

This is a postprint version of the published document at:

Antevski, K., Bernardos, C.J., Cominardi, L., Oliva, A. y Mourad, A. (2020). On the integration of NFV and MEC technologies: architecture analysis and benefits for edge robotics. *Computer Networks*, Vol. 175, 107274.

DOI: <https://doi.org/10.1016/j.comnet.2020.107274>

© 2020 Elsevier Ltd. All rights reserved.



This article is licensed under a [Creative Commons Attribution Non-Commercial No Derivatives License 4.0 International License](https://creativecommons.org/licenses/by-nc-nd/4.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

On the integration of NFV and MEC technologies: architecture analysis and benefits for edge robotics

Kiril Antevski^{a,*}, Carlos J. Bernardos^a, Luca Cominardi^b, Antonio de la Oliva^a and Alain Mourad^c

^aUniversity Carlos III of Madrid

^bADLINK Technology

^cInterDigital Europe Ltd.

Keywords:

NFV

MEC

Integration

Edge Robotics

Use case

ABSTRACT

Forthcoming networks will need to accommodate a large variety of services over a common shared infrastructure. To achieve the necessary flexibility and cost savings, these networks will need to leverage two promising technologies: Network Function Virtualization (NFV) and Multi-access Edge Computing (MEC). While the benefits of NFV and MEC have been largely studied as independent domains, the benefits of an harmonized system comprising these two technologies remains largely unexplored. In this article we first identify a set of reference use cases that would benefit from a joint use of MEC and NFV. Then, we analyze the current state-of-the-art on MEC and NFV integration and we identify several issues that prevent a seamless integration. Next, we consider a reference use case, namely Edge Robotics, to exemplify and characterize these issues in terms of the overall service life cycle: from the initial development, to deployment and termination.

1. Introduction

In the last years, significant effort has been pushed into the research and definition of the technologies shaping the next generation of communication networks, 5G. Although from the user perspective 5G will be essentially characterized by an increase in the data rate and a significant decrease in the delay, the network architecture has gone through major updates. The new characteristics of 5G have been designed in order to meet three kind of services, which differ significantly on their characteristics: *i*) Ultra-Reliable Low Latency Communications (URLLC), *ii*) enhanced Massive Broadband (eMBB) and *iii*) massive Machine Type Communications (mMTC). These three services require different performance characteristics, ranging from absolute performance in terms of bandwidth (eMBB), delay and reliability (URLLC) and scalability (mMTC).

URLLC requirements are defined in an end-to-end fashion. Therefore, it is not sufficient to reduce the delay in the air interface, but it is also needed to reduce the propagation time between the service provider and the application/user consuming it. In other words, the application servers must be physically close to the user to achieve ultra-low end-to-end delay. With this idea in mind, the ETSI Multi-Access Edge Computing (MEC) [14] group was brought to live. ETSI MEC focuses on creating a computing, storage and networking platform at near proximity of the user capable of providing services with a very low delay. Deployment options for ETSI MEC include the instantiation of the service in servers located in the cell sites or in data centers. Typical examples of MEC applications include virtual or augmented reality, traffic offload, video processing and any other that may ben-

efit from placing computing near the user [9].

ETSI MEC was designed as a technology to be deployed in a virtualization environment, taking advantage of all the features in terms of flexibility, scalability options and ease management provided by the ETSI Network Functions Virtualization (ETSI NFV) framework [7]. However, its development started when ETSI NFV was not mature enough and ETSI MEC and NFV have been evolving in parallel for some time without a tight coordination among them. Recently, the ETSI MEC started an initiative to clarify all the open points that need to be addressed in order to run the ETSI MEC framework in an ETSI NFV environment without hassle, finding several open issues that require effort to be solved [13].

In this article, we aim at providing an up to date view of ETSI MEC and ETSI NFV standards and their relation, focusing on the key architecture-related issues and open points for a proper integration of both technologies. In order to ease the understanding of the architectural analysis, while keeping it complete and follow a practical approach, we put the focus on the viewpoint of an application developer facing the different issues found when integrating NFV and MEC. We have chosen this point of view because the resolution of most of the integration issues will highly impact how an application is instantiated in the integrated platform. Therefore, the outcome of the analysis aims at simplifying the operations required by developers to deploy and manage their applications on such an integrated system. While different use cases would benefit from the integration of NFV and MEC, as presented in the next section, such as video analytics, augmented reality or connected vehicles, we adopt the use case of Edge Robotics as a meaningful scenario. Using Edge Robotics, we describe and support the decisions taken on the different trade-offs encountered for the integration of NFV and MEC technologies.

The main contributions of this article are:

This work has been partially funded by the EU H2020 5G-TRANSFORMER Project (grant no. 761536), the H2020 collaborative Europe/Taiwan research project 5G-CORAL (grant no. 761586) and the EU H2020 5GROWTH Project (grant no. 856709).

*Corresponding author, email address: kiril.antevski@uc3m.es

Table 1

Advantages of applying MEC and NFV for each use case and application

Benefits \ Use-cases	Intelligent video acceleration	Video stream analysis	Augmented reality	Connected vehicles	IoT gateway	Cloud Robotics
Radio-link quality	X	X				X
Low-latency	X	X	X	X		X
Local Computation	X	X	X		X	X
Scaling		X			X	
Mobility				X		X
Real-time analysis	X	X	X	X	X	X

- presenting the motivation of a bulletproof integration of MEC and NFV technologies, making use of real-life use cases;
- identifying the gaps in normative specification needed to enable MEC deployments in NFV environments;
- proposing a comprehensive walk-through for operators on the various deployment options for an integrated NFV-MEC system by considering the point of view of the application developers who are the end users of such system; and,
- presenting a detailed solution to properly integrate MEC and NFV taking the specific use case of Edge Robotics, including workflows of all procedures and some experimental considerations.

The reminder of the paper is organized as follows. We elaborate on the main motivation to integrate MEC and NFV technologies, by first shortly describing in Section 2 the benefits brought by MEC and NFV in different use cases. Then we review how ETSI MEC has proposed to integrate MEC and NFV at conceptual level in Section 3, to then discuss the issues of such a proposal and the architectural choices left intentionally open. In Section 4 we focus on a specific integration of Edge Robotics in a MEC in NFV environment with directions to overcome the integration issues supported by an experimental assessment of an Edge Robotics prototype. Section 5 focuses on the comparison with related works and finally, Section 6 draws some conclusions and enumerate ongoing future work.

2. A use-case based view on the need of MEC and NFV

Multi-access edge computing appeared as a framework to make possible and/or improve several use cases that were not feasible with classical networking/cloud solutions [9]. We next review how using MEC technology and integrating it with NFV proves beneficial in those use case scenarios.

2.1. Intelligent video acceleration

Radio information can be dynamically used to adapt a video downstream application according to the estimation of available throughput in the radio downlink interface. MEC enables this *intelligent video acceleration* by locating a radio analytics application in the radio access network (RAN), which monitors the radio downlink interface and sends the

monitoring data to a video server. The video server uses the information to prevent TCP congestion and allow the application-level coding to adapt to the radio downlink capacity. This way of acceleration boosts up the users' quality of experience by adapting the video coding so it uses the full capacity of the radio links. Besides, multiple applications may use the radio monitoring data in parallel, justifying the need of enabling it as a MEC service. The deployment of MEC in NFV is clearly useful in this kind of video streaming scenario, by facilitating the deployment of on-demand video caches closer to the highly loaded regions. This enables saving network resources and avoiding some traffic congestion events that might occur when video content is not located closer to the video consumers.

2.2. Video stream analysis

Today's video surveillance systems (such as the ones deployed in public areas, parking areas, highways, private properties, etc.) make an increasing use of intelligent video recognition systems (e.g., of faces, license plates, etc). A classical approach is to locally identify patterns and send the video data to a cloud system where it can be stored and a more powerful analysis can be performed. Sending all this raw video data to a centralized cloud server is expensive and inefficient, even more considering new 4K or even 8K resolution cameras which are starting to be commercialized. By using a MEC approach, where the video stream analysis can be performed locally, on the vicinity of the cameras, the cost can be reduced by performing a local computation (close to the access network), where small pieces of information are extracted and processed from each video upstream. Further on, the MEC application can decide to upload the processed information to a cloud monitoring service or act immediately (e.g., in case of emergency). Here also the use of NFV technology proves to be helpful, as video processing applications can also be deployed and scaled on-demand (e.g., if an emergency situation arises where more computing power is needed to perform image recognition) thanks to NFV.

2.3. Augmented reality

Various types of events, such as visits to museums, galleries, music shows or sport events, could benefit from using event-tailored mobile applications capable of offering live additional content related to what the mobile device is pointing at (e.g., using the device's camera). The actual recording of the mobile phone's camera can be enriched with additional content, which is normally referred to as *augmented reality (AR)*. In order to operate properly, the augmented re-

ality application processes the camera input, the precise device location and sight direction (using the accelerometer sensor from the mobile device). With this input, the applications generates the additional information, which is displayed in real-time over the content that the user is seeing. This requires, in order to meet a satisfactory performance, data processing at a very high rate with low latency. Therefore, implementing AR as a MEC application is clearly a feasible and optimal approach, as MEC can provide the exchange of fast data rate, high computing power with low-latency in localized areas. Actually, the localized nature of the augmented reality application (e.g., at a museum, sport event) allows the use of additional MEC services such as localization. Equivalently to what happens with video streaming applications, the demand of AR in terms of computing resources very much depends of the event itself and the number of users, which makes NFV a very suitable technology to allow for dynamic scaling and better use of shared resources.

2.4. Connected vehicles

With the advent of autonomous vehicles, the need for network connectivity for vehicles is very fast increasing through the years. Forecasts predict the drastic increase of data flows from and towards the sensors or processors of the connected vehicles. Cars will communicate and exchange data among themselves and also with the road infrastructure, so drivers will be aware of the status of the roads, road accidents, etc. Additionally, more value-added services like infotainment, car finder, parking location, etc., would be also provided. MEC is a key tool here, as it can be used to distribute large portions of the services closer to the access network, as well as to enable fast processing of information coming from multiple connected vehicles or sensors and deliver the necessary information to the vehicles or road-assisting units (e.g., in case of an accident). MEC applications would reside on servers deployed in small-cell sites or Long-Term Evolution(LTE)/5G base stations close to the roads. As vehicles move along the roads, the use of "follow-me" features will allow MEC applications to maintain Quality of Service (QoS) and connectivity to the connected vehicles while migrating through the hosting systems in the direction of the movement. It is clear that this "follow-me" features require NFV technology to become feasible, as network virtualization mechanisms support migration of functions and applications within the infrastructure.

2.5. IoT gateway

The continuous evolution of the Internet of Things (IoT) demands upgrades to the gateway devices connecting the different "things". Gateways need to scale to support more devices while keeping a level of QoS and security. Different protocol families and radio technologies are currently used for IoT, which requires the role of IoT gateway: a low-latency aggregation point that can support various types of protocols, radio-technologies and real-time processing of monitoring data. MEC is a considered a suitable approach to meet the requirements of various IoT setups, by benefiting from low-latency local processing of monitoring data and

enabling the application of real-time analytics. Again, the integration of MEC with NFV in this scenario would allow a faster, cheaper and more agile adaptation by dynamically deploying IoT gateways to support the different IoT devices present on a given deployment.

2.6. Cloud robotics

Cloud robotics is a field of robotics that leverages and integrates cloud computing, cloud storage, and other Internet technologies, into industrial and commercial applications. Cloud technologies enable robot systems to be endowed with additional capabilities by leveraging the powerful computation, storage, and communication resources available in the cloud. Consequently, it is possible to build lightweight, low cost, and smarter robots by placing an intelligent brain in the cloud which offers a converged infrastructure that can be also used to share services and information from various robots or agents. Though robots can benefit from various advantages of cloud computing, this presents several limitations when applied to the cloud robotics field: cloud-based applications can suffer from high-latency or unpredictable jitter in the network. Because of this, the use of MEC clearly helps in reducing latency and using context information to help in controlling the robots. In this context, integration with NFV is also considered helpful, as different robot control mechanisms can be deployed on demand upon necessity. This also involves, for example, the setup of replicated functions to improve resiliency and reliability, which are critical in this kind of use case. Summarizing, an integrated MEC and NFV system offers a simplified management (provided by the operator) to cloud robotics which, in turn, can significantly reduce deployment costs thanks to the hardware pooling and the virtualization of white boxes.

