"5G for Drone-based Vertical Applications"

D3.1 Report on infrastructure-level enablers for 5G!Drones

Document ID:	**D3.1**
Deliverable Title:	**Report on infrastructure-level enablers for 5G!Drones**
Responsible Beneficiary:	**OPL**

Topic:	**H2020-ICT-2018-2020/H2020-ICT-2018-3**
Project Title:	**Unmanned Aerial Vehicle Vertical Applications' Trials Leveraging Advanced 5G Facilities**
Project Number:	**857031**
Project Acronym:	**5G!Drones**
Project Start Date:	**June 1st, 2019**
Project Duration:	**42 Months**
Contractual Delivery Date:	**M18**
Actual Delivery Date:	**30.11.2020**
Dissemination Level:	**PU**
Contributing Beneficiaries:	**OPL, EUR, AU, CAF, DEM, DRR, FRQ, MOE, NOK, ORA, THA, UMS, UO**

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 857031.

Document ID: D3.1
Version: V1.0
Version Date: November 30th, 2020
Authors: Sławomir Kukliński (OPL) – leading editor, Lechosław Tomaszewski (OPL), Robert Kołakowski (OPL), Adlen Ksentini (EUR), Karim Boutiba (EUR), Serge Delmas (AIR), Tarik Taleb (AU), Oussama Bekkouche (AU), Hamed Hellaoui (AU), Tanel Järvet (CAF), Harilaos Koumaras (DEM), George Makropoulos (DEM), Stavros Kolometsos (DEM), Anastasios Gogos (DEM), George Xilouris (DEM), Piotr Dybiec (DRR), Paweł Korzec (DRR), Gregor Mogeritsch (FRQ), Ludwig Kastner (FRQ), Saadan Ansari (FRQ), Iris Roehrich (FRQ), Dimitrios Tzempelikos (MOE), Johannes Jyrkkä (NOK), Ilkka Kansala (NOK), Ivon Gourhant (ORA), Laurent Reynaud (ORA), Sabbir Ahmed (UO), Antti Pauanne (UO), Vesa Halonen (UO), Hafiz Abdul Hannan (UO), Jussi Haapola (UO), Jari Jääskelä (UO), Juha Röning (UO), Farid Benbadis (THA), Cyril Dangerville (THA), Pascal Bisson (THA), Alan Branchereau (THA), Tomas Gareau (UMS), Nemish Mehta (UMS)

Security: Public

Approvals

	Name	Organization	Date
Coordinator	Jussi Haapola	UO	30.11.2020
Technical Committee	Pascal Bisson	THA	27.11.2020
Management Committee	Project Management Team	FRQ, AU, THA, UMS, AIR, UO	30.11.2020

EXECUTIVE SUMMARY

The 5G!Drones is an innovative 42-month project focused on trials of several UAV use cases that cover eMBB, URLLC and mMTC 5G services, validation of 5G KPIs for supporting such challenging use cases, and their enhancements with powerful features. This deliverable is a report on the design and implementation of enabling mechanisms for 5G!Drones at the (5G) infrastructure level including:

- end to end network slicing;
- incorporation of MEC to facilitate UAV services;
- network and compute resources abstraction;
- facility federation.

The 5G!Drones project aims to explore, which of the 5G components need to be improved to support different UAV use cases. The large part of work is focused on how to build and secure network slices required for the realisation of the specific trials.

Network slicing enables the creation of parallel virtual telecommunication networks over a common distributed cloud infrastructure. Instead of handling all traffic fractions with completely different characteristics through one general-purpose network, these separate traffic fractions will be transmitted through parallel, separate but federated networks, architecturally and functionally adapted to requirements of their fractions. The main advantages towards this approach are the ability to create isolated networking solutions on-demand that are combined or tailored for specific applications and can be managed in a flexible manner, as well as flexible utilisation of underlying resources in an adapted way (e.g. through dynamic reallocation of resources to specific network slices, following the traffic demand). Efficient slices management can be achieved by monitoring a particular combination of network features aggregated in the form of KPIs. An important task to be realised by 5G!Drones project is also the definition and validation of the set of representative KPIs that will enable UAV verticals to monitor and manage Network Slices running UAV applications.

MEC provides cloud-computing capabilities and an IT service environment at the edge of the mobile network. The expected main advantages of the solution are the ability to achieve ultra-low latency, high bandwidths and real-time access to radio network information, which can be further leveraged by applications deployed in the ecosystem. MEC solution also facilitates the operators in terms of opening RAN edge to authorized third-parties that can deploy innovative applications and services towards mobile subscribers, enterprises and vertical segments in a fast and flexible manner. MEC is also perceived as necessary facilitation feature for latency-critical applications. Another, usually underestimated gain on LBO for MEC, is reshaping the transported traffic distribution for avoiding unnecessary transmission path loops, off-loading the TN and finally having less demand for installed TN capacity.

Interconnection of UAV and 5G ecosystems imposes the creation of the specific abstractions that could facilitate the drone actors with the access to the network and compute resources offered by telco operators or, in case of the 5G!Drones project, trial facilities. A unified view of network slice management services provided by the network together play a vital role in terms of forming facility federation out of the distinct 5G network slicing-enabled solutions provided by consortium members. Granting the aforementioned features is vital regarding performing 5G!Drones trials in a consolidated and efficient manner, in particular, to facilitate the connection between trial facilities and trial controller.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

3GPP	The Third Generation Partnership Project
4G	The Fourth Generation of Mobile Communications
5G	The Fifth Generation of Mobile Communications
5GC	5G Core network
5GPPP	Fifth Generation (5G) Public Private Partnership
ANM	Autonomic Network Management
AP	Application Plane
AppD	Application Descriptor
BVLOS	Beyond Visual Line Of Sight
CN	Core Network
CNM	Cognitive Network Management
CP	Control Plane
DASMO	Distributed Autonomous Slice Management and Orchestration
DN	Data Network
DNS	Domain Name Service
DP	Data Plane
EEM	Embedded Element Manager
EM	Element Manager
eMBB	Enhanced Mobile Broadband
eNB	Evolved Node B
EPC	Evolved Packet Core
ETSI	European Telecommunications Standards Institute
FCAPS	Fault, Configuration, Accounting, Performance, Security
HPLMN	Home PLMN
IMSI	International Mobile Subscriber Identity
ISM	In-Slice Management
ITU-R	International Telecommunication Union – Radiocommunication Sector
ITU-T	International Telecommunication Union – Telecommunication Standardization Sector
K8S	Kubernetes
KPI	Key Performance Indicators
KQI	Key Quality Indicators
LBO	Local Break-Out

LCM	Life Cycle Management
LTSM	Long Short-Term Memory
MANO	Management and Orchestration
MAPE	Monitor-Analyse-Plan-Execute
MEAO	Mobile Edge Application Orchestrator
MEC	Multi-access Edge Computing
MEP	MEC platform
MEPM	MEP Manager
MF	Management Function
MME	Mobile Management Entity
mMTC	Massive Machine Type Communications
NF	Network Function
NFV	Network Function Virtualization
NFVO	NFV Orchestrator
NS	Network Slicing
NSA	Non-standalone
NSI	Network Slice Instance
NST	Network Slice Template
OSS/BSS	Operation System Support/Base Station System Support
PDN	Packet Data Network
PGW	PDN Gateway
PLMN	Public Land Mobile Network
QoS	Quality of Service
RAN	Radio Access Network
RB	Resource Block
RNIS	Radio Network Information Service
SDN	Software Defined Network
SGW	Serving Gateway
SM	Slice Manager
SON	Self-Organizing Networks
SORA	Specific Operations Risk Assessment
TAC	Tracking Area Code
TMN	Telecommunication Management Network
TN	Transport Network

UAS	Unmanned Aircraft System
UASP	UAV Service Provider
UAV	Unmanned Aerial Vehicle
UE	User Equipment
UP	User Plane
UPF	User Plane Function
URLLC	Ultra-Reliable Low Latency
V2X	Vehicle to Everything
VIM	Virtualized Infrastructure Manager
VLOS	Visual Line Of Sight
VNF	Virtual Network Function
VNFM	VNF Manager
VPLMN	Visited PLMN

1. INTRODUCTION

1.1. Deliverable scope

The 5G!Drones context dictates an entire Work Package (WP3) to "Enabling mechanisms and tools to support UAV use cases". The main focus of WP3 is laid on the development of the 5G!Drones enablers that allow to run the UAV use cases and to meet their requirements identified within WP1 work. Specifically, the desired enablers include:

- **Scalable end-to-end slice orchestration, management and security mechanisms (T3.1)** with a particular focus on security aspects and extensions in network slicing and advanced slicing mechanisms.
- **MEC capabilities for the support of 5G!Drones trials (T3.2)** especially the necessary support for the inclusion of MEC application instances and related network and compute resources into an end-to-end UAV slice, obligatory enhancements regarding isolation of slices, as well as extensions concerning slice awareness, resource isolation and security in multitenant MEC environment. Furthermore, challenges for MEC related to UAV mobility aspects is to be thoroughly researched.
- **Mechanisms of infrastructure abstraction and federation of 5G facilities (T3.3)** specifically, a unified interface that will enable exposure of facility capabilities and deploy functions, there are to be defined and developed. The aim of the interface is to provide a single abstraction for the network (e.g. RAN) and compute resources (e.g. provided from a central or MEC datacentre).

This document reports work done regarding 5G enablers of concerns related to the topics mentioned above.

1.2. Organization of the document

The document is organized as follows:

- **Section 1 (current section)** is an overall introduction to the document and discusses the scope of WP3 as well as the role of interaction infrastructure enablers with aviation domain processes;
- **Section 2** focuses on selected issues related to end-to-end slice orchestration and management (T3.1);
- **Section 3** discusses MEC capabilities in terms of support for 5G!Drones trials (T3.2);
- **Section 4** discusses MEC and network slicing capabilities in terms of support for 5G!Drones trials and infrastructure abstraction and federation of 5G facilities (T3.3);
- **Section 5** concludes the report.

2. SCALABLE END-TO-END SLICE ORCHESTRATION AND MANAGEMENT

In the context of 5G!Drones, each use-case has very special requirements, in terms of latency, throughput, reliability or number of supported devices, for example. Those needs are specified by the four categories defined in 5G NR (New Radio) standards:

- eMBB – enhanced Mobile Broadband;
- mMTC – massive Machine Type Communication;
- URLLC – Ultra-Reliable Low Latency Communication;
- V2X – Vehicle to Everything for vehicle communications.

Therefore, end-to-end network slicing is a crucial element of the 5G!Drones architecture, because it ensures that these heterogeneous service types coexist and provide each user with its required quality of service (QoS). The concept of network slicing is a cornerstone of 5G NR to allow the coexistence of several verticals and different services on a single physical platform. Infrastructure virtualisation is the primary enabler of network slicing by enabling the deployment and reconfiguration of new services on the fly in standard equipment. Thus, a vertical can independently deploy and orchestrate its services on a network resources shared by several other verticals. More specifically, a slice can be deployed for each service with dedicated QoS guarantees.

This resource management technique ensures isolation between verticals and services and sharing of the infrastructure, which reduces the cost for operators. Performance isolation between slices means that insufficient resource in a slice will not affect the performance of another slice. The possibility of slice reconfiguration on-the-fly is also the main advantage that allows adequate management of infrastructure and services. In this context, 3GPP defines the "Network Slice Instances" (NSIs) distributed in the architecture in the form of "Network Slice Subnet Instances" (NSSIs), as depicted in Fig. 1.

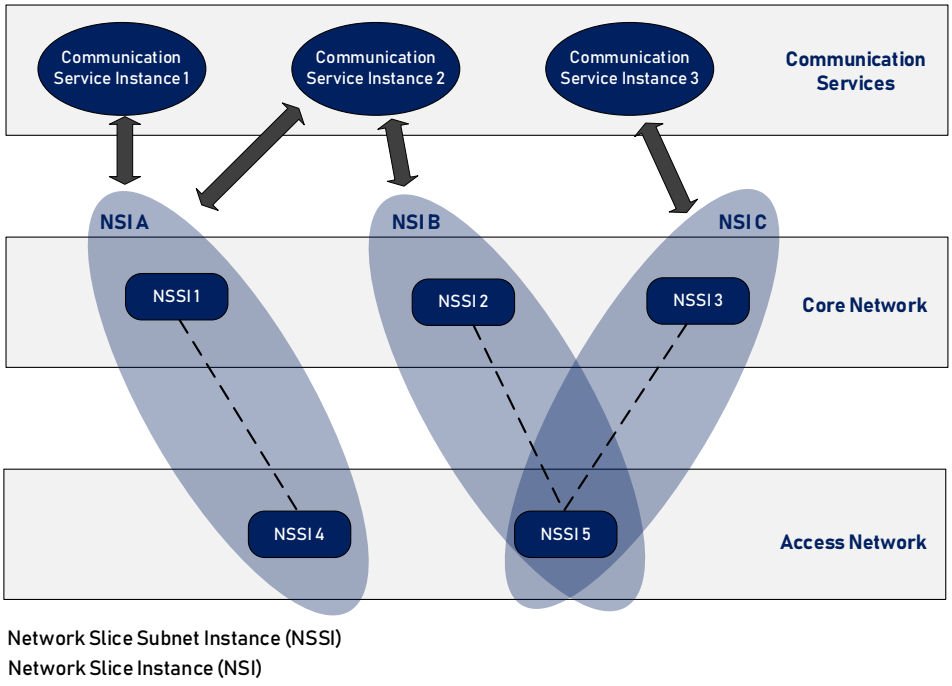


Fig. 1: 3GPP Slicing model [1]

Starting from Release 15, 3GPP introduced in its 5G core network (5GC) the set of Network Functions (NFs) that can constitute the Control Plane (CP) and the User Plane (UP) of an NSI. This includes, but is not limited to, the Access and Mobility Management Function (AMF), which is responsible for the

management of UEs registrations and handling user mobility. The AMF also provides interfaces towards the RAN. The Unified Data Management (UDM) holds the profiles and information of subscribers, while the Session Management Function (SMF) manages the life cycle of PDU sessions from the establishment to the release. The User Plane Function (UPF) is responsible for the routing and forwarding of packets that belong to the ongoing PDU sessions and plays the role of the interconnection points between the RAN and the Data Network (DN). Moreover, 3GPP introduced the Network Slice Selection Function (NSSF) that provides assistance for selecting NSs upon a network service is requested, and the Application Function (AF) that provides means for third parties applications to interact with the 5GC and NSIs. Fig. 2 depicts the service-based architecture of the 5GC.

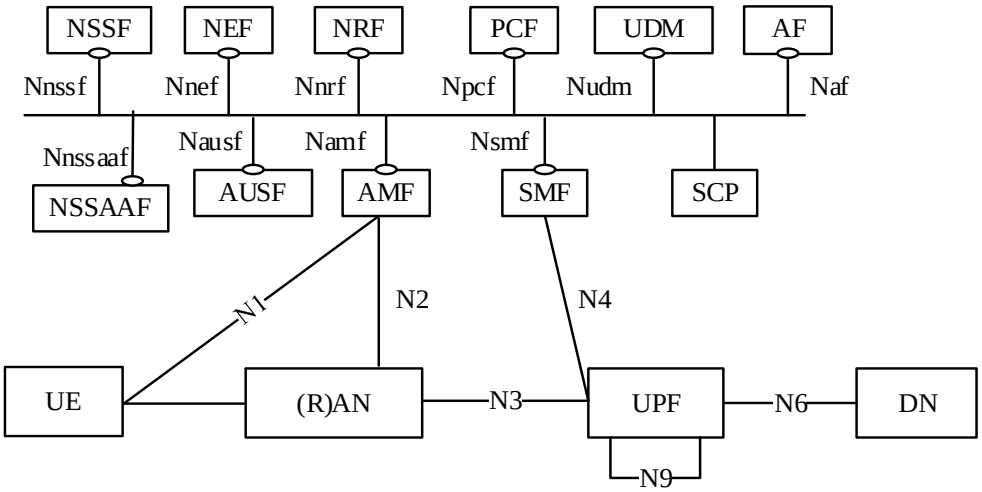


Fig. 2: 5G System Architecture [2]

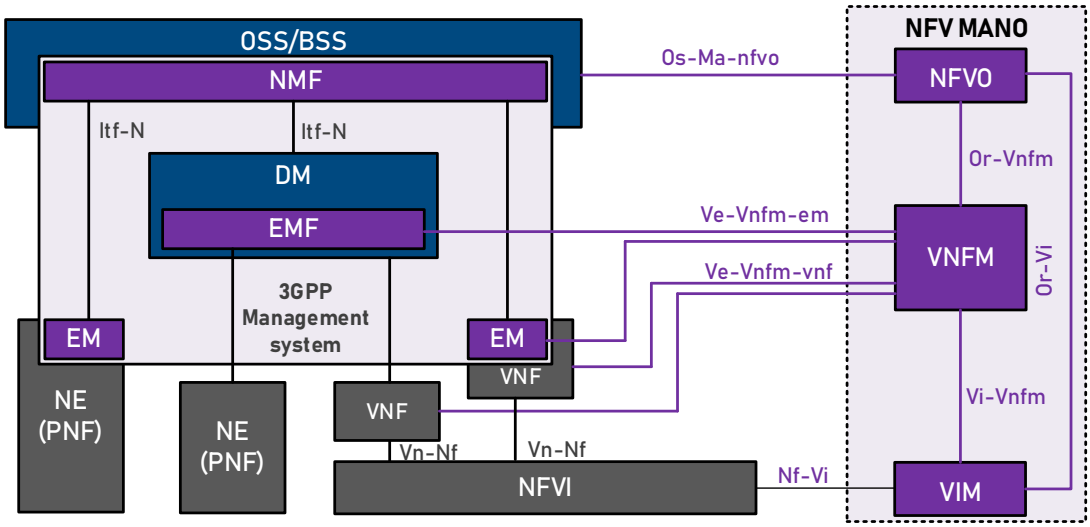


Fig. 3: The mobile network management architecture mapping relationship between 3GPP and NFV-MANO architectural framework [3]

The 3GPP has also started working on several aspects of management of the life cycle of network slices in the context of 5G network management and orchestration. The 3GPP has defined the following management functions (MFs) related to network slicing: Communication Service MF, Network Slice MF and Network Slice Subnet MF [4]. The report [5] lists the network slicing related issues that include FCAPS of slices, SON evolution for network slice management and orchestration of network slices across single or multiple administrative domains. The management architectures for network slicing

enabled softwarized communication networks by a principle following the ETSI NFV concept. The 3GPP view on management architecture is complementary to ETSI NFV MANO framework, where the 3GPP management system is an expansion of the OSS/BSS and EM part of MANO (see Fig. 3).

The hierarchical 3GPP management vision distinguishes between NF management (i.e. EM according to ETSI), NSSI management and NSI management. The last two can be considered as two levels of OSS/BSS, according to ETSI. Additionally, 3GPP acknowledges utilization of reference points and interfaces defined by ETSI NFV MANO – the 3GPP management system shall be capable of consuming NFV MANO interface (e.g. Os-Ma-nfvo, Ve-Vnfm-em and Ve-Vnfm-vnf reference points) [2].

A Fully-Fledged end-to-end NSI consists of the interconnection of multiple NSSIs, each NSSI belonging to a different technology domain. In general, an E2E NSI should be composed of three NSSIs: RAN NSSI, TN NSSI, and CN NSSI.

- **RAN NSSI** – access points that constitute the RAN of 5G networks should support the slicing of radio access services and radio resources. While the slicing of radio access services consists of creating a set of isolated Virtual Network Functions (VNFs) that run the access functions, the slicing of the radio resource is provided as a set of Radio Resource Management (RRM) policies that enforce the allocation of Radio Resource Blocks (RRBs) needed to run an NSI with a specific Service Level Agreement (SLA).
- **CN NSSI** includes the elements that constitute a full or partial instance of the CN. Each element in the CN NSSI runs as an isolated VNF on top of the virtualisation infrastructure. Moreover, the components of a CN NSSI are interconnected internally and externally (i.e. with TN and Data network) using a set of virtual links. The service differentiation is ensured at the level of CN NSSI by the customization of the number of computing resources allocated to each VNF, and the number of network resources allocated to the virtual links interconnecting the VNFs. Another important factor that can be used for achieving service differentiation at the level of CN NSSI is the customization of the placement of the composing VNFs, for instance, the placement of UPFs near to the end-users (i.e. at the edge of the network) can reduce the communication latency considerably.
- **TN NSSI** allows the interconnection of the distributed VNFs that constitute the CN NSSI with each other, as well as with the RAN NSSI. TN NSSI is enforced by the instantiation of a set of VNFs (e.g. switches, routers, firewalls, etc.) that realize the network UP, and by the configuration of traffic rules that realize the network control plane (usually using Software Defined Networking, SDN). Same as the CN NSSI, the service differentiation is ensured at the level of TN NSSI by the customisation of the number of computing resources allocated to each VNF, and the number of network resources allocated to the virtual links interconnecting them. Moreover, SDN-based control of traffic can enable comprehensive QoS management for the different TN NSSI.

The life cycle of E2E network slices is composed of the following phases:

- **Preparation phase** includes the design of the network slice, evaluation of network slice requirements, feasibility check (e.g. availability of resources), on-boarding of VNFs packages, configuration and instantiation of the dependencies required for instantiating the new network slice.
- **Commissioning phase** includes the reservation of computing and network resources required by the new network slice instance. Moreover, the commissioning phase may trigger NSSI(s) creation or using existing NSSI(s) and setting up the corresponding associations with the new NSI.
- **Operation phase** includes the activation/deactivation of the NSI to indicate its availability/unavailability for providing communication services. Moreover, in this phase, it would

be possible to modify the NSI, e.g. changes of NSI capacity, changes of NSI topology, NSI reconfiguration.

- **Decommissioning phase** includes the termination of the NSI by releasing all the resources used by the NSI.

Both ETSI and 3GPP visions are operator-centric and do not include a broader perspective. The point is that the network slicing brings a serious revolution in the way the communication networks will be designed and operated. From the network operator's point of view this is just splitting of one, universal and multi-service communication into parallel component networks that are adapted to support certain specific classes of services with distinct properties, and hence having separate requirements, which may be conflicting with requirements of other classes. There is also no simple 1:1 mapping between the operation of the communication network and ownership of the infrastructure as well as operating an NFV MANO stack. This is the reason for concerns about proper overall management architecture, especially with regard to its scalability, i.e. the ability of management environment to grow according to the managed entities expansion. Another issue is the optimization of the management in terms of information exchange, a delegation of tasks to shorten feedback loops, the ability of autonomous mechanisms implementation, exposure of management interfaces for slice users/tenants (not only for the host-operator) and finally resources consumption by the management itself.

Slice management differs from classical network management. In the network slicing case, there is a need to manage not a single, but multiple networks – this makes the scalability of management extremely important. Moreover, as it has been already mentioned, the management functions of a slice should be split between slice tenants and the network slicing system operator. Due to the software dimension of slices, there is also a need to provide cooperation of the management and orchestration systems, which functionalities partially overlap. From the management point of view, a single network slice (network instance) can be treated similarly as a classical network. Therefore, the generic scheme of Telecommunication Management Network (TMN), as defined in the ITU-T recommendation M.3000 [6], can be applied. However, some modifications related to the software nature of such networks are needed. As the network slices are mostly based on software entities, the management and orchestration of them can use the ETSI NFV MANO approach. In this framework, the management part of the system (OSS/BSS) drives the NFV Orchestrator (NFVO) to perform management and orchestration of MANO compliant solutions. The NFVO performs not only the NS LCM but also dynamically allocates resources to provide the required performance and handle faults. Recently, ETSI started working on incorporating network slicing within the Release 3 of NFV MANO specifications [7]. They plan to address the scalability of orchestration, multi-tenancy of NFVO and support for the creation of the multi-domain slices.

So far, we have found no approach that is looking into the **scalability of slice management** – the existing approaches are typically centralised ones (at least per domain level). We have found an approach to the integration of Cognitive Network Management (CNM) or Autonomic Network Management (ANM) with ETSI MANO, but not in the context of network slicing [8]. For example, the distribution of management functions according to the In-Network-Management concept (INM) [9], and the Autonomic Network Management (ANM) technique, can be used to solve the management scalability problem. The ANM concept was developed a long time ago in the context of autonomic computing [10]. The LTE SON (an ANM approach) is already used for automated RAN management (handover and coverage optimisation, energy-efficient operations or plug-and-play eNodeB deployment). Recently, an ANM variant that has learning capabilities, i.e. CNM, is popular in the context of 5G networks. It is worth noting that GANA (Generic Autonomic Networking Architecture) [11] is a subject of ongoing standardization by ETSI. Recently, ETSI started a new activity called Zero Touch Network that is also based on ANM/CNM [12].

2.1. Shared functions in network slicing

According to 3GPP [2], a Network Slice (NS) is defined as a logical network that provides specific network capabilities and network characteristics. Each Network Slice Instance (NSI) is defined within a Public Land Mobile Network (PLMN) as a set of Network Functions (NF) instances that constitute the Control Plane (CP) and User Plane (UP) of that instance. An NSI can have its own dedicated CP and UP network functions or can share a subset of them with other NSIs. Fig. 4 depicts the high-level architecture of network slicing with shared and dedicated NFs. The 5GC architecture, with the explicit split of the CP and UP, has some appealing features for network slicing. Indeed, with such design it would be possible to perform horizontal scaling of the CP and UP independently, for example, the operation of creating a new NSI can be considered as a simple horizontal scaling of the UP. Moreover, the CP/UP separation allows flexible deployment of network slices, for example, the CP can be deployed in a centralized location, whereas the UP can be distributed across edge servers for realizing a low-latency NSIs.

