselves. The application layer, which may provide functionalities such as telemetry, training and inference, are envisioned to be distributed across different computing tiers, including far edge constrained devices. Applications and functions may be hosted anywhere in the computing stratum (cloud, edge or far edge devices).

Clause 6 proposes the possible solutions for closing the gaps. Solutions are evaluated for feasibility and provide recommendation if there are multiple solutions.

Clause 7 finally concludes this study with recommendation for any future work.

This work considers developments of other standard and industry bodies such as 3GPP, GSMA, 5GAA, LF and all relevant work done in ETSI.

4.2 Constrained Devices Description

Constrained Devices, when compared to contemporary computing devices, have reduced or absent characteristics such as power, memory, processing capacity, and other related aspects as defined in IETF RFC 7228 [i.10].

Examples of "constrained devices", which may be considered for the study, are described below:

- Small cells deployed inside a building, mall, or enterprise, enhanced with edge capability.
- Vehicle enabled with computing capability, capable to provide far edge cloud service to users inside the vehicle, as well as to vehicular applications for autonomous driving and safety. Wireless access, such as 5G, may be used as a backhaul for the in-vehicle constrained device. Roadside units may also be considered in this group.
- Sensor-carrying devices, which may be fixed or mobile, such as a Robot, Camera, a wearable device or an Automated Guided Vehicle (AGV), etc., may be enhanced with edge compute capability. These sensors (possibly also collocated with actuators) may use fixed access or wireless access (e.g. 5G, Wi-Fi) to connect to the network.
- Flying objects such as a drone/ UAV equipped with computing capability may provide far edge service. These devices may use wireless access such as 5G to connect to the network.
- UE (e.g. smart phone), supporting a personal IoT/ online gaming/ V2X network.

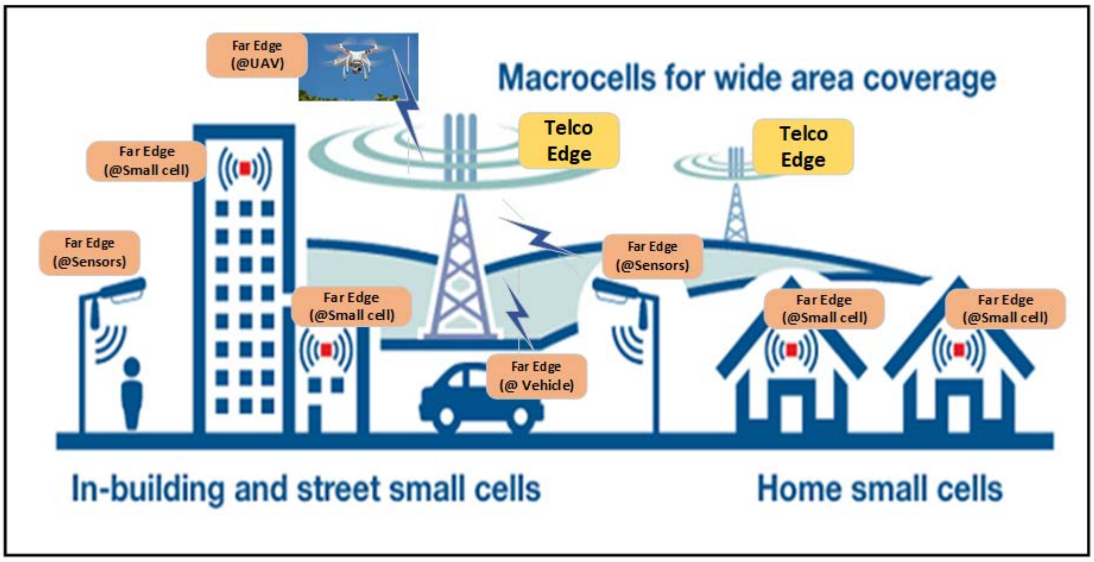

Figure 4.2-1: Far Edge deployment

Constrained devices, listed above, may be equipped with a "MEC platform" to provide far edge cloud services. Far edge service is anticipated to enable edge native computing, aiming to satisfy requirements of several applications, such as e.g. advanced AI/ML based mission critical applications. This study is analysing applicability of the MEC platform in these devices, which are limited by compute, storage, power, and connectivity, to enable edge native computing for such applications.

4.3 Assumptions in this study

In this study the MEC platform is assumed to be deployed at the constrained devices (i.e. terminal side) to provide far edge services. This is different from the basic concept of the MEC system as defined in ETSI GS MEC 003 [i.1] where the MEC platform is typically deployed at the network side. The overall MEC Framework and Reference Architecture need to be re-considered. In particular, the following concept and definitions, as specified in ETSI GS MEC 003 [i.1] cannot be assumed:

- MEC enables the implementation of MEC applications as software-only entities that run on top of a Virtualisation infrastructure, which is located in or close to the network edge.

- The MEC host is an entity that contains the MEC platform and a Virtualisation infrastructure which provides compute, storage and network resources for the MEC applications. The Virtualisation infrastructure includes a data plane that executes the traffic rules received by the MEC platform and routes the traffic among applications, services, DNS server/proxy, 3GPP network, other access networks, local networks and external networks.

Moreover, the existing features and functionalities specified in 3GPP in support for Edge Computing (e.g. as defined in ETSI TS 123 501 [i.2]) cannot be assumed.

5 Use cases

5.1 Use case #1: Smart Factory of the future

5.1.1 Description

This use case considers a production line in a smart factory. The production line is composed of machines that may be fixed or mobile (e.g. robots, robot arms, etc.) acting continuously or on-demand on the line. Additional sensors including video cameras are used for real-time monitoring and subsequent intervention by the machines, e.g. to stop the line or to remove a defective product. Such intervention by the machines is typically instructed/commanded by a remote factory worker acting upon real-time data received from the sensors and cameras over a local or wide area network (e.g. through a 5G network).

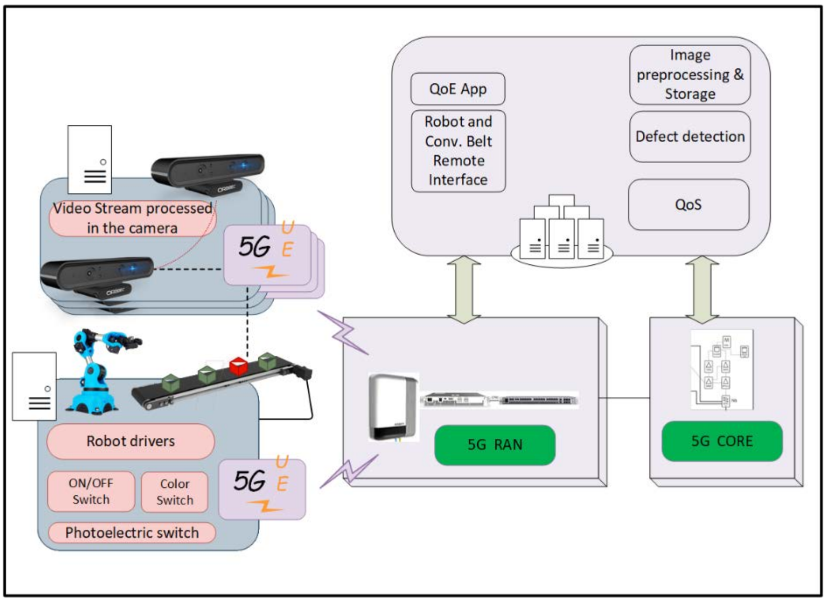

Figure 5.1.1-1: Smart Factory of the future

The machines and devices in the smart factory are assumed to have capabilities for networking, computing, and storage. The computing capability on the local machines and devices in the factory may support distributed data telemetry and intelligent functionalities locally. Numerous cameras and sensors, with the possibility for some cameras to be on-wheels (e.g. carried by guided vehicles), are continuously monitoring the production line. These cameras and sensors are capable of data storage, fast data analysis, including extracting and capitalizing on the corresponding knowledge in real-time. Running Federated Learning (FL) coupled with advancements in Deep Learning (DL) across multiple participating end devices opens possibilities for optimization of manufacturing processes in smart factory. Smart manufacturing process demands real-time inference of the data collected, to prevent delays, avoid mistakes and improve efficiency. To provide factory managers with the ability to quickly parse real-time data, make better informed decisions and recognize potential defects in production, a distributed localized edge computing solution is leveraged.

These local device capabilities are combined with additional (more sophisticated but mostly fixed) capabilities available in the end-to-end infrastructure (e.g. Telco Edge, Distant Cloud) connecting the smart factory to the remote digital worker. Running data analysis and detection in these devices enable real-time detection services. On the contrary, if data from a smart manufacturing unit is offloaded to the Telco Edge or any cloud service provider platform, the response may take between 50 to 200 milliseconds [i.16]

In achieving real-time data analysis for defect-free and improved production line, FL is adopted where privacy sensitive training data is generated and processed (possibly unevenly) across learning agents, instead of being transported and processed in a centralized edge cloud or distant cloud. This approach overcomes long propagation delays, choked backbone network and unacceptable latency. After the training phase, with model updates from the Telco Edge/cloud server, a fractional offloading approach is considered. This technique leverages the unique structure of Deep Neural Network (DNN). In the DL approach lower layers are computed on the device edge and higher layers are computed in Telco Edge. Augmenting DL with FL allows more accurate detection and fast convergence due to decreased number of communication rounds. Decrease in communication rounds may be attributed to additional computation performed on participating end devices. This approach may achieve detection and response time less than few milliseconds with greater geospatial specificity.

The distributed ML model enables detection of a defective manufactured piece and triggers a remote worker to command an intervention by some machines (e.g. robots or robotic arms) to stop the line or take the defective piece out of the production line to a certain destination. Such immediate intervention is done in real-time processing, with visualization of geometric features for manufactured parts at the remote worker location. Clearly the sooner a piece is detected as defective and taken out of the production line, the less scrap will be generated. Moreover, the faster the pieces are analysed, more pieces can be produced in each period.

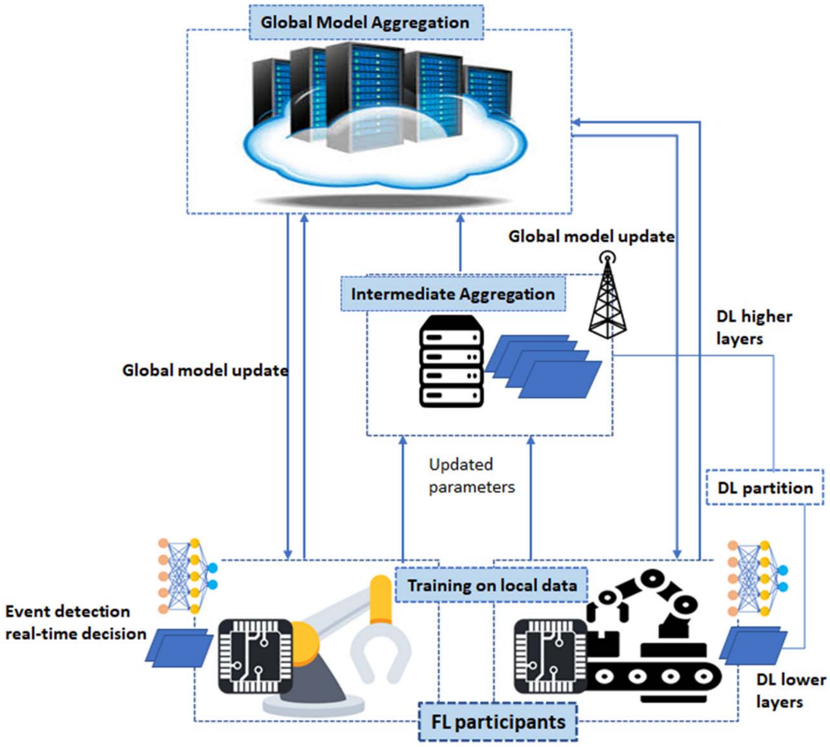

Figure 5.1.1-2: Smart Factory - Augmented FL-DL approach

To take advantage of such an ML implementation it is imperative that additional computation is performed on participating end devices. More than one learning agent, runs in parallel and participates in each round of training. Local updates are done in the end devices, prior to sending for global aggregation. Pre-processing with DL algorithm and Collaborative model training is conducted across the same set of FL participants with different data features. Quick convergence is achieved by limiting the number of local updates per interval of communication. During each training round, the global model is received by the participant and fixed as a reference in the training process at the device end. While training, the participant learns not just from local data, but also from other participants with reference to the fixed global model. This approach of hierarchical coordination in the process of data preparation, model training, evaluation and prediction improves the outcome of manufacturing by reducing waste, speeding up production and improving quality of goods produced.

5.1.2 Use Of MEC

5.1.2.1 Use of MEC in constrained device

5.1.2.1.1 Description

In the use case described above, there are many machines and sensors, such as Robotic arm, Camera, which collects data about the production line and processes this data. These machines fall under the category of Constrained Device, which can be used to provide far edge cloud service. Enabling far edge cloud service within these machines can allow service providers to deploy and manage advanced services. Application developers can also deploy their applications and use far edge service.

To promote application development and adoption, standardized interfaces to use far edge service can be useful. Standardized interface exposed by the far edge can enable:

- Deployment of application and services
- Service providers to provide services
- Application developers to consume services and develop new applications

5.1.2.1.2 Gap analysis

MEC platform already provides standard interfaces for application deployment and service consumption. MEC platform can be deployed in constrained device to provide such standard interfaces. However, due to limited resources of these devices, the MEC Platform can be reduced in functionality.

E.g. a reduced functionality MEC platform can have one or more of the following functions:

- Application deployment capability.
- Service producing capability.
- Service consumption capability, the consumed service can be deployed in Far Edge, as well as Telco Edge.

Whether MEC services, such as RNIS, BWM etc. are included in constrained device, can be left at the discretion of the service provider.

The following diagram illustrates the use of MEC in constrained devices within the Smart Factory of the Future use case, highlighting applicable MEC Management Plane interfaces. Additionally, User Plane interfaces are assumed to exist:

- Between the constrained devices, for example, enabling MEC App1 to communicate with MEC App 2.
- Between the constrained devices and the Telco Edge; enabling the MEC Applications on the constrained devices to communicate with other MEC Applications and consume available MEC services in the MEC system.

User Plane interfaces are not shown for simplicity.

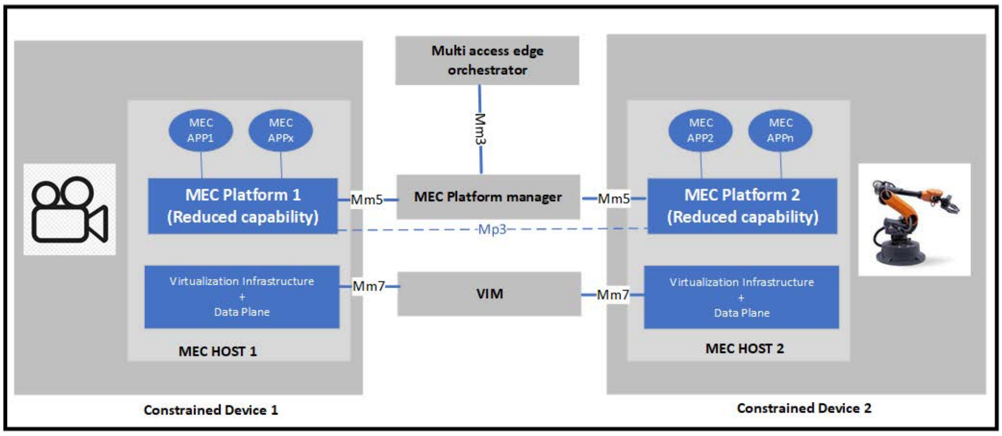

Figure 5.1.2.1.2-1: Constrained Device supported interfaces

Connectivity assumptions:

- A constrained device (such as a Camera or Robotic arm) can host both application clients and servers for those clients. MEC App1 and MEC App2, in constrained devices 1 and 2 respectively, are examples of such servers. Each server is associated with a MEC platform that is also hosted by the respective constrained device.
- Constrained Devices can support:
  - Management plane interfaces:
    - Mm5, Mm7 interfaces over 3GPP, Wi-Fi, Ethernet.
    - Mp3 interface can be over 3GPP, Wi-Fi, Ethernet or D2D.
  - User plane:
    - Between constrained devices over 3GPP, Wi-Fi, Ethernet or D2D technology.
    - Between constrained device and Telco Edge over 3GPP, Wi-Fi, Ethernet.

There are several open issues to enable the reduced functionality MEC platform in a MEC system:

[Open issue 5.1.2.1.2-1] Since the stable connection between the reduced functionality MEC platform and the rest of the MEC system cannot be always assumed, it is FFS handling instability with regards to:

- the connection between the MEC platform and the MEC platform manager then the MEC orchestrator (e.g. Mm5, Mm3, Mm2)
- the connection between the MEC host and the underlying virtualization infrastructure (e.g. Mp2)

[Open issue 5.1.2.1.2-2] It is FFS how to handle the unstable connection between a reduced functionality MEC platform and the other MEC platform.

[Open issue 5.1.2.1.2-3] It is FFS how to handle the unstable connection between an application client and the application server instance hosted in a reduced functionality MEC platform.

[Open issue 5.1.2.1.2-4] It is FFS how to handle the authentication and authorization related aspects.

These various connectivity options can give rise to multiple scenarios related to loss of connectivity calling for different remediations. For example, two constrained devices may lose management plane interfaces over 3GPP, Wi-Fi, but may still maintain user plane over D2D.

5.1.2.2 Telco Edge and Far Edge interaction

5.1.2.2.1 Description

The use case describes scenarios where the Far Edge capability of constrained devices is combined with additional capabilities available in the Telco Edge. MEC Applications running on the Far Edge hosted by constrained devices generate privacy sensitive data and process this data locally. E.g. a Federated Learning Agent (FLA) MEC app in the Far Edge processes sensitive sensor information locally. After an initial ML model training phase, the Far Edge FLA MEC app sends an updated model or ML parameters to a supporting MEC application(s) running on Telco Edge, as shown in Figure 5.1.1-2. Thus, one or more MEC applications in Far Edge, update an ML MEC application in Telco Edge.

The use case describes how the ML MEC application in Telco Edge exchange information with one or more instance of FLA MEC application(s) in the Far Edge. There can be many instances of the FLA MEC application in the Far Edge, but for a specific deployment instance, only few can exchange information with the ML MEC application in the Telco Edge. For every deployment instance a group (or set) of MEC applications in Far Edge and Telco Edge communicate and exchange information among themselves. MEC applications in Far Edge and Telco Edge can be part of more than one group, exchanging information among different group members.

5.1.2.2.2 Gap analysis

To enable the interaction between the application instances hosted in a Telco Edge and a far edge there are several open issues:

[Open issue 5.1.2.2.2-1] It is FFS how to handle an unstable connection between a far edge and a Telco Edge.

[Open issue 5.1.2.2.2-2] It is FFS whether additional support is needed for the communication between the applications hosted in the far edge and the Telco Edge, e.g.:

1) To create a group of MEC applications in Far Edge and Telco Edge, communicating among themselves for a particular deployment instance.
2) To create the group using a set of criteria, which is chosen based on deployment instance.
3) To allow one Far Edge MEC application can be part of multiple groups simultaneously for different deployment instances.
4) To allow group of MEC applications in Far Edge and Telco Edge for each deployment instance, can change with time without any impact on service.

5.1.2.3 Mobility of constrained devices

5.1.2.3.1 Description

There are numerous cameras and sensors on the factory floor, with the possibility for some cameras to be on-wheels (e.g. carried by guided vehicles), are continuously monitoring production lines in the factory. This set of mobile cameras and sensors are constrained devices, which can provide Far Edge service for one production line while another set of sensors can provide Far Edge service for another production line. As these mobile constrained devices move around, due to coverage and proximity, a MEC application (e.g. Federated Learning Agent or FLA) hosted in the Far Edge can change the deployment instance it was originally servicing. E.g. the FLA can be previously supporting Production Line 1, but later start monitoring a different production line, i.e. Production Line 2.

5.1.2.3.2 Gap analysis

Considering the potential mobility of the constrained devices, the following open issues need to be addressed:

[Open issue 5.1.2.3.2-1] How to notify, the service consumers potentially hosted in other MEC platforms, the change of the service availability produced by the MEC applications hosted in the Far Edge.

[Open issue 5.1.2.3.2-2] It is FFS whether additional support is needed to allow a MEC application in Far Edge join or leave a specific deployment instance when triggered by users, such as supervisor on the factory floor.

5.1.3 Evaluation

5.1.3.1 Use of MEC in Constrained device

Clause 5.1.2.1.2 describes how Management Plane and User Plane for Constrained Device can be connected over different technologies. MEC App1 in Constrained Device 1 registers as a service producing application, which is consumed by MEC App2 in Constrained Device 2.

Referring to the Figure 5.1.2.1.2-1 in clause 5.1.2.1.2, various scenarios can be identified based on Management Plane connectivity status and various user plane conditions under it. The following clauses identify the possible scenarios.

5.1.3.1.1 Scenarios related to Management Plane of one constrained device

In certain condition one Constrained Device can lose Management Plane connectivity while the other maintains it. If the management plane of Constrained Device 1 is considered, the following scenarios can be identified:

- Constrained Device 1, loses Management plane connection to MEC System and User Plane connection to Constrained Device 2.
- Constrained Device 1 loses Management plane connection to MEC System but User Plane connection to Constrained Device 2 exists over D2D type connection.
- Constrained Device 1 continues to maintain Management Plane connection to MEC System, but User Plane connection to Constrained Device 2 is lost.

If the management plane of Constrained Device 2 is considered, the following scenarios can be identified:

- Constrained Device 2 loses Management plane connection to MEC System and User Plane connection to Constrained Device 1.
- Constrained Device 2 loses Management plane connection to MEC System but User Plane connection to Constrained Device 1 exists over D2D type connection.
- Constrained Device 2 continues to maintain Management Plane connection to MEC System, but User Plane connection to Constrained Device 1 is lost.

5.1.3.1.2 Scenarios related to Management Plane of two constrained devices considered jointly

In certain situation both Constrained Devices can lose or maintain Management Plane connectivity simultaneously. When Management plane connectivity for Constrained Device 1 and Constrained Device 2 is lost, but:

- User plane exists over D2D.
- User plane is also lost.

In another scenario, Management Plane is available for Constrained Device 1 and Constrained Device 2:

- User plane is not available between the devices.

5.1.3.1.3 Consideration for the scenarios

Constrained devices can handle temporary loss of Management Plane by relying on the current configuration. Prolonged loss of Management plane can be assumed by the MEC system as a loss of the constrained device and try to reconfigure.

In certain situations, temporary loss of user plane connection can be handled by MEC Application instance while operating in autonomous mode.

NOTE: In autonomous mode a MEC Application instance can operate independently without consuming information from other MEC Application instance for the time of user plane connection unavailability.

But loss of connectivity in the user plane for prolonged period can be an issue.

In the present document, scenarios involving loss of user plane, while management plane exists or not, will be evaluated further.

5.1.3.2 Unstable Connection in Telco Edge and Far Edge Interaction

A Constrained Device can have connectivity to Telco Edge and MEC system over access technologies like 3gpp, Wi-Fi, etc. This can carry Management Plane and User Plane. Constrained Devices can also have D2D type connection to other Constrained Devices.

A constrained device losing management plane connection to MEC system may also imply that it lost user plane connectivity to Telco Edge. However, this may not be always true, and the constrained device can maintain user plane connectivity to Telco Edge and other constrained devices while management plane is lost.

Losing management plane will imply that from MEC system's point of view the MEC host and MEC App in the Constrained Device is not available for management purpose. In such condition, the MEC system may mark the MEC Host in the Constrained Device and the MEC App hosted in the Constrained Device as unavailable for management, application deployment, discovery, etc.

The Constrained Device can lose management plane, but the user plane with Telco Edge or other Constrained Devices can exist. The existence of the user plane will not be known by MEC system. The constrained device can continue interacting over the user plane, while management plane does not exist.

When the Constrained Device loses user plane with Telco Edge, the MEC App hosted in Constrained Device cannot send sensed data for ML model training to the MEC App hosted in Telco Edge. ML model updates will stop due to inaccessibility of training data. In FL, a combined model is created by averaging the parameters of the individual models and broadcasting to other MEC Applications hosted in constrained devices (e.g. for inference). Due to interruption from one MEC App, the global ML model broadcasting to Constrained Devices may be inaccurate and lead to in-efficient operation.

A Constrained Device can lose user plane connectivity with another constrained device. When ML MEC Applications in Constrained Devices are training over heterogenous dataset, model updates are exchanged between the ML MEC Applications hosted in the Constrained Devices. Losing user plane between Constrained Devices, alters the FL application deployment topology and may affect the performance of learning process of the Constrained Devices.

5.1.3.3 Group of MEC applications in Telco Edge and Far Edge interaction

In the smart factory use case, FL/ML applications are distributed across Telco Edge and Constrained Devices. FL applications are MEC applications hosted in MEC Hosts in Telco Edge and Constrained Devices (e.g. Camera and Robotic Arm). Some of the FL applications can register as a service with the MEC Host.

The smart factory use case can be assumed to be realized through the deployment of the FL MEC Application (MECappT) hosted at Telco Edge, FL MEC Application (MECappC) hosted in Camera1 and FL MEC application (MECappR) hosted in Robotic arm1.

MECappT requires updates from MECappC and MECappR, hosted specifically on Camera1 and Robotic arm 1 respectively. MECappC in Camera 1 updates MECappR in Robotic arm1. On the factory floor there may be Camera2 hosting MECappC and Robotic Arm2 hosting MECappR. But in the smart factory use case, it is assumed that only MECappC hosted in Camera 1 and MECappR hosted in Robotic arm1 can be used to monitor and operate on the production line. MECappT knows, it needs to interact with MECappC in Camera 1 and the MECappR in Robotic arm1. MECappT also knows how to reach the MEC Applications.

The selection of MEC Applications, which are required to interact among themselves, can be done based on certain contextual situation and resource-related criteria such as, the constrained device where it is hosted, capability of the constrained device, location, etc. A third-party service provider or application developer can provide the criterion to MEC system operator. Based on these input, MEC system operator can select the set of MEC applications in Telco Edge and Constrained device.

Application developers and third-party service providers can dynamically update, change, modify the FL MEC Applications running on MEC Hosts. MEC Applications can be dynamically reconfigured. E.g. a third-party application developer deploys a new version of the MECappC hosted in Camera1 and maintain the existing relationships with MECappT and MECappR. MEC systems do not allow dynamic reconfiguration of MEC Applications, therefore MEC Applications need to be redeployed each time.

A service provider or operator can change the deployment configuration, which is a set of MEC Applications hosted on specific MEC hosts, dynamically by:

- Adding FL MEC Application hosted in Camera2 (MECappC) in the monitoring of production line 1. MECappC in Camera1 and Camera2 becomes part of the monitoring of Production line 1.
- MECappC in Camera 2 can continue to monitor another production line, i.e. Production Line 2.
- Removing FL MEC Application hosted in Camera1 (MECappC) after adding Camera2.

5.1.3.4 Mobility of Constrained Device

MEC Applications hosted at the Telco Edge (MECappT) control the overall operation and monitoring of the production lines. The MECappT App may be specific to a production line and customer. Within the smart factory, constrained devices, hosting MECappC, are utilized to perform production line tasks.

MECappC is hosted on a mobile constrained device (Camera1), includes FLA functionality, and utilizes the devices' s cameras/sensors. MECappC may be initially providing monitoring for production line #1 and its associated MECappT Apps.

As Camera1 moves to a new location on the factory floor and can no longer provide monitoring service for production line #1. Camera 1 can be removed from monitoring production line#1 and a new Camera N, which can provide monitoring service, can be added to the task.

While Camera 1 moved to a new location, it can provide monitoring service to monitoring task of production line#2. Camera 1 can be added to the task of monitoring production line 2.

MEC Applications hosted on constrained devices depends on their location to provide monitoring service. As these devices move, the MEC Applications need to be dynamically added or removed from a task. Whether MEC system can support dynamic addition and removal of MEC Applications, hosted in constrained device, needs to be further analysed.

Additionally, a user in the factory (supervisor or technician, authorized by the service provider to manage the edge application) may be alerted when a MECappC App is removed from a deployment instance (e.g. production line #1 is not being monitored). In response, the user may introduce a new constrained device into the smart factory with MECappC hosted in Camera2 to resume monitoring for production line #1.

The MEC support of dynamic addition of MEC hosts needs to be further analysed.

5.2 Use case #2 Autonomous Vehicular Applications

5.2.1 Description

Autonomous vehicular applications demand real-time decision-making under high reliability and low latency, even when network connectivity is lost. For instance, applying breaks in a platoon of vehicles or robots cannot afford millisecond range latencies. Remotely controlled vehicles should stay operational even under temporary losses of connectivity. Increasingly vehicles are being equipped with computing resources, where distributed Machine Learning (ML) techniques, such as Federated Learning, with minimum signalling overheads are deployed.

Federated Learning (FL) is a distributed learning technique where privacy sensitive training data is generated and processed (possibly unevenly) across learning agents, instead of being transported and processed in a centralized edge cloud or distant cloud. Federated Learning allows each agent (e.g. deployed on a far edge constrained device) to compute a set of local learning parameters from the available training data, referred to as local model. Instead of sharing the training data, agents share their local models with a central entity (e.g. Edge cloud), which in turn does model averaging then sharing a global model with the agents (e.g. on the far edge constrained devices). As such, Federated Learning does not require exchanging training data, thus reducing the communication latencies.

Moving nodes, e.g. V2V, Edge Robotics or UAV Swarms, deploy decentralized learning to minimize central control and coordination. Use of low-latency distributed learning for such nodes enables real-time applications with limited battery power. Use of Distributed Multi-Task Learning (DMTL) and Federated Learning on these mobile nodes allows temporary loss of link or node failure to be un-noticeable.

FL and DMTL techniques have different requirements of local computation on the devices (including constrained devices) as well as communication interactions with the central entity. These requirements depend on convergence, accuracy, and robustness of the trained model. The resource constrained moving nodes collectively train a high-quality centralized model in a decentralized manner for different ML architectures. As operating condition changes dynamically, such as mobility, link quality, these nodes adapt and continue providing service with almost no degradation in quality and reliability.

5.2.2 Use Of MEC

Autonomous Vehicular applications require task to be distributed across different hosts in Data centre, Telco Edge, and on Vehicles. MEC hosts can be deployed in a vehicle. MEC system can be used to support vehicular applications deployed on in-vehicle MEC hosts. The In-vehicle MEC hosts can be a Constrained MEC host.

The In-vehicle Constrained MEC hosts can host vehicular MEC applications and MEC Application Groups, which execute vehicular application tasks and can be managed by MEC reference points, such as:

- Management plane interfaces:
  - Mm5, Mm7 interfaces over 3GPP, Wi-Fi, Ethernet.
  - Mp3 interface can be over 3GPP, Wi-Fi, Ethernet or D2D.
- User plane interfaces:
  - Between constrained devices over 3GPP, Wi-Fi, Ethernet or D2D technology.
  - Between constrained device and Telco Edge over 3GPP, Wi-Fi, Ethernet.

5.2.3 Evaluation

Analysis of the "Autonomous Vehicular applications" use case led to similar Gaps, Key issues identified in the analysis of Use case #1 and #3.

For example, the following gaps and key issues also apply for the "Autonomous Vehicular applications" use case:

- Loss of user plane and management plane
- Dynamic addition of constrained MEC hosts
- Managing group of In-vehicle constrained MEC hosts
- MEC Resource allocation and re-distribution
- Transfer or relocation of MEC applications

Solutions have been proposed in clause 6, to address these key issues. The same solutions can be used to address the key issues derived from the Autonomous Vehicular applications use case.

5.3 Use case #3: Multi player XR multimodal mobile gaming

5.3.1 Description

The future of gaming is "Community gaming". These games have inspired game developers and service providers to create such community games on mobile devices. People are hanging out at what is known as a "Digital Third Place", which more closely resembles a skate park than an arena for competition.

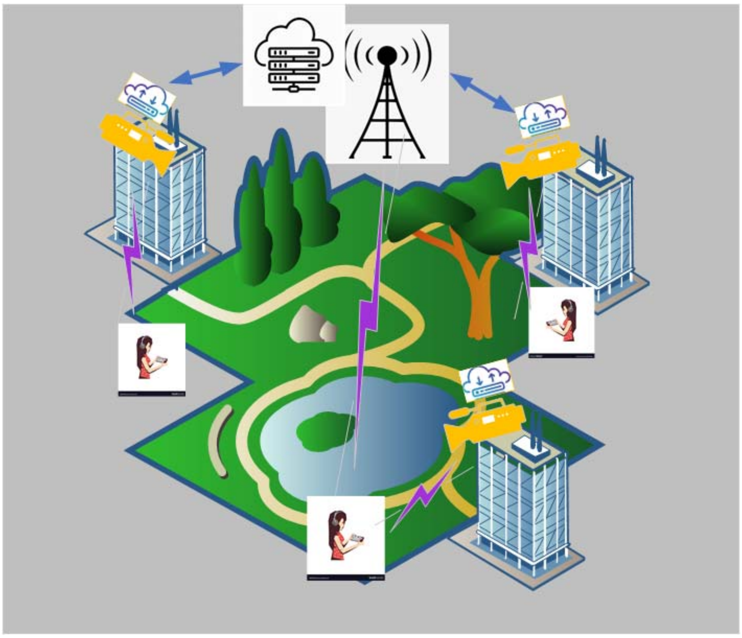

Figure 5.3.1-1: Multi player XR game play scenario

A common way to create this "Digital Third Place" is extended reality (XR). This new reality will unlock new social interactions. The quality of an XR game hinges, first and foremost, on the presence of reality. Artificial intelligence is no longer the stuff of science fiction, it is lodged in the fabric of our everyday lives and video games are no exception. The idea of AI has been expressed in gaming for decades, but now it is becoming more advanced and at a greater scale.

In an XR gaming application system, the XR application first processes the scene that a mobile player is watching in real-time and identifies objects that will be targeted to overlay of ultra-high-resolution video. It then generates high-resolution 3D image scenes related to the perspective of the mobile player in real-time. These generated video images are then overlaid on the view of the real-world as seen by the mobile player. A key goal of XR gaming is to provide the sufficient Quality of Experience (QoE) to all the players to provide good game play. This includes the individual-based experience of a player but also the game fairness requirements across all participating players in terms of both timeliness and quality of content delivered to player's devices at different network locations.