Table 1 summarizes the main identified advantages of applying MEC and NFV for each of the analyzed use cases and applications.

3. NFV and MEC - Background and Integration

Considering the benefits of the relevant use cases and applications, in this section we present an overview of the two technologies, NFV and MEC, and we explore how the integration of MEC in NFV can be done, identifying some issues at architectural level.

3.1. Network Function Virtualization

The main architectural framework for NFV widely adopted by industry is the one defined by the the ETSI Industry Specification Group (ISG) NFV. It is composed of three domains:

- Virtual Network Function (VNF), running over the NFV Infrastructure (NFVI).
- NFV Infrastructure (NFVI), including the diversity of physical resources and how these can be virtualized. NFVI supports the execution of the VNFs.

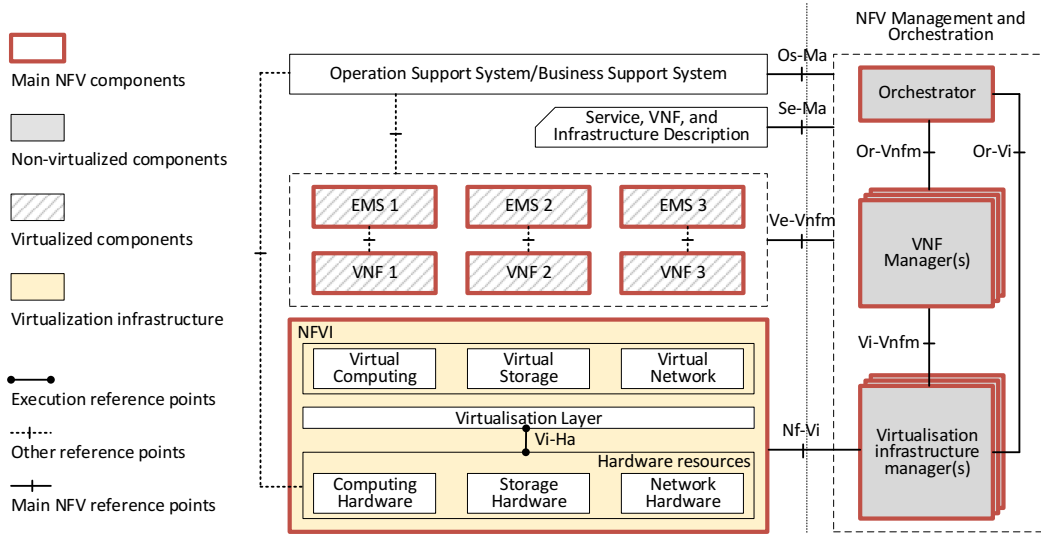


Figure 1: NFV architecture framework

- NFV Management and Orchestration (MANO), which covers the orchestration and life-cycle management of physical and/or software resources that support the infrastructure virtualization, and the life-cycle management of VNFs. NFV Management and Orchestration focuses on all virtualization specific management tasks necessary in the NFV framework.

This NFV architectural framework identifies functional blocks and the main reference points between such blocks as shown in Figure 1. Some of these are already present in current deployments, whilst others might be necessary additions in order to support the virtualization process and consequent operation. The functional blocks are explained in the following paragraphs.

The *NFV Infrastructure* includes the hardware and virtualized resources, and the Virtualization Layer. It encompasses the HW and SW resources that create the environment in which VNFs are deployed. The NFVI virtualizes physical computing, storage, and networking and places them into resource pools.

The *Virtualized Infrastructure Manager(s) (VIM)* is a functional block with the main responsibility for controlling and managing the NFVI compute, storage and network resources. It controls and manages the interaction of a VNF with computing, storage, and network resources under its authority, in addition to their virtualization.

The *NFV Orchestrator (NFVO)* is a functional block with two main responsibilities: the orchestration of NFVI resources across multiple VIMs, fulfilling the Resource Orchestration (RO) role, and the lifecycle management of Network Services (NS), fulfilling the Network Service Orchestration (NSO) role. The NFVO is responsible for installing and configuring new network services (NS) and virtual network function (VNF) packages, NS lifecycle management, global resource management, and validation and authorization of NFVI resource requests.

The *VNF Manager(s)* is a functional block with the main responsibility for the lifecycle management (e.g., instantiation, update, query, scaling, termination) of VNF instances.

The *Service, VNF and Infrastructure Description* is a data-set providing information regarding the VNF deployment template, VNF Forwarding Graph, service-related information, and NFV infrastructure information models. These templates/descriptors are used internally within NFV Management and Orchestration. The NFV Management and Orchestration functional blocks handle information contained in the templates/descriptors and may expose (subsets of) such information to applicable functional blocks, as needed.

The *Operations and Business Support Systems (OSS/BSS)* provide the operational and business support systems implemented by the VNF service provider.

As it can be evinced in Figure 1, some of these functional blocks are virtualized while others can be deployed in any form according to the operator's requirements (i.e., non-virtualized, virtualized, bare metal, etc.).

3.2. Multi-Access Edge Computing (MEC)

The main architectural framework for edge computing widely adopted by industry is the one defined by the ETSI ISG Multi-access Edge Computing (MEC) [14] and shown in Figure 2. The MEC framework is composed of two domains: *i)* the MEC Host level and *ii)* the MEC System level.

Similarly to the ETSI NFV case, the domains contain functional blocks that are either virtualized or that can be deployed in any form (i.e., non-virtualized, virtualized, bare metal, etc.). The main MEC component at host level is the *MEC Host*, which can be seen as an edge data center, and it consists of the *i)* Virtualization infrastructure, *ii)* the MEC Platform (MEP) and *iii)* the MEC Applications (MEC Apps).

The *virtualization infrastructure* provides compute, storage, and network resources for the MEC Applications. The virtualization infrastructure includes a data plane for routing the traffic among applications, services, local/external

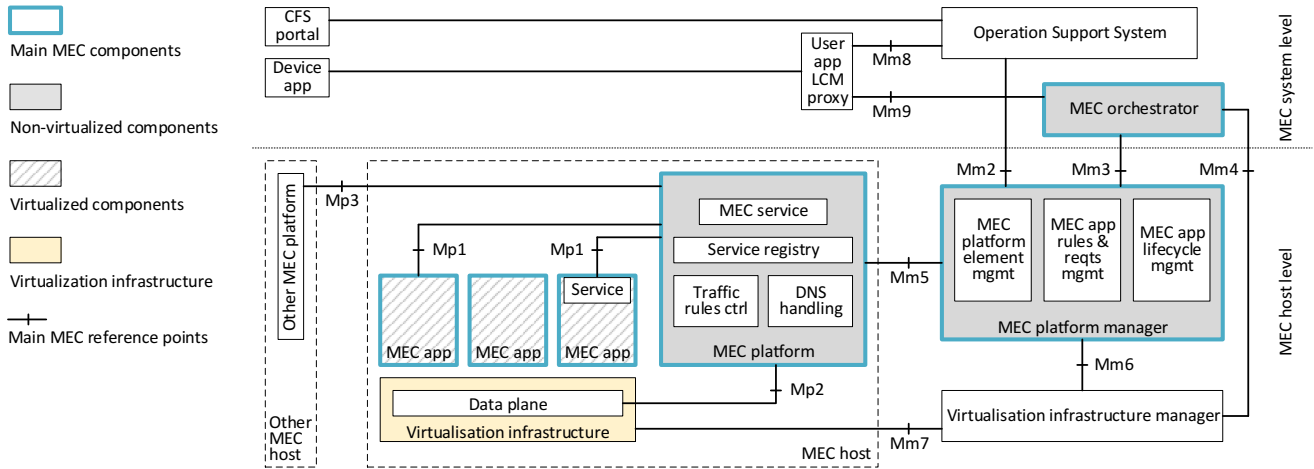


Figure 2: MEC architecture framework.

networks, and MEC Platform. The configuration of the data plane is done via the Mp2 reference point and it is managed by the traffic rules controller in the MEC Platform.

The *MEC Platform* offers an environment (i.e., service registry and DNS handling) where the MEC Applications can discover, advertise, consume and offer MEC Services.

MEC Applications run on top of the virtualization infrastructure provided by the MEC Host, and can interact with the MEC Platform to consume and publish MEC Services via the Mp1 reference point. How these MEC Applications retrieve the data to be published as a service is left unspecified by current ETSI MEC specifications. Finally, the MEC Host level encompasses two management components: *i*) the Virtualization Infrastructure Manager (VIM) and *ii*) the MEC Platform Manager. While the *Virtualization Infrastructure Manager* is in charge of controlling and managing the virtualized infrastructure, the *MEC Platform Manager* is in charge of controlling, managing, and configuring the MEC Platform for what concerns MEC Applications and MEC Services authentication and authorization.

While the MEC Host level operates on the single edge data center, the MEC System level operates across multiple MEC Host levels and its main components are *i*) the MEC Orchestrator, *ii*) the Operation Support System (OSS) and *iii*) the User application life-cycle management proxy.

The *MEC Orchestrator* is the main component for the MEC System level management and is in charge of maintaining an overall view of the MEC System based on deployed MEC Hosts, available resources and available MEC Services as well as on-boarding, instantiating, and terminating the applications. The *Operation Support System (OSS)* refers to the OSS of an operator. It receives requests via the Customer Facing Service (CFS) portal and from Device applications for instantiation or termination of applications, and decides on the granting of these requests. The Device applications can request life-cycle operations (on-boarding, instantiation, termination, modification/mobility) via the User application Life-cycle Management proxy. Granted requests

are forwarded to the MEC Orchestrator for further processing, which in turn may contact the MEC Platform Manager via the Mm3 reference point to manage the application lifecycle and enforce the application rules.

Ultimately, ETSI MEC defines a set of MEC Services such as the Radio Network Information Service, the Location Service and the Bandwidth Management Service. The Radio Network Information Service (RNIS) [12] provides radio network-related information, such as up-to-date radio network conditions, measurement and statistics information related to the user plane, and information related to users served by the radio nodes. While the original RNIS service was designed for 3GPP networks, a new service is being defined to cover also WiFi networks [15]. The Location Service [11] provides location-related information about the users (e.g., all of them or a subset) currently served by the radio nodes. The location information can be geo-location, Cell ID, etc. Finally, the Bandwidth Management Service [10] allows the allocation of bandwidth to certain traffic routed to and from MEC Applications and the prioritization of certain traffic. Additional services can be then defined upon necessity based on the same MEC framework.