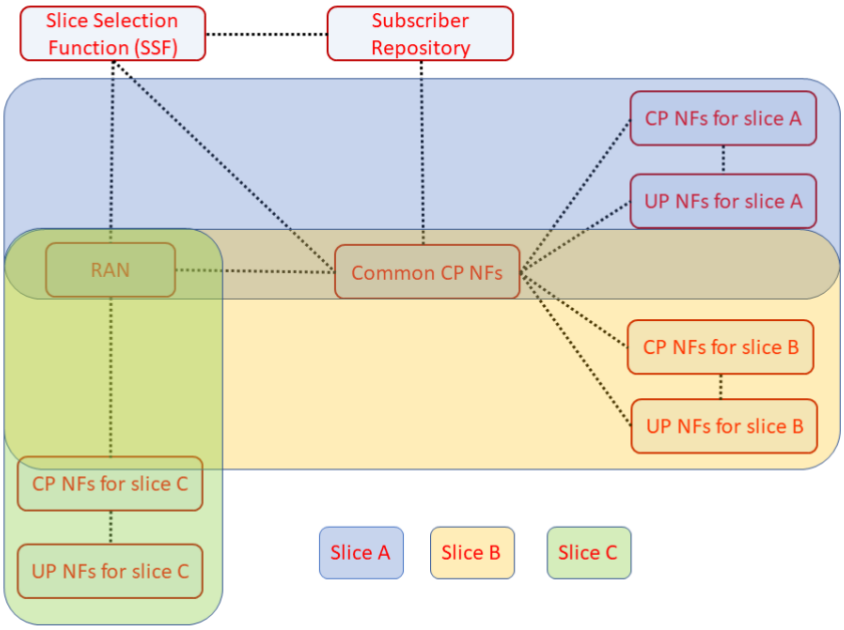


Fig. 4: The high-level architecture of network slicing with shared and dedicated NFs [1]

An example of network slicing with 5GC is depicted in Fig. 5. In the 5G NSA (Non-standalone) deployment mode, the 5G RAN and its New Radio (NR) interface is used in conjunction with the existing LTE and EPC infrastructure, making the NR technology available without network replacement. In such deployment, only 4G services are supported, but enjoying the capacities offered by the 5G NR (lower latency, etc.) is still possible [13]. However, even that several benefits induced by the microservices-based design of the 5GC are lost when using the NSA mode, it is still possible to follow the same aforementioned approach for network slicing. In fact, the principles of separating the CP and UP were first introduced by 3GPP in the latest version of the EPC [14] by adopting the Control and User Plane Separation (CUPS) architecture. The main enhancement that was introduced in the CUPS architecture is the separation of the CP and UP of the SGW (Serving Gateway) and PGW (Packet Data Network Gateway) network functions and the definition of new interfaces between the resulting network functions. That is, the SGW was split to two new network functions, namely the SGW-C and the SGW-U. Whereas, the PGW was split to PGW-C and PGW-U.

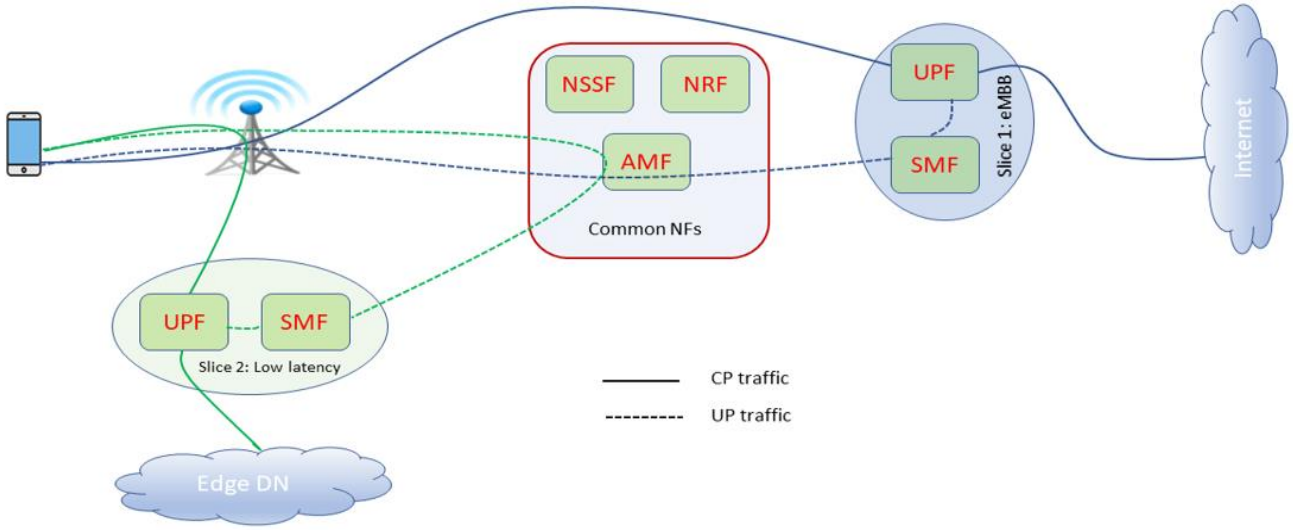


Fig. 5: Example of network slices

The communication between the newly introduced CP and UP function is based on a protocol that was specifically defined for this purpose, which is the Packet Forwarding Control Plane (PFCP) protocol. 5GC is considered as a natural evolution of the CUPS architecture. Hence, it is possible to map the NFs of the 5GC to the NFs of the CUPS EPC as depicted in Fig. 6.

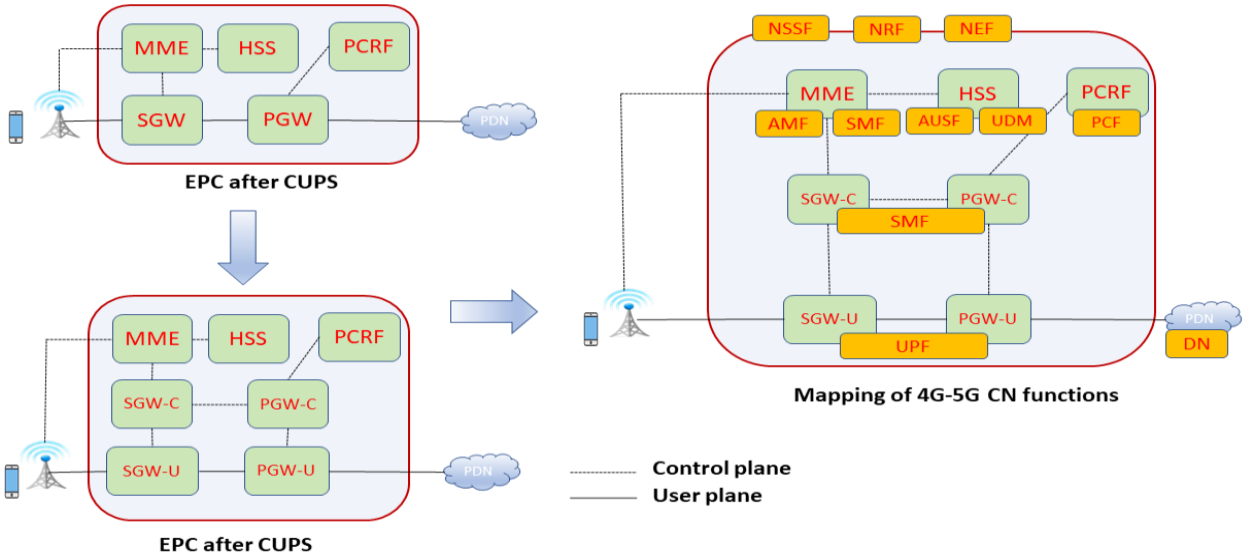


Fig. 6: Evolution of 3GPP CN

2.2. RAN slicing issues and their impact on management

While network slicing in the CN has been defined, thanks to the well-known concepts of SDN and NFV, the development of network slicing in the RAN is still in its early stages. The fact that part of the RAN relies on wireless communications brings new challenges to this concept, like resource management, spectrum sharing and isolation, for example. Indeed, on the radio segment, ensuring QoS mainly comes down to performing adapted scheduling to match the demands of each traffic. This allocation of radio/frequency resources (Resource Blocks) guarantees a bit rate for each service or slice. In order to

extend the QoS guarantees on the radio segment, a module called RAN controller has been introduced in the literature to allow the extension of the QoS already established in the core and at the edge of the network. Of course, the policies applied by the RAN controller must reflect those of the CN slice orchestrator. In order to address these challenges, we propose the architecture for implementing RAN slicing, which is described in the following section.

One of the key concepts of this architecture is the separation between the Control and Data Plane (i.e. User Plane). This enables the independent management of both planes and thus facilitating the scalability of DP nodes. In other words, a single RAN controller can be in charge of different base stations, through dedicated agents. And a base station can be added at any time. In the proposed architecture, the RAN Controller, presented in Fig. 7, is in charge of processing the information coming through its northbound interface from the management plane, and the network state information coming from the agents through its southbound interface. Based on the global network view built from this information, it provides configuration instructions to the Agents. One Agent is implemented in each base station. Its role is to implement the instructions issued by the controller.

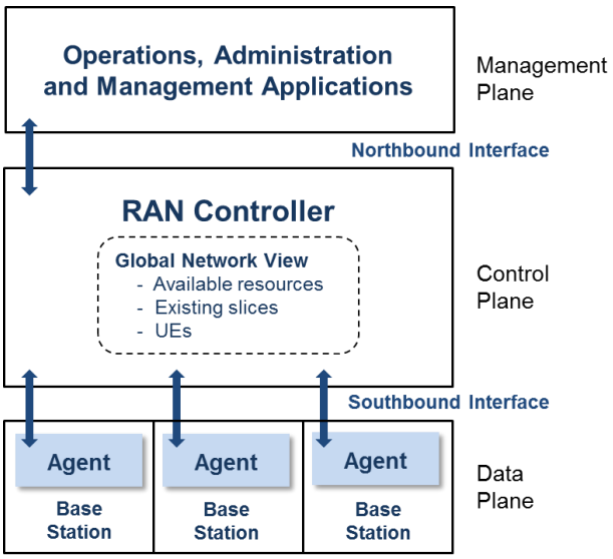


Fig. 7: High-level view of the proposed architecture

The main goals of our solution regarding resource management are:

- ensuring performance isolation between slices, meaning that insufficient resource in a slice should not affect the performance of another slice;
- allowing each slice to allocate its resource in its own way between the different UEs attached;
- efficient use of radio resources.

In order to achieve these goals, two levels of scheduling are performed. The first aims to allocate radio resources to each slice, and the second shares resources between UEs within a slice, depending on the slice's scheduling policy. Fig. 8 shows a high-level view of the scheduling model used in this architecture. It utilises the following entities:

- The **Slice Life Cycle Manager** responsible for the creation and destruction of slices. When the RAN controller receives a slice creation request, the Slice Life Cycle Manager checks that there is enough resource left and that the slice's configuration complies with the admission control mechanism.
- The **Hypervisor** being in charge of allocating the radio resources to the different slices. It provides an abstraction of the physical resources in the form of Resource Blocks (RBs), without specifying

their location on the grid. This way, the Hypervisor can reallocate the resource in real-time, and it will be invisible for the higher layers.

- Within each slice, a **Specific Scheduler** being in charge of allocating the virtual resources provided by the Hypervisor to the UEs. The default scheduling policy is a Round Robin mechanism, but it can be modified in the slice configuration.

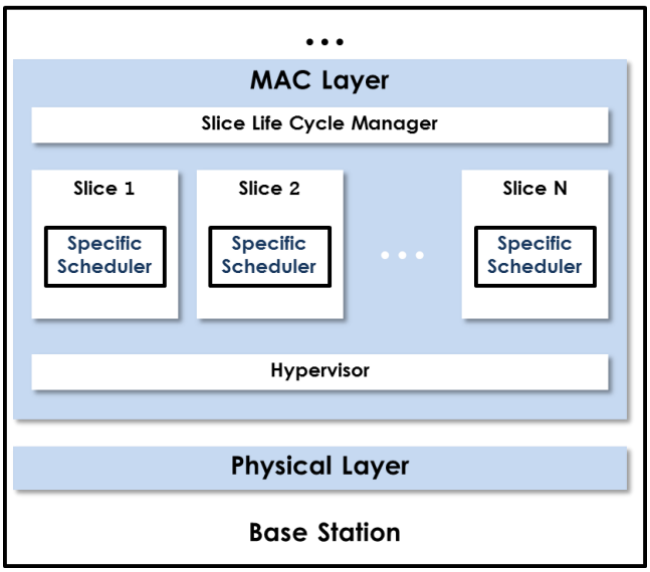


Fig. 8: Scheduling model

2.2.1. Functionalities of existing RAN controllers

The following section is a brief state of the art of available RAN controllers that could be used to implement slicing in the access network of the 5G!Drones environment.

FlexRAN

FlexRAN ([15], [16]) is an SD-RAN platform enabling slicing and separation of CP and DP in the RAN. In this architecture, each base station has its own DP embodied by a FlexRAN agent. All of these agents communicate with a centralised controller through its southbound API. This flexible and programmable control plane makes it easier to manage all the base stations belonging to the network and facilitates the development of control applications. FlexRAN includes a mechanism allowing the master controller to delegate scheduling decisions to the agents, leading to reduced latency and distributed computation. FlexRAN is based on Open Air Interface, a stack implementing the RAN as well as the CN in LTE or 5G NR.

Orion

Orion [17] is a RAN slicing architecture based on FlexRAN that enables the dynamic on-the-fly virtualisation of base stations. It introduces a hypervisor connecting each base station to the CP of each slice. This hypervisor must ensure that each slice has the resources necessary for its proper functioning and guarantee isolation from the other slices in a dynamic way. The resource can be reallocated to follow the requirements in real-time. In this system, the Physical Resource Blocks (PRB), radio resources to be allocated, are virtualised and allocated via pools of virtual resource blocks to the slices.

RAN Runtime

RAN Runtime [18] is a RAN slicing system also based on FlexRAN and developed by Eurecom. Its particularity compared to Orion, is the personalisation it offers to the slices. A common set of RAN modules, accessible through the RAN Runtime API, is shared between slices. They include different RAN functions and resources that can be used to customize a slice. The isolation level of a slice can also be determined. It can be completely isolated, shared across all network layers, or customized for a subset of CP and DP. RAN Runtime also allows more flexible allocation of PRBs than Orion. Indeed, it allows reallocating resources not allocated to other slices. Four levels of granularity are introduced in RAN Runtime for the allocation of resource blocks:

- Contiguous – resource blocks allocated according to this granularity are contiguous in the grid;
- Non-contiguous – resource blocks allocated according to this granularity may not be contiguous in the grid;
- Fixed position – the resource blocks have a fixed position in the grid of resources and cannot be reallocated in another place;
- Minimum granularity – for this granularity, the slice does not require resource blocks but a specific capacity. RAN Runtime will then allocate as few resource blocks as possible while complying with the demand.

Each slice chooses the granularity according to its needs. The main objective of RAN Runtime is to maximize the satisfaction of the slices in terms of allocation of requested resources as well as to maximize the number of unallocated resources in the event that another slice comes to request these resources. This technology allows greater customisation of slices and more flexible allocation of resource blocks.

5G-EmPOWER

5G-EmPOWER ([19], [20]) is an open-source platform supporting RAN slicing. It is composed of three main elements. On the DP (UP), an Agent is implemented in each base station to enforce the instructions issued by the controller. The latter is in charge of processing the information coming from the management plane through its northbound interface, and the network state information coming from its southbound interface. Based on the global network view built from this information, the controller provides configuration instructions to the Agents, using the OpenEmpower protocol. The last part is the management plane, relying on the REST API to manage slice parameters. Each slice can be configured independently with a number of allocated PRBs and a scheduling mechanism. The LTE stack used by default in 5G-EmPOWER is srsLTE. Implementation of OpenAirInterface is also possible. One of 5G-EmPOWER's strength is that it is able to reallocate unused resources in order to increase performance.

2.3. In-slice management – an example of scalable slice management

The heterogeneity, as well as the amount of data produced by each network slice, will raise significant issues related to management and orchestration scalability and complexity, making it infeasible to handle in a traditional way, i.e. by using one huge central OSS/BSS and/or MANO stack. Additionally, the network slicing is perceived as a key tool for creation of slices that are tailored to the needs of 3rd parties (verticals), who – in most cases – want to manage their slices (this is also the case of UAV Service Providers – UASPs) in an unhindered manner (adaptability to the current situation, customers profiling as well as confidentiality) do not need to be professional network operators (in some cases they may be even the end-users). Therefore, the management system provided to the tenants should aim for ensuring simplicity, specialized support, i.e. with embedded intelligence as well as high flexibility and dynamism.

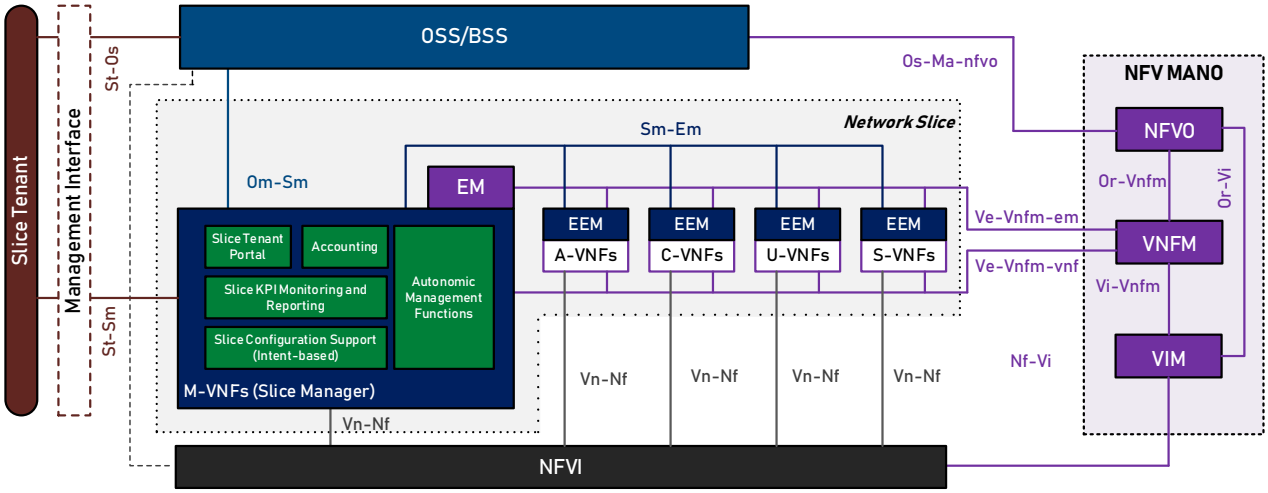


Fig. 9: Intent-based management framework with the internal structure of network slices – ETSI NFV MANO extensions (slice management plane shown in navy blue)

In [21], a reference architectural framework for network slicing, based on the ETSI NFV MANO architecture [22] and compliant with various communication network architectures has been proposed. The architecture facilitates vertical and horizontal slice expansion by incorporation of common/dedicated slice concepts, exposure of slice functions via slice API and slice stitching (slice concatenation, e.g. via API). The framework follows the paradigm of hierarchical multi-domain orchestration and supports tenant-oriented operations and interfaces based on embedded in-slice managers. In [23] the internal structure of slices has been further defined – the core part of the slice, consisting of functions composing the Application (AP), Control (CP) and User (UP) Planes (A-VNFs, C-VNFs and U-VNFs, respectively), is accompanied by two special functional blocks: Slice Manager (SM) and Slice Operation Support (SOS), both implemented as sets of VNFs (M-VNFs and S-VNFs respectively), belonging to slice template and sharing the life cycle of their slice. The described intent-based management framework is presented in Fig. 9. The central point of the slice management plane is SM that is linked to Embedded Element Managers (EEMs) of VNFs implemented within a slice. The EEM follows the principles of ETSI NFV concept of Element Manager (EM) but is enhanced with additional functionalities, which facilitate slice-level management support, VNF monitoring, actuating and autonomic control loop, etc. SM plays a role of slice OSS and implements the mechanisms responsible for slice-level monitoring, analysis, actuating and autonomic control loop according to the Monitor-Analyse-Plan-Execute (MAPE) [10] model (real-time feedback loop).

Moreover, SM implements tenant-oriented functions: accounting, KPI monitoring and reporting, configuration support (following the "intent-based management" paradigm), which are exposed via the Management Interface (Tenant Portal) functionality of SM. SM also exposes an interface to the global OSS/BSS, which is of importance, especially in multi-domain slicing. SOS functions support slice-level operations as slice selection, subscription, authentication and stitching of sub-slices to provide transparent communication between NFs belonging to different domains for creation of the end-to-end slice. The described architecture implements the In-Slice Management (ISM) concept, which ensures scalability by the hierarchical distribution of management tasks.

Implementation of the proposed intent-based management system poses specific requirements:

- To each VNF that is a part of the "Core" part of the slice (the VNFs that are used for the implementation of the UP, CP and AP), appropriate EEMs that implement the node level autonomic behaviour and send pre-processed monitoring data to SM have to be added.

- Each ”Core” part of a slice has to be supplied with individual management counterpart, i.e. the SM in the form of VNFs implementing most of the in-slice management functions, including the intent-based management by slice tenant. Further optimization of the management operations can be achieved by VNF distribution. The SM component should be a part of a slice blueprint, similarly to the SOS part of the slice.

The OSS/BSS, to be compatible with described intent-based management framework, has to be equipped by appropriate interfaces for handling multiple SMs. Nevertheless, in comparison to OSS/BSS-only management, instantiation of the concept enables migration of some of the slice-specific functionalities out of OSS/BSS. These components should be initiated together with the slice. Therefore, they should be added to the Network Slice Description but placed within the OSS/BSS that way, providing it with a certain level of programmability.

It is hard to define a priori an optimal split of management functions between EEMs, SM and OSS/BSS. The ultimate goal would be to obtain the OSS/BSS functionality slice agnostic and to keep the slice specific management handled by the Slice Manager. Such an approach will provide higher flexibility and seamless integration with any type of slice regardless of its particular properties. Nevertheless, achieving such a goal is problematic and split of management functionalities will be dependent on the implementation and instantiated slice type.

In case of the 5G!Drones project it is necessary to specify the management functionality of in-slice management as well as the roles of the interfaces Om-Sm (used by the trial controller) and St-Sm (may be used directly by the UASP) indicated in Fig. 9. There is also an interaction of the trial controller (i.e. OSS/BSS) with the MANO orchestrator. As it is described in Section 4.5.1., the trial controller has to use the OSS/BSS and indirectly MANO and MEC interfaces to deal with:

- Network slices lifecycle management;
- VNFs management;
- MEC applications management;
- KPIs monitoring.

All the interfaces should go to the central OSS/BSS. However, there is a runtime management interface that should be enabled via the tenant interface. This interface should be used by the UASP to obtain information about slice KPIs and should allow for UAV service management operations. These operations will have yet to be defined within the 5G!Drones project as they are specific to UAV use case service components within the slice. The Slice Manager embedded inside the network slice can be involved in the calculation of KPIs that are exposed to UASP and trial controller. The presented concept does not change the functionality it only changes the placement of functions.

2.3.1. In-slice management concept implementation

The implementation of the concept, as described in the previous section, is expected in the later phase of the project. Exemplary instantiation of intent-based management compliant with the ISM paradigm has been presented in Fig. 10. Instead of MANO orchestrator, as originally proposed in the concept, the VNFs are orchestrated by using lightweight K8S (both management framework VNFs as well as slice specific network functions) orchestrator and Docker as the containerization technology.

The software components used within the architecture can be grouped into the following categories:

- ISM components;
- Orchestration-related components;
- Tenant interface;

- A set of VNFs implementing the desired functionalities of the network slice;
- Infrastructure monitoring.

ISM software components include EEM, SM and Slice Creator (SC), Python and bash script-based applications. Modules that implement the logic for interaction with a tenant, i.e. SM or SC also utilize Flask for web API implementation. Moreover, SM uses a Postgres database for storing network slice specific management data. Newly introduced module in comparison to the original ISM concept is Slice Creator component, which is the part of OSS/BSS and enables resolving slice tenant requests regarding for instantiation of specific network slice. Based on the tenant's request, the appropriate template is selected, i.e. a set of K8S resources description that have to be created (e.g. deployments, stateful sets, services, config maps, secrets etc.). The set of configuration files is afterwards fed by SC to K8S via K8S API. Application images that are instantiated during slice creation are stored within local VNF repository (Gitlab Docker image registry) or downloaded from external registries, e.g. Docker Hub.

Each VNF package, which implements network slice functionalities is wrapped in a bundle containing: EEM component, the image of the software module logic (e.g. NRF) and management config file called Manifest, which contains the set of commands that can be executed on the running container. The connectivity between the EEM and SM is provided by using RabbitMQ – MQTT message broker.

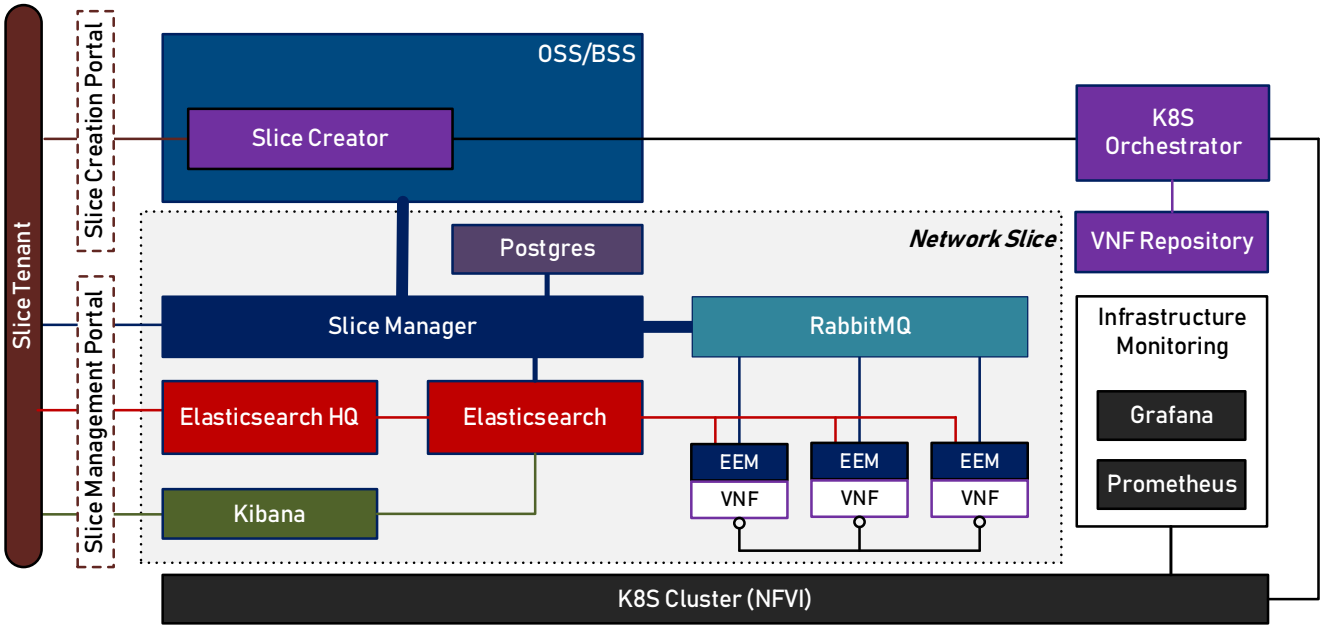


Fig. 10: Exemplary implementation of ISM utilizing K8S as the VNF orchestrator

To monitor the state and operation of each slice components, the following tools are used:

- Elasticsearch (ES) – a distributed, search and analytics engine using REST integrated with the database. It provides a robust set of features that enable to effectively search, index, and analyse data of diverse shapes and sizes—used for storing and indexing EEM and SM logs, which are directly sent to the ES cluster, for tenant analysis and provide necessary data for Autonomic Management mechanisms.
- Kibana – a free and open user interface that enables visualisation of Elasticsearch data and navigation throughout the Elastic Stack. It provides mechanisms for, i.a. data exploration and visualisation, management and monitoring of data, alerting, machine learning, e.g. anomaly detection etc.
- Elasticsearch HQ – a tool that provides mechanisms for management and monitoring of Elasticsearch clusters as well as stored data and rules applied to the incoming data, e.g. index

management, easy database access and data management, monitoring of cluster health, node load, consumed space etc.