To support XR gaming environment, it is anticipated that 5GS needs to support [i.17]:

- motion-to-photon latency in the range of 7 - 15 ms, while maintaining the required user data rate of 1 Gbps and more;
- motion-to-sound delay of 20 ms or less.

For some XR use cases like remote machinery control [i.17], the required latency may be lower than 5ms and depending on the frame rate and DoF (degree of freedom) the required bandwidth can be 1 Gbps to 5 Gbps. Current mainstream games require 60 frames per second, which translates to a frame interval of 16,67 ms. Taking out the delay for rendering and encoding/decoding processing, the Round-Trip Time (RTT) delay over 5G link should be less than 5 ms.

To provide realistic gaming experience, modern games capture video and user movements, gestures, proxemics, haptics, eye movement, etc. In a community game, other players' information is also captured and processed which influences the gameplay of other participants. Game developers feed this information to an AI engine and XR engine. The time to act upon the output of AI/XR engine is very limited. The XR games of the future (Mixed reality, Extended reality games) may operate at a minimum 90 fps [i.18]. To support these required frame rates, round-trip time between a user action and the rendered image being shown to the user is 11 ms. Considering XR processing and rendering takes between 2 and 10 ms [i.17], the access latency over the wireless link gets pushed down to 1 ms or lower. Until sub-millisecond latency on the wireless link can be achieved, service providers are finding other means to maintain the overall required latency.

Gaming service providers and network providers are following new application deployment models, where additional AI/XR services are made available closer to the user. These new service platforms are called "Device Edge", which can directly interact with users. These services bring down the processing time, support high data rate and support lower latency.

Users may obtain time critical "game scene information" from these services. The AI and XR engines, which are deployed on the device edge interacts and shares information for more accurate outcome. These localized services provide input to central game controller. Players still obtain non- critical information from the game services deployed in Telco Edge. The local AI and XR engines influence local play while the overall AI and XR engines coordinate the game play across many device edges.

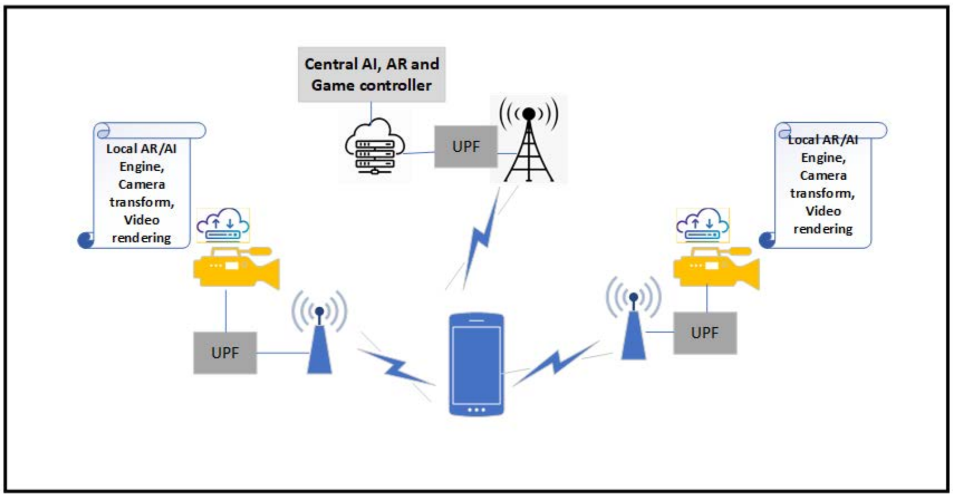

Figure 5.3.1-2: Deployment model

Following the Device Edge deployment model, service providers can provide AI, XR and rendering services on demand at the device edge. Localized XR, AI and rendering services can consume local data, reduce load on networks and meet the stringent bandwidth and latency requirements. The global AR and AI engines may combine the outcome from the local services to feed the main game coordinator.

Game developers/Service providers may follow a hybrid approach to flexibly deploy these functions depending on the type of game, device capability, battery power status, network condition, bandwidth, and latency requirements.

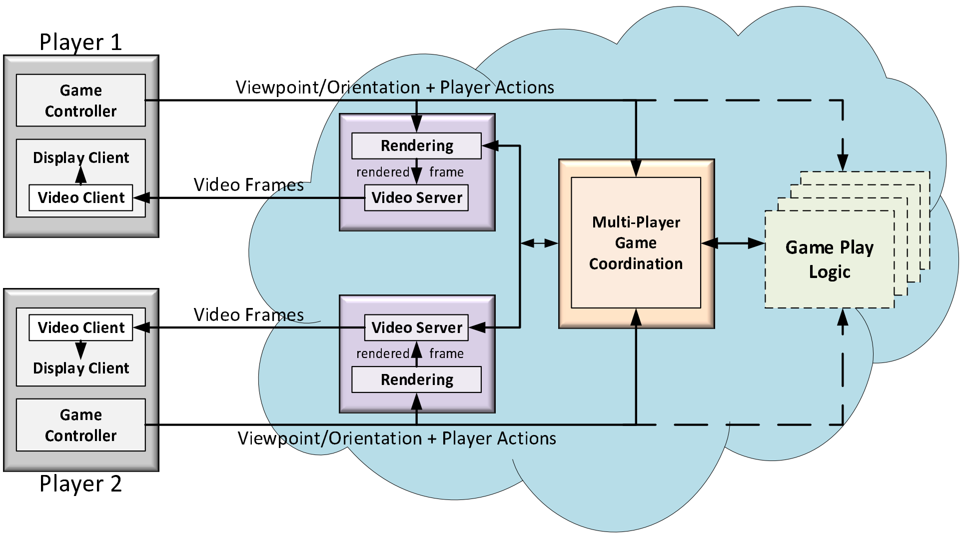

Figure 5.3.1-3: Multi-tier function portioning

Each player has a game controller which sends viewpoint, orientation, and actions to the edge network. The edge network accepts these inputs and renders a new frame which is sent to the video client. Upon receipt by the video client it is displayed to the player. In the edge network the player viewpoint, orientation and actions are also shared with the multi-player game coordinator as well as the game play logic. The game-play logic and multi-game coordination function interact to facilitate the game play.

5.3.2 Use Of MEC

A XR application requires task to be distributed across different hosts in Data centre, Telco Edge and on Constrained MEC hosts. MEC system can be used to support XR application, assuming XR tasks are deployed in Telco Edge MEC Hosts and Constrained MEC host.

The Constrained MEC hosts can host XR MEC Application, which execute XR tasks and can be managed by MEC reference points, such as:

- Management plane interfaces:
  - Mm5, Mm7 interfaces over 3GPP, Wi-Fi, Ethernet
  - Mp3 interface can be over 3GPP, Wi-Fi, Ethernet or D2D
- User plane interfaces:
  - Between constrained devices over 3GPP, Wi-Fi, Ethernet or D2D technology
  - Between constrained device and Telco Edge over 3GPP, Wi-Fi, Ethernet

As shown in Figure 5.3.1-3, the 'Game Controller' and 'Display Client' are client devices, which includes a terminal (e.g. XR glass) and a constrained MEC host hosting a MEC application, e.g. video client for game scene display. The Game controller and Display Client are not connected to each other.

In some scenarios a terminal and MEC application can be either co-located or separated. E.g. the display client can have the "XR Glass" and "constrained MEC host" co-located or the "XR Glass" can be connected wirelessly or through wire to the "constrained MEC host".

The 'Rendering' and the 'Video Server' components can be hosted on one or more constrained MEC hosts. These constrained MEC hosts could be deployed in a home (e.g. CPE), in a gaming arena (e.g. small cell), or in a vehicle and need to deliver video to the display clients with low latency, corresponding to a frame rate of 60 fps or even 90 fps.

The 'Multiplayer Game Coordination' component, which has moderate latency requirement, can be deployed as a MEC app in the Telco Edge or in Constrained MEC Host.

The 'Game Play Logic', which can perform its function despite of large latencies, can be deployed as MEC Applications in the Telco Edge or in the Cloud.

The impact of resource constraints and distribution of MEC applications in one or more constrained MEC hosts, hosting XR MEC applications, is discussed further.

5.3.3 Impact of resource constraints in constrained MEC hosts hosting XR MEC Applications

5.3.3.1 Description

In this use case, the XR MEC application hosted on a constrained MEC host needs to deal with the changing characteristics of communication (e.g. latency, bandwidth, unavailability) and availability of compute resources (e.g. CPU, GPU, TPU, memory). The XR tasks are computationally intensive, as well as energy intensive.

Computationally intensive XR tasks deployed in Telco Edge and Constrained MEC hosts faces varying computational requirement. Computational requirement includes number of CPU/GPU cycles required, the amount of memory required and the number of I/O requests per unit time. It is possible that during the operation sometime the compute requirement can be low and sometime the compute requirement can be very high. This compute requirement can be high because of computational algorithms that run to process XR data. On the other hand, the compute requirement can be low when the XR application is not using image processing algorithms or not using data intensive AI/ML tasks. The compute requirement is dynamic and can be impacted by the user action, mobility of the device, connectivity condition, etc.

When the compute requirement is high for constrained MEC hosts, they may not be able to handle the task and the overall operation of XR service will be impacted. The operation of the XR service is impacted because the processing slows down, and battery can drain out. The QoS level can degrade, and the service can be unacceptable.

5.3.3.2 Gap Analysis

The following open issue needs to be studied further:

[Open issue 5.3.2.1.2 - 1] Impact of varying resource requirements of an XR MEC application, hosted in constrained MEC host and Telco Edge, on the QoE experienced by user.

5.3.4 Impact of BW fluctuation on XR MEC Applications hosted in constrained MEC host

5.3.4.1 Description

The changing characteristics of the communication can result in the network bandwidth fluctuating at runtime as the client devices moves, while the user is moving.

The bandwidth fluctuation can happen between:

- Terminal and MEC application, if separated at deployment. A tethering link exists between the XR glasses (as display client) and constrained MEC host (hosting video client). This is internal to the XR application. The internal XR application bandwidth fluctuation is not in MEC system scope and is not considered in the present document.
- MEC application at the client device and the Rendering or Video server, e.g. deployed on a CPE. The bandwidth fluctuation between MEC application and Video server is considered for the evaluation.

The bandwidth fluctuation impacts the user's XR experience (for example, even making an XR game unplayable). This is because the fluctuating bandwidth will impact the response time (latency) as seen by the XR application.

Current XR applications are organized around a processing loop that starts with the client devices sending data generated by the user (e.g. user's commands, video captured from the user's viewpoint, etc.) and ends with the XR platform running on the edge returning a resultant XR rendered video frame. A video frame is the total amount of video information presented on a display at any time. Assuming that the frames are being generated at a rate of 60 fps, the time taken to return the resultant frame is 16,67 ms (1/60 approximately) + the network latency between the UE and the XR system running on the edge. This is essentially the "application response" time that this XR system takes from when the user makes some kind of movement (say turns their head to look around) to when the corresponding frame is rendered on the user's device.

The desired application response time is equal to or below 20 ms [i.12] for a realistic display of game play. If the time taken to generate the frame is equal to or below 16,67 ms, then the network latency should be equal to or below 4 ms.

Any fluctuation of the bandwidth that results in a network latency of more than 4 ms will cause the response time (latency) as seen by the XR application to become more than 20 ms resulting in a low QoS and making an XR game unplayable.

5.3.4.2 Gap Analysis

The following open issue needs to be studied further:

[Open issue 5.3.2.2.2 - 1] How to maintain sufficient XR quality of experience in the face of fluctuating bandwidth?

5.3.5 Impact of MEC application distribution across several MEC hosts

5.3.5.1 Description

An XR application, such as the multi-player game, can be decomposed into multiple MEC Applications as described in clause 5.3.1. These MEC Applications can support the following functionalities:

- Display client and/or video client as a MEC App
- XR Rendering service as a MEC App

- Video server as a MEC App
- Game coordinator as a MEC App
- Several game play logic entities as MEC Applications, depending on the internal XR application need

The MEC Applications listed above are components of an XR Application. The application components interact at application level. MEC system can support discovery of the application components, when registered as MEC service. The application components can be deployed in CMH and Telco Edge depending on service requirements. E.g. "Display Client", "XR Rendering" and "Game co-ordinator" can be deployed as MEC App in CMH and "Game play logic" MEC app in Telco Edge. The "Game play logic" MEC app in Telco Edge can communicate with components of the same application, like "Display Client", "XR Rendering" and "Game co-ordinator" MEC Applications deployed in CMHs. The interaction can be restricted within these set of application components.

The "Game play logic" MEC App in Telco Edge can have the ability to discover and learn about availability and un-availability of other application components, like the "Display Client", "XR Rendering" and "Game co-ordinator", which are part of the same application. Application components deployed on CMH, can be impacted by mobility of the host. E.g. the "Game play logic" MEC App in Telco Edge can be updated about availability and unavailability of "XR Rendering" MEC app, which can be relocated from one CMH to another due to resource constraints.

5.3.5.2 Gap Analysis

The following open issue needs to be studied further:

[Open issue 5.3.2.3.2 - 1] How Application components distributed as MEC Applications, among several MEC hosts, impacts discovery and management in a MEC system?

5.3.6 Evaluation

5.3.6.1 Impact of varying resource requirement for XR MEC Application

XR application is composed of computationally intensive XR tasks. The compute resource requirement of XR tasks changes dynamically due to user mobility and network connectivity.

Each XR task can require multiple resources. For example, the XR tasks might require resources such as Central Processing Unit (CPU), network bandwidth, battery energy, file cache state, and memory. The requirement of such resources change during the run time of the application for various reasons such as the algorithm being used, the amount of data to be processed.

When the compute resource requirement goes up, resource allocation can be optimized by re-allocating additional resources which were under-utilized. On the other hand, when compute resource requirement goes down, compute resources can be reclaimed back from a XR MEC application.

To maintain a minimum level of acceptable QoS, when compute requirement for a XR task changes, the XR tasks needs to be supported by appropriate allocation and re-allocation of compute resources.

So, the Key Issue that needs to be addressed:

- How XR MEC applications can communicate its need for additional compute resources or low compute resource requirement to MEC system?
- How MEC system can support allocation and re-allocation of compute resources to XR tasks when compute resource requirement changes?

5.3.6.2 Impact of high variability of bandwidth on XR MEC Application

XR applications that are running on constrained devices are impacted by the variability in the bandwidth of the network links, especially wireless connections. As discussed above, the consequence of the high variability of bandwidth is that when the bandwidth is low, the response time goes above 20 ms making the XR game unplayable.

An XR MEC application hosted on a constrained MEC host needs to adapt and consider changing characteristics of communication as a user moves around.

This bottleneck of low bandwidth can be tackled by speculation. Speculation is defined as performing an operation to generate data in advance of receiving a request on the possibility that it will be requested. For XR games, video frames are the generated data of interest, because of speculation. The speculated video frame can be pre-generated and when a request is made, rendered on the XR device with low latency.

Video frame speculation is to predict future frames(s) based on previous video frames, by tracking information about previous frames as well as game players movement, surrounding static and dynamic objects, relative motion among player and objects.

Referring to the use case Figure 5.3.1-3, MEC system can track the location and motion trajectory of the display client, which is hosting the video client on a constrained MEC host. Also, MEC system has the knowledge of available resources at the Display client, typical availability of bandwidth over time at different locations. With the availability of a wide range of information about the client devices, MEC system can support video frame speculation capabilities to maintain acceptable quality of experience for a user.

It is FFS, how MEC system can enable an XR application by supporting speculation capabilities, utilizing its knowledge of available resources to mitigate the fluctuating bandwidth problem?

5.3.6.3 Impact of MEC App distribution among several MEC hosts

In the XR Gaming application, when an application component MEC App in Telco Edge needs to interact with other application components of the same XR application, deployed as MEC Applications in CMH, number of signalling exchanges and processing within the MEC system increases.

The "Game play logic" MEC app in Telco Edge can communicate with components of the same application, like "Display Client", "XR Rendering" and "Game co-ordinator" MEC Applications deployed in several CMHs. The "Game play logic" MEC App in Telco Edge discovers the other application component MEC Applications, like "Display Client", "XR Rendering" which are part of the same XR application. There can be one or more discovery requests issued by the MEC App in Telco Edge. Multiple discovery responses can be exchanged between constrained MEC host and MEC host in Telco Edge. The MEC Applications in constrained MEC hosts can provide availability and un-availability notification to the MEC App in Telco Edge. Exchange of these notifications can increase, when the constrained MEC hosts are mobile or overloaded and low in compute capability. Hence, distribution of MEC Applications, which are components of an application, among many constrained MEC hosts, increases signalling exchanges between MEC host in Telco Edge and constrained MEC hosts.

Processing overhead increases as MEC system stores information about the distributed application components, deployed as MEC applications. The information about the MEC Applications can be updated frequently, which increases processing requirements. MEC system can process multiple discovery requests and responses, adding to the processing requirements.

Key Issue that needs to be addressed:

- How to reduce processing overhead and multiple signalling communication among MEC Applications, which are components of an application, deployed in Telco Edge and CMH?

5.3.6.4 Impact of MEC app relocation when deployed on a Constrained MEC Host

A XR application may include the several components, as described in clause 5.3.1. For example:

- A Display Client that presents XR video to an individual player. The Display Client may be realized as a Client Application that uses a XR Rendering MEC app in the MEC system.
- An XR Rendering MEC app that generates (or renders) video for a player based on the individual player's viewpoint and provides it to the Display Client.
- A Game Coordinator or Game Server MEC app that considers actions and context from all players and provides information to each player's XR Rendering MEC app to generate appropriate video for that player.

The XR MEC Applications may be deployed on CMHs or Telco MEC hosts, depending on their requirements, the configuration MEC system (location of CMH, Telco MEC hosts, etc.), and access network.

For example:

- Display client deployed as an application client on a player's mobile device (which includes a CMH).

- XR Rendering MEC app deployed on the CMH in a player's mobile device.
- Game Coordinator MEC app deployed on a local on-premises CMH, for example in a home or XR game studio.
- Other game play logic aspects may be deployed as micro-services in other MEC Applications in the Telco Edge or Cloud, as described in clause 5.3.1.

The XR Render MEC app is processing and power intensive. The CMH, where it is deployed, may run low on battery capacity, computing power, or other constraint. As a result and in order to maintain the player's XR quality of experience, the XR MEC app needs to be transparently relocated to another MEC host (CMH or Telco MEC Host). Transparent relocation is when a MEC app instance or application context is transitioned across MEC hosts (including CMH's) without loss of service to an application client. If battery levels (or other constraints) improve on the CMH, the XR Render MEC app may be relocated back in order to best utilize resources in the MEC system.

Key Issues that need to be addressed:

- How to transparently transition or relocate a MEC application instance from a CMH to another CMH or a Telco MEC host? Or vice-versa?
- How can a MEC application or an application client inform the MEC system about a need for transparent relocation?

Applicability of ETSI GS MEC 021 Application Mobility capabilities supporting user context relocation and/or application instance relocation from one Telco Edge MEC host to another needs to be considered for CMH relocation solutions.

5.4 Use case #4: In Vehicle Infotainment

5.4.1 Description

The automotive industry has been undergoing dynamic changes in the recent years. Technologies like In-vehicle infotainment, AR dashboard [i.19], and many such innovations are gaining immense popularity. Such services require a diverse embedded computing environment, responsible for handling various functionalities of different nature. To handle such diverse and complex processes, ETSI GS MEC 002 proposes availability of "In-vehicle MEC hosts supporting automotive workloads encountered in a passenger vehicle" [i.6]. Each passenger vehicle is enabled with MEC host, which allows in-vehicle applications to collaborate with applications and services running in a Telco MEC platform located at the eNB/RSU. However, "MEC Host" onboard passenger vehicles, are limited in computing capability, storage availability, power, and connectivity resources, compared to the Telco MEC platform. Resource limitation may impact vehicular applications, which may consume and produce enormous amount of data.

Applications like "Distraction free Waze" [i.9], performs real-time data collection for road monitoring without a human in the loop. Such service focus on detecting road hazards (e.g. potholes) or general road health condition. The main advantage of in-vehicle edge computing for such application is that it can process media rich sensor data, such as video feeds, information from on-board radar, etc. As a result, there is a great savings in uplink bandwidth due to transmission of only filtered non-repetitive data. Reports are sent to edge applications running in Telco MEC platform or in the distant cloud, along with co-ordinates, only when a hazard, which was not previously reported, is detected. The in-vehicle compute platform performs image/sensor processing, object detection and sends the information to the MEC host in the eNB/RSU.

High-mobility infotainment and automotive content streaming would require strong network uniformity (in terms of coverage, throughput, capacity, and reliability) with increased capacity to handle growing bitrate requirements [i.20]. In the case of live streaming a sporting event to an end user riding in the car with an enriched experience, the network is required to provide very dense coverage and high capacity. As vehicles maybe moving at a very high speed and travel to areas where network coverage may not be uniform, it will be difficult for infotainment applications to maintain the desired level of QOE. In-vehicle MEC host may be used to mitigate this situation and maintain the desired QOE for infotainment applications [i.21].

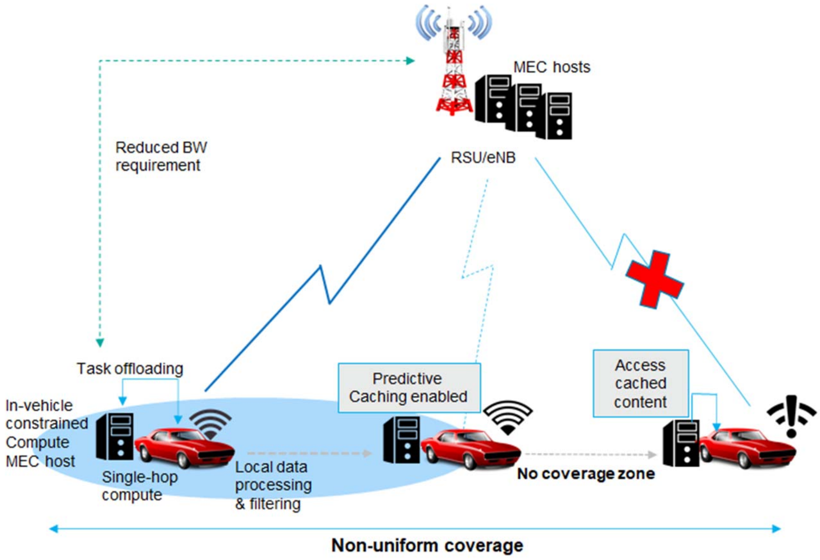

Figure 5.4.1-1: In vehicle infotainment

In-vehicle infotainment service includes live video streaming, accessing internet radio stations and subscription based streaming services, advanced gaming with AR/VR. In-vehicle MEC applications, like "Predictive In-vehicle Caching" of data, take advantage of local compute and storage to reduce the impact of connectivity loss and non-uniform coverage. The "Predictive In-vehicle Caching" application can predetermine no coverage zones the vehicle will pass through and accordingly cache the required amount of content in the car when it is in coverage zone. Some part of the application, which needs immediate processing and response, is made available in the in-vehicle MEC platform.

Users inside vehicle may be offered a selection of applications. Users may choose to run a single application or a combination of few. These applications may be deployed on demand in the In-vehicle MEC host. As number of applications increases, due to limited compute, storage, and power resources, few applications may need to be offloaded/swapped in and out between the In-vehicle MEC host and Telco Edge. While applications are swapped in and out from the in-vehicle MEC host, there should be no disruption in the service. In-vehicle MEC platform offers flexibility, allowing users to easily customize applications. This includes providing additional application features for meaningful improvement opportunities. For example, in the "Distraction free Waze" [i.9] application, vehicle can automatically provide updates without human intervention. Users may be allowed to choose and combine services on demand, e.g. user may combine video conferencing application with a white board and calendar application of his/her choice.

5.4.2 Use Of MEC

In Vehicle infotainment applications requires task to be distributed across different hosts in Data centre, Telco Edge and on Vehicles. MEC hosts can be deployed in a vehicle. MEC system can be used to support vehicular MEC applications and MEC Application Groups, deployed on in-vehicle MEC hosts. The In-vehicle MEC hosts can be a Constrained MEC host.

The In-vehicle Constrained MEC hosts can host vehicular infotainment MEC applications and MEC Application Groups, which execute vehicular application tasks and can be managed by MEC reference points, such as:

- Management plane interfaces:
  - Mm5, Mm7 interfaces over 3GPP, Wi-Fi, Ethernet
  - Mp3 interface can be over 3GPP, Wi-Fi, Ethernet or D2D
- User plane interfaces:
  - Between constrained devices over 3GPP, Wi-Fi, Ethernet or D2D technology
  - Between constrained device and Telco Edge over 3GPP, Wi-Fi, Ethernet.

5.4.3 Evaluation

Analysis of the "In Vehicle infotainment" use case led to similar Gaps, Key issues identified in the analysis of use cases #1 and #3.

For example, the following gaps and key issues also apply for the "In Vehicle infotainment" use case:

- Loss of user plane and management plane
- Dynamic addition of constrained MEC hosts
- Managing group of In-vehicle constrained MEC hosts
- MEC Resource allocation and re-distribution
- Transfer or relocation of MEC applications

Solutions have been proposed in clause 6, to address these key issues. The same solutions can be used to address the key issues derived from the "In Vehicle infotainment" use case.

6 Solutions for closing the gaps

6.1 Gap/Key issue #1 - Loss of management plane and User Plane connection of a constrained device

6.1.1 Description

The scenarios in clause 5.1.3 describe MEC App1 running on Constrained MEC Host 1 (CMH1) and MEC App2 is running on Constrained MEC Host 2 (CMH2). It is assumed that MEC App1 has registered with the MEC platform and produces MEC service, which is consumed by MEC App2. Due to mobility, the constrained device, where MEC App1 is hosted, losses connectivity. The CMH1, loses management plane connectivity to MEC system and user plane connectivity between MEC app1 and MEC App2. Losing management plane connectivity implies that the MEP in CMH1 cannot reach the MEC platform manager to report the loss of user plane. Loss of user plane implies MEC App2 running on CMH2 is unable to receive service from MEC App1.

The solutions describe how:

- the loss of user plane is detected and confirmed;
- to find an alternative MEC App1 service instance to support MEC App2, which was consuming the service provided by MEC App1.

6.1.2 Solution proposal #1-1, Detecting loss of user plane

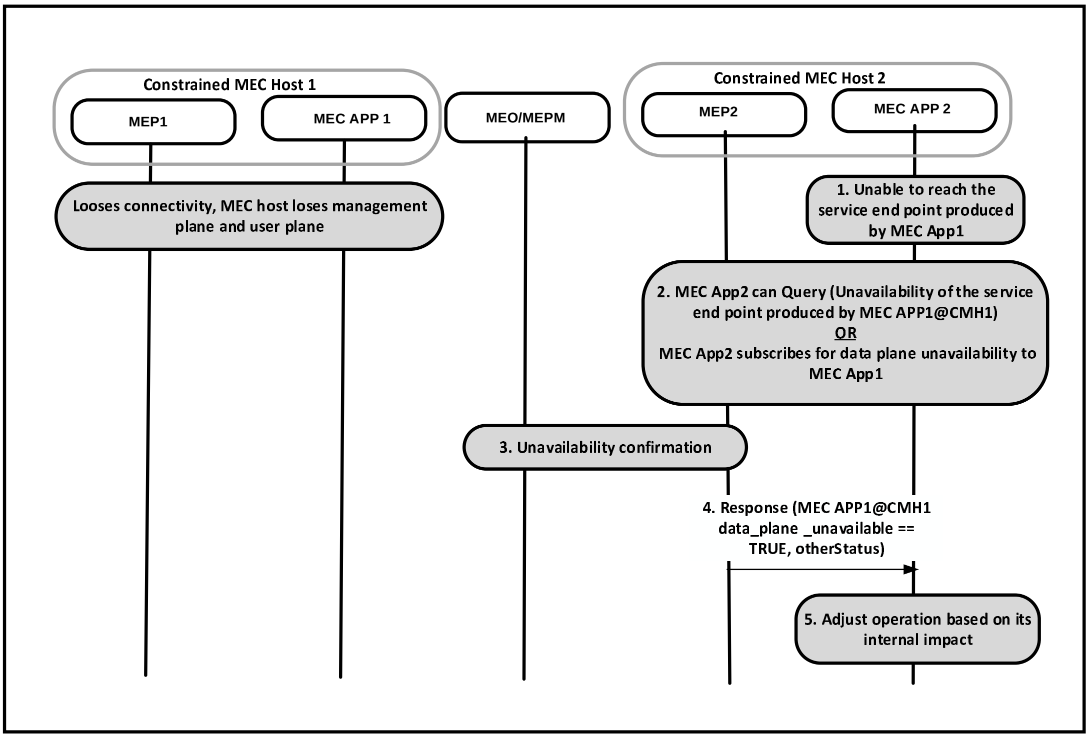

Figure 6.1.2-1: Data plane unavailability detection

Step 1: MEC App2 will not be able to reach the endpoint of the service produced by the MEC App1 instance hosted in CMH1 when the user plane connection between CMH1 and CMH2 is lost.

Step 2: To confirm that the unavailability of the service produced by MEC App1 is due to loss of user plane and not due to other events like, low compute and storage resource, crash of the application instance, etc., MEC App2, as shown in step 2, can:

- query MEP2 for the unavailability of the service produced by MEC App1 hosted on CMH1;
- subscribe to MEP2 for unavailability of data plane to MEC App1 hosted on CMH1.

In Step 3 MEP2 reaches out to the MEC management system (i.e. MEO/MEPM) to confirm the unavailability of the service produced by MEC App1. From the management point of view, loss of management plane with the host CMH1 is assumed to imply that the device is lost and unreachable.

In Step 4, on receiving a confirmation response from MEC management system via the MEPM, MEP2 forwards the response to MEC App2. This can include information confirming the loss of data plane or other status information if it is not really a loss of data plane.

In Step 5 on receiving MEC App1 unavailability information, MEC App2 adjusts its operation based on its internal impact from loss of access to the MEC service and its service dependency:

- Required MEC Service - MEC App2 cannot operate without the service. MEC App2 may transition into an autonomous mode of operation (i.e. limited function specific to MEC App2), while searching for another instance of the MEC service (e.g. via querying the MEP). If an instance of the MEC service cannot be accessed, the MEC application instance may need to be terminated by the MEO.
- Optional MEC service - MEC App2 can continue to operate without access to the MEC service. MEC App2 may decide to query the MEP to access another MEC service instance or may decide to operate without the service. This is MEC application specific and not in MEC system scope.

6.1.3 Solution proposal #1-2, Finding other MEC App to continue service

Instances of a MEC application can be instantiated on multiple constrained devices at specific locations for reliability. These hosts can be static or mobile. If the hosts are static, their information is available with MEC system. If the hosts are mobile, then they can update the MEC system about their availability, location, capability, available MEC Applications etc. In the following diagram, in Step 0 constrained devices update their availability with the MEC system.

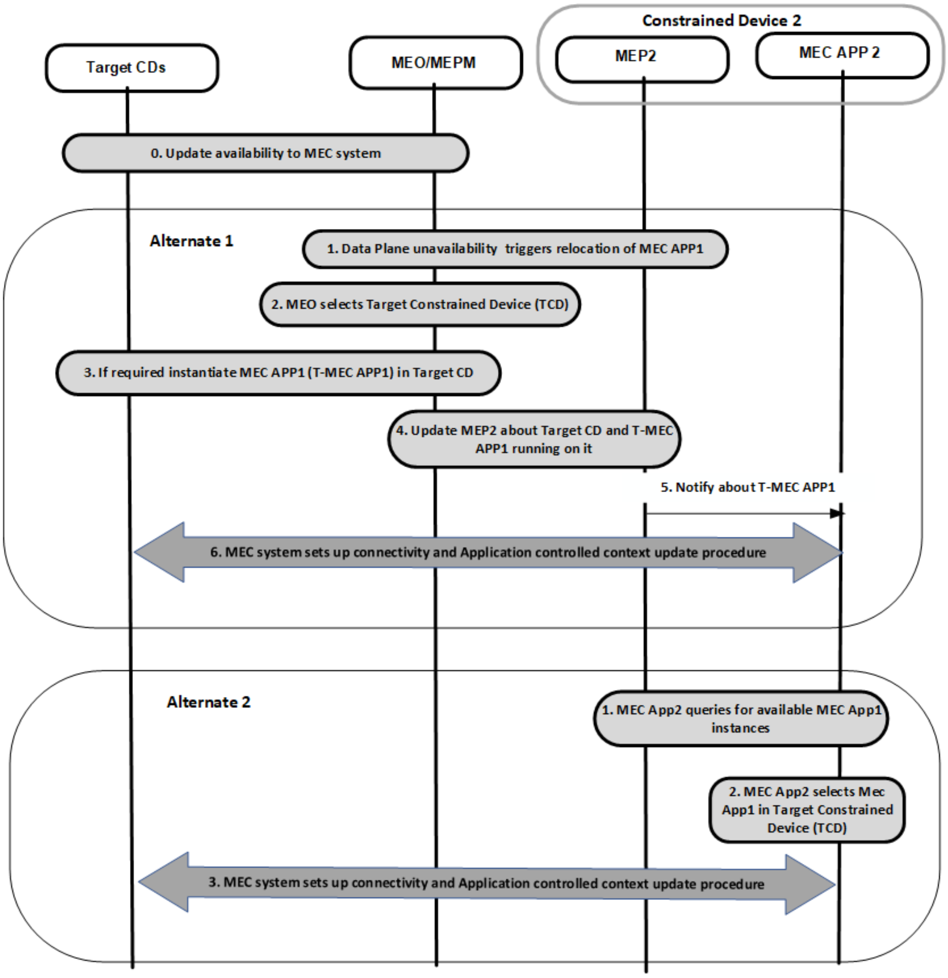

Figure 6.1.3-1: Finding other MEC Applications

Two alternate solutions are described. In Alternative 1, Step 1 user plane unavailability indication triggers a search by MEO for MEC App1 instances, which can produce the same service, based on criteria, such as location, service capability of the constrained device where it is hosted, etc.

In Step 2, when the criteria from step 1 are matched, MEO selects the suitable MEC App1.

If in Step 2 a suitable MEC App1 is not found by MEO, then in Step 3 it can first find a suitable Target Constrained Device (TCD) and instantiate MEC App1 on it. The TCD can be selected by MEO based on Location, Service capability, etc.

