Multi-access edge computing smart relocation approach from an NFV perspective

Cristina Bernad[1][0000−0001−9537−415X], Vojdan Kjorveziroski[2][0000−0003−0419−4300], Pedro Juan Roig[1][0000−0002−8391−8946], Salvador Alcaraz[1][0000−0003−3701−5583], Katja Gilly[1][0000−0002−8985−0639], and Sonja Filiposka[2][0000−0003−0034−2855]

[1] Department of Computer Engineering, Miguel Hernández University (Elche), Alicante, Spain cbernad,proig,salcaraz,katya@umh.es

[2] Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University (Skopje), Skopje, North Macedonia
vojdan.kjorveziroski,sonja.filiposka@inki.ukim.mk

Abstract. This paper analyses the virtualised entities migration process implementation within the ETSI-compliant edge framework, considering the necessary multi-access edge computing (MEC) modules information interchange required for instantiation, termination and migration of MEC applications. Based on the variant of the MEC-NFV architecture and the functions of each element of it, a communication process that includes network function virtualisation (NFV) interfaces is provided, as a step towards the unresolved challenge of modelling and developing a migration procedure that is aligned with the MEC standardisation process.

Keywords: Edge Computing · Optimisation · Standardisation · Process design · Mobile nodes.

1 Introduction

Edge computing is devised as a promising technology that will boost the development of latency sensitive and intensive computing mobile applications on localised premises instead of transmitting them to remote servers in the cloud. Extending virtualised compute and network resources from cloud infrastructures to edge microdatacentres closer to end users' mobile devices has the potential to enable highly dynamic service provisioning that will guarantee the performance of near real time heterogeneous services such as autonomous vehicles, industrial automation or extended reality. Effective integration of edge computing with a broad range of user equipment is expected to include Internet of Things (IoT) devices, unmanned aerial vehicle (UAVs), autonomous cars, etc. However, a lot more effort is needed in order to accomplish such expectations as edge computing requires the convergence of IT and telecommunications networking industry, and there are still many issues on the table for reaching an industry consensus.

Meanwhile, the academia started working towards new initiatives that open the door to providing intelligence to the edge infrastructure, directing its research efforts toward solutions that lead to an Artificial Intelligence (AI)-assisted edge [1]. Machine learning algorithms implemented at the edge of the network is the next step to complete the technological set that will guarantee proactive behaviour and the best possible performance in resource management optimisation at the edge layer.

Standardisation efforts are also being intensified during these last years from the European Telecommunications Standards Institute (ETSI) Multi-access Edge Computing (MEC) initiative Industry Specification Group (ISG), providing a MEC reference framework documentation [2]. With the advantages provided when coupling 5G with MEC, it becomes a necessity to align the 5G architecture with the MEC reference framework architecture so that their joint implementation can be done in a compatible, effective manner. 5G is promoting services such as telemedicine, smart cars, IoT or HD video that will be boosted when all 5G specifications such as high reliability (99.999%), high data rates (greater than 10Gbps) and ultra-low latency (less than 1ms) are resolved for heterogeneous devices and services. MEC FW provides a scenario where the computation and network resources are closer to the end user and thus it achieves a very low latency with the already available technology. Thus, the initial ETSI MEC reference architecture has recently been extended and aligned to the existing Network Function Virtualisation (NFV) interfaces that provide the basis of the 5G core implementation in order to achieve compliance with the reference documents provided by the ETSI ISG NFV group. Although the major steps towards implementing MEC with NFV have been defined, there are still a number of open issues that need to be addressed with the implementation of smart relocation using MEC application migration [3]. Effective implementation of smart relocation is essential for advanced MEC systems that are implemented in a highly mobile environment such as 5G, where the benefits of MEC such as low latency can be retained only if the MEC services continuously remain in the closest possible vicinity of the users. In this paper, we focus on the MEC-NFV reference framework to conceptually design the smart relocation migration problem following the latest ETSI guidelines. Smart relocation is defined as the necessity to migrate the MEC applications to adapt to the changing network performances for mobile end-users. Namely, as the user enters a new service area, the virtualised entities that provide MEC services to the user's mobile user equipment need to be moved to the closest MEC host in order to continue guaranteeing ultra-low latency.

