

Development and Implementation of an Unmanned Aerial System for a 5G Vertical Use Case

by

Christos Chasis

Submitted to the Department of Electrical and Computer Engineering
in partial fulfillment of the requirements for the degree of

Integrated Masters in Electrical and Computer Engineering

at the

UNIVERSITY OF PATRAS

March 2021

Author
Department of Electrical and Computer Engineering
March 10, 2021

Certified by
Spyros Denazis
Professor
Thesis Supervisor

Certified by
Grigorios Kalivas
Professor
Thesis Supervisor

Accepted by
Kyriakos Sgarbas
Director, Division of Telecommunications
and Information Technology

Development and Implementation of an Unmanned Aerial System for a 5G Vertical Use Case

by

Christos Chasis

Submitted to the Department of Electrical and Computer Engineering
on March 10, 2021, in partial fulfillment of the
requirements for the degree of
Integrated Masters in Electrical and Computer Engineering

Abstract

Recent advances in the utilization of semi or fully autonomous platforms have created an emerging area of interest regarding their utilization in various aspects of applications. Despite that fact, network connectivity, a crucial part to the successful application of their benefits, is still in early research stages. In this work, an overview of current trends and technologies regarding cellular networks is presented, followed by a detailed description of a system architecture that aims to showcase the feasibility of a remote-controlled vehicle over a network. Emphasis has been given to engineering a complete end-to-end system that is comprised of primarily open-source components, while being forwards compatible in regards to experimental or commercial 5G deployments.

Thesis Supervisor: Spyros Denazis
Title: Professor

Thesis Supervisor: Grigorios Kalivas
Title: Professor

Acknowledgments

I would like to wholeheartedly thank my thesis supervisor Spyros Denazis for entrusting me with the responsibility of developing part of the 5G-VINNI use case deliverable and providing me with invaluable resources and guidance to refine and materialize my task. Without his contributions, I would not have the privilege of working with a distinguished research team such as the Network Architectures and Management group, nor learn as much in terms of leadership and project management.

Additionally, I would like to thank my thesis co-supervisor Grigorios Kalivas, which instilled me the principles of a proper engineer and a passion for wireless technologies, way before I even was sure about what I wanted to pursue for my professional career. It has been truly a pleasure cooperating with him, from past design contests up to the biggest task so far of my academic endeavors.

Last, but certainly not least, I should credit the people I had the privilege of working alongside, which brought (and still bring) in their own ways their countless years of experience towards making the first experimental 5G facility in Greece a reality. These are in no particular order Christos Tranoris, Panagiotis Papaioannou, Takis Apostolopoulos and Athanasios Chamalidis.

This work is dedicated to those who seek to push the boundaries of technology and through their hard work, help make our everyday lives a little bit better.

Contents

1	Introduction	15
1.1	Objective	16
1.2	Structure	17
1.3	Pre-Existing Work	18
1.3.1	3rd Generation Partnership Project	18
1.3.2	5G-PPP Architecture Working Group	22
1.3.3	European Telecommunications Standards Institute	22
2	Network Slicing Concept and Requirements	25
2.1	Network Slicing as a Concept	25
2.1.1	Roadmap of Network Slicing	28
2.2	Network Slicing Types	30
2.2.1	Enhanced Mobile Broadband (eMBB)	30
2.2.2	Ultra-Reliable Low Latency Communication (URLLC)	32
2.2.3	Massive Internet of Things (mIoT)	34
2.3	Network Slicing Requirements	35
2.3.1	Network Slicing Use Cases	36
2.4	Isolation on Network Slicing	37
3	5G Verticals Innovation Infrastructure	45
3.1	Architectural Principles	46
3.2	Template Facility Architecture	48
3.2.1	User Equipment	48

3.2.2	Radio Access Network (RAN)	49
3.2.3	Core and Transport Infrastructure	51
3.2.4	Management and Orchestration	53
3.2.5	Security, Testing and Monitoring Architecture	56
3.3	Interconnection and Interoperability of sites	58
3.3.1	5G Network Slice Federation Scenarios	60
3.3.2	Use Case Verticals	63
4	Drone-based Verticals Use Case	67
4.1	Purpose	67
4.2	Use Case and Key Performance Indicators	68
4.3	Communication paths	70
4.4	System Architecture Overview	71
4.4.1	User Side	72
4.4.2	Vehicle Side	73
4.5	Intermediate Implementation Steps	74
4.6	Suggested Hardware	75
4.6.1	Suggested Hardware for xNodeB	76
4.7	Bill of Materials (BOM)	80
5	Unmanned Aerial System Architecture and Implementation	81
5.1	System Overview	81
5.1.1	Intel Aero Compute Board	81
5.1.2	Flight Controller	84
5.1.3	Cameras	85
5.1.4	Additional Software	86
5.2	Networking	97
5.3	Docker and Container Networking	98
5.3.1	srsue	98
5.3.2	gstreamer	99
5.3.3	mavlink-router	100

6	Future Work and Development	101
6.1	Miniaturized Software Defined Radio platforms	102
6.2	Leveraging new streaming protocols	103
6.3	Going cloud native	104
	Acronyms	105

List of Figures

2-1	Network Slices on a Common Underlying Network. Source: [1]	26
2-2	Roadmap of Network Slicing. Source: [2]	29
2-3	Comparison of Server Virtualization Technologies	40
2-4	VMs vs Unikernels vs Containers in Terms of Isolation. Source: [3] .	41
3-1	5G-VINNI Facility with Main and Experimentation Sites. Source: [4]	46
3-2	SDN Architecture According to IETF RFC 7426. Source: [5]	55
3-3	Service Inter-Operation across 5G-VINNI Sites. Source:	60
3-4	E2E Network Slice Federation across three different Facility Sites. Source: 61	
3-5	Multiple Tenants and Federated Network Slicing. Source:	62
3-6	Greece UC1 – First Person View Remote Control Vehicle Use Sase . .	64
3-7	Greece UC2 – 360 ideo Broadcast Use Case	64
3-8	Greece UC3 - Narrowband IoT Use Case	65
3-9	Greece UC4 – AR Use Case	66
4-1	Use Case Scenario	70
4-2	Detailed Approach for User Side of Use Case	72
4-3	Detailed Approach for Vehicle Side of Use Case	73
4-4	Implementation Step 1 - Verify Communication	74
4-5	Implementation Step 2 - Verify Streaming	74
4-6	Implementation Step 3 - SBC Requirements and Remote Control . .	74
4-7	View of the N310 SDR platform. Source: [6]	79

5-1	Respective module block diagrams. Source: [7]	82
5-2	Downloading and setting correct kernel version.	84
5-3	Vehicle-side overview of the software sub-systems.	86
5-4	Default Ground Control Station software state.	91
5-5	RTSP video stream of the front-facing camera.	92
5-6	User-side overview of the software sub-systems.	93
5-7	Result of proper calibration of all sensors to arm motors.	94
5-8	Verifying wireless controller adapter is recognized as input.	94
5-9	Setting parameter COM_RC_IN_MODE to accept Joystick input.	95
5-10	MAVLink Router implementation	96
5-11	Complete view of the Ground Control Station software.	96
5-12	Installing the modem and placing the additional antenna mounts.	97
6-1	XTRX Pro Board for Scale. Source: [8]	102

List of Tables

2.1	eMBB Scenarios and Requirements Identified by 3GPP	31
2.3	URLLC Scenarios and Performance Identified by 3GPP	33
2.5	Comparison Between Types of VDUs	42
3.1	Service Offerings in the Greece Portfolio	63
4.1	Bandwidth in Relation to Video Quality	71
4.2	Software Defined Radio Platforms for xNodeB	77
4.4	Suggested Bill of Materials	80
5.1	Intel Aero Compute Board Connector Layout. Source: [7]	82
6.1	Comparison Between Video Streaming Protocols in ms. Source: [9]	103

Chapter 1

Introduction

The 5G Infrastructure Public Private Partnership (5G-PPP) is a joint initiative between the European Commission and the European Information and Communication Technology (ICT) industry, mainly manufacturers, telecommunications operators, service providers, Small and Medium size Enterprises (SMEs) and research institutions. This partnership will deliver solutions, architectures, technologies and standards for the ubiquitous next generation communication infrastructures of the coming decade. A key challenge is to secure Europe's leadership in potential sources for new market creation such as smart cities, e-health, intelligent transport, education or entertainment & media. This will reinforce the European industry as a whole, in order to successfully compete on global markets and open new innovation opportunities.

This effort comes under the scope of the European Union's Horizon 2020 research and innovation framework, where €700 million [10] have been pledged for investment over the course of the programme. At the time of writing, Large Industry and SMEs in 2018 mobilised private investments that sum up to an amount 10.12 times the European Community's (EC) public investment in the 5G-PPP. Additionally, the entirety of stakeholders invested a total amount of money that is 7.24 times the public investment in the same period [11].

1.1 Objective

The 5G-PPP as of writing, is in its third phase where new projects were launched in June 2018. Of these, three have been selected from the 16 proposals received by the EC in response to the 5G-PPP ICT-17-2018 call [12]. The purpose of this program is to implement and test advanced 5G infrastructures in Europe for a duration of at least 3 years.

The subject of this thesis will constitute part of one of the aforementioned research projects, 5G Verticals Innovation Infrastructure (5G-VINNI) [13], the mission of which is to develop a large-scale, end-to-end (E2E) 5G facility. This experimental facility will be used to demonstrate that performance capabilities conforming to 5G Key Performance Indicators (KPIs) can be met at realistic usage situations.

To ensure that intended demonstrations are both representative of real-world usage and comprehensive to a wider audience, 5G-VINNI aims to achieve high use case diversity. For this reason, the engagement of industry verticals such as those taking part in 5G-PPP Phase 3 projects is key. These, will bring a wide variety of innovative use cases that will be used to test and validate the 5G-VINNI facility and its components against 5G KPIs.

Primary example of such scenario would be the project's facility site in Patras, Greece, which will present four use cases as proof of concept, one of which is the main focus of this work. More specifically, this thesis aims to describe in detail the development and implementation of an unmanned aerial system platform, able to be remotely controlled and transmit a real-time video feed through its network connection. Additionally, the whole software stack is able to leverage the latest containerization technologies in order to be preemptively well-suited for future applications that would require microservice-enabled endpoints.

1.2 Structure

The rest of this chapter will provide essential references to pre-existing work regarding standards-based network principles that have been developed in industry Standards Developing Organisations (SDOs)s and previous 5G-PPP projects. Additionally, an overview of the work that underpins the development of the model 5G-VINNI facility is presented, under which individual facility sites will be based on. Chapter 2 will expand on the key concept of network slicing and its accompanying requirements as to clearly define the KPIs that will be required in order for the use case scenarios to be carried out successfully. Chapter 3 will delve further into the specifics of the project under which this thesis is being developed as to provide a better understanding of the architecture and morphology of a 5G-VINNI facility site network. Chapter 4 will showcase the use case proposal in detail along with suggested hardware for the implementation, while Chapter 5 will give an overview of the drone system architecture that is being implemented for said use case. Chapter 6 finally, will provide a glimpse into concepts that could benefit from the work presented on this thesis along with suggestions on future work for further improvements.

1.3 Pre-Existing Work

While 5G-VINNI is comprised of research projects, the nature of ICT-17 as a whole, is to build test networks that are standards-based and are able to demonstrate practical implementation of work that has previously been carried out in conjunction with funded research projects of its past phases. For this reason, some of the key contributors in 5G research and development are briefly outlined below in order to provide a better understanding of the foundation that preceded this project.

1.3.1 3rd Generation Partnership Project

TSG SA1 - Service

3rd Generation Partnership Project (3GPP) Rel. 14 Technical Report (TR) 22.891 [14] identified the market for verticals, the use cases and requirements that the 3GPP system would need to support. Namely, these are:

- Enhanced Mobile Broadband (eMBB)
- Critical Communications
- Massive Machine Type Communications
- Network Operations
- Enhancement of Vehicle-to-Everything (V2X)

TSG SA2 - Architecture

The specification of 3GPP Rel. 15 5G system architecture included in 3GPP Technical Specification (TS) 23.501 [15], 3GPP TS 23.502 [16] and 3GPP TS 23.503 [17] Rel. 15 provides an overview of the "5G Phase 1 System", including the key features and functionalities needed to deploy a commercial 5G network. Included are some prominent 5G-specific features:

I. Architecture Modularisation

To improve flexibility over the traditional 4G reference model, where clusters of functions are gathered under network elements, a set of flexible Network Functions (NFs) with looser implementation restrictions are introduced.

II. Service Based Architecture (SBA)

A service-based reference model is introduced, where NFs are able to mutually provide services.

III. Network Slicing

A network slice refers to a complete E2E logical network (i.e. both Access and Core Networks) providing telecommunication services and network resources. This enables the network operator to deploy multiple, independent Public Land Mobile Networks (PLMNs) where each is customized by instantiating only the features, capabilities and services required to satisfy the subset of the served users or a related business customer needs.

IV. Multi-access Edge Computing (MEC) support

MEC enables operators and 3rd party services to be hosted closer to the User Equipment's (UE's) access point of attachment, in order to achieve efficient service delivery through reduced latency and load on the transport network. Additionally, enhanced capability is provided, enabling traffic steering from the User Plane to the local Data Network.

V. Common 5G Core

In order to facilitate a gradual deployment of the next generation 5G System, integration with existing legacy 4G systems and to allow independent deployments of 5G RAN and 5G Core, 3GPP specified a set of architecture options¹.

VI. 3rd Party Application Function (AF) Integration

¹Numbering of options is not incremental and reflects the 3GPP defined enumeration.

The flexibility of its architecture coupled with network slicing makes the 5G system ideal for vertical industries, satisfying diverse and dynamic requirements. Additionally, it is able to provide enhanced capabilities allowing AF from 3rd parties to interact and influence 5G system behaviour. This will further enhance the possibility of customising the architecture and features of a communication system.