In Step 4 MEO updates MEP2 via the MEPM about the TCD hosting the target MEC App1. The update can include information about the MEC App1 such as URL, IP address of TCD, etc.

In Step 5 MEP2 forwards the target MEC App1 availability and reachability information, such as URL, IP address to MEC App2.

In Step 6 MEC system sets up the connectivity between the target MEC App1 and MEC App2.

In Alternative 2, Step 1, MEC App2 can query for availability of service produced by MEC App1 in the proximity and able to serve. MEC system provides a list of available instances. In Step 2, MEC App2 uses the information to select a specific instance. MEC App2 connects to the instance of MEC App1. In Step 3, connectivity is setup between MEC App1 and MEC App2.

After the connection between MEC App1 and MEC App2 is setup, for stateful applications an application-controlled context update procedure or MEC assisted context update can be executed between MEC App1 and MEC App2 [i.11]. Once application context is synchronized between MEC App1 and MEC App2, service can resume.

6.1.4 Evaluation

The solution proposal #1-1 is technically feasible only if the following conditions are met:

- MEO is capable of informing MEP that a MEC App is unavailable due to loss of user plane.
- MEP is capable of notifying MEC App that service unavailability is due to loss of user plane.

The solution proposal #1-2 is technically feasible only if the following conditions are met:

- Mobile constrained MEC hosts can update availability to MEO with location and capability details.
- MEO can detect the loss of service is due to loss of user plane and act to:
  - Find a constrained MEC host with same capability as the lost constrained MEC host.
  - Deploy an instance of the MEC application on the selected constrained MEC host.
  - Inform MEP about availability of a MEC service instance, which has same capability as the MEC service which lost user plane connectivity.

MEO is capable to inform MEC App, that an alternate instance of the MEC service, hosted on a constrained MEC host with same capability, is available for continuing the service.

6.2 Gap/Key issue #2 - Support for dynamic addition of MEC hosts (to a MEC system)

6.2.1 Description

As described in clause 5.1.3.4, MEC applications deployed in the Telco Edge and in the Far Edge (i.e. on constrained MEC hosts) help to control the overall operation and monitoring of production lines in a smart factory. The mobility and location of constrained MEC hosts within the smart factory impacts their ability to participate and serve in specific application verticals (e.g. monitoring a specific production line). When a constrained MEC host can no longer serve an application vertical, another constrained MEC host needs to be dynamically added to the MEC System.

Solutions to support dynamic addition of MEC hosts include:

- Constrained MEC Host Description and Capabilities due to limited resources (Solution proposal #2-1 - clause 6.2.2)
- Dynamic MEC Host Capability Exposure Options (Solution proposal #2-2 - clause 6.2.3)
- Configuration of MEC Host Capability in a MEC System by authorized user (Solution proposal #2-3 - clause 6.2.4)
- Dynamic MEC Host Capability Registration with a MEC System, initiated by MEO (Solution proposal #2-4 - clause 6.2.5)
- Dynamic MEC Host Capability Registration with a MEC System, initiated by Constrained MEC Host (Solution proposal #2-5 - clause 6.2.6).

6.2.2 Solution proposal #2-1, Constrained MEC Host

As specified in ETSI GS MEC 003 [i.1], a MEC host is an entity that contains the MEC Platform (MEP) and a virtualization infrastructure which provides computing, storage, and network resources for MEC applications. MEC applications are deployed on MEC hosts by the MEO utilizing the MEC host level management functions of the MEPM and VIM. A constrained device may not be able to support all of the entities required of a MEC host due to its limited resources (computing, storage, etc.).

Due to their fundamental nature, MEC applications are virtualized software entities (e.g. VMs, containers), requiring a virtualization infrastructure to run. Some virtualization environments, such as hypervisors and VMs, may not be supported on constrained devices due to their resource requirements (memory, CPU). However, lightweight virtualization techniques, such as containers, are proven to successfully operate in constrained devices (e.g. IoT devices & gateways, small-scale computing devices like Raspberry Pi, etc.). As such, a constrained device can support the virtualization infrastructure functional entity required for a MEC host, depending on the device's capability. If a device cannot support any virtualization infrastructure, the device cannot host MEC applications. Such an ultra-constrained device can only be considered as a client device that utilizes MEC applications in the MEC system and cannot serve as a MEC host.

In addition to the virtualization infrastructure, a MEC host provides a MEC Platform which provides MEC application enablement capabilities, for example to advertise and discover MEC services. However, due to resource limitations, a constrained device may not be able to support local, on-device deployment of the MEP. In such a case, an "off-device" MEP may be utilized to provide Mp1 services to MEC applications on the constrained device. The "off-device" MEP may be located in the Telco Edge as a stand-alone entity or utilized from another fully functional MEC Host. An MEP Proxy may be used to facilitate communication between MEC applications and VI on the constrained device towards the "off-device" MEP.

A Constrained MEC Host is defined as an entity that hosts MEC applications on a virtualization infrastructure, while providing MEC platform services to MEC applications from a remote, off-device MEP. See Figure 6.2.2-1. A Constrained MEC Host may also be mobile, requiring the capability to dynamically connect to the MEC system.

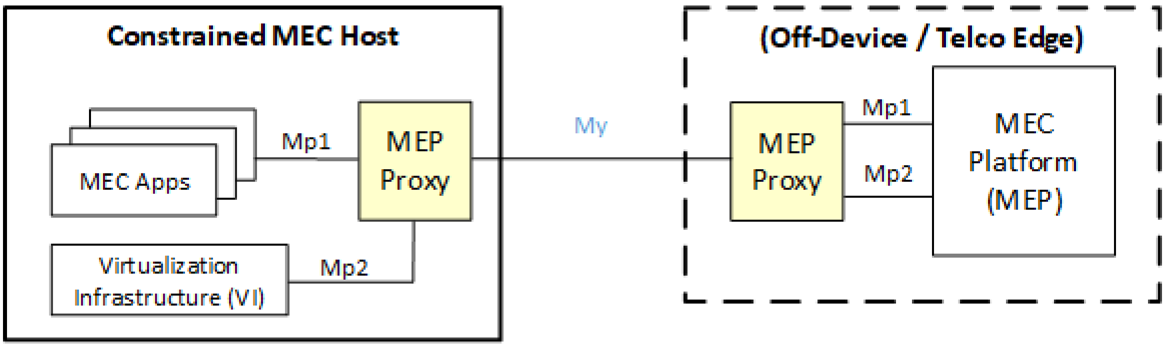

Figure 6.2.2-1: Constrained MEC Host and Off-Device MEP Platform

The purpose of MEP proxy function, in constrained MEC host and Off device MEP, is to forward and receive Mp1 messages between MEC applications and "off device MEPs".

To enable forwarding of Mp1 messages between MEP proxies, several deployment options and implementation considerations are described. A general-purpose interface configured at deployment time, shown as My (MEP Proxy interface) in Figure 6.2.2-1, can be used to forward Mp1 messages.

As one deployment option, the following diagram describes, MEP proxy function exposing Mp1 interface from one Off device MEP to MEC Applications (one or more) in the constrained MEC host.

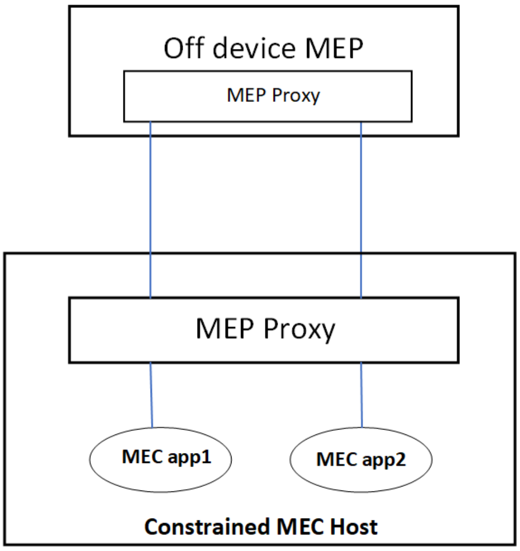

Figure 6.2.2-2: MEP proxy interacting with one Off device MEP proxy

As another deployment option, constrained MEC host, which hosts the MEP Proxy function, can support interaction with one or more instances of "off-device" MEP. Possible scenarios:

1) MEC app1 registering with Off device MEP1

2) MEC app2 registering with Off device MEP2

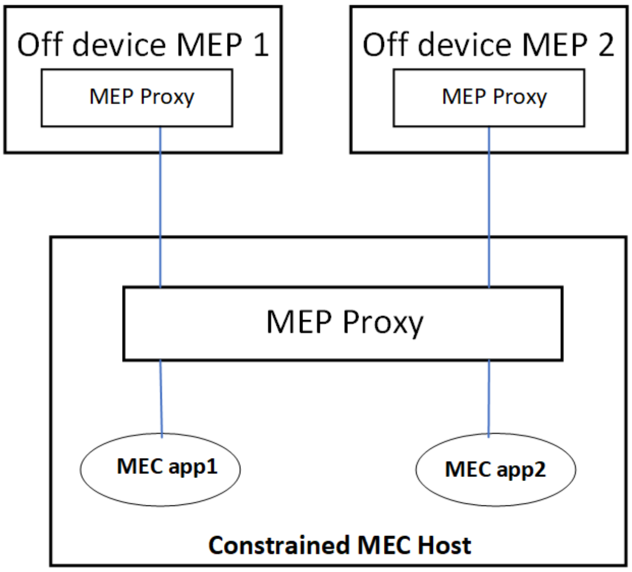

Figure 6.2.2-3: MEP proxy interacting with more than one Off device MEP proxy

To facilitate the communication between MEC applications on the constrained device towards the "off-device" MEP, the MEP proxy in the constrained MEC host needs information about:

- Which off device MEP proxy function the communication should be forwarded.
- Once it is setup, continue forwarding the communication from the same MEC app to the same off device MEP proxy.
- When receiving communication from Off device MEP, which MEC application it should be forwarded to.

Similarly, for the off-device MEP proxy, which is receiving the communication from MEP proxies in one or more constrained MEC hosts, needs information about:

- Which MEC application and MEP proxy in a Constrained MEC host, is sending the communication.

- While responding, which MEC application and MEP proxy in a constrained MEC host among many constrained MEC hosts, the communication can be sent.

MEP proxy in the constrained MEC host can be configured with the IP address of the Off device MEP towards which Mp1 messages are forwarded. This configuration can be done by the service provider, constrained MEC host owner during deployment.

Alternatively, Mp1 and Mp2 interface can be tunnelled and terminated at the MEP proxy. MEP proxy can be configured with the tunnel identifier, which can be used to forward Mp1 messages, at the time of deployment.

If MEP proxy can forward to more than one "Off device MEP" instances, then it can be configured with the information about the MEC application and the corresponding Off device MEP IP address or Tunnel identifier to be used for forwarding Mp1 messages.

The "Off device MEP" and the constrained MEC host may be provided by a single vendor or different vendors. The MEP proxy can be configured at run time, after being deployed by different vendors. A known configuration server can be used by MEP proxy to obtain information about Off device MEP proxy and how to reach it.

Figure 6.2.2-4 describes run time configuration of MEP proxy.

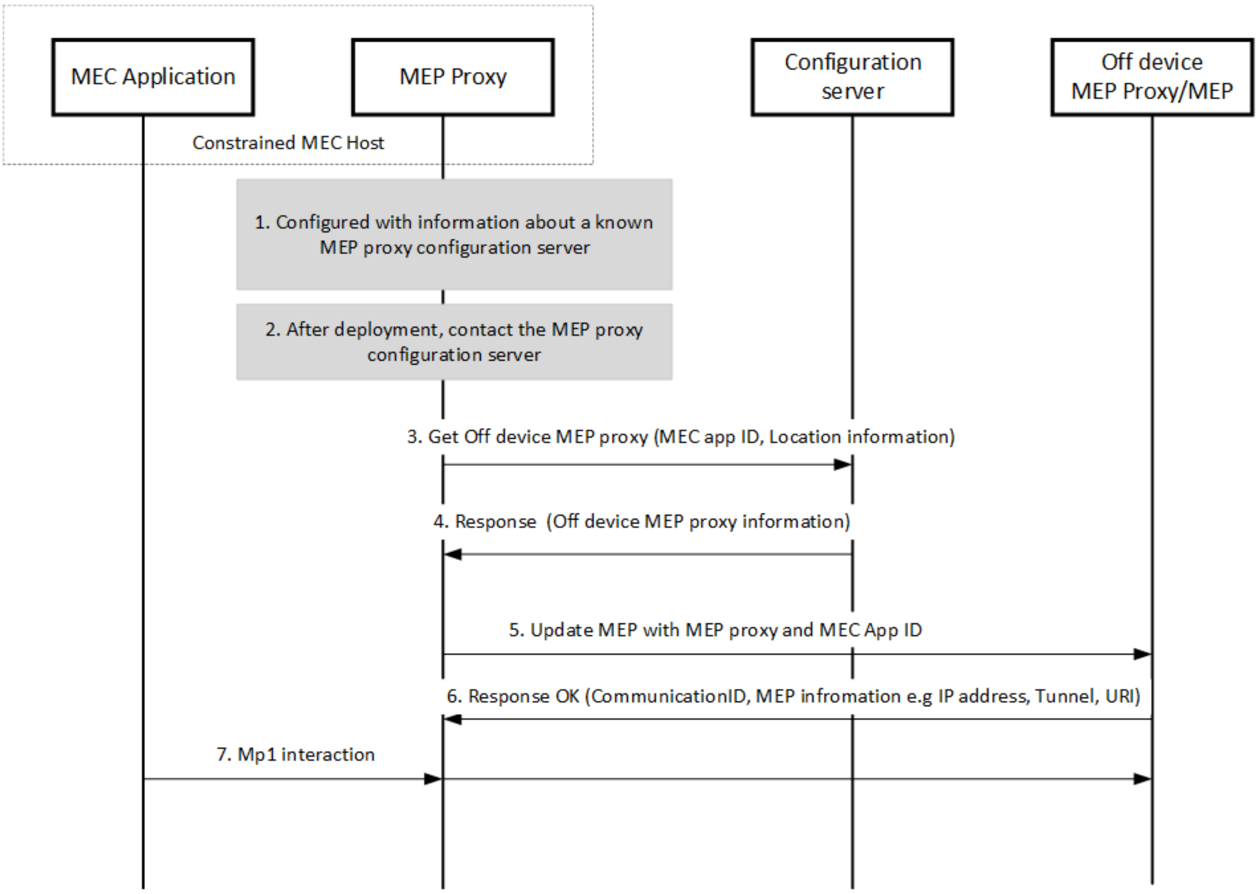

Figure 6.2.2-4: Configuration option of MEP proxy

Step1: MEP proxy is configured with a known Configuration server information.

Step 2: After deployment, MEP proxy obtains the Configuration server information and initiates communication with it.

Step 3: MEP Proxy requests the Configuration server to obtain information about a suitable Off device MEP proxy by providing MEC applications information hosted in the constrained MEC host and location information.

Step 4: The configuration server based on MEC application information and Location information identifies a suitable Off device MEP and provides the details of Off device MEP in the response message.

Step 5: MEP proxy contacts the Off-device MEP proxy and provides its own ID and hosted MEC application information.

Step 6: The Off-device MEP proxy updates MEP with MEP proxy ID and MEC App ID, and generates a communication identifier, which identifies MEC application, MEP proxy and the Off device MEP. The communication identifier and the Off-device MEP information like IP address, URL of MEP or tunnel ID is sent back to MEP proxy.

Step 7: MEP proxy starts forwarding the MP1 messages using the communication identifier and configured Off device MEP information.

6.2.3 Solution proposal #2-2, Dynamic MEC Host Capability Exposure Options

Constrained MEC Hosts may expose different levels of MEC capability options towards the MEC system. These may depend on the host's overall capabilities and what level of capability the host selects to provide to the MEC system. Some Constrained MEC hosts may be extremely constrained, only offering the ability to host MEC applications and do not include any MEC platform services. While other Constrained MEC Hosts may be more capable, such as including support for the MEC Platform. Additionally, some Constrained MEC Hosts may also offer on-device MEC host level management functions (MEPM and VIM).

The range of capabilities that a Constrained MEC Host offers to MEC system can be expressed by which and how many of the MEC functional elements and reference points the host exposes to a MEC system. As shown in Figure 6.2.3-1, these may include:

- **Constrained MEC Host (MEC Application Hosting only):** This MEC host provides limited capability. It can host MEC applications and only offers virtualization infrastructure. The device cannot support any MEC Platform capability. In this option, the constrained MEC host offers the Mm7 interface to a VIM in the MEC System. The Constrained MEC Host also offers the new MpX interface for remote MEP communication, same as My interface, as shown in Figure 6.2.2-1. The constrained MEC host being mobile, availability of Mm7 and MpX interface can be impacted.

- **Constrained MEC Host, with MEC Platform:** In addition to virtualization infrastructure, it hosts the MEC Platform (MEP) and offers Mp1 capabilities to MEC applications on the device. This device is a fully functional MEC host in the MEC system. However, it is mobile and may connect and disconnect from the MEC system dynamically. In this option, the Constrained MEC Host offers the Mm7 towards a VIM in the MEC system. Additionally, it offers the Mm5 interface towards a MEC Platform Manager and the Mp3 interface towards other MEC hosts, if needed.

- **Constrained MEC Host, with Host Management (i.e. MEC Platform, MEC Platform Manager, and VIM):** This option is for a more capable host, which needs to isolate its internal MEC host management functions. Here, the Constrained MEC Host includes the MEC Platform, the MEC Platform Manager, and the VIM. This type of host offers the Mm2, Mm3, and Mm4 interfaces towards the MEC system. The Mp3 interface towards other MEC hosts may also be offered, if needed.

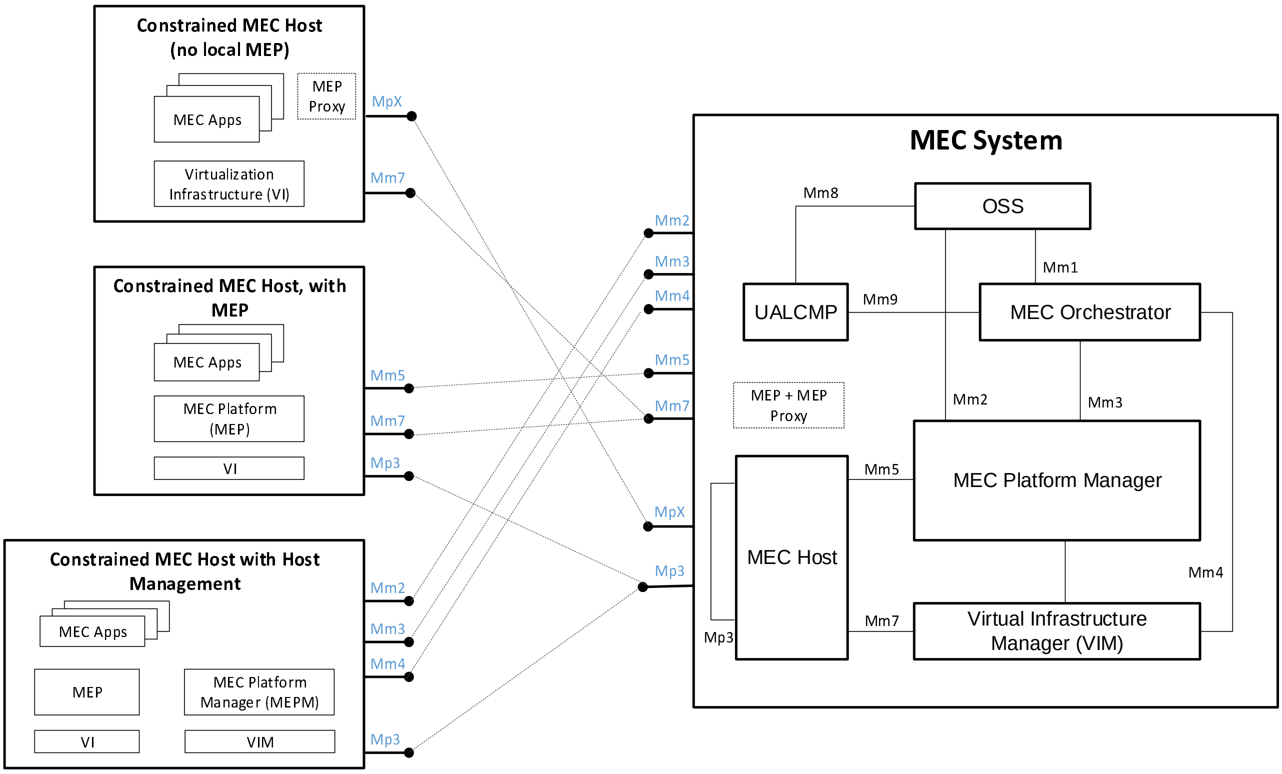

Figure 6.2.3-1: Dynamic MEC Host Capability Exposure Options

6.2.4 Solution proposal #2-3, Configuration of MEC Host Capability in a MEC System by authorized user

This solution proposes a method for an authorized user, system administrator to inform the MEC system (e.g. MEO via an OSS) about a new Constrained MEC host and configures the MEO with Constrained MEC host information.

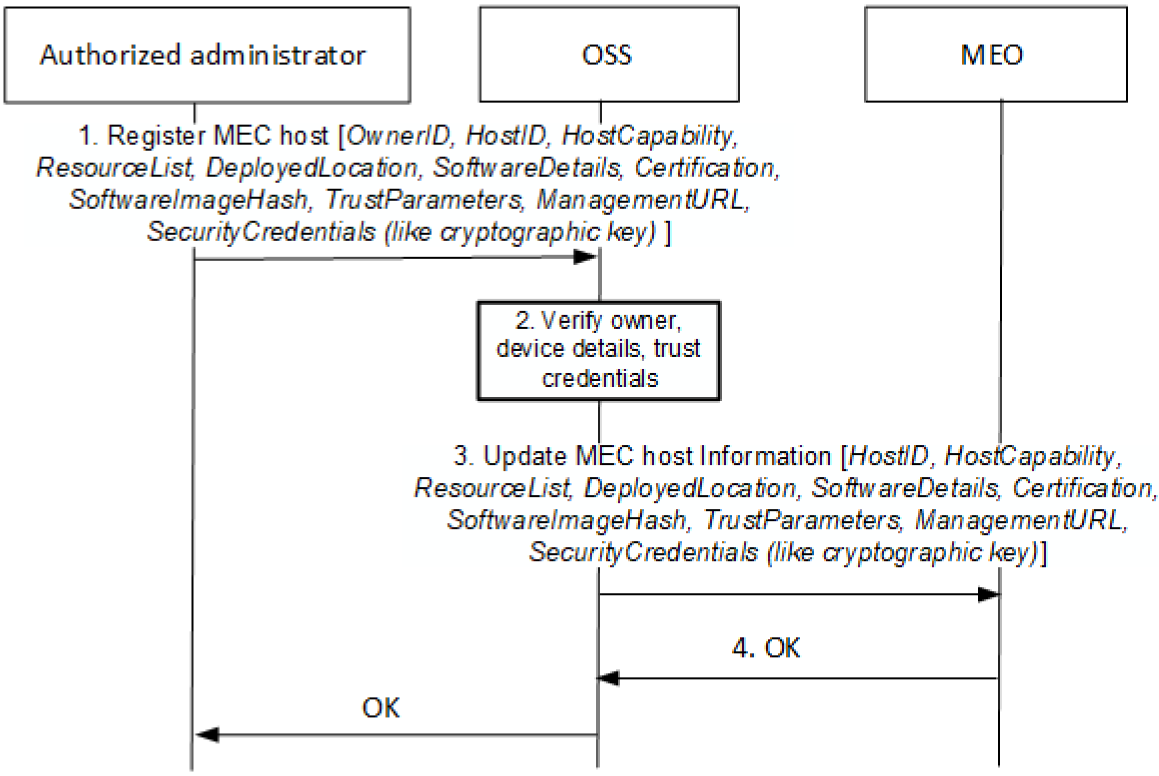

Figure 6.2.4-1: Configuration of MEO by authorized user

MEO maintains information about all MEC hosts in the MEC system, i.e. Telco Edge and Constrained MEC Hosts (CMH). MEO can be accessed through OSS by authorized users to add configuration information about constrained MEC hosts.

MEO can implement a host management function to manage information about MEC hosts (Telco Edge hosts and CMH). The details of the host management function is out of scope for the present document.

In step 1, an Authorized user, administrator, or owner of a constrained MEC host, provides details of the constrained MEC host to OSS.

The information about a constrained host details can include:

- Identity of the owner of the constrained MEC host, so that OSS can validate and authorize the request.
- Host ID, which can be used by MEO to identify and authenticate a host. The host ID can be a secure ID, which cannot be tampered or changed. It can also be a secure ID assigned by service provider, such as IMEI number.
- Host capability, such as if this is a camera, robot, vehicle, drone for air surveillance, etc. If it is a camera, what capability it has for vision processing, etc. This information can be used by MEO to determine if the host can be added to support a specific application service. E.g. in the Industry 4.0 use case, a production line is managed on the factory floor by deploying mobile camera and robots. A mobile robot, which was not serving the production line earlier, appears in proximity to the production line. MEO, with the knowledge that the constrained MEC host is a robot, can add it to monitor the production line.
- Resource list, which provides information about compute, storage and power available in the constrained MEC host. This information can be used by MEC system to determine if the host can be added to support an application service.
- Deployed location, if the constrained MEC host is deployed in a stationary (i.e. non-mobile) mode, it can include the exact location where the constrained MEC host is deployed. But if the constrained MEC host is mobile, then location information can be a wider area where the constrained MEC host is supposed to provide application service, such as, a civic address, geo-location information with a radius or other shape data defining an area, a mobility path that includes geo-location points with time of arrival, departure, speed, etc. or indoor location information (floor #, room # or area, indoor coordinates, etc.). For mobile constrained MEC hosts, the location information can be used by MEO to receive a notification when the constrained host becomes available in a specific location within the area. MEO can use the notification to initiate a procedure to add the constrained MEC host.
- Software details such as list of software (including MEC applications, MEC services, etc.), version, available in the host. It can also include a hash of the software image in the host. MEO can verify the software details and the software image hash to verify that the software in the host has not been tampered. If the software is not tampered, then MEO can add the host.
- Trust parameters and certificates, security credentials, keys about the constrained host, which MEC system can verify with the host, when adding the host to the MEC system.
- A secured management URL, which can be used by MEO to make initial contacts with the constrained MEC host, while adding the host to MEC system. This can be a new reference point or Mm5 can be re-used.
- The information can optionally include MEC host integration options supported by the constrained MEC host. MEC host integration option can be:
  - Constrained MEC Host (with MEC Application Hosting only)
  - Constrained MEC Host, with MEC Platform
  - Constrained MEC Host, with Host Management (i.e. MEC Platform, MEC Platform Manager, and VIM)

In step 2, the OSS can verify the user, administrator creating the request. It can also verify or check integrity of the information provided in the request. OSS can verify the capability, resources, software and hardware details, certificates, and security credentials of the constrained MEC host meets the MEC system requirements. If so, OSS forwards the request to MEO otherwise it can reject the request.

In step 3, OSS forwards the request from authorized users, constrained MEC host owners to MEO with the information received in step 1. OSS indicates that the constrained MEC host information can be added to the MEC system. OSS also indicates that this is a constrained MEC host, mobile or stationary and can be added by MEO dynamically, when MEO decides the constrained MEC host is required to support a specific deployment and an application or service in a factory, location, building etc. OSS can also indicate that the MEO report back the usage of the constrained MEC host, when it is added, what service it provides, etc. for charging purpose. It can also indicate the MEO to provide the status of the host such as which MEC system it is serving, which application being supported.

In step 4, MEO updates information about valid and authorized constrained MEC host, which can be added dynamically to the MEC system at a later time when it is needed. While updating its records, MEO can verify if the information received matches the MEC system requirement criteria. MEO sends an OK response to OSS to indicate that the database has been updated successfully. At that point OSS can also update the constrained MEC host owner, authorized user that the constrained MEC host information has been updated successfully.

6.2.5 Solution proposal #2-4, Dynamic MEC Host Capability Registration with a MEC System, initiated by MEO

This solution describes how MEO can initiate capability registration and integration of a mobile constrained MEC host into the MEC system.

Prerequisites: Before MEO can initiate the capability registration procedure, the following conditions should be met:

1) MEP (or the proxy function) is pre-installed on the CMH.

2) Constrained MEC Host information to be configured in the MEC system with the MEO by an authorized user, as described in solution #2-3.

3) Constrained MEC Host to be authorized to connect to the MEC system with security credentials (e.g. OAuth 2.0).

4) MEO has subscribed to obtain location information of Constrained MEC Host.

MEO initiates the procedure when the mobile constrained MEC host is available in a desired location and can support an application. MEO contacts the constrained MEC host to authenticate it and then requests its integration capability. The constrained MEC host responds with its supported integration capability. The integration capability received from constrained MEC host can be same as configured by the authorized user. Otherwise, the constrained MEC host can select one integration option from a list of other supported options. MEO can select from the host's capability exposure options and registers the Constrained MEC Host. Once registered, the MEC system can manage the Constrained MEC host via its exposed capabilities.

Several interface options may be utilized for the MEO to contact constrained MEC Host:

- Over Mm3 and Mm5
- A new interface for initial communication using the configured management URL

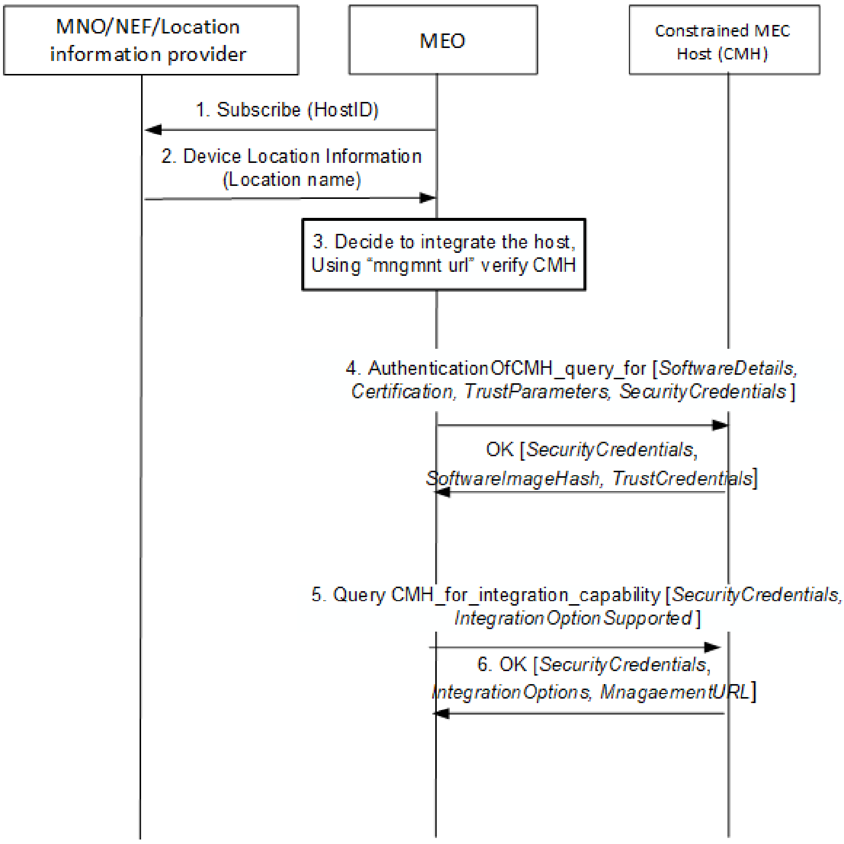

Figure 6.2.5-1: MEO initiated constrained MEC host capability registration.

In step 1, MEO subscribes with MNO (e.g. through NEF), via an Edge Enablement Layer like 3GPP EDGEAPP, or other Telco API, or Location service provider (e.g. via the MEC-012 Location service provider), to be notified about a Constrained MEC host being available in a specific location. MEO provides the Device ID and the area or location of interest. The area or location of interest is the location where MEO wants the constrained MEC host to support an application and provide the desired service.

In step 2, when the constrained MEC host is available in the desired area, MEO is notified by MNO, Location service provider about the host's availability in that specific location. The notification indicates the Host ID, which identifies the constrained MEC host. The notification can also include precise location of the constrained MEC host. It can indicate indoor locations such as floor, block, segment, etc. The precise location of the constrained MEC host will allow MEO to decide how to integrate and add the constrained MEC host to the MEC system.

In step 3, MEO, after receiving the Host ID of a constrained MEC host and the precise location where it is available, verifies the host's capability to support an application service, such as if it is a camera and can support vision analysis. MEO also determines resource availability of the constrained MEC host, if it has enough compute, storage, etc. to support the application service. MEO determines that the constrained MEC host can support the application service in a specific location, it decides to integrate the host into the MEC system. But before MEO can integrate the constrained MEC host, it needs to verify and authenticate the constrained MEC host. For verification and authentication, MEO selects the associated management URL (provided for the CMH in clause 6.2.4, step 1) of the constrained MEC host, supplied by authorized user while configuring MEO with constrained MEC host information.

In step 4, MEO uses the management URL provided by the user to contact the constrained MEC host to validate and authenticate the MEC host. MEO requests the constrained MEC host to send its security credentials to validate and authenticate the host. MEO also requests certificates and trust parameters to verify that the host has been not tampered. For example, MEO requests for software image hash to verify that the software entities available in the host are same as described by the user and has not been compromised. The constrained MEC host provides the requested information to MEO. MEO can verify the information received from constrained MEC host against the information provided by authorized user and OSS. If MEO is satisfied with the information and determines that the constrained MEC host is valid and not compromised, it proceeds further with this integration procedure. Otherwise MEO can abandon the process.

In step 5, MEO can determine the integration option supported by CMH from the information provided in clause 6.2.4, step 1. If the information is not available, MEO uses the management URL provided by the user to contact the constrained MEC host to request for which integration option the constrained MEC host supports. The integration option supported by the constrained MEC host indicates what all MEC interfaces it supports. Constrained MEC host integration option can be:

- Constrained MEC Host (with MEC Application Hosting only).
- Constrained MEC Host, with MEC Platform.
- Constrained MEC Host, with Host Management (i.e. MEC Platform, MEC Platform Manager, and VIM).