As mentioned before, the slice tenant can request a desired slice template to instantiate via Slice Creation Portal and afterwards manage the created slice via Slice Management Portal. Both Portals use NGINX as the reverse proxy, React.js for GUI implementation and are interconnected with the ingress providing secure, connectivity to the "world outside" the K8S cluster.

The ISM implementation also provides means for infrastructure monitoring, which is provided by:

- Prometheus – an open-source monitoring and alerting toolkit. It facilitates both machine-centric monitoring, i.e. infrastructure parameters as well as highly dynamic service-oriented architectures.
- Grafana – tool for visualisation of monitoring data collected from applications as well as infrastructure such as the utilization of resources (e.g. CPU, storage). In the described setup, Grafana enables exploration of data acquired from K8S cluster nodes by Prometheus agents.

2.3.2. Management role in SLA

UAV services impose specific requirements regarding quality and reliability. The specific needs can be however fulfilled by being compliant with appropriate Service Level Agreement (SLA) – a set of rules and parameters ensuring the particular telecommunication grade service. The overall picture of factors impacting service SLA has been presented in Fig. 11.

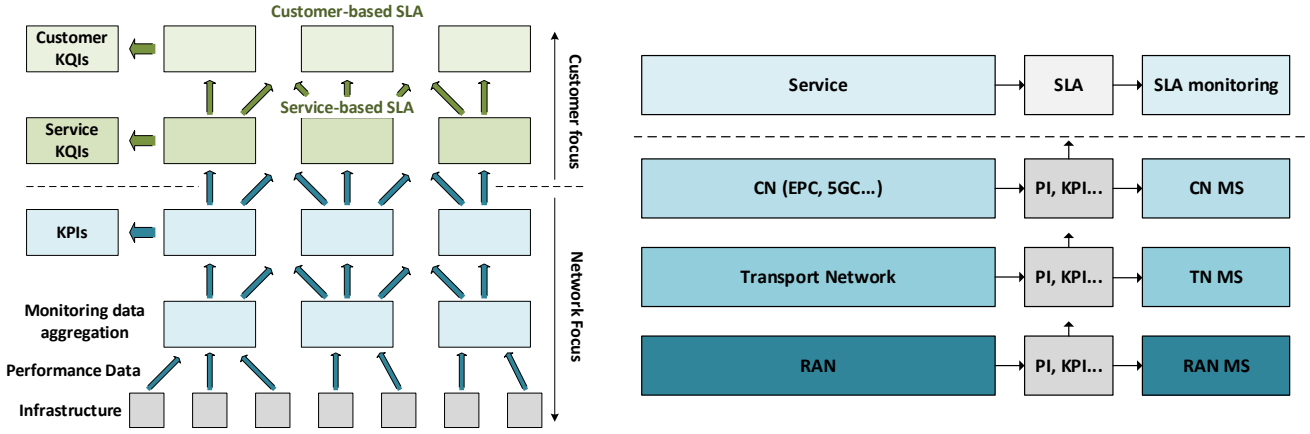


Fig. 11: The overall structure of SLA (left) and exemplary factors that can impact service-based SLA (right)

In general, the service-based SLA is contributed by several factors that are organised in a hierarchical way, which describe specific network qualitative parameters on different levels. Typically applied categories are Performance Indicators (PIs), Key Performance Indicators (KPI) and Key Quality Indicators (KQI). Separate domains, e.g. CN, RAN, TN etc. introduce their own sets of performance metrics that assess the quality of operation and contribute to the final SLA, i.e. low KPIs in one domain can impact KPIs of other domains and degrade the QoS. Maintaining the desired level of operation is, therefore of vital importance, especially in services that involve considerable risks and require very high reliability such as UAV or V2X scenarios.

Apart from the aforementioned domains, the important contribution to overall SLA is also added by slice orchestration and management system. Therefore, defining KPIs that regard slice management, slice runtime as well as slice LCM have to be defined. The exemplary solutions are described in the following sections.

2.3.3. Network Slice KPIs

Currently used runtime KPIs have been introduced by ETSI [24] and adopted by 3GPP [25], [26] using the definitions by ITU-T [27] and offer a framework to assess performance and quality of 2G/3G/4G services' from the end-to-end perspective. The work on defining 5G network KPIs are conducted within several research projects ([28], [29], [30], [31], [32], [33], and [34]). At present, a lot of standardization efforts is put into defining criteria for 5G network performance and quality assessment. Typically, the quality assessment approach is focused on 5G services' requirements and characteristics, as defined by ITU-R (cf. [35], [36]) and 3GPP (cf. [37], [38]). So far, the following 5G KPIs related to network slicing has been defined [39]:

- Utilisation KPIs: mean number of Protocol Data Unit sessions for NSI and virtualised resource utilization by an NSI;
- Accessibility KPIs: a number of registered subscribers, registration success rate per (NSI);
- Integrity KPIs: end-to-end latency of the 5G network, upstream/downstream throughput for NSI and at N3 interface, in case of RAN – the User Equipment throughput.

Having in mind a possibly large number of NSIs, the number of NS KPIs has to be kept to the minimum to minimise the collection and processing overhead. The network slicing KPIs can be split into two categories: slice run-time and slice LCM-related.

The runtime KPIs have been defined for virtualized EPC (vEPC) that can be implemented as a slice. This is, in fact, the case when the KPIs related to the certain solution implemented as a network slice should be the same as in non-sliced implementation. In fact, slice run-time KPIs regard performance of a network or a service that is implemented as a slice and typically do not differ in case of non-sliced implementation of the network or solution. The only new slice-agnostic (in the virtualised implementation) mechanisms are related to the consumption of virtual resources by a slice and orchestration operations. One of the key operations regarding resources management is resource scaling in accordance with their usage. The three types of virtual resources, namely, connectivity, computing, and memory, are typically considered. Additionally, the usage of RAM and swap space and disc measurements can be performed separately [40].

Definition of the network slice runtime KPIs has to be done in accordance with a certain functional model of network slicing as well as its implementation – in the 5G!Drones project case this concerns the capabilities and tools deployed at facilities. However, the proposed by the project abstractions should enable facilities independent KPIs exposure.

The proposed solution for slice type agnostic KPI calculations follows the NGMN functional approach [41] with some extensions and the ETSI NFV MANO approach. The proposed concept uses both the KPIs defined by 3GPP [39] and MANO performance measurements specified by ETSI [40]. The approach is focused on the runtime performance evaluation linked with resource consumption. The slice KPI can be correct in underutilization of the resources, but such case shows inefficient resource allocation. The opposite case is overutilization of resources which typically leads to the degradation of slice runtime KPIs. For such KPIs, some additional information can be used. For example, the information about the number of links in which a predefined threshold has been crossed, etc. In the case of memory KPIs, we propose that the resource KPI is affected if at least one type of VNF's monitored memory resources crosses a predefined threshold (underutilization or overutilization).

2.3.4. Network slicing KPIs in the NFV MANO environment

The performance management abilities of the ETSI NFV MANO framework allow for the direct collection of all resource-related defined metrics. The mechanism called Performance Management Job (cf. [42], [43]) enables the creation of measurements of specified parameters upon the OSS/BSS or EM

request. After the creation of relevant jobs, the OSS/BSS requests MANO (directly or via EMs) to set thresholds on these measurements and then only the threshold-crossing notifications are sent by MANO entities to the requester.

To calculate the MANO KPIs, the information about the resource allocation, usage and the occurrence of certain operations with their completion time is required that can be categorized into:

- Information related to resource consumption: computing, memory, storage and connectivity resources allocated and consumed.
- Information about the execution time of selected NFVO operations, which are driven by the OSS/BSS.
- Information about execution of VNFM operations.

The collection of information required for KPI calculation is done by the OSS/BSS, which has to interact with other components of the MANO architecture.

- The OSS/BSS can directly use the Os-Ma-nfvo reference point [42] for the purpose of Network Service LCM, Performance Management, Fault Management and the NFVI Capacity Information (querying and notifications about underlying infrastructure capacity and its shortage).
- VIM can expose the information about the underlying NFVI at the reference points Vi-Vnfm [45] and Or-Vi [46] to NFVO.

Hence, the OSS/BSS is able either to determine the life cycle operations performance based on a request-response time interval or subscribe the run-time performance/fault/capacity indicators. While the information exchange between NFVO and OSS/BSS is at the level of Network Service Instance, the individual VNF's Element Manager (EM) is partially able to exchange similar information with the VNFM at the level of its VNF/VNFCs via the reference point Ve-Vnfm-em [43] and to share further the information with its own OSS.

Mechanisms of premium importance for KPIs calculation has also been described in [40]:

- VIM uses reference points Vi-Vnfm and Or-Vi to report NFVI-related performance indicators to VNFM and VNFO, respectively. The performance metrics include mean/peak usage of virtual CPU, memory, disk, and virtual storage, number of incoming/outgoing bytes/packets on the virtual computer (split per virtual interface) or virtual network (split per virtual port);
- VNFM maps the above-mentioned information from VIM to specific VNFs/VNFCs and exposes the performance measurements at reference points Ve-Vnfm-em (for VNFs/VNFCs) and Or-Vnfm (for VNFs only). These are VNF/VNFC-specific mean/peak usages of virtual CPU, memory, disk and virtual storage, numbers of incoming/outgoing bytes/packets at VNF internal/external connection points;
- The performance measurements produced by NFVO can be transferred to OSS/BSS via the reference point Os-Ma-Nfvo. They include numbers of incoming/outgoing bytes/packets at Network Service border interfaces.

Other essential features have been presented in [44], where charging-related capabilities have been described. In general, MANO enables charging of two categories: Usage Events and Management and Orchestration Events. Both types of events can be used to calculate KPIs.

The presented capabilities of MANO enable data collection by OSS/BSS, necessary for network slicing KPIs calculation and correlation. These data, processed mainly by VNFM, can be obtained via several paths by the direct interaction of OSS/BSS with NFVO or through EM. The EM of VNF can also be

implemented in that way that it will calculate VNF-level KPIs directly. It is of particular importance for the in-slice management concept described earlier in this Section.

In some implementations, the OSS/BSS can interact with NFVI directly in order to obtain knowledge about resource allocation and consumption. The NFVI has additional tools that provide such information. The ways, in which the required information is collected by OSS/BSS, are partly implementation-dependent.

3. MEC CAPABILITIES FOR THE SUPPORT OF 5G!DRONES TRIALS

Edge computing comes with the promise of low latency, which is critical for the delay-sensitive components that are involved in many of the 5G!Drones use case (UC) scenarios. This part of the deliverable relies on the activities conducted in task T3.2 "MEC capabilities for the support of 5G!Drones trials", and highlights the MEC features and functions needed for 5G!Drones and UAV vertical industry. In this section we highlight the different MEC features needed in 5G!Drones (including security) and in general, by the UAV vertical to safely fly drones on top of a MEC infrastructure in 5G. This section includes description of MEC concept evolution by the integration with NFV and 5G network slicing as well as description of concepts of:

- UAV mobility by introducing a mobility management component, which will ensure that UAV service components that are deployed at the edge are appropriately migrated across edge clouds following UAV mobility in order to maintain the latency constraints of the respective slices.
- UAV flight planning considering MEC constraints.
- Slicing support for MEC to ensure slice awareness so that the appropriate level of (performance and other) isolation among coexisting slices is also enforced at the MEC level.

3.1. ETSI MEC architecture

Since its creation in 2013, the ETSI ISG MEC group has been working on the development of standardisation activities around MEC. The first released document of the group covers the reference architecture [47], which aims to specify the different necessary components; a high-level representation of the architecture is shown in Fig. 12.

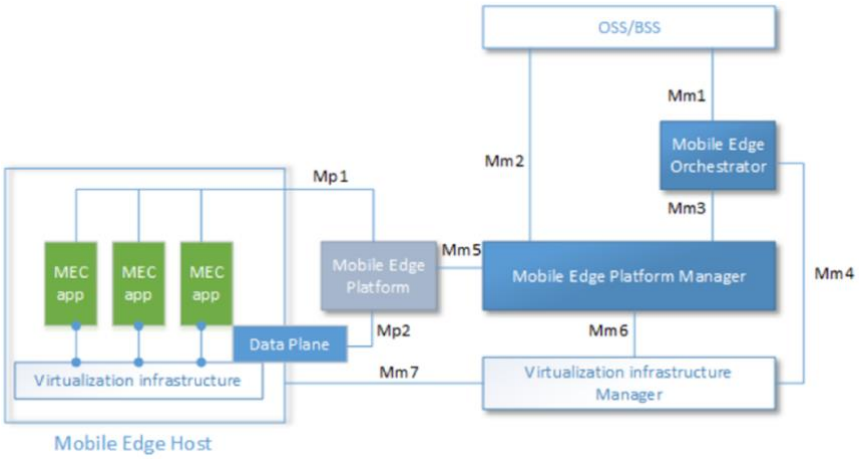


Fig. 12: A high-level view of the MEC architecture [47]

MEC introduces three main entities:

- The MEC host, which provides the virtualisation environment to run MEC applications, while interacting with mobile network entities via the MEC platform (MEP) to provide MEC services and data offload to MEC applications. Two MEC hosts can communicate via the Mp3 interface aiming at managing user mobility via the migration of MEC applications among MEC hosts.
- The MEC platform (MEP), which acts as an interface between the mobile network and the MEC applications. It has an interface (Mp1) with MEC applications so that the latter can expose and consume MEC services, and another interface (Mp2) to interact with the mobile network. The latter is used to obtain statistics from the RAN on UEs and eNBs, e.g. in order to provide the Radio

Network Information Service (RNIS) and the Location Service and to appropriately steer user-plane traffic to MEC applications.

- MEC applications that run on top of a virtualised platform.

Another concept introduced by ETSI MEC is the MEC service, which is either a service provided natively by the MEC platform, such as the RNIS and traffic control, or a service provided by a MEC application, e.g. video transcoding. MEC services provided by third-party MEC applications should be registered with the MEP and made available over the Mp1 reference point. Once registered, a service may be discovered and consumed by other MEC applications. Regarding the management and orchestration plane, ETSI MEC introduced the Mobile Edge Orchestrator (MEO), which is in charge of the life cycle of MEC applications (instantiation, orchestration and management), and acts as the interface between the MEC host and the Operation/Business Support System (OSS/BSS).

Several interfaces have been specified for the MEC management plane. The Mm1 interface is used to communicate with the OSS/BSS, allowing the latter to onboard MEC application packages and request application instantiation and termination. The MEO uses the Mm3 reference point to interface with the MEP Manager (MEPM) for application LCM and configuration, and Mm4 to manage application images at the edge Virtual Infrastructure Manager (VIM), which is in charge of launching application instances on the MEC host. The MEPM element is in charge of the LCM of the deployed MEC applications, and the configuration of the MEC platform, via the Mm5 interface. This includes MEC application authorization, specification of the type of traffic that needs to be offloaded to a MEC application, Domain Name Service (DNS) management, etc.

The Mm6 interface is used by the MEPM to obtain information on the virtual resources used by a MEC application from the VIM and implement their LCM. Such information can be passed on via Mm3 to the MEO to check the MEC application resource status, and, if deemed appropriate, add more resources to it. This information is also exposed to the OSS/BSS over the Mm2 reference point.

It should be noted that MEC allows the migration of MEC applications among MEC hosts using the mp3 interfaces, i.e. the mp3 is used to implement the migration processes.

As defined in ETSI MEC, a MEC application's LCM is handled by the MEO. If vertical wishes to deploy a network slice at the MEC, the first step is to onboard the MEC application image (i.e. VM or container image) at the MEO catalogue. The onboarding process consists of providing metadata on the MEC application and the location of the application image. These metadata are described in a specific format, which is known as the Application Descriptor (AppD) [48]. It includes information on the location of the virtual image, security information, and other fields related to the requirements of the MEC application, such as its maximum tolerated latency, traffic steering rules, and required MEC services. Since the MEC application image is on-boarded, the MEO creates an identifier for the MEC application, which is communicated to the vertical, and used by the latter to instantiate the MEC application. Following the request of the vertical to instantiate the MEC application, the MEO uses the AppD, and more specifically the three fields described earlier, to select the appropriate MEP that satisfies the combined requirements, and requests the deployment of the MEC application to the VIM (at the selected MEC host). Once the MEC application is up, the next step consists in allowing the latter to discover the MEP resources over the Mp1 reference point.

The MEC framework defines special service APIs exposed by MEP to MEC applications: Radio Network Information – RNIS [49] (PLMN information, E-RAB information, S1 Bearer information and L2 measurements), Location [50] (zonal presence and terminal location, including information about distance from a specific location or between terminals), UE Identity [51] and Bandwidth Management [52] (management of bandwidth on per application session basis). These services shall be provided via the Mp2 reference point, which will need special enablers within 5GC-CP. It has to be noted that the

ETSI MEC framework is currently defined for integration with the 4G network (it is especially reflected in RNIS data model, which is not radio technology-agnostic). Therefore, specifications of these APIs have to be updated, and corresponding 5GS-side enablers have to be available. This mainly applies to mechanisms provided by NEF, Network Data Analytics Function (NWDAF) [2] and Location Services (LCS) [53]. It is particularly important to ensure the availability of RAN related information. Although the 5G RAN physical layer measurements at UE have been specified [54], the mechanisms similar to 3G/4G radio measurements collection (MDT, cf. [55]) for further processing and use are still undefined, but they are in the scope of Release 17.

Additionally, it is hereby proposed to define the special MEP-facing gateway function located in 5GC-CP to provide a single and standardised interface for MEP and ensure smooth and optimized interaction (especially for avoiding excessive signalling exchange within 5GC-CP). Such initiative needs bilateral cooperation of the 3GPP and ETSI MEC group.

3.2. MEC in NFV

As described in section 3.1, the MEC architecture is defined to run independently from the NFV environment. However, the advantage brought by NFV, and aiming to integrate and run all MEC entities in a common NFV environment, has led the MEC ETSI group to update the reference architecture. The proposed document [56] updates the reference architecture, as shown in Fig. 13.

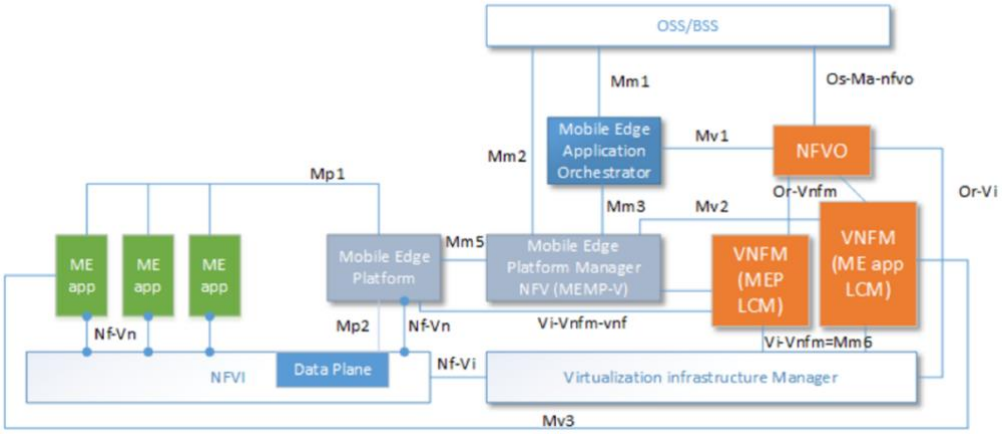


Fig. 13: An updated version of the MEC architecture featuring MEC in NFV [56]

As it could be noticed, the MEC platform and the MEPM are run as a VNF. The MEO became the MEAO (Mobile Edge Application Orchestrator); it keeps the main functions described before, except that it should use the NFVO to instantiate the virtual resources for the MEC applications as well as for the MEP. Consequently, all the process of instantiation and management of resources will follow the NFV well-defined interfaces. By doing so, the edge resources can be seen as classical computation and storage resources and managed by the same VIM software. Note that Table 1 summarizes the difference between the MEO and MEAO in terms of functionality.

In addition to MEC applications, the VNF Manager (VNFM) is also in charge of the LCM of MEP and MEPM. Finally, another important extension is the appearance of new interfaces (Mv1, Mv2, and Mv3), which allow communication between MEC and NFV components, in addition to the usage of the interfaces defined by the ETSI NFV.

Table 1. Differences between MEO and MEAO

Function	MEO	MEAO
Maintaining an overview of the MEC, available resources, available MEC hosts, topology	Yes	Yes
Selecting appropriate MEC host based on constraints (latency, available resource and available services)	Yes	Yes
Triggering application instantiation and termination	Yes	Yes (Via the NFVO)
Triggering application relocation as needed when supported (migration due to mobility)	Yes	Yes

3.3. MEC in 5G

The new 5G reference architecture introduces several NFs. The most prominent are Access and Mobility Management Function (AMF), Session Management Function (SMF), User Plane Function (UPF), User Data Management (UDM), Network Slice Selection Function (NSSF), Network capability Exposure Function (NEF), Policy Control Function (PCF), and Application Function (AF). All the NFs expose APIs to provide one or more services to other NFs, following the producer-consumer concept. Regarding the support for Network Slicing, we notice the appearance of the NSSF, which allows the RAN to select the appropriate AMF (slice-specific or common to all slices), when a UE indicates in the first attach request its S-NSSAI.

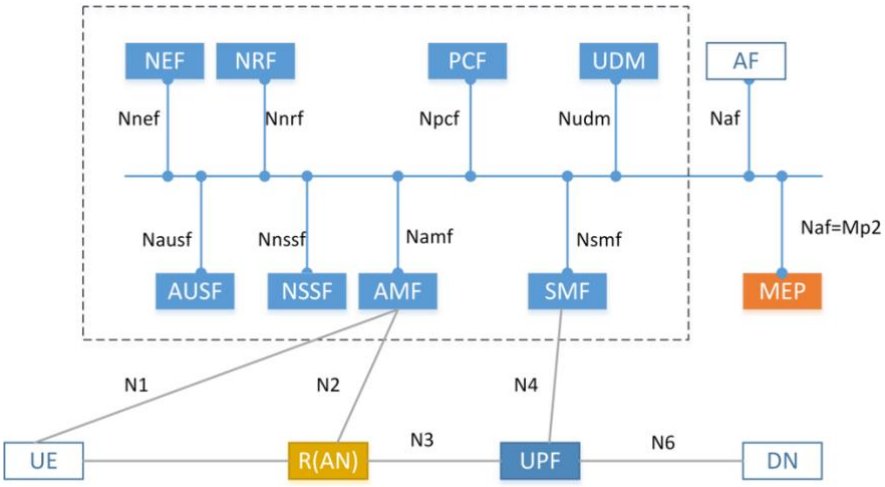


Fig. 14: MEC vision in 5G

In this work, we focus on user-plane functions (SMF, PCF, and UPF), as MEC requires the definition of traffic policies to redirect traffic to the appropriate MEC applications. More details on the other 5G functions can be found in [2]. The UPF is the function in charge of routing the UP traffic to the appropriate Data Network (DN). It gets its configuration from the SMF. The latter is considered as one of the key elements for user-plane traffic management. Among the various functions of the SMF, such as IP address allocation and management and session management, is the control of the UPF by configuring traffic rules. The SMF exposes service operations to allow another function or 5G AF to use policy and traffic rules to reconfigure the UPF, either directly via PCF or using NEF as an additional proxy, depending on the operator deployment (e.g. based on security architecture).

In the 5G architecture, the MEP will be integrated as a 5G AF (Fig. 14), trusted or not, depending on the use-case; this will be discussed later. The MEP requests traffic redirection for a MEC application as per the request of the MEAO via the MEPM. Therefore, if MEP is a trusted 5G AF, it can directly use the PCF to generate a policy to offload traffic towards the MEC application. If it is not considered as a trusted 5G AF, it uses the NEF to access the SMF, via its traffic filter policy exposed API and requests the traffic redirection.

3.3.1. Integration of MEC with 5G slicing

An overview of the current status of the standards in slicing and edge computing reveals that slicing support for MEC is still at a very early stage. Given that 5G!Drones makes heavy use of slicing in conjunction with edge computing, it is necessary to extend current MEC implementations for slice awareness so that the appropriate level of (performance and other) isolation among coexisting slices is also enforced at the MEC level. Stemming from the facts that (i) 3GPP has released a new architecture model to integrate NS in 5G, and a new framework to manage NS, and (ii) the ETSI MEC group has proposed a solution to integrate MEC in NFV, there is a need to update the current MEC architecture to comply with these evolutions, aiming at supporting NS at the MEC level (i.e. slicing the MEC). We distinguish two models for the support of Network Slicing in MEC. The first model assumes that the MEP is already deployed at the edge NFVI and is shared among the slices; we term it the *multi-tenancy* model. In the second model, the MEP is deployed inside the slice. This is what we call *in-slice* deployment. For both models, we assume that the MEP is deployed as a VNF. Both the MEP and MEC applications are described using a VNF Descriptor (VNFD) and Application Descriptors (AppDs), respectively.