The remainder of this paper is organised as follows. In Section 2 we review some of the latest contributions focusing on virtualised entities migration in MEC-NFV environments. Section 3 introduces the MEC-NFV reference architecture and focuses on the description of the communication processes required for instantiating, terminating and migrating virtualised entities. The paper ends with a discussion and conclusions section4.

2 Related Work

Efficient resource management has been a very popular research topic ever since the introduction of edge computing [4]. However, while lots of attention has been given to the efficient initial placement of an application on a particular edge host, the problem of efficient migration in mobile environments has received less scrutiny [5]. Lately, the work on edge computing use cases such as autonomic vehicles, where mobility is an intrinsic part of the scenario, has given a tremendous rise to the approaches that aim to solve the problem of smart relocation using optimised migration approaches [6]. A few examples include work focusing on optimising resource allocation and migration in multi-cell scenarios [7], delay and mobility-aware approaches based on probabilistic methods [8], as well as on mobile agents [9], and Markov decision processes [10]. When it comes to the implementation of these algorithms and strategies, the virtualisation infrastructure of MEC applications is mostly implemented using virtual machines. Lately, there are several studies that explore containers as means to implement MEC such as [11].

When considering NFV, there are some other initiatives that tackle the migration problem by taking advantage of the concept of Service Function Chains (SFC) that is used to propose optimisation algorithms based on Dijkstra for enabling seamless migration through SFC reconfigurations [12]. Another example is focusing on optimising resource allocation for VNF migrations using genetic algorithms [13]. Recently, there are a number of studies that use AI to resolve smart relocation problems such as [14] that uses deep reinforcement learning strategies and [15] that introduces cognitive edge computing.

Recognising that the 5G and MEC integration can be achieved more easily if NFV is used for both, implementations of this blended environment are also being studied [16]. However, it is important to remark that the previously mentioned contributions do not consider the NFV-based MEC generic architecture in their proposals and, therefore, their research is not specifically aligned to the "smart relocation function" defined in [17] that provides details defining the process of transferring an instance of a MEC application to maintain the quality of service for the users. And while there is literature that addresses the MEC-NFV implementation, very few papers discuss the implementation of smart relocation in this setting. In these examples, such as [18], the authors do not consider the existence of two separate orchestrators and the rest of the specific MEC components that are identified in the ETSI MEC-NFV reference architecture. Thus, these approaches can not be used as a standardised approach to implementing smart relocation and more work is needed to define this process.

3 MEC-NFV Management flows

Having in mind that the underlying architecture that is used to build the core 5G components is based on NFV elements, the MEC integration into the 5G system can be done in a smoother fashion if the MEC components can be implemented