3.3. Integration at architectural level

As it can be concluded from Section 3.1 and 3.2, ETSI NFV and ETSI MEC frameworks present several commonalities as also highlighted in Figure 1 and Figure 2, respectively. In an attempt to align and harmonize the two ISGs, ETSI MEC published a group report, namely MEC 017 [13], with goal of studying the deployment of MEC in an NFV environment. It is worth highlighting that this report represents just the first attempt from the industry to analyze the problem of integrating MEC and NFV technologies, and does not even aim at proposing a solution or set of solutions, but rather start identifying the main issues that would deserve additional work. The rationale of this exercise being done in ETSI MEC (and not in ETSI NFV) lies in the fact that the main focus of ETSI MEC is on the MEC platform/MEC

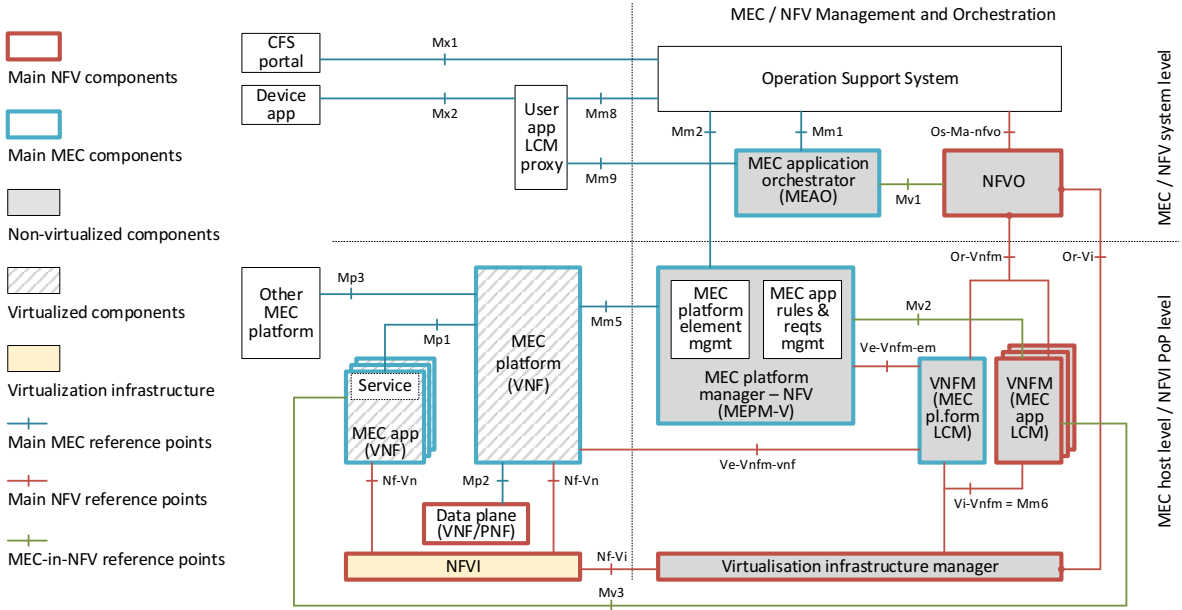


Figure 3: MEC reference architecture in a NFV environment as proposed in ETSI GR MEC 017 [13].

services and not on the virtualization infrastructure per se, which is instead the main focus of ETSI NFV. Therefore, ETSI MEC proposed a mapping between the MEC components and the NFV framework resulting in the MEC reference architecture illustrated in Figure 3, where the NFV components/reference points are highlighted in red while the MEC components/reference points are highlighted in blue. Moreover, Figure 3 highlights the components that are virtualized.

The assumptions for deploying MEC in NFV are:

1. The MEC platform is deployed as a VNF. For that purpose, the procedures defined by ETSI NFV are used;
2. The MEC applications behave as VNFs for the rest of ETSI NFV MANO components. This allows re-use of ETSI NFV MANO functionality;
3. The virtualization infrastructure is deployed as a NFVI and its virtualized resources are managed by the ETSI NFV defined VIM (part of the ETSI NFV MANO).

In the MEC architecture, the MEC host contains an instance of a virtualization infrastructure and runs an instance of the MEC platform. When integrating the MEC architecture into NFV, the concept of MEC host becomes obsolete and it is replaced by the concepts of NFVI-PoP (i.e., a data center) and zone (i.e., a set of co-located and well-connected physical resources within a NFVI-PoP). Moreover, the MEC platform manager as defined in [14] is transformed into a *Mobile Edge Platform Manager - NFV (MEPM-V)* that delegates the life-cycle management (LCM) part to one or more VNFM(s). Similarly, the MEC orchestrator is transformed into a *Mobile Edge Application Orchestrator (MEO)* that uses the NFVO for resource orchestration, and for orchestration of the set of MEC application VNFs as one or more NFV Network Services (NSs).

While most of the reference points defined in ETSI NFV and ETSI MEC do not require any changes, the ETSI MEC Mm3 reference point (see Section 3.2) needs to be extended and become *Mm3** to cater for the split between MEPM-V and VNFM for properly managing the MEC applications LCM. Likewise, the following new reference points are introduced between elements of the ETSI MEC and ETSI NFV architectures to support the management of MEC application VNFs:

- *Mv1*: It connects the MEO and the NFVO to allow the MEO to invoke operations towards the NFVO to manage MEC application VNFs;
- *Mv2*: It connects the VNF Manager that performs the LCM of the MEC application VNFs with the MEPM-V to allow LCM related notifications to be exchanged between these entities;
- *Mv3*: It connects the VNF Manager with the MEC application VNF instance, to allow the exchange of messages, e.g., related to MEC application LCM or initial deployment-specific configuration.

Finally, when MEC is deployed in a NFV environment, the data plane can be realized in two different ways. In the first option, the data plane is realized as a VNF which is integrated in the network service (NS) containing the MEC application VNFs. By doing so, Mp2 is kept as a MEC-internal reference point and it is agnostic to the way MEC is deployed. In the second option, the ETSI NFV MANO functionalities for configuring the data plane are used by the ETSI MEC components. That is, the MEC platform does not control the network configuration directly via Mp2 but rather requests it to the MEPM-V, which in turn requests the

MEAO. When receiving such a request, the MEAO contacts the NFVO to update the network configuration accordingly. Therefore, there is no dedicated VNF implementing the data plane as in the first option, making the Mp2 reference point unnecessary.

3.4. Integration issues

In the previous section we went through how MEC and NFV can be integrated at architectural level, starting from what the ETSI MEC ISG reported in [13]. In addition to the proposed integrated architecture, [13] identifies 14 key issues of different nature, proposing solutions for some, while leaving others for future study.

Next, we provide a summarized view of these issues, classifying them into three categories: *architectural*, *work-flow* and *communication* issues. This initial classification is intended to serve as baseline for a detailed analysis of the issues and gaps that exist today for the integration of MEC and NFV, which we perform in Section 4.3 together with our solution proposal.

3.4.1. Architectural (MEC as NFV) issues

The parallel development of ETSI MEC and NFV has yielded to two separated architectures with multiple duplicated functions played by MEC and NFV components. Such duplication is due to the need in ETSI MEC to orchestrate applications, which were developed without considering the parallel development of NFV technology. Moreover, although functionally the ETSI MEC and ETSI NFV architectures look very similar, they are based on slightly different assumptions which prevent a straightforward integration at architectural level. The main issue is related to the instantiation process, as the approaches followed by NFV and MEC differ. Considering the integration of the two architectures, as defined in Section 2, some responsibilities are moved from one component (i.e., MEAO) to another (i.e., NFVO). This creates the following issues:

- MEC/NFV descriptors partial compatibility;
- Mapping of MEC VNFs to network services;
- Connecting a MEC platform network service with a MEC application network service;
- Mapping of the concept of MEC host to NFV.

Starting with the descriptors compatibility, NFV uses a descriptor defining the information necessary to instantiate a VNF (VNF Descriptor, VNFD) while MEC uses one defining the information needed to instantiate an Application (Application Descriptor, AppD). Although the VNFD and the AppD present some similarities, the information in the AppD is not enough to instantiate a MEC Application in the form of a VNF. The information contained in an AppD focuses on the requirements of the infrastructure and the service availability for the MEC Application, while the information included in a VNFD is not tightly related to the infrastructure setup (e.g., no strict location constraints), but

more explanatory of the interconnection between internal and external components of a VNF. The mapping of the identifier data fields from an AppD to a VNFD is straightforward process, however, aligning network requirements and (part of) life-cycle management procedures from an AppD to a VNFD is not that simple due to some mismatches. For example, the AppD can only model a single virtual compute resource for a MEC Application, whereas the VNFD can define templates for multiple virtual compute resources per VNF to support scalability. Considering the above issues, we believe that the AppD and the VNFD should be integrated, having both descriptors the semantics to express each other behavior, enabling a loss-less translation between both descriptors.

However, the descriptor integration is not just a mere field matching exercise since the translation of an AppD to a VNFD implies a modification of the step-by-step on-boarding process of a MEC Application VNFD, which must consider the MEC specific orchestration entities (such as the MEAO). In Section 4.3 an exemplary mapping of AppD to VNFD is presented.

After an AppD is mapped/translated to a VNFD and then packaged into a VNF package, a ME app should be part of a network service in order to be instantiated in an NFV-MEC environment. This is the source of the second of the issues identified before (mapping of MEC VNFs to network services), which raises the concern of how to create a MEC Application VNF as part of a network service (NS). Considering the processing of an already on-boarded VNFD at instantiation time, in order to make the MEC application part of a network service that can be instantiated from scratch or that is already instantiated, the Mobile Edge Application Orchestrator (MEAO) and the NFVO have to be perfectly coordinated.

The main function of the MEAO is to orchestrate MEC application VNFs as part of a network service. In that sense, the MEAO takes care of assuring the presence of a MEC platform for each MEC application VNF that is instantiated. The MEC platform is assumed to be a VNF in the NFV-MEC environment [13].

Since every MEC application requires of a MEC platform, and both are VNFs, they can be orchestrated as part of an NS. The MEAO is responsible for assuring the correct instantiation of a MEC Platform VNF to any other MEC Application VNF. The connection between MEC Platform VNF and any MEC Application VNF can be established as a single composite Network service or as two individual Network services that are concatenated. The main difference between the composite solution and the concatenation approach is that in the composition case for each instantiation of a new MEC Application VNF, the general composite Network Service Descriptor (NSD) needs to be modified and the new MEC application VNF instance should be instantiated through a request for modification of already active network service instance. Whereas in the concatenation case, the MEC Application VNF is instantiated as an independent Network service and connected directly to the Service Ac-

cess Point (SAP). In the concatenation case, the life-cycle management is more complex with a benefit that each instantiated VNF is non-dependent of the status of the rest interconnected VNF instances.

In the initial MEC architecture, the MEC platform is instantiated on a specific host. In the NFV-MEC environment, the concept of a MEC host does not exist. This yields to the fourth of the issues identified at the beginning of this section. By eliminating the concept of MEC host, the issue is how to arrange nearby placement of the MEC platform and the MEC applications as part of a single network Service. Since both the MEC platform and MEC applications are deployed as VNFs, determining the placement of the MEC VNFs over a virtualized infrastructure is highly challenging.

3.4.2. Workflow issues

We now focus on the potential problems of the procedures such as: on-boarding, instantiation, modification (e.g., mobility) and termination of the MEC Applications in the NFV-MEC environment. As for the architectural issues, the problems arise with the presence of both the NFV and MEC orchestrators: NFVO and MEAO. Both orchestrators should assess an on-boarding VNF package of a MEC Application. In an NFV MANO environment, the NFVO is in charge of the NSDs and the VNF packages on-boarding procedures using the NFV Interfaces and Architecture (NFV-IFA) 013 specifications [16]. In an NFV-MEC environment, each on-boarded VNF package that contains MEC-related files needs to be processed by the MEAO as well. The main question is: which module should take care of the VNF package on-boarding? the NFVO or the MEAO?

A clean instantiation workflow demands instantiation of a NS that contains a MEC Application VNF at the start and applying the external MEC-specific features (such as modifying traffic rules, DNS rules, required/provided MEC services, etc.) in the final phase. Ideally the roles are equally split: initially the instantiation of NSs is orchestrated by the NFVO, and then it is handed over to the MEAO. If the NFVO handles the on-boarding of VNF packages, it sends a request to the MEAO to analyze the on-boarding VNF package and generate NSDs for instantiation of a MEC application VNF as part of a NS. It might require significant changes on the interfaces and in the internal workflow of the NFVO and MEAO. However, if the MEAO is the first to process on-boarded VNF packages and to generate NSDs, it orchestrates the whole instantiation procedure by requesting orchestration of NFV-related features (e.g., the NS instantiation) to the NFVO and applying the MEC-related features at the final phase. This would require applying additional features in the MEAO and less interventions in the NFVO. On the other hand, the termination has a reversed execution sequence from the instantiation procedure. First, the MEC-related features are removed and then the MEC Application VNF being part of a NS is terminated. To sum up, all procedures (on-boarding, instantiation and termination) present a different set of issues, that may occur depending on which module has the main orchestration role, the NFVO or the

MEAO.

The mobility feature has existed only in the MEC environment. The absence of a MEC host paradigm and non-existent VNF mobility in a NFV-only environment open a range of issues that need to be addressed. Enabling mobility features for VNFs is a completely open question.