TSG SA3 – Security

5G brings a number of new possibilities for Mobile Network Operators (MNOs) to enable new use cases and business models, by introducing mobility to new business sectors and devices. However, this in turn has the potential to open up new threats to customer and the network security, which will not be expanded further is mentioned for the sake of completeness.

TSG SA5 - Telecom Management

3GPP TS 28.530 [18] defines use cases to be supported by the Rel. 15 3GPP management system which also introduces the key concepts relating to slicing and services.

I. Roles in the 5G network

The roles identified by 3GPP within a 5G network are defined as:

- Communication Service Customer (CSC) – for example, a vertical company.
- Communication Service Provider (CSP) – the actual entity that provisions the communication service for the CSC. This could be an operator that specializes in providing the V2X service.
- Network Operator (NOP) – the traditional network operator that hosts communication services.
- Virtualisation Infrastructure Service Provider (VISP)

- Data Centre Service Provider (DCSP) – cloud provider where the VNF may be hosted.

II. The Concept of the Network Slice Instance (NSI)

An NSI can be composed of multiple Network Sub-Slice Instance (NSSI) which could be shared across multiple NSIs. One or multiple NSIs could be used to provide corresponding communications services. Before deployment, the NSI has to be initialized, which includes designing, on-boarding and other network-related processes. Finally, when it is no longer needed, the NSI can be terminated.

III. The Management Plane (MP)

3GPP SA5 specifications for Rel. 15 have pointed towards a service oriented framework in the interest of modularity and scalability of management services. Key concepts and design principles of management services are defined in 3GPP TS 28.533 [19]. These include provision, fault and performance management services for NSI, NSSI and NF.

IV. E2E KPIs

With the introduction of slicing and supported use cases for verticals Communication Service Customer (CSC), there is a need to provide data on the management of said E2E service. 3GPP SA5 has concluded that the fragmented collection of KPIs of existing infrastructure and services would not be of interest to the CSC, and has instead created a new specification on the standardized E2E KPIs in 3GPP TS 28.554 [20]. Most notably:

- Uplink and Downlink Latency
- Upstream and Downstream Throughput
- Resource Utilization of Network Slice Instance

Radio Access Network (RAN) decomposition

In some implementations of LTE, RAN networks have implemented a separation between the Radio Unit (RU) and the Baseband Unit (BBU), by exposing the Common Public Radio Interface (CPRI). This has been conceived to allow the BBU to be at a network consolidation point, which results in operational efficiencies and cost savings. However, the CPRI interface is constrained by the need to meet low delay and high bandwidth requirements, resulting in a need for high capacity transport links. In 5G, however, 3GPP TR 38.801 [21] included work to study possible options for decomposition of the RAN environment which amounts the separation of the RU from the BBU.

1.3.2 5G-PPP Architecture Working Group

The 5G-PPP architecture working group has acted as the point of consolidation for 5G-PPP Projects in Phase 1, for topics relating to 5G Architecture. This has led to the production of two white papers which aim at "capturing novel trends and key technological enablers for the realization of the 5G architecture" [22]. Architectural concepts developed in various projects and initiatives are also targeted, so as to provide a consolidated view on the technical directions in the 5G era.

1.3.3 European Telecommunications Standards Institute

The European Telecommunications Standards Institute (ETSI) Industry Specification Group (ISG) for Network Functions Virtualization (NFV) is the group responsible for the standardization of NFV technology. NFV decouples Network Function (NF)s from purpose-built (proprietary, closed, costly) hardware appliances and migrates them to software images that can run on commodity hardware. Leveraging IT virtualization technologies, NFV enables traditional NFs to be virtualized, resulting in the so-called Virtual Network Function (VNF). These VNFs can be deployed and operated with great agility on top of commodity hardware, and can be flexibly combined to define network services. To manage them, ETSI ISG NFV has defined a reference framework,

with different functional blocks and interfaces to exchange information. An overview on the foundations for operation in virtualized environments is provided below.

The Concept of the NFV Network Service

An NFV Network Service (NS) is a composition of NFs. A NF is a processing function in the network which has defined functional behaviour and external interfaces. According to ETSI NFV, an individual NF may be implemented as a VNF (i.e. a software image running on commodity hardware) or as a Physical Network Function (PNF) (i.e. a purpose-built hardware appliance). The components of a NS include NF (with at least one VNF), virtual links and VNF Forwarding Graphs (VNFFGs). Virtual links are abstractions of physical links to the Control Plane (CP), exposed by the different NFs, providing connectivity between them. For a fine-grained control of its performance, scalability, security and reliability, a VNF can be decomposed into one or more VNF Components (VNFCs), each performing a well-defined task within the VNF functionality. Each VNFC is hosted in a single virtual deployment unit (e.g. virtual machine, docker container) and connected with other VNFCs through internal virtual links [23].

ETSI NFV reference architectural framework

The deployment and operation of NSs in (highly diverse) virtual environments bring multiple points of management that need to be carefully addressed. Most of them depend on the NF composition mechanism used for NS definition. This mechanism creates a set of dependencies between a given NS and its components, which exerts influence on their management and orchestration of the underlying resources. To compensate for that, ETSI ISG NFV has defined a reference architectural framework for NFV in ETSI Group Specification (GS) NFV-MAN 001 [24], which provides basic architectural foundations. These allow for consistency and uniformity during NS deployment and operation and consist of three main working domains:

I. Infrastructure and NF Layers

These layers include the Network Function Virtualization Infrastructure (NFVI) and the different network services (NF - VNF and PNF). The NFVI is the collection of all hardware and software resources that build up the environment on top of which VNFs (and their constituent VNFCs) run. Additionally, the set of resources that make up the NFVI could be distributed over multiple Points of Presence (PoP). In such case, VNFs could be deployed and executed across geographically remote areas, enabling mutli-site NFV scenarios.

II. NFV Management and Orchestration

This domain focuses on all the virtualization-specific tasks that are needed for NS deployment and operation in NFV environments, and is made up of the NFV Management And Network Orchestration (MANO) architectural framework. NFV MANO is a software stack consisting of three types of functional blocks, each with a well-defined functionality - Virtualised Infrastructure Manager (VIM), VNF Manager (VNFM), and NFV Orchestrator (NFVO). At the lowest abstraction layer, the VIM controls and manages the NFVI resources on top of which VNFs are deployed and executed. At the intermediate abstraction layer, VNFM performs management actions over VNF instances, such as the lifecycle management of the VNFs (e.g. instantiation, scaling, healing, termination, performance and fault management). Finally, there is the NFVO.

III. Network Management System (NMS)

The NMS domain focuses on traditional (i.e. non-virtualization-related, vendor-agnostic) tasks. Unlike NFV MANO, which is focused on NS (and VNF) management and orchestration at the virtualized resource level, NMS focuses on application-aware NS (and VNF) configuration and management, and is responsible for the deployment and maintenance of PNFs.

Chapter 2

Network Slicing Concept and Requirements

This chapter contains architectural references to concepts that research facility sites utilize to implement 5G services through network slicing. These are mainly based on 3GPP specifications, but also take into consideration work in other SDOs for vertical requirements and the 5G evolution.

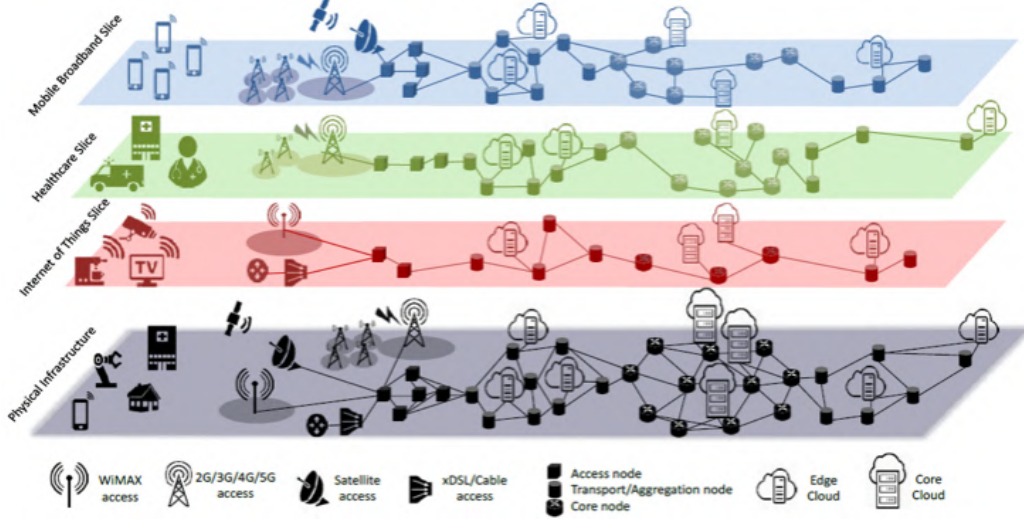
2.1 Network Slicing as a Concept

5G networks, in combination with network slicing, deliver business customers connectivity and data processing tailored to the specific business requirements that adhere to a Service Level Agreement (SLA) agreed with the network operator. The customisable network capabilities include data throughput, quality, latency, reliability and security. From a mobile operator's point of view, a network slice is an independent end-to-end logical network that runs on a shared physical infrastructure, capable of providing a negotiated service quality.

As previously mentioned in Section 1.3.1, we can define network slices as E2E logical networks running on a common underlying (physical or virtual) network, mutually isolated, with independent control and management as presented in Figure 2-1 and which can be created on demand. Such self-contained networks must be flexible to

simultaneously accommodate diverse business-driven use cases from multiple clients on a common network infrastructure, yet resilient enough to meet KPIs.

Figure 2-1: Network Slices on a Common Underlying Network. Source: [1]



The possibility to create on demand and in a programmable fashion, cost-efficient E2E network slices and dedicate them for the dynamic provisioning of diverse services is seen as a vital feature of 5G. In this vein, efforts are ongoing towards developing a 5G mobile system capable of deploying network slices of varying forms and sizes. Network slicing builds on top of the following seven main principles that shape the concept and related operations [25]:

- **Automation:** enables an on-demand configuration of network slicing without the need of fixed contractual agreements and manual intervention.
- **Isolation:** is a fundamental property of network slicing that assures performance guarantees and security for each tenant even when different tenants use network slices for services with conflicting performance requirements.
- **Customization:** assures that the resources allocated to a particular tenant are efficiently utilized in order to meet the respective service requirements.
- **Elasticity:** is an essential feature related with the resources allocated to a particular network slice, in order to assure the desired SLA under varying resources

and network conditions (e.g. the changes of network traffic or workload).

- Programmability: allows third parties to control the allocated slice resources (i.e. networking and cloud resources) via open Application Programming Interfaces (APIs) that expose network capabilities facilitating on-demand service-oriented customization.
- End-to-end: is an inherent property of network slicing for facilitating a service delivery all the way from the service providers to the end-user. Such a property stretches across different administrative domains and through that, unifies various network layers and heterogeneous technologies (e.g. RAN, core network, transport and cloud).
- Hierarchical abstraction: is a property of network slicing that has its roots on recursive virtualization, wherein the resource abstraction procedure is repeated on a hierarchical pattern with each successively higher level, offering a greater abstraction with a broader scope. In other words, the resources of a network slice, allocated to a particular tenant, can be further traded either partially or fully to yet another third party.

The network slicing process is broadly broken down into three main layers, namely the Service Instance layer, the Network Slice Instance (NSI) layer, and the Resource layer. Each service instance reflects a service provided by a vertical segment, application provider or mobile network operator. The NSI represents a set of resources customized to accommodate the performance requirements of a particular service and may contain none, one or a number of different sub-network instances. A sub-network instance can be a network function or a sub-set of network functions or resources realizing a part of an NSI.

3GPP refers to Network Sub-Slice Instance as NSSI and states that in order to be E2E, an NSI needs to consist of:

- NSSI RAN: an instance from a given (radio) access network slice subnet.

A radio access network slice subnet consists of one or more RAN nodes (i.e. eNB nodes when using LTE, and gNB nodes when using NR), each embedding and delivering full radio access functionality to interact with the UE over the radio interface. To introduce modularity and support for different deployment options, different initiatives have suggested the need to functionally split a RAN node into three types of units: Radio Unit (RU), Distributed Unit (DU), and Central Unit (CU), briefly mentioned in Section 1.3.1 as "RAN decomposition". For the most representative case, a RAN node may consist of a CU, each serving one or more DUs, each in turn serving one or more RUs.

- NSSI Core Network (CN): an instance from a given core network slice subnet.

A CN slice subnet includes the functions from the 3GPP CN that are in charge of providing connectivity between RAN nodes and the Data Network, as well as any other 3rd party AF providing value-added functionality.

- A Virtual Network providing connectivity within and across the two former NSSIs.

Focused on provided connectivity within the NSI in an E2E manner, the virtual network consists of a set of virtual links that may span across various network segments.

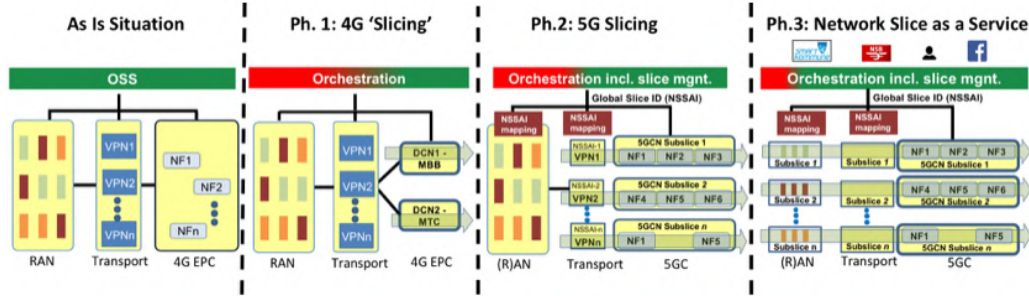
NSSIs resulting from the instantiation of the above-referred network slice subnets can be flexibly combined to provide different behaviors and performance levels, and hence different NSIs.