In step 6, Constrained MEC Host responds with the integration options it supports including the interfaces it supports. These interfaces are the typical MEC management interfaces, accessible by MEO. MEO uses these MEC management interfaces to manage and orchestrate the constrained MEC host.

6.2.6 Solution proposal #2-5, Dynamic MEC Host Capability Registration with a MEC System, initiated by Constrained MEC Host

This solution proposes a method for an authorized Constrained MEC Host to dynamically discover a MEC system and inform the MEC system which MEC host capability exposure options it may support. With this information, the MEC system selects constrained MEC host's capability exposure options and registers the Constrained MEC Host. Once registered, the MEC system can manage the Constrained MEC Host via its exposed capabilities.

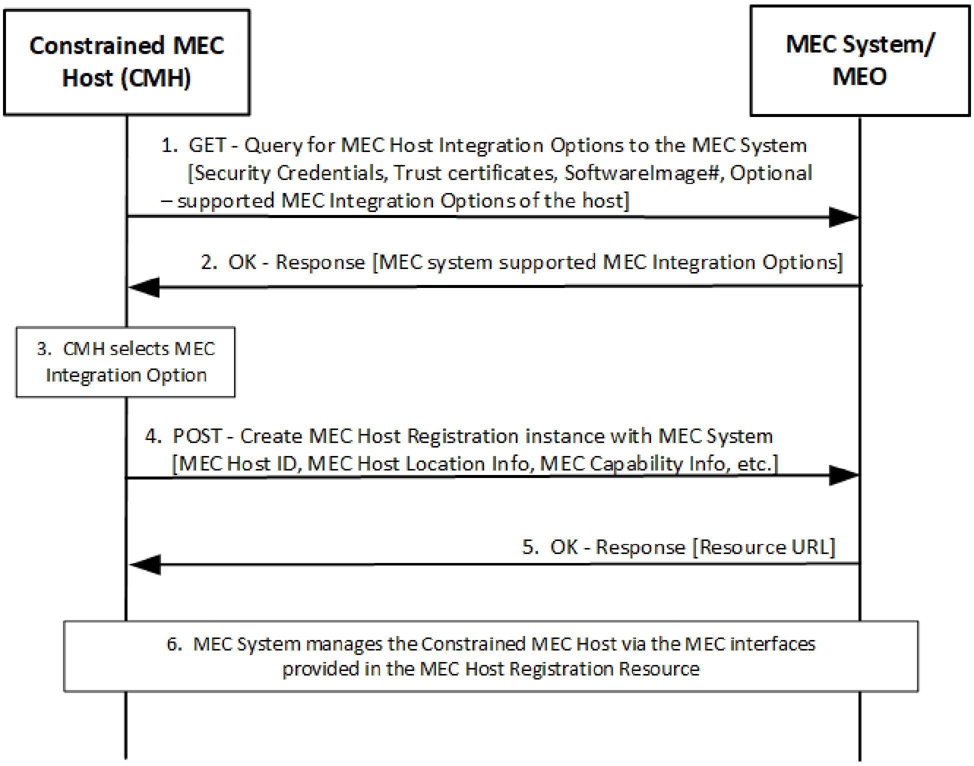

Figure 6.2.6-1: Constrained MEC Host Registration with a MEC System, initiated by the host

Prerequisites: Before dynamically registering with the MEC system, a constrained MEC host needs:

1) Preinstalled MEP (or the proxy function).
2) To be configured in the MEC system with the MEO, as described in solution #2-3.
3) To be authorized to connect to the MEC system with security credentials (e.g. OAuth 2.0).
4) To discover the MEC system to communicate (e.g. via a URI, FQDN, IP address, etc.).

Several options may be utilized for the MEC system discovery and authorization, including:

- Configured by a user (e.g. technician or supervisor in a smart factory).
- Configured by a host management system.
- Access network support (e.g. similar methods as: 3GPP Edge Configuration Server discovery, IETF Provisioning Domain configuration, Wi-Fi ANQP, etc.).
- Dynamic MEC Host Registration may be appended to an existing MEC reference point, for example Mx2. Existing MEC reference point discovery methods would be utilized. Dynamic MEC Host registration is currently not supported over Mx2 and can impact MEC UE Application interface.

Step 1: An authorized constrained MEC host queries the MEC system for dynamic MEC host integration options.

The constrained MEC host provides security credentials, software details, hash of software image, trust parameters, certificates to MEO with the request. This set of information should match, what the authorized user, administrator provided to MEC system in MEO configuration procedure. MEO will validate the request based on what was configured in MEO configuration procedure. MEO validates the request based on the security credentials.

The constrained MEC host may include optional information to assist the MEC system in determining appropriate dynamic MEC host integration options for the specific host, including: MEC host ID, MEC host locations, and supported dynamic MEC host integration options such as: Constrained MEC Host, Constrained MEC Host with MEP, etc. MEC system selects the dynamic host integration option(s) to be utilized based on its internal criteria.

Step 2: MEC system responds to the constrained MEC host query with information on the dynamic MEC host integration option(s) that the MEC system selects for the request. Dynamic MEC host integration options include a list of offered integration types:

1) Dynamic MEC Host Registration Not Supported (i.e. the constrained MEC host cannot register with the MEC system).
2) Constrained MEC Host.
3) Constrained MEC Host with MEP.
4) Constrained MEC Host with Host Management.

Step 3: The constrained MEC host evaluates its compatibility with the dynamic MEC host integration option(s) offered by the MEC System. If the constrained MEC host does not find any compatible dynamic MEC host integration options with its capabilities, this procedure ends. Otherwise, the constrained MEC host registers its MEC host capabilities with the MEC system in the next step.

Step 4: The constrained MEC host issues request to create a Dynamic MEC Host Registration instance with the MEC system. The constrained MEC host issues a post to create the resource including its dynamic MEC host capability exposure information. This information includes:

- Host ID.
- MEC Host Location Information, including anticipated mobility path or trajectory information.
- Dynamic MEC Host Integration option (e.g. selected from a list provided by the MEC system in step 2).
- MEC Interfaces - interface endpoint information:
    - Interface type (Mm1, Mm2, etc.).
    - Endpoint address or URI.
    - Endpoint authentication information.
- MEC Host Features:
    - Memory
    - Storage

  - Bandwidth
  - Latency
  - Hardware type
  - # of CPUs
  - Virtualization type: for example, containers
  - HW acceleration: supported FPGA, supported GPUs, etc.

- MEC Host Registration Validity Time - registration expires after this time.

Step 5: The MEC system checks the registration information from the constrained MEC host. If all parameters are valid and compatible with the MEC system, the MEC system accepts the registration. The MEC system creates a dynamic MEC host registration resource and returns its URL to the constrained MEC host.

Step 6: The MEC system manages the constrained MEC host (e.g. instantiating a MEC application on the constrained MEC host) using the interfaces exposed by the MEC host in its dynamic MEC host registration resource.

6.2.7 Evaluation

The solution proposal #2-1 describes a Constrained MEC Host that hosts MEC applications on a virtualization infrastructure, while providing MEC platform services to MEC applications from a remote, off-device MEP. It is technically feasible if the following conditions are met:

- It is possible to setup an interface between MEP proxy function exposing Mp1 interface from one Off device MEP to MEC Applications (one or more) in the constrained MEC host.
- The interface is used for forwarding Mp1 messages.

The solution proposal #2-2 lists the options for the Constrained MEC Hosts to expose different levels of MEC capability options towards the MEC system. This is the foundation for the solution proposals #2-3, #2-4 and #2-5.

The solution proposal #2-3 allows an authorized user, system administrator to inform the MEC system (e.g. MEO via an OSS) about a new Constrained MEC host and configures the MEO with Constrained MEC host information. The solution is technically feasible if the following conditions are met:

- MEO maintains information about all MEC hosts in the MEC system, i.e. Telco Edge and Constrained MEC hosts (CMH).
- MEO can be accessed through OSS by authorized users to add configuration information about constrained MEC hosts.
- MEO can implement a "host management function" to manage information about MEC hosts (Telco Edge hosts and CMH).
- OSS can verify the authorized user, system administrator creating the request. It can also verify or check integrity of the information provided in the request.

The solution proposal #2-4 allows MEO to initiate capability registration and integration of a mobile constrained MEC host into the MEC system. The solution is technically feasible if the following conditions are met:

- MEO can contact the constrained MEC host to authenticate it and, after successful authentication, requests the constrained MEC host's integration capability information (e.g. as proposed in solution proposal #2-2).
- Constrained MEC host can respond to MEO with its supported integration capability.
- MEO can use the host management function (proposed in solution #2-3) to add, update and retrieve information about constrained MEC hosts.

The solution proposal #2-5 allows an authorized Constrained MEC Host to dynamically discover a MEC system and inform the MEC system which MEC host capability exposure options it may support. The solution is technically feasible if the following conditions are met:

- An authorized constrained MEC host can contact the MEO to authenticate itself and provide the MEO with its CMH integration capability information (e.g. as proposed in solution proposal #2-2).
- MEO can select a constrained MEC host's capability exposure options and registers the constrained MEC host with in the MEC system.
- MEC system can manage the Constrained MEC Host via its exposed capabilities.

6.3 Gap/Key issue #3 - Support for a group of MEC application instances deployed on specific set of hosts

6.3.1 Description

In the smart factory, MEC applications (i.e. federated learning agents) in constrained MEC hosts interact with MEC applications (e.g. federated learning agent or server) in other Constrained MEC Hosts or Telco Edge. One or more Constrained MEC Hosts and Telco Edge MEC Host, hosting MEC applications, are assigned to execute an application task, such as "production line monitoring".

In the factory floor, the constrained MEC hosts are the "training hosts" running "training applications". These training applications on training hosts interact with FL server in the Telco Edge as well as among themselves. For every iteration of the training sequence a fixed set of training hosts with training applications are selected. This set should remain fixed, as long as possible, for better convergence of learning.

A group, which is a set of MEC application instances (can be instance of same or different MEC application) running on specific set of hosts (Telco Edge and CMH), can be defined by MEO. In the group, one instance of a MEC application runs on one specific host. E.g. assuming there are three hosts, Host1, Host2 and Host3, and seven instances from 3 different applications, namely, Inst1a, Inst2a, Inst2b, Inst3a, Inst3b, Inst3c, Inst3d. MEO can decide, Inst1a runs on Host1, Inst2a runs on Host2, Inst2b runs on Host3 and inform the application instances, they can only interact among these set or group i.e. Inst1a, Inst2a, Inst2b. In another example, MEO decides.

- Inst1a runs on Host1, Inst2a runs on Host1
- Inst2b runs on Host2, Inst3a runs on Host2
- Inst3b runs on Host3, Inst3c runs on Host3, Inst3d runs on Host3

The application instances can only interact among this set or group.

It is to be noted that the group, consists of two entities i.e. application instance and host, which are not independent. This is a group of two entities, e.g. MEC app instance and Host, coupled with each other. Group can be defined as a set of = {[app1instance, host1], [app2instance, host2], [app3instance, host3]}.

To create and operate a group, multiple solutions are described in clauses 6.3.2, 6.3.3, 6.3.4, 6.3.5 and 6.3.6.

6.3.2 Solution proposal #3-1, MEO supported group creation and configuration of constrained MEC hosts

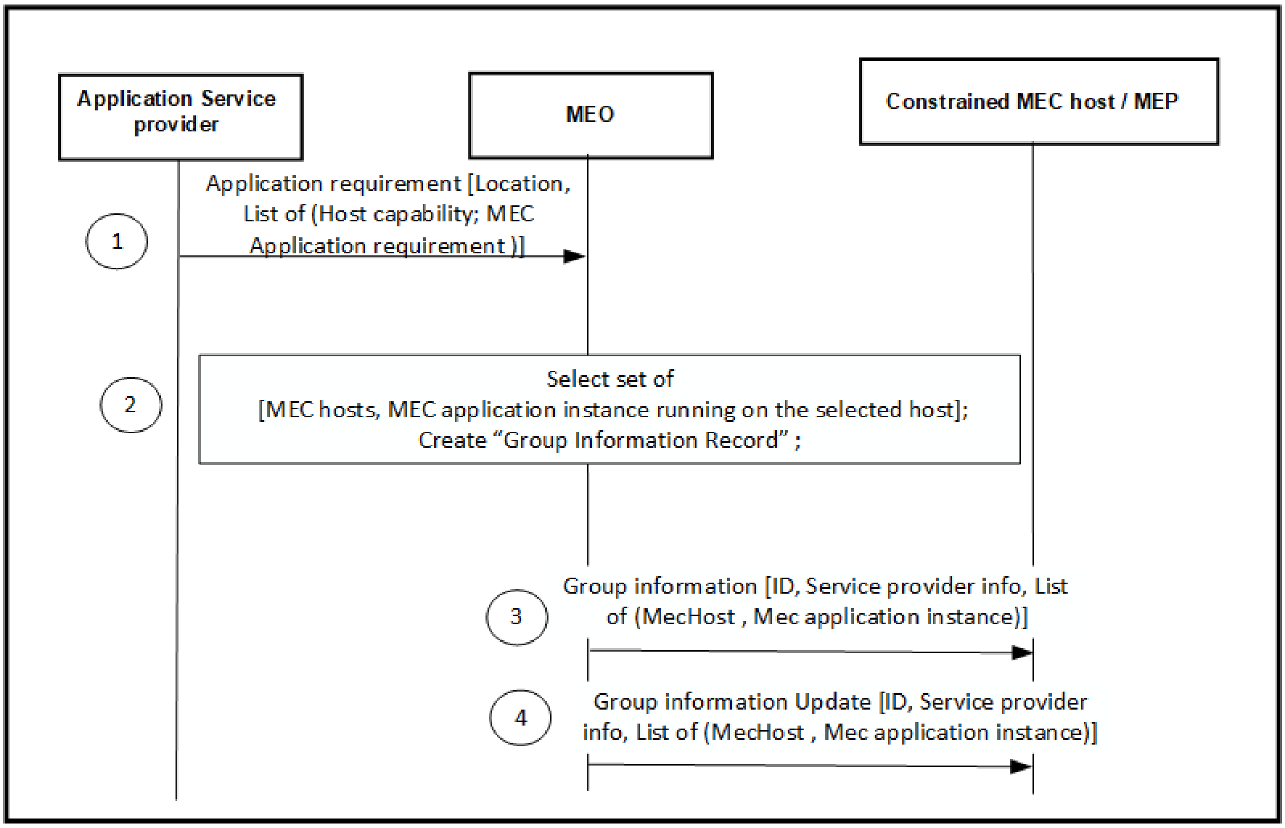

Figure 6.3.2-1: MEO supported Group creation and configuration

The Application Service Provider or the Factory owner can provide the application service requirement to the MEC system. E.g. an application service provider can be a third-party application service provider, who can be hired by factory owner to provide monitoring service.

In the MEC system, MEO can select MEC Applications in Constrained MEC Hosts to support the application service. It is assumed that all constrained MEC hosts are registered with MEO. MEO knows about the capability of the constrained MEC hosts, location, available MEC applications, etc. The group configuration information for each application service, can be stored, managed, updated by MEO.

Once the group of constrained MEC hosts and MEC application instances are selected by MEO, the information is distributed to the selected constrained MEC hosts.

An application service provider can change the group configuration information by adding or removing a constrained MEC host.

Step 1: Application Service provider provides application requirements such as location information, application description, constrained MEC hosts with certain capability and MEC applications to MEO.

Step 2: MEO, based on application requirements from step 1, determines available constrained MEC hosts in the specified location.

MEO checks if the constrained MEC hosts:

- have the required host capability.
- have deployed the required MEC application(s) instance.

MEO selects a set of hosts, which includes MEC host in Telco Edge and constrained MEC hosts, and MEC application instances running on these hosts.

MEO creates for each application task, a "Group information record", which consists of Host IDs, Host capability, MEC application instance ID. The information record is assigned a Group ID and associated with Service provider name, Application ID.

Step 3: MEO sends the "Group information record" to all the hosts which are part of the group. Constrained MEC hosts configure themselves with the group record. The Group information record can be used by constrained MEC hosts to interact with other MEC application instances in the group.

Step 4: If due to mobility or other conditions, group participant becomes unavailable, MEO can change the group configuration by adding or removing group elements. If the group configuration is changed, MEO sends updated Group information record to all the hosts.

Current ETSI GS MEC 010-2 [i.7] capabilities cannot be used to realize solution proposal #3-1. For example:

- In step 1 where the Application Provider provides information about application requirements (including constrained MEC host capabilities and MEC applications) to the MEC system.
- In Steps 3 and 4 where MEO configures and updates group information records with CMHs and Telco Edge MEC hosts.

To realize these steps, ETSI GS MEC 010-2 [i.7] does not include sufficient support or capabilities for groups of MEC applications or their associated MEC host requirements in either its MEC Application Package Onboarding or MEC Application Lifecycle Management operations on Mm1 or Mm3. ETSI GS MEC 010-2 [i.7] does define one and only one MEC app instance group operation in clause 5.5, Coordination Operations. The coordinated LCM operation (in ETSI GS MEC 010-2 [i.7] clause 5.5.1) is used to terminate one or more indicated MEC app instances when the MEO needs to terminate other MEC Applications in order to instantiate a new MEC app instance.

Two potential options are envisioned to utilize ETSI GS MEC 010-2 [i.7] to realize solution proposal #3-1:

1) **Option #1 - No change to ETSI GS MEC 010-2 [i.7] by isolating all group management operation in the OSS:**
   - The objective of this option is to minimize impact on ETSI GS MEC 010-2 [i.7] and the operation of the MEO.
   - All group management operation is isolated in the OSS.
   - Application provider requirements are communicated via the Mx1 interface to the OSS.
   - OSS creates and manages groups by utilizing existing 10-2 resources:
     - Specific AppD versions or instances are needed to create and manage separate MEC app configurations for different Constrained MEC host capabilities (e.g. varying compute, memory, or other required capabilities).
     - OSS executes individual MEC app operations on a per MEC app package and MEC app instance lifecycle basis.

     Potential drawbacks of this option include:
     - Requires many OSS/MEO interactions and possibly beyond to the customer over Mx1.
     - Causing complex operation for users of the MEC system (i.e. the application service provider).
     - Slow MEC system reaction to mobility of CHMs and changing conditions that require group configuration modification.

2) **Option #2 - Enhance 10-2 Lifecycle Management to support groups on Mm1/Mm3:**
   - The objective of this option is to:
     1) reduce complexity toward the MEC system customer (i.e. application service provider); and
     2) maximize MEC system efficiency (i.e. least number of interactions over MEC reference points).

- Potential 10-2 enhancements could include:

    - MEC application group onboarding methods that enable Application Packages with sets of CMH capabilities in a single operation by the OSS toward the MEO (Mm1). This reduces OSS/customer complexity and interactions with MEO.

    - Extend existing Coordinated LCM methods to include additional capabilities beyond termination, including group instantiation and operation. For example, such additional coordinated LCM operations on Mm3 would enable the MEO to lifecycle manage multiple MEC Applications as a group with an MEPM (e.g. that is managing one or more constrained MEC hosts).

    Potential benefits:

    - Reduced interactions between OSS and MEO - improved MEC system efficiency compared to option #1 and reduced OSS complexity.

    - Reduced interactions between MEO and MEPM - improved MEC system efficiency compared to option #1 and potential reduced MEO complexity.

    Potential drawbacks of this option include:

    - Change required to 10-2; however, this is not a technical drawback.

Furthermore, two additional solution proposals are described to address other 10-2 concerns, to realize solution #3-1. These solutions may apply generally beyond solution #3-1. They include:

1) Solution proposal #3-3, Constrained MEC Host Location Information in ETSI GS MEC 010-2 [i.7].

2) Solution proposal #3-4, Constrained MEC Host Capability Information in ETSI GS MEC 010-2 [i.7].

6.3.3 Solution proposal #3-2, Group operation through a Group management Function

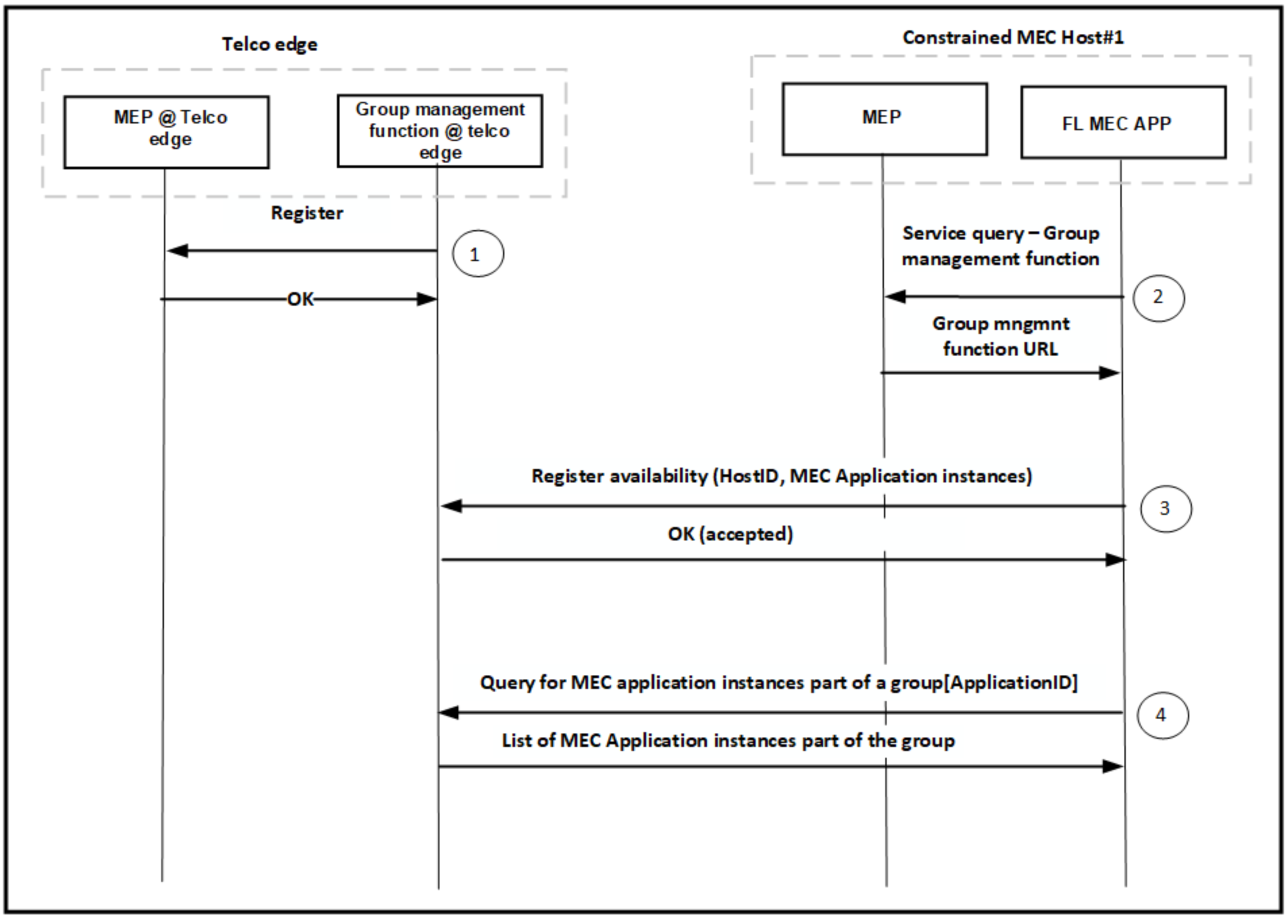

Figure 6.3.3-1: Group operation using Group Management Function

A Group Management Function (GMF) is assumed to support the group creation and discovery. GMF is a service available in the MEC Host, e.g. at Telco Edge. GMF maintains list of available MEC Hosts and MEC application instances running on them.

Step 1: GMF registers as a service producer with MEP at Telco Edge. By registering the service, it becomes discoverable across MEC system, including constrained MEC hosts.

Step 2: An FL agent MEC application in a CMH, wants to use group management service, queries the MEP for Group management function by providing the service name = GMF, service provider id, application id. MEP returns the URL of the Group management function.

Step 3: The FL agent MEC application instance running on CMH, registers with the GMF, to update its availability, by providing Host ID, its own application instance ID, service provider details, application id and optionally other available MEC application instance IDs. The GMF accepts the registration and sends a OK response back.

Step 4: The FL agent MEC application instance running on CMH queries GMF, for other MEC application instances, which are part of the group. The query can include the Service provider information, application details etc. GMF responds with MEC application instances, which are part of a group to support a specific application task. The FL agent MEC application can interact directly with the MEC application instances returned by GMF.

6.3.4 Solution proposal #3-3, Constrained MEC Host Location Information in ETSI GS MEC 010-2

ETSI GS MEC 010-2 [i.7] defines constraints for MEC application requirements related to MEC application deployment location in clause 6.2.2.2 - Type: LocationConstraints.

MEC host location constraint attributes include:

- Country Code
- Civic Address
- Geographic area - "Polygon" geometry object

These location constraints are not sufficient for Constrained MEC Hosts, which require greater location granularity than Telco Edge MEC hosts for MEC application requirements. To instantiate and manage MEC applications on Constrained MEC hosts (for a group or otherwise for a single MEC app instance), the attributes for 10-2 Location Constraints may need to consider:

- Network Location Information, such as zone information, connected access point information, etc. (as defined in ETSI GS MEC 013 [i.15]).
- Additional Geographic Location Information, such as latitude, longitude, altitude, additional shape information (such as area circle), etc. (as defined in ETSI GS MEC 013 [i.15]).
- Indoor Location Context Information, such as floor number, room number, aisle, etc., (as defined in in ETSI GS MEC 013 [i.15]).
- Relative Distance Information, for example between CMHs or between CHM and terminal devices (as defined in ETSI GS MEC 013 [i.15]).
- CMH Mobility Path Information.

6.3.5 Solution proposal #3-4, Constrained MEC Host Capability Information in ETSI GS MEC 010-2 [i.7]

ETSI GS MEC 010-2 [i.7] supports the needed data model information for MEC application requirements to specify Constrained MEC Host capabilities as follows.

Separate Application Packages with varying AppD configurations could be used to define and select MEC applications against a range of constrained MEC host capabilities:

- Each configuration would be in a separate Application Package with a specific AppD combination.
- However, this may incur the onboarding of several redundant application packages with a small number of AppD attribute differences, requiring redundant storage and run-time management.
- Potential 10-2 optimization is to enhance the application package with a list of AppD configurations that could be selected in LCM operations.

The Application Descriptor data model appears sufficient to express Constrained MEC Host capabilities and MEC application requirement for such capabilities, including:

- AppExternalCpd → external interfaces exposed by a MEC app, useful for interconnecting a group:
    - However, this defined at onboarding time, which may not be useful to update groups based on CMH mobility.
- appServiceRequired/appServiceOptional/appServiceProduced → group interconnection via MEC services.
- LogicalNodeRequirements → CMH node-level requirements.
- mcioConstraintParams → utilized for MCIO constraints for CMH capabilities such as GPU, FPGA, SSD, DPDK, etc.
- RequestedAdditionalCapabilityData → other specific CMH required capabilities.

6.3.6 Solution proposal #3-5, Group operation through MEC Application Slice management function

ETSI GR MEC 044 [i.12], "Study on MEC Application Slices", defines the following terms:

- MEC Application Slice (MAS): a logical MEC application service environment, which provides specific MEC application functions and related MEC service characteristics.
- MEC Application Slice Instance (MASI): A set of MEC Application instances, MEC service instances, as well as the required resources (e.g. compute, storage, and networking resources) which form a deployed MEC application slice.
- MEC application slice instance ID: an identifier of the MEC application slice instance.

The study also describes MEC Application Slice management functions:

- MEC Application Slice Communication Service Management Function (MAS-CMSF) responsible for customer-oriented management services of MEC application slices.
- MEC Application Slice Management Function (MAS-MF) responsible for the management and orchestration of MEC application slice instances.

As described in KI#3, the group is a set of MEC Applications deployed on specific MEC Hosts. These MEC applications can interact among themselves. It is assumed that a MEC application slice instance can be equivalent to a group. Figure 6.3.6-1 describes the use of MEC Application Slice management functions to create a group, based on the ETSI GR MEC 044 architecture "a" configuration (MAS-CSMF within OSS and MAS-MF within MEO).

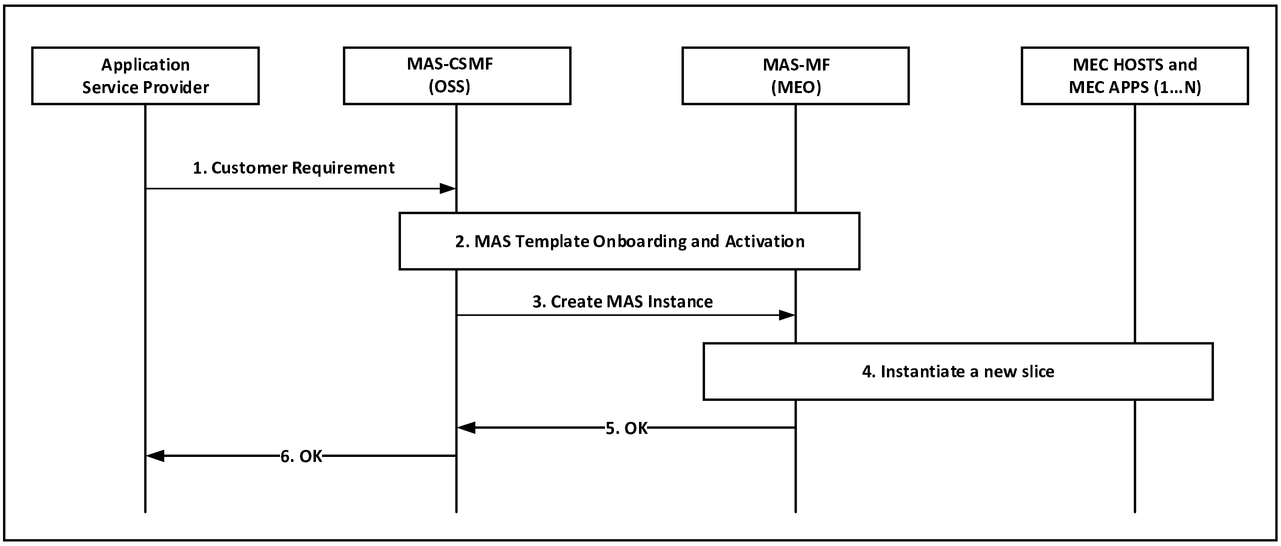

Figure 6.3.6-1: Group creation through MEC App Slice management function - new MEC Application Slice

Step 1: The MAS-CSMF receives group requirements from a customer or application service provider and translates them into slice requirements, captured in a MAS template which is equivalent to the group of MEC applications and MEC Hosts. Based on the customer requirements, either an existing MAS can be used, or a new slice can be created. This procedure is assuming a new MAS is needed.

Step 2: The MAS-CSMF onboards and activates the MAS template, representing a group, with the MAS-MF.

Step 3: After successful MAS template onboarding and activation, the MAS-CSMF sends a request to create a MAS instance for the MAS template in step 2.

Step 4: MAS-MF instantiates a new MAS with the needed MEC hosts (CMH and Telco MEC hosts) and MEC Application instances. As part of the instantiation of the MAS, communication among MEC Applications can be enabled.

Steps 5 and 6 are responses to the request indicating the operation was successful.

NOTE: For ETSI GR MEC 044 [i.12] architecture "f": The MAS-CSMF and the MAS-MF are combined. The procedure, above, applies without steps 2, 3, 4 and 5 as separate operations.

6.3.7 Evaluation

The solutions described to solve the KI #3, proposes various ways to create and manage a group of MEC application instances. A group is a set of MEC application instances (which can be instances of same or different MEC applications) running on specific set of hosts (Telco Edge and CMH).

Solution #3-1, #3-2 and #3-5 are alternate solutions and do not depend on each other. Solution #3-1 describes how MEO can create and manage a group based on application service provider requirements. Solution #3-2 describes use of a MEC application or MEC service to create and manage group. Solution #3-5 describes use of the MEC slice management function to create and manage a group.

Solutions #3-3 and #3-4 are specific to the use of ETSI GS MEC 010-2 [i.7] in the context of constrained MEC hosts (CMH), including MEC application instance deployment location and MEC host capabilities. Although these solutions were identified while addressing this key issue, they apply to the use of CMHs in general.

The solution proposal #3-1 describes how an Application Service Provider can provide the application service requirements to the MEC system. MEO can select MEC Application instances in constrained MEC hosts to support the application service. After selection, MEO shares the information with selected constrained MEC hosts.

The solution is technically feasible, if the following conditions are met:

- Application Service Provider can provide application requirements such as location information, application description, constrained MEC host capability information, and MEC applications to the MEO.
- MEO can have the knowledge of the capability of the constrained MEC hosts, location of the constrained MEC hosts, available MEC application instances hosted in the constrained MEC hosts, etc.
- MEO can create a group based on application requirement and knowledge of constrained MEC hosts.
- MEO can create, store, update the group configuration information.
- MEO can share the group configuration information with constrained MEC hosts and constrained MEC hosts stores the configuration information.
- MEO can know unavailability of a constrained MEC host in a group, update group configuration, and sends the updated information to the impacted constrained MEC hosts in the group.

For solution #3-1, ETSI GS MEC 010-2 [i.7] capabilities are evaluated to identify if it can be applied to realize,

- How can an Application Service Provider provide information about application requirements?
- How MEO can configure and update group information records?

The evaluation concluded that ETSI GS MEC 010-2 [i.7] does not include sufficient support or capabilities for groups of MEC applications or their associated MEC host requirements in either its MEC Application Package Onboarding or MEC Application Lifecycle Management operations.

Two options are proposed:

- No change to ETSI GS MEC 010-2 [i.7] by isolating all group management operation in the OSS.
- Enhance 10-2 Lifecycle Management to support groups on Mm1/Mm3.

The option to enhance ETSI GS MEC 010-2 [i.7] Lifecycle Management to support groups is technically feasible, if the following conditions are met:

- OSS can support in a single operation toward the MEO (Mm1) to onboard a group of MEC application instances by creating an application package with sets of CMH capabilities.
- Coordinated LCM methods can be extended to include additional capabilities beyond termination, including instantiation and operation for multiple MEC application instances (i.e. group), which can be instances of same or different MEC applications.