3.4.3. Communication issues

Here we exemplify the potential issues on the newly introduced interfaces in the NFV-MEC architecture. With the convergence of both MEC and NFV, the interfaces that connect some of the modules may produce errors. We mainly focus on the set of interfaces:

- Mv1 - interface between the NFVO and MEAO;
- Mv2 - interface between the VNFM (Life-Cycle Management for MEC Apps) and MEPM-V;
- Mv3 - interface between the VNFM (LCM for MEC Apps) and MEC App VNFs;
- Mp2 - interface between the MEC Platform VNF and the Data plane.

The Mv1 interface is used by the MEAO to communicate with the NFVO for the deployment of a MEC application VNF as a NS along with the MEC Platform VNF. The Mv1 interface is used in all procedures with MEC application VNFs (on-boarding, instantiation, termination and modification). In the case of the NFVO being the main orchestrator, the Mv1 interface should support the demanded additional features where (i) the NFVO asks the MEAO to analyze an on-boarded VNF package for MEC-related features; (ii) the NFVO requests the MEAO to provide continuous feedback for the duration of the instantiation of a MEC application or MEC platform VNF as part of a NS.

If the MEAO has the main orchestrator role, the Mv1 interface would be an extension to the NFV-IFA 013 [16]. The MEAO generates NSDs that would fit the concatenation or composition of a NS that contains the MEC application VNF coupled with a MEC platform VNF. Upon generation of all NSDs, the MEAO requests on-boarding, instantiation/termination and modification of an NS on the Mv1 interface. The NFVO in that case is transparent to the presence of the MEC-related features and treats all requests the same way as for regular NSs.

The Mv2 interface is placed between MEPM-V and VNFM modules. Both modules are derived originally from the MEPM of the MEC architecture, where the MEPM-V is responsible for the MEC-specific part and the VNFM responsible for Life-Cycle Management (LCM) of the MEC Application VNFs. In order the MEPM-V to be aware of the LCM operations performed by the VNFM, the Mv2 interface allows to send direct requests, such as (i) requests for performance monitoring and fault information related to a specific MEC Application VNF instance; (ii) subscribe for notifications or query for life-cycle information of a MEC Application VNF.

The NFV IFA 008 [17] specification can be used for implementation of the Mv2 interface, but not all the specified operations are allowed. For example, invoking a instantiation of a MEC application VNF towards the VNFM can produce an error. Thus the implementation of the Mv2 interface must be synchronized to meet the requirements while taking care not to introduce errors.

While the Mv2 interface is used for extracting LCM information, the Mv3 interface is used to execute the LCM operations towards the deployed MEC Application VNFs and to allow MEC Applications to communicate directly to the VNFM. A "day-zero configuration" feature is enabled through the Mv3, which is a provision procedure that allows the VNFM to provide pre-boot configuration parameters of the MEC Application VNF instance via the VIM. It remains an open question the usage of Mv3 in an opposite direction or whether the MEC application VNF instance should invoke healing or scaling operations towards the VNFM.

The Mp2 interface, the interface between MEC Platform VNF and data plane, is the most complex for implementation. A possible implementation is via an indirect workflow: requests for data plane operations are sent from the MEC platform VNF towards the MEPM-V (via Mm5) which, acting as a proxy, will redirect them (via Mm3*) to the MEAO and the NFVO (via Mv1). Setting up this workflow demands invasive adjustments in each module and its internal workflows. Another option is to setup the data plane as an independent VNF. In that way, the implementation of the Mp2 interface would be dynamic and set up as a virtual link between two VNFs. A deeper analysis of these options is provided in the next section.

4. Nuts and bolts: NFV/MEC for edge robotics

While the previous section analyzed the integration of NFV and MEC, describing the main issues of such integration and outlining some of the possible solutions, this section takes a deep dive into the issues and solutions by considering a specific use case and dwelling on all the relevant considerations.

4.1. Major role: MEAO vs. NFVO

As pointed out in MEC-in-NFV [13], either the NFVO (Option 1, in the following) or the MEAO (Option 2, in the following) can play the role of the master module that oversees the procedures of on-boarding, instantiation, life-cycle management, termination and migration of MEC Applications. Therefore, two different approaches for the NFV/MEC integration can be followed, each approach with its pros and cons. To compare both approaches, we next elaborate on an analysis based on three key points or metrics: MEC awareness, Application Program Interface (API) call requests and states. As conclusion of this analysis, we propose the MEAO to be in charge of the orchestration procedures.

MEC Awareness: The first metric considered is the need to modify current implemented behaviours to account for the presence of MEC in the platform. In both approaches, the

NFVO is aware of the existence of the MEAO through the Mv1 interface that interconnects them. However, Options 1 and 2 differ in the degree of MEC awareness needed:

- Option 1: In the case that the NFVO is the master of all procedures for Network Services and MEC Applications, the NFVO has different degrees of MEC awareness. If the user (the MEC Application developer) uploads the MEC package, the NFVO has to be aware that it is a MEC package and redirect the package to the MEAO, so it can transform the MEC package into a VNF package. Then the MEAO would onboard the newly created VNF package. Even in the case that the MEC developer performs itself the transformation from MEC to VNF package, it has to obtain MEC related information via the NFVO. That means that the NFVO must be aware of MEC related requests and redirect them on the Mv1 interface towards the MEAO.
- Option 2: In the case that the MEAO is in charge of performing all procedures, the NFVO is oblivious to the existence of the MEC entities and constructs, such as MEC Packages, MEC Platform or MEC Applications. MEC packages are directly on-boarded to the MEAO where the transformation is performed according to the availability of MEC Platform VNFs. The generated VNF package is later on-boarded on the NFVO by the MEAO and included in an already existing NS that contains a chosen MEC Platform VNF by a simple NS update call towards the NFVO.

To sum up, in the case of NFVO having the master role (Option 1), the NFVO itself needs to be upgraded to satisfy the MEC related procedures, whereas in the case of MEAO being in charge (Option 2), the existence of the MEC related procedures is transparent to the NFVO. Note that the MEAO needs to be extended to understand the NFV procedures in any case. As conclusion, Option 2 is preferred since it reduces the amount of modifications to the current infrastructure.

Number of API calls: The second metric used in the comparison is the *number of API calls* that needs to be triggered on each MEC related procedure (summarized in Table 2). Considering the different possible master roles, we can observe that for Option 1 (NFVO as master):

- On-boarding procedure: there are a total of 6 API calls.
- Instantiation process: there are a total of 11 API calls.
- Termination process: there are a total of 8 API calls.

While for Option 2 (MEAO as master):

- On-boarding procedure: there are a total of 3 API calls.
- Instantiation process: there are a total of 10 API calls.
- Termination process: there are a total of 7 API calls.

Table 2

MEAO vs. NFVO: Number of API calls per procedure

Procedure	MEAO major role	NFVO major role
On-boarding	3 calls per MEC package	6 calls per MEC package
Instantiation	10 calls per MEC App VNF	11 calls per MEC App VNF
LCM	2 calls per MEC App VNF	2 calls per MEC App VNF
Termination	7 calls per MEC App VNF	8 calls per MEC App VNF

Regarding the rest of life-cycle management operations, there is no difference in terms of number of API calls.

Summarizing, the case of MEAO playing the major orchestration role involves significantly less API calls compared to the NFVO being the one playing the major role, therefore Option 2 is preferred.

Number of states: The last metric analyzed is the *number of states* that each module needs to be aware when performing all procedures. In general, the total number of states that the whole system needs to hold for each MEC Application remains the same in both cases, but the level of complexity needed in both approaches is not the same. For Option 1, when the NFVO plays the master role, all state is held by the NFVO, preventing the MEAO of controlling the state of the MEC Applications and Platform. Therefore, to implement Option 1, both the MEAO and the NFVO need to be refactored to transfer the MEAO capability of holding the states for each MEC Application or MEC Platform to the NFVO. For Option 2 (MEAO is in charge), the NFVO does not need to hold any MEC related states other than the usual tracking of the status related to NSs and VNFs.

From our point of view, although the total number of states remains the same, we argue that the NFVO performing the central orchestration role would mean that the workflow of each NFVO implementation should be significantly changed, as well as, the workflow of the MEAO due to transferring the functionality of recording states for each MEC Application/Service/Platform to the NFVO. As conclusion, Option 2 is also preferred for this metric.

As a consequence of the three mentioned aspects, our view is that the joint-implementation should have the MEAO as the major orchestration entity, being the first module to receive requests from the OSS and orchestrate all NFV-MEC procedures, so Option 2 is preferred. Although this approach places the orchestration burden mainly in the MEAO, it demands little updates in the NFVO module and have more natural evolution of joining the MEC in NFV as a future implementation.

Stepping forward with the approach of having the MEAO as major orchestration entity, a detailed explanation of the procedures for Edge Robotics MEC Application is presented in the next sections. First, the description of the Edge Robotics scenario is presented, followed by the detailed description of all the procedures to establish a Robot Control MEC Application on top of MEC in an NFV platform.

4.2. Edge robotics: Scenario setup

The Edge Robotics scenario envisages a fleet of mobile robots remotely controlled and coordinated to perform different tasks in a multi-access indoor or outdoor environment. An example of this environment would be a shopping mall, where robots are used to move goods and perform cleaning tasks. A similar scenario setup is used in [1]. Figure 4 illustrates the main components and services of the Edge Robotics platform. In this scenario, the robots act only as sensors and actuators, that is, all the robotics intelligence controlling the navigation, driving speed, driving direction, video surveillance, angle of sight, etc., is executed in a MEC server located at the Edge of the network. Therefore, the robots embed only minimal applications for (i) executing driving commands (e.g., drive forward, turn right, increase speed, etc.) received from the MEC application in charge of the control of the robot, and (ii) sending application specific data towards the MEC applications, such as encoding and streaming video from the on-board camera to a Video Surveillance application. An important component, and key feature of the MEC platform used on this application, is the radio information service used to gain knowledge of the status of the radio connectivity. This information may be used for closed control-loop between the robots forming the fleet and the MEC applications. The performance of the overall system depends on the quality of the radio channel which can be a limiting factor in terms of maximum number of robots, cameras, or video stream quality in certain areas.

To react upon radio connectivity variance, the Edge Robotics use case leverages on the Radio-Network Information (RNI) service provided by the MEC platform which offers context information about the robots connectivity. Regardless of the radio access technology (e.g., LTE or Wi-Fi), the RNI service monitors the connectivity and reports real-time information about the signal strength, MAC layer parameters, packet loss, etc., for each robot (UE or STA). This context information is then consumed by two exemplary robotics MEC applications for this specific use case: (i) Robot Control application and (ii) Video Surveillance application. The Robot Control application implements all the logic for coordination, navigation, and control of the movement of the robots in the physical environment. The precision of the control is determined by the quality of the radio signal, therefore the RNI can be used to optimize the control of the robots accounting for variance in the signal level. Clearly, this application takes benefit of the close proximity and on the contextual (RNI) information available at the MEC platform. The