2.1.1 Roadmap of Network Slicing

The implementation of network slicing needs to be gradual due to the evolution approach followed by the 3GPP 5G architecture in Rel. 15 and 16. Based on the

network slicing roadmap proposed in [2], some of the most important phases expected are presented.

Figure 2-2: Roadmap of Network Slicing. Source: [2]



"As-Is Situation" represents the current scenario where no network slicing and a single core network instance are present. In this scenario, QoS differentiation is offered at the Bearer level and Access Point Names (APNs) are used for traffic separation while some form of isolation is implemented via a Virtual Private Network (VPN). Traffic can be separated for different customers but E2E Quality of Service (QoS), provisioning and the flexibility to have tailored network for vertical use cases cannot be implemented in an efficient manner.

Phase 1 is referred as "4G Slicing" which allows some level of traffic separation for overload mitigation and operational flexibility independent of other traffic.

Phase 2 will reference the specifications provided by the 3GPP Release 15 Standalone (SA), whose general concepts are outlined in Section 1.3.1. This phase is denominated as "5G Slicing" and consists of 5GCore. RAN and Transport layers support slicing via Network Slice Selection Assistance Information (NSSAI), with the latter leveraging VPNs, each mapped according to said NSSAI. Orchestration includes slice management but is not fully automated.

Phase 3 is labelled "Network Slice as a Service" (NSaaS) and is characterised by a fully automated network slicing operation. Unlike Phase 1 and 2 in which RAN and transport are not sliced, this phase involves slicing of the RAN and TN. In addition, it is expected to be able to host 3rd party NFs, which enables new business opportunities.

2.2 Network Slicing Types

The International Telecommunication Union (ITU) [26] and 5G Infrastructure Public Private Partnership (5G-PPP) [5] have identified three broad use case families: enhanced mobile broadband, massive machine-type communications, and critical communications. Differences between these use cases amount to a set of heterogeneous and often contradictory requirements that cannot be satisfied by a one-size fits-all architecture. Diverse use cases need to be mapped to suitably tailored network structures. With this in mind, the architectural proposals for 5G aim to accommodate these use cases. The following sections describe how network slicing may utilise resources in order to serve these main use cases, while simultaneously respecting their requirements.

2.2.1 Enhanced Mobile Broadband (eMBB)

eMBB is a set of services characterized by their need for high capacity links. This service type aims at supporting performance requirements of high data rates and high traffic densities. It is expected that the mobile capacity should increase 100 times over the capacity offered by current state of the art 4G systems to meet these requirements in urban user densities. Users should be able to achieve high speeds even at challenging scenarios such as on-the-go content streaming, while being at the cell edge and in crowded areas. In order to deliver NSaaS that provides this service, the CSP will place an emphasis on providing specific performance guarantees for KPIs in the following areas: bandwidth, throughput, coverage and stability of connection.

Some eMBB services/applications will put more stress on the network from a User Plane (UP) traffic capacity point of view than others and thereby are more interesting from a Packet Core capacity perspective. This mandates specifying characteristics based on the need of each eMBB service, Table 2.1 [27] describes the performance requirements for high data rate and traffic density scenarios defined by 3GPP SA1 [28]. Depending upon the transmission delay, there is consideration on distributing User Plane on the Access Layer while hosting the rest of the functions centrally.

Table 2.1: eMBB Scenarios and Requirements Identified by 3GPP

Scenario	Experienced data rate		Area traffic capacity		User density	Activity factor	UE speed	Coverage
	Download	Upload	Download	Upload				
Urban macro	50 Mbps	25 Mbps	100 Gbps/km ² ₄	50 Gbps/km ² ₄	10 000/km ²	20%	Pedestrians and users in vehicles (up to 120 km/h)	Full network ¹
Rural macro	50 Mbps	25 Mbps	1 Gbps/km ² ₄	500 Gbps/km ² ₄	100/km ²	20%	Pedestrians and users in vehicles (up to 120 km/h)	Full network ¹
Indoor hotspot	1 Gbps	500 Mbps	15 Tbps/km ²	2 Tbps/km ²	250 000/km ²	²	Pedestrians	Office and residential ¹ ³
Broadband access in a crowd	25 Mbps	50 Mbps	3.75 Tbps/km ²	7.5 Tbps/km ²	500 000/km ²	30%	Pedestrians	Confined area
Dense urban	300 Mbps	50 Mbps	750 Gbps/km ²	125 Gbps/km ²	25 000/km ²	10%	Pedestrians and users in vehicles (up to 60 km/h)	Downtown ¹
Broadcast-like services	Maximum 200 Mbps (per TV channel)	N/A	N/A	N/A	[15] TV channels of [20 Mbps] on one carrier	N/A	Pedestrians and users in vehicles (up to 500 km/h)	Full network ¹
High-speed train	50 Mbps	25 Mbps	15 Gbps/train	7.5 Gbps/train	1 000/train	30%	Users in trains (up to 500 km/h)	Along railways ¹

¹ For users in vehicles, the UE can be connected to the network directly, or via an on-board moving base station.

² A certain traffic mix is assumed; only some users use services that require the highest data rates [2].

³ For interactive audio and video services, for example, virtual meetings, the required two-way end-to-end latency (UL and DL) is 24 ms while the corresponding experienced data rate needs to be up to 8K 3D video [300 Mbps] in uplink and downlink.

⁴ These values are derived based on overall user density. Detailed information can be found in [10].

2.2.2 Ultra-Reliable Low Latency Communication (URLLC)

URLLC is a set of services with strict latency and reliability requirements. Basically, this service type aims at supporting performance requirements of low-latency and high-reliability while maintaining availability. Reliability increase can be achieved by dedicating appropriate resources to signalling, higher redundancy encoding and retransmissions. The above results in increased latency, proportional to packet size. If bandwidth requirements are low, by reducing packet size, low latency and high reliability simultaneously can be achieved, while reducing transmission times.

In order to guarantee the service is delivered at the required performance target, the CSP needs to prioritise resources to the URLLC slice, especially for mission-critical applications. It also needs to dynamically scale slices in the case of possible service degradation and may achieve high reliability by redundant transmission in the User Plane. URLLC slices will be provided by CSP for real-time control and feedback for the customer.

From a communications perspective, two factors are of utmost importance and shall impact the definition of URLLC slice architecture in each Facility Site:

- I. Latency: Sub-10ms E2E latency is required for all use cases that can be mapped to the URLLC network slice. Note that many of the applications would require even sub-5ms latency, as seen on Table 2.3.
- II. Reliability: The reliability is specified by the failure probability of packets (packet error rate) which are not successfully delivered to the receiver within the latency bound, as they are either erroneous, lost or arrive too late.

Table 2.3: URLLC Scenarios and Performance Identified by 3GPP

Scenario	Max. allowed latency ²	Survival time	Service availability ³	Reliability ³	User expe- rienced data rate	Payload size ⁴	Traffic density ⁵	Connection density ⁶	Service Area dimension ⁷
Discrete automation	10 ms	0 ms	99.99%	99.99%	10 Mbps	Small to big	1 Tbps/km ²	100 000 km ²	1000 x 1000 x 30 m
Process automation - remote control	60 ms	100 ms	99.9999%	99.9999%	1 to 100 Mbps	Small to big	100 Gbps/km ²	1 000/km ²	300 x 300 x 50 m
Process automation - monitoring	40 ms	100 ms	99.9%	99.9%	1 Mbps	Small	10 Gbps/km ²	10 000/km ²	300 x 300 x 50 m
Electricity distribution - medium voltage	40 ms	25 ms	99.9%	99.9%	10 Mbps	Small to big	10 Gbps/km ²	1 000/km ²	100 km along power line
Electricity distribution - high voltage	5 ms	10 ms	99.9999%	99.9999%	10 Mbps	Small	100 Gbps/km ²	1 000/km ²	200 km along power line
Intelligent transport systems	30 ms	100 ms	99.9999%	99.9999%	10 Mbps	Small to big	10 Gbps/km ²	1 000/km ²	2 km along road

¹ Currently realised via wired communication lines.

² This is the maximum end-to-end latency allowed for the 5G system to deliver the service in the case the end-to-end latency is completely allocated to the 5G system from the UE to the Interface to Data Network.

³ Communication service availability relates to the service interfaces, and reliability relates to a given system entity. One or more retransmissions of network layer packets may take place in order to satisfy the reliability requirement.

⁴ Small: payload typically (\leq) 256 bytes.

⁵ Based on the assumption that all connected applications within the service volume require the user experienced data rate.

⁶ Under the assumption of 100% 5G penetration.

⁷ Estimates of maximum dimensions; the last figure is the vertical dimension.

⁸ In dense urban areas.

⁹ All the values in this table are example values and not strict requirements. Deployment configurations should be taken into account when considering service offerings that meet the targets.

2.2.3 Massive Internet of Things (mIoT)

Heterogeneous access allows IoT devices to connect using whatever radio or fixed connectivity is available from both the device and the network. 5G-NR utilizes spectrum bands for both low and very high bandwidth, as well as co-existence with cellular and non-cellular access networks but also other un-licensed radios. The operator is thus required to create and maintain the various instantiated IoT slices to offer services and enable the following scenarios:

- Enable individual needs of an IoT slice in terms of bandwidth, latency.
- Distribute efficiently resources of network slices involving resources at the Edge. Operators can offer resources and services at the edge of the network for IoT vendors to deploy (virtualised) applications and store collected data.
- Enable IoT Vendors to become “virtual operators” and build their own virtual network on top of a mobile operator and manage resources and services across a 5G network.

The mIoT service type aims at supporting performance requirements of large number and high density of IoT devices. Some key aspects of these devices are their fully automated nature and the minimal interaction with humans, sending small amounts of information to the cloud. Some of these IoT devices are expected to be installed in remote, hard to access locations with bad coverage. Thus, a key requirement is the ability to use of very robust modulation and coding schemes. Median battery life is around a decade and the overall device cost should be low. Finally, their number is expected to explode in the coming years, thus base stations should be able to handle the additional overhead in signalling. In order to deliver NSaaS that provides this service, the CSP will place an emphasis on providing specific performance guarantees for KPIs such as connection density (1,000,000 devices per km²) and coverage (95%) [55] while also guaranteeing security of the slice. Consumer privacy and security of the information proves of extreme importance for this type of service.

2.3 Network Slicing Requirements

E2E network slicing inevitably adds an additional degree of complexity in the ecosystem, where each operator needs to deploy network slices internally and also facilitate the E2E deployment across several facility sites. Following there is a list of initial network slice requirements to be considered:

- **Orchestration:** Slice-based services will require, not only the deployment across several administrative and operational domains, but also the integration of different technology domains. Their E2E nature requires orchestration of different computing and transport technologies.
- **Operation:** Each slice must behave as a dedicated network while sharing underlying resources, physical and virtual. Mechanisms need to be defined in order to abstract each customer slice.
- **Scalability:** The application of different Service Level Agreement (SLAs) on the offered capabilities of management, control and customization of slices will directly impact scalability. Different levels depend on whether a network slice is inter or an intra-site.
- **Service layer:** The interaction with the vertical customers has an important requirement in the definition of the proper abstraction and templates to have a consistent service portfolio supported by different slice classes.
- **Security and Isolation:** Different network slices will be sharing the same infrastructure. The main requirement from the Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) is logical isolation of slices via appropriate networking, security grouping and multi tenancy implementation.

2.3.1 Network Slicing Use Cases

5G systems will support a wide range of use cases to vertical industries and their applications that wish to take advantage of their capabilities. These use cases will be supported by communication service providers (CSPs) who may offer Network Slice as a Service (NSaaS) as a means of delivery. NSaaS is where the CSP offers network slices to their customers directly, rather than simply using slicing as an underlying capability to support other communication services. Each NSaaS will have a service description and SLA which will define the nature of the interaction between the CSP and said consumer. Slice providers and consumers can be described in terms of the business models between them and may fall into one of the following four categories:

- B2B (Business to Business): in which the CSP provides service to another business entity.
- B2C (Business to Consumer): in which the CSP provides service to an individual user.
- B2H (Business to household): in which the CSP provides service to a collection of individual users who form a single household and thus have some common characteristics.
- B2B2X (Business to Business to Everything): in which the CSP provides service to other business (such as other CSPs) who then offer a further service to their customers.

From an architecture perspective, these four business models utilise one of two arrangements for the way in which the provider integrates their capabilities in order to deliver the end service.

2.4 Isolation on Network Slicing

The use of a single, shared multi-domain network infrastructure makes isolation a key requirement in supporting network slicing. Isolation across NSIs ensures that congestion, failures, attacks and lifecycle-related events (e.g. scaling in/out) of one NSI does not negatively impact other NSIs. Isolation is a broad concept embracing many dimensions, and will be studied from three main perspectives: performance, management and security.

As previously mentioned, 5G network infrastructure is built out of a set of radio, computing, storage, and connectivity resources that can be flexibly combined to set up different NSIs. These resources can be arranged into three main resource domains:

- **Radio Resource domain:** consists of one or more antennas hosting RU functionality. A given antenna handles the operation of one or more cells, each assigned with specific radio resources to serve the UE that are within the coverage area. These radio resources include multiple Radio Frequency (RF) carriers distributed across one or more spectrum bands which are consisted of several sub-carriers arranged into time-frequency resource grids known as Physical Resource Block (PRB).
- **Data Center (DC) domain:** consists of all the computing, storage, and networking resources that are defined within a DC. Some are bundled together in physical, purpose-built devices ready to host PNF instances, while others can be logically partitioned with the help of an abstraction layer. It is possible to have geographically distributed hierarchies, including edge and central DCs.
- **Transport Network (TN) resource domain:** consists of multiple physical TN links providing connectivity throughout the entire hierarchical network infrastructure. These links can be similarly abstracted and partitioned, resulting in virtual links. Some of the TN links will be defined within each administrative domain, providing connectivity between RAN nodes and DCs, while others span across different domains, connecting central DCs from different Facility Sites.