Solution #3-3 proposes updates to the MEC application requirements related to MEC application instance deployment location in ETSI GS MEC 010-2 [i.7] that are needed for the more detailed location granularity of constrained MEC hosts. The solution is technically feasible, if the following conditions are met:

- ETSI GS MEC 010-2 [i.7] location constraints can include Network Location Information, such as zone information, connected access point information.
- ETSI GS MEC 010-2 [i.7] location constraints can include Additional Geographic Location Information, such as latitude, longitude, altitude, additional shape information.
- ETSI GS MEC 010-2 [i.7] location constraints can include Indoor Location Context Information, such as floor number, room number, aisle.
- ETSI GS MEC 010-2 [i.7] location constraints can include Relative Distance Information, for example between CMHs or between CHM and terminal devices.

Solution #3-4 concludes that the existing ETSI GS MEC 010-2 [i.7] data model information for MEC application requirements, including the Application Descriptor, is sufficient to specify constrained MEC host capabilities. The solution is technically feasible, if the following conditions are met:

- If separate Application Packages with varying AppD configurations can be used to define and select MEC application instance against a range of constrained MEC host capabilities.
- Each configuration would be in a separate package, for a single MEC application instance, with a specific AppD combination.

Solution #3-4 proposes a potential ETSI GS MEC 010-2 [i.7] optimization to enhance the Application Package with a list of AppD configurations that could be selected in lifecycle management operations. This would reduce the number of redundant application packages needed.

The solution proposal #3-2 proposes a MEC application function, called "Group Management Function (GMF)", to support group creation and group discovery. GMF can be a MEC service available in the MEC Host, e.g. at Telco Edge. GMF maintains list of available MEC Hosts, including constrained MEC hosts, and MEC application instances running on these hosts.

The solution is technically feasible, if the following conditions are met:

- A "Group Management Function (GMF)" can create and update a group based on application service provider requirements, which can be provisioned by an application service provider.
- A "Group Management Function (GMF)" can store information about availability of MEC application instances hosted on constrained MEC hosts, including Host ID, its own application instance ID, service provider details, application id, and optionally other available MEC application instance IDs.
- The "Group Management Function (GMF)" based on application service provider requirements and availability information of MEC application instances, can create a group.
- MEC application instances can query the "Group Management Function (GMF)" for group information.
- The "Group Management Function (GMF)" can provide information to MEC application instances about the group and other MEC application, which are part of the group.

The solution proposal #3-5 considers a group a MEC application instances as a slice and proposes use of a slice management function to create a group based on the ETSI GR MEC 044 [i.12] architecture "a" configuration (MAS-CSMF within OSS and MAS-MF within MEO). The solution also applies to the ETSI GR MEC 044 architecture "f" configuration (MAS-CSMF and the MAS-MF).

The solution is technically feasible, if the following conditions are met:

- MAS-CSMF can receive group requirements from a customer or application service provider and translates them into slice requirements, captured in a MAS template.
- MAS-CSMF can onboard and activate the MAS template, representing a group, with the MAS-MF.
- MAS-CSMF can send a request to create a MAS instance for the MAS template.

- MAS-MF can instantiate a new MAS with the needed MEC hosts (CMH and Telco MEC hosts) and MEC Application instances.
- As part of the instantiation of the MAS, communication among MEC Applications can be enabled.

6.4 Gap/Key issue #4 - Telco Edge Far Edge (constrained device) Interaction

6.4.1 Description

In the smart factory, MEC Applications (i.e. federated learning agents) in constrained MEC hosts interact with MEC applications (e.g. federated learning server or aggregator) in Telco Edge. The event of disruption in interaction between a MEC app in constrained MEC host and MEC app in Telco Edge can result in interruption or degradation of the federated learning infrastructure or process.

To be able to continue the learning process, the MEC System can select another constrained MEC host in the local area with similar capability (e.g. with the federated learning agent MEC App) to the constrained MEC host, which lost connectivity (as described in solution proposal #1-2). However, if another constrained MEC host with similar capability is not available, the learning process stops or degrades due to lack of FL updates from the disrupted device, which can impact the operation of the factory.

However, another constrained MEC host, with direct connectivity to the disrupted constrained MEC host, may offer a relay function to enable the federated learning application to continue operation.

6.4.2 Solution proposal #4-1, Configuration of an Intermediate Device Relay Function to Support Constrained MEC Host Updates towards Telco Edge

A constrained MEC host (i.e. Far Edge) loses user plane connectivity with a MEC host in the Telco Edge, impacting a federated learning application. No similarly capable constrained MEC hosts are available to transition and continue the FL agent MEC app from the disrupted device. To continue the learning process, another constrained MEC host offering a Relay Function, which collects FL updates and transfers these updates between the Telco Edge and Far Edge (constrained MEC host), is proposed.

The second constrained MEC host is an **Intermediate Device**. An intermediate device is:

- A constrained MEC host with direct connectivity to the impacted constrained MEC host (with lost or degraded connectivity).
- Supports the **Relay Function** (RF), which can relay FL updates between the impacted constrained MEC host FL agent MEC application and the Telco Edge MEC application.

As illustrated in Figure 6.4.2.1, a FL agent MEC app on a constrained MEC host (CMH #1) stops interacting with a FL MEC app in a Telco Edge MEC host. The MEC system becomes aware of the connectivity loss between the constrained MEC host and Telco Edge MEC host. The MEC system (e.g. OSS/MEO) selects a suitable intermediate device (direct connectivity, relay function instantiated) in the vicinity of CMH #1 which can support FL data relaying with the Telco Edge. The MEC system configures the relay function on the intermediate device with information about Telco Edge MEC host and CMH #1. Once the relay function is configured, it relays application data between the FL MEC Applications at the Telco Edge and the Far Edge CMH #1. Further details are depicted in Figure 6.4.2-1, followed by description of the procedure.

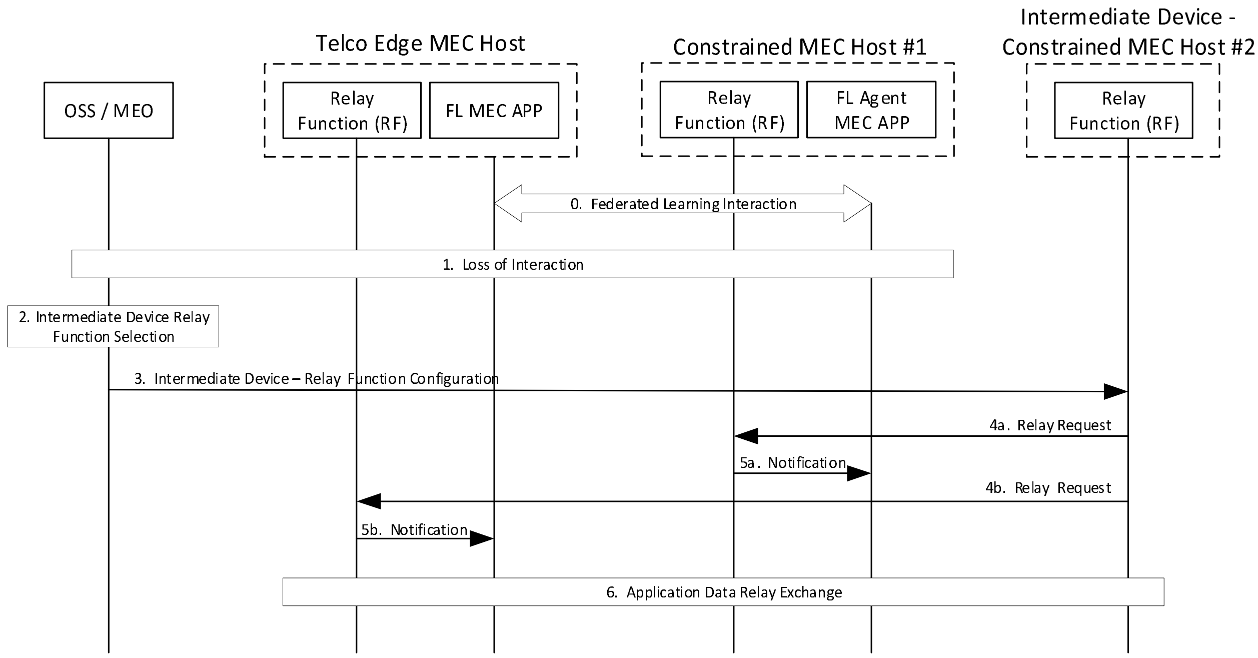

Figure 6.4.2-1: Configuration of intermediate constrained device relay function

Step 0: Prerequisites

- FL MEC app is running on a Telco Edge MEC Host and interacting with FL Agent MEC app on constrained MEC host #1 (CMH #1).
- Another constrained MEC host #2 (CMH#2) is available in the vicinity of CMH#1 with direct connectivity.
- The Relay Function (RF) is available on the Telco MEC Host, CMH#1, and CMH#2. The Relay Function may be realized as a MEC service or as a MEC Platform function.
- The FL MEC app and the FL Agent MEC app are sensitive to connectivity loss. As such, they have subscribed to relay notifications from the RF. They may have discovered the RF via the MEP, as a MEC service or a MEP function.
- Within the MEC system, the OSS/MEO are aware of the host capabilities of the CMH #1 and #2 (storage, compute, etc.), their location (e.g. as defined in ETSI GS MEC 013 [i.15]), and connectivity (including available direct connectivity).

The format of a constrained MEC host location information may follow ETSI GS MEC 013 [i.15], including geographical coordinates, location zone identity, civic location information, relative location, and/or connected access point identities.

Constrained MEC host location information may be provided by a MEC service (such as ETSI GS MEC 013 [i.15]), by the underlying multi-access network (which is out of scope the MEC system), or via the constrained MEC host configuration and registration methods in clause 6.2.

Step 1: Loss of interaction is detected between FL MEC app running on a Telco Edge MEC Host and FL Agent MEC app on CNH #1.

- Loss of interaction may be detected at the application level by the FL MEC app and the FL Agent MEC app.
- Loss of interaction may also be detected and indicated by the underlying access network. This is out of scope of the MEC system.
- Loss of interaction may be detected via the data plane unavailability solution in clause 6.1.2.

Step 2: The OSS/MEO selects an appropriate and available intermediate device based on:

- Location of CMH#1 and CMH#2.

- Availability of direct connection between CMH#1 and CMH#2.
- Availability of RF on CMH#2.
- Storage, computing, and other host capabilities of CMH#2.

Since CMH#2 fits the criteria to serve as an intermediate device, OSS/MEO selects CMH#2 as an intermediate device.

Step 3: The OSS/MEO configures the RF in CMH #2 to support and initiate application data relaying between the FL MEC app in the Telco Edge and FL Agent MEC app in the Far Edge. Configuration information includes:

- Constrained MEC Host #1 Identifier: identifier of the constrained MEC host that lost connectivity.
- Constrained MEC Host #1 Relay Function API: Interface to the RF on CMH#1. RF on the intermediate device can interact with the CMH#1 RF using this API.
- Constrained MEC Host #1 IP Address: IP address of the constrained MEC host that lost connectivity.
- Telco Edge MEC Host Identifier: identifier of the MEC host that lost connectivity with CMH#1.
- Telco Edge MEC Host Relay Function API: Interface to the RF on the Telco MEC host. RF on the intermediate device can interact with the Telco Edge RF using this API.

The OSS/MEO configuration of the RF may be realized by the existing reference points (e.g. Mm2/Mm3/Mm5).

Steps 4a and 4b: The intermediate device RF issues a relay request to the CMH #1 RF and to the Telco Edge MEC host RF. Step 4a depicts the relay request to CMH#1 and step 4b depicts the relay request to the Telco MEC host RF. These steps may occur in parallel and in any order. Relay requests include:

- MEC App Identifier: identifier of the MEC app for the relay request (e.g. FL MEC app or FL Agent MEC app).
- Intermediate Device (CMH#2) Identifier: identifier of the intermediate device.
- Intermediate Device (CMH#2) Relay Function API: Interface to the RF on intermediate device.

Relay functions on the Telco Edge MEC host and constrained MEC hosts may interact via the Mp3 interface or via a relay function service interface.

Steps 5a and 5b: The Telco Edge and CMH#1 RFs issue notifications for application data relaying to their associated MEC applications. The notification may include a request for information to coordinate relaying. Information may include:

- App priority: priority for the relay updates/data.
- App data size: size of relay updates/data.
- App update interval: frequency of sending relay updates/data.
- App update snapshot: initial application data to relay if requested and available.

The Telco Edge and CMH#1 RFs return this information to the intermediate device RF (e.g. in response to the relay request in step 4). The intermediate device may allocate appropriate buffer space for relay data and request enhanced access network capacity for transferring relay data in the uplink or downlink. This is out of scope of the MEC system.

Step 6: Application data relay exchange between the Telco Edge MEC app and CMH#1 MEC app initiates and continuously operates via the relay function until connectivity may be restored between CMH#1 and the Telco Edge.

When connectivity is restored between CMH#1 and the Telco Edge, the RF is terminated automatically. FL MEC App and the FL Agent MEC app detects connectivity is restored and resumes direct communication. No interaction happens with the RF and after certain period of inactivity, RF may terminate automatically, e.g. the socket between FL MEC App and RF can be closed automatically due to certain period of inactivity. Restoration of connectivity may be detected by:

- At the application level by the FL MEC app and the FL Agent MEC app.

- Indicated by the underlying multi-access network. This is out of scope of the MEC system.

6.4.3 Solution proposal #4-2, Relay Function Registration with a Relay Discovery MEC Service

This solution is an alternative to Solution Proposal #3-1 which defines a solution where an Intermediate Device with a Relay Function (RF) assists MEC applications (e.g. FL apps) to continue to interact under connection loss conditions between the Far Edge and Telco Edge. In Solution Proposal #3-1, the MEC system (e.g. OSS/MEO) selects a suitable intermediate device (direct connectivity, relay function instantiated) in the vicinity of constrained MEC host when connectivity is lost to support application data relaying with the Telco Edge.

This solution is MEC Service based and does not include the OSS/MEO selecting an intermediate device and relay function instance.

The Relay Function (RF) is a function of an Intermediate Device which offers a varying range of capabilities to consumer MEC applications.

For example:

1) relay for certain MEC app types or specific MEC app instances;

2) relay between specific hosts, in specific locations, or between network domains;

3) relaying within certain traffic types or limits (size of data, frequency of data, etc.).

This solution introduces a new MEC service: Relay Discovery Service. The Relay Discovery Service may be deployed in a more centralized location in the MEC system, e.g. the Telco Edge. The Relay Discovery Service gathers capabilities, location, connectivity state, and other information from specific Relay Function instances on intermediate devices. With this information, the Relay Discovery Service can match relay discovery requests from MEC applications (based on their input requirements and characteristics) with appropriate Relay Function instances.

In the context of the Relay Function, two sets of information are needed:

1) Relay Function Capabilities:
    - This is information about the capabilities of a specific RF instance.
    - Examples: relay type (e.g. relay to Telco Edge, relay to a specific Telco Edge host, relay to a CMH host, etc.), ability to reach certain domains, supported MEC applications (e.g. types, names, instance IDs), traffic types, traffic limits, service areas, locations, etc.

2) Relay Function Configuration Information:
    - This is information that RF instance needs to be configured for a consuming MEC app.
    - Examples: application type, destination/source MEC application for relaying, application data size, application data interval, relay priority, etc.

Figure 6.4.3-1 shows how a Relay Function on an Intermediate Device (Constrained MEC Host) discovers Relay Discovery Service as a MEC service via the MEP. It also shows how the Relay Function registers its capabilities, required configuration information, and endpoint information with the Relay Discovery Service. This information is needed for a MEC application to select an appropriate RF instance, as described in clause 6.4.4.

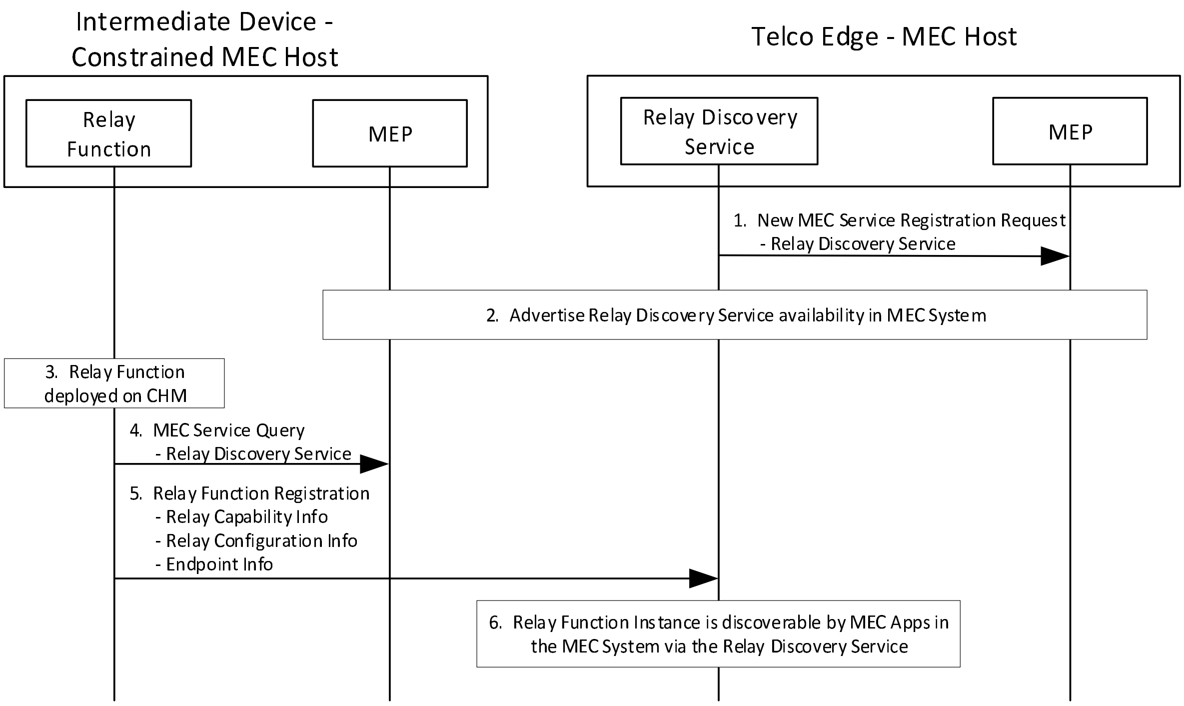

Figure 6.4.3-1: Intermediate device relay function registration with Relay Discovery Service

Step 0: Prerequisites

- The RF is a MEC application that can be instantiated on a Constrained MEC Host that can serve as an Intermediate Device, as described in clause 6.4.2.
- The RF MEC application is authorized to offer the Relay Function (e.g. to FL apps) in the MEC system.
- The RF MEC application is authorized to use the RF Discovery Service.
- The RF Discovery Service is a MEC service produced by a MEC application deployed on a MEC Host in the Telco Edge.
- The RF Discovery Service is a MEC Service that can be registered and discovered in the MEP over Mp1.

Step 1: MEC app instance (that produces the RF Discovery Service) in the Telco Edge sends a new service registration request to the MEC platform in the Telco Edge. The MEP verifies the new service registration request and adds the RF Discovery Service to its service registry.

Step 2: The MEP in the Telco Edge MEC Host advertises or updates the MEO/OSS or other MEPs (including CMHs in Far Edge) about the availability of the registered RF Discovery Service. The RF Discovery Service is available within the MEC system and discoverable by MEC applications on Constrained MEC Hosts and Telco Edge MEC Hosts.

Step 3: The RF is deployed as a MEC application instance on a CHM host by the MEO/MEPM.

Step 4: The RF sends a service discovery query to the MEC platform in the CHM for the RF Discovery Service. The MEP responses with the service information for the RF Discovery Service, including its service endpoint.

As an alternative, the RF may discover the RF Discovery Service via a service availability notification from the MEP.

Step 5: The RF on an Intermediate Device (CMH) sends a new RF registration request to the RF Discovery Service in the Telco Edge. The RF includes the following RF instance-specific information in the registration request:

- Relay Function Capabilities:
  - Relay type (e.g. relay to Telco Edge, relay to a Telco Edge host, relay to a CMH host)
  - Supported MEC applications for relaying (app names, app descriptor IDs, app providers, app instance IDs)

- Relay traffic types supported by the RF service
- Relay traffic limits: transfer rate, size, latency, packet loss, update interval and frequency supported
- Relay service area: civic addresses, indoor locations, a set of geographic coordinates or shape, a topological network location
- Host identifier: where the RF is available

• Relay Function Configuration Information (this information needs to be configured by a consuming MEC app, as described in clause 6.4.4):
  - App type (app name, app descriptor ID)
  - App instance ID
  - App priority: priority for the relay updates/data
  - App data size: size of relay updates/data
  - App traffic type
  - App update interval: frequency of sending relay updates/data
  - Relay endpoints: source and destination

• Relay Function Endpoint Information
  - Endpoint information that consuming MEC application can used to access the RF

Step 6: The RF instance is discoverable by MEC Applications in the MEC System via the Relay Discovery Service.

6.4.4 Solution proposal #4-3, Discovery, Configuration, and Use of an Intermediate Device, hosting Relay Function to Support Constrained MEC Host Updates towards Telco Edge

In a Smart Factory, a constrained MEC host (i.e. Far Edge) loses user plane connectivity with a MEC host in the Telco Edge, impacting a federated learning application. Without any similarly capable constrained MEC hosts available to transition and continue the FL agent MEC app from the disrupted device, a Relay Function on an Intermediate Device (with direct connectivity to the disrupted device) is used to collect and transfer FL updates between the Far Edge and Telco Edge.

This solution is an alternative to Solution Proposal #3-1 which defines a solution where an Intermediate Device with a Relay Function (RF) assists MEC applications (e.g. FL apps) to continue to interact under connection loss conditions between the Far Edge and Telco Edge. In Solution Proposal #4-1, the MEC system (e.g. OSS/MEO) selects a suitable intermediate device (direct connectivity, relay function instantiated) in the vicinity of constrained MEC host when connectivity is lost to support application data relaying with the Telco Edge. In this solution, the impacted MEC app discovers the Relay Function on the Intermediate Device via the RF Discovery Service, presented in clause 6.4.3 without any OSS/MEO configuration.

As shown in Figure 6.4.4-1, a FL agent MEC app is deployed on a constrained MEC host (CMH #1 in the Far Edge). Additionally, a FL MEC app is deployed on a MEC host in the Telco Edge. These MEC Applications part of a federated learning application in the Smart Factory. The FL MEC app in the Telco Edge transfers a global FL model to the FL agent MEC app on CMH#1. When the FL agent MEC app completes its local training, it transfers a local FL model update to the FL MEC app in the Telco Edge. Since these MEC Applications are sensitive to connectivity loss, the FL agent MEC app discovers and configures a Relay Function service instance that is available on an Intermediate Device.

While engaging in federated learning, the FL agent MEC app on CMH #1 and the FL MEC app on the Telco Edge MEC host detect a loss of interaction. To continue the FL process, the FL agent MEC app issues a relay request to the Relay Function on CMH#2. In response to the request, the relay function relays application data (FL model updates) between the FL MEC Applications at the Telco Edge and the Far Edge. Further details are depicted in Figure 6.4.4-1 with a procedure description.

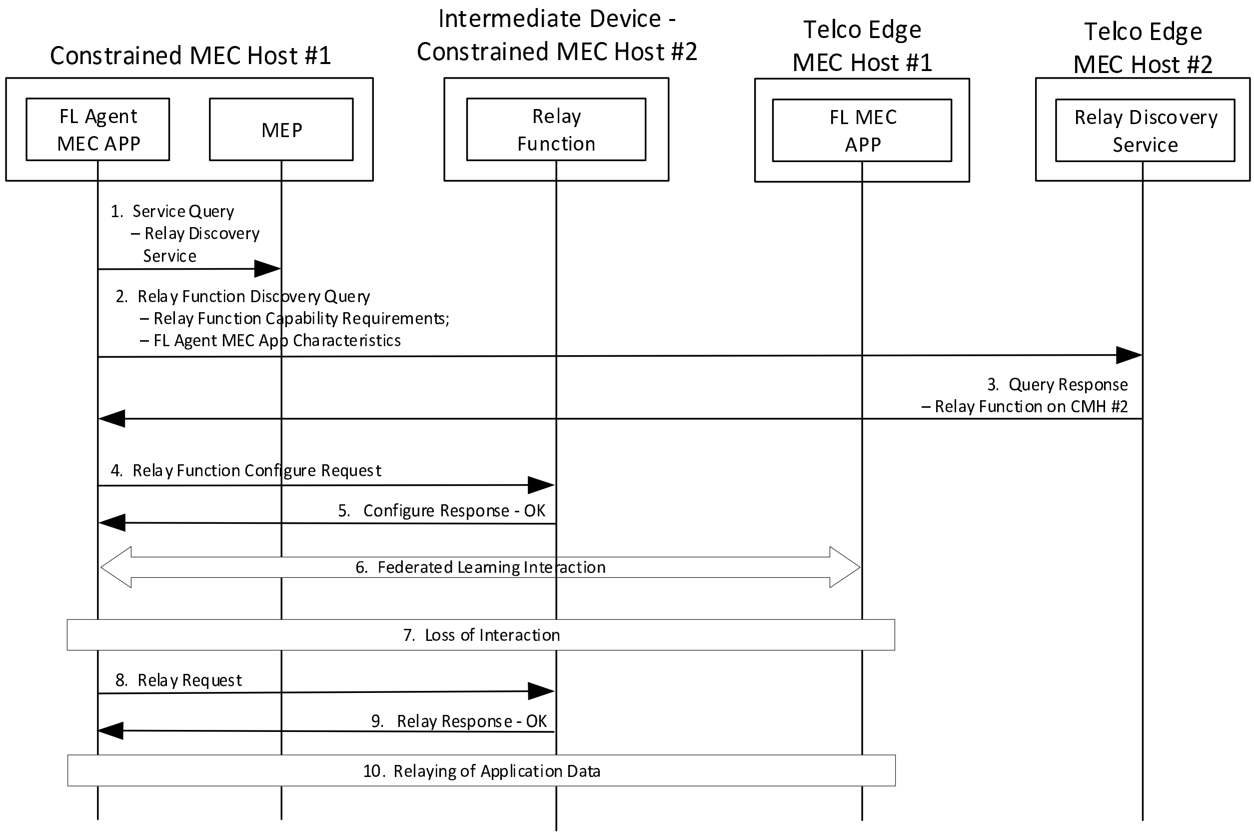

Figure 6.4.4-1: Discovery, Configuration, and Use of an Intermediate Device, Only Relay Function

Step 0: Prerequisites

- FL MEC app is instantiated and running on a Telco Edge MEC Host.
- FL Agent MEC app is instantiated and running on CMH #1 in the Far Edge.
- These MEC Applications are both deployed as part of a federated learning application in the Smart Factory.
- They have discovered each other and have sufficient authorization to interact with each other.
- Another constrained MEC host #2 (CMH#2) is an Intermediate Device that is available in the vicinity of CMH#1 with direct connectivity.
- A Relay Function (RF) is a MEC App that is deployed on Intermediate Device (CMH#2). The Relay Function is registered and available in the MEC system to the FL MEC app and the FL Agent MEC app via the Relay Function Discovery Service, as described in clause 6.4.3.

Step 1: The FL agent MEC app on CMH#1 discovers the Relay Function Discovery Service via the service query procedure with the MEP. The MEP responses with the service information for the RF Discovery Service, including its service endpoint.

Step 2: The FL agent MEC app on CMH#1 discovers the Relay Function via the Relay Function Discovery Service in a RF discovery query request. In the RF discovery query request, the FL agent MEC app includes information to help the Relay Function Discovery Service select a RF instance(s) that can service the FL agent MEC app, based on the RF capabilities and RF configuration information provided in registration (see clause 6.4.3). The RF function discovery query request includes:

- RF Capability Requirements, including:
    - Specific host where the relay service is hosted, if known by the FL agent MEC Application.
    - Capability of the relay service to reach a specific domain such as abc.com, xyz.com.
    - Capability to reach a specific host, where the destination MEC application for the relay is hosted.

   - Processing capability, such as capability to process FL/ML data, forward data at a certain rate.

- FL agent MEC Application Characteristics, including:
   - Type of application; FL agent MEC app, e.g. FL application.
   - FL agent generates data at a certain rate.
   - FL agent generates data of certain size.
   - Tolerable delay.
   - Location information, where the source FL agent MEC app is running such as Civic address, geo-location information, etc. so that a service can be found, which is in proximity of the MEC app.

The FL agent MEC app on CMH#1 may be aware of the Intermediate Device (CMH#2) and provide the identifier of the Intermediate Device (CMH #2) in the RF discovery query request. Alternatively, FL MEC App in the Telco Edge and the FL agent MEC app on CMH#1 may be unaware of the Intermediate Device (CMH #2). The selection of the Intermediate Device (CMH#2) may be realized via the Relay Function Discovery Service.

Step 3: The Relay Function Discovery Service responds with a list of RF instances that satisfy the FL agent in the discovery query (e.g. RF on CHM #2). The query response includes any Relay Function Capabilities and Relay Function Configuration Information that may have been included in the RF registration, e.g. using the procedure in clause 6.4.3. If more than one RF instance is returned in the query response, the FL agent MEC app selects an instance based on its internal criteria and using the query response, including any Relay Function Capabilities and Relay Function Configuration Information.

Alternatively, the FL agent MEC app on CMH#1 may discover the RF via an availability notification from the Relay Function Discovery Service (not shown in Figure 6.4.4-1).

Step 4: The FL agent MEC app sends a configuration request to the RF instance selected from step 3, including:

- Application Characteristics of the FL agent, including:
   - App name and instance ID
   - App type/category
   - App priority
   - Location information
- Relay Function Configuration Information, including:
   - Target MEC app, indicates target MEC application, where data is to be relayed.
   - Target Host, the host where the Target MEC Application is hosted.
   - CallbackURL, how to reach the MEC Application for information and notifications related to the service.
   - Application Data Size, indicates the size of the data the Relay Service can expect from MEC application.
   - App Update Interval, indicates the frequency of relay data updates the Relay Service can expect.
   - Application Relay Data, current snapshot of the application data to relay, if available.

Step 5: The RF configures the necessary parameters and capabilities based on the request and replies to the FL agent MEC app with an OK response. The RF is ready to accept relay requests, when needed from the FL agent MEC app. For example, the intermediate device may allocate appropriate buffer space to relay data and request enhanced access network capacity for transferring relay data. This may occur at the time of configuration to ensure required resources are available or may be deferred until the FL agent MEC Applications issues a relay request (step 8). How the intermediate device allocates additional resources is out of scope of the MEC system.

Step 6: The FL MEC app in the Telco Edge and the FL agent MEC app on CMH#1 engage in federated learning by exchanging model updates from Telco Edge to Far Edge and vice versa.

For example, in the FL process:

- The FL agent MEC app, after it completes a local training iteration, sends a local model update to the FL MEC app in the Telco Edge.
- This exchange executes continuously over training iterations.

FL interactions are at the application level.

Step 7: The FL agent MEC app on CMH#1 detects loss of interaction with the FL MEC app in the Telco Edge. This loss of interaction is detected at the application level by the FL agent MEC app.

Step 8: The FL agent MEC app on the CMH#1 issues a relay request to the Relay Function on the Intermediate Device (CMH#2) to continue to transfer its local model updates to the FL MEC app in the Telco Edge.

The relay request may update the relay parameters from the relay configuration:

- Application Priority: priority for the relay updates/data - priority of the FL local model update
- Application Data Size
- Application Update Interval
- Application Relay Data: update snapshot, initial application data to relay if requested and available - current FL local model update (if available)
- Location information

The intermediate device may update its parameters, if needed. For example, allocate appropriate additional buffer space for the FL local model update relay data and request enhanced access network capacity for transferring the relay data to the Telco Edge MEC host. This is out of scope of the MEC system.

Step 9: The RF replies to the relay request with an OK response.

Step 10: Application data relay exchange between the Far Edge CMH#1 MEC app and Telco Edge MEC app continuously operates via the relay function until connectivity may be restored between CMH#1 and the Telco Edge.

When connectivity is restored between CMH#1 and the Telco Edge, the RF is terminated automatically. FL MEC App and the FL Agent MEC app detects connectivity is restored and resumes direct communication. No interaction happens with the RF. After certain period of inactivity, RF may terminate automatically, e.g. the socket between FL MEC App and RF can be closed automatically due to certain period of inactivity.

Restoration of connectivity may be detected by:

- At the application level by the FL MEC app and the FL Agent MEC app.
- Indicated by the underlying multi-access network. This is out of scope of the MEC system.

6.4.5 Evaluation

The solutions described to solve the KI #4, utilizes an application, called "Relay function", and application-level mechanisms to mitigate connection loss between the Telco Edge and Far Edge constrained MEC host. This solution can complement other access network solutions that may provide connectivity relay capabilities.

Solution #4-1 utilizes a Relay Function that is deployed in the Telco Edge, on the Far Edge Constrained MEC Host, and on an Intermediate Device. This solution uses the MEO to select and configure a Relay Function instance that enables MEC application to continue to interact when connection loss occurs.

Solutions #4-2 and #4-3 are alternative solutions, based on a Relay Discovery Service, which may be deployed in the Telco Edge. The Relay Discovery Service enables MEC applications on constrained MEC hosts to register, discover, configure, and utilize a Relay Function Service to mitigate connection loss between the Telco Edge and Far Edge. This solution does not impact the MEO and utilizes existing MEP service enablement functionality. The Relay Discovery Service and the Relay Function Service could be standardized as a MEC Service.

The solution proposal #4-1 **uses a Relay Functions (RF) deployed in constrained MEC host, Telco Edge and on an Intermediate device**. The solution describes how a MEC system (e.g. OSS/MEO) selects a suitable intermediate device in the vicinity of the constrained MEC host, which lost connectivity to the Telco Edge. The intermediate device supports FL data relaying between Telco Edge and impacted constrained MEC host. The MEC system configures the relay function on the intermediate device with information about Telco Edge MEC host and the constrained MEC host, which lost connectivity.

