Mobile edge computing in Internet of Unmanned Things (IoUT)

Nina Slamnik-Kriještorac and Johann M. Marquez-Barja

Abstract The role of this chapter is to describe the promising role of Multi-Access Edge Computing (MEC) in enhancing the operations of various applications that fall under the umbrella of the Internet of Unmanned Things (IoUT). The value that MEC systems bring to 5G and beyond networks is unprecedented, and as such, it is extensively studied and recognized by both industry and academia, due to the opportunities to dynamically create means for deploying services in close proximity from the mobile end users, through the optimal deployment of edge servers that offer both computing and communication capabilities with enhanced Quality of Service (QoS) and Quality of Experience (QoE). Although MEC enables a significant reduction of end-to-end latency, and optimal utilization of bandwidth compared to traditional cloud computing deployments, maintaining service continuity for mobile users still requires complex management and orchestration systems to coordinate the deployment and operation of services on different MEC servers that are placed on the fixed locations. Thus, blending the concepts of MEC with Unmanned Aerial Vehicles (UAVs) is expected to play an essential role in mitigating the aforementioned challenge, thereby achieving ubiquitous connectivity for mobile users. Leveraging their flexibility and mobility, UAVs can be efficiently spawned at critical locations to boost wireless communication for various applications. The MEC-enhanced IoUT application domains covered in this chapter aim to enable scalability for boosting the operations in smart cities and suburban areas, through studying the advanced architectures, benefits, and challenges, for various use cases such as disaster management, intelligent healthcare, intelligent traffic, and smart transport & logistics, among the others.

Nina Slamnik-Kriještorac
University of Antwerp-imec, IDLab-Faculty of Applied Engineering, Sint-Pietersvliet 7, e-mail: nina.slamnik@imec.be

Johann M. Marquez-Barja
University of Antwerp-imec, IDLab-Faculty of Applied Engineering, Sint-Pietersvliet 7, e-mail: johann.marquez-barja@uantwerpen.be

1 Multi-Access Edge Computing (MEC) concepts

The improvements that digital technologies have brought over the past few decades are unprecedented, as they are aiming to improve *quality of life* by transforming various spheres of modern societies such as health, transportation, logistics, and automotive, into much safer and more efficient sectors. One of the main prerequisites for digital transformation is the connectivity of devices (e.g., phones, vehicles, sensors, drones, facilities, wearables) that are step-wise becoming technological and intelligent, thereby forming a ubiquitous Internet of Things (IoT) system.

This connectivity of devices enables IoT systems to instantaneously share data via the network. However, the number of connected devices is increasing, which leads to a tremendous increase in network traffic. Such an increase is affecting the response of network services, thereby producing delays that might not be acceptable for mission-critical services such as emergency and e-health ones, or services addressing natural disasters, which all together impose stringent requirements on network latency, reliability, and bandwidth. To address the aforementioned issues, i.e., to accommodate critical services with required network performance, edge computing technologies, and in particular Multi-Access Edge Computing (MEC), are emerging as an essential asset of 5G and beyond ecosystems that are offering both the computing resources and the computing capabilities at the network edge, i.e., significantly closer to the end-users (e.g., vehicles, pedestrians, vessels, or IoT devices). Thus, these MEC systems are constructing both *terrestrial* and *aerial* edge cloud platforms, which can be tailored to particular verticals (e-health, automotive, transport & logistics, etc.), and as such can be accessed with a decreased latency, an improved bandwidth, and a significantly decreased backhaul network utilization [1].

All the above-mentioned services that are tailored to verticals can be deployed in MEC-enhanced 5G and beyond systems, by being mapped into the three main types of network slices: massive Machine Type Communication (mMTC), ultra-Reliable Low Latency Communication (uRLLC), and enhanced Mobile Broadband (eMBB) [2]. The process of mapping vertical services to these three slice categories depends on the service requirements on latency, bandwidth, jitter, and reliability, among other Quality of Service (QoS) parameters. By assigning a particular vertical service to the network slice, a set of computing and network resources needs to be allocated from an end-to-end perspective, thus, entailing User Equipment (UE), network edge, radio, core and cloud. This accommodation requires networks to be more adjustable and programmable, enabling more flexibility in the way how resources are being managed and allocated. In particular, network programmability is enabled through network softwarization, which is based on the principles of i) virtualizing services and network functions, i.e., the Network Function Virtualization (NFV) [3, 4, 5], and ii) programming connectivity and traffic flow between the aforementioned Virtual Network Functions (VNFs), i.e., by achieving Software Defined Networking (SDN). These VNFs are further utilized as building blocks for designing complex and robust network services that can run either at cloud or MEC systems.

Being built upon the same programmability and flexibility principles, MEC technology is entirely based on NFV and SDN, and it is bringing an on-demand accessible cloud-alike computing system closer to the end users. As such, MEC technology is considered a key 5G and beyond component for enabling ultra-reliable and low-latency connectivity to distributed services [1, 6]. In particular, MEC systems are combining computing engines that are less resourceful and less powerful than distant cloud systems, but locating them either within the Radio Access Network (RAN) of mobile operators, or their transport network.

The ultimate goals of designing and deploying MEC platforms at the edge of 5G and beyond ecosystems are i) to reduce the latency in accessing services that were previously deployed on top of the distant locations in the communication systems, such as clouds or private data centers, and ii) to enable offloading heavy computing tasks from a UE, and eliminating the need for running complex and resource consuming tasks at the user side. In the following sections, we introduce the standardized MEC framework defined by European Telecommunications Standards Institute (ETSI), and we also discuss various types of MEC-based network architectures for Internet of Unmanned Things (IoUT) systems that are defined in the relevant literature.

1.1 The standardized MEC framework

To standardize the work on creating comprehensive MEC systems, including sets of Application Programming Interfaces (APIs) for essential MEC interfaces, the ETSI created an Industry Specification Group (ISG), i.e., ISG MEC [6, 7]. The goal of creating this standardized and open MEC environment is to boost the incorporation of MEC systems into upcoming new generations of communication networks and to create a comprehensive knowledge base for a better understanding of MEC service and resource orchestration. Such a framework is creating means for the efficient and seamless integration of diverse applications from different vendors, service providers, and third parties [7], regardless of the type of verticals (e.g., automotive, e-health, transport & logistics) for which these applications and services are built.

In particular, ETSI NFV ISG defines an NFV-based architectural framework, as illustrated in Figure 1, which consists of various building blocks and reference points needed for hosting applications within the MEC platform.

As presented in our previous work [8], the two main components of ETSI NFV MEC architecture are NFV Orchestrator (NFVO) and VNF Manager (VNFM), which are altogether forming a so-called ETSI NFV MANO [9]. On one hand, the NFVO is in charge of performing the orchestration operations, i.e., making decisions for the service deployments and their reconfiguration while taking into account resource availability within the NFV Infrastructure (NFVI) at the designated MEC platform. On the other hand, VNFM is responsible for translating those decisions made by NFVO into the life-cycle management actions of VNFs, i.e., performing the operations such as VNF instantiation, scaling, relocation, and termination, which are ultimately making changes on the allocated NFVI. Thus, VNFM is following the orchestration decisions and instructions provided by NFVO, thereby managing all network service instances running in the terrestrial and aerial MEC systems.

Another important building block of the ETSI MEC architectural framework is the Virtual Infrastructure Manager (VIM), which represents a management system for NFVI that is used for the deployment and running of services and VNFs. In particular, the roles of VIM are: i) to perform resource allocation, management, and release of previously allocated virtualized resources, ii) to prepare underlying NFVI for running the required software images as a base for the requested VNFs, and iii) to collect fault reports and performance measurements about allocated and available virtualized resources.

One of the main advantages of this open source architecture is an increased opportunity for facilitating the implementation of MEC-based network architectures that are interoperable among diverse network operators, vendors, and vertical industries [8]. If applications and services are designed in a cloud-native manner, as a Virtual Machine (VM) or a container-based application, it is important to note that MEC platforms can be used for deploying and managing services regardless of the vertical industry, defining the same set of rules for any type of MEC application that can be further tuned depending on the specific application requirements. We refer to those vertical-oriented MEC applications as Edge Applications (EdgeApps) later in this chapter.

The compatibility of MEC with various vertical stakeholders is particularly important to emphasize since different MEC platforms comprise a plethora of virtualized and physical resources, diverse services, and applications of various stakeholders. In a such strongly heterogeneous environment, interoperability plays a crucial role, which can be assured only by following the standardization guidelines and recommendations.

1.2 MEC in the context of IoUT

In the previous section, we elaborated more on the general concepts of MEC technology and its standardized architectural framework, which makes it applicable to any IoT system, including the IoUT services and applications. In this section, we steer the focus to the IoUT systems and different architectural styles that can be adapted to create on-demand edge computing systems in an IoUT fashion, where edge computing units are not necessarily statically deployed (terrestrial MEC), but instead could be on-boarded dynamically on the Unmanned Aerial Vehicles (UAVs).

The increasing demand for vertical services, such as those aimed at providing support for assisted or automated driving/navigation, remote control of IoT systems, etc., requires computing applications to help and serve the end users. These applications might be computation-intensive, and as such, they cannot run on the on-board units of UE, due to their low computational capabilities and finite battery capacity [11], which might not be sufficient especially if the application is running some Artificial Intelligence (AI)/Machine Learning (ML) model for forecasting or decision-making.

Thus, UAV-based MEC has started to gain an increased interest from both industry and academia when it comes to offloading computation-intensive tasks from user devices to distributed edge nodes in users' close proximity [11, 12]. In this case, MEC capabilities are embedded into UAVs, which are becoming mobile MEC systems that can be spawned at any location where more computational power or support from specific vertical MEC applications (e.g., automatic or assisted navigation) is needed.

However, a UAV used as an aerial MEC server is not the only architectural style in UAV-enabled MEC-based systems. According to Zhou et al. [11], UAVs can be also used as relays to assist users to offload their computation task to the terrestrial MEC system, but they can also act as users of vertical applications running at the MEC nodes or users that need to offload their computation task to closest MEC. In Figure 2, we illustrate the following three architectural styles.