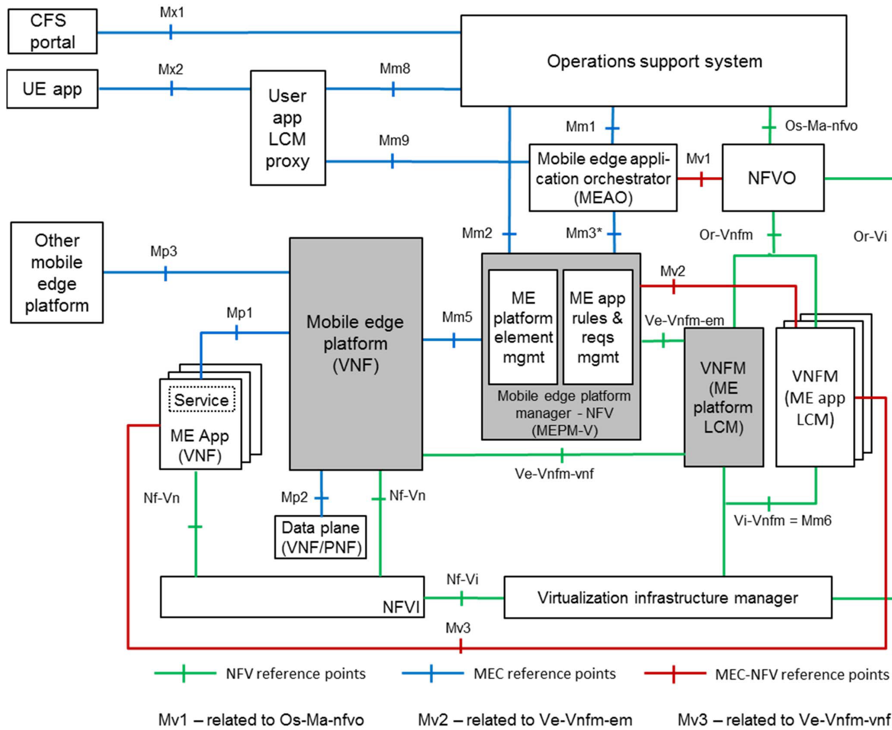

Fig. 1. NFV-based MEC generic architecture.

using NFV entities [16]. For this reason, ETSI has been working for years on defining a framework and an architecture based on MEC-NFV [2].

The main idea of the MEC-NFV implementation is to implement MEC applications as Virtual Network Functions (VNFs), see Figure 1. In this setting, there is a hierarchical setup of two separate orchestrators, the Mobile Edge Application Orchestrator (MEAO) that decides when and where to instantiate, migrate and terminate a MEC application, and the Network Function Virtualisation Orchestrator (NFVO) that takes over the responsibilities to create and manage the MEC applications as VNFs. The management of each individual MEC host, i.e. server hosting MEC applications, is done using the Mobile Edge Platform implemented as a separate VNF. Each set of co-located MEC hosts are managed by a virtualised implementation of the Mobile Edge Platform Manager (MEPM-V) that is in charge of specifying the MEC applications rules and requirements. The lifecycle management of each MEC application is delegated to the VNFM.

In an urban, dense 5G settings consider one or more base stations collocated with microdatacenters made out of several MEC hosts making up a mobile edge platform. Each service area has a MEPM-V that manages the grouped MEC hosts providing virtual MEC resources for that area and the orchestrators are used for a higher-level network-wide management.

Each element of the NFV-based MEC architecture uses a reference point to communicate with another entity as represented in Figure 1. Initially, the ETSI MEC document [2] defined the MEC reference points (in blue colour) for the generic reference architecture, which can be divided into reference points referring to the functionality of the MEC platform (`Mp`), management reference points (`Mm`) and the reference points that connect to external entities (`Mx`). Later, the reference points for the NFV-based MEC architecture variant (green color) were also included and defined in the document and some more reference points (red colour) were added to join the new entities of the MEC-NFV variant with the entities of the general architecture (`Mv2` and `Mv3`).

When analysing the migration process of VNFs, we find that there are specific details to be defined since they have not been subject to the standardisation carried out by ETSI MEC ISG yet. The main issue is that NFV based components do not include the explicit interfaces and processes for transparent VNF migration since this has not been a requirement for the NFV architecture. Thus, a series of open issues remain on how to implement the MEC application smart relocation for mobile end user scenarios in an efficient and uniform manner.

Aiming to tackle this problem, we have decided to start by defining the workflows that need to be implemented in order to instantiate, migrate and terminate a MEC application using the definition of the standardised MEC-NFV entities and the reference points between them. These workflows are intended to make it easier to understand which entities are involved in the migration process and what are their roles, as well as how all interfaces come together and where we should focus our attention to be able to successfully implement smart relocation.

3.1 MEC Application Instantiation

When we talk about a virtualised MEC-NFV environment, MEC applications are implemented as VNFs reached via user accessible MEC services. We detail out the steps taken during the instantiation process flow (see Figure 2) that is activated when a new MEC application needs to be created.