The resource (i.e. Radio, DC and TN) and management domains (i.e. 5G-RAN Controller, 5G-Core Controller, Transport controller, NFV-MANO, and the E2E Services Operations and Management) present the impact the three isolation dimensions have, without considering the presence of administrative domains. This means that the resource and management domains could be from the same or different sites.

Isolation in terms of Performance

Performance isolation in the network slicing context means that service-specific performance requirements are always satisfied on each NSI, regardless of the congestions and workloads of other NSIs running parallel. The level of performance committed for a NSI needs to be assured in an E2E manner (i.e. from the UE to the data center where the last NSI component is allocated), across all the NSSIs that take part in the NSI, including the connectivity between them.

I. Radio Resource Domain

Different NSSIs may share the limited set of radio resources available, mandating the definition of isolation mechanisms if performance isolation is desirable among NSIs. This sharing is motivated by the fact that NSSIs are served within the same coverage area of a given antenna, and hence need to make a shared usage of the radio resources operated by this antenna. The isolation mechanisms vary depending on the scenario considered and include traffic isolation (i.e. avoid situations when traffic overload in one NSSI negatively affects the behavior of the rest of NSSIs) and electrical isolation (i.e. avoid mutual interference at the radio interface between the transmissions of different NSSIs) [29]]. Spectrum planning is the one that has the highest level of maturity, and hence can be seen as a feasible solution to be adopted for 3GPP's Network Slicing Architecture.

II. Datacenter Resource Domain

DCs include physical hardware that can be used to accommodate the NFs of the different NSIs. These NFs can be either from the access part (AN NSSIs), or from the core part (CN NSSIs). The physical hardware of a DC include both commodity and purpose-built hardware, while the latter can be seen as PNF instances. Virtual resources result from the abstraction of the underlying commodity hardware, including computing, storage, and network equipment. The fact that VDUs run on a common substrate brings potential risks on performance isolation, as failure or workload increase in one VDU may decrease the performance of the rest of VDUs sharing the same substrate. In virtualized environments, two approaches can be followed for VDU hosting instantiation:

- VNF Instances from Different NSIs on Separate Hardware Nodes.

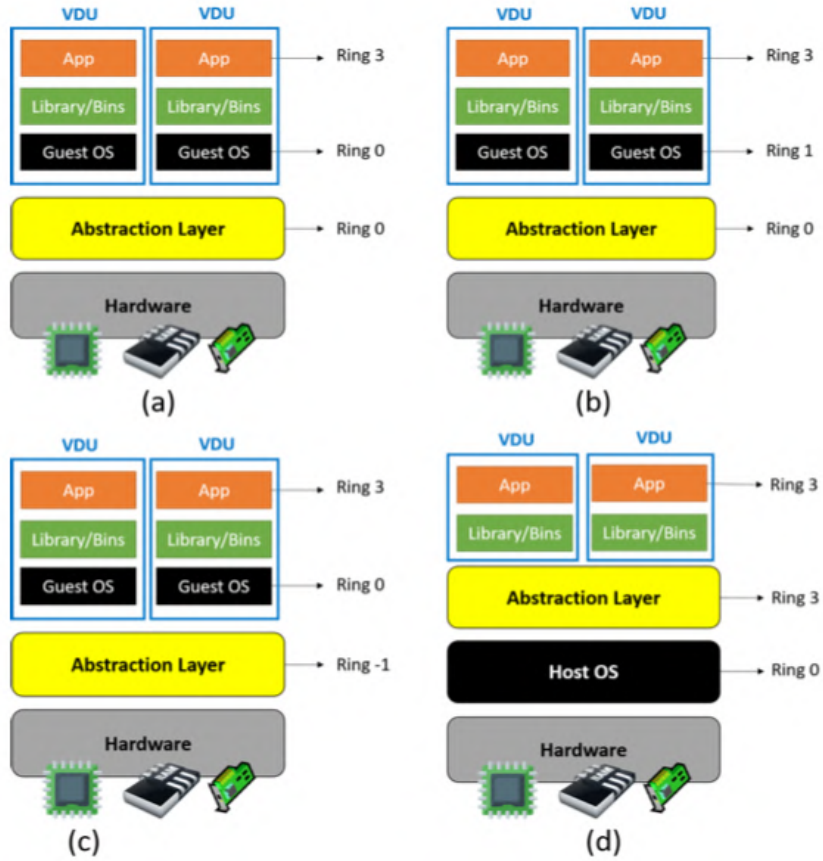
The first approach provides the most isolated environments for VDU execution, as performance decrease and hardware failures in one node only affect the VDU this node accommodates and not the rest of VDUs. For the most critical NSIs, compute nodes can be located in a dedicated availability zone, separated from any other zones in the DC. Although allocating VDUs from different NSIs in different hardware nodes means achieving very great level of isolation among them, this approach commands high resource and energy usage.

- VNF Instances from Different NSIs Executed on a Shared Compute Node.

The second approach allows for a more effective resource usage and energy efficiency, at the cost of providing less isolated environments for VDU execution. Unlike the first approach, hardware failures and decreases of performance in one node may affect all the VDUs running on this node.

The introduction of server virtualization allows definition of VDUs on top of the same compute node, providing them with separated execution environments. The isolation that each execution environment provides depends on the type of virtualization technology used. Currently, four types of virtualization technologies are available, each with different implications on the security ring-model as depicted in Figure 2-3.

Figure 2-3: : (a) Para Virtualization, (b) Full Virtualization, (c) Full Virtualization with Hardware Assist¹ and (d) OS Virtualization. Source: [3]



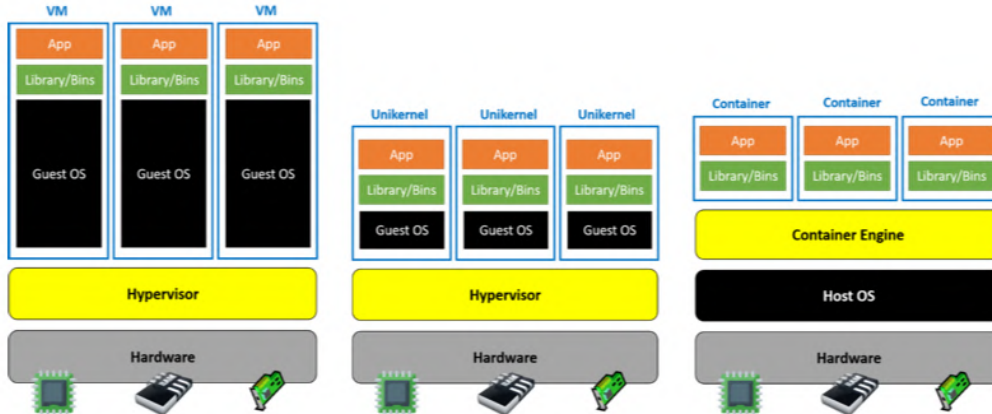
As result of the application of the four above-referred technologies, two types of VDUs are outlined: VDUs with their own guest OS and VDUs sharing the same host OS as depicted in Figure 2-4.

¹Note that full virtualization technologies with hardware assist (i.e. Intel VT and AMD-V) introduces a more privileged ring level (ring -1) to the original x86 ring model. In this new ring level, the abstraction layer uses processor extension to intercept and emulate privilege instructions, leaving ring 0 available for VDUs, whereas this would be reserved by the operating system's kernel.

Those result in two types of abstraction layer being identified:

- Hypervisor: applicable to para virtualization, full virtualization, and full virtualization with hardware assist. This abstraction layer allows the definition of VDUs, each with its guest OS. The VDUs resulting from the hypervisor are traditional Virtual Machines (VMs). However, in recent years, a new type of VDU, named "unikernel", resembles a VM but is built with a lightweight and specialized guest OS. Unlike VMs, a unikernel can only run a single process, and cannot benefit from full virtualization with hardware assist, as it contains only absolutely necessary components to run a specific application.
- Container engine: applicable to OS virtualization, this abstraction layer allows the definition of VDUs running on top of the same OS, without the possibility to host their own. These VDUs resulting from the abstraction carried out by container engine are referred to as containers.

Figure 2-4: VMs vs Unikernels vs Containers in Terms of Isolation. Source: [3]



As seen from Table 5.2, the most isolated environment is provided with a VM at the cost of less speed of operation, followed by the unikernel that provides more agility and reduced image size. Containers provided the least degree of isolation, although they allow for great agility in run-time operations. The selection of each VDU type for VNF hosting depends on the necessity of the VNF operation,

Table 2.5: on Top of which VNF Instances Run

VDU Type	Ring Level	Isolation provider	Image size	Operation agility
VM	Level 0 or -1	Hypervisor	Large	Low
Unikernel	Level 0	Hypervisor	Small	Medium
Container	Level 3 enforced at Level 0	Host OS	Medium	High

and the degree of isolation required. The use of separate physical appliances and VMs are known to be matured solutions for providing isolation in the DC resource domain. Unikernels, although bringing some benefits compared to VMs, need to reach more maturity and popularity before considering them as firm candidates to replace VMs. Finally, containerization technology is gaining some maturity in the last years, empowered by the adoption of Service Based Architecture (SBA) for 3GPP's 5GC. The adoption of this solution is expected for 3GPP Rel. 15 and 16 SA.

III. Transport Network (TN) Resource Domain

Virtual links resulting from the abstraction of the underlying TN links can be arranged into virtual networks, each providing intra-NSI connectivity (i.e. connectivity across P/VNF instances from the NSSI RAN and NSSI CN) with performance guarantee. Both wired and wireless TN technologies can be used, with the former including optical and gray optics, whereas the latter encompasses both wireless (i.e. packet microwave and mm-wave) and satellite solutions as Physical Layer (L1) technologies. It is to be expected that, in some cases, the common TN infrastructure will be shared between multiple 5G services. Network slicing isolation is thereby achieved by the introduction of some form of encapsulation, for the differentiation of virtual links operating on top of a shared substrate and the selection of corresponding Layer technologies towards the realization of these virtual networks. The introduction of encapsulation enables different virtual networks to be deployed on top of physical links that compose

the underlying TN infrastructure, differentiated by the degree of isolation they provide.

Soft isolation is implemented by statistically multiplexing the traffic from two or more VNs onto a common circuit switched connection using a packet technology (e.g., Ethernet Virtual Local Area Network (VLAN), Multiprotocol Label Switching (MPLS) tunnel). Isolation is enforced at the packet layer, permitting sharing of resources across different virtual links, since resources are only dedicated on a temporary basis. This allows greater statistical multiplexing on network resources in different links and leads to better economy at the cost of opening up the possibility to suffer from congestion in the network, in all the virtual links making use of those resources.

Hard isolation is implemented by providing independent circuit switched connections for the exclusive use of one VN (e.g. Time Division Multiplexing (TDM) based approaches, assignment of different fibre cables etc.). Hard isolation solutions allow reaching high degree of isolation between virtual links, at the expense of allocating dedicated resources on a long term and end-to-end basis. This means that the full cost of the resources must be borne by the NSI whose VL is allocated with those resources.

Isolation in terms of Management and Orchestration

Isolation at the MANO level assumes the existence of tenants at the orchestration level (e.g., Multitenancy in OSM, ONAP). Each tenant has an exclusive view and management of specific VNFs, NSs and the isolated resources at the RAN, Core and TN, enabling the management and orchestration of a particular slice. A default master tenant usually called “administrator”, is in charge of the management and orchestration of all slices and the physical and logical resources that comprise them. Multitenancy at the VIM level articulates the isolation mechanisms defined in compute, storage and tenant networks, allowing for a tenant to only see and administrate the resources that are assigned to it. One important concept is the shared part. It

is important to employ the isolation mechanisms mentioned in previous sections (e.g. RAN slicing at the carrier and PRB level, transport slicing at the wavelength, time slot, logical bandwidth level). From previous sections, it is apparent that shared resources is a critical part of operation. As to be expected, isolation cannot be guaranteed in all cases (i.e. when sharing same antenna, cable and compute node across multiple slices). Tenants may demand isolation of resources allocated to management regarding isolation of compute and storage resources. This makes it possible to obtain all properties of virtualized computing systems, as mentioned in the previous section, enabling tenants with different levels of isolation at the management level.

Isolation in terms of Security

Security as a term, encompasses different aspects in the provisioning, operation and use of any network services. For that reason, slices must adhere to general security properties, based on the three dimensions of a network service. These dimensions have obvious implications to slice isolation and namely are:

- **Protection:** in what implies that a service is immune to attacks from any adversary attempting to distort in any means its functionality or features. Isolation in terms of protection ensures each slice is immune from attacks on other slices or parts of the external network.
- **Privacy:** requiring that no data from any actor in the service (provider, consumer, end user) is accessed by unauthorized parties. Isolation in terms of privacy ensures each slice has adequate mechanisms to protect integrity and confidentiality and sensitive information (i.e. configuration, management, subscriber, and accounting information) from being accessed by other entities.
- **Accountability:** which translates into enforcing proper authentication, authorization and accounting, so no action is performed without verifying the identity, access rights and properly maintaining a record for further auditing.

Chapter 3

5G Verticals Innovation Infrastructure

The 5G-VINNI concept is to develop an E2E 5G facility that can be used to primarily demonstrate the practical implementation of infrastructure to support the key 5G KPIs and to accelerate the uptake of 5G in Europe by lowering the entry barrier for vertical industries to pilot use cases dependent upon those KPIs. However, 5G-VINNI is not intended to be simply a group of interconnected test facility sites as seen on Figure 3-1 [4] – it is underpinned by principles that will allow for highly dynamic and flexible network architectures, service deployment and testing, that will create new technical and commercial service deployment models. These will in turn drive inter-facility interconnection to enable virtualized functions from the network and service layer to be called upon from any facility, with complete location agnosticism – a truly cloud-based network instantiation that has no functional boundaries, implemented across multiple facility sites.