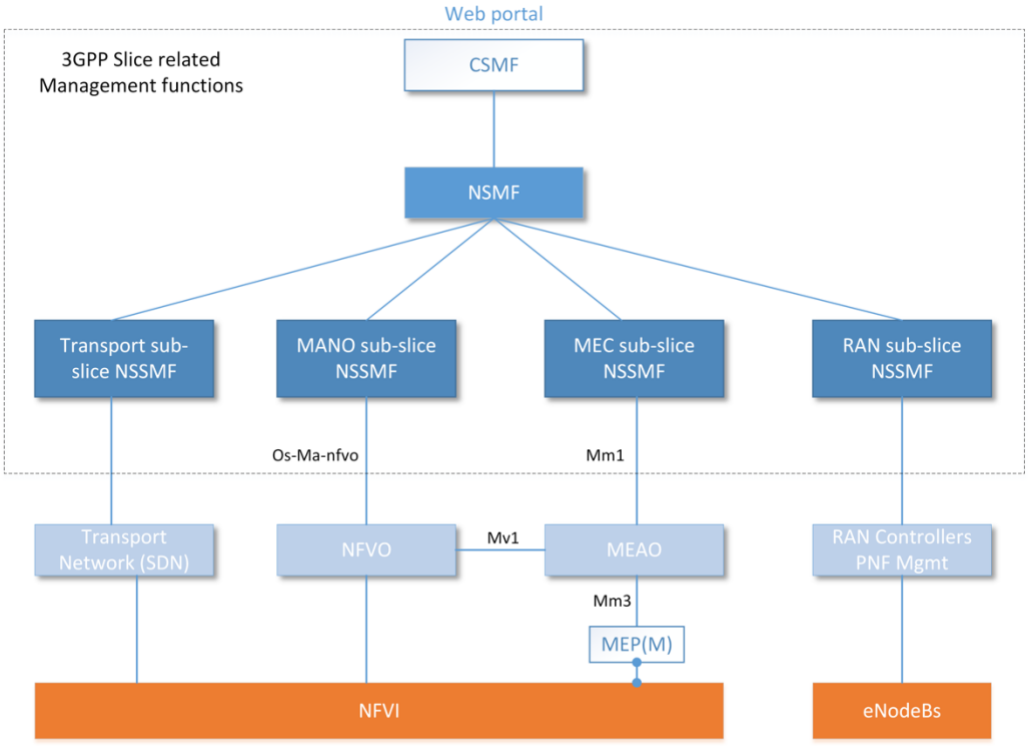


Fig. 15: The proposed network slicing orchestration/management architecture, including MEC, in a 5G environment

The VNFD and AppD describe the necessary information required by the NFV Orchestrator (NFVO) and VIM to deploy instances of virtual applications, either at centralised clouds or the edge. AppD is specific to MEC applications. It contains specific fields related to MEC, such as traffic steering rules and MEC services required by the application. Note that we consider the MEPM as the Element Manager

(EM) of the MEP. CSMF shows the global picture highlighting the envisioned network slicing orchestration/management architecture as proposed by 3GPP and featuring MEC slicing. In terms of interfaces, we mainly highlight those needed to orchestrate and manage core and edge virtual applications. The RAN controller is the element that provides a northbound control interface to manage eNBs, while using a southbound protocol, such as FlexRAN [57], in order to remotely configure eNBs (e.g. to associate to a new AMF of a slice) or to obtain RAN-level information, such as UE statistics, which can be used by the operator or exposed to interested applications over the RNIS MEC API.

We assume that a vertical first accesses a front-end interface (such as a web portal) to request the creation of a network slice, using the Network Slice Template (NST) made available by the CSMF. The NST could be extended according to the vertical needs and by integrating network functions displayed by the CSMF through its network functions store or catalogue (i.e. add more MEC applications). The CSMF forwards the NST to request the creation of an end-to-end network slice composed of several sub-slices that span the RAN, CN, MEC, and TN. The NSMF organises the NST into sections corresponding to each sub-slice. The Management and Orchestration (MANO) NSSMF component covers the CN functions and VNFs that need to be deployed over the cloud. All the network functions that need to be deployed over MEC should be managed by the MEC NSSMF. The NSSMF accepts as input a Network Service Descriptor (NSD) [58] that contains VNFDs as well as AppDs. The NSMF requests the creation of each sub-slice to the corresponding NSSMF, as illustrated in Fig. 15. The RAN NSSMF is in charge of updating the configuration of the RAN, via a RAN controller that interacts with the involved eNBs (PNF) indicated in the NST. The NSSMF in charge of CN and VNF instantiation requests the instantiation of the NSD to the NFVO using the Os-Ma-nfvo interface [42]. The MEC NSSMF interacts with the MEAO by providing the AppDs of the applications that need to be deployed at the edge NFVI. The MEAO will use the same NFVO (as specified in [56]) to request the creation of the AppD instance at the selected edge NFVI. Among the available edge NFVIs, the MEAO selects the appropriate one for the instantiation of a MEC application, according to its internal placement algorithm that may consider different criteria, such as latency and service availability [59]. To recall the AppD includes important information related to the MEC application to be deployed, such as *appLatency*, *appTrafficRule*, *appRequiredService*.

Once the application is instantiated, the MEAO is informed of the MEC application's IP address, which it communicates to the MEC platform along with parameters such as specific traffic filters to enforce traffic steering. The last sub-slice is about the TN part, where we assume that the NSSMF managing it interacts with Software Defined Networking (SDN) controllers to isolate and forward NS traffic to the Internet. Once each sub-slice is created, the NSMF is in charge of stitching them together to build the end-to-end slice. The stitching process consists of interconnecting the different sub-slices using a sub-slice border API, as described in [60].

The MEP can be implemented as multi- or single-tenant MEP (see Fig. 16).

Multi-tenant MEP

In the case of MEP multi-tenancy, the MEP and UPF are already deployed. The MEP is already aware of the IP addresses and interface endpoints of the NEF or PCF for traffic redirection, as well as those of the RAN controller, from which it can gather the necessary RAN-level data to provide MEC services, such as the RNIS and the Location Service. Once the MEC application is deployed by the NFVO, the latter informs the MEAO about the successful instantiation of the MEC application, along with its IP address. The MEAO then, via Mm3, requests the MEP to enforce traffic redirection rules as indicated in the AppD. Based on the description presented in section 3.1, the MEP, via the PCF's API, requests the redirection of specific traffic (via a traffic policy) toward the newly created MEC application. Here, the MEP uses the PCF, as it is considered a 5G AF: the MEP has been deployed by the network operator as a common 5G AF for all slices.

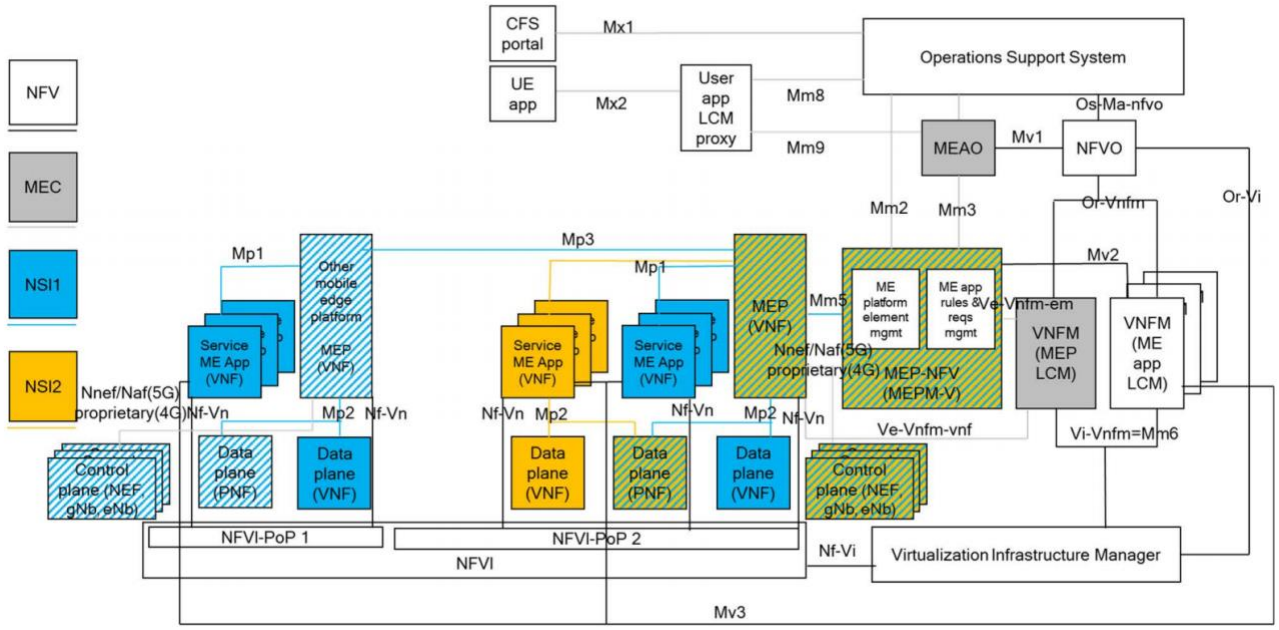


Fig. 16: Example of MEC in NFV supporting slicing [61]

Single-tenant (In-slice) MEP

In this case, the MEP has to be deployed along with the MEC application at the edge NFVI. Unlike the multi-tenancy model, here the MEAO requests the instantiation of both the MEP and MEC applications at the same time. The NFVO deploys both and ensures that there is a virtual link between them. As in the previous case, the NFVO acknowledges the creation of the MEP and MEC application instances and indicates their IP addresses.

Here, we differentiate between two cases: (i) all the CN elements (including the UPF) are deployed inside the slice; (ii) the UPF is already deployed. In the first situation, the UPF is deployed also at the edge (for the sake of performance), and the MEP can implement traffic redirection using the internal PCF of the network slice. For the second scenario, the MEP has to discover the NEF of the operator, if the MEP is not considered as a trusted 5G AF. To solve this, we propose that the DNS running at the edge NFVI may help in this direction: Once instantiated, the MEP sends a DNS request to discover the NEF's IP address and communicates with the latter to apply traffic redirection rules.

Regarding the needed access to the eNBs in order to provide MEC services (e.g. RNIS, Location Service), we propose to use the concept of zones, as introduced in [50]. A zone indicates an area covered by a group of eNBs associated with a MEC host. These eNBs are assumed to be managed by a single RAN controller. For both scenarios, we propose that the MEP uses DNS to discover the RAN controller that corresponds to the zone where it is instantiated, which in turn allows the MEP to retrieve RAN-level information from all eNBs of the zone.

3.3.2. A new proposal of MEC and network slicing integration

The proposed new MEC-enabled 5G network slicing architecture is based on the following principles:

- MEC services, similarly as NSIs, have limited geographic scope and are focused on a specific service – this is in line with the network slicing philosophy, which emphasises customisation of NSI to its service or a group of services with similar characteristics. In more complicated use cases, like UAV or V2X, the overall service uses several NSIs of a different type. Utilisation of MEC as a platform offers useful mechanisms to provide a specific service. Consequently, in the case of

network slicing, the number of MEC applications will be limited, and they will be defined during the slice creation. Therefore, the orchestration of MEC applications during the NSI run-time will be rather rare.

- Flexible architectural approach, adapted to NSI characteristics (complexity, longevity, critical deployment time, etc.), is required. As a result, the coexistence of various architectural variants can be expected.
- Implementation of MEC applications as a part of slice AP – the same NFVI is used by CP/UP, and no separated MEC orchestration domain is needed. Therefore, the orchestration of MEC applications belongs to slice-level orchestration activities.
- Tight integration on an equal basis of MEC APIs (RNIS, Localisation, etc.) with information obtainable from 5GC via NEF, to extend the amount of information available for slice creation and for the avoidance of duplication of 5G and MEC functions like Network Repository Function (NRF), etc.

Fig. 17 shows the proposed generalised architecture of MEC and 5G integration. All VNFs are implemented in the VNF space, using common NFVI managed by VIM (omitted in the picture for simplification). NFVI can be single- or multi-domain (cf. [62]). All VNFs have their EMs (symbolized by red dots) connected to OSS/BSS (red arrows). In the case of MEC applications, their management functions may be embedded in applications, externalised or non-existent. VNFs and their EMs are also connected to VNFM(s) (single- or multi-VNFM options are possible, cf. [62]), which are responsible for LCM of both MEC applications and other VNFs (VNFM* in Fig. 17). Even if the ETSI MEC framework assumes Ve-Vnfm-vnf variant (light) of MEC App–NFVO reference point, it may be potentially useful in specific cases to implement fully functional Ve-Vnfm-em variant, instead.

Orchestration of MEC is located at OSS/BSS together with the management of a 5G network and Network Slice (Subnet) Management Function – NS(S)MF. Therefore, all interactions with the ETSI NFV MANO stack are performed via one common OSS–NFVO interface. As MEAO and User app LCM proxy are functional modules of OSS/BSS, some ETSI MEC reference points are internalized. OSS/BSS opens both interfaces Mx1/Mx2 to the customer domain. MEP exposes platform's services to MEC applications (Mp1) and in case of 5GS-interacting ones, acts as a mediator to 5GC-CP via NEF (Mp2, considered as Naf at the 5GC-CP bus).

The described generalised architecture is valid both in case of 5G network with its own MEP/MEPM-V (Variant 1) and for MEP/MEPM-V sharing by multiple networks (Variant 2). In the case of Variant 1, the "VNF space" in Fig. 17 can be simply renamed to "5G network".

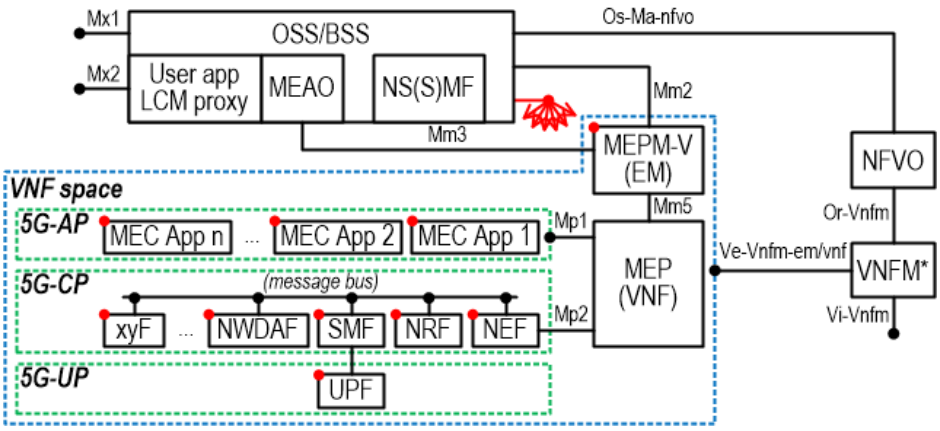


Fig. 17: General slicing architecture of MEC-enabled 5G network

As MEP/MEPM-V are dedicated, they can be a part of the template of the virtualised 5G network and share its life cycle. In case of Variant 2 (suitable rather for short-lived and simple slices), they will be external to 5G networks (now consisted of AP, CP and UP only). As the shared MEP is interfaced with CPs and APs of separate networks, it has to provide mechanisms for mutual isolation between these networks, i.e. their reciprocal unawareness and prevention of cross-exchange of information or unauthorised access to foreign 5GC-CP. The issue of protection of individual networks privacy is an additional factor for externalization of MEP towards all connected networks in Variant 2. Additionally, inter-App privacy should be ensured in both variants (e.g. awareness of users, their sessions metadata, etc.), but it can be provided by their own 5GC-CP. If network slicing is enabled (the case of multiple-NSI networks, providing services with different characteristics), both MEP/MEPM-V and MEAO have to be NSI-aware, i.e. recognize and distinguish NSIs, as it is required from all 5GC-CP entities (cf. [61]).

In geographically distributed architecturally complex communication networks, moving network functions of high granularity towards the edge have positive consequences for user traffic transport and performance but at the expense of the control and management planes. Centralised management of highly distributed networks is vastly inefficient, especially due to the necessity of transporting huge volumes of data needed for analysis, decision-making and execution of automated management processes.

The single-domain scalable MEC-enabled slicing architecture (intent-based management framework extended with MEC) is presented in Fig. 18. All VNFs of the slice have their own EEMs, as described in section 2.3. EEMs are connected to SM, to provide the slice management plane communication. MEP/MEPM-V belong to the SOS area because their role is in line with the SOS definition, especially the exposure of transparent mechanisms for slice VNFs interconnection. MEAO and User app LCM proxy are located in SM because it plays the role of slice OSS.

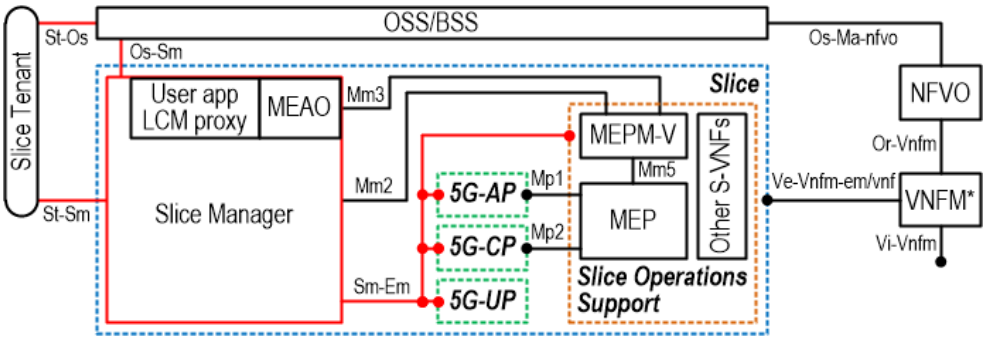


Fig. 18: MEC-enabled intent-based management architecture

The important task of SM is the proper routing of the MEC framework-related exchange. The Mm1 communication will be forwarded to the global OSS/BSS, which concentrates the exchange with NFV MANO. The Mx1/Mx2 reference point communication will be exposed through the St-Sm interface. Alternatively, it may be forwarded to the global OSS/BSS if the Slice Tenant prefers interactions that way (e.g. utilisation of multiple separate NSIs; the consolidated global view is then desired).

It has to be noted that the described ISM architecture also supports the multi-domain sliced networks. The global OSS/BSS contains the Multi-Domain Management and Orchestration Support functions composed of Multi-Domain Slice Configurator (MDSC) and Multi-Domain Orchestrator ("Umbrella NFVO", cf. [62]). MDSC, during the slice run-time, keeps monitoring of the end-to-end slice and coordinates its reconfiguration, also taking care of MEC-related activities. It is responsible for the proper configuration of local SOS entities for inter-domain operations.

To enable operations in a multi-domain environment, it is essential to provide means of horizontal end-to-end slice stitching, i.e. concatenation of sub-slices from different domains. Inter-Domain Operations Support (IDOS), a functional part of SOS, is defined for this purpose. IDOS acts as an inter-slice gateway, implementing information exchange between neighbouring domains, i.e. exposure of domain abstracted view and support for inter-domain communication (relevant protocols, transcoding, mediation, etc.). In the MEC-enabled intent-based management architecture, the Mp3 reference point control information transfer between MEPs shall be carried out via IDOS.

3.4. MEC security

In this section, we shed light on security requirements and mitigations in MEC. For each security requirement, we provide risks, way of mitigation and impact on 5G!Drones.

3.4.1. Physical security

MEC nodes are more exposed – indeed directly exposed - to physical attacks than the CN since they are closer to the end-users and could be considered with the RAN as the first line of defence of the whole 5G network. In some cases, MEC facilities are not fully owned or controlled by the 5G network operator, sometimes event part of the customer's or partner's on-premises equipment, therefore the security management very likely weaker and more vulnerable. In this case, it is not only exposed to outside attackers but also inside attackers from third party organizations. It is critical that MEC facility owners have hardware hardening measures in place for all servers and perimeter security controls, including physical access control, as well as physical-layer and link-layer network access control mechanisms such as IEEE 802.1x. Such security controls should be combined with well-defined access control policies to guarantee that only the users with proper administration clearances may access the facilities and servers. Besides, the MEC facility owner should have physical surveillance and monitoring system up and running (CCTV, etc.) to monitor and keep track of who accessed what where and when. Last but not least, the facility should be audited regularly to make sure the physical security and operations are maintained in good condition and compliance with the security policy. Although such physical security enablers are stretching the enablers in primary scope of 5G!Drones, they would become critical in near future regarding a real-life deployment scenario.

3.4.2. MEC infrastructure security

At the level of the MEC infrastructure, we shall consider security risks on:

- Hosts;
- Virtualisation infrastructure;
- Cloud infrastructure, esp. administration APIs or any kind interfaces (e.g. SSH, RDP, etc.).

In particular, we should make sure that no bad (malicious) VNF or MEC App image is loaded, that there is no third-party tampering on such images, on the runtime processes, and on the data flows within the infrastructure, i.e. no violation of confidentiality, integrity and availability (DoS). We should emphasize the risk of weak isolation in virtualised/cloud infrastructures, between VMs or containers, allowing one tenant to see others' assets or data; or an application/user in one tenant's security domain to see or leak data to another security domain of that tenant. Mitigating such risks starts with a hardware root of trust (TPM, HSM, SE, etc.) and mechanisms of Secure Boot or image signature validation to make sure that genuine software images (OS, container, VNF, MEC app) are loaded tamper-proof and certified by a trusted authority. General host/OS/virtualisation/container infrastructure hardening measures must be enforced (see NIST recommendations and Cloud Security Alliance). Use of TEE (Trusted Execution Environment) should be used as well for protecting software execution of sensitive processes (e.g. key management, encryption). Security controls at the network layer must be enforced as well: firewalling,

IDS, IPS, VPN, IEEE 802.1x, etc. At the application layer, especially for infrastructure management data flows, data confidentiality, integrity and availability should be protected at rest (encrypted storage and/or hardware security modules), in-transit (see previous network security controls and application-layer encryption/signature) from unauthorised access.

In this regard, 5G!Drones Enablers such as PKI, Identity and Access Management, as well as Policy Enforcement Proxy Enablers can provide data-in-transit security and protect MEC infrastructure API against such security risks.

3.4.3. MEP security

Similarly, in the MEP, there are security risks involved with VNFD and AppD if they are tampered with, also the potential vulnerability of the platform to DoS, and multi-tenancy/isolation issues. API security controls should be implemented (authentication, authorisation, input validation, data security, e.g. with TLS). Security (confidentiality and integrity) of data flows in MEP-MEPM communications must be guaranteed by ingress and egress gateways or lower-level VPNs. The MEP should also enforce tenant/domain/resource isolation, using granular encryption, or virtual isolation techniques, depending on the level of sensitivity (e.g. data classification) to prevent unauthorised access or data leaks.

In this regard, 5G!Drones Enablers such as Identity and Access Management, as well as Policy Enforcement Proxy Enablers, can provide data-in-transit security and protect MEP APIs against such security risks.

3.4.4. Service ME App security

As for the ME App, there are major risks of unauthorised operations on the application life cycle (CRUD), e.g. unauthorised application creation, removal, or update; and third-party tampering (application-to-application, user-to-application access); and excessive use of resources by an application, causing DoS to others. Such risks may be mitigated with proper application and tenant isolation, and security controls on the application LCM APIs (access control). Also, resource restriction (quotas) is critical to prevent DoS of a shortage of resources for other applications/tenants.

Again, 5G!Drones enablers such as PKI, Identity and Access Management, as well as Policy Enforcement Proxy Enablers, can provide data-in-transit security and protect APIs to mitigate such security risks.

3.4.5. User plane data security

The MEC UP is an obvious attack path to the CN. Also, communication between MECs involves many intermediaries, including the CN, therefore particularly exposed to third-party tampering or eavesdropping.

End-to-end (e.g. MEC-to-MEC or MEC-to-end-user) security, especially confidentiality and integrity, is critical and must be enforced for sensitive user communications through end-to-end encryption and signature. For different security domains (slices/tenants/sub-domains), the key management should allow encryption with different key materials, algorithms and protocols, for each security domain to guarantee their isolation. Also, proper key and certificate distribution, as well as identity federation protocols, should allow multiple domains and multiple tenants to establish trust with each other to achieve end-to-end trust.

5G!Drones enablers such as PKI, Identity and Access Management, as well as Policy Enforcement Proxy Enablers, can provide end-to-end data-in-transit security to mitigate such security risks.

3.4.6. MEC MANO (MEAO and MEPM) security

A major risk to MEAO and MEPM APIs is an abuse of privileged access. Privileged access control with strong authentication (multi-factor), and management API security (e.g. input validation, especially App image validation) is a must-have, as well as continuous vulnerability scan of App images in the image repository.

3.5. 5G-MEC usage for UAV services

3.5.1. Application mobility in demanding use cases

In [63] it has been demonstrated that the total time needed for MEC application deployment can vary from ~60 s (application instantiation only) to ~180 s (onboarding and instantiation) or even ~440 s (full onboarding and instantiation of both MEP and application). In high-mobility use cases (speeds of several kilometres per minute, which is typical for drone, railway or automotive ones) MEC applications cannot just follow the UE, but they must overtake it. Utilization of standard location tracking mechanisms, even with additional prediction, will not be sufficient. Therefore, integration with UAV traffic management system, which is aware of flight plan, with UE context-awareness mechanisms driven by mechanisms of Artificial Intelligence and Geographic Information Systems to deduce, e.g. following a motorway or railway line, or with onboard navigation, aware of the desired route, can be utilised.

3.5.2. Service continuity in roaming

Special concern should be dedicated to roaming cases. Maintaining service continuity requires replication of its architecture at Visited PLMN (VPLMN) and an acceleration of the re-registration process during the operator change. This issue is partially discussed in [64]. In case of MEC-enabled service architecture, the entire NSI, along with the MEC App residing in the AP, must be instantiated on VPLMN resources in a Local Break-Out (LBO) mode. To some extent, service architectures (i.e. NSI templates) standardization together with MEC applications porting mechanism can be a solution, but a general mechanism for any NSI portability will be needed.