1. A user, through an application that is running on his device (device app), sends a request to his service provider to instantiate a new MEC application via the `Mx2` reference point, that connects with the user application lifecycle management (LCM) proxy which authorises requests from device application.
2. The user application LCM proxy exchanges, then, information with the Operation Support System (OSS) and the Multi-access edge orchestrator (MEAO) for the management of this instantiation request via the `Mm8` in the MEC system.
3. The OSS decides whether to grant the request based on the user subscription information and service provider policies. OSS forwards approved requests to the MEAO for further processing through the `Mm1` reference point.

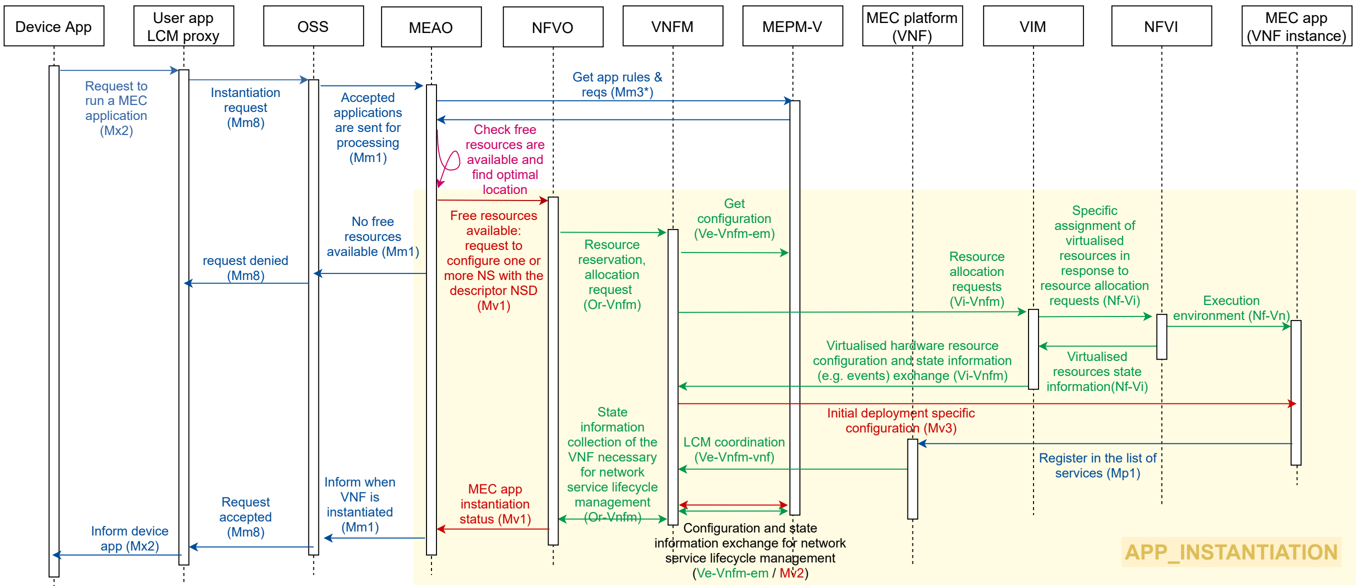

Fig. 2. MEC application instantiation process flowchart.

4. The MEAO communicates with the MEC Platform manager-NFV (MEPM-V) via `Mm3` and obtains the application rules, requirements and defined traffic rules, if any.
5. Since the MEAO is the centre of the MEC system at the management level, it must have a global view of the MEC system's free resources, available services and topology. The MEAO decides on the optimal location for the new MEC application based on this information. In case there are free resources, the MEAO requests the Network Function Virtualisation Orchestrator (NFVO) to configure one or more Network Services (NSs) to manage the VNFs of the MEC application, generating and on-boarding an Network Service Descriptor (NSD), and requesting the instantiation of an NS. The descriptor is a template used by the NFVO with information for the instantiation of the NS, formed by one or more VNFs. In case there are no free resources, the MEAO informs the OSS through `Mm1` which, in turn, will inform the User application LCM proxy through the `Mm8` interface.
6. The next step is to reserve the necessary resources for deploying the requested MEC service and then, the allocation of the application is requested. The flow of this communication starts at the NFVO via `Or-Vnfm` to the Virtual Network Function Manager (VNFM), since the NFVO manages all VNF LCM operations with VNFM.
7. As the ME Platform VNF is considered as a network function, the VNFM gets VNF configuration of the MEPM-V through the `Ve-Vnfm-em` interface.