UAV acting as a relay node

In this architectural style, UAV is deployed as a relay node between a UE and terrestrial MEC system. In the case illustrated on the left-hand side of Figure 2 (e.g., use case of vehicular emergency systems), UE is an emergency vehicle that is consuming mission-critical services running at the MEC, towards supporting emergency situations on the roads. In particular, these MEC services can be used to proactively notify other civilian vehicles on the road about the arrival of an

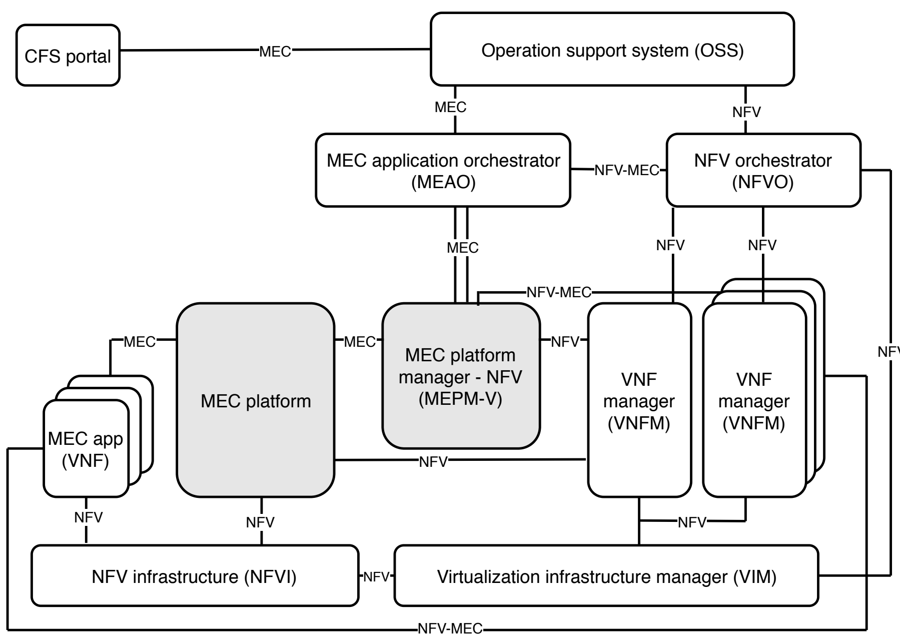

Fig. 1 ETSI MEC framework [8, 10].

emergency vehicle (fire brigade, ambulance, or police), as described in [13], so that they can properly and calmly maneuver out of the lane where this emergency vehicle is driving. The main goal of such services is to increase awareness about emergency vehicles on the road and to increase mission success by significantly decreasing the overall response time.

The response time can be decreased if all vehicles in front of the emergency one clear the lane based on the timely received notification from the MEC server. To make this mission successful, emergency and civilian vehicles need to be equipped with communication capabilities, e.g., with a 5G modem that will establish 5G connectivity with the MEC server, and with an on-board unit for message production and processing. Also, the emergency vehicle needs to produce emergency messages e.g., based on Intelligent Transportation System (ITS) standards, and to feed them into auxiliary services running at the terrestrial MEC servers. In this case, UAV can be the first contact point for emergency vehicles, where emergency messages are received, processed, and only relevant data is further sent to MEC server for processing. Given the increased load on terrestrial MEC servers, which are receiving requests and messages from various users, UAV-based MEC can further decrease the

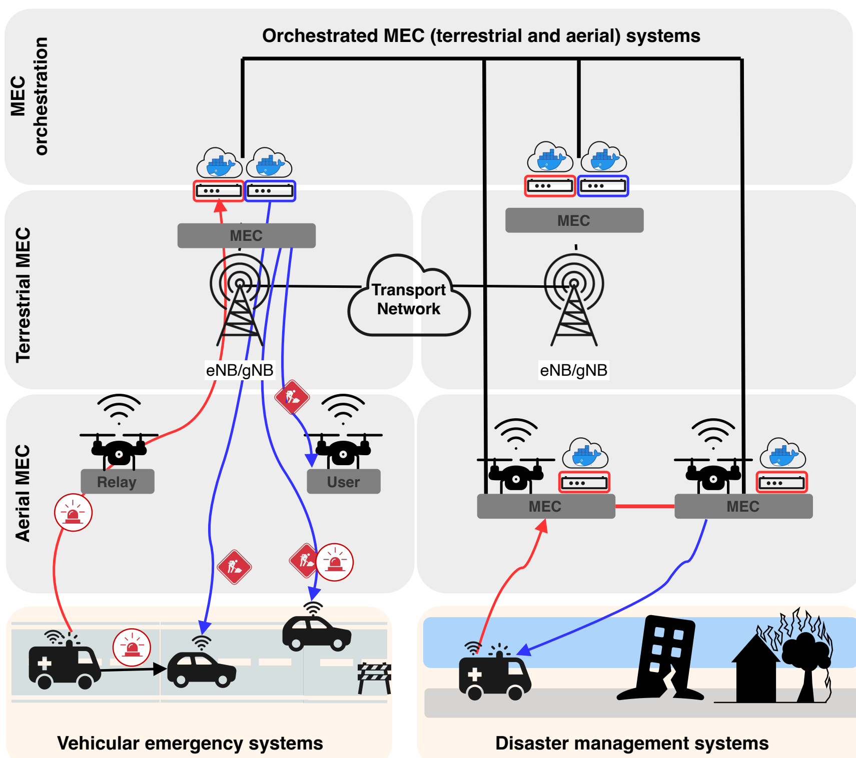

Fig. 2 A high-level overview of MEC system deployments for IoUT.

delay in message transfer from emergency vehicles. This setup does not only decrease the delay in message transfer due to the single-hop transmission delay compared to multi-hop in the case of terrestrial MEC nodes, but it might also improve uplink bandwidth utilization as UAV may decrease the payload sent to the MEC servers. This can be particularly valuable in the case of video streaming as if the emergency vehicle is transferring video data from on-board cameras to the MEC, UAVs can help with some image recognition tasks that process camera feed data, and instead of sending raw camera data to the MEC servers, they can only relay important findings (number of vehicles in front, obstacles on the road, etc.) that can further help MEC services to derive navigation decisions.

UAV acting as user

UAV can be also a user that either requires more computational power for executing certain tasks or acts as a consumer of services running at terrestrial MEC systems. Nowadays, UAVs are advantageous in terms of their low cost, on-demand deployment and high maneuverability [12], and as such, they can be deployed on-demand as mobile users that are in charge of executing critical tasks in various scenarios, such as emergency situations, target tracking, smart delivery, among others [12]. In case of task offloading, sometimes UAVs require more power and battery capacity to compute their own tasks, such as trajectory optimization [11, 40, 41, 47] to deliver packages in an optimal manner, image recognition to support target tracking mission, and sensor data fusion for object detection to avoid obstacles when landing. In such a scenario, UAVs can partially or fully offload their tasks to terrestrial MEC systems. In parallel, UAV can also act as consumers of MEC services from which they can collect valuable data about the traffic conditions on the roads or weather conditions collected from weather forecasting systems, which are further used to optimize UAV's maneuver trajectory [40].

UAV acting as MEC system

In this case, UAV is being used as an aerial MEC system that delivers services for the end/ground users, which can be emergency vehicles, other civilian vehicles, Vulnerable Road Users (VRUs), among others, and/or provides them with task offloading capabilities [41, 47]. On the right-hand side of Figure 2, we illustrate a scenario where UAVs are deployed as a distributed edge cloud to support disaster situations, thus, exhibiting *close-to-users* edge computing capabilities that provide infrastructural means for disaster management systems. In this case, UAVs have computational and battery capacity, and act as miniaturized MEC servers [12], which either run MEC services that are consumed by the ground users or execute the tasks that are offloaded from the ground users due to their computational limitations. One example scenario of ground users consuming MEC services from UAV-based MEC servers is shown in Figure 2, where an emergency vehicle (i.e., UE) is retrieving the most recent updates about the emergency situation caused by a natural disaster, whereas the updates are produced by MEC services that collect environmental data from the sensors or other users such as VRUs and other citizens in the surrounding area. Instead of using distant cloud or MEC systems that are not deployed in close proximity, and thus, cannot guarantee required levels of service latency and bandwidth, UAV-based MEC deployment is boosting the rescue operation by selecting an optimal trajectory with improved communication channel conditions, thereby increasing the chance of mission success.

Furthermore, Song et al. [12] emphasize the importance of multi-UAV MEC deployments, as in the case of aerial MEC, computing resources and available energy can be easily depleted if all the ground users rely on the single UAV for edge computing. In addition to energy constraints and potential unavailability of computing resources in case the battery needs to be recharged, it is important to properly design the aerial MEC system, such that MEC services are distributed [46, 48], thereby ensuring redundancy, service reliability of five nines (99,999%), and the required latency for task execution. In Figure 2, we also point at the role of orchestrators, such as NFVO described in 1, which needs to orchestrate all MEC resources and services either deployed on the terrestrial or aerial MEC systems, in order to ensure proper operation of all services and offloaded tasks. However, there are various challenges that arise in all of the listed UAV-based MEC architectural styles, and especially in the case of multi-UAV settings due to the need for cooperation among all UAVs in the system. These challenges are tackled later in Section 4, while orchestration is closely studied in Section 3.

2 Applications for UAV-enhanced MEC systems

UAV technology has emerged as a prominent choice for commercial applications in various scenarios, including those without network infrastructure as in the case of rural areas and natural disasters [24]. This popularity has risen mainly due to the flexibility of UAV deployment and Line-of-Sight (LoS) capabilities that are improving channel characteristics, thereby boosting signal and service quality, while maintaining high mobility [24, 25, 26, 27, 28, 29]. In Section 1.2, we discussed different architectural styles for aerial MEC deployments, while this Section presents various application scenarios for UAV-based MEC deployments, and elaborates on the MEC application design principles.