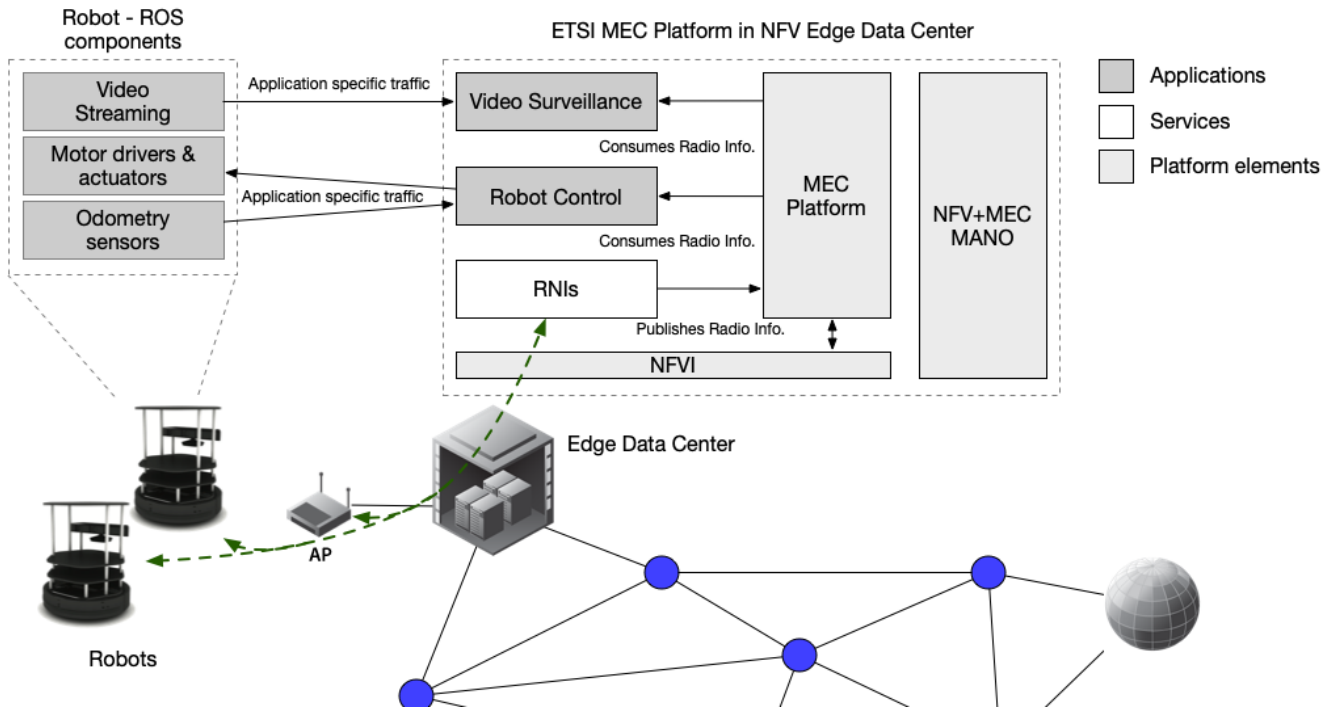


Figure 4: Edge Robotics scenario

Video Surveillance application controls the on-board cameras installed on the robots, collects the video streams, and cooperates with the Robot Control application on navigating the robots for better Video Surveillance of certain areas of interest. Using the real-time information from the RNI service, the Video Surveillance application can reduce or increase the up streaming quality (e.g., changing encoding, frame-rate, etc.,) improving the video quality in order to meet certain application specific constraints.

In addition, this use case makes explicitly visible the new business roles that the use of MEC technology enables. Let's take a shopping mall as a potential deployment scenario (to be further described next). In this case, the owner of the infrastructure (e.g., the micro-datacenter at the Edge) might be a Shopping Mall owner, or even a third party deploying infrastructure and managing it for the Shopping Mall owners. The Shopping Mall requires a service, which can be delivered through robots, such as cleaning, Video Surveillance or transport of goods. This service is provided by a third party robotics application provider, which delivers its applications in the infrastructure located at the Shopping Mall. To enable this future scenario, the underlying infrastructure of the access network and the network's edge should be virtualized. Enabling multiple providers to leverage a common infrastructure to deploy services. A virtualized underlying infrastructure is ready to adopt the MEC in NFV solution. The co-existence of multiple virtualized applications allows the users to easily extend the exemplary scenario by using different services or deploying different applications on the MEC platform.

4.3. Edge Robotics in NFV-MEC: issues and solutions

In this section we explore: 1) the initial setup of the NFV-MEC environment; the procedures of 2) on-boarding; 3) instantiation; 4) life-cycle management (LCM); and 5) termination of an exemplary MEC Application. From the Edge Robotics scenario, the observed procedures apply to both the Robot Control and Video Surveillance MEC Applications.

We first present the initial setup of the NFV-MEC environment on top of which the MEC application VNFs are deployed.

The main focus is on the step-by-step deployment procedures that are presented in the subsequent sections, explaining in detail the workflow of message exchange for each procedure. Note that the sequencing of the message exchange in the workflows is continuous and globally determined, meaning that the on-boarding procedure starts with number 1), but the termination procedure starts with sequence number 30).

4.3.1. Initial setup

Throughout the description of the workflow procedures we refer to the Edge Robotics scenario. In that sense, the following assumptions are made regarding the scenario setup shown in Figure 4 and deployed in a shopping mall:

- The robots have already been configured and ready to be used.
- Two embedded software features are up and running on the robots: (i) motor drivers & actuators, and (ii) odometry sensors, together with the communication

protocol stack, as explained in section 4.2. Both embedded features can be developed in various ways (e.g., as Robot Operating System (ROS) applications, firmware software, etc.) and they are out of scope for this work.

- The access network (e.g., Wifi or LTE) network has already been configured to cover the whole shopping mall area or at least the operating area of the robots. The access network is connected to the underlying virtualized infrastructure (e.g., small data center).
- It is assumed that the virtualized infrastructure contains the NFV-MEC environment and it is able to run MEC application VNF instances on top of it. The virtualized infrastructure can be owned by an arbitrary operator or the shopping mall itself.
- The MEC applications that offer Robot Control or Video Surveillance services can be provided by a robotic manufacturer (that provide the robots) or a third-party entity.

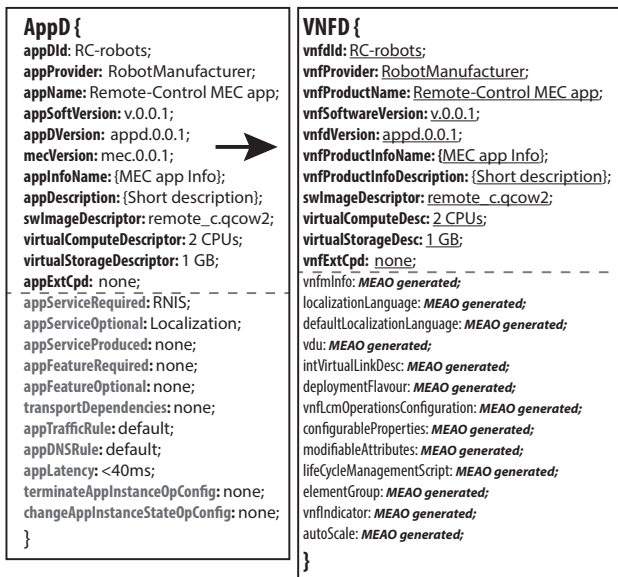


Figure 5: Mapping of AppD to VNFD.

In this scenario, the following stakeholders relationships are assumed: the operator is providing the virtualized infrastructure, the shopping mall owns the access network, and the robot manufacturer provides the robots and the MEC applications (Robot Control and Video Surveillance).

The overall workflow relies on the following assumptions:

- The MEC Platform VNF descriptor (VNFD) is already on-boarded.
- An Network Service Descriptor (NSD) that contains the MEC platform VNFD is generated by the MEAO.
- The generated NSD is used by the NFVO to instantiate the NS containing the MEC platform VNF.

- The NS instance is up and running.
- All MEC services are instantiated and initialized.

After describing the general scenario setup and its assumptions, in the next sections the step-by-step message exchange per procedure are described.

Please note that each numeric bullet corresponds to a message in the corresponding workflow.

4.3.2. On-boarding procedure

The on-boarding procedure is the first procedure that the application provider (in this case the robot manufacturer) has to perform successfully before being able to instantiate the MEC Application on the MEC in NFV environment. The workflow exchange of messages is presented in Figure 6 and a step-by-step detailed description is provided below:

1. First, the robot manufacturer defines the development of the MEC applications for the Robot Control and Video Surveillance. The work in [20] proposes guidelines for the development of MEC applications. In this case, the application will use one of the defined MEC services, the RNIS.
2. Afterwards, the MEC application is developed by including all function calls towards the required MEC services in the internal app-workflow. For example, the Robot Control MEC Application is requesting information from the RNIS MEC Service regarding the radio-link quality and requests the information of the current robot position in the shopping mall area from the Localization MEC Service.
3. Once the MEC application is completed, the developer assembles the application descriptor (AppD) and the MEC App Package. AppD parameters provide a description of the MEC features to be used by the application (e.g., required MEC Services, DNS rules, traffic re-direction rules, etc.). An example of those can be found in the leftmost part of Figure 5. In the exemplary case of the Robot Control MEC Application, the developer will include the RNIS as required ("appServiceRequired") and Localization as optional ("appServiceOptional") MEC services (leftmost part of Figure 5). As a last assembling step, the manufacturer packs the AppD along with all necessary files into a MEC App Package.
4. At this point, the application is ready to be deployed in a MEC infrastructure. Therefore, a robot manufacturer may initiate the on-boarding procedure at any time. This procedure starts by sending an on-boarding request to the OSS/BSS of the operator on the Mx1/Mm8 interface. The MEC App Package or a reference pointer (e.g. URL to the package) is attached to the request. The authenticity of the request is checked by the OSS/BSS and it is redirected towards the MEAO on the Mm1 interface. This step in the process defines that the MEAO has the main role of orchestrating and managing the requests.

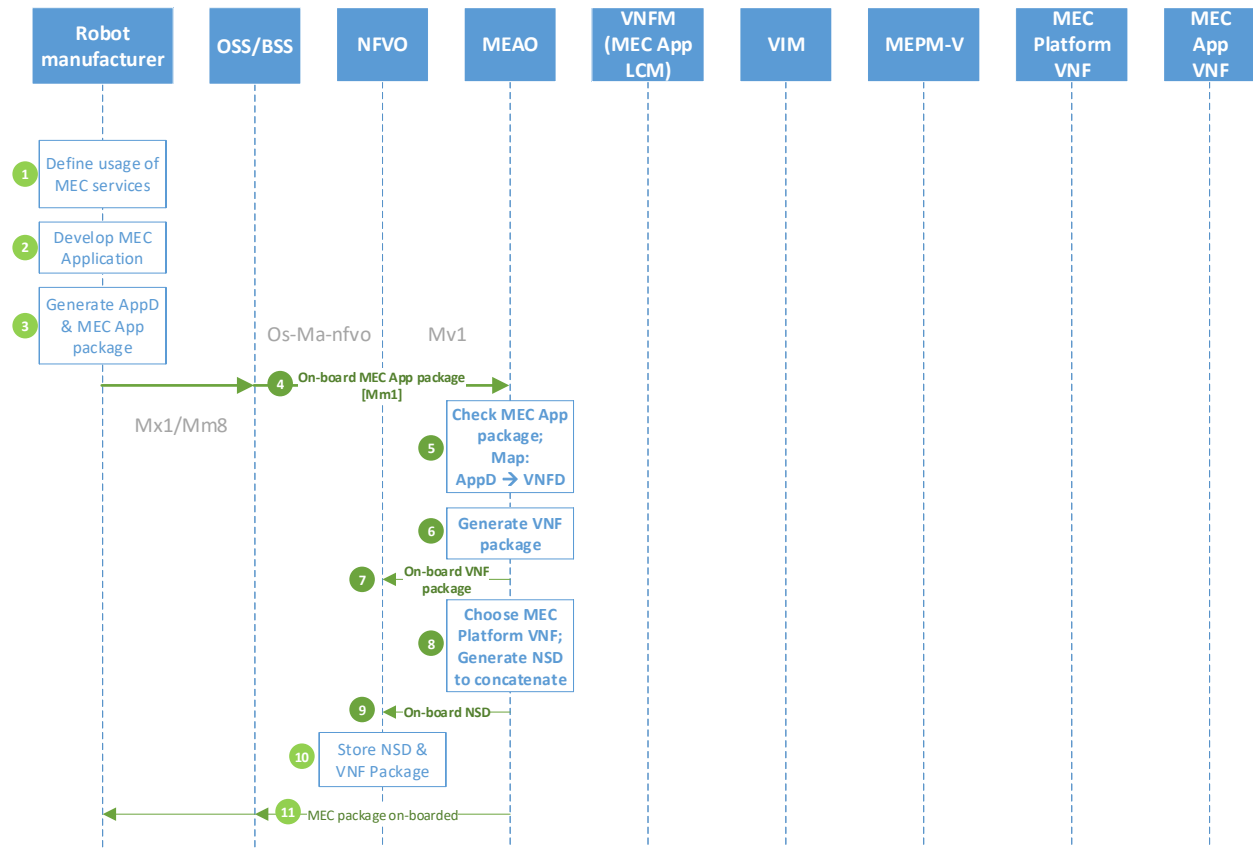


Figure 6: On-boarding workflow.