8. Likewise, the VNFM via `Vi-Vnfm` exchanges information with the Virtualisation Infrastructure Manager (VIM) about resource allocation. The VIM is responsible, among other functions, of allocating, managing and releasing virtualised resources of the Virtualisation infrastructure.
9. Therefore, the VIM performs the allocation of specific resources for the VNF in the Network Functions Virtualisation Infrastructure (NFVI), in response to the request to allocate resources through `Nf-Vi` interface.
10. Finally, the MEC application is instantiated on the virtualisation infrastructure based on the requirements and configuration described in the NSD.
11. Once the MEC application is instantiated, the system is informed. First, the NFVI reports the status of the virtualised resources to the VIM through the `Nf-Vi` interface that connects them.
12. The next step, the VIM sends through Vi-Vnfm to the VNFM the configuration of virtualised hardware resources and the exchange of state information such as events.
13. Then, through the `Mv3` reference point, the VNF Manager and the VNF instance of the MEC application exchange data related to the initial configuration specified.
14. The MEC application instance is then registered in the list of services to the MEC platform via the `Mp1` reference point. Since, as it has been said previously, VNF instances act in virtualised environments as another service managed by the VNFM.
15. Through the `Ve-Vnfm-vnf` interface, the VNF manager coordinates with the MEC LCM platform to manage the subscription on the LCM event.
16. Next, configuration and status information is exchanged between the MEC-NFV platform manager and the VNFM for network service lifecycle management via two interfaces that are: `Mv2` and `Ve-Vnfm-em`.
17. Then, the NFVO, as responsible for the management of the life cycle of the network services, exchanges information with the VNFM about the VNF status required for network service lifecycle management through `O-Vnfm`.
18. Through the `Mv1` interface, the NFVO informs the MEAO about the ME app VNF mapping and state as the NFVO is in charge of the orchestration of the set of ME app VNFs as one or more NFV NSs.
19. Likewise, the MEAO informs the OSS through the `Mm1` interface that the requested instance has been carried out.
20. Finally, the OSS informs the user application LCM proxy that the request has been accepted using the Mm8 interface and this, in turn, informs the Device application via `Mx2`.

3.2 Migration

In case the MEC application has to be migrated to guarantee Quality of Service (QoS) requirements such as low latency when changing service areas as the device is moving, a clone of the MEC application must be instantiated on a more optimal host with enough available resources. Therefore the information from the old VNF must be copied to the new instance. Once the transfer is complete,