2.1 Application scenarios

Given the opportunities for low-latency services that support various verticals, such as automotive, e-health, and transport & logistics, aerial or IoUT-based MEC systems can be flexibly spawned at any location, providing low-latency access to computing

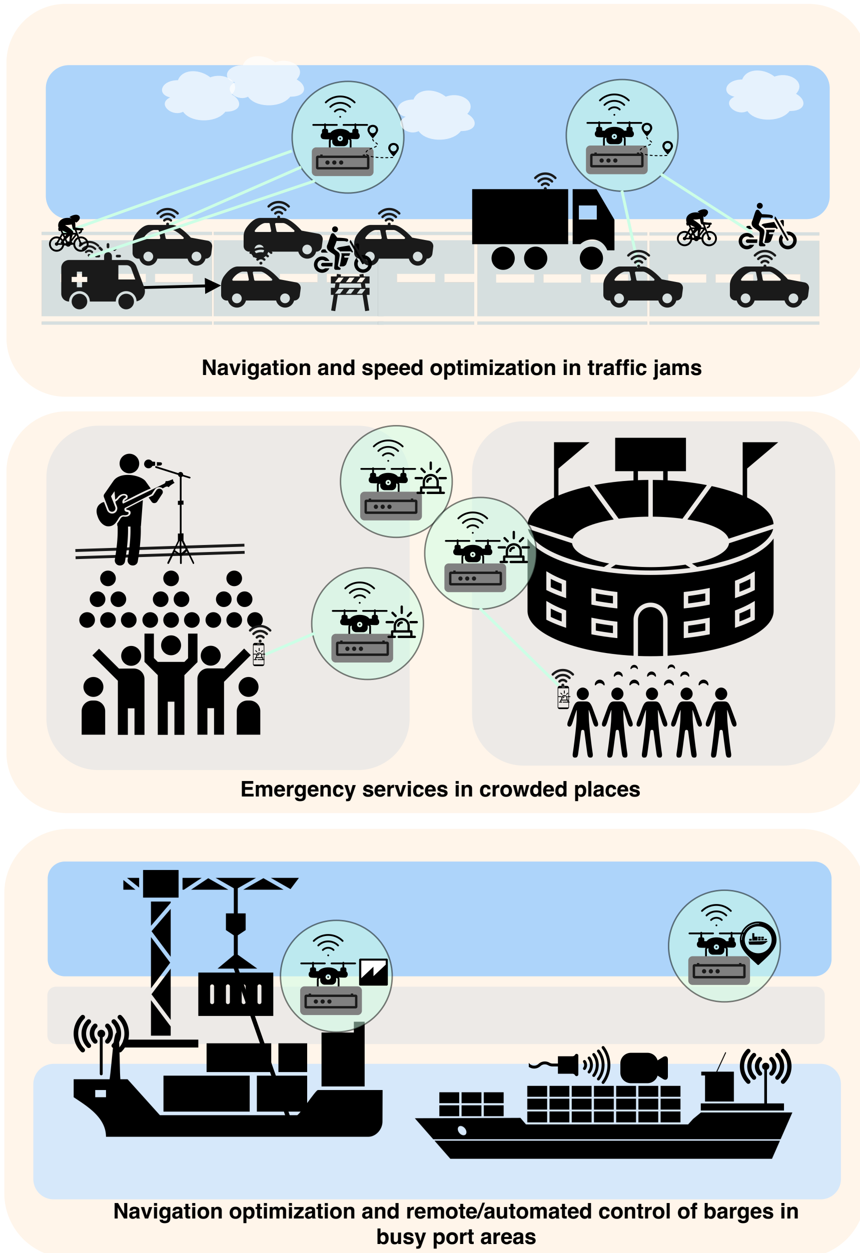

Fig. 3 Application scenarios for aerial MEC deployments - part 1.

engines either for consuming vertical services (e.g., navigation support, emergency notifications, and driving assistance) or for offloading computation-intensive tasks from ground users. Some of the application use cases are illustrated in Figures 3 and 4 and described below.

- **Navigation and speed optimization in traffic jams:** Highways are becoming excessively busy even out of the rush hours, and there can be a massive number of vehicles that demand computation-intensive services. In busy city and suburban areas, there can be hundreds of vehicles on the highways that are simultaneously using navigation support during rush hours. To help drivers, as well as fully autonomous vehicles, to derive optimal decisions on the maneuvering process, various types of data need to be collected from infrastructure (base stations, roadside units, traffic lights, etc.) and other vehicles (sensor and lidar data, speed/heading/location, etc.), in order to increase situational awareness on the

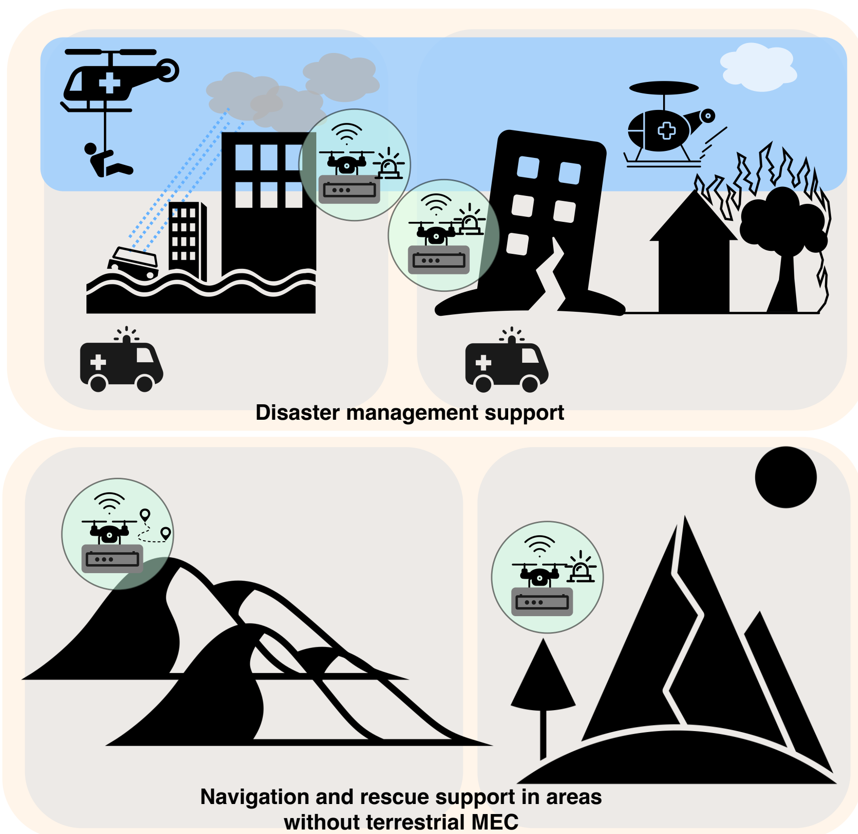

Fig. 4 Application scenarios for aerial MEC deployments - part 2.

highways. Thus, there is an increased amount of data that needs to be processed, while leveraging advanced data analytics or ML techniques, which usually go beyond the computing capabilities of a single vehicle. In addition, services which support vehicles are extremely sensitive when it comes to latency (order of milliseconds), which means that deployments of such services need to be located as close as possible to the end users. Therefore, due to such requirements on latency and computing capabilities, UAV-based MEC nodes seem as a viable solution to exploit towards assisting existing terrestrial MEC nodes in providing services for the ground users. Such distributed MEC deployment can be spawned at various locations on the highways [43]. The cooperation of UAVs within the distributed MEC deployment can be further optimized so that the energy consumption is balanced among UAVs, as well as their computational load, in order to avoid delays in response to service demands from the ground users (i.e., vehicles). Another important type of service that could support vehicles in traffic jams is speed optimization service, which optimizes fuel consumption by defining an optimal speed for vehicles in different areas on the highways. Such support from aerial MEC systems assists ground users by decreasing the outage probability and improving service quality perceived by those users [11].

- **Emergency services in crowded places:** Due to heavily saturated radio channels, crowded places such as concerts or sports events usually suffer from decreased signal strength. As this can be fatal in case of emergency situations, such places demand some service performance boosters in order to properly react when people need medical support. Thus, MEC nodes can be used for hosting emergency responding services that provide support for end users, by communicating with emergency responders (e.g., emergency management authorities) [47, 48]. Such services could lead to a much faster response from the ambulances and authorities that provide emergency support. By leveraging UAVs, MEC system can be quickly established and scaled up if needed, depending on the number of people and type of occasion.
- **Optimization of port operations and remote/automated control of barges:** Busy port areas suffer from large waiting times for barges in the docks, due to unoptimized mechanisms for loading/unloading and docking. In such areas, safety measures are significantly important, and digital technologies need to be leveraged towards providing more support to staff. There are some recent efforts that reflect on utilizing support from 5G-enhanced vertical services in port environments to remotely operate barges, cranes, trucks, and skid-steers [32]. Due to strict requirements on downlink latency for control operations (¡20 ms), and uplink bandwidth for collecting camera feeds from the ports and barges (25 Mbps per user), 5G and MEC emerged as promising technologies to optimize such operations. Given that distribution of barges in the large ports can be scarce, while the presence of trucks and skid steers can be dense in the loading/unloading areas, UAV-based MEC nodes are a practical solution when optimization service or task offloading mechanisms are instantaneously required at various locations that might have low signal strength due to heavy metal constructions.

- **Disaster management support:** In case of natural disasters (e.g., earthquakes, floods, and fires), all terrestrial network infrastructure usually gets destroyed, including data centers and fixed MEC systems collocated within base stations or roadside infrastructure [11, 12]. However, it is extremely important that rescue and reconstruction operations are executed in time, and for that purpose, UAV-based MEC systems play an important role.
- **Navigation and rescue support in areas without terrestrial MEC:** In distant areas like deserts, mountains, wildernesses, and rural areas, the governments need to provide support for the citizens by monitoring the environment and deriving strategies that protect both the citizens and the environment. Given that terrestrial MEC systems cannot be efficiently placed in such areas, while the cost of deployment also remains high [11], dynamic spawning of aerial MEC platforms seems to be a valid alternative, especially for providing navigation and rescue support for citizens.

2.2 MEC Application design principles

Given the ever-increasing number of users and end devices, 5G and beyond networks are designed to be entirely flexible and programmable by defining network services as software pieces that communicate through software-driven procedures. In such communication ecosystems, the need for a cloud-native architecture of services and applications becomes a defacto choice, as it enables rapid and flexible deployment of services, with reduced Capital Expenditure (CapEx) and Operational Expenditure (OpEx) through network automation. Therefore, the MEC systems and MEC applications need to follow the same principles, since resources are limited compared to resourceful cloud computing systems, while MEC applications are still expected to yield a high level of flexibility and short response time.

As the entire idea of employing UAVs to serve as flexible MEC nodes with the ability to get spawned dynamically at any location, it is of utmost importance that deployment of services and applications on the UAV MEC infrastructure is efficient and as simple as possible. Thus, to achieve such application deployment efficiency, it is important to properly modularize the overall MEC service and/or EdgeApp, designing it as a set of loosely coupled microservices, which communicate to each other via internal interfaces, while being separately managed, i.e., migrated/scaled according to the decisions made by orchestrators.