3.5.3. Availability of 5G enablers for MEC

Majority of R&D projects are based on popular 5GS implementations, such as OpenAirInterface, Open5GCore or free5GC. However, these solutions implement fundamental functionalities of the 3GPP 5G architecture, but unfortunately NEF, NWDAF or LCS are missing there. Even handover support can be somewhat problematic. Additionally, whenever not-UE-based positioning is required, the Network-Assisted Positioning Procedure shall be used, which has to be supported by gNB (positioning based on RAN measurements, cf. [65]). Individual efforts on implementation of these mechanisms or an initiative on public-domain tools are needed. The list, review and status of open source tools for 5G (3GPP Release 15) can be found in [66].

3.5.4. Service mobility management for UAV

UAVs can benefit from the latest advances in edge cloud technologies, such as Multi-Access Edge Computing (MEC). By leveraging the benefits of MEC at the network edge (i.e. access points and base stations), a dramatic decrease in the communication latency can be induced between the edge-hosted applications and the connected UAVs. For instance, MEC can be used to host UAVs control services or to offload intensive computation from UAVs (e.g. image processing). Also, it is expected to use the computing power on the base station to run an application “near” UAVs. Application examples are CARS (Common Altitude Reference System) conversion systems, DTM (Data Terrain Model)/DSM (Data Surface Model) distribution.

Unlike centralised cloud data-centres, resources at edge hosts are limited and hence smart exploitation is encouraged. In the general case, where edge cloud is used to host traditional users' applications (e.g. video streaming application), this is mainly achieved through optimal service placement and replication, where the aim is to find the optimal distribution of a given service over a set of edge hosts in such a way that this service can be available for the maximum number of mobile user with a sustainable Quality-of-Experience (QoE) [67]. However, when users are characterised with high mobility and when the mobility patterns are not known, service replication may result in over-usage of the edge resources due to having the same service instantiated over a large number of edge nodes. Another alternative to service replication is service relocation, in which the services are moved from one edge host to another. As a mobile device (e.g. UAV) connected to a mobile network moves around within the network, it can result in the device connecting to the network entity associated to a different edge host from the serving host. Consequently, there is a necessity of relocating the application instance and/or user context associated with the device to a new edge host to continue offering the best performance of service [68]. One of the main challenging steps of service relocation is "Relocation Initiation", at which the decision of relocating an application between two hosts is made [69]. Indeed, the relocation process takes time to finish; thus, the relocation must be triggered before that the mobile device changes the network entity associated with original edge host (e.g. before the handover) and end only when the mobile device has effectively associated with the new network entity corresponding to the new edge host. The second challenge in the initiation phase is to identify the target edge host, which is not obvious when the mobility pattern of the mobile device is unknown.

Meanwhile, Application mobility in MEC is the capability of resuming the service to the UE once the UE's context and/or application instance is transferred from one MEC host to another MEC host aiming to continue the service seamlessly.

Traffic path updating is the key point for supporting mobility in a 5G MEC system. There are two scenarios of mobility support, one is for the intra-operator scenario, and the other one is for the inter-operator scenario. In the 5G!Drones project we address the aforementioned challenges for service relocation by harnessing the prior knowledge of the mobility pattern of the UAVs. Indeed, in the 5G!Drones project and mostly in the commercial field for automated drone flights since the deployment of the UTM system in the host country, UAVs will perform their missions according to approved flight plans reported to the UAVs Traffic Management (UTM) system. A flight plan consists of a set of waypoints (i.e. GPS coordinates) that describe the mobility plan of the UAV. Hence, it is possible to predict the set of access points that are candidates for serving the UAV during its flight. Moreover, the telemetry data collected from the UAVs during their missions such as the GPS location and the speed can be used to decide on the relocation trigger time. The User Equipment (UE), whether handsets or modems on UAVs, are usually mobile, and hence application mobility is one of the key features in systems where serving applications are hosted in MECs.

3.5.5. Service migration of a video application deployed in MEC

Service migration in MEC is one aspect of mobility management. As MEC brings the services close to the end-users, at the same time brings new implementation challenges with it. For instance, with user mobility, the limited coverage of an edge server may cause the connectivity issue leading to an interruption of the service. To address such kind of issues, service migration has been proposed to transfer the user's active service from the current Edge server to the destination Edge server, which is closer to the UE. However, although service migration seems very promising in the MEC system, many challenges have to overcome to make it functional. Figuring out the exact moment of migration, the migration process itself, and method how to select the destination Edge server are the main focusing points in the development of service migration, which in turn guarantees interruption less service for the customer. Regarding this topic, an experiment has been carried out at the 5GTN facility, where the aim

was to test the trade-off between resource usage and latency when there is an active application that needs to migrate from one Edge server to another. More specifically, an experiment was executed to minimize the use of virtual resources while enabling service migration while maintaining a low latency transmission. Therefore, the communication between a video stream source and Edge servers has been established via 5GTN for that experiment. The tools used for the experimental setup as follows:

- OBS: Open Broadcaster Software, Video Recording and Streaming application;
- NGINX: Multimedia server on Edge Servers;
- RTMP: Real-Time Messaging Protocol;
- VLC: VideoLAN Client for watching video streams – Client.

In the experiment, both resource management and latency minimization to be optimised were tried. To achieve service migration, considering both the cases, several different setups, and methodologies were tried out. The most suitable setup is described below.

The setup (Fig. 19); follows the underneath sequence of operations:

- Configure Master NGINX to push streams to all other two NGINX servers;
- Start primary NGINX (NGINX 1) and establish a connection with client 1;
- Start secondary NGINX (NGINX 2) and establish a connection with client 2;
- Stop primary NGINX and kill client connections of primary NGINX.

According to the configuration, there are 3 NGINX apps running in the system. This architecture measures the delay of transition between NGINX 1 and 2. However, an issue was faced; NGINX Master configuration needs to be changed in order to change the destination of the sent stream. NGINX does not support configuration change while it is being used. It was assumed that the reason is RTMP protocol, which is based on Transmission Control Protocol (TCP), is used, and a handshake cannot be interrupted. In order to address this issue, NGINX can be started so that it can push streaming to two NGINX servers on the same time starting from time T1.

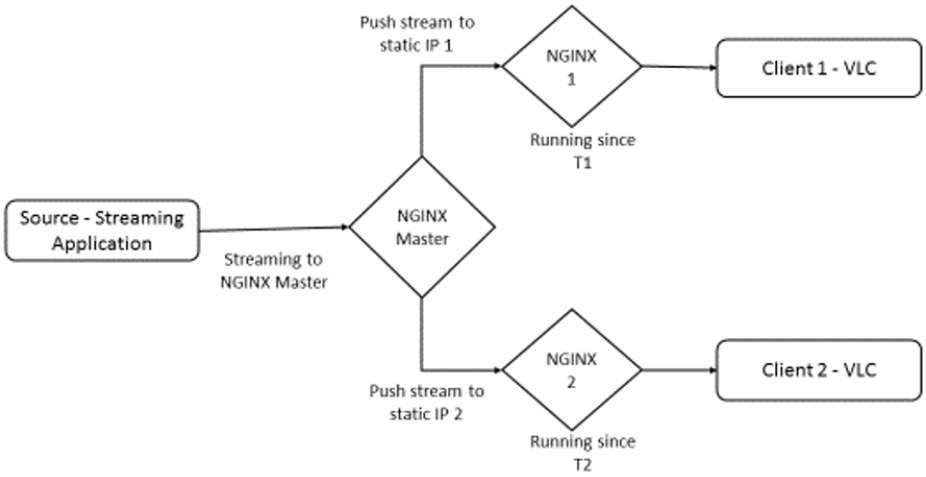


Fig. 19: Service migration while maintaining low latency transmission

Tests were executed like this: Start stream from a source, send frames to NGINX 1. At time T2, change the flow, send frames to NGINX 2. At time T2, let's assume the number of the last frame that NGINX 1 received is 20. Check when frame number 20 reaches NGINX 2. Thus, the difference is the delay.

This methodology might be one of the ways to measure the transition delay between two servers. It might be possible to include the MEC features in the system. Also, a new Virtualised Network Function (VNF) can be created, which will establish another connection in parallel, enabling to manage resources and achieve possible zero latency.

3.5.6. Follow Me Edge Cloud for UAVs control

One of the techniques that can be used for service relocation is live process migration, in which a hierarchy of processes (e.g. a running container) is detached from the kernel of the original host, transferred over the network to the target host, and finally reattached to the kernel of the new host. Live migration techniques can be applied to all kinds of processes, although, in our case, we consider only the hierarchy of processes that represent containers, which are used to run UAV applications in edge cloud hosts. During the live migration of a container instance from one host to another, the states of the processes running in the namespace of that container are serialized into compact files and transferred over the network while preserving the running state of the containerized applications and maintaining open network connections. The transferred states include CPU state, memory state, network state and disk state. One of the major concerns for live process migration algorithms is the migration of the memory states. In fact, the size of the memory state depends on the application running inside the container and can reach several gigabytes; this can cause several issues during the migration procedure, not only because the time required to transfer such amount of data but also due to the complexity of tracking the changed memory pages during the migration [70]. It has to be noted that live migration algorithms are evaluated based on two KPIs, the first one is the total migration time that represents the elapsed time between the start and the end of the migration process. Whereas, the second KPI is the downtime that represents the time interval, during which the migrated application is running neither on the source nor on the destination server. Herein, we consider live migration based on the "iterative pre-copy" algorithm, which is available out-of-the-box in the Checkpoint-restore in User space (CRIU) tool used for live containers migrations. In the iterative pre-copy migration, the memory state is copied to the target host over several rounds, during the first round, the whole memory is copied, after that and during each subsequent round, only the memory pages that were changed after finishing the last round are transferred to the target host.

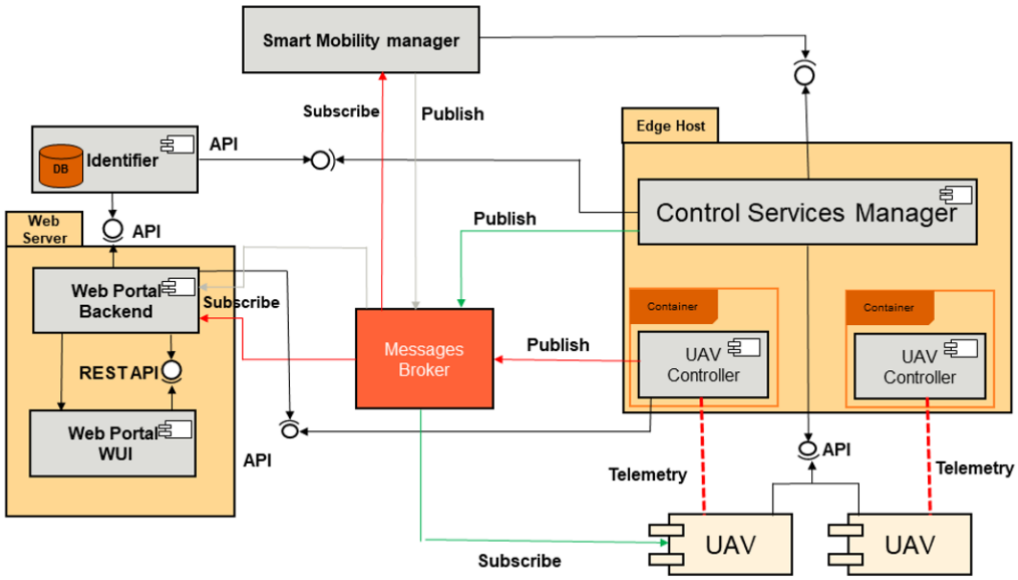


Fig. 20: Architecture of the Follow-me Edge Cloud platform

In the 5G!Drones project we propose a new platform that allows the relocation of UAVs control services hosted at the edge of the network, using live container migration and harnessing the prior knowledge of the mobility plans of UAVs to decide on the relocation time and target. Moreover, the proposed platform is intended to be independent of the mobile network in the sense that the entity, which is responsible for triggering service relocation relies only on the mobility plans of the UAVs and on the collected telemetry data instead of relying on specific network services such as the Radio Network Information Service (RNIS). Nevertheless, the UAVs control services to be relocated are assumed to be running as containers hosted in the edge cloud of the mobile network, with LBO to the edge hosts ensured by the mobile network UP.

As depicted in Fig. 20, the proposed platform is composed of the following building blocks:

- **Smart Mobility Manager (SMM):** Responsible for the initiation of relocation process. It collects and analyses the telemetry data published by the UAVs to decide on whether to trigger the relocation at a given timestep or not. Moreover, it uses the flight plans of UAVs to decide on the relocation target host.
- **Control Services Manager (CSM):** Manages the life cycle of the control services inside the edge host, it provides interfaces that allows the creation, deletion, and relocation of the edge hosted controllers. It is also responsible for configuring the traffic rules inside the edge host for communicating of the control service.
- **UAV Controller:** Runs as containerized application and exchanges control traffic with one UAV using a telemetry protocol. The UAV controller publishes the telemetry data (location, speed, battery level etc.) to the smart mobility manager and to the "Web Portal" backend, where it will be made available to the end users (i.e. the UAV operator).
- **UAV:** Establishes a telemetry connection to the UAV controller for exchanging C2 traffic, and a side link connection to the broker to receive notifications on any possible relocation. Indeed, when the SMM initiate the relocation process, it sends a request to the CSM to perform the relocation and a notification to the concerned UAV through the side-link to establishes a new telemetry connection towards the new edge host. The new connection is established before breaking the old telemetry connection to the original edge host (i.e. make-before-break) ensuring service continuity.
- **Message Broker:** Used for publishing and receiving the telemetry data, as well as for the make-before-break mechanism.
- **Web Server:** Hosts the WUI that allows the UAV operator to manage the edge hosts, the UAV controller, and the UAVs.
- **Identifier:** Used to authenticate the UAVs operators and the registered edge hosts.

3.5.7. Flight optimization for drones considering MEC

The deployment of MEC [70], [71] is assumed to be very distributed, i.e. several MEC servers will be deployed close to end-users. One MEC server will cover a set of base stations, hence covering a limited geographical area. However, as drones are highly mobile, they can go out of the coverage area of a MEC server, which may increase the latency; hence perturbing the C2 link (lead to a threat in the safety of the drone's flight). One solution usually employed to keep the benefit of the MEC, in terms of low latency connection, is to migrate the Drone Pilot application among MEC servers (known as service migration [72], by following drones' mobility. This will ensure that the Drone Pilot application is always hosted by the MEC server covering the area (set of base stations) where the Drones are located. Nevertheless, service migration has a negative effect. Indeed, during the migration, the service is down (called

downtime) for a few seconds. This may affect the C2 link negatively, and thus service migrations need to be limited to a minimum (only when deemed appropriate).

In 5G!Drones, we propose to reduce the service migration downtime by minimising the number of Drone Pilot migrations among MEC servers. To achieve this objective, we propose an algorithm to be used offline and during the mission planning phase, where the Drone Operator prepares the plan of the flight in accordance with the 5G Network Operator. The proposed algorithm aims at selecting the flight path, from the start point to the landing point (drone's flight plan), by considering not only the shortest path but also reducing the number of service migrations.

UAV flight planning in 5G – on reducing MEC service relocation

In general, the deployment of drones should follow several steps [72], [73], divided into three blocks: (i) Scope Definition Block; (ii) Drone Block consisting of Flight Planning, Flight Implementation, and Data Acquisition; (iii) Software Block where Data Analysis, Data Interpretation and Optimization are conducted. In the first block, the mission statement is clearly established, and precise objectives are defined; e.g. for network traffic analysis and behavioural studies. The second block consists of preparing the flight, considering Safety & Environment conditions, and route planning. The third and final block proceeds during the flight, where all the operations (Analysis, Interpretation and Optimization) on the Data acquired by the Drones are executed.

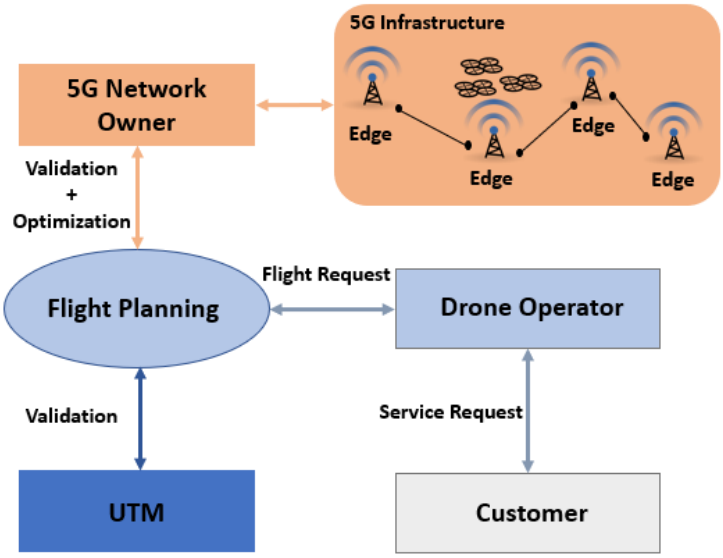


Fig. 21: Flight Planning actors

Flight Planning: This task is extremely important, for security reasons, since the conditions of the flight are negotiated at this level. The main concerned stakeholders [74] are, as shown in Fig. 21.

- Customer: The entity or user wanting to benefit from a drone service. It can be an individual (for entertainment), a company (delivery purposes, etc.) or even a government. The application on the UAV can vary depending on the customer.
- Drone Operator: The entity responsible of controlling the drones, and offering UAV-based services, and proposing flight plans depending on the needs of the Customer. The Drone Operator is the entity, which deploys the Drone Pilot as an application at the edge.
- Network Operator (NOP): The entity holding the 5G infrastructure and offering 5G coverage.
- UAV Traffic Management (UTM): A centralised entity responsible of the management of drone's flights, since it holds information about all the drones flying in the areas; all information such as presence of drones, their trajectories, locations [73], [74] are centralised at the UTM.

The preparation of a flight plan for the mission consists of proposing a path (or route) followed by the drone during a given interval of time in accordance with Network Operator infrastructure information. The flight plan needs to be later validated by the UTM. The information that the NOP has to indicate to prepare the flight plan is the network coverage of the flying area, to check whether the infrastructure can offer the needed 5G network coverage and states. For instance, if at a given instant, the network infrastructure is overloaded with huge traffic, ensuring very low latency to the C2 link for UAV may be difficult. In this case the flight plan should be modified, and another flight time should be proposed. Obviously, we assume that the NOP is aware of the state of network in its infrastructure during the day long. Another information that the NOP should provide is the number of MEC servers and their mapping with 5G base stations. At this step, our proposed algorithm proceeds, by helping the drone operator to find a path that reduces the number of service relocations, and avoid overloaded base stations.

Regarding the UTM, the validation consists of checking if the area is safe in terms of environmental conditions (people in places where the drone flies, weather, restricted zones etc.), and UAVs collisions; for example, if the proposed flight plan is in an close area where other drones are flying in the same altitude, then there will be a risk of conflicts and collisions between drones, which means that the flight plan will be rejected and another flight plan should be proposed. The Flight Preparation step ends with a validated and agreed on flight plan, which contains the list of cells followed by the drone, and the corresponding time.

As stated earlier, our proposed algorithm intervenes at the route planning step of the mission preparation. As usually modelled, the mobile network is composed of a set of base stations, where each base station has a hexagonal coverage [75]. In Fig. 22, we show an example of a mobile network topology, where each MEC Server (noted edge) covers an area composed of a group of cells. The Drone Pilot application can be deployed at the MEC server to ensure low latency. We used colours to show the relation between a MEC server and a group of cells it covers.

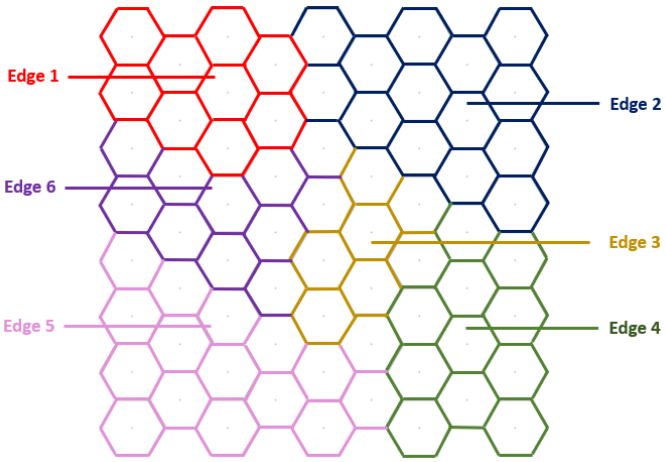


Fig. 22: Topology Edges

Let us suppose that a mission consists of flying a drone from a point A to point B (see Fig. 23, in different areas. We consider then that the Drone Pilot application is first instantiated in Edge1 server as it covers the initial position of the drone. Then, the Drone Pilot application is migrated among the servers according to the drone mobility. The straight path between the two points consists in minimizing the distance travelled by the drone. But, in some cases, like the one depicted in Fig. 22, the straight path between the two points will require not less than three service migrations, since the drone will pass through both Edge1, Edge6, Edge3 and finally Edge4; which requires to migrate the Drone Pilot application accordingly. This would impact the performances of the C2 link as the duration of downtime could be consequent. However, from Fig. 23, we can see that another path is much more interesting, in

terms of service migrations; if the drone goes from Edge1 to Edge2, then to Edge4, it will get to the final destination with only two service migrations, which will considerably reduce the downtime duration. Hence, there is a need of an algorithm that returns the best path between the two points, in terms of minimizing the number of service migrations.

We propose to model the network topology of Fig. 22 by an oriented graph, where: the vertices represent the cells, the edges are weighted with either the distance between the two cells or the cost of service migration in the case where the two cells are under the coverage of different edges. The objective is to find the optimal trajectory, i.e. a set of cells to cross through that reduces the service migrations from the starting to the landing point.

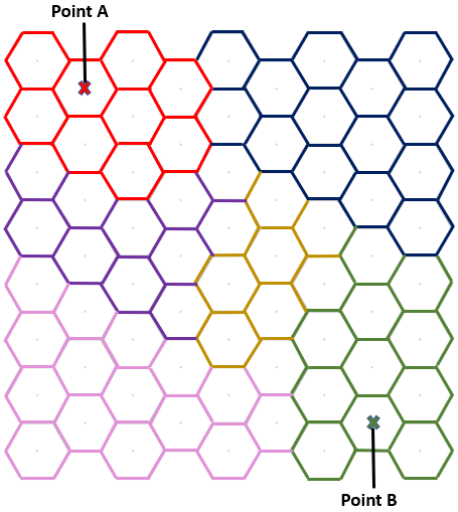


Fig. 23: Topology Points

We denote by E_i, the Edge server covering the area $\{C_{1i}, C_{2i}, ..., C_{ni},\}$, which consists of a set of cells, where C_{ji} represents the cell identified by j in the area covered by the Edge E_i,. As indicated earlier, we model the topology as a non-oriented graph (V, E) where V is the set of cells $\{C_{11}, C_{21}, \ldots, C_{n1}, ..., C_{1k}, C_{2k}, ..., C_{nk}\}$, k is the number of cells per Edge node, and n is the number of Edges.

We denote by $w_{(i,j),(k,m)}$ as the weight between two neighbouring, i.e. C_{ij} and C_{km}, which represents the cost of the service migration if the two cells are not under the same edge coverage, or the distance between them if they are in the same area. In our case, since the topology has a hexagonal form, all the distances are similar and equal to 1 for simplicity. Such a graph is depicted in Fig. 24, where C is the fixed service migration cost between two edges.

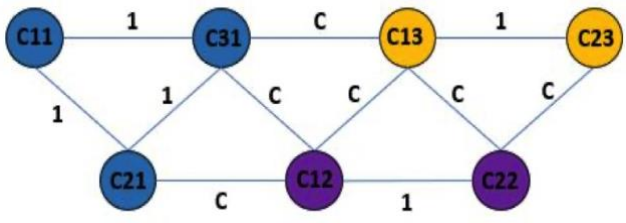


Fig. 24: Graph topology weights

As indicated earlier, another parameter that may impact the C2 link performances (i.e. latency) is the fact that a selected cell may be overloaded during the flight period, by other types of network traffic. Therefore, we add another parameter to the model, which is the cell overload probability (noted $P(t)$) that indicates the overload of a cell at an instant t. For example, at a given time, every cell of the topology

will have a probability of being overloaded; indeed, the probability of finding a cell busy at the rush time is different from other periods of the day. To introduce this parameter to our model, we include it in the weight of the edge between the cells. This way, the selected path will consider the distance, the service migration cost, and the overloading of the destination cell. *P(t)* can be computed with the use of a forecasting model, trained using collected data on the mobile network traffic dynamic [76]. We derive the weight $w_{(i,j),(k,m)}$ of an edge (Fig. 25) as follows:

$$w_{(i,j),(k,m)}(t) = \begin{cases} 1 + (t) & \text{if } j = m \\ C + P_{km}(t) & \text{else} \end{cases} \quad (1)$$

where C_{ij} is the source cell, C_{km} the destination cell, t the requested instant and $P_{km}(t)$ the probability of the cell C_{km} being overloaded at the instant t.

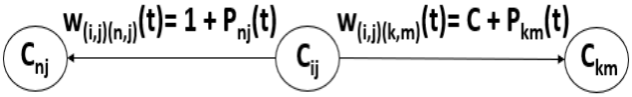


Fig. 25: Graph topology weights deriving

To tune the impact of C and P_{km} on the edge's weight, which will drive the solution, we introduce a coefficient, noted α. Now the edge weight is expressed as follows:

$$w_{(i,j),(k,m)}(t) = \begin{cases} 1 + (1-\alpha)P_{km}(t) & \text{if } j = m \\ \alpha C + (1-\alpha)P_{km}(t) & \text{else} \end{cases} \quad (2)$$

where $(0 \leq \alpha \leq 1)$.

Thus, one can use the value of α to steer the solution by giving more priority for reducing service migration, or more priority for visiting less loaded cells. Since $P_{km}(t)$ expresses a probability, its value is between 0 and 1, which is not the case for the service migration cost (C). Hence, the two values are normalised to give them the same scale, to make them influence the model in the same way.