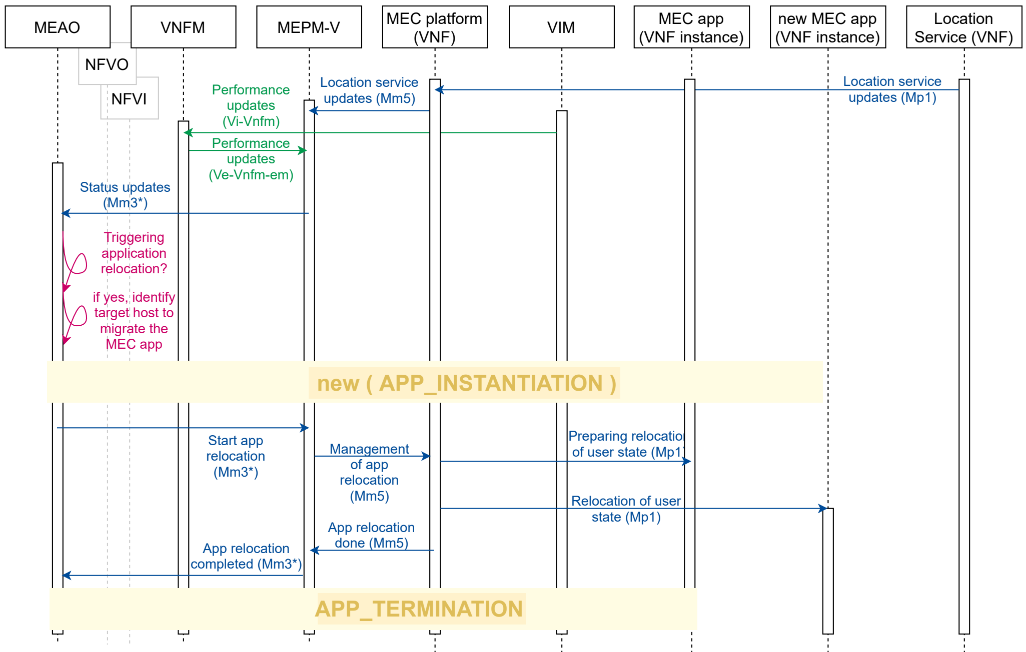

Fig. 3. Migration process flowchart of a MEC application.

the old VNF must terminate. Figure 3 shows the flow of the migration process based on the trigger from the location service [2] which provides location-related information for authorised applications. This information is sent through the `Mp1` interface to the MEC platform (VNF) where one of its functions is to host MEC services. Thus, the location of the MEC application will be updated.

In this way, the MEC platform is informed where are the users that use the hosted MEC services. This location information travels through the `Mm5` interface from the MEC Platform (VNF) to the MEPM-V that manages the life cycle of the MEC applications. To ensure QoS, the system must keep track of the performances as the users interact with the MEC applications. For these purposes, the VIM sends performance information about the virtualised resources to the VNF Management via `Vi-Vnfm`. From there these performance updates are sent to the MEPM-V via `Ve-Vnfm-em`. The performance status together with the location information for all MEC applications are sent to the MEAO through the Mm3 interface. The MEAO needs to have this information since it must have an overview of the MEC system and must detect when it is necessary to migrate a MEC application when its QoS requirements are not met. In case the QoS requirements of a MEC application are not met, the MEAO must identify a new target host where the MEC application is going to be migrated. To implement the smart relocation feature the MEAO needs to start a new MEC application on the target host, initiate relocation and then terminate the old application instance on the original host.

Once an instance of the new MEC application has been created, the MEAO starts relocating the application context and user state in coordination with the MEPM-V via `Mm3`. At the same time, the current running application instance status must be maintained as this process must be transparent to the user. The MEPM-V with the information received from the MEAO, such as traffic rules, manages the relocation of the MEC application through the `Mm5` interface together with the MEC Platform (VNF). Next, the MEC Platform (VNF) through `Mp1` prepares the relocation of the user state. Once the user sessions have been transferred, the MEC Platform (VNF) performs the relocation of the user state to the new instance of the MEC application via `Mp1`. Next, the MEC platform (VNF) informs the MEPM-V that the relocation has been completed through the `Mm5` interface. And the MEPM-V informs the MEAO via `Mm3` that the relocation process has been completed. Finally, the MEAO must terminate the old MEC application. To do so it launches a termination process of the old instance of the application.