5. Upon reception of the MEC App Package, the MEO starts the translation of the AppD to a new VNFD. To do so, the MEO performs a one-to-one mapping of the different parameters of the AppD to the ones defined in the VNFD. This is a crucial step to enable seamless co-existence of the MEC functionalities in the NFV environment. As a consequence of the issues described in Section 3.4, the one-to-one mapping is not entirely feasible. Therefore, the MEO performs a partial mapping, including in the VNFD only the matching parameters, while the other parameters are placed in a separate (external) file. In addition, for some specific fields of the VNFD, the MEO must generate values to generate a complete VNFD. An example of mapping the Robot Control MEC AppD to the Robot Control MEC VNFD is presented in Figure 5.
6. After the VNFD is finalized, the MEO generates a VNF package, which contains the VNFD along with all external files, indexed by a manifest file. The external files consists of all files included in the MEC App package plus the file with unmapped AppD parameters, generated in the mapping process. The generation of the VNF package is an important process which might require a lot of MEO processing.
7. The MEO sends an on-boarding request, including the VNF package, to the NFVO on the Mv1 interface. The NFVO is not aware of any MEC related information and processes the request as a normal VNF on-boarding request.
8. Simultaneously, the MEO performs a check on the requirements imposed by the AppD parameters (e.g., required MEC Services, traffic rules, DNS rules, etc.). Since the MEO knows all up-and-running MEC Platform NS instances, it can make the selection of the MEC Platform to be used by the newly on-boarded MEC Application, based on the requirements described in the AppD. Once the MEC Platform to be used has been selected, the MEO generates a new MEC App Network Service Descriptor (NSD), that contains the new MEC App VNFD (generated in step 5) and the definition of the different Virtual Links (VLs) to connect the MEC App NS with the chosen MEC Platform NS instance (at instantiation time).
9. The generated MEC App NSD is on-boarded to the NFVO on the Mv1 interface and similarly processed by the NFVO, as in step 5.
10. The NFVO stores the NSD and VNF package in an internal database.
11. The MEO concludes the MEC Application on-boarding

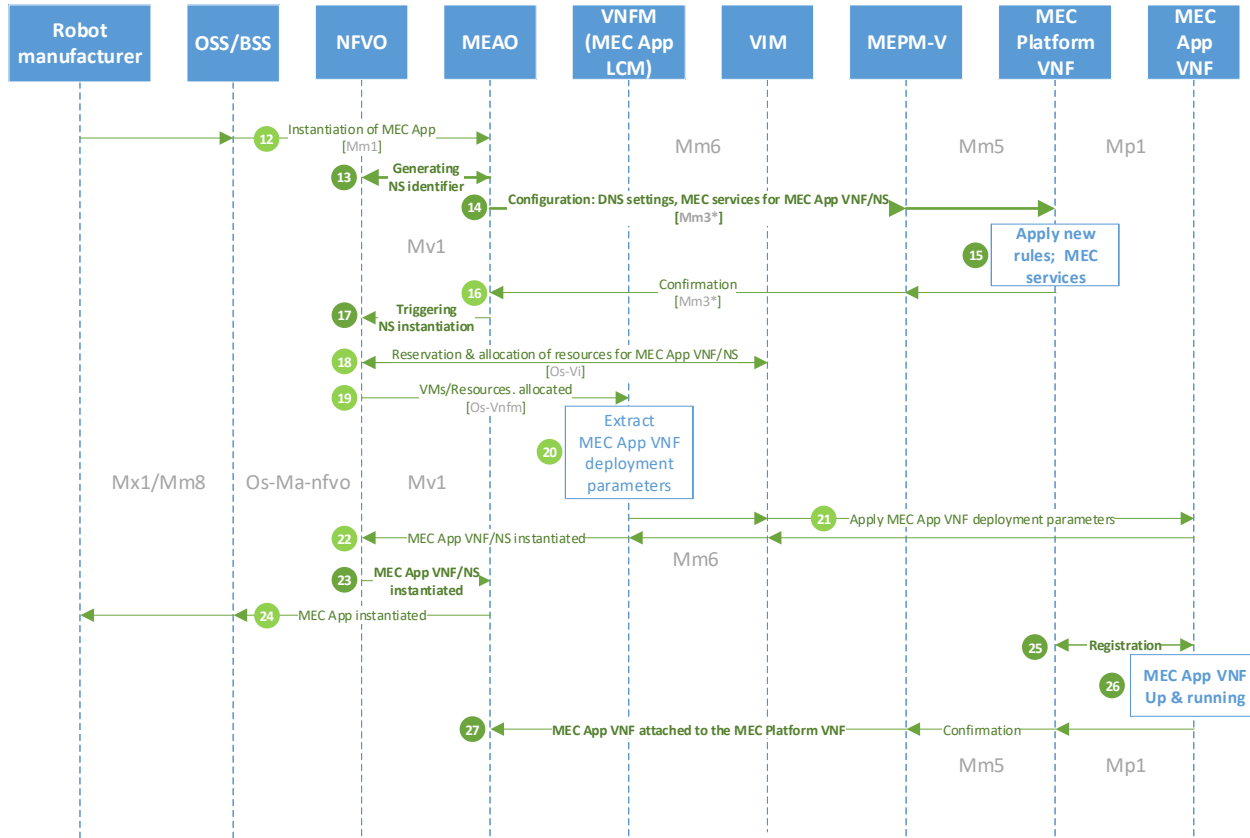


Figure 7: Instantiation workflow.

procedure by sending a confirmation of successful MEC package on-boarding via the operator’s OSS/BSS (on the Mm1 interface) back to the Robot manufacturer (on the Mx1/Mm8 interface).

Note that in this example, it is assumed that the MEAO chooses the MEC platform VNF to associate the MEC application before the actual instantiation of the application. This may cause some problems if the resources available for the MEC platform change between the time the MEC Application is on-boarded and the time of its instantiation. An alternative workflow (not presented in this work) performs steps 8-11 on instantiation time.

4.3.3. Instantiation procedure

At this point of the workflow, the application has already been on-boarded and is ready to be instantiated, or it is not yet running. In Figure 7, the following steps are taken in order to instantiate and boot up the application:

12. The Robot manufacturer requests the instantiation of the MEC application through the OSS/BSS (interface Mx1/Mm8) via the Mm1 interface to the MEAO.
13. The MEAO requests the NFVO to generate an NS identifier for the previously on-boarded NSD (as defined in [16], operation *CreateNsIdentifierRequest(NSD)*).

14. Based on the information of the descriptor, the MEAO instructs the Mobile Edge Platform Manager - VNF (MEPM-V) to configure the MEC Platform to include all the requirements for the application, such as traffic redirections, DNS rules or to configure the required MEC services. The MEPM-V redirects the request to the MEC Platform VNF on the Mm5 interface.
15. The MEC Platform VNF analyzes the information from the request, applies all the traffic rules and configures the access for the required MEC Services. For example, the Robot Control MEC application needs that the platform supports the RNIS MEC Service. It is important step to prepare the MEC Platform to accept association of the new MEC Application and interconnected all necessary MEC Services.
16. The MEC Platform VNF sends confirmation of the applied configurations and rules to the MEPM-V on the Mm5 interface, which are redirected to the MEAO through the Mm3* interface between the MEPM-V and MEAO.
17. The MEAO generates an instantiation request to the NFVO (on the Mv1 interface), using the previously generated *NSid* (as defined in [16], operation *InstantiateNsRequest(NSid)*). The NFVO accepts the request

and checks if it is feasible considering current resources availability.

18. In case the application instantiation is feasible, the NFVO requests the VIM to reserve the computational resources needed by the application. Note that according to the specified location constraint in the NSD, the resource reservation is executed at the same NFVI-PoP where the MEC platform VNF instance resides. Once the computational resources are reserved on the specific NFVI-PoP, the NFVO requests for the allocation of networking resources in order to enable network connectivity between the MEC application VNF/NS instance and the MEC platform VNF/NS instance. The operations are part of a common ETSI NFVO MANO instantiation workflow procedure [8]. After the computational resources are reserved and the networking connectivity is up and running, the NFVO requests the VIM to allocate the components of the MEC Application VNF. Since the MEC Application NS is a single VNF, there is no need to differentiate between the VNF and NS for this example. The VIM deploys the software images (e.g., Virtual Machines, VMs) and connects them to the networking fabric. Finally, the VIM notifies the NFVO for the instantiated VNF components.
19. The NFVO notifies the VNFM (MEC App LCM) of the successful allocation of the resources (VMs). In addition, it forwards the VNFD to the NFVO complemented with some extra information regarding the MEC Platform used, such the IP address to be used by the application to connect to the platform.
20. The VNFM (MEC App LCM) extracts all necessary MEC App VNF deployment parameters from the VNFD, and creates a set of configuration parameters, including MEC Platform specific information, to configure the MEC App VNF.
21. The VNFM (MEC App LCM) performs the provision of the MEC App VNF configuration via the VIM. Once applied, the MEC App VNF/NS is started.
22. The successful completion of the MEC Application VNF instantiation is acknowledged by the VNFM (MEC App LCM) to the NFVO.
23. The NFVO confirms the instantiation of the MEC Application VNF/NS to the MEAO on the Mv1 interface. This step concludes the instantiation of the Robot Control MEC Application as a VNF orchestrated by the NFVO.
24. The MEAO sends confirmation of instantiated MEC Application VNF/NS to the Robot manufacturer via the OSS/BSS. From this point on, the Robot Control MEC Application mainly depends of the implementation of the MEC functionalities in the MEC in NFV environment.
25. Meanwhile, at bootstrap, the Robot Control MEC Application registers to the MEC platform VNF. The Robot Control MEC Application will use the IP address of the MEC Platform VNF provisioned by the VNFM.

The registration request is received by the MEC Platform VNF and the registration is confirmed.

26. The Robot Control MEC Application is up & running, consuming all the requested MEC Services. The Robot Control MEC Application is ready to accept connection for robot devices. When a robot is powered on, it attaches to the Wi-Fi access network deployed in the Shopping Mall. Then a connection is established between the robot and the Robot Control MEC application, as result of the traffic redirection executed by the MEC platform. From this point on, robots continuously send sensor data towards the Robot Control MEC Application, where the data is analyzed. Along with the sensor data, the Robot Control application obtains near real-time information regarding the quality of the radio connection of each robot through the RNIS MEC Service, and obtains near real-time localization information from the Localization MEC Service.
27. The MEC Application VNF/NS confirms the correct set-up to the MEC Platform VNF/NS via the Mp1 interface. The MEC Platform VNF/NS confirms the correct set-up to the MEPM-V via the Mm5, which in turn redirects the confirmation to the MEAO via the Mm3*.

4.3.4. Life-cycle management (LCM) procedure

Once the Robot Control MEC Application is up & running, it performs its activity while being monitored for performance issues by the Orchestration system (specifically by the MEPM-V). A couple of messages are used for maintaining the life-cycle management of the application shown in Figure 8.

28. The MEPM-V polls for monitoring information to the VNFM (MEC App LCM) via the Mv2 interface.
29. The monitoring information for the Robot Control MEC Application is periodically updated by the VNFM (MEC App LCM) through periodically polling the Robot Control MEC Application via the Nf-Vn interface.