Having defined the weight of the link connecting two adjacent nodes (neighbour cells), we denote by $P=(c_1, c_2, c_3, ..., c_n\}$ a path in V; where $c_i \in V$ and c_i, c_{i+1} are two adjacent nodes. We note by $f(c_i, c_{i+1})$ the function that returns the weight of the link between c_i and c_{i+1} as defined in equation (2). Now the problem consists in finding a path P that minimizes $\sum_{i=1}^{n-1} f(c_i, c_{i+1})$.

To find the optimal path P, considering both the service migration and the cell overload, from the initial point to the landing point, we propose two algorithms. The first one is based on the well-known Dijkstra algorithm [77]; while the second one is based on a greedy algorithm (Prim) [78]. The two proposed algorithms are not sensitive to any use-case, since they just compute the best path regarding the chosen metrics.

The Dijkstra-based algorithm (Fig. 26) calculates for each node, the shortest distance from the source node to it. To do that, it first initializes the initial node with a current distance of 0 and the distances of the remaining nodes with infinity. Then, it sets the non-visited node with the smallest current distance as the current node (S). For each neighbour (N), it adds the weight of the connection between S and N to the distance from the source to S. If the new value is smaller than the previous distance from the source to N, it updates the latter with the calculated value. It repeats this process until all nodes are visited. Our contribution to the Dijkstra algorithm is the usage of the overload probability of the target

node when we compare and update the distances from the source node. Indeed, instead of taking only the weight of the connection between the nodes, we add the probability value to that connection.

Algorithm 1: Djikstra-based Algorithm

```
P := {};
d[S] := ∞ for all S in the graph;
d[start] := 0;
while There is a node out of P do
    Choose a node S out of P with min distance d[S];
    Put S in P;
    for Each node B out of P and neighbor of a do
        if d[B] > d[S] + weight(S,B) +
         overload_probability(B) then
            d[b] := d[a] + weight(S,B) +
             overload_probability(B);
            previous[B] := S;
        end
    end
end
```

Fig. 26: Dijkstra-based Algorithm

In addition, to the Dijkstra-based algorithm, we also introduce a greedy one (Fig. 27), namely Prim algorithm [77], [78]. It creates from a given graph, the Minimum Spanning Tree (MST), which is another graph extracted from the initial one, where all the vertices are connected via a path, and where the sum of all the weights is the minimum, taking into consideration the service migration as well as the overload probability. This algorithm first initializes the MST as an empty set, and then takes at every step the minimum weight edge from the initial graph, and add it to the MST in the case that an edge is valid. A valid edge between two nodes is when one end of it is already included in the MST and the other one is not. These steps are repeated, and the number of edges in MST (*nbEdges*) is incremented at each step until the MST holds a number of edges equal to the number of nodes in the initial graph (*graphSize*) minus 2 (*nbEdges* = *graphSize* - 2). Once the MST is formed, we reconstruct the path from a source to a target node using Breadth-first search algorithm [79], which is an algorithm for exploring a graph by going through all of the neighbour nodes at the present depth, then moving on to the nodes at the next depth level and so on, until we find the target node. This way we give the expected path using Prim algorithm.

Algorithm 2: Prim-based Algorithm

```
nbEdges := 1;
MST := {};
while nbEdges < graphSize - 1 do
    Find minimum weight edge E;
    If E is valide edge then add it to MST;
    nbEdges := nbEdges +1;
end
```

Fig. 27: Prim-based Algorithm

To evaluate the performances of both algorithms, we executed them on the topology of Fig. 23. Note that this topology is just an example; we assume that each NOP has such a model for the geographical locations covered by its mobile network.

We considered two scenarios for tuning the expected solution; i.e. giving more priority for minimizing the service migrations, or for avoiding overloaded cells. To achieve this, we selected different values of α. It is worth noting that for all the scenarios, the migration cost *C* and the overload probability *P(t)* are normalised to give the same scale to the two parameters, by simply multiplying *P(t)* by 10, since the chosen default value of the service migration cost is 10, and the overload probability is ranged between 0 and 1. Any other way of normalization between the two variables can be easily done. From this point, all the values of *C* and *P(t)* are normalised.

- Scenario 1: In this scenario, more weight is given to the service migration, i.e. the expected solution tends to pass through an overloaded cell than to migrate a MEC service. The value that we used for alpha is α=0.8, so that the migration cost is greater than the overload probability.
- Scenario 2: Unlike Scenario 1, in this scenario, more weight is assigned to the overload probability, i.e. the expected solution avoids overloaded cells, and accepts more service migrations. We used for this scenario α=0.2, which will make the overload probability greater than the migration cost.

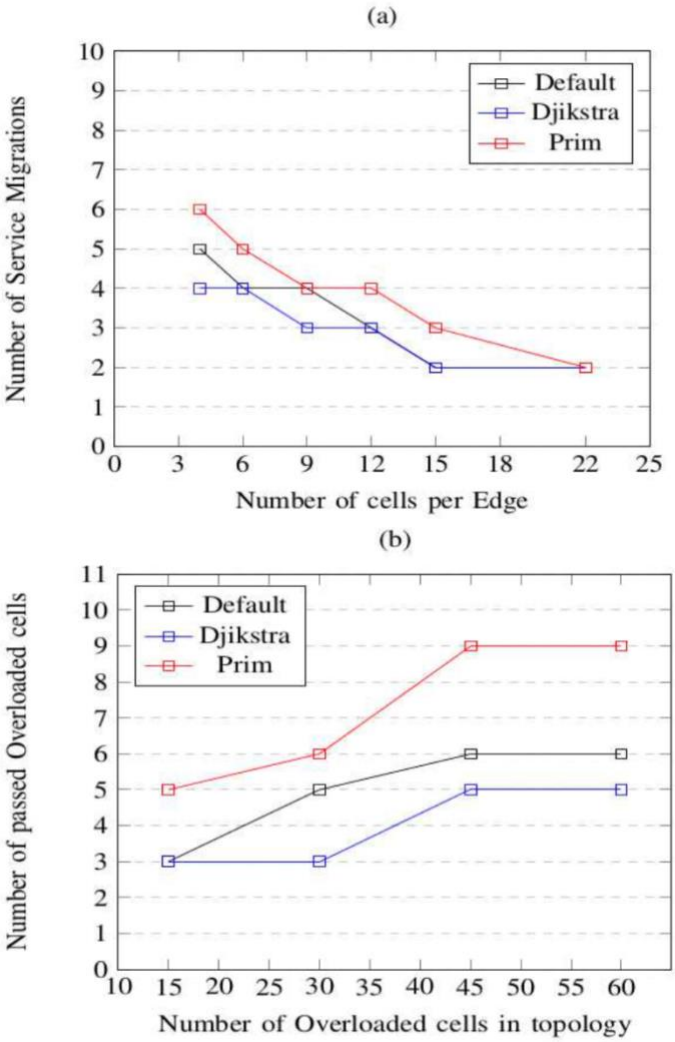


Fig. 28: Metrics evolution for Scenario 1

As stated earlier, we use the example of Fig. 23, which consists of a topology formed by 68 cells. For each scenario, we measure the performances of the three solutions in terms of the number of service migration, and the number of used cells that are overloaded. To measure the number of service migration, we first generate in a random way the overload probabilities of the cell, and we vary the number of cells per MEC Edge from 4 to 22; while for the number of overloaded cells we fixed the number of cells (9 per Edge server) covered by an Edge server and vary the number of overloaded cells in the topology.

We assume that a cell is overloaded when its overload probability exceeds a certain threshold fixed by the network owner (0.5 for example). For the sake of comparison, we also execute the Dijkstra algorithm (as default in the figures) to find the shortest path, i.e. the weights of the graph edges are all equal to 1. This way, we obtain the shortest path, in terms of distance, between the initial and landing point. We then compute the number of service migrations and overloaded cells used by the path found by the three algorithms.

Fig. 28 and Fig. 29 shows the performances of the three solutions, in terms of the two metrics, for scenario 1 and 2, respectively. The algorithms were implemented in Python. For both scenarios the number of service migration decreases as the number of cells per edge increases, which is logical as the higher the number of cells inside an edge is, the lesser the number of edges is, and the lesser the number of service migration is. Similarly, the number of overloaded cells selected in the proposed path increases as the number of overloaded cells in the topology increases.

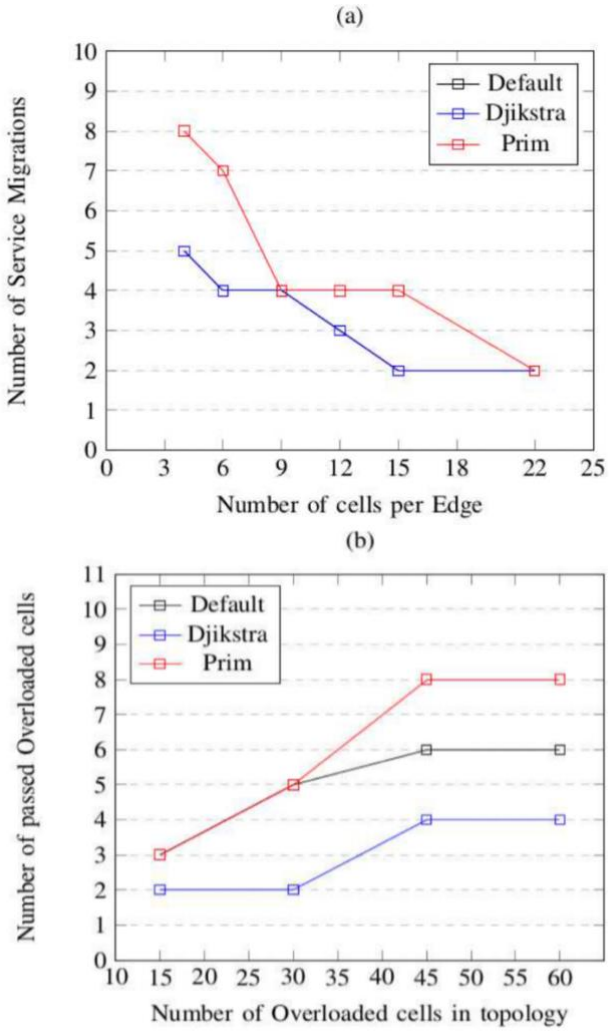


Fig. 29: Metrics evolution for Scenario 2

This is obvious as the higher the number of overloaded cells in the whole topology is, the higher the number of overloaded cells selected in the proposed path is. In addition, we remark that in Scenario 1, both algorithms (i.e. Dijkstra-based and the default) achieve similar results and the best performances in terms of the number of service migrations. However, Dijkstra-based algorithm gives better results in terms of the number of overloaded cells held in the path. We argue this by the fact that the default algorithm always finds the same solution as the objective is to select the shortest path; while the Dijkstra-based algorithm takes into consideration both metrics, by giving more weight for reducing the number of service migration. We also note that the greedy algorithm gives the worse solution than the two other solutions for both metrics.

For scenario 2, we see in Fig. 29 (a) that the Dijkstra-based algorithm behaves like the default solution in terms of the number of service migration, while it achieves the best results in terms of the number of overloaded cells selected in a path (Fig. 29 (b)). We justify this by the fact that using the shortest path allows having interesting solutions for service migrations, while the Dijkstra-based algorithm is seeking solutions that rather find a trade-off between the two metrics, with more weight given to reduce the number of overloaded cells. Indeed, we observe clearly in Fig. 29 (b) that the Dijkstra-based algorithm achieves the best results in terms of the number of overloaded cells in the topology. It is worth mentioning that in the case of the Dijkstra-based algorithm, the number of service migration in Scenario 2 is higher than in Scenario 1 (they are mostly between 3 and 8 in Scenario 2, while in Scenario 1 they are between 2 and 4). We argue this by the fact that the Dijkstra-based algorithm gives more importance for avoiding the overloaded cells, which means that the model favours a path with less overloaded cells. For the same reason, the number of overloaded cells runs through is less in Scenario 2 than in Scenario 1 (they are between 2 and 4 in this scenario while in Scenario 1 they are between 3 and 5).

These results clearly prove that our model is sensitive to the parameters' weights, which allows the drones operator to tune the model depending on its wishes, whether it wants to reduce the service migrations or the overloaded cells, or even equitably considering both parameters. Note that the difference between those numbers in the two scenarios is not consequent due to the used topology, where the edges are split in a way that the service migration cannot exceed a certain number. The difference should be more consequent and visible on a bigger topology, where the edges are numerous, which is expected in 5G.

4. UAV SERVICE COMPONENTS INTERACTION WITH THE INFRASTRUCTURE ENABLERS

There are already published research and standardisation documents that relate to the support of UAS operations by mobile networks describing the proposed/potential way of interaction between legacy network infrastructure components and UAS/UTM systems. Such documents as : "Study on application layer support for Unmanned Aerial Systems (UAS)" [80], "Study on supporting Unmanned Aerial Systems (UAS) connectivity, Identification and tracking" [81], and "Unmanned Aerial System (UAS) support in 3GPP" [82] should be mentioned and considered. However, the main concern is that all aforementioned publications and researches are still at the infancy stage.

There are several implementation challenges that would need to be addressed without clear guidance from the available recommendations or technical reports. Even following the general architecture proposal, as depicted on Fig. 30, specific adjustments would need to be done to provide the necessary functionalities required by particular use case scenarios and to be aligned with still not fully established legislative regulations related to UAVs. Among faced challenges, the most considerable include:

- Support for MEC application mobility – to ensure that MEC based applications can be smoothly and instantly migrated between MECs to support UAV's mobility.
- Assurance of necessary QoS for communication (direct and network-assisted) of C2 and telemetry information (including a retransmission) as a basis for each type of the mission.
- Support for communication of broadcasting type, which is crucial for manned aviation (mentioned below U2U, UAV to UAV).

In Fig. 30, a diagram of U-space ecosystem as described by 3GPP is presented. In the presented architecture, the 3GPP shows currently foreseen interactions than involve UAVs, UAC, 3GPP network as well as entities linked with aviation ecosystem, i.e. UTM and Third-Party Authorised Entity (TPAE).

Based on the above reference architecture model, 5G!Drones implementation of U-space related interfaces should be as follow:

- **UAV6** interface is used for the purpose of sending notifications/alarms from the PLMN network to UAS/UTM (U-space);
- **UAV6** interface is also used to provide network-related information like radio coverage and related quality KPIs used for SORA analysis;
- **UAV9** is used for telemetry data gathering and non-verbal bi-directional communication between ATC and UAV operator (e.g. Controller – Drone Data Link Communication).

The information exchanged between UTM system, and PLMN network can have strategical (mission planning related, e.g. information about radio coverage) and tactical (during mission execution, when the network failure would have an impact on ongoing flights) meaning. The other point refers to the tracking support of UAVs via UAV6 interface: in 5G!Drones testbed environments tracking information will be provided within the telemetry data through the UAV9. It is also not planned to implement network-based identification, neither authorisation via UAV6. The interface between UTM and PLMN, UAV6 are used to provide alarms/notifications from PLMN and eventually to provide access to radio coverage and related quality KPIs. Tracking information through the network (utilising UAV6) will be used for specific use case scenarios.

The other important aspect is that some of the components specific to 5G!Drones trials will be deployed as VNF applications. A common example for all project's use cases is the C2 and telemetry application. Hence to provide reliable C2 service (and telemetry) it is not the only question of low latency link, but

also appropriate MEC infrastructure that would provide and support VNFs mobility across different MECs, including full applications' runtime context. This topic, as well as proposals for some advanced flight routing improvement algorithms taking into consideration network topology of MECs, is covered in the section dedicated to UAV supporting MEC specific mechanisms (cf. section 3.5.7).

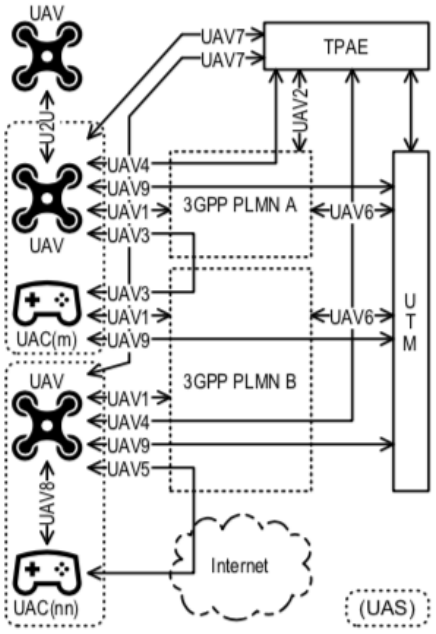


Fig. 30: 3GPP reference architecture of the U-space ecosystem (based on [81])

The UAS operator service components are implemented as a set virtual network functions (VNFs) on the network edge and user equipment (UE) on the access side. Drone UEs on the access side is managed on an application-level by UAS operators and hence require no infrastructure-level management outside providing them with network connectivity such that they can communicate with VNFs on the network edge. UAS operator VNFs on the network edge such as software pilots or supporting services (e.g. video analysis) are managed by infrastructure-level enablers only in the sense that infrastructure-level enablers are expected to provide services for the instantiation of those VNFs. Once the VNFs have been initialised, application-level management (e.g. mission configuration, drone deployment, application life cycle transitions) is the responsibility of WP2 enablers or UAS operators and hence is not in the scope of WP3.

Ensuring safety in common airspace is a matter of prime importance regarding making commercial UAV services a reality. Many legislation and standardisation bodies have been putting significant efforts into creation of rules and procedures that could guarantee high level of security for unmanned flights. One of the results, originally proposed by the Joint Authorities for the Rulemaking of Unmanned Systems (JARUS), but also strongly supported by the European Aviation Safety Agency (EASA) is the set of rules to assess risk in conducting specific flights called Specific Operations Risk Assessment (SORA). SORA is the approach on how to safely create, evaluate and conduct an Unmanned Aircraft System (UAS) operation. SORA is based on assigning to a UAS-operation two classes of risk, a ground risk class (GRC) and an air risk class (ARC), which are used to determine the so-called Specific Assurance and Integrity Levels (SAIL) for ARC and GRC. The SAIL, on the other hand represents the level of confidence that the UAS operation will stay under control within the boundaries of the intended operation.

4.1. Supported scenarios

4.1.1. Use Case 1: UAV traffic management

To progress further, the UAV industry needs to pass from VLOS to BVLOS mode of operation. It means that operations, which are now performed or supervised by human pilots, must be automated in the future, which high degree of confidence. Apart from the algorithms designed to react and answer to all kind of situations, the latency factor will play the highest importance. And the answer is Edge Computing solution, plus additional functionality, like slicing and mobility. This Use Case is allowing remote supervision and control of the autonomous drone flight from any place in the world, using the 5G mobile network (the link between UAV and GCS) and internet connection (the link between GCS and UTM). The main responsibility for collision avoidance, de-conflicting and reacting to unforeseen situations is placed in GCS, which makes decisions related to the flight. In the experiment setup, pilot on the place is required for safety backup, where an unforeseen situation can happen, which cannot be managed by GCS software. Hence, for this scenario, the main importance is the low latency of the communication between drone and GCS, assured by the short distance between UAV in the air – GCS running in the MEC. Other monitoring functions and UTM are placed on the internet, and the max requirement in terms of latency is 2 s if a drone flies with speed up to 20 m/s (which is the maximum speed of most commercial drones) to use minimal 50 m safety distance.

4.1.2. Use Case 2: Public safety

In the public safety area, there is a huge potential for MEC services, for example, for analysing big data sets (e.g. video and gas sensors analysing) in near real-time and in a confidential environment. Dedicated security enablers are expected from 5G MEC by public authorities. MEC services can be used to analyse data captured by the drone and compare these data with public authority data that can be securely stored at the MEC level (e.g. wanted car numbers, rescue area hazard indicators and patterns of hazardous activities, wanted person photos, patterns of suspicious behaviour etc.), but it is not permitted to store such data on the drone's onboard computer. This is where the advantages of MEC come in terms of data security over data processing on the device itself or onboard a drone. It is noteworthy that UC2 tests are also planned to control IoT sensors and process data with MEC services.

4.1.3. Use Case 3: Situation awareness

In this use case, MEC services are used in the following scenarios:

- 3.1 Infrastructure Inspection;
- 3.2 UAV-enhanced IoT Data Collection;
- 3.3 Location of UE in non-GPS Environments.

In Scenario 3.1., it is important to process large data arrays (for example, LIDAR produces about 100 Mbps of data) in a very short time so that the drone can be redirected to collect additional data or adjust the flight plan or flight speed already while the drone is in the air. From the above, there is a need for the MEC application to share processing tasks between different 5G MEC cells or to process data collected during the UAV flight prior to the UAV mission. However, in Scenario 3.2, the processing of data collected by IoT sensors by the MEC service is important. In Scenario 3.3, priority is given to very low latency MEC services to transmit real-time data locations that allow for the most accurate location calculations. In smart city applications relevant to this use case, the MEC services will facilitate cloud applications for data analysing in near real-time and autonomous UAV management. Mapping can be used to update a real-time model of the environment used by the UAVs and the control applications.

The ETSI MEC will bring many benefits to these scenarios since they are latency-sensitive and require RNIS, Location API, at the edge etc. ETSI MEC will further improve the scalability and allows the sensor and components involved in these use cases to maintain a consistent and reliable connection.

4.1.4. Use Case 4: Connectivity extension & offloading during crowded events

The purpose of this scenario is to demonstrate how UAVs through 5G network capabilities can improve connectivity services in a highly crowded environment, e.g. during large events. The concept relies on providing end-to-end dedicated and reliable communication targeting specific user groups such as the event organisers to supervise and manage large events in an unhindered manner. At the same time, and with the proper dimensioning of the deployed solution in terms of capacity, the connectivity services can also be offered to the spectators. Controlling a drone via software components demands a guaranteed low-latency communication link, and deploying the UAS at the edge seems the perfect fit. In this case, it is also important to run the automated video analysis service as a MEC service to determine which locations have more people and where the 5G service should be strengthened.

4.2. Characteristics of UAV specific service components

The 5G!Drones project develops service components for four use cases with a total of 10 scenarios. Each scenario has its own service components. Table 2 lists the five main service components that interact with infrastructure enablers.

Table 2. UAV use-cases service components

Service component	5G slice type	MEC service
U-space component	URLLC – Ultra-Reliable Low Latency Communication	U-space application
Tactical deconfliction	C-V2X – Vehicle to Everything	Vehicle/drone communication protocol and application
IoT devices management and data processing	mMTC – massive Machine Type Communication	IoT devices C2 application
Command and Control	URLLC – Ultra-Reliable Low Latency Communication	C2 application
Real-time transmission and processing of large-scale data (LIDAR datasets, video analysing, etc.)	eMBB – enhanced Mobile Broadband	Applications for processing big data in near real-time

From the U-space perspective, when the UTM as a use case service component is considered, there are few types of interactions with infrastructure enablers. They might have different characteristics: some of them are related to strategical (pre-flight) information exchange, and others are related to dynamic, tactical (in-flight) information flows. On a strategical level, when flight security evaluation and approval is considered, it should be ensured that network-related information like radio coverage and related quality KPIs are provided to the UTM system for SORA analysis purposes. This kind of information most probably would be updated occasionally, whenever updates to network coverage or related network configuration are changed in the way, that flight-related KPIs will be impacted. This information will be used for SORA analysis. On the other side, there is a “dynamic” (online, real-time) exchange of the information between U-space and infrastructure that impacts on-going missions. This information covers:

- Telemetry data – this information must be provided from all UAVs, constantly during the mission with short time intervals (every 1-3 s) and passed to U-space for the purpose of traffic monitoring in the airspace;
- Immediate alarms from infrastructure to U-space related to the infrastructure failures, which might impact vital KPIs of the service and thus all ongoing missions in the impacted area;
- Notifications passed between U-space and UAVs operators (e.g. emergency requests to change/abort the mission).

Use case service components provided by UAV operators interact with infrastructure enablers in two locations: the network edge and the access side. The access side provides UAVs with the network connection required to connect to services hosted at the edge. To infrastructure enablers, UAVs on the access side are generic user equipment (UE). The network edge hosts supporting services for UAV flights. These services include both primary flight services, which enable the control and command of UAVs and auxiliary services, which support the requirements of the use case or vertical. Primary flight services include software pilots or ground-control stations (GCS) and are responsible for coordinating and controlling associated UAVs. In this capacity, these services typically require low latency communication to the UAVs. Auxiliary services cover a broader range of use-case specific functionality. Examples include video analysis services, which provide, e.g. real-time object recognition for in-flight use or mapping services, which use measurements obtained during a flight to provide real-time information to the operator or experimenter. Given that these services cover a broad range of functionality, their slicing requirements vary and hence should be specified on a case-by-case basis. In both cases, these services are provided as generic virtual network functions (VNFs) and are managed or deployed by infrastructure enablers like any other VNF. Requirements such as latency or bandwidth should be specified as generic slicing requirements as opposed to providing special consideration to these services. This promotes a healthy separation of concerns between infrastructure components and UAV use-case components.

From the UAS operator point of view, the standardised, reliable channel for communication with UTM should be available. Reliability is provided by the mechanism of request-confirmation type of communication: each request must be clearly acknowledged by the UAV operator. This channel is used to pass emergency information and notifications between the operator and air traffic controller (ATC) that typically include:

- Check-in request/approval;
- Notification about lost control of the drone;
- Request for immediate landing or leaving the zone.

Example of the existing bidirectional, non-verbal communication protocol to be used for this purpose in 5G!Drones project is CDDLC (Controller-Drone Data Link Communication). UAS operator service components are modelled as virtual network functions (VNFs) on the network edge and user equipment (UE) on the access side. Drone UEs on the access side is managed on the application level by UAS operators and hence require no infrastructure-level management outside, providing them with network connectivity such that they can communicate with VNFs on the network edge. UAS operator VNFs on the network edge such as software pilots or supporting services (e.g. video analysis) are managed by infrastructure-level enablers only in the sense that infrastructure-level enablers are expected to provide services for the instantiation of those VNFs. Once the VNFs have been initialised, application-level management (e.g. mission configuration, drone deployment, application life cycle transitions) is the responsibility of WP2 enablers or UAS operators and hence is not in the scope of WP3 (except the operations like MEC mobility management described in more details in previous section). There is a

number of software components that a UAS operator wants to use with MEC, based on the facilitation it provides.