3.3 Termination

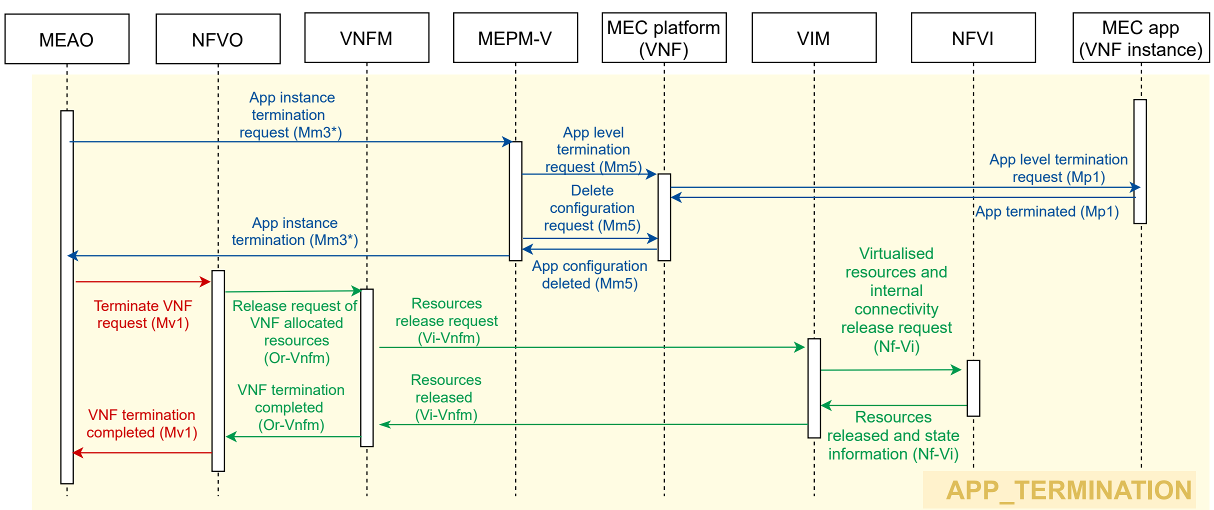

Fig. 4. Termination process flowchart of a MEC application.

The application instance termination process takes place in two steps (see Figure 4): first, the ME platform configuration and data plane must be removed, and the second step is to terminate the ME app VNF. To do this, the MEAO communicates with the MEPM-V through `Mm3` to request the termination of the application instance. The request arrives at the MEC platform (VNF) through the `Mm5` interface. The MEC platform then sends an application level termination request to the old MEC application instance via `Mp1`. After the application has been terminated, the MEPM-V asks the MEC platform (VNF) to remove the platform configuration via `Mm5`. Once the confirmation is received, MEPM-V informs the MEAO that the application instance termination process has been completed by `Mm3`. The next step takes the MEAO to request the NFVO to terminate the VNF of the application via `Mv1`. To do this, the NFVO communicates with the VNFM by `Or-Vnfm` to request the release VNF allocated resources. The VNFM requests the release of these resources to VIM through `Vi-Vnfm`. Then the VIM, via `Nf-Vi`, requests the NFVI to release the visualised resources and internal connectivity. The NFVI releases the virtualised resources and sends status information back to the VIM, which forwards the confirmation that the resources have been released to the VNFM through `Vi-Vnfm`, that sends the confirmation that the termination of the VNF has been completed to the NFVO via `Or-Vnfm`. The NFVO by `Mv1` communicates it to the MEAO, which must be aware of the status of the whole MEC system.

4 Discussion and Conclusions

The emergence of new generation devices with ultra low latency requirements is increasing every day. There are many works that are committed to uniting MEC NFV-based and 5G technologies but there are still some concepts to be defined. With this work we want to provide a solution to the orchestration of the migration process in order to further the work done by ETSI and tackle the smart relocation issue in MEC-NFV implementations. The identification of the roles of each MEC component and the activation of the defined interfaces needed to implement smart relocation are a first step of the research. Our future work will focus on bridging the gap of implementing smart relocation in MEC-NFV by detailed analysis of the data exchanged via the identified interfaces and, finally, a practical implementation of the outlined workflows.

The ETSI has not yet defined the smart relocation process in MEC NFV-based environments, even though their documents indicate that the MEC system that supports smart relocation must be able to perform relocation of a MEC application instance from one MEC host to a different host within the system or to cloud system outside the MEC system. Therefore, in this work we assign to each of the involved components of the NFV-based MEC generic architecture a specific part in the smart relocation process. We produce a mapping of functions and blocks of the NFV-based MEC architecture, thus continuing the work addressed by the ETSI. As it can be seen in [2] this standardisation work continues with a new variant of the general MEC-NFV reference architecture extending it with federations. In a federated environment MEC services and applications are shared with each federation having its own computing resources at the edge. In this new variant of the MEC architecture a new element is added, MEC Federator (MEF), that allows the MEC system to communicate with other MEC systems and other non-MEC systems. In this constellation, the smart relocation function must allow the migration of the MEC application between MEC systems as well as between a MEC system and another non-MEC system, which will be addressed in future works.