The solution is technically feasible, if the following conditions are met:

- OSS/MEO can detect loss of interaction between FL MEC app running on a Telco Edge MEC Host and FL Agent MEC app on constrained MEC host.
- OSS/MEO can select an appropriate and available "intermediate device" based on information about the constrained MEC host such as location, availability of direct connectivity, availability of MEC applications and services.
- OSS/MEO has configuration information and can configure the RF in intermediate device to support application data relaying.
- The **Intermediate device RF** can issue a relay request to the constrained MEC host RF and to the Telco Edge MEC host RF.
- The Telco Edge RF and constrained MEC host RFs are capable of issuing notifications for application data relaying to their associated **MEC applications**.
- When connectivity is restored between constrained MEC host and the Telco Edge, the RF can be terminated.

The solution proposal #4-2 **proposes a MEC Service to support the Relay Function (RF) and does not involve the OSS/MEO in selecting an intermediate device** and configuring the relay function instance. The proposed RF is a function of an Intermediate Device capable of relaying data for certain MEC app types or specific MEC app instances. A service called, "Relay Discovery Service" is also proposed, which may be deployed in a more centralized location in the MEC system, e.g. the Telco Edge. The Relay Discovery Service gathers capabilities, location, connectivity state, and other information from specific RF instances on intermediate devices.

The solution is technically feasible, if the following conditions are met:

- The availability of Relay Discovery Service can be advertised. The Relay Discovery Service is available within the MEC system and discoverable by MEC applications on Constrained MEC Hosts and Telco Edge MEC Hosts.
- RF can discover the Relay Discovery Service, including its end point information.
- RF can register its capabilities, required configuration information, and endpoint information with the Relay Discovery Service.

The solution proposal #4-3 **describes RF operation using the Relay Discovery Service** solution, described in proposal #4-2.

The solution is technically feasible if the following conditions are met:

- MEC application (FL Agent) can discover the Relay Discovery Service.
- MEC application (FL Agent) can query the Relay Discovery Service for a suitable RF by including RF Capability Requirements and FL agent MEC Application Characteristics.
- Relay Discovery Service can select or responds with a list of RF instances that satisfy the FL agent in the discovery query. The query response includes any RF Capabilities and RF Configuration Information that may have been included in the RF registration.
- The FL agent MEC app can send a configuration request to the selected RF instance including Application Characteristics of the FL agent and RF Configuration Information.
- The RF can configure the necessary parameters and capabilities based on the request received from FL Agent MEC app.

- The FL agent MEC app can detect loss of interaction with Telco Edge and issue a relay request to the RF on the Intermediate Device. The relay request can update the relay parameters from the relay configuration.
- When connectivity is restored between constrained MEC host and the Telco Edge, the RF can be terminated.

6.5 Gap/Key issue #5 - Resource allocation and re-distribution to XR MEC application(s)

6.5.1 Description

A multi-player XR game, can be decomposed into multiple MEC Applications, that form a MEC Application Group, as described in clause 5.3.1. As shown in Figure 5.3.1-3, the MEC Applications within the XR MEC Application Group have the following functionalities:

- Display client and/or video client as a MEC App on a CMH
- XR Rendering service as a MEC App on a CMHor MEC Host
- Video server as a MEC App on a CMHor MEC Host
- Game coordinator as a MEC app on a MEC Host
- Several game play logic entity MEC Applications, depending on the internal XR game needs

As players move around, the XR MEC Application Group faces runtime changes in resource conditions. In response, the XR MEC Application Group needs to adapt to such varying conditions without impacting critical application properties and maintaining sufficient XR experience quality for the players. To assist in achieving this, the XR MEC Application Group includes an XR adaptation module that runs as a MEC App. This XR adaptation MEC app has MEC Application Group wide visibility towards XR performance from the application-level perspective of players across all MEC Applications within the XR MEC Application Group. This information can be used to make requests to the MEC system for re-allocation of resources to one or more of the MEC application instances within the XR MEC Application Group.

To assist runtime resource reallocation, the MEC system can expose an XR Enabler Service. This service can be used by the XR adaption module to make requests for MEC resource reallocation for one or more MEC applications in the XR MEC Application Group. Specifically, the XR Enabler Service can expose the following functionality:

- Request for adaptation of MEC resources to one or more MEC applications at runtime. The request for resource adaptation could use the OSContainerDescriptor [i.7] and [i.13] or its attributes such as:
  - requestedCPUResources (number of CPU resources for the container e.g. in milli-CPUs);
  - requestedMemoryResources (amount of memory resources requested e.g. in MB);
  - requestedEphemeralStorageResources (size of ephemeral storage requested for the container e.g. in GB).

Although this key issue was identified by evaluating an XR use case, it may apply to other application verticals requiring runtime MEC resource adjustment.

6.5.2 Solution proposal #5-1, MEC application resource reallocation via an Enabler Service

This solution proposes a method where XR MEC Applications within an XR MEC Application Group deployment can inform the MEC system, via an XR Enabler Service, about their needs for resource allocation adjustment at runtime.

When a MEC application detects performance degradation at an application level, it requests for resource modifications that may impact itself, or other MEC application instances deployed in the XR MEC Application.

Although this solution is based on an XR use case, it may apply to other application verticals requiring runtime MEC resource adjustments.

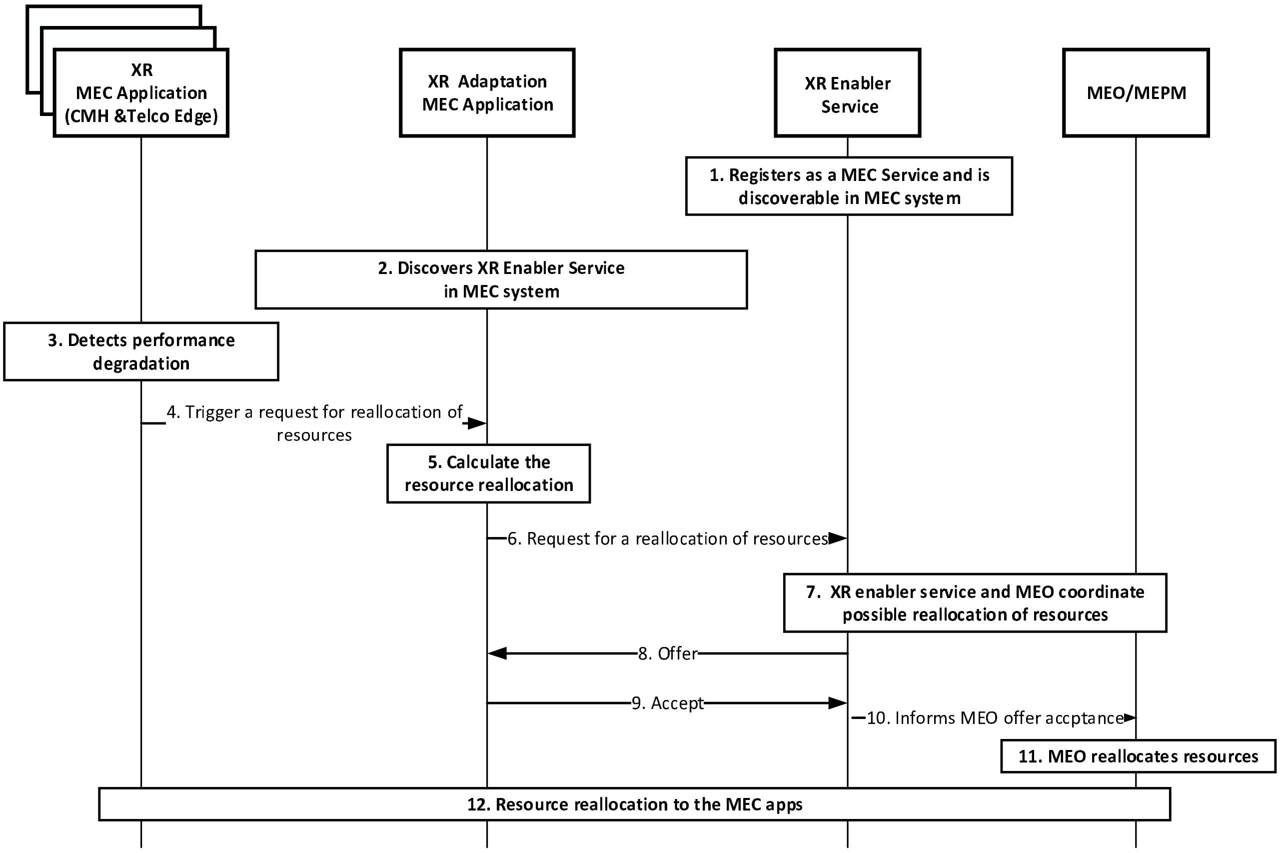

Figure 6.5.2-1: MEC application resource reallocation via an Enabler Service

Prerequisites:

1) An XR MEC Application Group is instantiated within the MEC system as a set of MEC applications deployed on CMHs in the Far Edge and Telco Edge MEC Host.

2) A XR adaptation module realized within a dedicated MEC application is included in the deployment. This MEC application could be deployed on a CMH or a Telco Edge MEC host.

3) An XR Enabler Service is available and authorized to the XR MEC Application Group, specifically the XR adaptation MEC app.

Step 1: The XR Enabler Service registers as a MEC service via MEP and is available in the MEC system to the XR Adaption MEC app.

Step 2: The XR Adaption MEC app discovers the XR Enabler Service in the MEC system via MEP.

Step 3: One or more MEC Applications within the XR MEC Application Group detect a degradation in their performance at an application level. For example, a XR Rendering Service may detect its response time of processing video is higher than acceptable for gameplay.

Step 4: One or more MEC Applications within the XR MEC Application Group trigger a request for a reallocation of resources (e.g. compute or memory) via its XR Adaption MEC app at an application level. This request is outside of MEC scope.

Step 5: The XR Adaption MEC app, using the information from step 4 from one or more XR MEC Applications, determines an optimized resource combination (compute, memory, storage, etc.) for the overall XR MEC Application Group (i.e. all MEC Applications in the group deployment). This may include updates to one or more XR MEC Applications in the MEC Application Group.

Step 6: The XR Adaption MEC app sends a request to the XR Enabler Service for a new runtime combination of resources the each of the XR MEC Applications identified in step 5. Request for a combination of resources include attributes such as:

- requestedCPUResources (number of CPU resources for the container e.g. in milli-CPUs);
- requestedMemoryResources (amount of memory resources requested e.g. in MB);
- requestedEphemeralStorageResources (size of ephemeral storage requested for the container e.g. in GB); etc.

In addition or alternatively, the request could include a particular optimization algorithm and its parameters. These parameters could include the system utility function values, the list of decomposed tasks, their resource combination requirements, and their associated quality values.

NOTE: The details about the optimization algorithm, utility function and parameters to the MEC system and the XR Enabler Service are out of scope of the present document.

Step 7: The XR Enabler Service in coordination with the MEO is unable to meet the resource reallocation request for the XR MEC Applications in step 6, for example utilizing Solution proposal #5-2, MEC application resource availability query.

Step 8: In coordination with the MEO, the XR Enabler Service makes a counteroffer to the XR Adaption MEC app for resource combinations that are currently available for the XR MEC Application Group, for example utilizing Solution proposal #5-2, MEC application resource availability query. The counter-offer attributes may match those in the original request in step 6.

Step 9: The XR Adaption MEC app determines that the counteroffer will satisfy the needs of the MEC Application Group application vertical. It accepts the counteroffer by informing the XR Enabler Service.

Step 10: The XR Enabler Service informs the MEO that the resource reallocation offer has been accepted, for example utilizing Solution proposal #5-3, MEC application resource assignment and notification.

Step 11: The MEO reallocates the resources as agreed to in the previous steps.

Step 12: The impacted XR MEC Applications are informed, for example utilizing Solution proposal #5-3, MEC application resource assignment and notification, that the appropriate resources have been allocated to them and begin updating their internal application behavior accordingly.

6.5.3 Solution proposal #5-2, MEC application resource availability query

This solution proposes a method for the XR Adaption MEC Application and XR Enabler Service to query the MEPM/MEO to discovery the level of MEC resources available for reallocation. This solution may be utilized in Solution proposal #5-1, for example for step 7 to determine if sufficient resources are available and if not to make a counter offer as described in step 8.

Although this solution is based on an XR use case, it may apply to other application verticals requiring runtime MEC resource adjustments.

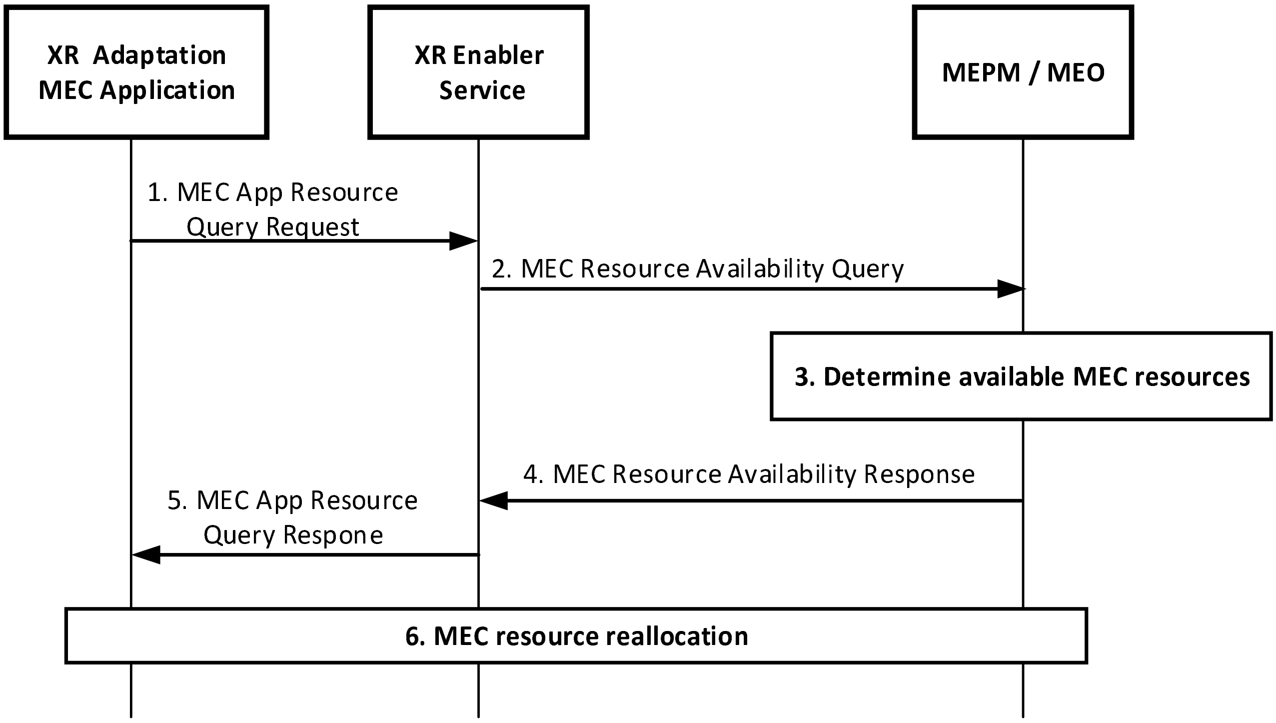

Figure 6.5.3-1: MEC application resource availability query via an Enabler Service

Prerequisites:

1) An XR MEC Application Group is instantiated within the MEC system as a set of MEC applications deployed on CMHs in the Far Edge and Telco Edge MEC Host as described in Solution proposal #5-1.

2) A XR adaptation module realized within a dedicated MEC application as described in Solution proposal #5-1.

3) An XR Enabler Service is available and authorized to the XR MEC Application Group as described in Solution proposal #5-1.

4) The XR adaptation MEC application needs to determine the level of available resources for reallocation to the XR MEC Application Group, as described in step 5 in Solution proposal #5-1.

Step 1: The XR Adaption MEC app sends a query request to the XR Enabler Service to discover what resources may be available for reallocation. The query may request all available resources or may request specific resource available for each MEC app within the XR. Resources attributes include CPU resources, memory resources, storage resources, etc.

Alternatively, the request could include a particular optimization algorithm and its parameters. These parameters could include the system utility function values, the list of decomposed tasks, their resource combination requirements, and their associated quality values.

NOTE: The details about the optimization algorithm, utility function and parameters to the MEC system and the XR Enabler Service are out of scope of the present document.

Step 2: The XR Enabler Service verifies the query request and determines resource availability with the MEPM. The XR Enabler Service sends a MEC resource query to the MEPM, including information on which resources are of interest from step 1. XR Enabler Service can use Mm5 interface to interact with MEPM. Mm5 interface is not standardized in MEC system. Alternatively, XR Enabler Service can use a new interface, which is not available in the MEC system, to interact with MEPM.

Step 3: The MEPM determines resource availability on the MEC host against deployed MEC application instances and their resource reservation or utilization. If the query includes MEC app instances on multiple MEC hosts, the MEPM may forward the query other MEPMs or the MEO.

Step 4: The MEPM responds to MEC resource query from the XR Enabler Service, indicating the level of resources available for run-time reallocation.

Step 5: In turn, the XR Enabler Service responds to XR Adaption MEC app, indicating the level of resources available for run-time reallocation for the XR MEC Application Group.

Step 6: The XR Adaption MEC app utilizes the resource availability information to request for resource reallocation, for example to maintain XR game performance, using Solution proposal #5-1.

6.5.4 Solution proposal #5-3, MEC application resource assignment and notification

This solution proposes a method for the XR Enabler Service to request and coordinate resource modifications with the MEPM and MEO. This solution may be utilized in Solution proposal #5-1, for example in step 10 to inform the MEO on resource reallocations and in step 12 to inform the impacted MEC Applications on their new resource allocations.

Although this solution is based on an XR use case, it may apply to other application verticals requiring runtime MEC resource adjustments.

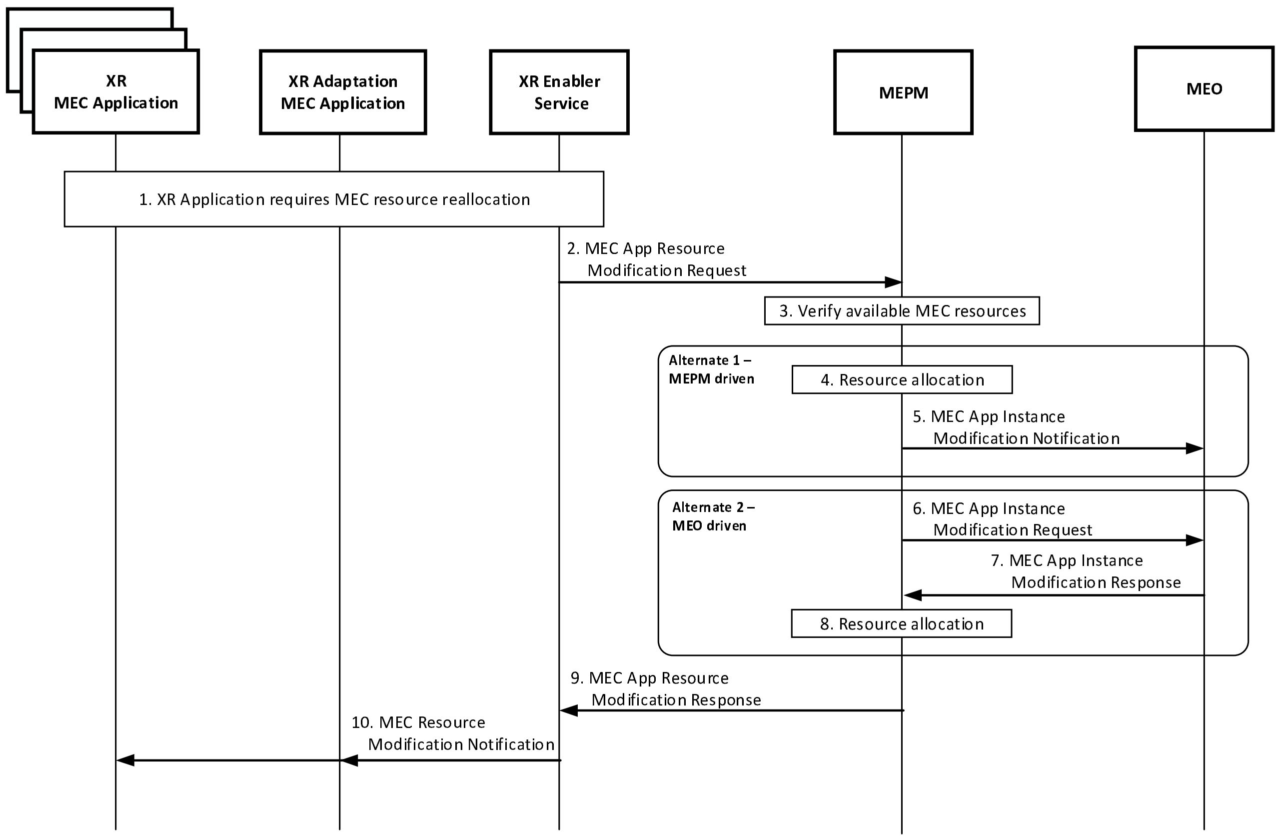

Figure 6.5.4-1: MEC application resource assignment via an Enabler Service

Prerequisites:

1) An XR MEC Application Group is instantiated within the MEC system as a set of MEC applications deployed on CMHs in the Far Edge and Telco Edge MEC Host as described in Solution proposal #5-1.

2) A XR adaptation module realized within a dedicated MEC application as described in Solution proposal #5-1.

3) An XR Enabler Service is available and authorized to the XR MEC Application Group as described in Solution proposal #5-1.

Step 1: One or more of the MEC applications within the XR MEC Application Group detect the need for resource allocation adjustment at runtime, for example to maintain or achieve a certain level of application performance. They coordinate requests for resource reallocation with the XR Adaption MEC app and the XR Enabler Service as described in Solution proposal #5-1. The XR Enabler Service may query to determine if the needed level of resources are available via Solution Proposal #5-2. If resources are available, then the resource modification request is sent in step 2.

Alternatively, the XR Enabler Service may skip this step and directly send resource modification request, as described in step 2. On failure to the modification request, MEPM can respond with an error indication and indicate what level of resources are available, as described in step 9.

The reallocation request may adjust resources of a MEC app within the XR MEC Application Group. Resources include CPU resources, memory resources, storage resources, etc.

Alternatively, the request could include a particular optimization algorithm and its parameters. These parameters could include the system utility function values, the list of decomposed tasks, their resource combination requirements, and their associated quality values.

NOTE 1: The details about the optimization algorithm, utility function and parameters to the MEC system and the XR Enabler Service are out of scope of the present document.

Step 2: The XR Enabler Service sends a MEC application resource modification request to the MEPM, including information on which resources to reallocate from step 1. XR Enabler Service can use the Mm5 interface to interact with MEPM. Mm5 interface is not standardized in the MEC system. Alternatively, XR Enabler Service can use a new interface, which is not available in the MEC system, to interact with MEPM.

Step 3: The MEPM verifies resource availability on the MEC host against deployed MEC application instances and their resource reservation or utilization. If the request includes MEC app instances on multiple MEC hosts, the MEPM may forward the request to the MEO.

If the MEPM determines that there are sufficient resources to satisfy the request, two options may be used to this satisfy the resource modification request: 1) MEPM driven in Steps 4 - 5, where the MEPM modifies the resources per the request and informs the MEO; 2) MEO driven in Steps 6-8, where the MEPM checks with the MEO before reallocating resources. The selection between utilizing the MEPM driven option or the MEO driven option is out of scope for the present document.

Alternate Option 1 - MEPM driven

Step 4: The MEPM sends a resource allocation request to the VIM, with the modified resources. VIM allocates the resources according to the request of the MEPM.

Step 5: The MEPM sends a MEC app instance modification notification to the MEO, including information on the modified resources. MEPM can use the Mm3 interface with the MEO.

Alternate Option 2 - MEO driven

Step 6: Before adjusting any resource allocations, the MEPM sends a MEC app instance modification request to the MEO, including information on the requested resources for modification. MEPM can use the Mm3 interface with the MEO.

Step 7: MEO verifies the modification request and acknowledges acceptance by sending a MEC app instance modification response to the MEPM. MEPM can use the Mm3 interface with the MEO.

Step 8: The MEPM sends a resource allocation request to the VIM, with the modified resources. VIM allocates the resources according to the request of the MEPM.

Step 9: The MEPM sends MEC application resource modification response to the XR Enabler Service, indicating success and that resources are reallocated per the request in step 2.

Step 10: The XR Enabler service sends a MEC resource modification notification to the XR adaptation module, indicating the reallocated resources. The XR adaptation module informs the impacted XR MEC app regarding their resource updates. Alternatively, XR Enabler service may send a MEC resource modification notification directly to each impacted XR MEC app.

The XR MEC Applications adjust their internal application behaviour according to the new resource allocation to achieve the needed level of application performance (from step 1).

NOTE 2: The current assumption for ETSI GS MEC 010-2 [i.7], where sufficient resources would be reserved when a MEC application is instantiated, is not applicable for the solution proposal #5-3.

6.5.5 Evaluation

Solution #5-1 describes a method where XR MEC Applications within an XR application vertical deployment can inform the MEC system, via an XR Enabler Service, about their needs for resource allocation adjustment at runtime.

The solution is technically feasible, if the following conditions are met:

- XR MEC Application Group can be instantiated within the MEC system as a set of MEC applications instantiated on CMHs in the Far Edge and/or Telco Edge MEC Host. The MEC applications in the group are pre-provisioned, available and authorized to be accessed by other MEC applications within the group.
- XR Enabler Service can register as an MEC service via MEP and is available in the MEC system to the XR Adaption MEC app.
- XR Adaptation MEC app can inform XR Enabler service, for a new runtime combination of resources for each of the XR MEC Applications in the XR MEC Application Group.
- XR Enabler service can negotiate with MEO about the new runtime resource requirement (e.g. as described in solution #5-2) and provide a counteroffer to XR Adaptation MEC app.
- XR Adaptation MEC app can accept the offer and inform XR Enabler service.
- XR Enabler Service can inform MEO that the resource reallocation offer has been accepted, for example utilizing Solution proposal #5-3.

MEO can reallocate resources and inform the impacted XR MEC Applications (e.g. as in solution #5-3).

Solution #5-2 proposes a method for the XR Adaption MEC Application and XR Enabler Service to query the MEPM/MEO to discovery the level of MEC resources available for reallocation.

The solution is technically feasible, if the following conditions are met:

- XR MEC Application Group can be instantiated within the MEC system as a set of MEC applications instantiated on CMHs in the Far Edge and/or Telco Edge MEC Host. The MEC applications in the group are pre-provisioned, available and authorized to be accessed by other MEC applications within the group.
- XR Adaptation app can query XR Enabler service to discover what resources are available for reallocation.
- XR Enabler service can query the MEPM for resource availability, indicating the resources of interest.
- Resource related information is available in MEPM and can provide resource information to XR Enabler Service.
- MEPM can obtain information about MEC app instances on multiple MEC hosts, by querying the MEO.
- XR enabler service can respond to XR Adaptation app with resource availability information.
- XR Adaption MEC app utilizes the resource availability information to support resource reallocation.

Solution #5-3 proposes a method for the XR Enabler Service to request and coordinate resource modifications with the MEPM and MEO.

The solution is technically feasible, if the following conditions are met:

- XR MEC Application Group can be instantiated within the MEC system as a set of MEC applications instantiated on CMHs in the Far Edge and/or Telco Edge MEC Host. The MEC applications in the group are pre-provisioned, available and authorized to be accessed by other MEC applications within the group.
- XR Enabler Service can register as an MEC service via MEP and is available in the MEC system to the XR Adaption MEC app.
- XR Adaptation MEC app can coordinate with XR Enabler service about resource requirement for each of the XR MEC Applications in the XR MEC Application Group.

- XR Enabler Service can send MEC application resource modification request to the MEPM, including information on which resources to reallocate.
- MEPM can verify resource availability on the MEC host and if required can forward the request to MEO.
- MEPM can send resource allocation request to the VIM, with the modified resources and inform MEO about the modification.
- Alternatively, MEPM can inform MEO about the requested resources for modification. MEO verifies the modification request and acknowledges acceptance. MEPM can send a resource allocation request to the VIM, with the modified resources. VIM allocates the resources according to the request of the MEPM.

MEPM can send MEC application resource modification response to the XR Enabler Service, which informs XR adaptation module, indicating the reallocated resources.

6.6 Gap/Key issue #6 - Maintaining sufficient XR Quality of Experience in the face of Bandwidth Fluctuation.

6.6.1 Description

As discussed in clause 5.3.3.1, XR applications, especially when running on constrained devices, are impacted by the variability in the bandwidth of network links and other conditions. As consequence, for example when bandwidth is low or latency is high, the quality of an XR game will decrease, even to the point of being unplayable.

As network bandwidth fluctuates, an XR game (including its various individual MEC Applications deployed within a XR MEC Application Group on CMHs and the Telco Edge) may adapt its behaviour, which may in turn impact the XR MEC Application Group's need for MEC resources. For example, the XR game may increase or decrease video resolution and frame rate. This may lower or increase its need for CPU, memory, storage, or other MEC resources. Under such conditions, the XR MEC Application Group may utilize Solution #5-1 in clause 6.5.2 to request for MEC resource reallocation at run time. However, this may occur after the user is experiencing degraded XR performance. Even to the point where the game is unplayable, and the user simply stops using it.

Speculation is a useful technique to predict future MEC resource needs ahead of time, so they may be available to MEC applications or MEC Application Groups, before user experience or performance is impacted. This would potentially avoid XR performance degradation and keep the user engaged with the XR game, even when faced with fluctuating network conditions. Currently, speculation is a common technique utilized in XR applications to deal with varying conditions while maintaining sufficient gameplay experience for users. For example, speculation is used to predict video frames and player movements to reduce compute load, time to render video, and deal with low-bandwidth conditions. Speculation within the XR MEC Application Group may impact MEC resource needs. To illustrate using the multi-player game shown in Figure 5.3.1-3, a video client MEC app may decide to speculate video frames due to low-bandwidth conditions with a XR video server MEC app. This speculation would increase the MEC resource needs of the video client MEC app.

MEC resource speculation may be done within a MEC Application Group to predict its future MEC resource needs based on application behavior, information from available MEC services or the MEP, etc. However, this prediction is bound by information available within a single MEC app or within a MEC Application Group deployment (i.e. MEC app instances in the group). In this study, this prediction is classified and referred as "low-fidelity" speculation since it has limited visibility to a single MEC app or MEC Application Group.

The MEC system can provide additional MEC resource speculation by considering information available within the system. For example, the prediction of MEC resources for one XR MEC Application Group may be complimented by considering MEC resource allocations of other XR MEC Application Group deployments, availability of resources in the MEC system, availability of MEC hosts and their locations, etc. This will improve or better maintain MEC application performance compared to using low-fidelity speculation alone. Since this prediction considers additional parameters and information in the MEC system, it is classified and referred to as "high-fidelity" speculation in this study.

To provide runtime MEC resource speculation, the MEC system can expose an enhanced XR Enabler Service, as presented in Solution #5-1 which enables MEC applications to request adjustments in their runtime MEC resource allocations. An enhanced XR Enabler Service can expose a service that provides high-fidelity speculation of MEC resources and, in turn, proactively adjust these allocations for the MEC application.

With needed resources available, the MEC application will continue to provide a user with sufficient XR quality of experience, even when facing dynamic conditions, such as access network bandwidth fluctuation.

Although this key issue was identified by evaluating an XR use case, it may apply to other application verticals requiring runtime MEC resource adjustment and where MEC resource speculation is possible.

6.6.2 Solution proposal #6-1, Future reallocation of MEC runtime resources using speculation.

This solution proposes a method where an XR MEC application and the MEC system coordinate on the speculation of MEC resources to meet the performance needs of the application. This solution predicts the need for MEC resource reallocation based on application-level considerations and system insights. It ensures that MEC resources are best allocated ahead of need to maintain sufficient application performance and avoid XR experience degradation.

Although this solution is based on an XR use case, it may apply to other application verticals requiring runtime MEC resource adjustments.

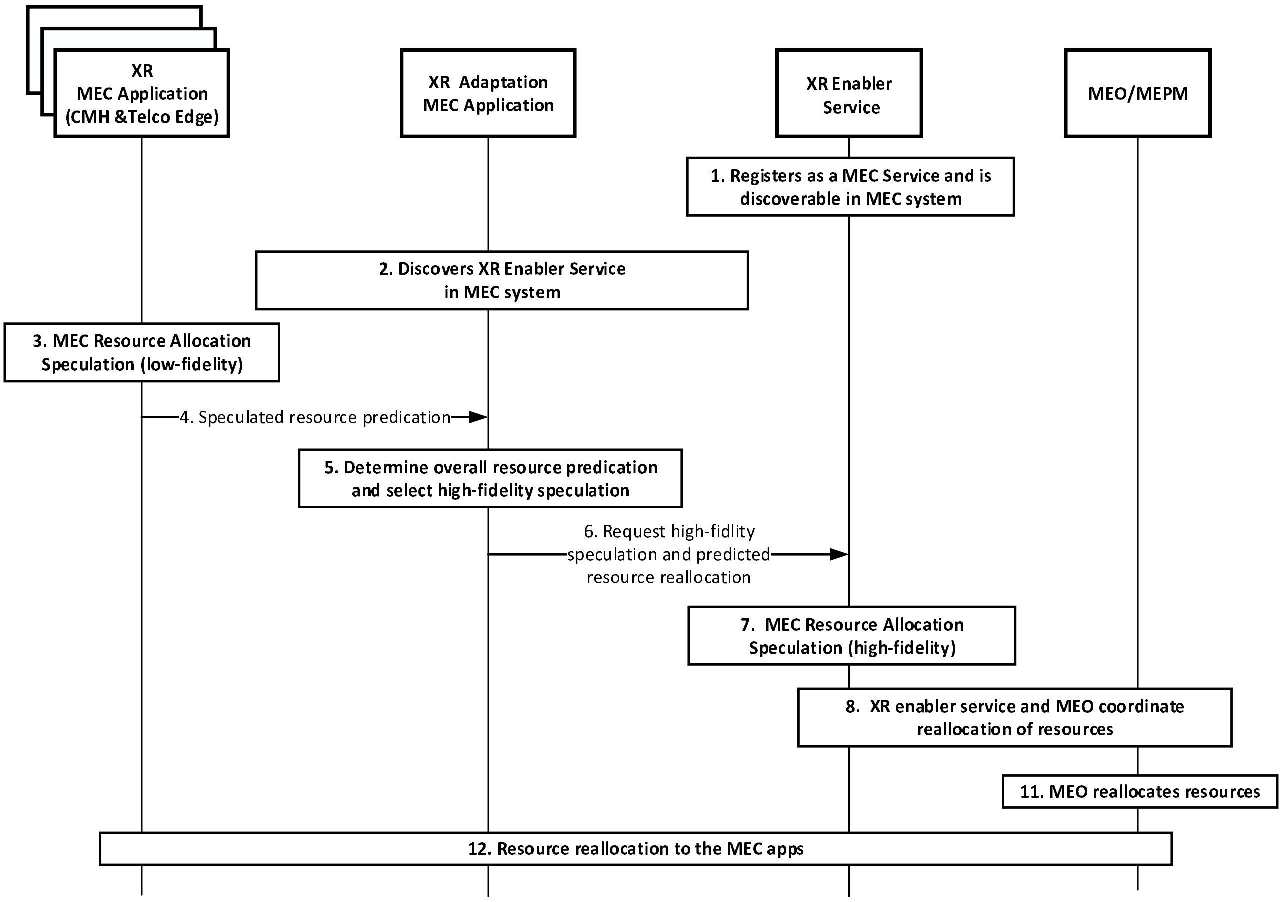

Figure 6.6.2-1: Future reallocation of MEC runtime resources using speculation

Prerequisites:

1) An XR MEC Application Group is instantiated within the MEC system as a set of MEC applications deployed on CMHs in the Far Edge and Telco Edge MEC Hosts.