In this section, we present the methodology of designing and developing MEC applications for vertical services that can be hosted at UAV-based MEC systems for the purpose of serving ground users and providing task offloading mechanisms [41, 47]. In particular, these MEC services can be defined as a set of EdgeApps, which are virtualized network functions that are designed to i) abstract the complexity of vertical services, and ii) make vertical services network-aware, through defining network-specific requirements (e.g., imposing limits on tolerable latency and minimum bandwidth for service operation over 5G).

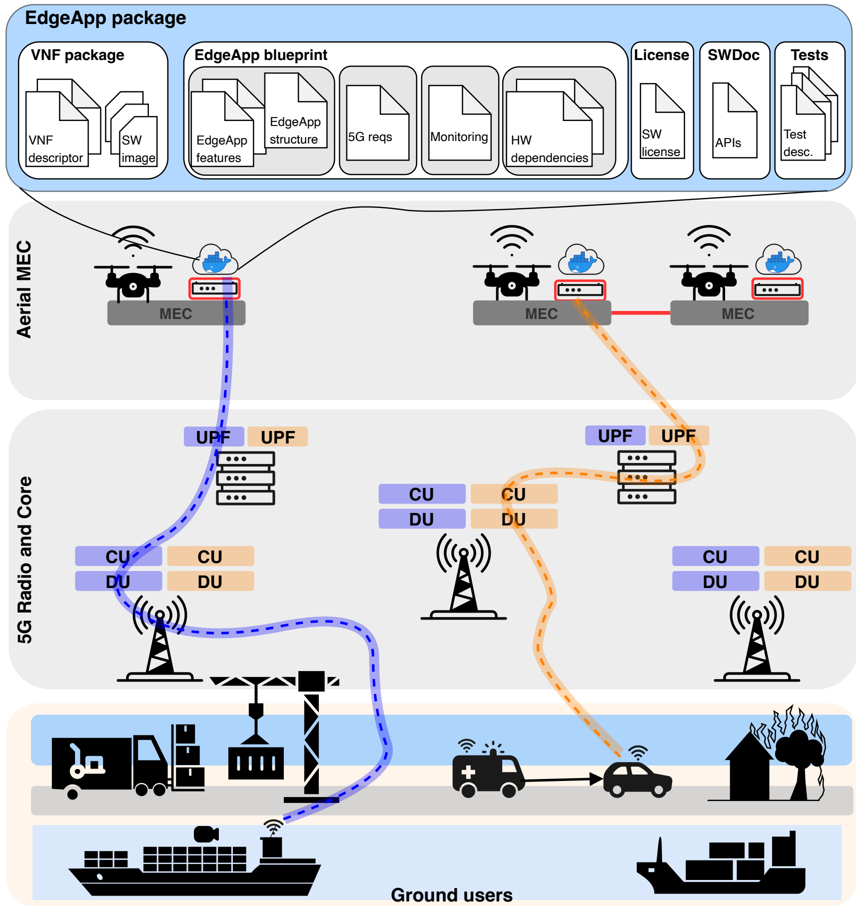

Fig. 5 Deployment of a MEC service as an EdgeApp, and its usage in 5G ecosystems leveraging on eMBB (purple) and uRLLC (orange) network slices; Definition of EdgeApp package;

If we focus on UAV-based MEC systems that are placed within a 5G ecosystem, vertical services running on top of these MEC nodes are enabled to achieve ultra-low latency (1-10 ms), ultra-high reliability (99.999%), and high data rates (up to 20 Gbps) [30], by leveraging on logical and virtualized networks, i.e., network slices, over the common network infrastructure. Thus, by designing and implementing the following types of network slices: uRLLC, eMBB, and mMTC, 5G is extending the perspectives for industry verticals (e.g., automotive, e-health, and Transport & Logistics (T&L)), and it is fostering new use cases (e.g., autonomous driving, remote navigation, teleoperation) that have not been possible with the previous generations of mobile communications systems, given the too stringent connectivity requirements that could have not been met [31].

However, to be able to benefit from 5G and beyond networks, the design of vertical services needs to be tailored to particular use cases, taking into account vertical service-specific requirements towards 5G. Some of these requirements could be: uplink bandwidth of 25 Mbps per user, and service interruption for automated barge control lower than 150ms [32]. To take into account these specific requirements, and to abstract the underlying complexity of networks, the design methodology for MEC vertical services introduces EdgeApps, as basic building blocks or microservices that form a MEC vertical service.

The external interfaces of those microservices are used for communication with i) end users, i.e., vehicles, VRUs, or barges, so that they can connect to the MEC service and send their real-time messages, ii) dissemination services, which will be used for message dissemination towards vehicles, and iii) orchestration entities that orchestrate MEC applications, and dynamically receive notifications from such applications to improve their life-cycle management (e.g., NFVO introduced in Section 1). To support dynamic UAV-based MEC environments where both MEC systems and ground users are moving, EdgeApps should rely on middlewares that provide location-transparent communication and data access that is not hindered by the ever-changing underlying network topology and infrastructure. Also, EdgeApps dealing with real-time constraints in mission-critical operations (rescue operation or emergency situations on the roads [47]) should be built on top of time-aware frameworks (e.g., Zenoh-flow[1]) to react effectively to any event in the system (e.g., network error, server congestion), allowing critical applications to fall back to default safe mode. One example of such operation could be a MEC service experiencing some unpredictable performance, which requires autonomous vehicles or barges to stop or slow down while orchestrates migrates/scales the involved microservices, i.e., EdgeApps.

The overall concept of deploying MEC services as EdgeApp is facilitating the creation, design, provisioning, life-cycle management, and performance evaluation, of vertical services in 5G network infrastructures [33]. Deployment-wise, a single EdgeApp can be considered as an atomic component of MEC services designed for verticals. This component can be i) dynamically deployed on the virtualized network infrastructure, ii) re-used and shared while building new vertical services, as well as iii) combined with 5G network slices to deliver the required performance for the mobile connectivity (e.g., required uplink bandwidth for camera streams, and end-to-end latency for remote or autonomous navigation) [33].

Such deployment is achieved by extending the orchestration-oriented models proposed by ETSI NFV, i.e., VNF descriptors (VNFDs) and Network Service Descriptors (NSDs), which are service-agnostic, and limited to the definition of computing resources, network functions in the chain, forwarding graphs and paths, virtual links, and internal/external connection ports [34]. In particular, EdgeApps extend the original VNF concept declaring i) service level information to simplify the design of vertical services, and ii) mobile connectivity requirements. This additional infor-

[1] Zenoh-Flow aims at simplifying and structuring (i) the declaration, (ii) the deployment and (iii) the writing of ”complex” applications that can span from the Cloud to the Thing (or close to it). More information: https://github.com/eclipse-zenoh/zenoh-flow.

mation is encoded as metadata in a EdgeApp blueprint, which is included in the EdgeApp package.

In Figure 5, we illustrate the EdgeApp package that includes i) the references to the VNF package that defines the orchestration procedures for an EdgeApp, ii) the EdgeApp blueprint, and iii) software licenses, software documentation, test cases, and target KPIs for automated validation. This Figure also provides an insight into the EdgeApp modeling, using as an example two EdgeApps, where the first one collects upstream camera data from the barges in the port environment (eMBB slice), and the second one distributes emergency notifications to vehicles on the roads (Ultra-Reliable and Low-latency Communication (URLLC) slice). Both EdgeApps are running on the virtual computing infrastructure within UAV-based MEC nodes, which are interconnected via 5G to the ground users. In particular, EdgeApps are composed of internal components that are realized as containers or VMs implementing the EdgeApp logic. The external endpoints of EdgeApps are providing them with connectivity to the 5G network, using the N6 interface[2] of the 5G system [35]. As illustrated in Figure 5, these endpoints are further associated to one or more 5G slice profiles, describing the network slice characteristics, as defined in the 3GPP Network Resource Model [36]. The EdgeApp model illustrated on top of Figure 5 is defined as an abstract model, in order to offer a service-oriented description of the EdgeApps, and to facilitate the verticals in the selection and composition of EdgeApps towards creating new vertical services for various use cases they want to build and test [33].

3 MEC orchestration

As we elaborated in the previous sections, MEC systems rely on the virtualized infrastructure resources (i.e., NFVI) to optimally deploy MEC applications, i.e., EdgeApps tailored to various use cases that need UAV-based MEC support. Thus, the main enablers of MEC in general, including aerial MEC, are NFV and SDN. While NFV is virtualizing MEC infrastructure resources, thereby creating means for deploying network functions as VNFs, SDN is in charge of programming the network links between those VNFs, allowing them to connect to each other via a network and decoupling the network control from the data plane. Therefore, one of the main conveniences of leveraging on NFV and SDN is the opportunity for achieving effective, flexible, and dynamic resource management within modern computing environments.

One example of a such computing environment is a 5G ecosystem, which spans a large variety of infrastructure resources (stretching from radio access network, over edge and transport, to the core and data network). Relying on both SDN and NFV to enable full network programmability, such an ecosystem becomes a robust software-based scheme whose components and their interconnecting links are vir-

[2] In the 3GPP 5G architecture, the N6 reference point is connecting User Plane Function (UPF) in the 5G Core, with the Data Network, where the EdgeApps are running.

tualized and programmable. However, by virtualizing network resources that are distributed across various network segments (from radio to the core), a pool of resources becomes extremely large and heterogeneous, and proper management and orchestration of those resources are inevitable. In this Section, we focus on a subset of the overall end-to-end resources, i.e., on orchestrating MEC resources. If we take a look at Figure 1 again, MEC resources are placed at both terrestrial and aerial MEC systems. Thus, a proper MEC orchestration is crucial for ensuring optimal resource usage and optimal operation of services deployed on the MEC nodes. Therefore, this Section elaborates on the MEC orchestration principles, detailing further the specifics of each MEC orchestration operation.

3.1 NFV orchestration principles

The 5G ecosystem is binding together the fifth generation of the cellular mobile communication system (5G) with the distributed virtualized network infrastructure in the edges and clouds to deliver services to the end users with low latency and high bandwidth. In the case of UAV-based MEC systems illustrated in Figure 1, these end users are ground users that can be mobile (vehicles, barges, VRUs, etc.) and static (road infrastructure and deployed sensors). Such users do not only represent mobile devices but also the entire vertical industries that have diverse requirements in terms of resources and service performance.