References

1. K. Koufos *et al.*, "Trends in Intelligent Communication Systems: Review of Standards, Major Research Projects, and Identification of Research Gaps", J. Sens. Actuator Netw. 2021, 10, 60.
2. ETSI GS MEC 003: "Multi-Access Edge Computing (MEC); Framework and Reference Architecture". V3.1.1 (20220-03).
3. C. Bernad, P.J. Roig, S. Filiposka, S. Alcaraz , K. Gilly, "Challenges of Implementing NFV-based Multi-Access Edge Computing Environments," In 2021 29th Telecom. Forum (TELFOR), 2021, pp. 1-4.
4. C.H. Hong, B. Varghese, "Resource management in fog/edge computing: a survey on architectures, infrastructure, and algorithms", ACM Computing Surveys (CSUR), 2019, 52, 5, pp.1-37.
5. S. Wang, J. Xu, N. Zhang, Y. Liu, "A survey on service migration in mobile edge computing", IEEE Access, 2018, 6, pp.23511-23528.
6. A.I. Jehangiri *at al.*, "Mobility-aware computational offloading in mobile edge networks: a survey", Cluster Computing, 2021, 24, 4, pp.2735-2756.
7. Z. Liang, Y. Liu, T.M. Lok, K. Huang, "Multi-cell mobile edge computing: joint service migration and resource allocation", IEEE Transactions on Wireless Communications, 2021.
8. M. Xu *et al.*, "PDMA: Probabilistic service migration approach for delay-aware and mobility-aware mobile edge computing", Software: Practice and Experience, 2022, 52, 2, pp.394-414.
9. Y. Guo, C. Jiang, T.Y. Wu, A. Wang, "Mobile agent-based service migration in mobile edge computing", International Journal of Communication Systems, 2021, 34, 3, p.e4699.
10. S. Wang *et al.*, "Dynamic Service Migration in Mobile Edge Computing Based on Markov Decision Process," in IEEE/ACM Transactions on Networking, vol. 27, no. 3, pp. 1272-1288, June 2019.
11. Meng, X. and Lu, W. Container-Based Fast Service Migration Method for Mobile Edge Computing. Journal of Ciruits, System and Computers. 2021, p.2250117
12. Li, B., Cheng, B. and Chen, J. "An Efficient Algorithm for Service Function Chains Reconfiguration in Mobile Edge Cloud Networks". In proc. of the 2021 IEEE International Conference on Web Services, ICWS 2021, pp. 426-435.
13. Kiran, N., Liu, X., Wang, S., Yin, Ch. "Optimising resource allocation for virtual network functions in SDN/NFV-enabled MEC networks". IET Communications, 15(13), 1710-1722. 2021.
14. Z. Gao *at al.*, "Deep reinforcement learning based service migration strategy for edge computing," In 2019 IEEE International Conference on Service-Oriented System Engineering (SOSE), 2019, pp. 116-1165.
15. M. Chen *at al.*, "A dynamic service migration mechanism in edge cognitive computing", ACM Transactions on Internet Technology (TOIT), 2019, 19, 2, pp.1-15.
16. B. Blanco *et al.*, "Technology pillars in the architecture of future 5G mobile networks: NFV, MEC and SDN", Computer Standards and Interfaces, 2017, 54, pp.216-228.

17. ETSI GS MEC 017: “Deployment of Mobile Edge Computing in an NFV enviroment”. V1.1.1 (2018-02).
18. S.N *et al.*. Afrasiabi, S. Kianpisheh, C. Mouradian, R.H. Glitho, A. Moghe , ”Application components migration in NFV-based hybrid cloud/fog systems,” In 2019 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), 2019, pp. 1-6.