2) A XR adaptation module, realized within a dedicated MEC application, is included in the deployment. This MEC application could be deployed on a CMH or a Telco Edge MEC host.

3) An XR Enabler Service is available and authorized to the XR MEC Application Group, specifically the XR adaptation MEC app.

4) The XR MEC Applications in the MEC Application Group deployment are able to perform "low-fidelity" speculation for future MEC resources based on their limited visibility to a single MEC app instance, single MEC host, etc.

5) The XR Enabler Service can provide "high-fidelity" MEC resource speculation by considering a wider set of considerations and parameters, including its visibility of the MEC system (resource requests from other XR MEC Application Group deployments, current and predicted resource allocations, etc.).

Step 1: The XR Enabler Service registers as a MEC service via MEP and is available in the MEC system to the XR Adaptation MEC app.

Step 2: The XR Adaptation MEC app discovers the XR Enabler Service in the MEC system via MEP.

Step 3: One or more MEC Applications within the XR MEC Application Group may perform speculation at an application level, for example predicting upcoming video frames or player actions. Additionally, one or more of the XR MEC Applications may perform speculation on their MEC resource needs based on application-level considerations, input from MEC Service APIs (Location, RNIS, etc.), etc. MEC resources include those that can be reallocated at runtime as is described in Solution #5-1 (e.g. compute, memory, storage, etc.). This speculation is a low-fidelity prediction due to limited visibility to a single MEC application, single MEC host, etc.

If this prediction indicates the MEC application's resource needs are within its current allocation, then no action needs to be taken and this procedure ends.

Step 4: A XR MEC app sends its low-fidelity speculation prediction for MEC resource reallocation to the XR Adaptation MEC application. The resource prediction could be a point estimate or a probability distribution. This request is at an application level and is outside of MEC scope.

Step 5: The XR Adaptation MEC app, using the information from step 4 from one or more XR MEC applications, considers the MEC resource speculation (compute, memory, storage, etc.) for the overall XR MEC Application Group (i.e. all MEC Applications in the application vertical deployment). This may include predicting the needed MEC resource allocation for one or more XR MEC Applications in the MEC Application Group in order to maintain sufficient XR experience for the user. If the predicted MEC resource need is within current MEC system allocation, then no action needs to be taken and this procedure ends.

The XR Adaptation MEC app may consider factors such as game quality, network performance thresholds, or other criteria when deciding to request additional high-fidelity speculation of MEC resources from the XR Enabler Service. For example, it may utilize MEC services to query for network information (RNIS, WAIS, etc.) to determine if sufficient latency is available. If network performance towards the XR Enabler Service is insufficient for high-fidelity speculation, the XR Adaptation MEC app may:

1) Request the MEC system to reallocate resources with the "low-fidelity" prediction using Solution #5-1 in clause 6.5.2. Afterward, this procedure ends.

2) Or, retain the current MEC resource allocation, and this procedure ends.

This step takes place within the XR Adaption MEC app and is outside of MEC scope.

Step 6: If network performance is sufficient for high-fidelity speculation in step 5, the XR Adaptation MEC app sends a request to the XR Enabler Service for high-fidelity MEC resource runtime speculation. It provides the resource combination prediction determined in step 5. The predicted resource combination includes similar attributes as in Solution #5-1, including:

- requestedCPUResources (number of CPU resources for the container e.g. in milli-CPUs);
- requestedMemoryResources (amount of memory resources requested e.g. in MB);
- requestedEphemeralStorageResources (size of ephemeral storage requested for the container e.g. in GB);
- Etc.

The predicted resource combination may be a point estimate of the MEC resource attributes. Alternatively, it may be a probability distribution (e.g. memory between 8 MB to 10 MB with 98 % probability).

Step 7: The XR Enabler Service combines the low-fidelity resource prediction from step 6 with its visibility of the MEC system (resource requests from other XR app deployments, current and predicted resource allocations, etc.) to produce a high-fidelity resource prediction (i.e. high-fidelity speculation). If the high-fidelity resource prediction is within the current MEC system allocation, then no action needs to be taken and this procedure ends.

Step 8: The XR Enabler Service coordinates with the MEO to reallocate MEC resources for the XR MEC application vertical using the high-fidelity resource prediction, for example utilizing Solution proposal #5-2 to determine available resources and Solution proposal #5-3 to reallocate MEC application resources. This will ensure sufficient resources are available before the application experiences any XR performance degrade towards the user.

Step 9: The MEO reallocates the resources as coordinated to in the previous step.

Step 10: The impacted XR MEC Applications are informed, for example utilizing Solution proposal #5-3, MEC application resource assignment and notification, that the appropriate resources have been allocated to them and begin updating their internal application behavior accordingly (i.e. avoiding XR user experience degradation).

6.6.3 Evaluation

Solution #6-1 describes a method where an XR MEC application and the MEC system coordinate on the speculation of MEC resources to meet the performance needs of the application. This solution predicts the need for MEC resource reallocation based on application-level considerations and system insights. The solution co-ordinates with MEC system to allocate MEC resources ahead of need to maintain sufficient application performance and avoid XR experience degradation.

The solution is technically feasible, if the following conditions are met:

- XR MEC Application Group can be instantiated within the MEC system as a set of MEC applications instantiated on CMHs in the Far Edge and/or Telco Edge MEC Host. The MEC applications in the group are pre-provisioned, available and authorized to be accessed by other MEC applications within the group.
- XR Enabler Service can register as a MEC service via MEP and is available in the MEC system to the XR Adaption MEC app.
- XR Adaptation MEC app can discover and inform XR Enabler service, about predicted resource combination for all the MEC Applications in the XR MEC Application Group.
- XR Enabler service can produce a high-fidelity resource prediction for all the MEC Applications in the XR MEC Application Group.
- XR Enabler Service can coordinate with the MEO to reallocate MEC resources for the XR MEC application vertical using the high-fidelity resource prediction (e.g. as in solutions #5-2 and #5-3).
- MEO can reallocate resources and inform impacted MEC applications. (e.g. as in solution #5-3).

6.7 Gap/Key issue #7 - Reducing processing and signalling among MEC Applications, which are components of an application

6.7.1 Description

Clause 5.3.3.1 describes how XR application components, realized as distributed MEC applications on CMHs and Telco MEC hosts, may increase signalling exchange and processing within MEC system. The application components can provide and consume services between each other using MEC platform enablement services.

In ETSI GS MEC 011 [i.4], as part of "**Service availability update and new service registration**" procedure:

- When a MEC application instance (i.e. an XR application component), producing a service, becomes available for the first time (via service registration), the **MEC platform notifies the authorized, relevant MEC application instances** (e.g. MEC applications that indicate the service(s) as "optional" or "required") about the newly available service(s).

- A service producing MEC application instance updates the MEC platform about status changes of the produced MEC services; and the **MEC platform notifies all authorized, relevant MEC application instances** about the service availability changes.

- In the XR use case, where MEC applications (as components of the XR application) are running on other MEC hosts, the MEC platform identifies the relevant MEC platforms for these updates and informs about new service availability notifications or changes in service availability.

- The XR application components are MEC applications deployed in different hosts. These updates and notifications will be exchanged over communication and network links.

As described in ETSI GS MEC 011 [i.4], MEC applications are bound to a transport that is either provided by the MEC platform, or by the application itself. Examples of transports are REST-HTTP, and message passing systems that support the Publish-Subscribe mode for the communication between MEC application component instances and the MEC platform, or between MEC application component instances.

Exchange of messages over transports or message passing system (such as pub/sub), that need traverse through communication network links, may not be efficient. Network-related issues such as latency, network partitions, and the risk of lost messages can significantly impact performance. These can lead to unpredictable and varied service performance. Pub/Sub systems can introduce latency due to the message routing and delivery process. Minimizing latency while maintaining scalability and reliability can be challenging, especially in real-time applications, such as XR, where low latency is critical.

Implementing a Pub/Sub architecture can introduce complexity. Managing subscriptions, message routing, and ensuring consistency across distributed components require careful design and management. It increases processing to manage subscription, derive/calculate routes, etc.

To reduce signalling exchanges over communication links between CMHs and MEC hosts in Telco Edge, solutions are proposed where the messages from the CMH are managed locally, abstracting out from the MEC Application component in Telco Edge.

6.7.2 Solution proposal #7-1, Managing signalling from application components on CMH locally

6.7.2.1 Description

In this solution, a XR game specific MEC Application, (e.g. Remote Component Aggregator, RCA), is proposed. The RCA registers services with MEC platform to collect and process notifications from remote MEC Application components deployed on CMH. RCA can expose information about available APIs from the remote MEC application components. The MEC application components are pre-provisioned as part of the XR application.

Figure 6.7.2-1 shows an example deployment of RCA in CMH and Telco Edge.

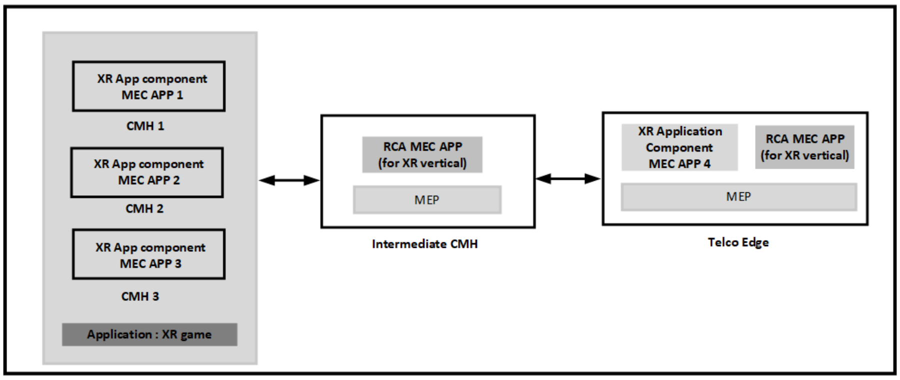

Figure 6.7.2-1: Example deployment of MEC Application components in CMH and Telco Edge

As shown in Figure 6.7.2-1, XR Application components MEC App1, MEC App2, MEC App3 are deployed on a CMH. These application components are considered as remote application components. The XR Application component MEC APP4 is deployed on a Telco Edge MEC host. The application components can consume and produce services among themselves. For example, MEC App4 in Telco Edge consumes service from MEC App1, MEC App2 and MEC App3 on the CMH.

RCA can register MEC services, on a CMH and in the Telco Edge. In the above diagram, a CMH, named "Intermediate CMH", hosts the RCA MEC app. The RCA MEC App registers MEC service with the MEC platform in Intermediate CMH. The RCA deployed in Intermediate CMH, collects and process information about the APIs produced by the remote application components and notifications from remote Application components MEC App1, MEC App2 and MEC App3.

All remote MEC application components, which are part of the XR application, inform the RCA (e.g. deployed in an Intermediate CMH) about the produced service APIs. The MEC application components also inform RCA, about the "service availability" and "service availability changes".

RCA registers MEC service (producer) with the MEC Platform in an Intermediate CMH. MEC Application component (i.e. MEC App4) in Telco Edge discovers RCA services via the MEC Platform. RCA provides information about APIs produced by remote MEC application components to the MEC application component in Telco Edge. It is assumed that RCA will maintain list of APIs produced by remote application components and manage "availability status changes" of the remote MEC application components locally.

RCA can also be deployed in Telco Edge and can register with the MEC platform in Telco Edge. The RCA for an XR application in Telco Edge can have a counterpart RCA MEC App in CMH. The RCA in CMH can update the RCA in Telco Edge about available APIs and availability status. This interaction is out of scope of the MEC system. The MEC application component in Telco Edge (e.g. MEC App4) will discover the local RCA (i.e. RCA MEC App deployed in Telco Edge) and interact with it locally.

6.7.2.2 RCA deployed in Intermediate CMH

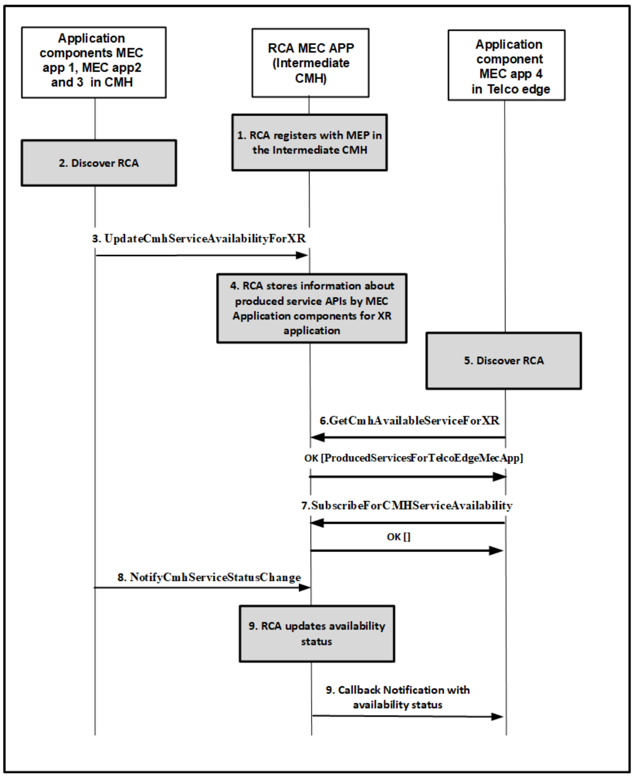

Figure 6.7.2.2-1: Use of RCA produced services, deployed in Intermediate CMH, to reduce signalling

RCA, registers MEC services in Intermediate CMH, can be used to collect and process availability information of produced APIs by MEC application components e.g. MEC App1, MEC App2 and MEC App3 deployed on a CMH. RCA can be a generic service producer or it can support a specific MEC Application Group or MEC application.

E.g. in this solution, RCA supports the XR game. RCA processes the availability information of the produced APIs, consolidates them, and informs MEC application component MEC App4 in Telco Edge.

Step 1: RCA registers MEC service (producer) with MEP in the Intermediate CMH, indicating support for XR application. If XR application components MEC App1, MEC App2, MEC App3, hosted in CMH, indicated RCA for XR as "optional" or "required", then these applications will be notified through "new service notification" from the MEP.

RCA produces following services:

- Collect availability of produced services by application components in CMH, to be exposed to other application components (*UpdateCmhServiceAvailabilityForXR*).
- Publish available services produced by application components in CMH to other application components in Telco Edge (*GetCmhAvailableServiceForXR*).
- Collectavailability status of services produced by application components in CMH (*NotifyCmhServiceStatusChange*).
- Subscription mechanism by MEC application component in Telco Edge to get notified about availability status change of produced services in CMH.

Step 2: XR application components MEC App1, MEC App2, MEC App3, hosted in CMH, can send a request to query the availability of RCA service to the CMH MEP. (*Assuming MEC App1, MEC App2, MEC App3 indicated RCA as "optional" or "required"*). MEC platform in the CMH, responds with the message body containing the information about the RCA service, including the information needed to access RCA service.

Step 3: MEC application components (MEC App1, MEC App2, MEC App3, hosted in CMH) uses the service "*UpdateCmhServiceAvailabilityForXR"* to inform RCA about the produced services and their availability information. E.g. a data type can be defined, "**ProducedServicesByApplicationComponents**", which include:

Table 6.7.2.2-1: ProducedServicesByApplicationComponents

Attribute name	Data type	Cardinality	Description
serviceName	Structure (inlined)	0..1	Name of the MEC application component deployed in CMH
>*hostLocality*	String	0..1	Host of service producing MEC application component, e.g. Telco Edge, CMH
>*producedServiceAPI*	String/URI	0..N	API of the produced services by MEC application components
>*availabilityStatus*	Boolean	0..N	0 = Not available for consumption 1 = Available for consumption
>*exposeToTelcoEdge*	Boolean	0..N	0 = Not exposed to App component in Telco Edge 1 = Exposed to application component in Telco Edge (see note)
NOTE: Some services produced by application components in CMH is for consumption by application components in CMH only. In some scenarios, application component may not want the availability status to be managed by RCA.			

Step 4: RCA stores or updates the information received in "**ProducedServicesByApplicationComponents**".

Step 5 is similar to step 2, where XR application components MEC App4 hosted in Telco Edge, can send a request to query the availability of RCA produced services for the XR application. (*Assuming MEC App4 indicated RCA as "optional" or "required"*). MEC platform in the Telco Edge, responds with the message body containing the information about the RCA produced services, including the information needed to access RCA service.

Step 6: MEC application component MEC App4, hosted in Telco Edge, uses the service "*GetCmhAvailableServiceForXR"* to obtain information from RCA about the available services produced by MEC application component in CMH.

MEC App4 sends the following data type with the request, "**QueryForCmhServices**".

Table 6.7.2.2-2: QueryForCmhServices

Attribute name	Data type	Cardinality	Description
applicationGroupName	String	0..1	MEC Application Group which sends the query, e.g. XR gaming application
serviceName	String	0..1	Application component, which sends the query, e.g. MECApp4
hostLocality	String	0..1	Host of the application component, which initiated the query e.g. Telco Edge, CMH, host

RCA provides a list of produced services, which are available and indicated the service can be exposed to Telco Edge, to MEC App4. RCA selects produced services by application components in CMH, from "**ProducedServicesByApplicationComponents**",

- Where "*hostLocality*" set to CMH
- Which are available, as indicated by the attribute "*availabilityStatus*"
- Which can be informed to MEC application component in Telco Edge, based on the attribute "*exposeToTelcoEdge*".

RCA sends OK response to MEC App4, including a data type "**ProducedServicesForTelcoEdgeMecApp**" in the body, as shown in Table 6.7.2.2-3.

Table 6.7.2.2-3: ProducedServicesForTelcoEdgeMecApp

Attribute name	Data type	Cardinality	Description
serviceName	Structure (inlined)	0..1	Name of the MEC application component deployed in CMH
>producedServiceAPI	String/URI	0..N	API of the produced services by MEC application components

Step 7: MEC App4 subscribes with RCA, to be notified about service availability status for the produced services by MEC application components hosted in CMH. E.g. MEC App4 can use the data type "**AvailabilityStatusNotificationSubscription**", to subscribe with RCA.

Table 6.7.2.2-4: AvailabilityStatusNotificationSubscription

Attribute name	Data type	Cardinality	Description
applicationGroupName	String	0..1	MEC Application Group which sends the query, e.g. XR gaming application
serviceName	String	0..1	Application component, which subscribes for availability notification, e.g. MECApp4
hostLocality	String	0..1	Host of the application component, which subscribes for availability notification e.g. Telco Edge, CMH, host
callBackUrl	String/URI	0..1	Callback for notification, to send notification to the app component (e.g. MEC App4), which subscribed for availability status update or availability status change
notificationConfiguration	Structure {}	0..1	Application component, which subscribes for notification, specifies how notifications can be sent to it, e.g. Frequency, consolidation level, duration, etc.

RCA verifies the "serviceName" and "hostLocality". If the application component is allowed to subscribe for service availability notification of application components in CMH, RCA accepts the subscription and sends OK, indicating success or failure of the subscription.

Step 8: MEC application components, informs RCA about any status changes, by using "*NotifyCmhServiceStatusChange*", including the data type, "**AppComponentStatusChangeUpdate**", as shown in Table 6.7.2.2-5.

Table 6.7.2.2-5: AppComponentStatusChangeUpdate

Attribute name	Data type	Cardinality	Description
serviceName	Structure (inlined)	0..1	Name of the MEC application component deployed in CMH
>producedServiceAPI	String/URI	0..N	API of the produced services by MEC application components
>availabilityStatus	Boolean	0..N	0 = Not available for consumption 1 = Available for consumption

Step 9: RCA updates and stores availability information from MEC application components, received in step 8. RCA updates the data type, "**ProducedServicesByApplicationComponents".**

Step 10: RCA checks for any active subscription. E.g. MEC App4 subscribed for notification about MEC application component availability status. RCA checks for notification configuration and use the call back URI, received in "AvailabilityStatusNotificationSubscription"**,** to notify MEC App4 about change in status of MEC application components in CMH.

E.g. data type for call back notification, "**ServiceAvalabilityUpdate**"as shown in Table 6.7.2.2-6.

Table 6.7.2.2-6: ServiceAvalabilityUpdate

Attribute name	Data type	Cardinality	Description
availabilityStatus	Structure (inlined)	0..N	Information about availability status of services produced by MEC application component deployed in CMH
> serviceName	String	0..1	Name of the MEC application component deployed in CMH
> availability	Boolean	0..1	0 = Not available for consumption 1 = Available for consumption
causeForStatusChange	Enum	0..1	e.g. Mobility, compute unavailability, etc.
applicationActions	Enum	0..1	e.g. Continue, Pause, Restart

6.7.3 Evaluation

Solution #7-1, describes a MEC application, called "RCA (Remote Component Aggregator)", which registers services with MEC platform, to support:

- Information collection about available produced services by MEC application components, which are pre-provisioned to be part of a MEC Application Group.
- Information exposure about produced service APIs from MEC application components to other MEC application components, where the application components are pre-provisioned to be part of a MEC Application Group.
- Processing of notifications from MEC Application components about the "service availability" and "service availability changes".
- Notifying other MEC application components about service availability changes.

The solution is technically feasible, if the following conditions are met:

- RCA can be specific to a MEC Application Group and application component from the same group can discover and use the services produced by RCA.
- Application components in CMH and Telco Edge can discover an RCA in another CMH, including its end point information. As part of discovery, application component becomes aware of the service APIs produced by RCA.
- Application component can use the RCA service APIs to update about available services, which are produced by the application component and allowed to be managed by RCA.
- Application component can use the RCA service API to inform about the service availability status and any change to the service availability status of the produced services by the application component.
- Application component can subscribe to RCA for notification about changes to availability status or other control information related to the operation of the MEC Application Group based on availability status.
- RCA can inform about availability status of services produced by application components to other application components, which subscribed to receive notification.
- Application components can configure the notification mechanism in RCA, the details of the notification, level of details requested, etc.

6.8 Gap/Key issue #8 - Transfer of a MEC application instance between Constrained MEC Hosts

6.8.1 Description

Clause 5.3.1 describes how an XR game may be realized as several components composing a MEC Application Group, such as:

- A Display Client that presents XR video to an individual player. The Display Client may be realized as a Client Application that uses a XR Rendering MEC app in the MEC system.
- An XR Rendering MEC app that generates video for a player based on the individual player's viewpoint and provides it to the Display Client.
- A Game Coordinator or Game Server MEC app that considers actions and context from all players and provides information to each player's XR Rendering MEC app to generate appropriate video for that player.

Furthermore, clause 5.3.3.4 evaluated the impact of MEC application transfer when deployed on a Constrained MEC Host within the XR gaming MEC Application Group. For example, when the battery level of a CMH ran low, the XR Render MEC application needs to be transparently (or seamlessly) transferred from that CMH to another host that can satisfy gameplay experience requirements of the XR user.

Three solutions are proposed:

- Proposal #8-1 presents a solution where a Device Application can initiate transfer of a MEC application context from one CMH to another.
- Proposal #8-2 is a supporting proposal to enhance MEC-016 Location Constraints with information sufficient for CMH deployments, similar to Solution proposal #3-3.
- Proposal #8-3 presents a solution where the transfer of a MEC App user context information from a MEC App instance in a CMH to a MEC App instance in another MEC Host is initiated by the XR Enabler Service upon the detection of low battery.

6.8.2 Solution proposal #8-1, Device App initiated MEC application transfer on a Constrained MEC host

This solution proposes a method where an XR Display Client Application may initiate the transfer of an XR Rendering MEC Application from one MEC host to a second MEC host, in order to maintain the needed user experience for the XR gaming player. It ensures the user is experiencing sufficient XR gaming application performance.

Although this solution is based on an XR use case, it may apply to other application verticals requiring transfer of MEC applications or user application contexts when deployed on CMHs.

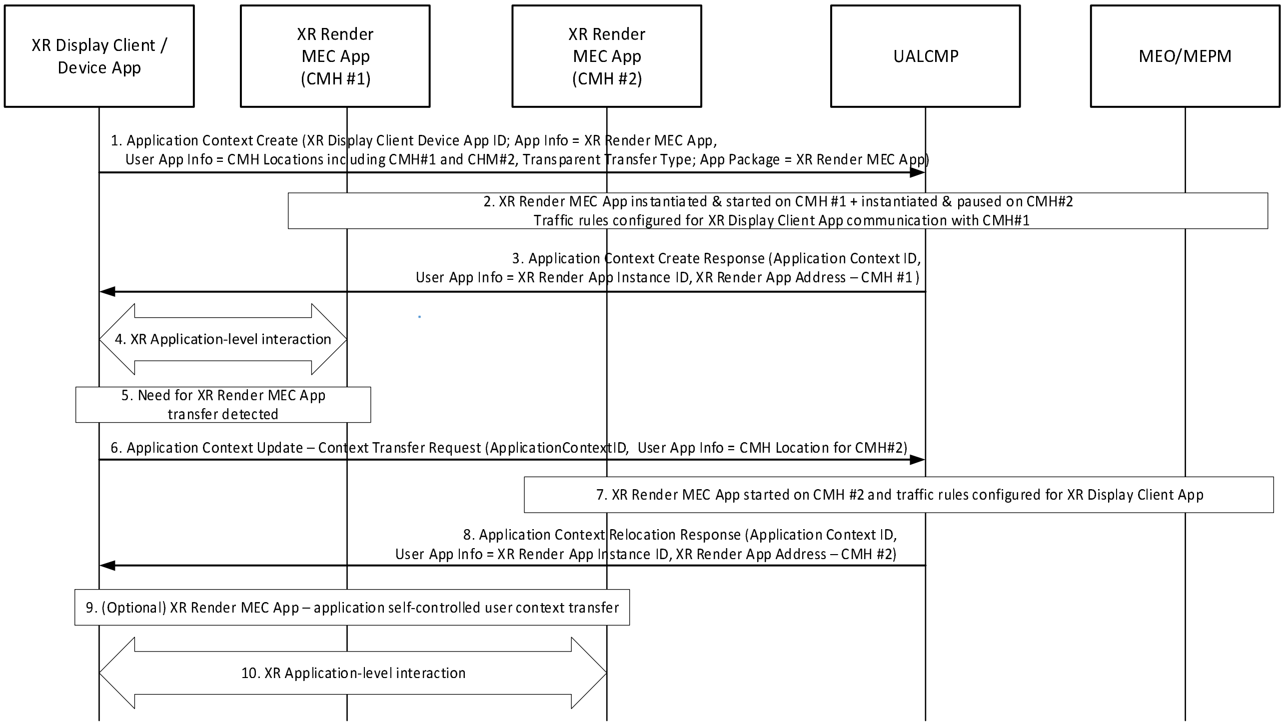

Figure 6.8.2-1: Client/Device App initiated MEC application transfer on a Constrained MEC host

Prerequisites:

1) An XR Display Client application is on a user device and uses the service of an XR Render MEC application to acquire XR video for display to an individual player. The XR Display Client may be considered as a client application and a device application as defined in ETSI GS MEC 016 [i.14].

2) The XR Render MEC application is a user application in the MEC system.

3) An XR Render MEC application instance is not deployed on a CMH or Telco Edge host for the individual player.

4) The XR Display Client application may be aware of nearby CMHs in its local area, for example CMHs within a home or within a gaming arena (e.g. CMH#1 and CMH#2).

Step 1: The XR Display Client requests creation of a User Application Context for the XR Render MEC app via the UALCMP, providing App Context information to instantiate the XR Render MEC app, including:

- XR Display Client Device App ID.
- XR Render MEC App Information: AppD ID, App Name, AppD version, etc.
- XR Render App Instance Information: including location constraints that indicate relevant CMHs as follows:
  - Deployment and start on CMH#1.
  - Deployment on CMH#2 (paused and available for application context transfer).
  - Location Constraints indicate CMH location - see solution #8-2.

The XR Display Client also indicates that it may request transparent transfer of the XR Render MEC application context to another host in the MEC system. This informs the MEC system to configure traffic rules with an underlying multi-access network, so the XR Display Client maintains communication with the XR Render MEC application after transfer to another host.

Step 2: The MEO/MEPM deploys an instance of the XR Render MEC app to CMH #1 and CMH#2, based on the CMH Location Constraints in step 1 and configures traffic rules for communication with the XR Display Client App. Traffic rule configuration is specific to an underlying multi-access network and is out of MEC system scope. For a MEC system that support fast MEC app deployment (e.g. serverless virtualization), the MEO/MEPM may defer deployment to CMH #2 until a context transfer is initiated.

Step 3: The UALCMP responds to the Application Context request from the XR Display Client in step 1, indicating:

- Application Context ID for the created context with the XR Render MEC app
- XR Render MEC App's Instance ID on CMH#1 and CMH#2
- XR Render MEC App's reference address for the XR Display Client to access the XR Render MEC app on CMH #1

Step 4: The XR Display Client and XR Render MEC app on CMH #1 interact with each other at an application level, with the XR Render MEC app providing video the XR Display Client (which is out of scope the MEC system).

Step 5: The XR Display Client, independently or via interaction with other XR MEC Applications, decides that the XR Render MEC app needs to be transferred to an alternative CMH (CMH #2) in order to maintain XR application overall performance. This determination is at an application-level and is out of scope for the MEC system.

Step 6: The XR Display Client requests the MEC system via UALCMP to transfer its User Application Context with the XR Render MEC app to CMH #2. The XR Display Client performs a User Application Context Update with the UALCMP, providing:

- Application Context ID - received in step 3.
- Updated XR Render App Instance Information: updated App Location Constraints:
  - Requesting transfer to CMH #2 (informing to start instance, if paused).
  - See solution #8-2 for CMH location.

If the XR Display Client is aware of additional CMHs, it may include them in the XR Render App Instance Information (location constraints = deployed, paused, and available for context transfer). Additionally, the XR Display Client may remove CMHs that it deems inappropriate for its performance or other requirements.

Step 7: The MEO/MEPM starts the XR Render MEC app instance on CMH#2. Alternatively, for a MEC system that supports fast MEC app deployment or transfer (e.g. serverless virtualization), the MEO/MEPM may transfer the XR Render MEC app instance to CHM#2 or deploy a new instance of the XR Render MEC app to CMH #2. MEO/MEPM configures traffic rules for the XR Render MEC App on CMH#2 to maintain communication with the XR Display Client.

Step 8: The UALCMP responds to Application Context transfer request from the XR Display Client in step 6, indicating:

- Application Context ID for the XR Render MEC app (this is unchanged).
- XR Render MEC App's Instance ID on CMH#2.
- XR Render MEC App's reference address for the XR Display Client to access the XR Render MEC app on CMH #2. Since transparent transfer is indicated, the reference address may be maintained from step 3.

Step 9: (Optionally) The XR Client App and XR Render MEC app instances may execute an application self-controlled user context transfer from the XR Render MEC app instance on CMH #1 to the XR Render MEC app instance on CMH #2 without assistance from the MEC system.

Step 10: The XR Display Client and XR Render MEC app on CMH #2 interact with each other at an application level, with the XR Render MEC app providing video the XR Display Client (which is out of scope the MEC system).

6.8.3 Solution proposal #8-2, Constrained MEC Host Location Information in ETSI GS MEC 016

ETSI GS MEC 016 [i.14] defines constraints for MEC application requirements related to MEC application deployment location in clause 6.5.2 - Type: LocationConstraints.

MEC host location constraint attributes include:

- as a country Code
- as a civic address combined with a country code
- as an area, conditionally combined with a county code

These location constraints are not sufficient for Constrained MEC Hosts, which require greater location granularity than Telco Edge MEC hosts for MEC application requirements. To create and manage MEC application user app contexts on Constrained MEC hosts (for a groups or otherwise for a single MEC app instance), the attributes for ETSI GS MEC 016 [i.14] Location Constraints may need to consider:

- Network Location Information, such as zone information, connected access point information, etc. (as defined in ETSI GS MEC 013 [i.15]).
- Additional Geographic Location Information, such as latitude, longitude, altitude, additional shape information (such as area circle), etc., (as defined in ETSI GS MEC 013 [i.15]).
- Indoor Location Context Information, such as floor number, room number, aisle, etc., (as defined in in ETSI GS MEC 013 [i.15]).
- Relative Distance Information, for example between CMHs or between CHM and terminal devices (as defined in ETSI GS MEC 013 [i.15]).
- CMH identifier(s).
- CMH Mobility Path Information.

6.8.4 Solution proposal #8-3, XR Service Enabler initiated MEC application transfer on a Constrained MEC Host

This solution proposes a method to transfer the user context information from a XR MEC App instance in a CMH with a low battery, referred to as CMH #X, to the XR MEC App instance in another MEC Host with sufficient battery to ensure XR application availability and performance. The solution considers that the XR MEC App instance in CMH #X detects a battery shortage in CMH #X and indicates it to the XR Enabler Service, introduced in clause 6.5.1. The XR Enabler Service, together with the MEC system (MEO/MEPM), identifies a target MEC Host and a second XR MEC App instance in the target MEC host to transfer the XR MEC App user context information. This transfer requires that an instance of the XR MEC App is running in the target MEC Host, so in the case where an instance is not available, a new XR MEC App instance is instantiated in the target MEC Host. Then, the user context information of the XR MEC App at CMH #X is transferred to the XR MEC App instance in the target MEC Host, following the application self-controlled user context transfer procedure in [i.11].