These include:

- C2 software for conducting drone flights etc.;
- 5G QoS mapping software;
- Video analysing software;
- IoT devices management and data processing etc.

Therefore, the UAS operator needs an interface, through which it can order MEC services and install MEC-based applications, monitor application performance and integrate them into its company's ICT systems. Such access must be flexible, but at the same time, sufficient security must be ensured for the MEC infrastructure. A possible solution is described in Table 3.

Table 3. Core functions of UAS operator management interface

Function of MEC interface	Description	5G MEC requirements
Installation of MEC applications and ordering services	Install, configure, modify, and test applications. Ordering MEC services and resources with specific parameters	When granting access to the MEC, the security of the rest of the MEC infrastructure must be ensured
Monitoring of MEC applications performance and data analytics	Analysing the applications Possibility to analyse 5G network KPIs	Technical possibilities for identifying KPIs
Integration of MEC services into the UAS operator's own ICT systems.	Integrations that ensure both the management of MEC applications by the UAS operator's own ICT systems and automated information exchange.	Security and load limitations must be ensured, considering the MEC infrastructure capabilities.

4.3. MEC requirement of UAV use-cases

ETSI MEC considers three types of MEC deployment requirements:

1. MEC application that requires traffic redirection to access the user-plane traffic, and request a MEC Service (e.g. Radio Network Service Information (RNIS) or location API). To recall a MEC service is provided by the MEC Platform via mp1 interface (JSON format).
2. MEC application that requires only access to UP traffic (need traffic redirection), which could correspond to the C2link Command/Control application.
3. MEC application that requires MEC service without needing to access the user- plane traffic.

According to the above classification, in Table 4, we provide for each use-case the needed type of deployment.

Table 4. Type of needed MEC deployment for 5G!Drones use-cases

5G!Drones UC	Type of deployment	Application
UC1: Scenario 1 – Deployment 1	Type 2	C2 link for command & control
UC1: Scenario 1 – Deployment 2	Type 2	C2 link for command& control
	Type 2	Video link for First Person View (FPV)
UC1: Scenario 2	Type 1	Location API and RNIS for outdoor trial
	Type 2	C2 link for indoor trial
UC1: Scenario 3	Type 1	Location API and RNIS and C2 link for automated mission command& control and Video link for First Person View (FPV) for medicaments delivery.
UC2: Scenario 1	Type 2	C2 link for command& control, Video
UC2: Scenario 2	Type 1	C2 link for automated mission command& control and Video analyser and MCS application
UC2: Scenario 3	Type 1	Location API for flying between buildings and Video analytics services and C2 UAV flight control service (C2 link) and Video link for First Person View (FPV) for Police operation.
UC3:Scenario11SSC1	Type 1	3D mapping service, 5G QoS measuring and processing applications and C2 UAV flight control service (C2 link).
UC3: Scenario 1 – SSC2	Type 2	LIDAR mapping and processing services and C2 UAV flight control service (C2 link).
UC3: Scenario 1 – SSC3	Type 2	C2 link for command & control, and Video link for First Person View (FPV)
UC3: Scenario 2	Type 2	UAV flight control service (C2 link).
	Type 2	IoT data aggregation and analysis
UC3: Scenario 3	Type 1	Location API, RNIS.
UC4: Connectivity during crowded events	Type 1	C2 UAV flight control service (C2 link), 3D Mapping application, 5G QoS application, Video analysing application and connectivity planner application

4.4. 5G!Drones facilities support for MEC and network slicing

As a trial (ICT-19) project, 5G!Drones conducts trials implicating UAVs on two ICT-17, namely 5G-EVE and 5Genesis facilities. The project also extends its trials to 5GTN and X-Network testbeds in Finland. All four platforms embed ETSI MEC and Edge cloud capabilities. We survey herein the pre-existing MEC and edge cloud features in the four considered facilities then discuss the required MEC enablers required to trial UAV based services.

4.4.1. 5G-EVE testbed

ETSI MEC edge computing solution is used in 5G-EVE as it is compliant with the 3GPP architecture and includes several recommendations on how to offload the traffic to the Edge application. In addition, the ETSI MEC includes specifications on how to describe a MEC application via the AppD and the process of its LCM via the MEC Edge Orchestrator.

4.4.2. 5GENESIS testbed

The Athens Platform integrates edge computing infrastructure in various locations within its topology, for the deployment of edge applications and Network Service components. The overall deployment architecture of Athens Platform can be broken down into the two separate sites namely the NCSRD and Cosmote. Regarding NCSRD site the current solution only supports edge computing but not following concretely specific ETSI standardization. Adopting virtualisation and Service Function Chaining capabilities offered by NFV will enable the creation of an LBO point. As a result, traffic that would normally reach the services sitting behind the 4G/5G core utilizing the backhaul connection will be steered locally and either reach services instantiated at the edge or reach through the internet using local connections. In order to achieve that there is a need to deploy a 5GC function locally at the edge computing infrastructure. On the other hand, Cosmote site integrates a hybrid 4G/NSA 5G/MEC testbed complemented with an OpenStack-based SDN/NFV Cloud infrastructure. More specifically, the OTE Group 4G/5G testbed is composed of:

- A lightweight 4G/5G EPC/IMS CN (running on 2 VMs on a Dell R630 server)
- Two flavours of MEC implementation:
  - Via second SPGW;
  - Via SGW-LBO;
- Nokia Airscale 4G/5G BTSs for providing 5G radio connectivity;
- Eight NOKIA 4G/WiFi Flexi-Zone Multiband Indoor Pico BTS, supporting standard network interfaces (such as S1 and X2), 5/10/15/20 MHz LTE carriers with 2x2 MIMO, along with Wi-Fi connectivity @2.4 and 5GHz delivering thus a HetNet solution.

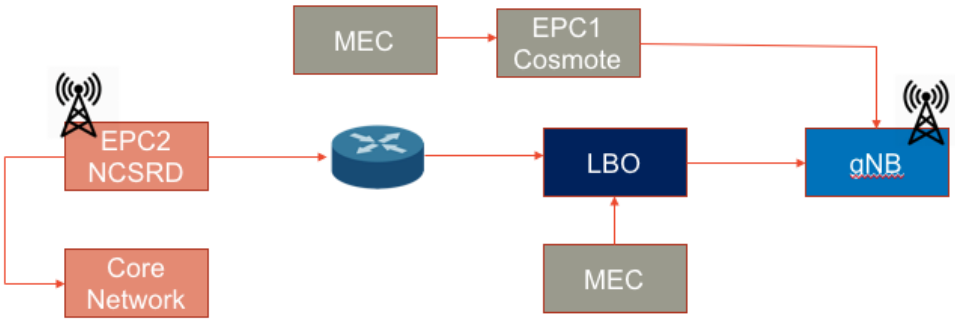


Fig. 31: Interconnection of NCSRD and Cosmote architecture

Fig. 31 illustrates the next iteration of updates on Athens platform that is to interconnect the two sites under a common architecture that will allow MOCN functionality, connectivity extension and offloading, as well as a proper LBO approach, following MEC specifications. Moreover, the current infrastructure uses an NSA deployment that will be shifted to a SA setup.

4.4.3. 5GTN testbed

The Nokia vMEC, based on ETSI MEC architecture (RGS/MEC-0003v211Arch), is used in the University of Oulu 5GTN because of its current availability in the facility. It also brings many benefits to the implementation of several use cases in the 5G!Drones project. It includes a rich software suite that provides different MEC services. In terms of MEC implementation for the 5G!Drones project, there are two phases at the University of Oulu 5GTN infrastructure (the full system architectures and their brief explanations are given below). The first phase is the MEC deployment in 5G Non-standalone (NSA) mode, and the second one is the MEC deployment in 5G Standalone (SA) mode. Currently, the MEC

platform for the NSA setup is supported in the 5GTN facility while it is being fine-tuned regarding the network configuration. The MEC platform will be shifted to the SA setup as soon as SA becomes available in the facility.

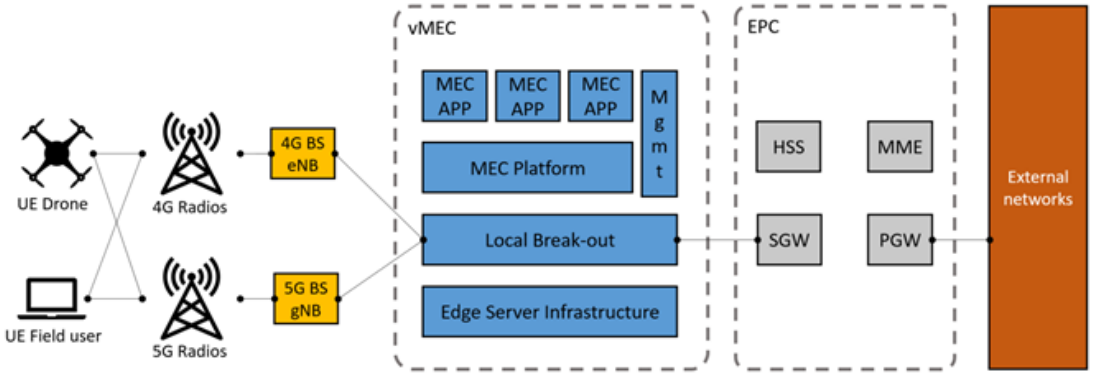


Fig. 32: 5GTN MEC deployment in NSA mode

5GTN MEC deployment in NSA mode: In NSA mode of 5G New Radio (NR) configuration, the control-plane is first established between the UE and eNB, and then the user-plane is established between the UE and gNB. In this configuration (Fig. 32), the UE connects to eNB and/or gNB, and that is connected to the EPC. The Nokia vMEC is located at the edge server. According to ETSI MEC distributed SGW-LBO scenario, the MEC host co-locates with the SGW where both the SGW-LBO and the MEC applications are hosted as virtualised network functions (VNFs) in the same MEC platform. The offered MEC applications can include, for instance, C2link Service, Video Analysis Service, 3D Mapping Service, AR/VR Visualization Service, etc. For this setup, the 5GTN implementation is based on the following components:

- 4G and 5G mobile phones as well as modems;
- eNB: Nokia Pico BTS;
- gNB: Nokia Indoor 5G BTS and/or Macro BTS;
- MEC: Nokia vMEC;
- EPC: NextEPC.

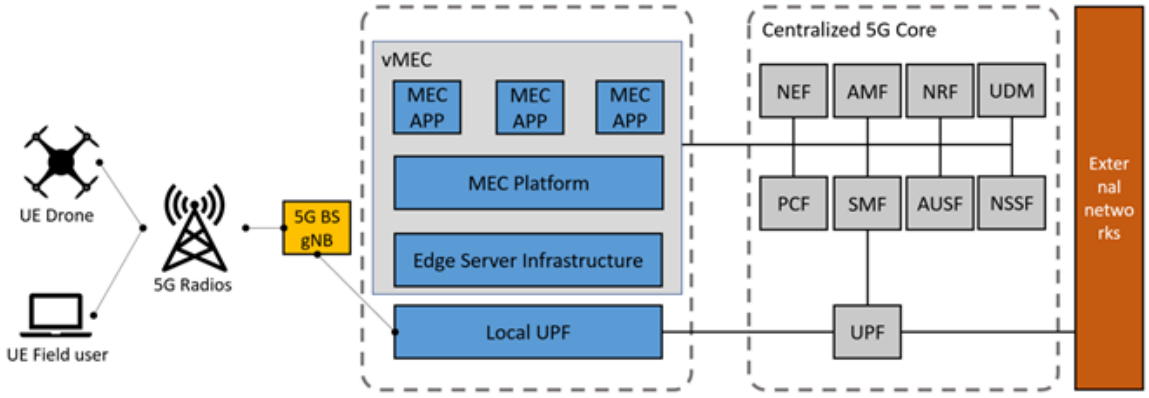


Fig. 33: 5GTN MEC deployment in SA mode

5GTN MEC deployment in SA mode: SA mode means a 5G network where the gNB is used for both the signalling and data transfer. In this setup (Fig. 33), the UE is connected to the gNB, and that is connected to the 5GC. The Nokia vMEC is deployed at the edge server where the MEC host's data plane is mapped to 5GC's UP functional entity, UPF. All the applications and functions, including the MEC

applications (Video Analysis Service, 3D Mapping Service, etc.) are VNFs. The 5GTN SA implementation is built using the following components:

- 5G mobile phones and modems;
- gNB: Nokia Indoor 5G BTS and/or Macro BTS;
- MEC: Nokia vMEC;
- 5GC: Open5GS.

4.4.4. X-Network testbed

The MEC/edge solution of Aalto University's trial site is deployed between the data centre and the radio access network. This deployment is not considered as ETSI compliant and allows hosting vertical application near UP network functions.

For X-Network testbed, the same approach discussed in the previous section has been followed for the slicing of the CN, where a CUPS-based EPC is used to create multiple NSIs that share a common control plane NFs and have dedicated UP NFs. Fig. 34 depicts an example of two network slices created in X-Network testbed. It is to be noted that the SGW-C and PGW-C are deployed as a single N,F which is the SPGW-C and that the SGW-U and PGW-U are also deployed as a single NF, which is the SPGW-U.

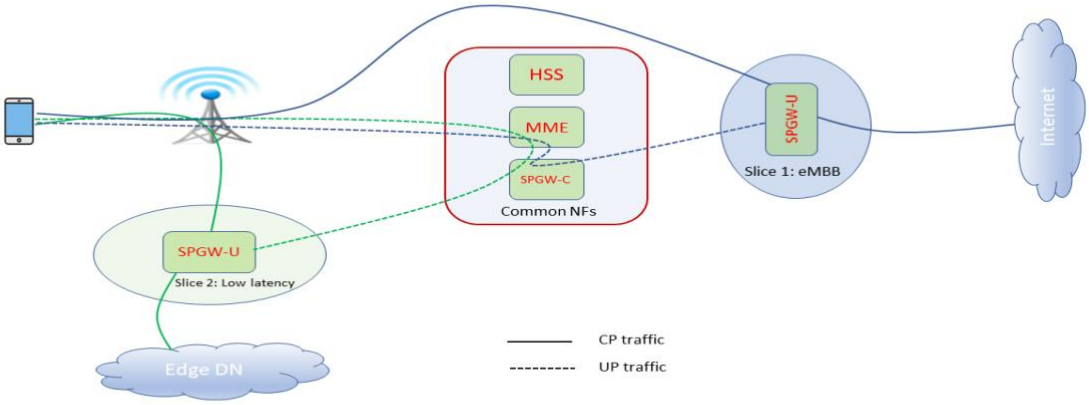


Fig. 34: Example of network slicing with CUPS-based CN

The CUPS-based EPC used in X-Network implementation is the Open Mobile Evolved Core (OMEC) [83] from Open Networking Foundation (ONF). OMEC is an open-source project in which the components that constitute the CN are provided by other sub-projects. Below is a short description of the OMEC's network functions:

- HSS: Cassandra-based Home Subscription Service that is provided as part of the C3P0 project that includes other optional NF such as the Charging Data Function (CDF) and the Policy Charging Rules Function (PCRF).
- MME: Provided by the Nucleus project, which is a ground-up implementation of the mobility management entity. Its design is performance optimised for high-speed mobility events over the S1-MME interface with the RAN.
- SPGW-C: Provided as part of the Next Generation Infrastructure Core (NGIC-RTC) project, which CUPS-based implementation of the SGW and PGW.

- SPGW-U: Provided by the UPF-EPC project, which is a revised version of the SPGW-U provided by the NGIC-RTC project. The UP is built on top of Berkeley Extensible Software Switch (BESS) programmable framework, where each submodule in the SPGW-U pipeline is represented by a BESS-based module. Thus, this version of SPGW-U is very flexible, and other functionalities can be added as BESS-based module, for example, it is possible TO extend the data processing pipeline by adding modules that perform Deep Packet Inspection (DPI), or that can encrypt plain UP traffic. Customizing the UPFs with different flavours enables the customization of NSI with different features.

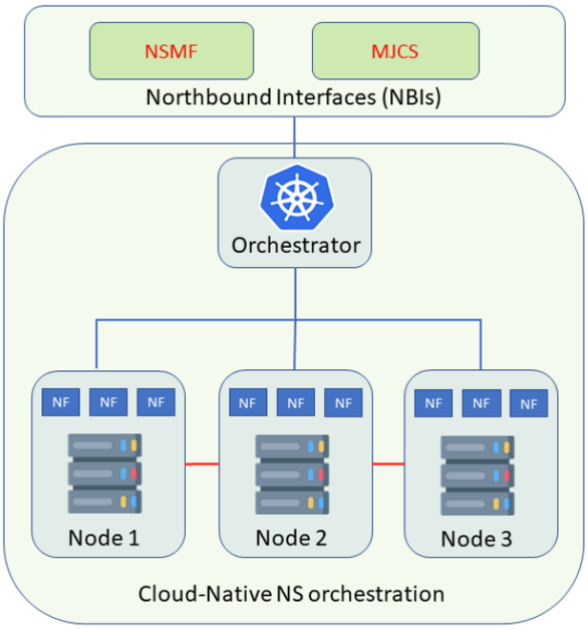


Fig. 35: NS platform in X-Network testbed

As depicted in Fig. 35, the X-Network testbed adopts a "Cloud-Native" approach for deploying and running the OMEC's network functions, with as objectives to improve the speed of NSIs deployment process, better scalability of NSIs, and ensuring fault tolerance of NSIs, some of the building blocks of X-Network orchestration platform are the following:

- **Containers:** A lightweight virtualisation alternative to Virtual Machines (VMs). In contrast to VMs that package an entire operating system along with the network functions, containers only package the network functions with their dependencies and use the shared kernel of the host machine. Hence, containers are characterised by less consumption of computing resources and with a smaller start-up, recovery, and upgrade time. In X-Network testbed, the network functions are deployed as containers using Docker, which is the most complete and popular containers engine.
- **Orchestrator:** Responsible for scheduling the virtualised network functions (i.e. Docker containers) on top of the physical infrastructure. The scheduling process includes two main steps: i) finding the order, on which the network functions need to be created; ii) finding the optimal placement of the network functions across the physical infrastructure. The orchestrator adopted in X-Network is Kubernetes, which is widely used for the orchestration of Docker-based services. In Kubernetes terminology, the physical infrastructure is called a Kubernetes cluster.
- **Networking:** Traditional IP routing-like networking cannot cope with the new requirements of Cloud-Native systems where instances of virtual services can be created, moved, and stopped very quickly. In this context, a new networking model suited for containers has emerged, which is called

Container Network Interface (CNI) and that allows the automation of the configuration of containers network interfaces and the overlay network connecting them. For example, using a CNI, containers can have an IP address in the same subnet even if they are deployed on top of different physical networks, and all this is available out-of-the-box. CNIs are considered as a plugin for the orchestrator and can be easily customised. In the X-Network testbed, the networking services that ensure the communication between the different NFs of an NSI are provided by the Calico CNI.

The **NSMF** acts as a network slice management service provider [84], it provides Northbound Interfaces (NBIs) to the Communication Service Management Function (CSMF) that plays the role of a network slice management service consumer. In our case, the 5G!Drones trial controller acts as a CSMF, and it is supposed to consume the services provided by the NSMF. The NSMF relies on a lower-level orchestrator to perform the different operations related to the LCM of NSI. The NSMF communicates with the orchestrator via its Southbound Interfaces (SBIs). The NSMF developed for the 5G!Drones project provides interfaces for the trial controller to perform the following operations:

- Checking the feasibility of a network slice instance;
- Creating a new network slice instance;
- Activating a network slice instance;
- Modifying a network slice instance;
- Deactivating a network slice instance;
- Deleting a network slice instance.

The **MJCS** is responsible for the LCM of measurement jobs, which are created to collect a set of KPIs related to a running network slice instance. The MJCS exposes interfaces to the 5G!Drones trial controller for performing the following operations [85]:

- Creation of measurement job for NSI;
- Termination of measurement job for NSI;
- Query of measurement jobs for NSI.

The measurement jobs in X-Network testbed use Prometheus exporters to collect KPIs such as computing resources usage (i.e. CPU and RAM) and network resources usage.
In the CUPS-based EPC, and throughout the establishment of a PDU session during the UE attach procedure, the SPGW-C will select the SPGW-U that will be used for the PDU session based on a set of criteria that are discussed in [14].

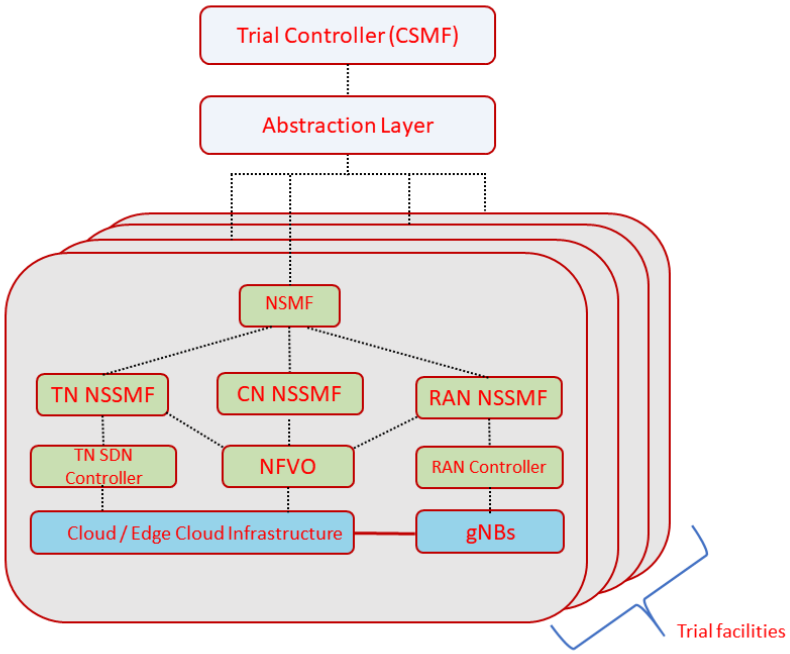
```
[DP_SELECTION_RULE_1]
DPID = 1
DPNAME = NSI-1
MCC = 244
MNC = 52
TAC = 1
DNS_PRIMARY = 1.1.1.1
DNS_SECONDARY = 8.8.8.8
STATIC_IP_POOL = 10.250.254.0/24
IPV4_MTU = 1450
NUM_IMSIs = 3
IMSIs = 244524567891201

[DP_SELECTION_RULE_2]
DPID = 2
DPNAME = NSI-2
MCC = 244
MNC = 52
TAC = 1
DNS_PRIMARY = 1.1.1.1
DNS_SECONDARY = 8.8.8.8
STATIC_IP_POOL = 10.250.255.0/24
IPV4_MTU = 1450
NUM_IMSIs = 2
IMSIs = 244524567891202
```

Fig. 36: Network slices selection rules in X-Network testbed (example)

For example, in order to reduce the communication latency, the SPGW-C can use the TAC sent in the session creation request to select an SPGW-U that is close to the RAN. However, in the 5G!Drones project there is a need to assign each UAV to its dedicated network slice instance (SPGW-U), but such functionality is not supported in the CUPS—based EPC. In this regard, we have extended the selection rules at the level of the UP to include the IMSI of UE. Thus, the SPGW-C will be able to select the appropriate UP for the UE, Fig. 36 depicts an example of UP selection rules used in the X-Network testbed, according to which, the UE with IMSI **244524567891201** is assigned to the UP identified by **NSI-1**, and the UE with IMSI **244524567891202** is assigned to the UP identified by **NSI-2**. It has to be noted that each time a new NSI is created, a new corresponding selection rule is injected in the SPGW-C.

4.5. Abstraction and federation of 5G facilities

In this context, the abstraction layer enabler allows the trial controller to have a unified view about the way the network services provided by the trial facilities are managed. As shown in Fig. 37, the orchestration architecture responsible for the management of the life cycle of end-to-end network slices on top of the trial facilities can be divided into three parts.



.

Fig. 37: Overall orchestration architecture

The trial facilities expose the network slicing management interfaces via the Network Slice Management Function (NSMF), which is responsible for the management of the life cycle of NSIs within a specific trial facility. For managing the life cycle of an E2E NSI, the NSMF delegates the management of each part of the slice (i.e. RAN NSSI, TN NSSI, and CN NSSI) to a specific Network Slice Subnet Management Function (NSSMF) that corresponds that part. The CN NSSMF makes use of the facility's NFVO to manage the life cycle of the VNFs that constitute the CN. It has to be noted that the NFVO implementation is facility-dependent, and it may be compliant to ETSI NFV or not. The RAN NSSMF makes use of the RAN controller to translate slice requirements into radio resource allocations and carry out high-level RAN resource management. Moreover, it makes use of the NFVO to manage the life cycle to virtual RAN access functions. Finally, the TN NSSMF interacts with the network control plane (i.e. SDN controllers) and the NFVO to manage the provisioning and isolation of the virtual network connecting the VNFs of the access and core networks.

The trial controller [86] responsible for the execution, automation, and monitoring of the UAV trials will be communicating with the trial facilities through an intermediate layer that allows the abstraction of the heterogenous nature and capabilities of the trial facilities. The controller plays the role of the Communication Service Management Function (CSMF), which is, according to 3GPP [84], responsible for triggering different operation related to the management of the life cycle of NSIs (i.e. creation, termination, modification, etc.). Moreover, this function is responsible for translating the communication service-related requirements to network slice related requirements during preparation phase [5]. The CSMF consumes the services provided by the NSMF.

The Abstraction Layer receives the generic requests sent by the trial controller to the trial facilities and translates them to facility-specific requests. Indeed, since each trial facility has its own implementation of network slicing, it is mandatory to abstract this heterogeneity by adding an abstraction layer between the trial controller and the trial facilities. This will provide a unified interface to the trial controller for accessing, per facility, network slices management services.