The NFV has been already mentioned as one of the main 5G and MEC technology enablers, which enables the separation of control and data planes. This separation between the two planes is allowing 5G systems to perform life-cycle management of services in an automated and agile manner. As MEC systems are being exploited throughout 5G deployments in order to deliver localized access to EdgeApps, i.e., MEC services by deploying them in close proximity to the users, it is important to closely monitor the performance of those MEC systems and to derive corrective measures that will ensure required service performance.

Thus, regardless of the low latency accommodation for MEC services and EdgeApps enabled by deploying services close to the mobile ground users, MEC deployments still impose challenges in terms of efficient orchestration of services in a resource-constrained and highly distributed environment. Especially given their mobility and distributed locations, UAV-based MEC systems need to be properly managed since their malperformance can have a serious impact on the end-to-end service latency and service reliability.

Therefore, one of the biggest challenges is to design and create an overarching framework for the automated deployment and orchestration of edge service with requested levels of QoS and Quality of Experience (QoE). Such a framework enables full control of the network between services deployed at the mobile edges, i.e., UAV-based MEC nodes, and mobile devices, which connect to these services through the cellular 5G network. In such a network setting, orchestration entities can collect and process data locally where it is generated and needed, while reducing the communication path as well as latency. However, in such highly mobile environments where both users and MEC nodes are moving, real-time monitoring and seamless reconfiguration are required for maintaining service continuity. Besides efficient mechanisms for service reconfiguration, relocation of the connection to a MEC node closer to the ground user also needs to be provided. On the contrary, reactive approaches for service continuity are reconfiguring service chains or EdgeApp only after an event happens (e.g., VRU experiencing delayed notifications about speeding cars in near proximity, or vehicle moving to a location which is out of service for the MEC host [43]). Such approaches are more and more complemented or even replaced by proactive solutions, which leverage data analytics and AI/ML to anticipate such events and prepare MEC resources in advance.

The responsibility of the NFV management and orchestration processes is to coordinate and manage the deployment of network services and/or EdgeApps [14, 8]. In Section 1, we already introduced the NFV orchestration framework proposed by ETSI, which stretches over the following domains: i) VNFs, as pieces of MEC service or EdgeApp deployment, ii) NFVI, which combines hardware/software resources for deploying VNFs, i.e., MEC services or EdgeApps, and iii) orchestration entities such as NFVO and MEC Application Orchestrator (MEAO) that are responsible for organizing and managing NFVI, thereby performing the life-cycle management of VNFs [15, 16].

In the comprehensive survey on orchestration, de Sousa et al. [14] provided different types of classification of the orchestrators. First, the NFV orchestration elements can be classified based on the type of responsibility:

- *service orchestrator* that is responsible for composing and decomposing services and EdgeApps, by selecting the VNFs that need to be chained to deliver the service,
- *life-cycle orchestrator* that is in charge of managing the workflows, processes, and dependencies across VNFs that are selected and chained by the service orchestrator, and
- *resource orchestrator* that is making sure that every designed chain of VNFs is translated to virtual and physical resources that it requires for its smooth operation.

Second, in the following way, orchestrators can be classified based on their operational scope:

- *domain orchestrators* are entirely responsible only for those resources and services that belong to their domains (e.g., edge domain),
- *multi-domain orchestrators* stretch over multiple edge domains, and as such, they have a broader scope of action, which makes them more complex, but yet more powerful as they are enabling end-to-end service orchestration while spanning different administrative domains [14, 38].

It is important to note that this classification does not necessarily mean that one NFV orchestrator can embody only one of the designated roles; thus, it can be responsible for various roles, thereby orchestrating services and resources, and making decisions on their allocation and efficient placement. However, the second classification depends more on the type of the use case, and whether the cross-domain operation is required or not.

In Section 3.2, we will detail more on particular orchestration operations, but the role of the orchestrators, in general, is: i) to identify the resource needs for MEC deployment of an EdgeApp, and ii) to configure service on top of the selected resources in an efficient and dynamic manner, thereby performing proactive service reconfigurations (e.g., scaling, migration, and service teardown, which will be further detailed as orchestration operations). Thus, service and resource orchestration is performed on top of the *collaborative and distributed MEC environments* to enable *openness* and *programmability* of 5G and beyond ecosystems. The word **'collaborative'** refers to the collaboration between different edge/MEC orchestrators in performing orchestration of services and EdgeApps deployed at the 5G edges, which belong to different administrative (e.g., two mobile operators) and/or different technological domains (e.g., the same operator, but different technologies used, such as OpenStack and Kubernetes). Since 5G is mainly designed to boost the operation of verticals (automotive, transport & logistics, e-health, etc.), vertical services that are built to deliver new use cases for those verticals can be deployed at the network edge, in order to experience lower latency and higher bandwidth. As UAV-based edges can be spawned anywhere in the network infrastructure, depending on the location of their mobile users, vertical services and their constituting EdgeApps can be deployed in a distributed manner, stretching over multiple edge platforms or being migrated from one to another depending on the service performance.

Thus, such distributed infrastructure resources and services/EdgeApps need to be orchestrated by **distributed orchestration elements**, i.e., edge orchestrators that are collaborating with each other while being distributed across the 5G edges. Leveraging on NFV and SDN to achieve network programmability, **edge networks** are enabling **virtualized** and **programmable** service chains, i.e., vertical services and EdgeApps that are loosely coupled via **open** interfaces, which can be efficiently reconfigured based on the decisions made by orchestrators.

Therefore, to manage the distributed EdgeApp deployments across multiple MEC sites, two prerequisites need to be met: i) coordination between the adjacent MEC orchestrators is required, especially if MEC service or EdgeApp deployments stretch over MEC sites that belong to different administrative domains (e.g., countries), and ii) strict isolation between EdgeApp instances needs to be ensured without affecting QoS, as these instances may belong to different verticals or different ground users. The aforementioned challenges can be mitigated by enabling collaboration between orchestrated edges via the distribution of orchestration tasks, which provides proactive multi-domain service deployments with support for service continuity. Such collaboration is illustrated in Figure 6, showing the workflows of operations for EdgeApp deployment on selected UAV-based MEC hosts, and cross-domain collaboration between orchestrators and EdgeApps.

3.2 Orchestration operations

This section provides insights into a baseline group of management and orchestration operations that consists of the service/application instantiation/placement, scaling, migration/relocation, and termination. These operations can be grouped into *deployment* and *runtime* operations. The first group reflects on the operations of service onboarding and instantiation (e.g., preparation of service descriptors and images that will be used for deployment). The second group comprises the operations that are executed during the service runtime (users consuming MEC service). At the moment when the MEC application is required, either to provide service to the ground users or to offload and execute users' tasks, orchestrators proceed first with the deployment procedure, which is then followed by runtime operations while the application is being consumed by the ground users.

Service instantiation

As MEC applications or EdgeApps are realized as chains of loosely coupled VNFs, service instantiation is based on the VNF placement procedure that is realized through the two following phases [8]: i) the design and creation of Service Function Chain (SFC) that combine several VNFs that are inevitable for MEC service performance, and ii) the SFC embedding on the substrate network that consists of the physical MEC resources (CPU, memory, compute). These two phases are successive; however, they cannot be isolated from each other as the overall process of VNF instantiation needs to be coordinated by NFVO and MEAO (as briefly described in Section 1.1. As elaborated in our previous work [8], the process of embedding the designated SFC on the substrate infrastructure resources is performed as follows: i) the traffic paths are determined for each VNF in the chain, ii) bandwidth is allocated on the links that constitute the determined traffic paths, iii) VMs or Docker containers are spawned at different nodes on the determined paths, and iv) VNFs are installed on the instantiated VMs or Docker containers.

Although more resourceful than computing units on the user side, the resources at the MEC systems are limited as well, and it is important to use them efficiently. Thus, to maximize resource utilization, resource management needs to determine the number of resources that will be enough to obtain a satisfactory level of service quality. To obtain more efficient resource utilization, sharing of VNFs between various SFCs can be exploited, thereby taking care of the VNF functionalities and specific limitations which are mostly defined for security reasons. The aforementioned concerns utilization of computing resources, however, VNF placement procedure should also take into account network conditions and the impact of the network on the MEC node selected for placement. As the network has a significant impact on the service performance, usually measured in terms of end-to-end latency, and uplink/downlink throughout, Cziva et al. [37] studied VNF placement at the distributed edges as a problem of allocating VNFs that is supposed to minimize end-to-end latency from all users to their associated VNFs. In their work [37], Cziva et al. used optimal stopping theory to resolve the problem of dynamic re-scheduling and the optimal placement of VNFs based on temporal latency fluctuations.

Runtime operations

The runtime orchestration operations such as VNF *migration*, *scaling*, and *termination*, are performed by NFVO and MEAO orchestrators during the MEC service runtime.

In particular, if more resources are needed for service operation, **scaling** procedure is invoked, and as such it is in charge of assigning more resources to the running VNF. This procedure refers to scaling up and out, which refers to assigning more resources to the existing VM or container, or to spawning more VMs or containers towards supporting the MEC service, respectively. Similarly, if MEC service does not require the allocated amount of resources, procedures of *scaling down* and *in* are invoked. The first is reducing the number of resources assigned to the VM or container, while the latter terminates the unnecessary VMs or containers. The decisions on whether to perform scaling operations are made by the orchestrators, which take into account real-time monitored performance metrics such as available computing resources, network performance (end-to-end latency, uplink/downlink throughput), user locations, and energy utilization, among others. The same applies to the other runtime operation, i.e., service **termination**, which is in charge of releasing the resources allocated for the performance of MEC service, thereby deleting the deployed VM or container.

Concerning service **migration** or **relocation**, this orchestration operation is usually considered a scale-out operation that stretches over multiple MEC hosts. Service migration/relocation sometimes requires synchronized work between orchestrators that are managing different MEC hosts, thereby demanding a controlled environment with time-sensitive operations of proactive service deployment on the target MEC system in order to minimize the service downtime. Given such complexity, this operation requires more discussion.

In the context of multiple edge domains, service relocation implies relocation of all VNF chains, from source MEC servers in one domain to selected target MEC servers in another. In general, the service migration should be invoked anytime Service Level Agreement (SLA) for the network service is not ensured.