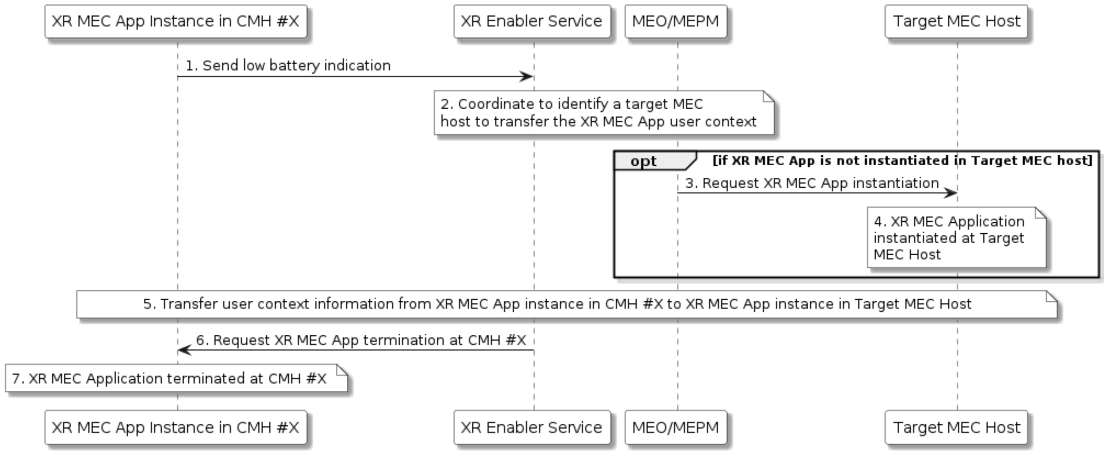

Figure 6.8.4-1: XR MEC App user context information transfer through the XR Enabler Service due to battery shortage detection

Prerequisites:

- An XR MEC Application Group is instantiated within the MEC system as a set of MEC Applications deployed on CMHs in the Far Edge and Telco Edge MEC Host.
- One of the components of the XR application, referred to as XR MEC App instance, is instantiated and running on a constrained MEC host #X, denoted as CMH #X. The XR MEC App instance can either be the XR Rendering service or the Video server.
- An XR Enabler Service is available and authorized to the XR MEC Application Group.
- The battery status of CMH #X is accessible by the XR MEC App instance in the CMH and the indication of low battery to the XR Enabler Service is provided by using XR Enabler Service APIs.

Step 1: The XR MEC App instance in CMH #X sends a low battery indication to the XR Enabler Service upon the detection of battery shortage. This detection can be based on a pre-defined threshold defined at the application level.

Step 2: The XR Enabler Service coordinates with the MEO/MEPM to identify a target MEC Host to transfer the XR MEC App instance's user context information. This target host could be another CMH with sufficient available battery or a Telco Edge MEC host.

NOTE 1: The interaction between the XR Enabler Service and the MEO would be done through the MEPM, supported by the Mm5 and Mm3 reference points. The specific coordination between the MEO and MEPM to determine the target MEC Host is FFS.

Alternate option - If XR MEC App instance is not instantiated in target MEC Host, continue with step 3. Otherwise, proceed to step 5.

Step 3: The MEO requests the instantiation of the XR MEC application on the selected target MEC Host.

Step 4: The XR MEC application is instantiated on the target MEC host.

Step 5: User context information is transferred from the XR MEC App instance in CMH #X to the XR MEC App instance in the target MEC host. This information could include the current application state, user settings, and other data necessary to ensure a seamless transition.

NOTE 2: This user context information transfer is assumed to be done according to the application self-controlled user context transfer procedure in [i.11], which assumes that the application is able to execute the context transfer without assistance from MEC system. The specific interactions between the XR enabler service, the CMH #X and target MEC Host would be given by internal implementation of the XR application, so they are out of the scope of the present document.

Step 6: The XR Enabler Service requests the termination of the XR MEC application on CMH #X.

Step 7: The XR MEC application on CMH #X is terminated.

6.8.5 Evaluation

Solution #8-1 proposes a method, where the XR Display Client Application interacts with UALCMP to create context for XR Render MEC app and provides a list of CMH where the XR Render MEC app instances can be deployed. MEO/MEPM deploys XR Render MEC app and configures traffic rule. MEC system, on receiving request from XR Display Client through UALCMP, transfers user application context to XR Render MEC app to CMH #2 and starts the XR Render MEC app instance on CMH#2.

The solution is technically feasible, if the following conditions are met:

- MEC system allows a set of MEC applications instantiated on CMHs in the Far Edge and/or Telco Edge MEC Host. The MEC applications are pre-provisioned, available and authorized to be accessed by other MEC applications.
- XR Display Client Application can interact with UALCMP to request creation of a User Application Context for the XR Render MEC app, providing application context information to instantiate the XR Render MEC app.
- UALCMP can provide the application context information to MEC system, MEO/MEPM can deploy XR Render MEC app on MEC hosts and configure traffic rules.
- UALCMP can inform XR Display Client Application about the Application Context ID and XR Render MEC App instance IDs.
- XR Display Client can determine that the XR Render MEC app needs to be transferred to an alternative CMH, based on application criteria.
- XR Display Client can request the MEC system via UALCMP to transfer its User Application Context with the XR Render MEC app to the alternate CMH and can update the user application context.
- MEO/MEPM can start XR Render MEC app on alternate CMH and configure traffic rules for communication with the XR Display Client Application.

UALCMP can respond to XR Display Client to inform about the application context ID, MEC App instance ID, and the reference address of XR Display client on alternate CMH.

Solution #8-2 analyses ETSI GS MEC 016 [i.14], which defines constraints for MEC application requirements related to MEC application deployment location, to determine if it is sufficient for constrained MEC hosts.

The solution proposes additional location constraints, which are required for constrained MEC hosts, to be considered in ETSI GS MEC 016 [i.14]. The proposed CHM location constraints in ETSI GS MEC 016 [i.14] are consistent with proposed CHM location information for ETSI GS MEC 013 [i.15].

The solution is technically feasible, if the following conditions are met:

- ETSI GS MEC 016 [i.14] location constraints can include Network Location Information, such as zone information, connected access point information.
- ETSI GS MEC 016 [i.14] location constraints can include Additional Geographic Location Information, such as latitude, longitude, altitude, additional shape information.
- ETSI GS MEC 016 [i.14] location constraints can include Indoor Location Context Information, such as floor number, room number, aisle.
- ETSI GS MEC 016 [i.14] location constraints can include Relative Distance Information, for example between CMHs or between CHM and terminal devices.
- ETSI GS MEC 016 [i.14] location constraints can include CMH identifiers and CMH Mobility Path Information.
- ETSI GS MEC 016 [i.14] location constraints do not conflict with the location constraint information as described in solution #3-3, Location constraints in ETSI GS MEC 010-2 [i.7].

Solution #8-3 proposes a method, where an XR MEC App instance in CMH #X detects a battery shortage in CMH #X and indicates it to the XR Enabler Service. The XR Enabler Service, together with the MEC system (MEO/MEPM), identifies a target MEC Host and a second XR MEC App instance in the target MEC host to transfer the XR MEC App user context information. If an instance is not available, a new XR MEC App instance is instantiated in the target MEC Host. Then, the user context information of the XR MEC App at CMH #X is transferred to the XR MEC App instance in the target MEC Host.

The solution is technically feasible, if the following conditions are met:

- MEC system allows a set of MEC applications instantiated on CMHs in the Far Edge and/or Telco Edge MEC Host. The MEC applications are pre-provisioned, available and authorized to be accessed by other MEC applications.
- XR Enabler Service is available and authorized to serve other MEC applications.
- The XR Enabler Service, after receiving low battery indication, can coordinate with the MEO/MEPM to identify a target MEC Host with sufficient battery power, to transfer the XR MEC App instance's user context information to another XR MEC App instance running on the target MEC host.
- The XR Enabler Service coordinates with the MEO/MEPM to identify a target MEC Host and can determine that XR MEC App instance is not available on the target MEC host, then MEO can request instantiation of the XR MEC application on the selected target MEC host.
- XR Enabler service can request termination of the XR MEC application on the first CMH.

7 Conclusion

7.1 Description

The present document describes various use cases to study MEC in constrained devices. Two use cases:

1) smart factory of the future; and
2) multi player XR multimodal mobile gaming define key issues and propose potential solutions based on an analysis of the current ETSI MEC architecture.

This clause will conclude the study of these use cases and provide recommendations.

The conclusion and recommendations are not meant to influence 3GPP work but to align ETSI MEC work with 3GPP.

7.2 Smart Factory of the future use case

The use case assumes use of MEC system to realize the use case. Three main areas of the use case are identified for further analysis.

The three main areas are:

- Use of MEC in constrained devices. For example, constrained devices can be assumed to be a MEC host and exposes standard MEC interfaces. Constrained devices with a MEC host are termed as constrained MEC Hosts.
- Telco Edge Far Edge Interaction, how to mitigate loss of user plane of the constrained MEC host.
- Mobility of constrained device, adding mobile constrained MEC hosts dynamically.

In summary, all recommendations are technically feasible but may depend on the work of other SDOs such as 3GPP SA2 or 3GPP SA6.

Table 7.2-1 summarizes the "Smart Factory of the future" use case areas, gaps, evaluation, and key issues.

Table 7.2-1: Summary of use case areas, gaps, evaluation and key issues

Use case areas	Gap Analysis	Evaluations and corresponding key issues
Use of MEC in constrained devices	• Impact of unstable connection between constrained MEC host and MEC system on Management Plane and User Plane interaction • Impact of unstable connection between two constrained MEC hosts on Management Plane and User Plane • Impact of unstable connection between an application client and the application server instance	Constrained devices can handle temporary loss of Management Plane by relying on the current configuration. Prolonged loss of Management Plane can be assumed by the MEC system as a loss of the constrained MEC host and MEC system is reconfigured to select a different constrained MEC host. Analysis of scenarios involving loss of User Plane, while Management Plane exists or not, lead to the Key Issue #1. **Gap/Key issue #1 - Loss of Management Plane and User Plane connection of a constrained device**
Telco Edge Far Edge Interaction	• Handling unstable connection between Far Edge and a Telco Edge Support for communication among specific MEC application instances (of same or different MEC applications) hosted in Far Edge and Telco Edge	Loss of Management Plane between Constrained device and Telco Edge can be mitigated. But losing User Plane can have a big impact on the application and needs to be mitigated. This led to Key Issue #4. **Gap/Key issue #4 - Telco Edge Far Edge (constrained device) Interaction** The smart factory use case can be assumed to be realized through the deployment of a set of MEC application instances in Telco Edge and Far Edge. The MEC application instances in a set can be of different MEC applications. The MEC application instances in a set can be removed or added dynamically. How MEC system can support deployment of a set of MEC application instances led to the Key Issue #3. **Gap/Key issue #3 - Support for a group of MEC application instances deployed on specific set of hosts**
Mobility of constrained device	• How to allow a MEC application instance hosted in a constrained MEC host to join or leave a deployment set (as described in KI#3), as the constrained MEC host moves or as triggered by authorized users, such as supervisor on the factory floor?	A Constrained MEC Host (CMH), which was previously unavailable, becomes available, e.g. a mobile CMH comes in the network coverage of an MNO. MEC applications in a CMH cannot be discovered, consumed or produce service, until the CMH is registered and management interfaces to MEC system has been setup. This led to the Gap/Key Issue #2. **Gap/Key issue #2 - Support for dynamic addition of MEC hosts (to a MEC system)**

The mapping of the key issues to their associated solutions is provided in Table 7.2-2, highlighting identified gaps within the current scope of ETSI MEC.

Table 7.2-2: Summary of key issues, solution, gaps and recommendations

Key issue	Clause	Solution	Gaps within current ETSI MEC scope and recommendations
Gap/Key issue #1 - Loss of management plane and User Plane connection of a constrained device	6.1.2	Solution proposal #1-1, Detecting loss of user plane	The solution addresses that a MEC application wants to ensure or learn that the loss of interaction with a MEC service is due to loss of user plane and no other reasons like compute overload. The capability for a MEC application to know the reason for loss of interaction with a remote MEC service, is not supported in the current MEC system.

Key issue	Clause	Solution	Gaps within current ETSI MEC scope and recommendations
			The solution proposes MEC Application obtains information about the loss of interaction from MEP. This interaction can be supported over Mp1 interface. ETSI GS MEC 011 [i.4] can be enhanced to support this capability.
	6.1.3	Solution proposal #1-2, Finding other MEC App to continue service	The solution assumes MEO has the knowledge that a MEC Application instance is consuming service from an instance of MEC Service and the interaction is lost because of unavailability of user plane. MEO can search for an equivalent MEC service instance which is available and can provide service to the MEC application. This approach is different from what is defined in ETSI MEC reference architecture, where a MEC application discovers a MEC service. However, this procedure may be internal to MEO operation and does not impact any ETSI MEC interfaces. Informing about the availability of a MEC service instance, which the MEC application instance should use, can be implemented over Mp1 interface. ETSI GS MEC 011 [i.4] can be enhanced to support this capability.
Gap/Key issue #2	6.2.2	Solution proposal #2-1, Constrained MEC Host	The solution describes how to operate a constrained device, without a MEC platform. The solution proposes use of a MEP proxy function in the constrained device. An interface between MEP proxy function and an off-device MEP to forward Mp1 messages is described. The solution does not impact the MEC architecture. The interface between MEP proxy function and the off-device MEP can be considered as an implementation option and may require further consideration when additional scenarios like one-to-many and/or many to many interactions are considered.
	6.2.3	Solution proposal #2-2, Dynamic MEC Host Capability Exposure Options	Constrained MEC hosts can have different capabilities. The solution describes list of options, which indicates different levels of MEC capability for constrained MEC hosts. A MEC system can use this information to manage constrained MEC hosts. This solution enables solutions #2-3, #2-4 and #2-5. Within the scope of the ETSI MEC architecture, a MEC host with different capabilities is not considered. The management system is not aware of managing MEC hosts with different capabilities. ETSI GS MEC 002 [i.6] can be updated to include the requirements for MEC hosts with different capabilities. ETSI GS MEC 003 [i.1] can be evaluated further to identify if the assumption of MEC host with different capabilities impacts the existing architecture and procedures.

Key issue	Clause	Solution	Gaps within current ETSI MEC scope and recommendations
	6.2.4	Solution proposal #2-3, Configuration of MEC Host Capability in a MEC System by authorized user	The solution describes how an MEO can implement a host management function to manage constrained MEC host information. Constrained MEC host information can be provided by an authorized user over (Mx1) to the OSS. OSS can use Mm1 interface to update MEO with constrained MEC host information. Mx1 and Mm1 is not standardized and open to implementation, hence there will be no impact within the scope of ETSI MEC specifications. Authorized users can use the host capability options as described in solution #2-2 to provide information about a constrained MEC host to MEO. The capability option of a constrained MEC host can be defined within the scope of ETSI MEC. The standard data model can be used across MEC system, e.g. by CMH, MEO, Host management system and Application Life cycle management.
	6.2.5	Solution proposal #2-4, Dynamic MEC Host Capability Registration with a MEC System, initiated by MEO	The solution describes how MEO initiates capability registration and integration of a mobile constrained MEC host into the MEC system. MEO can contact the constrained MEC host to authenticate and request host's capability information. Constrained MEC host responds with host capability information to MEO. ETSI MEC architecture supports interaction between MEO and MEC host through MEPM. Mm2 ETSI GS MEC 010-1 [i.22] reference point support platform configuration capability, allowing the OSS to configure the mobile edge host. Mm2 can be enhanced to support configuring MEC platform with different capabilities, including configuring MEC management interfaces at run time. Initial discovery and communication between CMH and MEC System is not possible, because management interfaces, such as Mm5, are not available. Initial discovery can be implemented over Mm5, but it is not standardized. Alternatively, Mx2 ETSI GS MEC 016 [i.14] (interface can be used to support initial discovery and handshaking between CMH and OSS or MEO, using Device applications. The Mx2 interface terminates to an Application Lifecycle management proxy, which can be enhanced to support host capability registration and host integration. Evaluating the use of Mx2, ETSI GS MEC 016 [i.14] for discovery and registration of CMH is further recommended.

Key issue	Clause	Solution	Gaps within current ETSI MEC scope and recommendations
	6.2.6	Solution proposal #2-5, Dynamic MEC Host Capability Registration with a MEC System, initiated by Constrained MEC Host	The solution describes a constrained MEC host dynamically discovering a MEC system and informs the MEC system about which MEC host capability exposure options it may support. MEC host discovering and contacting MEO for registering capability information and the MEO responding back to MEC host is not supported in the current scope of the ETSI MEC architecture. The analysis for the previous solution, applies for this solution too. It is recommended to evaluate the use of Mx2, ETSI GS MEC 016 [i.14] for initial discovery and registration of CMH.
Gap/Key issue #3	6.3.1	Solution proposal #3-1, MEO supported group creation and configuration of constrained MEC hosts	Solution #3-1 is an independent solution, describes how MEO can create and manage a group of MEC application instances based on application service provider requirements. MEO obtains requirements from an application service provider and creates a group of MEC application instances. MEO creates a group configuration information. The creation of group configuration information is internal to MEO operation and does not impact MEC system. MEO provides the group configuration information to the MEC application instances, which are part of the group. MEO contacting a MEC application instance to provide group information is not supported in the current scope of the ETSI MEC architecture. ETSI GS MEC 003 [i.1] describes Mm3 and Mm5 interfaces, which can be used by MEO to reach the MEP through MEPM. MEO can provide group configuration information to MEP. MEP can update MEC application instances about group configuration information over Mp1 interface. Since Mm5 is not specified, the information exchange between MEO and MEP can be assumed internal function. The Mp1 interface can be enhanced to support sharing of group configuration information between MEC application instance and MEP. Alternatively, Mx2 interface can be used to support the interaction between a MEC application instance and MEO. Evaluating the enhancement of Mp1 interface for sharing group information is further recommended.

Key issue	Clause	Solution	Gaps within current ETSI MEC scope and recommendations
	6.3.2	Solution proposal #3-2, Group operation through a Group management Function	Solution #3-2 describes a MEC application function, called "Group Management Function (GMF)", to support group creation and group discovery. GMF can be a MEC service available in the MEC Host, including constrained MEC hosts. Within the scope of the ETSI MEC architecture, creating a group of MEC application instances and group management function is not considered. ETSI GS MEC 002 [i.6] can be updated to include the requirements for managing group of MEC application instances and group management feature can be an optional feature. Additional functions and interactions can be described as part of the feature enablement. The solution #3-2 is recommended because it allows dynamic group creation, modification to handle mobile constrained MEC hosts.
	6.3.3	Solution proposal #3-3, Constrained MEC Host Location Information in ETSI GS MEC 010-2 [i.7]	Solution #3-3 proposes updates to the MEC application requirements related to MEC application instance deployment location in ETSI GS MEC 010-2 [i.7]. It is recommended to enhance ETSI GS MEC 010-2 to include the proposed location granularity information.
	6.3.4	Solution proposal #3-4, Constrained MEC Host Capability Information in ETSI GS MEC 010-2 [i.7]	Solution #3-4, concludes that the existing ETSI GS MEC 010-2 [i.7] data model information for MEC application requirements, including the Application Descriptor, is sufficient to specify constrained MEC host capabilities. No impact on ETSI GS MEC 010-2 [i.7].
	6.3.5	Solution proposal #3-5, Group operation through MEC Application Slice management function	The solution considers a group a MEC application instances as a slice and proposes use of a slice management function to create a group based on the ETSI GR MEC 044 [i.12] architecture "a" configuration (MAS-CSMF within OSS and MAS-MF within MEO). Dynamic update and modification of a slice to manage mobile constrained MEC hosts may not support real time constraints. This solution is not recommended.
Gap/Key issue #4	6.4.2	Solution proposal #4-1, Configuration of an Intermediate Device Relay Function to Support Constrained MEC Host Updates towards Telco Edge	Solution #4-1 is an independent solution, not related to solutions #4-2 and #4-3. The solution utilizes an application-data relay function, which may complement access network solutions that provide connectivity level relay capabilities. The relay function can be a MEC application deployed in constrained MEC hosts and Telco Edge. The relay function in an intermediate device relays data at application level between constrained MEC host and Telco Edge and vice versa.

Key issue	Clause	Solution	Gaps within current ETSI MEC scope and recommendations
			The solution proposes that MEO detects the loss of user plane between constrained MEC host and Telco Edge. MEO selects an intermediate device with the relay function and has connectivity with the MEC hosts. This function is internal to MEO operation and do not impact ETSI MEC architecture. MEO configures the selected relay function at run time. To configure the relay function in intermediate device, MEO should be able to interact with the intermediate device. ETSI GS MEC 003 [i.1] describes Mm3 and Mm5 interface, which can be used by MEO to reach MEC application through MEPM and MEP. This interface can be enhanced to support the capability registration procedure. Alternatively, Mx2 interface, ETSI GS MEC 016 [i.14], can be used to support the interaction between constrained MEC host and MEO. This may require updating ETSI GS MEC 016 [i.14], for configuration of relay function. This solution is not recommended because interaction between MEO and MEC application is not supported in ETSI MEC architecture.
	6.4.3	Solution proposal #4-2, Relay Function Registration with a Relay Discovery MEC Service	Solutions #4-2 and #4-3 are related and these two combined, provide another solution to support relay function. The solution proposes the relay function to be realized as a MEC service. The relay MEC service can be offered by constrained MEC hosts in intermediate devices. The relay MEC service can be discovered by another MEC service called relay discovery service. The relay function can be a feature, optionally supported by the MEC system. The feature is enabled by the use of Solutions #4-2 and #4-3, which may be realized as MEC services. ETSI GS MEC 002 [i.6] can be updated to include the requirements for relay function. ETSI GS MEC 003 [i.1] can be evaluated further to identify additional functions that may be required to enable the feature. Platform enablement APIs as described in ETSI GS MEC 011 [i.4] can be updated to support the feature. The details need further study.

Key issue	Clause	Solution	Gaps within current ETSI MEC scope and recommendations
	6.4.4	Solution proposal #4-3, Discovery, Configuration, and Use of an Intermediate Device, Only Relay Function to Support Constrained MEC Host Updates towards Telco Edge	This solution describes the operation and interaction among MEC applications, relay function, relay discovery service from solution #4-2. This is dependent on solution #4-2 and acceptance of the feature. Hence same set of recommendation applies as in solution #4-2. ETSI GS MEC 002 [i.6] can be updated to include the requirements for relay function. ETSI GS MEC 003 [i.1] can be evaluated further to identify additional functions that may be required to enable the feature. Platform enablement APIs as described in ETSI GS MEC 011 [i.4] can be updated to support the feature. The details need further study.

7.3 Multi player XR multimodal mobile gaming use case

The use case assumes use of MEC system to realize the use case. XR MEC applications are distributed across different hosts in Cloud, Telco Edge and on devices with constrained MEC hosts. The Constrained MEC hosts can host XR MEC Applications, which execute XR tasks and can be managed by MEC reference points. Analysis of the ETSI MEC architecture to realize this use case identified two main areas for study.

The two main areas are:

- Area #1: Impact of resource constraints on XR MEC applications hosted in constrained MEC hosts.
- Area #2: Distribution of MEC applications in one or more constrained MEC hosts.

In summary, all recommendations are technically feasible but may depend on the work of other SDOs such as 3GPP SA2 or 3GPP SA6.

Table 7.3-1 summarizes the "Multi player XR multimodal mobile gaming" use case areas, gaps, evaluation, and key issues.

Table 7.3-1: Summary of use case areas, gaps, evaluation and key issues

Use case areas	Gap Analysis	Evaluations and corresponding key issues
Area #1: Impact of resource constraints on XR MEC applications	• Impact of varying resource requirements of an XR MEC application, hosted in constrained MEC host and Telco Edge, on the QoE experienced by user • Maintaining sufficient XR quality of experience in the face of fluctuating bandwidth	When MEC resource requirement goes up, resource allocation can be optimized by re-allocating additional resources which are under-utilized. On the other hand, when MEC resource requirement goes down, resources can be reclaimed back from a XR MEC application. The issues identified were: • Communicating the need to modify MEC resources (either increase or decreasing allocation) to the MEC system by MEC applications. • Modification of MEC resources allocation to XR tasks (i.e. MEC applications), when resource requirements change. This led to key issue #5. **Gap/Key issue #5 - Resource allocation and re-distribution to XR MEC application(s)** **XR MEC application hosted on a constrained MEC host needs to adapt and consider changing characteristics of communication.** This problem can be addressed by speculation, i.e. generating data in advance. How MEC system can enable an XR application by supporting speculation capabilities, utilizing its knowledge of available resources to mitigate the fluctuating bandwidth problem led to key issue #6. **Gap/Key issue #6 - Maintaining sufficient XR Quality of Experience in the face of Bandwidth Fluctuation**

Use case areas	Gap Analysis	Evaluations and corresponding key issues
Area #2: Impact of MEC application distribution across several MEC hosts	How Application components distributed as MEC Applications, among several MEC hosts, impacts discovery and management in a MEC system?	Distribution of MEC Applications, which are components of an application, among many constrained MEC hosts, increases signalling exchanges between MEC host in Telco Edge and constrained MEC hosts. Processing overhead increases as MEC system stores information about the distributed application components, deployed as MEC applications. How to reduce processing overhead and multiple signalling communication among MEC Applications, which are components of an application, deployed in Telco Edge and CMH lead to key issue #7. **Gap/Key issue #7 - Reducing processing and signalling among MEC Applications, which are components of an application** CMH, where MEC applications are deployed, may run low on battery capacity, computing power, or other constraints. As a result, and in order to maintain the player's XR quality of experience, the XR MEC app needs to be transparently relocated to another MEC host (CMH or Telco MEC Host). The issues which need to be addressed are transparent transition or relocation of a MEC application instance from a CMH to another CMH or a Telco MEC host and vice-versa. It also needs to be addressed how a MEC application or an application client can inform the MEC system about a need for transparent relocation. This led to key issue #8. **Gap/Key issue #8 - Transfer of a MEC application instance between Constrained MEC Hosts**

The mapping of the key issues to their associated solutions is provided in Table 7.3-2, highlighting identified gaps within the current scope of ETSI MEC.

Table 7.3-2: Summary of key issues, solution, gaps and recommendations

Key issue	Clause	Solution	Gaps within current ETSI MEC scope and recommendations
Gap/Key issue #5 - Resource allocation and re-distribution to XR MEC application(s)	6.5.2	Solution proposal #5-1, MEC application resource reallocation via an Enabler Service	Solution #5-1 describes a MEC application function, called "XR Enabler Service", to support other MEC application's need for resource allocation adjustment at runtime. XR Enabler Service can be a MEC service available in the MEC Host, including constrained MEC hosts. The "XR Enabler Service" interacts with MEO for run time resource allocation adjustment. Within the scope of the ETSI MEC architecture, resource allocation adjustment at runtime is not considered. ETSI GS MEC 002 [i.6] can be updated to include the requirements for run time resource allocation adjustment, which can be an optional feature. Additional functions and interactions can be described as part of the feature enablement.

Key issue	Clause	Solution	Gaps within current ETSI MEC scope and recommendations
			MEC application contacting MEO for resource allocation adjustment is not supported in the current scope of the ETSI MEC architecture. ETSI GS MEC 003 [i.1] describes Mm3 and Mm5 interfaces, which can be used by MEO to reach the MEP through MEPM and vice versa. MEC application can send resource allocation adjustment request to MEP, to be forwarded to MEO. MEO can provide resource allocation adjustment information to MEP. MEP can update MEC application instances about resource allocation adjustment information over Mp1 interface. Since Mm5 is not specified, the information exchange between MEPM and MEP can be assumed internal function. ETSI GS MEC 10-2 [i.7] over Mm3 reference point can be enhanced to support resource allocation adjustment request and response between MEPM and MEO. ETSI GS MEC 011 [i.4] over Mp1 reference point can be enhanced to support the interaction between MEC application and MEP for resource allocation adjustment request and response.
	6.5.3	Solution proposal #5-2, MEC application resource availability query	The solution proposes a MEC application, e.g. XR Adaption MEC Application and XR Enabler Service query the MEPM/MEO to discover the level of MEC resources available for reallocation. Gaps and recommendations for solution #5-2, is same as solution #5-1. MEC application contacting MEO for resource allocation adjustment is not supported in the current scope of the ETSI MEC architecture. ETSI GS MEC 002 [i.6] can be updated to include the requirements for run time resource allocation adjustment. Same proposed enhancements to ETSI GS MEC 10-2 [i.7] over Mm3 and ETSI GS MEC 011 [i.4] over Mp1 reference points are recommended.
	6.5.4	Solution proposal #5-3, MEC application resource assignment and notification	This solution proposes a method for the XR Enabler Service to request and coordinate resource modifications with the MEPM and MEO. Gaps and recommendations for solution #5-3, is same as solution #5-1. MEC application contacting MEO for resource allocation adjustment is not supported in the current scope of the ETSI MEC architecture. ETSI GS MEC 002 [i.6] can be updated to include the requirements for run time resource allocation adjustment.

Key issue	Clause	Solution	Gaps within current ETSI MEC scope and recommendations
			Same proposed enhancements to ETSI GS MEC 10-2 [i.7] over Mm3 and ETSI GS MEC 011 [i.4] over Mp1 reference points are recommended.
Gap/Key issue #6 - Maintaining sufficient XR Quality of Experience in the face of Bandwidth Fluctuation	6.6.2	Solution proposal #6-1, Future reallocation of MEC runtime resources using speculation	This solution proposes a method where an XR MEC application and the MEC system coordinate on the speculation of MEC resources to meet the performance needs of the application. This solution predicts the need for MEC resource reallocation based on application-level considerations and system insights. Gaps and recommendations for solution #6-1, is same as solution #5-1. MEC application contacting MEO for resource allocation adjustment is not supported in the current scope of the ETSI MEC architecture. ETSI GS MEC 002 [i.6] can be updated to include the requirements for run time resource allocation adjustment. Same proposed enhancements to ETSI GS MEC 10-2 [i.7] over Mm3 and ETSI GS MEC 011 [i.4] over Mp1 reference points are recommended.
Gap/Key issue #7 - Reducing processing and signalling among MEC Applications, which are components of an application	6.7.2	Solution proposal #7-1, Managing signalling from application components on CMH locally	The solution proposes an XR MEC Application, (e.g. Remote Component Aggregator, RCA), which collects and process notifications from remote MEC Application components deployed on CMH. RCA can expose information to other MEC applications, about • APIs available from the remote MEC application components, • Outcome of the processing of the notifications, received from remote MEC applications. Within the scope of the ETSI MEC architecture, collecting and processing of notifications from multiple MEC applications, providing processed information to another MEC application is not considered. ETSI GS MEC 002 [i.6] can be updated to include the requirements for aggregation and processing of MEC application information and make the processed information available to other MEC application. This can be an optional feature. Additional functions and interactions can be described as part of the feature enablement.

Key issue	Clause	Solution	Gaps within current ETSI MEC scope and recommendations
Gap/Key issue #8 - Transfer of a MEC application instance between Constrained MEC Hosts	6.8.2	Solution proposal #8-1, Device App initiated MEC application transfer on a Constrained MEC host	This solution proposes a method where a MEC application, e.g. XR Display Client Application and Device Apps, can initiate the transfer of another MEC application e.g. XR Rendering MEC Application from one MEC host to a second MEC host. A MEC application or a Device App, e.g. XR Display Client interacts with UALCMP to: • Create context for XR Render MEC app and provides a list of CMH where the XR Render MEC app instances can be deployed • Request the MEC system via UALCMP to transfer its User Application Context with the XR Render MEC app to the alternate CMH To support the interaction between Device App and UALCMP, Mx2 interface can be used and ETSI GS MEC 016 [i.14] can be updated to support the interaction between XR Display Client and UALCMP. The Mx2 interface terminates to an Application Lifecycle management proxy, which can be enhanced to support interaction with MEO to provide application context information to MEO, initiate deployment of MEC Applications, transfer user application context to another CMH. Evaluating the use of Mx2, ETSI GS MEC 016 [i.14] for application context transfer by Device App is further recommended.
	6.8.3	Solution proposal #8-2, Constrained MEC Host Location Information in MEC 016	Solution #8-2 proposes additional location constraints, which are required for constrained MEC hosts, to be considered in ETSI GS MEC 016 [i.14]. It is recommended to enhance ETSI GS MEC 016 [i.14] to include the proposed location granularity information.

Key issue	Clause	Solution	Gaps within current ETSI MEC scope and recommendations
	6.8.4	Solution proposal #8-3, XR Service Enabler initiated MEC application transfer on a Constrained MEC Host	Solution #8-3, proposes a MEC application like, XR Enabler Service, gets notified about low battery indication from a first XR MEC App and interacts with the MEC system (MEO/MEPM) • To identify a target MEC Host hosting a second XR MEC App instance • Transfer the XR MEC App user context information from the first XR MEC App to the second XR MEC App. User context information transfer is assumed to be done according to the application self-controlled user context transfer procedure in ETSI GS MEC 021 [i.11] MEC application contacting MEO to identify a target MEC Host hosting a second XR MEC App instance is not supported in the current scope of the ETSI MEC architecture. ETSI GS MEC 002 [i.6] can be updated to include the requirements for MEC application contacting MEO to identify a target MEC Host hosting a second XR MEC App instance. This can be an optional feature. Additional functions and interactions can be described as part of the feature enablement. ETSI GS MEC 003 [i.1] describes Mm3 and Mm5 interfaces, which can be used by MEO to reach the MEP through MEPM and vice versa. MEC application can send request to identify target MEC host to MEP, to be forwarded to MEO. MEO can provide information about target MEC hosts to MEP. MEP can update MEC application instances about target MEC hosts over Mp1 interface. Since Mm5 is not specified, the information exchange between MEO and MEP can be assumed internal function. ETSI GS MEC 10-2 [i.7] over Mm3 reference point can be enhanced to support identification of target MEC host by introducing new interaction between MEPM and MEO. ETSI GS MEC 011 [i.4] over Mp1 reference point can be enhanced to support the interaction between MEC application and MEP for identification of target MEC hosts.

History

Version	Date	Status
V4.1.1	August 2025	Publication