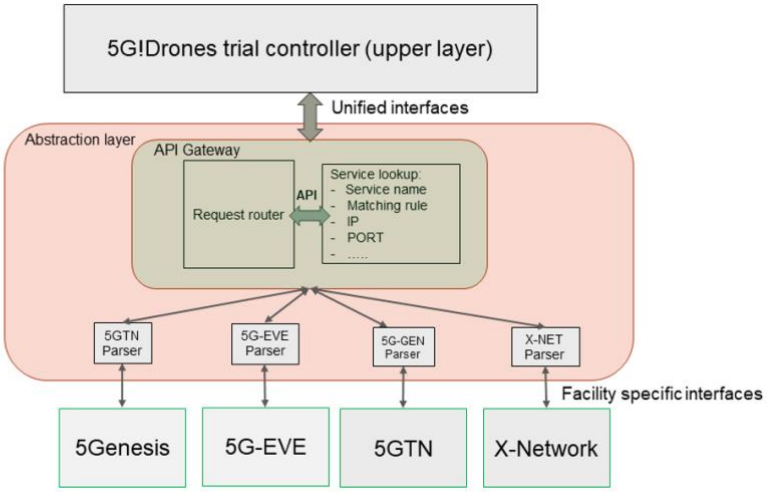


Fig. 38: Architecture of the abstraction layer

As depicted in Fig. 38, the proposed Abstraction Layer is composed of the following modules:

- **The API gateway:** Responsible for receiving the generic requests from the trial controller and routing them to the appropriate parser module. It is composed of two sub-modules:
  - **Requests router:** Uses the information stored in the parsers lookup module to route the requests to the appropriate parser.
  - **Parsers lookup:** Stores information on how to route each request to the appropriate parser, as well as information on how to reach the parsers.
- **Facilities parsers:** Translate the generic request routed by the API gateway to a facility-specific request and send it to the corresponding facility.

4.5.1. Abstraction Layer interfaces

When defining abstracted interfaces of the 5G!Drones overall solution, two contexts should be taken into consideration: the aviation multi-domain context and the 3GPP mobile network context. There are plenty of publications and ongoing research projects related to this topic. One of the most recent summaries and the synthesized view is provided in [87]. From the 3GPP perspective, the key reference documents are [81] and [82].

The first step for abstracting the heterogonous nature of trial facilities is the identification of the interfaces required by the trial controller and exposed by each facility. All the identified interfaces are subject to abstraction, wherein the aim is to provide unified interfaces to the trial controller for accessing, per facility, management, monitoring, and control, services. The interfaces required by the trial controller can be grouped into four categories:

- Network slices management interfaces;
- VNFs management interfaces;
- MEC applications management interfaces;
- KPIs monitoring interfaces.

In this section, the interfaces required by the trial controller to support the launch and LCM of UAV trials on top of trial facilities are described. The interface list is based on the concepts described in the 5G!Drones paper [88].

Network slices management interfaces

These interfaces are mainly used by the enforcement module to manage the life cycle of NSIs on the different trial facilities. According to 3GPP's Technical Specifications 28.531 [84] NSIs can be managed using a set of interfaces provided by the Network Slice Management Function (NSMF). This includes the following interfaces:

- **NSI feasibility check interface:** used by the validator sub-module to check whether the NSI requirements can be satisfied by a given trial facility at the starting time of the UAV's trial.
- **NSI creation interface:** used by the enforcement module as per a request from the trial controller LCM to deploy an NSI. This includes the reservation and configuration of all resources required by the NSI.
- **NSI activation interface:** used by the enforcement module to change the state of an NSI to the active state, which means that the NSI is ready to provide communication service to the UAV application.
- **NSI modification interface:** used by the enforcement module to modify the running NSI. This can map to several workflows, e.g. changes of NSI capacity, and NSI reconfiguration.
- **NSI deactivation interface:** used by the enforcement module to change the state of an NSI to the deactivated state, which means that the NSI is not available for providing communication services, e.g. NSI deactivation is mandatory before a NSSI modification.
- **NSI termination interface:** used by the enforcement module as per a request from the trial controller LCM to terminate its respective NSI. This mainly includes releasing the resources originally allocated for the NSI.

VNFs management interfaces

This set of interfaces is ensured by the orchestrator of the trial facilities to manage the life cycle of VNFs needed to support a specific UAV application (e.g. video streamer, IoT data collector, flight controller) deployed in the facilities central cloud as network services. The trial controller uses theses interfaces to provide the UAV verticals with the ability to deploy their own applications, either as a part of the NSI or as third-party's application running in the operator's data network. Based on ETSI NFV-IFA 013, these interfaces include the following:

- **VNFs packages management interfaces:** used by the enforcement module to on-board, enable, disable, delete, and fetch a VNF package. A VNF package is the file that includes the software image of the VNF and the VNF descriptor (VNFD). Initially, it is stored in the VNFs repository at the level of the trial controller.
- **Network services descriptors (NSDs) management interfaces:** used by the enforcement module to onboard, enable, disable, update, delete, and fetch an application descriptor (i.e. network service descriptor) that describes how the application must be deployed, i.e. constituent VNFs and the interconnections between them.
- **Network Services (NS) management interfaces:** used by the enforcement module to instantiate, scale, update, and terminate an application deployed as a network service.

Note that ETSI terminologies [89] are used to describe the life cycle of UAVs' applications, wherein each UAV application is considered as a network service composed of one or more VNFs. However, equivalent interfaces can be defined for all orchestration systems that are based on declarative configuration.

MEC management interfaces

In addition to the applications deployed in the operator's data networks, UAVs may need to communicate with applications characterised by URLLC requirements (e.g. UAV flight control services) that cannot be satisfied when the applications are hosted in distant data centres. Moreover, MEC can also be used to push UAV's specific applications that utilize high bandwidth (e.g. video streamer) to the edge in order to minimize traffic within the operators' network. Therefore, the trial controller requires access to interfaces that allow the management of such applications at the edge of trial facilities, i.e. near the base stations. MEC applications can be managed using the following interfaces:

- **Applications packages management interfaces** allow the management of the applications packages that bundle the files required for the instantiation of the UAV applications:
    - Application package onboarding interface: used by the trial controller to make the application package, stored in the VNFs repository, available to the MEC system.
    - Application package enabling interface: used to mark the application package is available for instantiation.
    - Application package disabling interface: used to mark the application package as not available for instantiation.
    - Application package deletion interface: used to delete the application package from the MEC system.
- **Applications instances management interfaces:**
    - Application instance creation interface: used to create a new instance of an application whose package has been already on-boarded and enabled.
    - Application instance operation interface: used to start and stop an already created application instance.
    - Application instance termination interface: used to delete a running application instance.

Note that, these interfaces are defined based on ETSI MEC [48] specifications. However, equivalents interfaces can be defined for all edge orchestration systems that are based on declarative configuration.

Key Performance Indicators KPI(s) monitoring interfaces

In addition to NSIs and applications management interfaces, the trial controller requires access to interfaces that allow the real-time collection of performance data. Indeed, the collected data will be used by the KPI monitoring module of the trial controller to analyse the effective performance so to take the appropriate actions accordingly. In [85], 3GPP specifies how the performance of 5G systems can be monitored by third parties' applications. The described procedure consists of creating measurement jobs on generic objects (e.g. NSI, or a VNF instance), and waiting for the data stream to be sent to the stream target specified in the measurement job creation request. Hence, the proposed trial controller requires access to the following set of interfaces:

- **Measurement job creation interface:** allows the creation of one measurement job that can collect the values of one or multiple KPIs.
- **Measurement job termination interface:** used to terminate a running measurement job after the end of the UAV trial.
- **List measurement jobs interface:** used to list the running measurement.

5. CONCLUSIONS

This document is reporting on infrastructure-level enabling tools and techniques for 5G!Drones resulting from activities of WP3 performed in the context of network slicing, MEC and network abstractions needed for 5G!Drones trials. It includes both research- and implementation-related topics.

The network slicing technology is still evolving, even at 3GPP. Regarding network slicing, the topics related to RAN slicing, scalable network slicing and usage of shared functions for network slicing have been described in this deliverable. It is worth noting that the 3GPP has not defined the standardized way in which RAN slicing should be implemented. Nevertheless, a standardized API for the interaction with RAN for slice operations has been introduced.

The in-slice management concept enables isolation of slice management spaces and provides slice runtime management interface to slice tenant, which in our case is UASP. This interface can be used for reporting to UASP service-related KPIs and making high-level reconfigurations of the slice. We have also included generic discussion on the role, standardization and methods of KPI calculation in a network slicing environment.

The MEC description includes the evolution of the concept from 4G towards 5G networks with the inclusion of network slicing and integration with NFV. As the integration of MEC with 5G is not yet fully standardized, we have proposed a new way of such integration. MEC security issues have also been briefly discussed.

Solutions that involve techniques for service relocation, i.e. research on the Follow Me Edge concept or a simple algorithm for MEC hosts selection during flight planning procedures and performing the service migration with accordance to the established placement with simulation results have been presented. Such mechanisms are of premium importance for UAV services.

Interconnection of UAV and 5G ecosystems in the 5G!Drones project is using the underlying system abstractions. In this context, the introduced abstraction layer allows the trial controller to have a unified view of the way the network services provided by the trial facilities that are managed. Such abstraction plays a vital role in terms of forming facility federation out of the distinct 5G network slicing-enabled solutions supplied by consortium members. The abstracted view, i.e. the abstraction layer APIs, will be used for executing 5G!Drones trials in a uniform way.

This deliverable (D3.1) presents infrastructure-level enablers for 5GDrones as outcomes of work performed on the matter. The vertical service-level enablers are now under investigation; they will get introduced in next deliverable, namely D3.2 due on M26. The 5G!Drones Enablers will then be released through a Software Suite due on M32 (deliverable D3.3).

References

[1] 3GPP: 3GPP TR 23.799 "Study on Architecture for Next Generation System", V14.0.0 (Dec. 2016).

[2] 3GPP: 3GPP TS 23.501 "System Architecture for the 5G System", V16.6.0 (Sep. 2020).

[3] 3GPP: 3GPP TS 28.530 "Management and orchestration; Concepts, use cases and requirements", V16.3.0 (Sep. 2020).

[4] 3GPP: 3GPP TR 28.800 "Telecommunication management; Study on management and orchestration architecture of next-generation networks and services", V15.0.0 (Jan. 2018).

[5] 3GPP: 3GPP TR 28.801 "Telecommunication management; Study on management and orchestration of network slicing for next-generation network", V15.1.0 (Jan. 2018).

[6] ITU-T: ITU-T M.3000 "Telecommunications management network, Overview of TMN Recommendations" (Feb. 2000).

[7] ETSI: "ETSI adds extra dimensions to virtualization of communication networks with continued NFV specification activity" (Oct. 2017), [Online]. Available:

http://www.etsi.org/news-events/news/1220-2017-10-news-etsi-adds-extra-dimensions-to-virtualization-of-communication-networks-with-continued-nfv-specification-activity

[8] L. Xu, H. Assem, I. Grida Ben Yahia, T. S. Buda et al.: "CogNet: A network management architecture featuring cognitive capabilities", 2016 European Conference on Networks and Communications (EuCNC), pp. 325-329. **https://www.doi.org/10.1109/EuCNC.2016.7561056**

[9] FP7-4WARD project: "D4.2 In-Network Management Concept" (Mar. 2009).

[10] IBM: "Autonomic Computing White Paper: An architectural blueprint for autonomic computing", 3rd edition, IBM White Paper (Jun. 2005).

[11] ETSI NFV ISG: ETSI GS AFI 002 "Generic Autonomic Network Architecture (An Architectural Reference Model for Autonomic Networking, Cognitive Networking and Self-Management)", V1.1.1 (Mar. 2013).

[12] ETSI: "Zero touch network & Service Management (ZSM)", [Online]. Available:

http://www.etsi.org/technologies-clusters/technologies/zero-touch-network-service-management

[13] 3GPP: 3GPP TR 21.915 "Release description; Release 15", V15.0.0 (Oct. 2019).

[14] 3GPP: 3GPP TS 23.214 "Architecture enhancements for control and user plane separation of EPC nodes", V16.2.0 (Sep. 2020).

[15] X. Foukas, N. Nikaein, M. Kassem, Mohamed, M. Marina, K. Kontovasilis: "FlexRAN: A Flexible and Programmable Platform for Software- Defined Radio Access Networks", 12th ACM International on Conference on emerging Networking EXperiments and Technologies (CoNEXT 2016), pp. 427–441. **https://www.doi.org/10.1145/2999572.2999599**

[16] "Mosaic5G and FlexRAN" Accessed on 23.11.2020. [Online]. Available:

http://mosaic-5g.io/flexran/

[17] X. Foukas, M. Marina, K. Kontovasilis,: "Orion: RAN Slicing for a Flexible and Cost-Effective Multi-Service Mobile Network Architecture", 23rd ACM Annual International Conference on Mobile Computing and Networking (MobiCom 2017), pp. 127-140. **https://www.doi.org/10.1145/3117811.3117831**

[18] C. Chang, N. Nikaein: "RAN Runtime Slicing System for Flexible and Dynamic Service Execution Environment", in: IEEE Access, vol. 6, 2018, pp. 34018–34042. **https://www.doi.org/10.1109/ACCESS.2018.2847610**

[19] E. Coronado, S. N. Khan, R. Riggio: "5G-EmPOWER: A Software-Defined Networking Platform for 5G Radio Access Networks", in: IEEE Transactions on Network and Service Management, vol. 16, no. 2, Jun. 2019, pp. 715-728. **https://www.doi.org/10.1109/TNSM.2019.2908675**

[20] "5G-EmPOWER" Accessed on 23.11.2020. [Online]. Available:

https://5g-empower.io

[21] S. Kukliński, L. Tomaszewski et al.: "A reference architecture for network slicing", 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft 2018), pp. 217–221. **https://doi.org/10.1109/NETSOFT.2018.8460057**

[22] ETSI NFV ISG: ETSI GS NFV 002 "Network Functions Virtualisation (NFV); Architectural Framework", V1.2.1 (Dec. 2014).

[23] S. Kukliński, L. Tomaszewski: DASMO: "A scalable approach to network slices management and orchestration", NOMS 2018 – 2018 IEEE/IFIP Network Operations and Management Symposium, pp. 1–6. **https://www.doi.org/10.1109/NOMS.2018.8406279**

[24] ETSI: ETSI TS 102 250 series of standards "Speech and multimedia Transmission Quality (STQ); QoS aspects for popular services in GSM and 3G networks", ETSI (May 2015).

[25] 3GPP: 3GPP TR 32.862 "Study on Key Quality Indicators (KQIs) for service experience", V14.0.0 (Mar. 2016).

[26] 3GPP: 3GPP TR 26.944 "End-to-end multimedia services performance metrics", V15.0.0 (Jun. 2018).

[27] ITU-T: ITU-T E.800 "Definitions of terms related to quality of service" (Sep. 2008).

[28] H. Koumaras et al.: "5GENESIS: The Genesis of a flexible 5G Facility", 2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD 2018), pp. 1-6. **https://www.doi.org/10.1109/CAMAD.2018.8514956**

[29] ONE5G project: "Deliverable D2.1 Scenarios, KPIs, use cases and baseline system evaluation" (Nov. 2017).

[30] 5G-MoNArch project: "Deliverable D6.1 Documentation of Requirements and KPIs and Definition of Suitable Evaluation Criteria" (Sep. 2017).

[31] 5GCHAMPION project: "Deliverable D2.2: 5GCHAMPION Key Performance Indicator and use-cases defined and specification written" (Mar. 2017).

[32] 5GCAR project: "Deliverable D2.1 5GCAR Scenarios, Use Cases, Requirements and KPIs" (Aug. 2017).

[33] 5G Applications and Devices Benchmarking (TRIANGLE) project: "Deliverable D2.6 Final Test Scenario and Test Specifications" (Sep. 2018).

[34] Euro-5G project: "D2.6 Final report on programme progress and KPIs" (Oct. 2017).

[35] ITU-R: ITU-R M.2083-0 "IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond" (Sep. 2015).

[36] ITU-R: ITU-R M.2410-0 "Minimum requirements related to technical performance for IMT-2020 radio interface(s)" (Nov. 2017).

[37] 3GPP: 3GPP TS 22.261 "Service requirements for next generation new services and markets", V18.0.0 (Oct. 2020).

[38] 3GPP: 3GPP TR 38.913 "Study on scenarios and requirements for next generation access technologies", V15.0.0 (Jul. 2018).

[39] 3GPP: 3GPP TS 28.554 "Management and orchestration; 5G end to end Key Performance Indicators (KPI)", V17.0.0 (Sep. 2020).

[40] ETSI NFV ISG: ETSI GS NFV-IFA 027 "Network Functions Virtualisation (NFV) Release 2; Management and Orchestration; Performance Measurements Specification", V2.4.1 (May 2018).

[41] NGMN Alliance: NGMN Final Deliverable "Description of Network Slicing Concept", V1.0 (Sep. 2016).

[42] ETSI NFV ISG: ETSI GS NFV-IFA 013 "Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Os-Ma-nfvo reference point – Interface and Information Model Specification", V3.3.1 (Sep. 2019).

[43] ETSI NFV ISG: ETSI GS NFV-IFA 008 "Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Ve-Vnfm reference point – Interface and Information Model Specification", V3.1.1 (Aug. 2018).

[44] ETSI NFV ISG: ETSI GR NFV-EVE 008 "Network Function Virtualisation (NFV) Release 3; Charging; Report on Usage Metering and Charging Use Cases and Architectural Study", V3.1.1 (Dec. 2017).

[45] ETSI NFV ISG: ETSI GS NFV-IFA 006 "Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Vi-Vnfm reference point – Interface and Information Model Specification", V3.1.1 (Aug. 2018).

[46] ETSI NFV ISG: ETSI GS NFV-IFA 005 "Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Or-Vi reference point – Interface and Information Model Specification", V3.1.1 (Aug. 2018).

[47] ETSI MEC ISG: ETSI GS MEC 003 "Mobile Edge Computing (MEC); Framework and Reference Architecture", V2.1.1 (Jan. 2019).

[48] ETSI MEC ISG: ETSI GS MEC 010-2 "Multi-access Edge Computing (MEC); MEC Management; Part 2: Application lifecycle, rules and requirements management", V2.1.1 (Nov. 2019).

[49] ETSI MEC ISG: ETSI GS MEC 012 "Multi-access Edge Computing (MEC); Radio Network Information API", V2.1.1 (Dec. 2019).

[50] ETSI MEC ISG: ETSI GS MEC 013 "Multi-access Edge Computing (MEC); Location API", V2.1.1 (Sep. 2019).

[51] ETSI MEC ISG: ETSI GS MEC 014 "Mobile Edge Computing (MEC); UE Identity API", V1.1.1 (Feb. 2018).

[52] ETSI MEC ISG: ETSI GS MEC 015 "Mobile Edge Computing (MEC); Bandwidth Management API", V1.1.1 (Oct. 2017).

[53] 3GPP: 3GPP TS 23.273 "5G System (5GS) Location Services (LCS); Stage 2", V16.7.0 (Jul. 2020).

[54] 3GPP: 3GPP TS 38.215 "NR; Physical layer measurements", V16.3.0 (Oct. 2020).

[55] 3GPP: 3GPP TS 37.320 "Universal Terrestrial Radio Access (UTRA) and Evolved Universal Terrestrial Radio Access (E-UTRA); Radio measurement collection for Minimization of Drive Tests (MDT); Overall description; Stage 2", V16.2.0 (Oct. 2020).

[56] ETSI MEC ISG: ETSI GR MEC 017 "Mobile Edge Computing (MEC); Deployment of Mobile Edge Computing in an NVF environment", V1.1.1 (Feb. 2018).

[57] X. Foukas et al. "FlexRAN: A Flexible and Programmable Platform for Software-Defined Radio Access Networks", Proceedings of the 12th International on Conference on emerging Networking Experiments and Technologies (CoNEXT '16), pp. 427-441. **https://www.doi.org/10.1145/2999572.2999599**

[58] ETSI NFV ISG, ETSI NFV-MAN 001 "Network Functions Virtualisation (NFV); Management and Orchestration", V1.1.1 (Dec. 2014).

[59] L. Yala, P. A. Frangoudis A. Ksentini: "Latency and Availability Driven VNF Placement in a MEC-NFV Environment", 2018 IEEE Global Communications Conference (GLOBECOM), 2018, pp. 1-7. **https://www.doi.org/10.1109/GLOCOM.2018.8647858**

[60] S. Kukliński et al.: "A reference architecture for network slicing", 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), pp. 217-221. **https://www.doi.org/10.1109/NETSOFT.2018.8460057**

[61] ETSI MEC ISG: ETSI GR MEC 024 "Multi-access Edge Computing (MEC); Support for network slicing", V2.1.1 (Nov. 2019).

[62] ETSI MEC ISG: ETSI GS NFV-IFA 009 "Network Functions Virtualisation (NFV); Management and Orchestration; Report on Architectural Options", V1.1.1 (Jul. 2016).

[63] A. Ksentini, P. A. Frangoudis: "Towards Slicing-Enabled Multi-access Edge Computing in 5G", in: IEEE Network 34(2), 99–105 (2020). **https://www.doi.org/10.1109/MNET.001.1900261**

[64] L. Tomaszewski, R. Kołakowski, P. Korzec: "On 5G support of cross-border UAV operations", 2020 IEEE International Conference on Communications Workshops (ICC Workshops), Dublin, Ireland, 2020, pp. 1-6, **https://www.doi.org/10.1109/ICCWorkshops49005.2020.9145262**

[65] 3GPP: 3GPP TS 38.305 "NG Radio Access Network (NG-RAN); Stage 2 functional specification of User Equipment (UE) positioning in NG-RAN", V16.2.0 (Oct. .2020).

[66] 5G Americas: "The Status of Open Source for 5G", 5G Americas White Paper (Feb. 2019).

[67] O. Bekkouche, T. Taleb, M. Bagaa and K. Samdanis: "Edge Cloud Resource-aware Flight Planning for Unmanned Aerial Vehicles", 2019 IEEE Wireless Communications and Networking Conference (WCNC), Marrakesh, Morocco, 2019, pp. 1-7. **https://www.doi.org/10.1109/WCNC.2019.8886000**

[68] ETSI MEC ISG: ETSI GS MEC 021 "Multi-access Edge Computing (MEC); Application Mobility Service API", V2.1.1 (Jan. 2020).

[69] ETSI MEC ISG: ETSI GS MEC 018 "Mobile Edge Computing (MEC); End to End Mobility Aspects", V1.1.1 (Oct. 2017).

[70] R. Stoyanov, M. R. Kollingbaum: "Efficient Live Migration of Linux Containers", in: Yokota R., Weiland M., Shalf J., Alam S. (eds) High Performance Computing. ISC High Performance 2018. Lecture Notes in Computer Science, vol. 11203. Springer, Cham, pp. 184-193. **https://www.doi.org/10.1007/978-3-030-02465-9_13**

[71] R. Roman, J. Lopez, M. Mambo et al. "Mobile edge computing, Fog et al.: A survey and analysis of security threats and challenges", in: Future Generation Computer Systems, vol. 78, part 2, 2018, pp. 680-698. **https://www.doi.org/10.1016/j.future.2016.11.009**

[72] A. Aissioui, A. Ksentini, A. M. Gueroui, T. Taleb: "On Enabling 5G Automotive Systems Using Follow Me Edge-Cloud Concept", in: IEEE Transactions on Vehicular Technology, vol. 67, no. 6, Jun. 2018, pp. 5302-5316. **https://www.doi.org/10.1109/TVT.2018.2805369**

[73] M. A. Khan, W. Ectors, T. Bellemans, D. Janssens, G. Wet: "UAV-Based Traffic Analysis: A Universal Guiding Framework Based on Literature Survey", Transportation Research Procedia, vol. 22, 2017, pp. 541-550. **https://www.doi.org/10.1016/j.trpro.2017.03.043**

[74] Federal Aviation Administration. Unmanned Aircraft System (UAS) Traffic Management (UTM)," Concept of Operations" (2020), [Online]. Available:

https://www.faa.gov/uas/research_development/traffic_management/media/UTM_ConOps_v2.pdf

[75] D. A. Levine, I. F. Akyildiz, M. Naghshineh: "A resource estimation and call admission algorithm for wireless multimedia networks using the shadow cluster concept", in: IEEE/ACM Transactions on Networking, vol. 5, no. 1, Feb. 1997, pp. 1-12. **https://www.doi.org/10.1109/90.554717**

[76] I. Alawe, A. Ksentini, Y. Hadjadj-Aoul, P. Bertin, "Improving traffic forecasting for 5G core network scalability: A machine learning approach", in: IEEE Network, vol. 32, no. 6, Nov./Dec. 2018, pp. 42-49. **https://www.doi.org/10.1109/MNET.2018.1800104**

[77] E. W. Dijkstra, "A note on two problems in connection with graphs", Numerische Mathematik (1959).

[78] R. C. Prim, "Shortest connection networks and some generalizations", Bell System Technical Journal (1957).

[79] K. Zuse: "Der Plankalkül", Gesellschaft für Mathematik und Datenverarbeitung, no. 63, BMBW-GMD-63 (1972).

[80] 3GPP: 3GPP TR 23.755 “Study on application layer support for Unmanned Aerial Systems (UAS)”, V0.11.0 (Oct. 2020).

[81] 3GPP: 3GPP TR 23.754 “Study on supporting Unmanned Aerial Systems (UAS) connectivity, identification and tracking”, V1.1.0 (Oct. 2020).

[82] 3GPP: 3GPP TS 22.125 “Unmanned Aerial System support in 3GPP; Stage 1”, V17.2.0 (Oct. 2020).

[83] Open Networking Foundation: “Open Mobile Evolved Packet Core”, [Online]. Available: **https://opennetworking.org/omec/**

[84] 3GPP: 3GPP TS 28.531: “Management and orchestration; Provisioning”, V16.7.0 (Sep. 2020).

[85] 3GPP: 3GPP TS 28.550: “Management and orchestration; Performance assurance”, V16.6.0 (Sep. 2020).

[86] 5G!Drones D2.1: “Initial definition of the trial controller architecture, mechanisms, and APIs”, H2020 5G!Drones project (2019).

[87] L. Tomaszewski, R. Kołakowski, S. Kukliński: “Integration of U-space and 5GS for UAV services”, 2020 IFIP Networking Conference (Networking), 2020, pp. 767-772.

[88] T. Taleb, A. Ksentini, H. Hellaoui, O. Bekkouche: “On Supporting UAV based Services in 5G and Beyond Mobile Systems”, in: IEEE Network Magazine (Oct. 2020).

[89] ETSI NFV ISG: ETSI GR NFV 003 “Network Function Virtualisation (NFV) Release 3; Terminology for Main Concepts in NFV”, V1.5.1 (Jan. 2020).