3.1 Architectural Principles

5G-VINNI is underpinned by key architectural principles identified in the 5G-VINNI proposal document. These are:

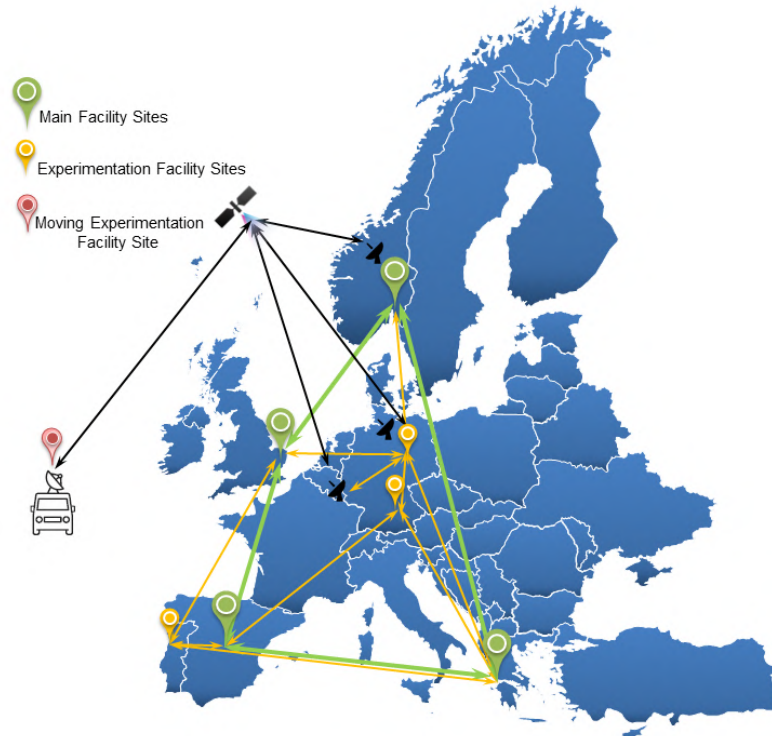
- Common and consistent sets of capabilities.

All 5G-VINNI facility sites support a core set of functional components and capabilities, meaning that partners that perform trials can be assured a baseline set of capabilities at any facility.

- Interworking between facility sites.

A key strength is the ability to perform E2E testing between sites. This mandates a common set of interfaces to be supported by all facility sites to allow interaction between control plane systems and the flow of traffic.

Figure 3-1: 5G-VINNI Facility with Main and Experimentation Sites. Source: [4]



- Optional elements to support specific use cases, testing and KPIs.

In order to achieve this, additional functions may be supported at sites that require them, in order to meet proprietary requirements.

- Support for Network Slicing. A key component of 5G-VINNI is the support of network slicing. This is essential for each site in order to allow simultaneous testing of multiple use cases on common infrastructure.
- Ability for each site to evolve during the project.

5G has an underlying principle of networks built using VNFs. The iterative nature of functional capabilities permits quick and with frequency changes to the network. 5G specifications is currently a work in progress and so, within the timeline of the project, standards and state-of-the-art implementations will potentially shift significantly. Therefore it is fundamental that 5G-VINNI facility sites are designed in such a way so as to allow implementations to keep pace with technological evolutions. This will maintain the 5G-VINNI facility's relevance to vertical industry partners' testing requirements throughout the duration of the project.

- Openness and accessibility of the framework.

The design of the each facility aims to allow the flexibility and accessibility for vertical industries to run and test novel pilot use cases.

With virtualisation, orchestration and network slicing all included as underpinning architectural principles and identified as the basis for implementation as early as 5G-VINNI Release 0, it is clear that a multi-layered implementation is expected with network management and orchestration as disciplines that are pivotal for the instigation of network functions and network slices within one or multiple facility sites.

3.2 Template Facility Architecture

In the lower layer is the infrastructure, VNFs and PNFs in the RAN and Core infrastructure for the transport domain, which are expanded upon on Sections 3.2.2 and 3.2.3 respectively. Above the Network Domains, NFV-MANO focuses on virtualization-specific tasks (i.e. management at the virtualized resource level) while domain controllers operate on non-virtualization-related operations (i.e. management at the application level). This architecture can be implemented in a range of ways, which allows individual facility sites to realize a network that can deliver a set of KPIs, whilst adhering to the principles of 5G-VINNI as a whole.

3.2.1 User Equipment

For the most part, end-user devices are outside of the scope of 5G-VINNI as devices tend to be tied to vertical use cases and as such are in the domain of prospective ICT-19 projects. It is clear that for a device to connect to a 5G-VINNI facility site's infrastructure, it should support reciprocal interface requirements mandated upon the RAN. Where required, 5G-VINNI facility sites may support devices that are specific to use cases and hence need to meet specific requirements. These include (but are not limited to):

- LTE-M
- Narrowband IoT (NB-IoT)
- Fixed Wireless Access (FWA)
- Vehicle-to-Everything (V2X)
- Experimental functionality for future capabilities (R16, R17 and beyond)
- Air interface and connectivity testing.

3.2.2 Radio Access Network (RAN)

Initial 5G-VINNI RAN architecture is aligned with what is being planned for early launches of commercial 5G networks in Europe. This will create a 5G research environment that, in some ways, closely reflects functionality and performance offered by early commercial 5G networks. Some of the 5G-VINNI facility sites may implement SA architecture from start while other facilities may align with the development related to early commercial networks then migrate from SA architecture during the later Releases of the 5G-VINNI project.

The 5G-VINNI RAN will initially offer service opportunities in the 3.5 GHz frequency band (3GPP n77/n78, referred to as "mid-band") and in subsequent phases also in the 26 GHz frequency band (3GPP n258, referred to as "high-band"). Specific details of what will be supported at the different facility sites depends on NSA or SA implementation as well as spectrum availability and device capabilities. Most notable features that need to be supported are:

- LTE – NR Dual Connectivity

Dual Connectivity (EN-DC) is a mandatory part of 5G NSA operation that allows early introduction of 5G by overlaying NR to existing LTE networks connecting to the 5G enabled CN.

- Uplink/Downlink Decoupling

5G-VINNI NSA architecture will support uplink/downlink decoupling allowing LTE spectrum (using lower frequency bands than NR) with superior coverage to be used for uplink traffic while NR spectrum with superior peak data rate and latency is used for downlink. By combining the high speed and low latency of the NR downlink with the larger coverage and high reliability of the LTE uplink, the coverage versatility provided by NR spectrum is significantly extended.

- LTE – NR Aggregation

Increases user peak bitrates and app coverage by combining one or more LTE carriers with one or more NR carriers to allow users to experience higher speeds in the network.

- Multiple Input Multiple Output (MIMO)

Massive numbers of antenna elements is a key feature for NR systems. Especially for high frequency bands, the introduction of multi-antenna arrays enables numerous benefits. Most notably, beamforming improves data transmission coverage and link performance, while interference is minimized further improving throughput for cell-edge users.

- Numerology support

3GPP TR 38.912 [30] Sub-Carrier Spacing (SCS) enables RAN systems to deliver lower latency compared to legacy commercial LTE networks.

- NR Carrier Bandwidth

Mid and low frequency carrier bandwidth is naturally narrower than high frequency bands. As a result, up to 100 MHz carrier bandwidth can be supported for mid-band versus 50 to 100 MHz in high frequencies¹.

- Modulation

64 Quadrature Amplitude Modulation (QAM) in uplink and downlink shall be supported, with some experimental uses of 256QAM for downlink transmission.

- Network slice aware RAN

For RAN, a network slice provides additional information on the policy for handling traffic in addition to the regular QoS for radio bearers.

- RAN integration with NR Standalone (SA) Core architecture

When 5G-VINNI facility sites implement SA architecture, RAN functionality shall be adopted to fulfil applicable requirements. This requires that RAN and UE progression needs to be aligned with the introduction of 5G Core, as this introduces new interfaces and protocols.

- Dynamic spectrum sharing

Radio access node will dynamically determine if an NR or LTE user is to be served, allowing NR to use the same legacy frequency bands as LTE.

3.2.3 Core and Transport Infrastructure

As previously discussed, 5G System as defined in 3GPP Rel. 15 can be realized either by 5G-NR RAN connected to EPC (NSA) or 5G-NR connected to 5G Core (SA). While EPC is used in NSA as an evolution of the current LTE Core network, 5G Core is a service-based architecture with new functional nodes.

¹Actual implementation needs to conform with spectrum availability at the different 5G-VINNI facility site regions.

Transport is the part of the network that comprises the intermediate links between the CN or backbone and the sub-networks at the "edge" of the hierarchical network. Backhaul plays a vital role in mobile networks by acting as the link between RAN and Core.

Wireless backhaul

Wireless technologies operating at microwave and millimetre-wave frequencies, allow operators freedom regarding base station and small cell wireless-based backhaul.

Packet microwave or mm-wave technologies are increasingly prevalent elements of Heterogeneous Networks (HetNets), offering ample IP capacity for backhaul. Point-to-Point (PtP) systems can also be used as relays in a multihop backhaul network. Various topologies can be used in a backhaul deployment, depending on each specific network's need (e.g. point-to-point line, tree structure, mesh, ring or a combination). Reliability can be achieved by using redundancy in equipment and appropriate topologies. In 5G-VINNI, the wireless transport network can provide a versatile solution for backhaul as a fibre substitute, as well as transport for IoT applications.

Optical Backhaul, Midhaul and Fronthaul

In 5G-VINNI an optical transport solution, can be used to support backhaul, midhaul and fronthaul alternatives. Different functions in the gNB result in different demand for bandwidth and latency, thus requiring different transport alternatives.

The mid and front-haul options are currently under development (outside the scope of 5G-VINNI) and are foreseen to be integrated in the facility infrastructure along the course of the project.

Satellite backhaul

Mobile Network Operators (MNOs) likely perceive satellite backhauling services as islands of costly, difficult to integrate and inflexible connectivity. However, with the significant recent advances in satellite technology (e.g., High-Throughput Satellite)

satellites can deliver cost-effective high-performance solutions to unserved and under-served areas.

Backhaul connectivity

Any backhaul connectivity deployment includes an edge node and a central node connected using a backhaul link, seen as a transport layer for the messages between the edge and the central node.

The UE connects to a local terrestrial access network which is represented in this case by a 5G gNB. In order to assure the connectivity to the local terrestrial access network through the different backhauls, the 5G Core Network is deployed with a functional split between the edge and central nodes. One of the NFs associated to the backhaul connectivity architecture is the edge router which acts as a Software-defined Wide Area Network (SD-WAN) router. Data traffic is then steered according to a set of routing policies over the available backhauls previously mentioned.

3.2.4 Management and Orchestration

Management and Orchestration is a key capability for provisioning network slices. Once the underlying programmable infrastructure has been established, there is a need for components that are able to manage and orchestrate the resources within and across domains [5]. These functions are:

NFV MANO

From a resource management point of view, services are deployed and operated as NFV NSs. With NFV MANO, facility operators are able to deploy and execute services in the NFVI with great agility and flexibility. The NFVI in 5G-VINNI facility spans across the seven facility sites and consists of different PoPs for VNF deployment and execution, ranging from cell sites to regional DCs. Each facility site will have its own NFV MANO stack to manage and orchestrate NFV NSs within its administrative domain while exposing a set of SBIs and NBIs to facilitate the interaction with the rest

of facility site components. The stack deployed at each facility site has three types of functional blocks (NFVO, VNFM, and CIM) and four data repositories (NS Catalog, VNF Catalog, NFV Instances Repository, and NFVI Resources Repository). This guarantees interoperability across 5G-VINNI facility sites at NFV level, regardless of any implementation-specific issues. The fact that ETSI NFV defines NFV MANO stack as a reference framework not only allows network operators and vendors to implement its components using open source or vendor specific solutions but also introduce some architectural modifications.

Network-Domain Controllers

I. RAN and Core Domain Controllers

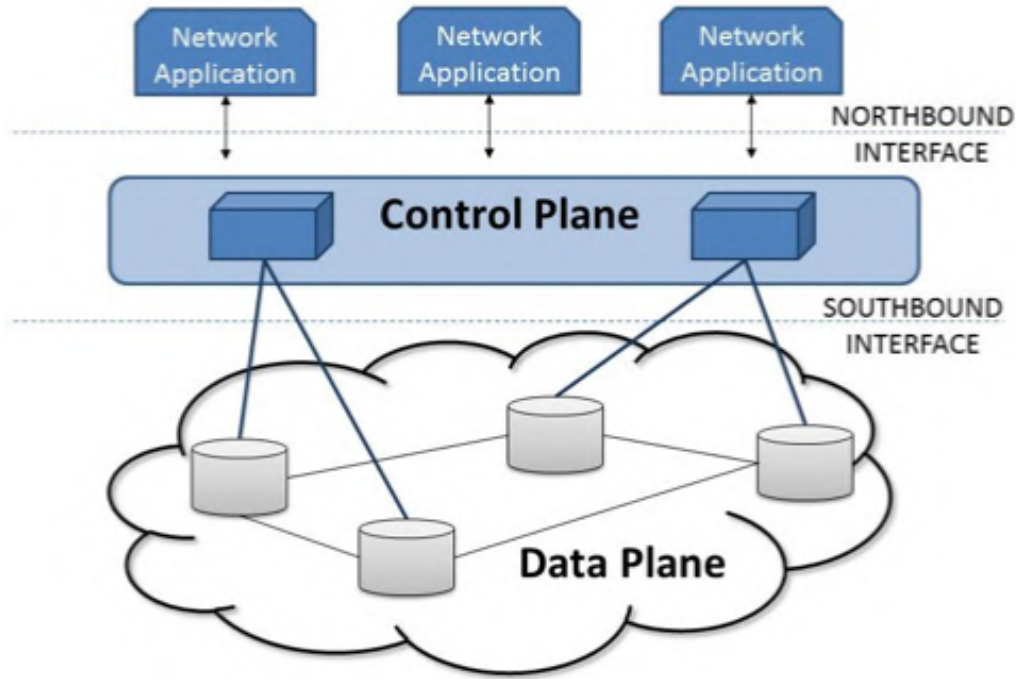
The Domain Controllers at the RAN and Core are in charge of managing the different NFs at the application level (independently of their deployment), and in general to provide control on all the non-virtualization-related operations. Core & RAN domain controllers can be mapped to the EMS function group of FCAPS (Fault, Configuration, Accounting, Performance and Security management) for the functional/application part of the underneath deployed PNFs and VNFs. In this way we can say that these domain controllers perform changes at the application level in a “vertical” way on functions “horizontally” deployed by the NFVO.