4.3.5. Termination procedure

At some point of time, the Robot Control MEC Application is no longer be needed. The termination process involves MEC and NFV specific signaling as shown on Figure 9 and defined in the following steps:

30. To terminate the Robot Control MEC Application VNF/NS, the Robot manufacturer issues a termination request via the OSS/BSS (using the Mx1/Mm8 interface) to the MEAO (using the Mm1 interface).
31. The MEAO forwards the termination request to the NFVO.
32. The NFVO using the common ETSI NFV MANO termination workflow procedures (described in [8]) that instructs the VIM to de-allocate all the resources used by the Robot Control MEC Application VNF/NS. The VIM de-allocates the resources assigned to the Robot

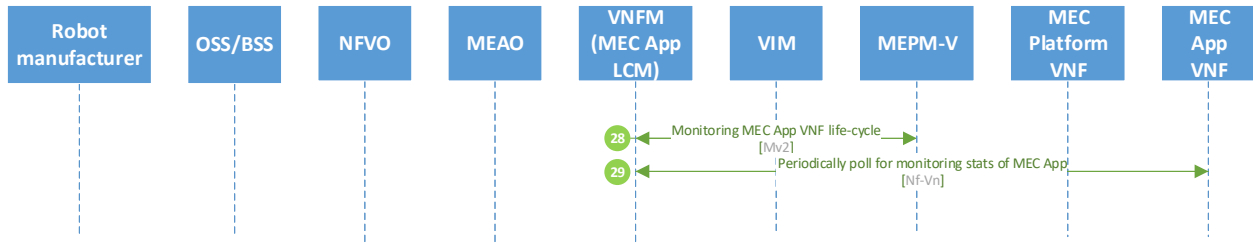


Figure 8: Life-cycle workflow.

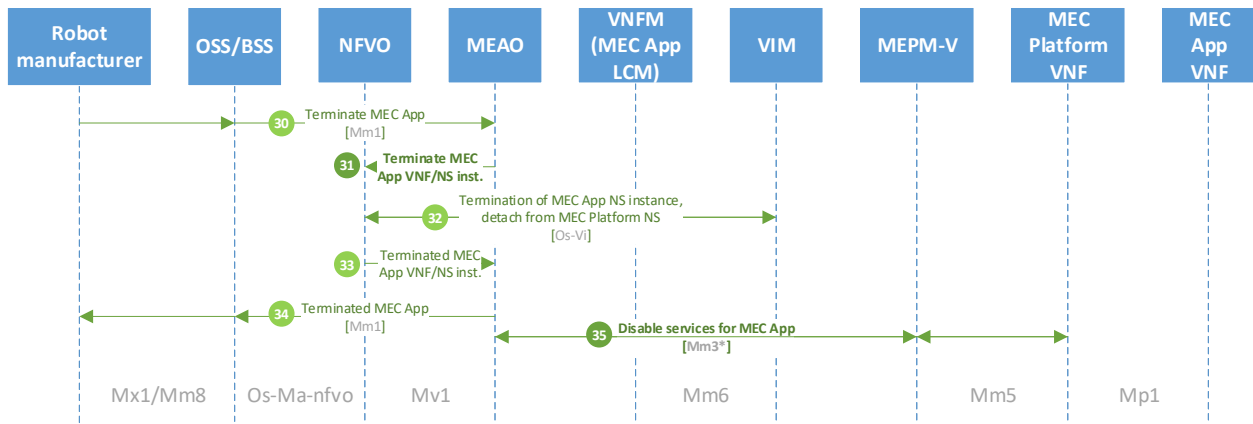


Figure 9: Termination workflow.

Control MEC application VNF/NS and confirms the operation to the NFVO.

33. The NFVO confirms the termination of the Robot Control MEC Application VNF/NS instance.
34. The MEAO acknowledges the termination procedure to the Robot manufacturer via the OSS/BSS.
35. The MEAO instructs the MEC platform VNF to disable the provision of MEC services and all traffic/DNS rules configurations. This is done by sending a request on the Mm3* interface to the MEPM-V which is redirected to the MEC platform VNF/NS instance. The MEC Platform un-registers the Robot Control MEC Application and acknowledges the completion of the procedure to the MEAO (Mm3*) via the MEPM-V (Mm5).

4.4. Edge robotics in NFV-MEC: experimental considerations

This article departed from the idea that integrating ETSI MEC in an NFV platform brings benefits that can not be achieved by any other mean. From the application developer point of view (following the same rationale as the rest of this work), the benefits of using an ETSI MEC platform can be summarised as: *i*) experiencing a lower end to end delay due to the presence of computing power that can be used to host the application in the vicinity of the user, and

ii) the possibility of using contextual information to improve the user experience, information that cannot be obtained or that loses its validity if the processing is done far away from the user. In the same way, the use of a virtualized platform to host the MEC platform benefits from the NFV concept by enabling: *i*) the possibility to scale up or down the resources available for the application based on its current needs, *ii*) the possibility of moving or migrating the application to the best location to serve the users (maybe at some point the application is needed in a per user basis and in another it is better to locate it slightly further away for aggregation purposes), and *iii*) the use of a common white box platform that can host a heterogeneity of applications under a common framework.

In order to provide an experimental proof of these benefits, the authors performed a validation of the Edge Robotics use case in a prior work [1]. This work showcases and measures, under real conditions, the benefits of deploying a Robotics application in a MEC-like platform on top of virtualized resources. This prior work tackles the above benefits, showing how the use of contextual information can be used to improve driving performance of a robot. The experimental setup is similar to the described use-case in this work. The scenario is deployed on virtualized infrastructure, and makes use of a MEC Application for Robot movement control and a RNIS MEC Service for extracting context data regarding the radio connectivity of the robots. The goal is to make robots

go as fast as they possibly can based on their radio conditions. Obtained results shows that the use of the context information through the RNIS MEC Service improved significantly the driving performance of the robot. The control-loop algorithm of the MEC Application adjusts the speed of the robot so it adapts to the the radio quality of the link. Hence in case the robot entered in an area of low radio coverage, the speed would smoothly drop from the maximum of 0.75 m/s to the minimum of 0.1 m/s. In the event of no context information provided by the RNIS MEC service, the robot starts to perform drive-stop movements that changed the trajectory and driving smoothness, which for a real environment (such as factories, labs, etc.) questions the usability of the robot itself. In addition, the smooth control of the robot was achieved thanks to the reduced latency provided by the MEC platform. The control software has been developed using standard Linux tools and deployed in a set of virtual machines and containers running in heterogeneous hardware, therefore proving that the use of a virtualization approach was useful.

This previous work did not considered a full NFV platform, but relied on a simpler virtualization platform, which could be extended to become a full NFV platform. However, while conducting these experiments, we discovered the different caveats and holes in the current MEC/NFV integration specifications, yielding to the work presented in this article. By applying the proposed MEC-in-NFV integration following the workflows proposed in this work, the experiment could easily evolve into a complete platform where new MEC Application and/or MEC Services can be instantiated. For example, the experimental system can be upgraded with the Video Surveillance MEC Application and Localization MEC Service. The combination of the Localization MEC Service and the RNIS MEC Service can produce a radio heat map of the experimental area and assist the MEAO for performing the migration of MEC Application VNFs.

5. Comparison with previous work

In this section, we identify other existing efforts, and compare our work with them.

In [13], which presents the initial idea of joint MEC in NFV architecture, most of the key issues are listed and provided with some *possible* solutions. We have extended this work by categorizing the issues (Table 3) and going deeper into the solution space by taking a specific use-case and explaining the step-by-step workflow procedures. Following this approach, we have been able to identify an additional issue that was not foreseen by the time [13] was published. For some of the cases where multiple approaches were proposed, we have favored one solution over the others, and gone further in terms of detailing it, based on our implementation and use-case experience.

In the work presented in [4], the authors similarly propose a deployment of a specific use-case (immersive video) on an integrated MEC-in-NFV platform. The integration of the MEC in NFV environment is broadly described with

small effort in evaluation of the issues described in [13]. The experimental results evaluate the performance enhancement through the use of MEC Application running close to the end users over NFV Infrastructure. The improvement is clearly noticeable through the upload improvement from order of tens of minutes to order of seconds. By using GPU acceleration, the application significantly improves the performance of transcoding several parallel media streaming sessions. In our view, modifying the workflow procedures can even more increase the efficiency of the system. Although the used MEC-in-NFV platform is a modified prototype version of the described architecture in [13], some relevant points related to both the on-boarding and the instantiation procedures are missing. In the on-boarding procedure, it is assumed that the MEC app developer would develop the application and generate a set of descriptors accordingly. Both AppD and VNFD are packed in a MAP (MEC Application Package). This is on-boarded together with generated NSD onto the MEC-in-NFV platform. The MEAO is in charge of the analyzing of the on-boarded files (MAP + NSD), which basically separates the AppD from the VNF package, stores the bindings of the AppD, VNFD and NSD; and then on-boards the VNF package plus the NSD to the NFVO. In the instantiation procedure, the dependency between the MEC Platform VNF and the MEC App VNF is out of scope. The assumption is that the MEC App VNF would be included as part of the NS (defined by the on-boarded NSD). Once the NS is instantiated, the connection to a already deployed MEC Platform NS instance would be handled by the NFVO. Although the main focus of the work is not to tackle most of the problems, it manages to describe the coupling between a MEC Platform VNF and the data plane into a single VNF (SGWLBO). This approach is extension of the optional deployment of MEC in 4G [19], which is suitable for applying the traffic and the DNS rules for the proposed (video streaming) use case.

The work done in [18] showcases the adaptation of MEC in NFV architecture as a key driver for expansion of Vehicle-to-Everything (V2X) applications and its development. The work lists several emerging challenges that need to be overcome for successful adaptation for the V2X Applications. With the tutorial perspective of our work, we contribute towards tackling one of the major emerging challenges in [18]: the on-boarding and running MEC applications provided by developers not aware of NFV procedures.

A different realization of the MEC in NFV environment is proposed in [21]. The authors, in similar manner as our work, propose a parallel extension of the ETSI NFV MANO platform towards a new enhanced MANO platform that can support MEC Applications without additional hardware or coordination overhead. Additional feasibility analysis is provided for several use-cases (i.e., migration of gaming applications, user mobility). The proposed architecture is not inline with [13], despite proposing the MEC Platform to be realized as VNF, other elements such as the Mobile Edge Orchestrator is proposed to devolve into the NFVO and maintain only application orchestration capabilities into a new or-

Table 3
Issues description

Issues	Description	MEC 017	MEC 017 solution	Solution provided in workflow
Conceptual	Mapping of MEC app VNFs to NSs	#1	Multiple	X
	Usage of NFV NS	#2	Multiple	X
	AppD vs. VNFD for MEC App VNFs	#6	Partial	X
	VNF Package vs. MEC application package	#7	Multiple	X
	Comparison of AppD and VNFD data structures	#10	Multiple	X
	NFV construct that corresponds to MEC Host	#11	None/Partial	
Workflow	VNF package on-boarding	#8	Multiple	
	MEC package on-boarding	-	-	X
	MEC App VNF Instance Relocation	#12	None	
	MEC App instance instantiation	#13	Multiple	X
	MEC App instance termination	#14	Multiple	X
Communication	Communication between MEAO and NFVO via Mv1	#3	Multiple	X
	Communication between VNFM and MEPM-V via Mv2	#4	Partial	X
	Communication between VNFM and ME app instance via Mv3	#5	Partial	X
	Managing traffic redirection	#9	None/Partial	X

chestrator - Application Functions Virtualization Orchestrator (AFVO). Hence the workflow procedures would be incompatible with this work or with the proposed integration in [13].

Similarly, the authors in [5] propose a novel approach of designing flexible and independent framework for MEC applications that collaborates with NFV frameworks. The work proposes a novel architecture design with introduction of similar concepts of MEC Applications realized as NSs or VNFs. The work envisions the novel Automated Provisioning Framework for MEC (APMEC) framework on top of existing NFV environment as an extension that is able to communicate directly to multiple VIMs and coordinate in parallel with existing MANO platforms. The procedures aim to provide MEC Applications as part of NS by breaking a received NS onto MEC and NFV parts and deploying each part in parallel, using existing MANO platforms or the APMEC framework. However, the descriptors used are not defined and used only by the APMEC framework. A MEC Platform that provides MEC Services towards MEC Applications is non-existing concept in the APMEC framework. In [6] is showcased the parallel orchestration using the implementation of the APMEC framework.