This type of orchestration operation needs to be performed efficiently and in real-time, as it can involve potential disruptions in service continuity, which results in undesirable effects on the service performance, thereby affecting QoS and QoE. By monitoring service performance in real-time, changes in various parameters can trigger service relocation. In particular, migrating MEC applications and their binding VNFs is necessary when application performance requirements are not met anymore [17]. For instance, the service relocation becomes an essential process for maintaining service continuity for mobile users on the ground, as different UAV-based MEC nodes can be selected for serving ground users based on the MEC resource availability, UAV location, as well as its energy efficiency. In general, service migration results in moving an ongoing service or application from one UAV host to another, while application state relocation refers to copying the state of the service from the source to the target UAV. In Figure 6, we present an overview of operations that enable smooth EdgeApp relocation, thereby performing cross-domain collaboration between orchestrators and EdgeApps.

In their study on service migration, Wang et al. [18] state that the decision on migrating service should be triggered based on the overhead that this migration brings, as well as the QoS requirements. Given the synergy of 5G and MEC promises an ultra-low latency (i.e., 1 ms-10 ms) and high throughput (i.e., above 100 Mbps), service management and orchestration systems are pressured to maintain service continuity by following the user mobility, and placing network services and EdgeApps always at the most suitable MEC platforms [19, 20].

Although both processes are initiated by the user's movement from one area to another, the handover and service migration should be differentiated and treated differently. In their latest survey, Wang et al. [18] also highlight the differences between these two processes, which are summarized as follows:

- *Amount of data to be transferred*: the data to be exchanged between the source and target hosts during handover usually contains only the signaling messages between UE and gNB, or two gNBs that handle the handover process, while in the case of service or state migration the memory data and/or application data image messages should be transferred, burdening the system with traffic that is usually a way larger than signaling,
- *Technology diversity*: in cellular networks, the handover is always performed between two neighboring cells with the same technology, while service migration is independent of the technology, and usually occurs in a heterogeneous environment with different network topologies and technologies in edge domains,
- *Triggering the process*: while the handover is required anytime UE exits the coverage area of a particular gNB, in case of service migration, UEs can still exchange data with the remote edge server, thereby bringing additional complexity in the whole system.

Furthermore, under the umbrella of service migration, there are a few practical concepts that are widely studied and adopted in industry and research, i.e., i) VM migration, ii) container migration, and iii) stateful process migration [21]. As an application is usually realized in the form of a set of execution environments (e.g., operating system) and services that are required for an application to run, the aforementioned concepts differ in the components of the overall application that are migrated [21]. Hence, in the case of VM migration, all application components need to be migrated from the source to the target host, which due to the amount of data takes more time.

In addition, there are different types of VM migration, such as cold migration, pre-copy live migration, and post-copy live migration, which are elaborated by Abdah et al. in [20]. On the other side, in the case of container migration, the execution environment is not migrated but only the service (i.e., stateless and stateful). Finally, in the case of stateful process migration, only the stateful processes in the application

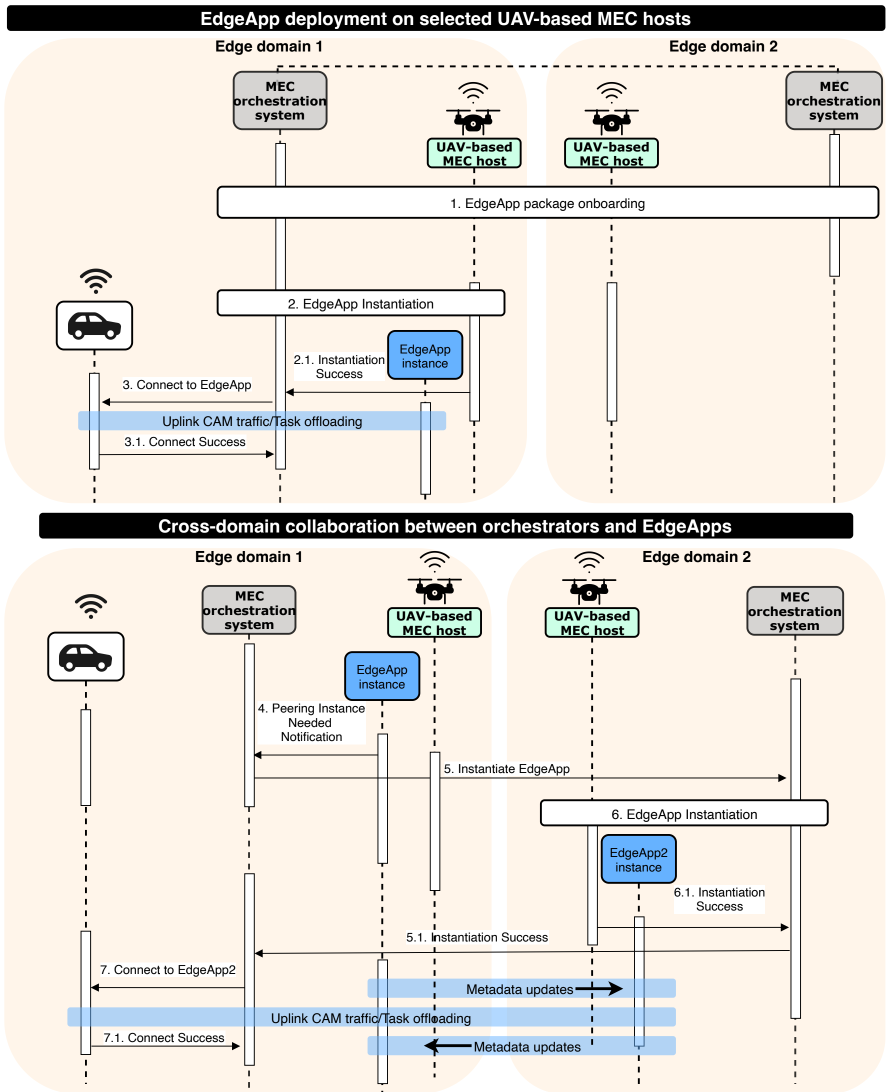

Fig. 6 Workflow of operations for EdgeApp deployment on selected UAV-based MEC hosts, and Cross-domain collaboration between orchestrators and EdgeApps.

are migrated from one host to another. As network edge is usually characterized by constraints in both network and computing resources, service migration also needs to be network and resource-aware. Therefore, Horri et al. [21] studied the concept of separating stateless and stateful processes inside an application, allowing them to talk to each other via inter-process communication channels that, once stateful processes are migrated, need to be re-established on the destination server.

Furthermore, according to Addad et al. [19], this migration of stateful processes can be obtained in two manners: i) stateful service migration with a predefined path, in case the system can anticipate the source and the target MEC nodes for any migration along the way of the user, and ii) stateful service migration based on undefined path, which is a more generic approach since service providers usually do not know the movement patterns of their users [19]. While in the first case, the need for service migration can be anticipated and thus preemptively triggered and performed, in the second case it becomes impossible and Addad et al. [19] study and present the fast and efficient migration process with a shared file system/pool that stores the container's file system. Finally, concerning the migration costs, Strunk provides an overview of all contributing factors to the overall service migration cost in [22]. The costs that vastly influence the service performance are the total migration time, the downtime, the energy overhead, but also the impact on the performance of VMs after migration, such as execution time and throughput of processes running inside a VM during migration [22].

The total migration time is studied and evaluated in different migration approaches [22, 23, 19, 21, 20] and it highly depends on the total amount of memory that has to be transmitted from source to target hypervisor/host and average link speed between these hosts [23], but also on the CPU resources of the source host due to the increase of processing cycles caused by migration.

3.3 Smart EdgeApps

In [39], we introduced the concept of *smart* EdgeApps as a particular design feature leveraged for boosting orchestration operations. This concept allows orchestrators to improve their decision-making process by leveraging notifications that are created by orchestrated EdgeApps themselves. This way, orchestrators can retrieve some application-specific insights, such as the exact locations of the ground users, change in route of connected vehicles, proximity to the other users, detection of obstacles on the road, and network re-selection. These insights are usually not known by orchestrators [38], and as such, they can be useful for tailoring the orchestration decisions to the particularity of service performance.

To be smart in the 5G context, MEC application needs to be aware of:

- the *edge environment*, such as the MEC orchestration elements, e.g., MEAO and NFVO,
- the *other MEC applications or EdgeApps*, which are involved in the same operation (e.g., emergency situations on the roads), as well as
- the *clients running on the users' side*, which connect to those EdgeApps and either use their service or offload tasks.

Such awareness allows an EdgeApps to increase its situational awareness and to be actively involved in making decisions about its own operation. With such a concept, EdgeApps are capable to generate important warnings and notifications for the orchestrators, thereby improving and boosting orchestrators' decisions about the life-cycle management of EdgeApps. The notifications that EdgeApps produce can be derived using either data analytics and/or different ML models that are executed by applications.

Studying the above-listed features that make an EdgeApp smart, the following implications can be derived. An EdgeApps is capable of:

- retrieving the topological and service coverage of the orchestrators (e.g., coverage of one edge domain), which is a relevant input for determining the boundaries of the service regions covered by MEAO and NFVO, thereby used by smart edge applications to timely trigger their relocation e.g., if the user is approaching the border between two edge domains, and
- passing the notifications to the orchestration entities, which these entities can further use to optimize their orchestration decisions that trigger operations such as EdgeApp instantiation, scaling, migration, or termination, and thus, maintain QoS at a required level.

The awareness of the other EdgeApps that are involved in the same operation enables extending the application service operation beyond the boundaries of one edge domain. As illustrated in Figure 6, EdgeApps can use service-based interfaces to exchange application metadata with other instances of EdgeApps running in the other domains. Some examples of this metadata are the location/speed/heading of mobile users connected to them. Concerning the network characteristics, an EdgeApps can collect statistics and relevant network data from MEC Value-added Service (VAS) such as Radio Network Information Service (RNIS) (per radio cell, or UE). Such connectivity to MEC VASs helps EdgeApps to retrieve network connectivity information about a particular user that is about to move out of the domain of one UAV and to re-select the network. Given such information, a smart EdgeApp can apply a suitable ML model to preempt network re-selection, which would have broken the service connectivity. Such a decision is further used for proactive triggering EdgeApp instantiation/migration in the target domain.

4 Open questions

In this chapter, we have tackled various aspects of MEC systems, putting the focus on different types of UAV-based MEC architectures, applications, and the necessity of efficient MEC orchestration. However, there are numerous challenges in network settings that combine MEC and IoUT. Based on the literature search, we present several of those challenges that require prominent attention in research and industry circles before building MEC systems that are entirely based on UAVs.