II. Transport Domain Controllers

The Domain Controllers at the transport include components such as SDN controllers or MPLS management and control components. In 5G-VINNI the Transport network will provide advanced backhaul functionality, to interconnect for instance different locations from a common site. MPLS networks for instance are well-known solutions used for several years in the transport network. MPLS allows the visualization of the E2E path instead of a hop by hop vision.

The CP and the data plane are separated from each other. In addition, the CP is logically centralized in a software-based controller defined as the “network

Figure 3-2: SDN Architecture According to IETF RFC 7426. Source: [5]



brain”, while the data plane is composed of network devices (“network arms”) that forward packets. The CP includes both northbound and southbound interfaces. The northbound interface provides a network abstraction to network applications, and the southbound interface (e.g., OpenFlow) standardizes the information exchange between the control and data planes [31].

E2E Service Operations and Management

E2E Service Operation and Management, within the 5G-VINNI architecture, encompasses the management and operations of E2E services that span multiple domains. It translates service design into configuration of resources (physical and virtualized) needed for service establishment. Using orchestration to coordinate between domain controllers and NFV-MANO, E2E services may span multiple network domains provided by different administrative entities (e.g., different network service providers or external partners). Effectively, E2E MANO connects the business needs of verticals to the resource and network capabilities exposed by management and partner

provider domains. It thereby enables orchestration and management across multiple domains, doing so by offering customer facing services northwards via standardised API's to fulfil service requirements. The E2E MANO translates business services into their corresponding component network services and requests the services from the appropriate domain. In this way the E2E MANO abstracts how the business services are realised at the resource level, thereby allowing verticals to focus on their Service Level Agreements (SLAs) rather than the service implementation itself.

3.2.5 Security, Testing and Monitoring Architecture

With the use of NFV and SDN in 5G, one of the main priorities is for a secure cloud environment between tenants and slice instances. Security needs to be present in all sub-domains of the cloud to address the current threat landscape with sophisticated actors, adhere to privacy and security regulations in the different markets and ensure isolations of tenants and network slices. The main security controls that need to be established in the cloud domain are:

- Infrastructure network zoning model that applies to all layers of the infrastructure.
- Solid multi-tenancy capabilities through the infrastructure platform.
- Security monitoring through flow based network analytics.
- Centralized logging with analytics capabilities.
- Server integrity checking and configuration auditing.

Note that there are other important security functions, such as DDoS protection, Intrusion Detection and Prevention Systems (IPS/IDS), antivirus, antimalware etc. The aim is not to cover all security requirements and features in this section. In the following, briefly described are architectural principles for the zoning model and multi tenancy.

Zone Model

The infrastructure network zoning model is fundamental in addressing the shared technology challenge in a multi-tenant platform. The defined zone model regulates how the cloud infrastructure is built and how segmentation is achieved. A security domain is an entity that defines the level of control and governance present; it does not itself have any policies, but exists to define where boundaries are present. A domain can consist of one or more security classes. Security domains defined in level of trust are: un-controlled, controlled, restricted, secure, and externally controlled domain with traffic not allowed to flow directly between non-adjacent domains. Security Zone Classes is a concept of aggregating network zones with the same exposure and security levels. Traffic from one security class to another will need to pass two different layers of security with one being logical or virtual and with the second layer being physical.

Multi-tenancy

Multi-tenancy enables separation between tenants running services in a shared environment. In a multi-tenant deployment, the resources controlled by one tenant are physically or logically separated and secured from other tenants. Multi-tenancy is a key requirement for services across both public and private cloud environments.

The testing architecture has a purpose that is primarily establishing automation services that encompasses every phase of the infrastructure deployment. Nowadays, this is considered a very important stage given the number of performance and interoperability variables at stake in modern virtualized systems. The second purpose of the testing infrastructure is to provide testing and experimentation services to the vertical service operators but also to the mobile operators themselves for network maintenance purposes. This is a fundamental step for ensuring reliability, resilience, and performance of the slice components and of the slice as a whole during the slice deployment phase, a due process before handing the slice over to the customer. Such system is used to monitor all the components, from the virtualised infrastructure to

the Quality of Experience (QoE) of the traffic carried by the network. While its presence would be highly suggested, the complexity of virtualized networks make a full monitoring extremely heavy and resource-demanding.

3.3 Interconnection and Interoperability of sites

Interconnection Between Sites

This section explains the architectural requirements in terms of connectivity between sites. As a main principle, it is important that all sites have IP connectivity with each other. For such purpose, each site shall have one or more facility site gateways (FSGW) that fulfils the following requirements. In the long term solution, two FSGWs can be preferred for increased load balancing and resilience capabilities.

- Public IP.
- Routing protocols, especially exterior gateway protocols, such as Border Gateway Protocol (BGP).
- Generic Routing Encapsulation for the establishment of GRE Tunnels.
- Internet Protocol Security (IPsec) protocol suite.

For QoS Differentiation on the connection between sites the following options are under evaluation.

- Best Effort Internet².
- SD-WAN, where the traffic can be sent over a combination best-effort Internet paths and some way of using MPLS Core or direct DiffServ-enabled traffic exchange between the operator domains.

²In the initial phase, all sites will use Best Effort internet. However, the possibility that the connection between some specific pair of sites will be upgraded gradually to MPLS-VPN and/or SD-WAN with MPLS Core is under evaluation.

This will enable SD-WAN for Assured Service Quality (ASQ) interconnection paths between sites and value added connectivity (VAC) triggered and handled on-demand for end-point traffic flows whose traffic can be steered. For the support of use cases and scenarios beyond the basic service offerings (based on best-effort Internet interconnection) the support for enterprise VPN is a viable alternative. These services can be relevant for enterprise customers managing their applications as tenants or connecting to partner enterprises that operate as tenants of one or more facility sites.

Some characteristics of the MPLS and SD-WAN concepts are:

- Classic MPLS VPN with QoS

MPLS is a very well-known connectivity solution. With MPLS, the operator can make separate bandwidth reservations for different classes of traffic and give different forwarding behaviour based on the class [32].

- SD-WAN with MPLS Core

This concept uses software-defined networking in a Wide Area Network (WAN), and is based on the traditional Software Defined Networking (SDN) principle of separating the CP from the data plane. One of the advantages that it offers is to enable the use of private WAN connections such as those based on MPLS with conventional internet access. [33, 34]

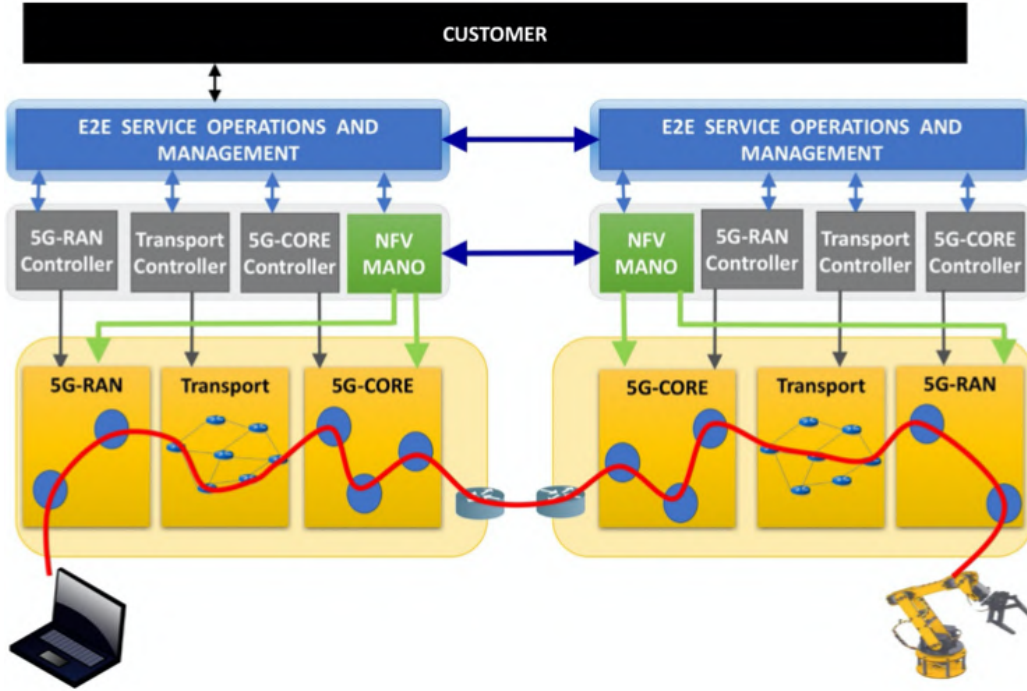
Interoperability Between Sites

Once the different 5G-VINNI sites are interconnected as presented in the previous section interoperability is in order for network services to be deployed and operated across said sites.

In order to enable such interoperability, the use of north-south bound and east-west bound interfaces are needed in order to guarantee the proper understanding of the service needs on each of the involved sites. While the arrows are here drawn directly to and from the NFV MANO in the neighbour domain this will typically be realized via inter-provider APIs³ (e.g. along the lines of 5GEx Project [35]).

³In this regard, it is important to note also that the specifications of these inter-provider APIs

Figure 3-3: Service Inter-Operation across 5G-VINNI Sites. Source:



3.3.1 5G Network Slice Federation Scenarios

This section presents two types of Network Slice Federation Scenarios. First is how an E2E 5G Network slice federation is created involving Access, Transport and Core by presenting a paradigm of another project, namely the 5GinFIRE H2020 project [36] and how it can be adapted for other research projects. The second presents how an E2E 5G Service Level Federation can be deployed with an E2E cross-domain orchestrator between facilities.

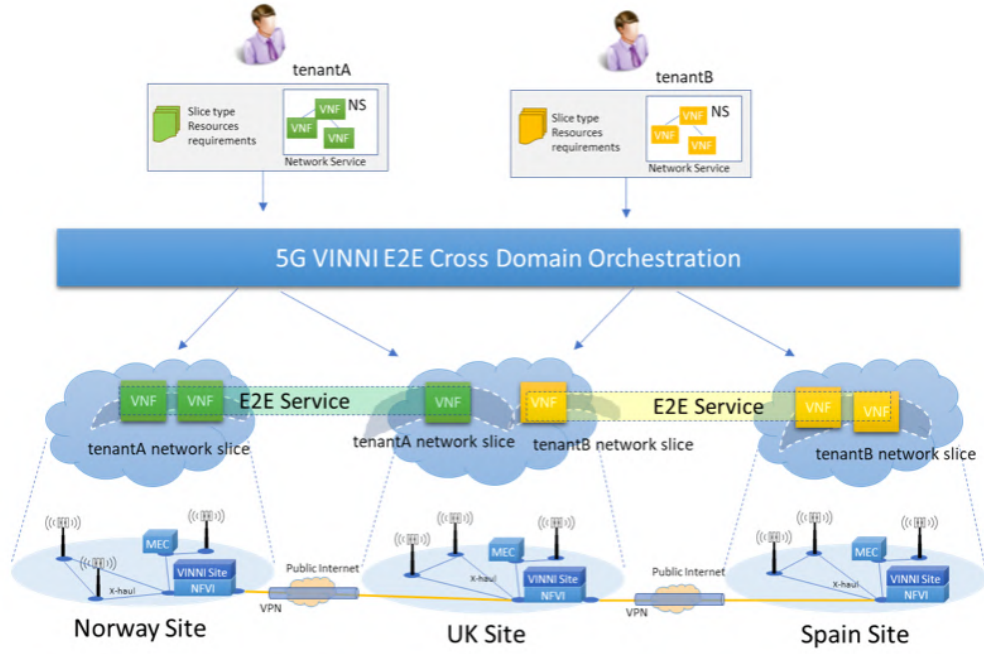
Network Slice Federation Across 5G-VINNI Facility Sites

An example scenario of E2E network slice federation across three different 5G-VINNI Facility Sites is illustrated in Figure 3-4 which involves three facilities (e.g., Norway Site, UK Site and Spain Site) providing a single E2E network slice.

In order to have such a scenario as per Figure 3-4, the facilities would need to

are still at an early stage and to be considered as a key activity and work in progress by the industry forums and specification defining organizations.

Figure 3-4: E2E Network Slice Federation across three different Facility Sites. Source:



have established agreements in the business (e.g. what type of resources are allowed and at what cost) and connectivity level (e.g. Layer 2 or 3). With respect to the latter, related work of such deployment has been done by 5GinFIRE [36] where a single MANO is responsible for orchestrating multiple interconnected and physically separated NFVIs. In 5G-VINNI, however, each facility has its own MANO stack.

In 5GinFIRE, partner testbeds have mutual agreements in regards to joining and accepting NFs and also allowing access of their VIM's APIs to a centralized MANO. This cross-site interconnectivity supports the exchange of control and data-plane information among sites, utilizing an overlay of a network architecture based on VPNs. This approach [37], given that sites are physically separated at different network locations, means that inter-site communications may traverse multiple untrusted network domains. Inter-site data-plane communications will require the creation of specific CIM networks at each site, with network connectivity towards the appropriate VPN endpoint of the site. These networks will then be used to attach any VNFs needing this type of communications, while guaranteeing all subsequent security principles.