Another framework for parallel orchestration of NFV and MEC is envisioned in [3]. The focus of the work is mainly orchestration of multi-domains than integration of MEC in NFV. The proposed framework does not envision any changes to the NFV and MEC architectures, but the focus is more how to implement an integrated solution for simultaneous orchestration.

In [22], the authors propose specific use-case driven orchestrator module that would bridge the gap between orchestration in NFV and MEC environment. The work proposes some ideas how to overcome the orchestration gaps between the NFV and MEC environment without focusing on the main challenges, suggesting that the new module is adaptable to a MEC in NFV environment.

The implementation of a MEC Radio Network Information Service (RNIS) aimed for a MEC-in-NFV environment is explored in [2]. That work presents how a standard-compliant RNIS can be realized on top of virtualized infrastructure and compares different message brokers that enable the RNIS MEC Service to provide information towards MEC Appli-

cations. The work does not dive into the implementation details of a MEC-in-NFV environment. Similarly, in [23] the focus is on proposing a VNF placement scheme for the purpose of optimizing the latency and reliability of MEC Applications deployed as VNFs in MEC-in-NFV environment, but not in examination of the MEC-in-NFV environment itself.

6. Conclusion and Future Work

Virtualization, and more specifically, its application to bringing resources to the edge of the network, is one of the key technologies in 5G and future beyond-5G networks. The combined use of MEC and NFV enables a virtualization layer providing the features needed for fine grade customization of the networks. This customization is key for the flexibility demanded from vertical industries.

Despite of the clear inter-relation of MEC and NFV technologies, they have been evolving with isolated, and quite parallel, tracks until very recently. This separate evolution has created integration issues that need to be addressed, as early identified in [13]. We have performed an architectural analysis identifying all possible MEC-NFV integration issues, including feasible approaches to address each of them. In order to better scope this work, we have adopted the specific use case of Edge Robotics to dive into the detailed workflows and mechanisms to combine MEC and NFV. Besides, while doing so we have tried to put the focus point into application developers, as they are the final users of combined MEC-NFV deployments.

A key aspect of the conducted analysis of the combination of MEC and NFV technologies is which logical entity has to be in control of all the orchestration procedures (on-boarding, instantiation, general life-cycle management and termination). Based on an detailed analysis of the following aspects: MEC awareness, number and type of API call requests and states, we have concluded that the MEAO is better suited to be in control of the orchestration. Detailed workflows for the case of Edge Robotics are provided to support this design decision.

Compared with previous related work looking at integration aspects of MEC and NFV, our work is not just limited to a high level analysis of the issues appearing when combining the technologies, but it goes into actual design and

validation using a specific use case. Future work includes conducting a proof-of-concept with a prototype of the edge robotics use case. Additionally, we will also explore the issues connected to the the data plane and applying various traffic rules via the MEC Platform VNF (e.g., traffic redirection, DNS rules, etc.).

References

[1] Antevski, K., Groshev, M., Cominardi, L., Bernardos, C., Mourad, A., Gazda, R., 2018. Enhancing edge robotics through the use of context information, in: Proceedings of the Workshop on Experimentation and Measurements in 5G, ACM. pp. 7–12.

[2] Arora, S., Frangoudis, P.A., Ksentini, A., 2019. Exposing radio network information in a mec-in-nfv environment: the rnisaa concept .

[3] Carella, G.A., Pauls, M., Magedanz, T., Cilloni, M., Bellavista, P., Foschini, L., 2017. Prototyping nfv-based multi-access edge computing in 5g ready networks with open baton, in: 2017 IEEE Conference on Network Softwarization (NetSoft), pp. 1–4. doi:10.1109/NETSOFT.2017.8004237.

[4] Cattaneo, G., Giust, F., Meani, C., Munaretto, D., Paglierani, P., 2018. Deploying cpu-intensive applications on mec in nfv systems: The immersive video use case. *Computers* 7, 55.

[5] Doan, T.V., Nguyen, G.T., Kropp, A., Fitzek, F.H.P., 2018. APMEC: an automated provisioning framework for multi-access edge computing. CoRR abs/1805.09251. URL: <http://arxiv.org/abs/1805.09251>, arXiv:1805.09251.

[6] Doan-Van, T., Kropp, A., Nguyen, G.T., Salah, H., Fitzek, F.H.P., 2019. Programmable first: Automated orchestration between mec and nfv platforms, in: 2019 16th IEEE Annual Consumer Communications Networking Conference (CCNC), pp. 1–2. doi:10.1109/CCNC.2019.8651871.

[7] ETSI, 2013. Network Functions Virtualisation (NFV); Architectural Framework. Group Specification (GS) NFV 002 v1.1.1. European Telecommunications Standards Institute (ETSI).

[8] ETSI, 2014. Network Functions Virtualisation (NFV); Management and Orchestration. Group Specification (GS) ETSI GS NFV-MAN 001 V1.1.1. European Telecommunications Standards Institute (ETSI).

[9] ETSI, 2015. Mobile-Edge Computing (MEC); Service Scenarios . Group Specification (GS) MEC-IEG 004 V1.1.1. European Telecommunications Standards Institute (ETSI).

[10] ETSI, 2017a. Mobile Edge Computing (MEC); Bandwidth Management API. Group Specification (GS) 015 v1.1.1. European Telecommunications Standards Institute (ETSI).

[11] ETSI, 2017b. Mobile Edge Computing (MEC); Location API. Group Specification (GS) 013 v1.1.1. European Telecommunications Standards Institute (ETSI).

[12] ETSI, 2017c. Mobile Edge Computing (MEC); Radio Network Information API. Group Specification (GS) 012 v1.1.1. European Telecommunications Standards Institute (ETSI).

[13] ETSI, 2018a. Mobile Edge Computing (MEC); Deployment of Mobile Edge Computing in an NFV environment. Group Report (GR) 017 V1.1.1. European Telecommunications Standards Institute (ETSI).

[14] ETSI, 2018b. Mobile Edge Computing (MEC); Framework and Reference Architecture. Group Specification (GS) 003 v2.1.1. European Telecommunications Standards Institute (ETSI).

[15] ETSI, 2018c. Multi-access Edge Computing (MEC); WLAN Information API. Group Specification (GS) 028 v2.0.1. European Telecommunications Standards Institute (ETSI).

[16] ETSI, 2018d. Network Functions Virtualisation (NFV); Management and Orchestration; Os-Ma-Nfvo reference point - Interface and Information Model Specification. Group Specification (GS) ETSI GS NFV-IFA 013 V2.4.1. European Telecommunications Standards Institute (ETSI).

[17] ETSI, 2018e. Network Functions Virtualisation (NFV); Management and Orchestration; Ve-Vnfm reference point - Interface and Information Model Specification. Group Specification (GS) ETSI GS NFV-IFA 008 V3.1.1. European Telecommunications Standards Institute (ETSI).

[18] Giust, F., Sciancalepore, V., Sabella, D., Filippou, M.C., Mangiante, S., Featherstone, W., Munaretto, D., 2018a. Multi-access edge computing: The driver behind the wheel of 5g-connected cars. CoRR abs/1803.07009. URL: <http://arxiv.org/abs/1803.07009>, arXiv:1803.07009.

[19] Giust, F., Verin, G., Antevski, K., Chou, J., Fang, Y., Featherstone, W., Fontes, F., Frydman, D., Li, A., Manzalini, A., et al., 2018b. MEC deployments in 4G and evolution towards 5G. ETSI, White Paper, under publication in February .

[20] Reznik, A., Arora, R., Cannon, M., Cominardi, L., Featherstone, W., Frazao, R., Giust, F., Kekki, S., Li, A., Sabella, D., et al., 2017. Developing software for multi-access edge computing. ETSI White Paper 20.

[21] Sciancalepore, V., Giust, F., Samdanis, K., Yousaf, Z., 2016. A double-tier mec-nfv architecture: Design and optimisation, in: 2016 IEEE Conference on Standards for Communications and Networking (CSCN), pp. 1–6. doi:10.1109/CSCN.2016.7785157.

[22] Taleb, T., Frangoudis, P.A., Benkacem, I., Ksentini, A., 2019. Cdn slicing over a multi-domain edge cloud. *IEEE Transactions on Mobile Computing* , 1–1doi:10.1109/TMC.2019.2921712.

[23] Yala, L., Frangoudis, P.A., Ksentini, A., 2018. Latency and availability driven vnf placement in a mec-nfv environment, in: 2018 IEEE Global Communications Conference (GLOBECOM), pp. 1–7. doi:10.1109/GLOCOM.2018.8647858.



Kiril Antevski received the B.S. degree in telecommunication engineering from the The Saints Cyril and Methodius University of Skopje, Macedonia in 2008 and the M.S. degree in telecommunication engineering from the the Politecnico di Milano, Milan, Italy in 2016. He is currently pursuing the Ph.D. degree in telematics engineering at University Carlos III Madrid (UC3M), Spain.

His research interest includes the development of mechanisms to integrate and enhance NFV and MEC for 5G Networks in dynamic and heterogeneous environments.

Mr. Antevski's has won best-paper award on the 9th IFIP Wireless and Mobile Networking Conference (WMNC) in 2016.



Dr. Carlos J. Bernardos received the degree in telecommunication engineering and the Ph.D. degree in telematics from the University Carlos III of Madrid (UC3M), in 2003 and 2006, respectively.

From 2003 to 2008, he was a Research and Teaching Assistant with UC3M, where he has been an Associate Professor, since 2008. He has published over 100 scientific papers in prestigious international journals and conferences. He is an active contributor to IETF since 2005, e.g. to AUTOCONF, MEXT, NETEXT, DMM, MULTIMOB, SDNRG and NFVRG working/research groups, being co-author of more than 30 contributions, 10 RFCs, has co-chaired the IETF P2PSIP WG, and currently co-chairs the IPWAVE WG and the Internet Area Directorate (INTDIR).

He visited the Computer Laboratory of University of Cambridge in 2004 and the University of Coimbra in 2005. He has worked in several EU funded projects, being the technical manager of the FP7 MEDIEVAL and H2020 5G-Exchange

projects, and the Project Coordinator of the 5G-TRANSFORMER and 5Growth projects. His current work focuses on virtualization applied to 5G networks.



Dr. Luca Cominardi received his Bachelor's and Master's degrees in Computer Science at the University of Brescia, Italy, in 2010 and 2013, respectively. He received his Master's degree and Ph.D. in Telematics Engineering from the University Carlos III of Madrid (UC3M), Spain, in 2014 and 2019, respectively.

From 2013 he covered various research positions focusing on network softwarization and related topics like Software Defined Networking (SDN), Network Function Virtualization (NFV), and Multi-access Edge Computing (MEC). Starting from 2019 he is a Senior Technologist at ADLINK Technology working on fog computing and distributed systems.

He is an active contributor to the ETSI MEC working group where he has also served as rapporteur. He has published several papers in international journals and issued 5 patents.



Dr. Antonio De La Oliva received his telecommunications engineering degree in 2004 and his Ph.D.

in 2008 from the Universidad Carlos III Madrid (UC3M), Spain, where he has been an associate professor since then. He is an active contributor to IEEE 802 where he has served as Vice-Chair of IEEE 802.21b and Technical Editor of IEEE 802.21d. He has also served as a Guest Editor of IEEE Communications Magazine. He has published more than 30 papers on different networking areas.



Dr. Alain Mourad is a Director Engineering R&D at InterDigital. He is currently leading the research and development of Next Generation Radio Access Networks (NG-RAN) at InterDigital Labs in the UK and Germany. Prior to joining InterDigital, Dr. Mourad was a Principal Researcher at Samsung Electronics R&D (UK) and previously a Senior Researcher at Mitsubishi Electric R&D Centre Europe (France). He is a prolific inventor with over 50 issued patents and a thought leader with numerous talks and publications. He received the Inventor of the Year Award from Samsung Electronics R&D in 2012 and 2013, InterDigital Innovation Awards in 2016 and 2018, and the 2018 Global Telecoms Awards "Highly Commended" in the category "Advancing the road to 5G".