- **Resource allocation and cross-MEC collaboration**: According to study conducted by Zhou et al. [11] and Tun et al. [45], most of the proposed aerial MEC schemes are taking into account only one UAV and one user. This setting is not realistic, given that all operations we studied in Section 2.1 need to include more

UAVs to cover larger areas, and certainly more than one user. One of the reasons for creating multi-UAV MEC systems is a limited set of resources and limited battery of each UAV, where tasks need to be shared by different MEC systems towards producing a single outcome for a ground user. Also, given the mobility of ground users, it becomes imperative to lay the foundation for multi-aerial MEC systems that collaborate with each other towards maximizing users' satisfaction with the service/EdgeApp. The distribution of users to different MEC systems is a complex non-convex optimization problem [11, 44], and as such, it needs to be further studied and tested in realistic environments. The whole collaboration among MEC system can be complex and needs to be carefully designed, defining the agreements among MEC orchestrators.

- **Computation efficiency** Although UAV-based MEC nodes are more resourceful than computing engines placed within UEs, they are not as resourceful as cloud computing systems. Thus, it is important to closely monitor how intensive the tasks offloaded to MEC systems are, and how much load is imposed on those systems when running EdgeApps that serve the ground users. As the main motivator for running EdgeApps on the network edge is to improve QoS, in particular end-to-end latency and throughput, these QoS parameters need to be carefully monitored while MEC systems perform any computation, as corrective decisions need to be proactively made (e.g., to relocate task and EdgeApps from one UAV to another) to prevent QoS deterioration.
- **Signal interference management** Despite reliable wireless connectivity of UAVs due to their high altitude and limited scattering [12], there is a strong air-ground interference that is particularly large in case of increased density of UAVs. According to Song et al. [12], this interference is one of the most critical issues in case UAVs are acting as users, as they need to offload their tasks to terrestrial MEC systems, thereby connecting to base stations and thus causing interference to other base stations that are occupying the same spectrum. One of the solutions they propose is to exploit directional instead of omnidirectional antennas, where useful LoS signals can be amplified.
- **Application of AI/ML** Applying AI/ML mechanisms on both MEC orchestration and EdgeApp operations is promising when it comes to optimizing the resource consumption and advancing decision-making processes based on large amounts of collected data. Some of the techniques such as federated learning and multiple agent reinforcement learning seem suitable for aerial MEC systems that consist of various UAVs. However, applying these techniques requires a significant amount of computing resources, which might not be present at the network edge. Thus, more study is needed on the applicability of some of the prominent AI/ML algorithms on aerial MEC systems.
- **Energy optimization** Given their extremely short battery life (30 minutes of flying [12]), the energy consumption of UAVs needs to be optimized, thus, distributing the load across multiple MEC nodes and offloading tasks to terrestrial MEC systems as well [46, 48]. Some of the promising techniques for extending the battery lifetime are wireless charging, and laser-powered UAVs, but their investigation is currently limited only to theoretical and simulation concepts [12].

- **Security** Even though MEC systems impose fewer security vulnerabilities than cloud systems due to their close proximity to the end users, there is still a large amount of sensitive data that needs to be protected. Some of the major security concerns when it comes to UAV-based MEC systems are passive eavesdropping, active interfering, information leakage or manipulations, and denial of service attacks [12]. In their survey, Song et al. [12] study the criticality of those security issues, especially because the detection of malicious attempts becomes even more difficult in dynamic network topologies such as those where MEC nodes can be spawned at any location. However, most of the studies on security are rather general, and more attention needs to be given to the particularity of attacks in aerial MEC systems.

5 Conclusion

The 5G and beyond ecosystems are comprised of the cellular 5G system along with a properly managed and orchestrated deployment of virtualized instances of EdgeApps. One of the major contributors to decreased end-to-end latency and higher throughput in 5G systems is the involvement of MEC systems. Being deployed close to the end users, MEC paradigm became a promising solution for on-demand service deployments and task offloading from the resource-constrained computing engines at the UE side. Thus, in this Chapter, we discussed the MEC technology in general, focusing more on its application to IoUT systems where dynamically spawned UAV-based MEC nodes are used as flying/mobile and highly flexible NFV infrastructure suitable for deploying MEC services at optimal locations for the ground users.

A heterogeneous ecosystem with 5G leveraging both terrestrial and aerial MEC nodes enables customized deployment and operation of services for different sectors of the vertical industries. In Section 1.1 we introduced a baseline framework for MEC systems, which serves as a foundation for all MEC-based deployments. In Section 1.2, we discussed different architectural styles where UAVs are exploited in MEC settings, acting as a user, a relay node, and MEC node itself. Furthermore, the entire Section 2 is dedicated to various types of application scenarios in which UAV-based or aerial MEC system may play a significant role in future network deployments, focusing on emergency situations in both urban and rural environments [47], optimizing navigation in traffic jams, providing disaster management support and navigation support in distant areas without terrestrial MEC systems.

In Section 3, we described a solution for the orchestration of MEC services or EdgeApps within such a 5G ecosystem to meet the stringent requirements of moving users on the ground, which connect to services in the network infrastructure. A key objective of such EdgeApp orchestration solution is the availability and continuity of low-latency services at the network infrastructure edges for highly dynamic UAV-based MEC scenarios and the associated management and orchestration of these services in distributed edge clouds. Finally, we close this chapter by making an overview of open challenges in aerial MEC settings.

References

1. E. Coronado, Z. Yousaf, and R. Riggio, “LightEdge: Mapping the Evolution of Multi-Access Edge Computing in Cellular Networks,” *IEEE Communications Magazine*, vol. 58, no. 4, pp. 24–30, 2020, doi: http://dx.doi.org/10.1109/MCOM.001.1900690.
2. 5G-ACIA, “5G for Connected Industries and Automation (White Paper - Second Edition),” no. November, p. 28, 2018, [Online] Available: https://bit.ly/2BGSMLA.
3. J. Van De Belt, H. Ahmadi, and L. E. Doyle, “Defining and Surveying Wireless Link Virtualization and Wireless Network Virtualization,” *IEEE Communications Surveys and Tutorials*, vol. 19, no. 3, pp. 1603–1627, 2017, doi: http://dx.doi.org/10.1109/COMST.2017.2704899.
4. K. Samdanis, X. Costa-Perez, and V. Sciancalepore, “From network sharing to multi-tenancy: The 5G network slice broker,” pp. 32–39, 2016, doi: http://dx.doi.org/10.1109/MCOM.2016.7514161.
5. D. Zhang, *Virtual Resource-Sharing Mechanisms in Software-Defined and Virtualized Wireless Network*, 2018, [Online] Available: https://bit.ly/2UJXa6K.
6. F. Giust, V. Sciancalepore, D. Sabella, M. C. Filippou, S. Mangiante, W. Featherstone, and D. Munaretto, “Multi-Access Edge Computing: The Driver Behind the Wheel of 5G-Connected Cars,” *IEEE Communications Standards Magazine*, vol. 2, no. 3, pp. 66–73, 2018, doi: http://dx.doi.org/10.1109/MCOMSTD.2018.1800013.
7. A. Reznik. Multi-access Edge Computing (MEC). [Online] Available: https://www.etsi.org/technologies/multi-access-edge-computing?jjj=1573496932482.
8. N. Slamnik-Kriještorac, E. de Britto e Silva, E. Municio, H. Carvalho de Resende, S. Hadiwardoyo, and J. Marquez-Barja, “Network Service and Resource Orchestration: A Feature and Performance Analysis within the MEC-Enhanced Vehicular Network Context,” *Sensors 2020*, vol. 20, 2020, doi: https://doi.org/10.3390/s20143852.
9. A. Reznik. Open Source MANO. [Online] Available: https://www.etsi.org/technologies/nfv/open-source-mano.
10. ETSI, “Mobile Edge Computing; Market Acceleration; MEC Metrics Best Practice and Guidelines,” *ETSI GS MEC-IEG 006 V1.1.1*, 2017.
11. F. Zhou, R. Q. Hu, Z. Li and Y. Wang, ”Mobile Edge Computing in Unmanned Aerial Vehicle Networks,” *IEEE Wireless Communications*, vol. 27, no. 1, pp. 140-146, February 2020, doi: 10.1109/MWC.001.1800594.
12. Z. Song, X. Qin, Y. Hao, T. Hou, J. Wang, X. Sun, , ”A comprehensive survey on aerial mobile edge computing: Challenges, state-of-the-art, and future directions,” *Computer Communications*, vol. 191, pp. 233-256, 2022, doi: 10.1016/j.comcom.2022.05.004.
13. N. Slamnik-Kriještorac, G. M. Yilma, F. Zarrar Yousaf, M. Liebsch, and J. M. Marquez-Barja, “Multi-domain MEC orchestration platform for enhanced Back Situation Awareness,” in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2021, pp. 1–2, doi: https://dx.doi.org/10.1109/INFOCOMWKSHPS51825.2021.9484632.
14. N. F. Saraiva de Sousa, D. A. Lachos Perez, R. V. Rosa, M. A. Santos, and C. Esteve Rothenberg, “Network Service Orchestration: A Survey,” *Computer Communications*, vol. 142-143, p. 69–94, Jun 2019, doi: http://dx.doi.org/10.1016/j.comcom.2019.04.008. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2019.04.008
15. T. Taleb, S. Member, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, “On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration,” vol. 19, no. 3, pp. 1657–1681, 2017, doi: http://dx.doi.org/10.1109/COMST.2017.2705720.
16. B. Blanco, J. Oscar, I. Giannoulakis, E. Kafetzakis, S. Peng, J. Pérez-romero, I. Trajkovska, P. S. Khodashenas, L. Goratti, M. Paolino, E. Sfakianakis, F. Liberal, and G. Xilouris, “Computer Standards & Interfaces Technology pillars in the architecture of future 5G mobile networks : NFV , MEC and SDN,” *Computer Standards & Interfaces*, vol. 54, no. January, pp. 216–228, 2017, doi: http://dx.doi.org/10.1016/j.csi.2016.12.007. [Online]. Available: http://dx.doi.org/10.1016/j.csi.2016.12.007