Service Level Federation across 5G-VINNI Facility Sites

In these scenarios, two 5G-VINNI facility sites support different types of network slices in order to create an E2E service for 5G-VINNI customers. Although the federation can be seen at the Service Layer, the interconnection and interworking between the two slices would be required to support E2E service operation.

Figure 3-5: Multiple Tenants and Federated Network Slicing. Source:

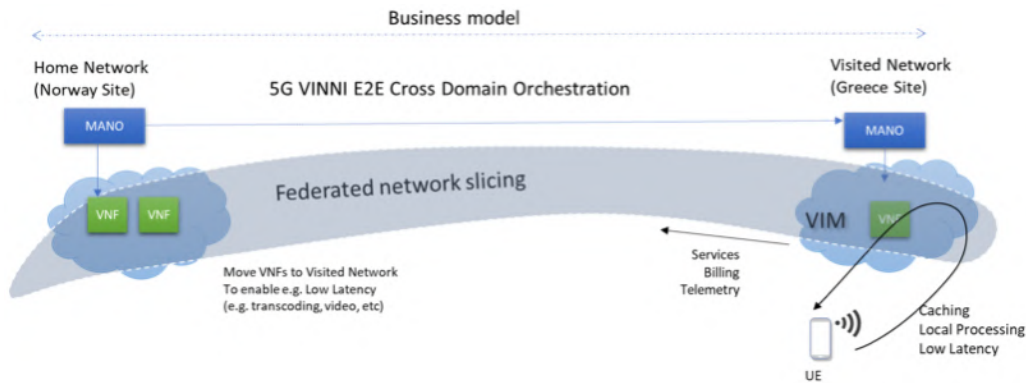


Figure 3-5 displays a generic view of how multiple tenants can use the 5G-VINNI E2E Facility to deploy an E2E service on top of a network slices types. This connectivity provides a unified service experience for the creation and roaming extension of network slices optimized for demanding scenarios. Today such scenarios are exclusively enabled for cloud services, but with the introduction of network slice requirements, services must move to the edge. In order to ensure the performance demands, edge processing is necessary to maintain latency within limits and make sure the 5G application behaves as usual, even in roaming scenarios.

3.3.2 Use Case Verticals

The 5G-VINNI facility site in Patras (Greece) will be an exemplary Open Source 5G-IoT facility. This means that most of the installed components will be offered as Open Source but there will be also dedicated components and services to support 5G-IoT scenarios. Numerous partners will deploy their technologies in the Patras/Greece facility, thus creating a unique 5G playground for KPI validation and support on future verticals.

Verticals Summary

Additionally, please note that these use cases could be combined and create different scenarios, by using some aspects of each distinct use cases. The service offerings referred in the Greece portfolio are detailed in Table 3.1 including information on their service exposure level and their ability to support the aforementioned use cases.

Table 3.1: Service Offerings in the Greece Portfolio

Service offerings	Service exposure	Supported use cases
eMBB network SaaS	Level 3-4	UC1 ¹ , UC2, UC4
uRLLC network SaaS	Level 3-4	UC1 ¹
mIoT network SaaS	Level 3-4	UC3 (IoT slicing)
Customised network slice	Level 3-4	UC1 ¹ , UC3 (NB-IoT slicing)

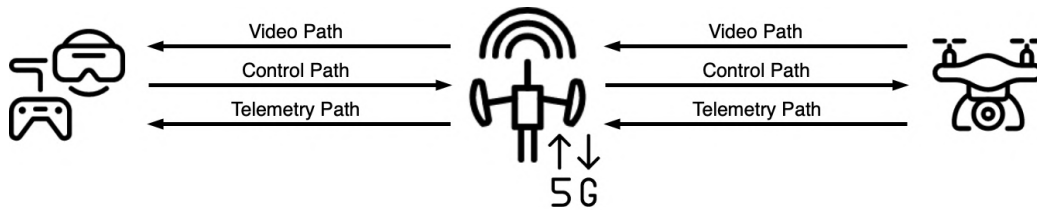
¹ Due to uRLLC and eMBB characteristics, the UC1 should be executed in a customised network slice. Until this service offering becomes available in the Greece portfolio, UC1 will be executed (with some limited performance and/or functionality) in eMBB and uRLLC network slices offered as a service to the CSC.

In Greece facility site, the following use cases are selected as the main candidates for 5G-VINNI validation purposes:

UC1 – First Person View Remote Control Vehicle (Automotive)

This use case consists of a remote-controlled vehicle with an on-board camera that provides real-time video streaming, allowing for remote operating. The camera provides real time frontal view of the vehicle, so the user can steer the vehicle by means of a controller. The video streaming aspect of the use case requires a significant amount of data received through the 5G network, while the remote-control aspect requires low latency and high reliability delivery of the movement commands. To satisfy these requirements, capabilities provided by eMBB and URLLC slices are critical.

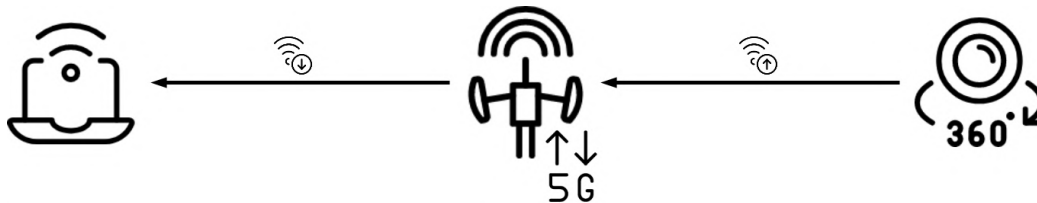
Figure 3-6: Greece UC1 – First Person View Remote Control Vehicle Use Sase



UC2 – 360-degree Video broadcast (Media and Entertainment)

This use case involves video streaming with a 360-degree camera. A possible scenario is during Patras carnival festivities, where participants can be equipped with 5G-enabled devices with 360o cameras in order to transmit the local festivities to remote locations while being able to move around downtown Patras. The main requirements of this use case are high throughput and high connection density in crowded scenarios.

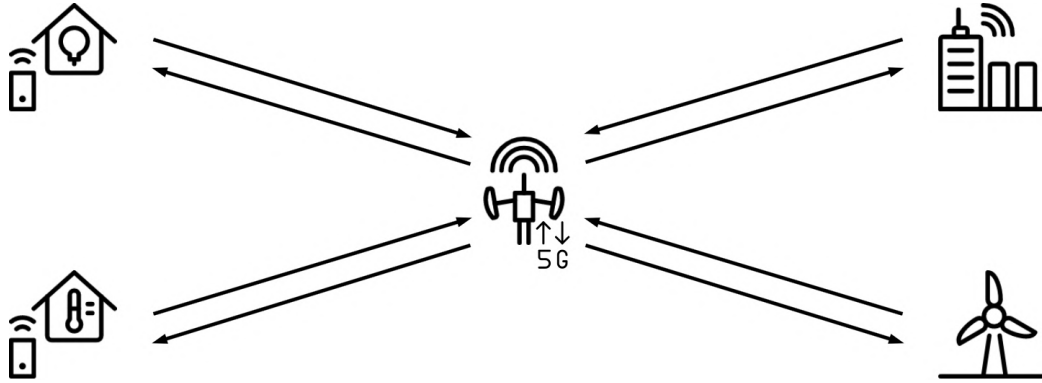
Figure 3-7: Greece UC2 – 360 ideo Broadcast Use Case



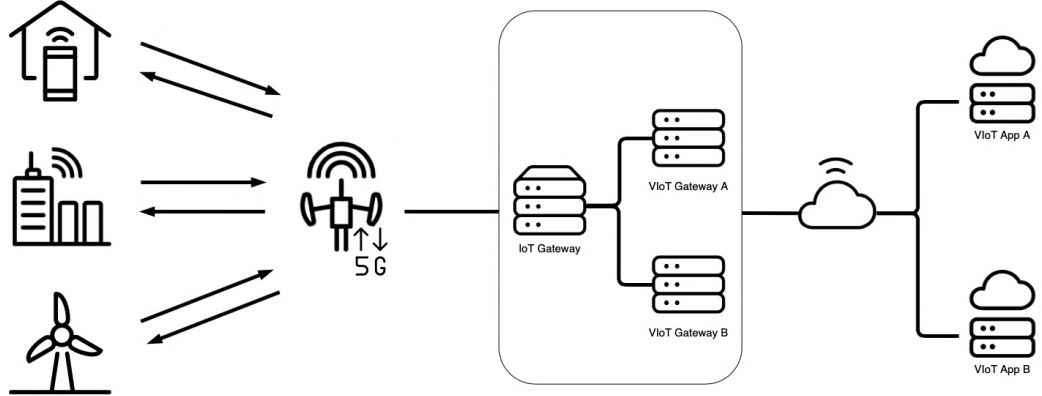
UC3 – NB-IoT Coverage and IoT Slicing study

The suggested use case for NB-IoT consists of two phases. In the first phase, a coverage study of NB-IoT devices will be carried out on the University of Patras (see Figure 3-8a). In this study, a large number of NB-IoT devices will be deployed and the network coverage will be examined, taking into consideration various factors including (but not limited to) distance from the antenna and penetration power. Once these factors have been analysed, then the second phase will apply the concept of IoT slicing, making use of distinct IoT applications provided by corresponding virtualized IoT Gateway instances deployed in separated NSIs (see Figure 3-8b).

Figure 3-8: Greece UC3 - Narrowband IoT Use Case



(a) Greece UC3 – NB-IoT Coverage

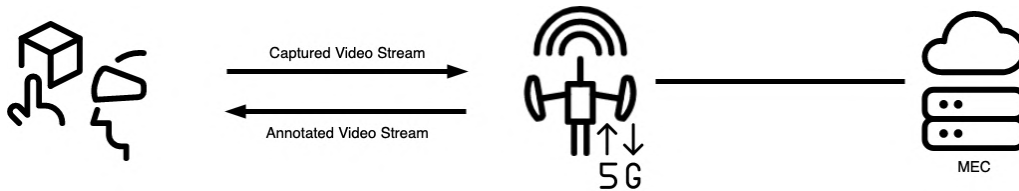


(b) Greece UC3 – Concept of IoT Slicing

UC4 – AR Use Case (Media and Entertainment)

This use case describes the delivery of AR annotations on top of a video stream captured by an end user of the service. The end user captures video, which is delivered to an AR application backend responsible for object recognition, tracking and annotation of the video with related information. The video annotation is performed taking into account the position of the identified objects in the video and delivered as an enhanced video stream to the user. This scenarios' main requirement is the very high throughput. To satisfy these requirements, the operator of this use case may rely on the communication capabilities provided by an eMBB slice. This use case is expected to be supported by DFKI, as a member of the Executive Board. The exact application context will be later defined, choosing from the wide DFKI portfolio. A candidate application targets the touristic domain, providing informational annotations on top of identified monuments. The use case targets the use of a MEC location within the Patras facility, as a means to lower latency.

Figure 3-9: Greece UC4 – AR Use Case



Chapter 4

Drone-based Verticals Use Case

4.1 Purpose

This chapter will serve as an in-depth look in the general architecture and implementation steps as outlined by the 5G-VINNI principles. Below the overall setup that will comprise the first use case of the Greece main facility is documented, aiming to showcase in a practical manner the capabilities, advancements and even limitations of this paradigm shift that network slicing aims to deliver. Following this high-level yet thorough description of the main focus of this thesis, Chapter 5 will delve deeper in the configuration parameters and the actual realization of the Unmanned Aerial System (UAS) platform.

4.2 Use Case and Key Performance Indicators

Use Case Definition

This use case briefly mentioned in Section 3.3.2 is comprised by two distinct parts that are required in order to successfully complete its task. On one end, a ground station unit will provide the user with the ability to control in real time a vehicle (in this case a quadcopter), through the cellular network. Compared to more traditional radio frequency control schemes, the user is not required to maintain line of sight of said vehicle, as the cellular-connected onboard computer is able to receive commands through the network. This leads us to the other part of the system which is of course the vehicle itself, able to transmit live video feeds back to the ground station in order to provide the user with appropriate feedback of its course, to which he can react accordingly. Legacy systems utilize alternative radio frequency bands to transmit said video feeds which are prone to interference, low quality and medium to high latency. These get replaced with modern technologies that alleviate all of the issues previously mentioned. By leveraging V2X connectivity, the user stands to benefit from the emerging 5G NR network, as increased coverage, low latency and quality of service are some of its main attributes. Such scenarios can be encountered in many cases with the most notable ones being search and rescue operations, remote takeover of autonomous vehicles or even human supervision of areas of interest.

Design-wise, the use case requires the implementation of the user and vehicle-side aspects while simultaneously satisfying network and open-source platform requirements. As an added bonus, various accessories that can augment the overall experience such as first person view (FPV) goggles and 360o cameras shall be supported in future iterations.

Key Performance Indicators (KPI) Definition

As previously mentioned, this use case aims to primarily showcase the benefits provided by network slicing technologies, as its implementation on current cellular network technologies would be impossible or to the least, impractical. For said reason, we need to establish some of the main key performance indicators that will need to be achieved in order to achieve full functionality of the platform. These are briefly listed below:

- **Throughput:** The minimum data rate required for the end-user to receive video feedback of adequate quality¹ and latency for the use case.
- **Latency:** This equates to the maximum end-to-end delay ² from when the user issues a command to when the vehicle is able to receive said information and respond accordingly.
- **Reliability:** The resilience of said communications system to all kind of errors introduced by interference and channel imperfections.
- **Service Instantiation Time:** The delay measured by the time that is required by the platform to initiate, configure and deploy said required service on command.
- **Coverage:** The area³ for which the vehicle will be able to fully operate within, while still adhering to the full specifications mentioned above. Area for which the vehicle will be able to move and stream video. The control path can potentially be done remotely as well.