17. Z. Tang, X. Zhou, F. Zhang, W. Jia, and W. Zhao, “Migration modeling and learning algorithms for containers in fog computing,” *IEEE Transactions on Services Computing*, vol. 12, no. 5, pp. 712–725, 2019, doi: https://doi.org/10.1109/TSC.2018.2827070.
18. S. Wang, J. Xu, N. Zhang, and Y. Liu, “A Survey on Service Migration in Mobile Edge Computing,” *IEEE Access*, vol. 6, pp. 23 511–23 528, 2018, doi: http://dx.doi.org/10.1109/ACCESS.2018.2828102.
19. R. A. Addad, D. L. Cadette Dutra, M. Bagaa, T. Taleb, and H. Flinck, “Towards a Fast Service Migration in 5G,” in *2018 IEEE Conference on Standards for Communications and Networking (CSCN)*, 2018, pp. 1–6, doi: http://dx.doi.org/10.1109/CSCN.2018.8581836.
20. H. Abdah, J. P. Barraca, and R. L. Aguiar, “QoS-Aware Service Continuity in the Virtualized Edge,” *IEEE Access*, vol. 7, pp. 51 570–51 588, 2019, doi: http://dx.doi.org/10.1109/ACCESS.2019.2907457.
21. M. Horii, Y. Kojima, and K. Fukuda, “Stateful process migration for edge computing applications,” in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, 2018, pp. 1–6, doi: http://dx.doi.org/10.1109/WCNC.2018.8377072.
22. A. Strunk, “Costs of virtual machine live migration: A survey,” in *2012 IEEE Eighth World Congress on Services*, 2012, pp. 323–329, doi: http://dx.doi.org/10.1109/SERVICES.2012.23.
23. H. Maziku and S. Shetty, “Towards a network aware vm migration: Evaluating the cost of vm migration in cloud data centers,” in *2014 IEEE 3rd International Conference on Cloud Networking (CloudNet)*, 2014, pp. 114–119, doi: http://dx.doi.org/10.1109/CloudNet.2014.6968978.
24. L. Yang and H. Yao, and X. Zhang, and J. Wang, and Y. Liu “Multi-UAV Deployment for MEC Enhanced IoT Networks,” in *IEEE/CIC International Conference on Communications in China (ICCC)*, 2020, pp. 436-441, doi: 10.1109/ICCC49849.2020.9238870.
25. L. Gupta and R. Jain, and G. Vaszkun “Survey of Important Issues in UAV Communication Networks,” in *Commun. Surveys Tuts.*, 2016, pp. 1123–1152., doi: https://doi.org/10.1109/COMST.2015.2495297
26. H. Yao, L. Wang, X. Wang, Z. Lu and Y. Liu,“The Space-Terrestrial Integrated Network: An Overview,” in *IEEE Communications Magazine*, vol. 56, no. 9, pp. 178-185, Sept. 2018, doi: 10.1109/MCOM.2018.1700038.
27. J. Zhang et al.,“Aeronautical *Ad Hoc* Networking for the Internet-Above-the-Clouds,” in *Proceedings of the IEEE*, vol. 107, no. 5, pp. 868-911, May 2019, doi: 10.1109/JPROC.2019.2909694.
28. X. Li, H. Yao, J. Wang, S. Wu, C. Jiang and Y. Qian, “Rechargeable Multi-UAV Aided Seamless Coverage for QoS-Guaranteed IoT Networks,” in *IEEE Internet of Things Journal,*, vol. 6, no. 6, pp. 10902-10914, Dec. 2019, doi: 10.1109/JIOT.2019.2943147.
29. J. Wang, C. Jiang, Z. Han, Y. Ren, R. G. Maunder and L. Hanzo, “Taking Drones to the Next Level: Cooperative Distributed Unmanned-Aerial-Vehicular Networks for Small and Mini Drones,” in *IEEE Vehicular Technology Magazine*, vol. 6, no. 6, pp. 10902-10914, Dec. 2019, doi: 10.1109/JIOT.2019.2943147. vol. 12, no. 3, pp. 73-82, Sept. 2017, doi: 10.1109/MVT.2016.2645481.
30. N. Toon, “5G Requirements and Key Performance Indicators,” *Journal of ICT Standardization*, 2018, 6, doi: https://doi.org/10.13052/jicts2245-800X.612
31. J. M. Marquez-Barja, S. Hadiwardoyo, B. Lannoo, W. Vandenberghe, E. Kenis, L. Deckers, M. C. Campodonico, K. dos Santos, R. Kusumakar, M. Klepper, and J. Vandenbossche, “Enhanced Teleoperated Transport and Logistics: A 5G Cross-Border Use Case,” in *2021 Joint European Conference on Networks and Communications 6G Summit (EuCNC/6G Summit)*, 2021, pp. 229–234, doi: https://doi.org/10.1109/EuCNC/6GSummit51104.2021.9482459.
32. J. Marquez-Barja, D. Naudts, V. Maglogiannis, S. A. Hadiwardoyo, I. Moerman, M. Klepper, G. Kakes, L. Xiangyu, W. Vandenberghe, R. Kusumakar, and J. Vandenbossche, “Designing a 5G architecture to overcome the challenges of the teleoperated transport and logistics,” *IEEE 19th Annual Consumer Communications & Networking Conference (CCNC). pp 1-4. January, 2022. Las Vegas, United States of America.*, 2022, Online [Available]: https://www.marquez-barja.com/en/publications.

33. N. Slamnik-Kriještorac et al., “Network Applications (NetApps) as a 5G booster for Transport & Logistics (T&L) Services: The VITAL-5G approach,” *2022 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, 2022, pp. 279-284, doi: 10.1109/EuCNC/6GSummit54941.2022.9815830.
34. ETSI, “VNF Package and PNFD Archive specification,” *ETSI ISG NFV, ETSI GS NFV-SOL 004 v3.5.1*, 2021, Online [Available]: https://www.etsi.org/deliver/etsi_gs/NFV-SOL/001_099/004/03.05.01_60/gs_NFV-SOL004v030501p.pdf
35. 3GPP, “Technical Specification Group Services and System Aspects; Procedures for the 5G System (5GS) Stage 2,” *3GPP TS 23.502 V16.6.0*, 2020, Online [Available]: https://www.3gpp.org/ftp//Specs/archive/23_series/23.502/
36. 3GPP, “Management and orchestration; 5G Network Resource Model (NRM); Stage 2 and Stage 3,” *3GPP TS 28.541 v17.5.0*, 2021, Online [Available]: shorturl.at/hjlp8
37. R. Cziva, C. Anagnostopoulos and D. P. Pezaros, “Dynamic, Latency-Optimal vNF Placement at the Network Edge,” *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, 2018, 693-701, doi: http://dx.doi.org/10.1109/INFOCOM.2018.8486021
38. N. Slamnik-Krijestorac, G. M. Yilma, M. Liebsch, F. Z. Yousaf, and J. Marquez-Barja, “Collaborative orchestration of multi-domain edges from a Connected, Cooperative and Automated Mobility (CCAM) perspective,” *IEEE Transactions on Mobile Computing*, pp. 1–1, 2021, doi: http://dx.doi.org/10.1109/TMC.2021.3118058.
39. G. M. Yilma, N. Slamnik-Krijestorac, M. Liebsch, A. Francescon, and J. Marquez-Barja, “No Limits – Smart Cellular Edges for Cross-Border Continuity of Automotive Services,” *2022 IEEE Future Networks World Forum (FNWF)*, 2022, Online [Available]: https://www.researchgate.net/publication/363134905_No_Limits_-_Smart_Cellular_Edges_for_Cross-Border_Continuity_of_Automotive_Services.
40. B. Liu, Y. Wan, F. Zhou, Q. Wu and R. Q. Hu, “Resource Allocation and Trajectory Design for MISO UAV-Assisted MEC Networks,” *IEEE Transactions on Vehicular Technology*, vol. 71, no. 5, pp. 4933-4948, May 2022, doi: 10.1109/TVT.2022.3140833.
41. D. Wang, J. Tian, H. Zhang and D. Wu, ”Task Offloading and Trajectory Scheduling for UAV-Enabled MEC Networks: An Optimal Transport Theory Perspective,” *IEEE Wireless Communications Letters*, vol. 11, no. 1, pp. 150-154, Jan. 2022, doi: 10.1109/LWC.2021.3122957.
42. Q. Luan, H. Cui, L. Zhang and R. E. Welsch, ”A Scheme for Flexible-Hybrid Subtask Offloading in a Two-Tier UAV- Assisted MEC Network,” *ICC 2022 - IEEE International Conference on Communications*, Seoul, Korea, Republic of, 2022, pp. 1-6, doi: 10.1109/ICC45855.2022.9838442.
43. A. Bera, P. J. Sanchez-Cuevas and M. A. Olivares-Mendez, ”MEC-assisted Dynamic Geofencing for 5G-enabled UAV,” *2022 IEEE Wireless Communications and Networking Conference (WCNC)*, Austin, TX, USA, 2022, pp. 160-165, doi: 10.1109/WCNC51071.2022.9771716.
44. Z. Yang, S. Bi and Y. -J. A. Zhang, ”Online Trajectory and Resource Optimization for Stochastic UAV-Enabled MEC Systems,” *IEEE Transactions on Wireless Communications*, vol. 21, no. 7, pp. 5629-5643, July 2022, doi: 10.1109/TWC.2022.3142365.
45. Y. K. Tun, T. N. Dang, K. Kim, M. Alsenwi, W. Saad and C. S. Hong, ”Collaboration in the Sky: A Distributed Framework for Task Offloading and Resource Allocation in Multi-Access Edge Computing,” *IEEE Internet of Things Journal*, vol. 9, no. 23, pp. 24221-24235, 1 Dec.1, 2022, doi: 10.1109/JIOT.2022.3189000.
46. Z. Xie, X. Song, J. Cao and W. Qiu, ”Providing Aerial MEC Service in Areas Without Infrastructure: A Tethered-UAV-Based Energy-Efficient Task Scheduling Framework,” *IEEE Internet of Things Journal*, vol. 9, no. 24, pp. 25223-25236, 15 Dec.15, 2022, doi: 10.1109/JIOT.2022.3195855.
47. Q. Luan, H. Cui, L. Zhang and Z. Lv, ”A Hierarchical Hybrid Subtask Scheduling Algorithm in UAV-Assisted MEC Emergency Network,” *IEEE Internet of Things Journal*, vol. 9, no. 14, pp. 12737-12753, 15 July15, 2022, doi: 10.1109/JIOT.2021.3138263.
48. G. Wu, Q. Liu, J. Xu, Y. Miao and M. Pustišek, ”Energy Efficient Task Caching and Offloading in UAV-Enabled Crowd Management,” *IEEE Sensors Journal*, vol. 22, no. 18, pp. 17565-17572, 15 Sept.15, 2022, doi: 10.1109/JSEN.2022.3182779.