¹For the use case High Definition (HD) quality with 24 frames per second (FPS) should be supported. This equals to a minimum throughput requirement of approximately 5 Mbps.

²No more than the latency experienced by legacy remote controlled systems.

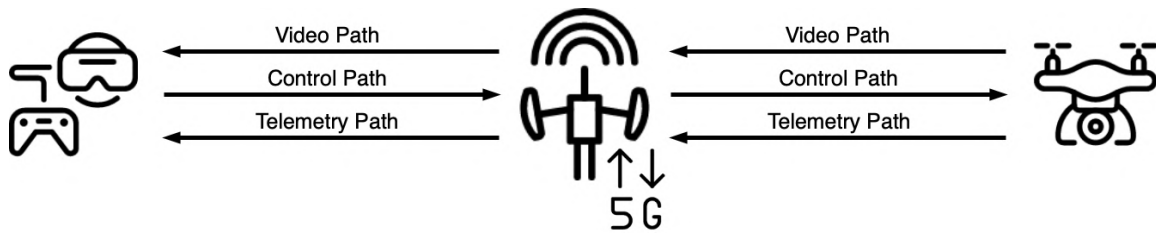
³No more than 90m² will suffice for demonstration purposes. This implies that a basic off the shelf device can be utilized as a base station without requiring additional amplification and a range of about 15m.

4.3 Communication paths

The main communication paths will be:

- Video path: Providing adequate bandwidth and latency for live video streaming from the vehicle to the user.
- Telemetry path: Providing information regarding the vehicle status.
- Telecommand path: Enabling remote vehicle operation from the user.

Figure 4-1: Use Case Scenario



In our use case we need to satisfy eMBB (to allow seamless video streaming) and uRLLC to enable an acceptable remote control of the vehicle.

Video Path

The deployed 5G network must be able to successfully handle various video streams from the devices, satisfying bandwidth requirements. The user must be able to download (Rx) while the vehicle side must be able to upload (Tx) at similar rates to carry out the use case demo successfully. It goes without saying that the xNodeB should support at minimum these values in both directions. The values in Table 4.1 are calculated by using 24 FPS which is considered acceptable for video streaming. Consequently, should the use case require true FPV or VR/AR, more FPS and/or higher quality, more bandwidth would be required.

Table 4.1: Bandwidth in Relation to Video Quality

Bandwidth (Mbps) ¹	Usage scenario
0.5	Minimum bandwidth required for live streaming
1.5	Minimum bandwidth required for a broadband connection
3.0	Standard Definition (SD) bandwidth
5.0	High Definition (HD) bandwidth
25	Ultra HD (4K UHD) bandwidth

¹ Due to uRLLC and eMBB characteristics, the UC1 should be executed in a customised network slice. Until this service offering becomes available in the Greece portfolio, UC1 will be executed (with some limited performance and/or functionality) in eMBB and uRLLC network slices offered as a service to the CSC.

Telecommand Path

Depending on the type of vehicle used, the 5G solution should exhibit the same or less latency of transmission and reception of commands than traditional radio control solutions. This would allow for a paradigm shift in either reduced latency, theoretically unlimited range or both, over legacy systems.

Telemetry Path

The telemetry path aims to showcase an optional implementation of a mIoT slice for a less-demanding path in terms of network resources. Telemetry data such as drone location, altitude and orientation would prove extremely vital for flight path logging, as well as any potential use case specific sensors that could be mounted to the drone platform at each time.

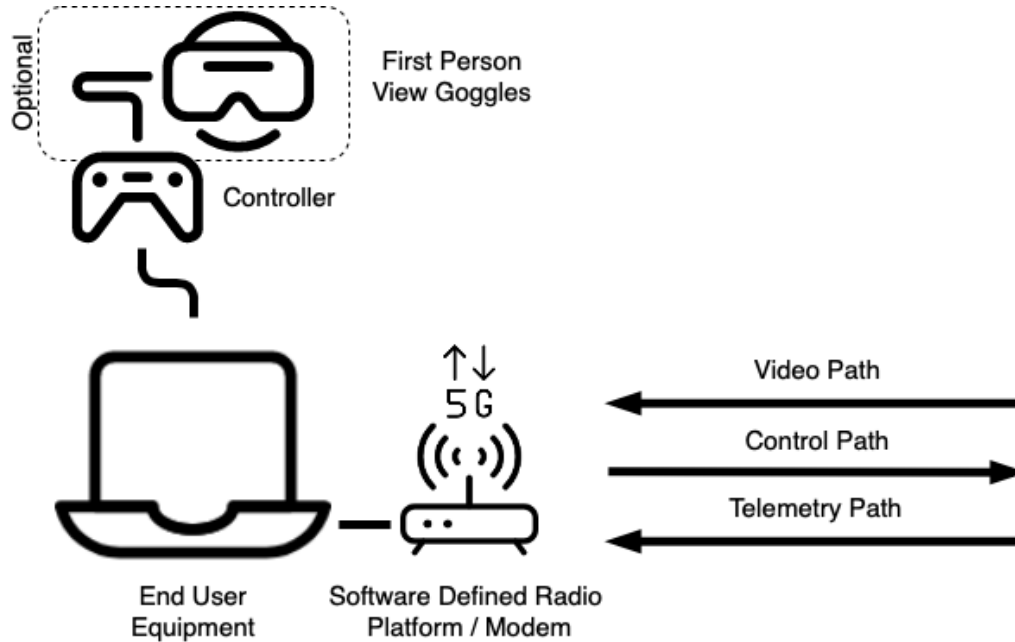
4.4 System Architecture Overview

A more thorough description of the use case is provided on the next sections. Please note that final implementation concepts or hardware can vary, but the overall system architecture will remain the same.

4.4.1 User Side

For the User Side, a configuration as shown in Figure 4-2 is proposed:

Figure 4-2: Detailed Approach for User Side of Use Case

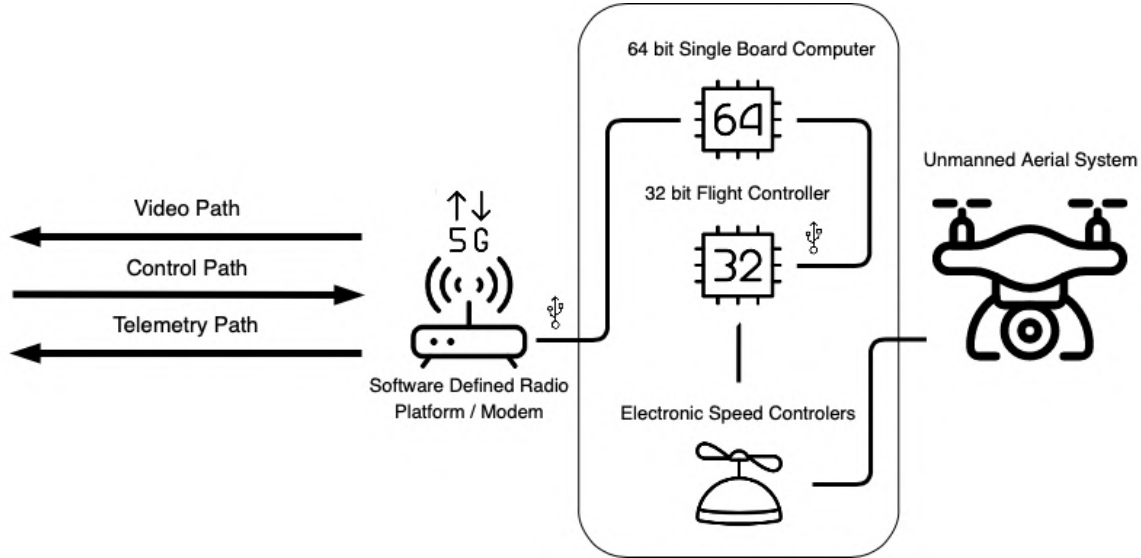


In our use case scenario, the goggles are replaced with a laptop for the sake of simplicity. This is in part for removing an extra layer of complexity from the development cycle and mainly for the reason that no 5G-compatible standalone headsets exist commercially. Most of the currently available products either use radio frequencies on the 2.4 or 5.8 GHz bands, suffering of course greatly from reduced range and reception quality. In future iterations, the First Person View goggles could be connected to a video out port on the end user equipment (in this case a laptop) to alleviate this problem. Due to the nature of analog control of vehicles, the laptop keyboard should be avoided as a means of steering at all costs, as it would prove too "rough" of an input device (either full throttle or none). For this reason, a wireless adapter will be used to relay a traditional two-stick radio controller's commands to the ground station software running on the laptop. Commercial joysticks aimed for video game use have demonstrated allot of jitter on their input so they are also disqualified for the sake of safety of both the drone and the pilot.

4.4.2 Vehicle Side

Regarding the vehicle side, the following setup is proposed:

Figure 4-3: Detailed Approach for Vehicle Side of Use Case



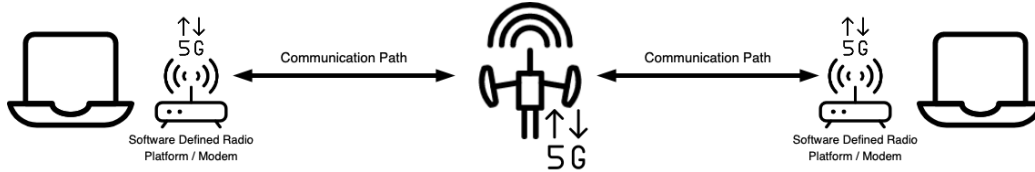
One of the most important parts of the use case is the vehicle side implementation. This is because all workload-intensive tasks such as video encoding and streaming while maintaining timely reaction to received commands are vital to its operation. For this reason, commercial off-the-shelf solutions that are geared towards recreational usage prove to be severely incapable of conforming to our platform needs. After extensive market research, only two solutions satisfied all the demands that our use case would require, those being either a custom-made solution from readily available parts or a drone research kit offered by Intel. Both of those proved to have their own distinct advantages, but the Ready to Fly (RtF) solution was preferred over the tedious task of tinkering with PID controllers and troubleshooting hardware bugs. In short, the Intel Aero platform provides a powerful, yet light enough Single Board Computer (SBC) for video encoding and streaming, as well as a combined takeoff weight of 1.9kgs. This would allow for more than enough headroom for future expansions, as additional cameras or sensors and easily be mounted and still maintain nominal 20 minutes flight time. Finally, the kit comes bundled with Intel's own flight controller, that connects over a serial port to the companion computer.

4.5 Intermediate Implementation Steps

Step 1

By only using laptops, the first step ensures the integrity of communication between the UE through the experimental cellular network.

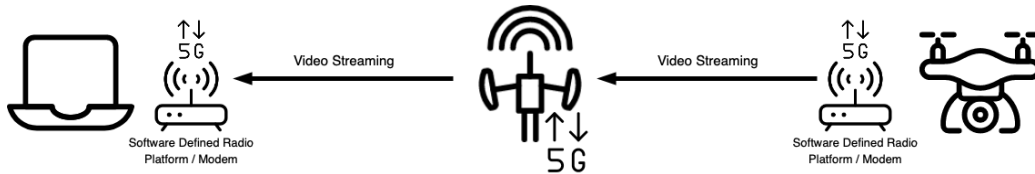
Figure 4-4: Implementation Step 1 - Verify Communication



Step 2

Using the same setup as before, replacing data with a complete video streaming pipeline between the UE and ground station in order to verify its quality.

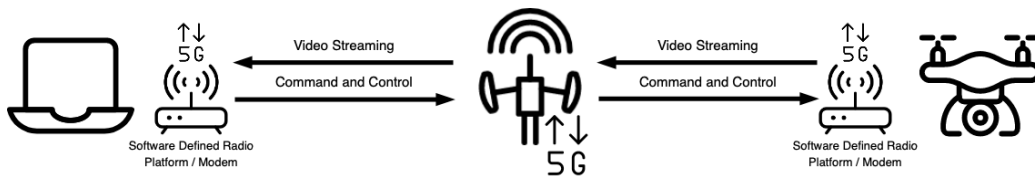
Figure 4-5: Implementation Step 2 - Verify Streaming



Step 3

Finally, replacing the laptop with a Single Board Computer (SBC), will allow to verification of the minimum system requirements. Simultaneously, the flight controller module can be also tested to verify cohesion of the control path.

Figure 4-6: Implementation Step 3 - SBC Requirements and Remote Control



4.6 Suggested Hardware

The combination of digital processing and analog RF has always made up communication systems. In today's modern systems signal processing has progressed to such an extent that a majority of baseband functionality is being implemented in software. The flexibility of the RF hardware to be re purposed and reconfigured has led to one radio font-end handling most RF systems. Normally the RF front-end is software controlled rather than software defined. This modern combination of flexible RF front-ends and signal processing in software has lead to the birth of software-defined radio. In general, a digital communications system consists of an interdependent sequence of operation responsible for taking some type of information and transmits it over-the-air to a receiver for processing and decoding into a reconstructed version of the original information signal.

There has been a growing amount of interest with respect to combining SDR technology with software-defined networks (SDNs), where the latter focuses on adapting the higher communication layers to the prevailing operational environment. This enables things like modification of the routing to be tied to heuristics provided by the physical layers.

Next-generation communications systems introduce new challenges that require solutions beyond what can be achieved through individual device optimization. Integrating more software control and cognitive abilities to the radio demands a more frequency- and bandwidth-flexible RF design. To achieve this goal static filters need to be removed and replaced with tunable filters. Similarly, the concept of a common platform would allow for shorter development times, reduced manufacturing costs, and provide greater interoperability between systems. The common platform demands that the RF system be capable of providing full performance for applications that traditionally had very different architectures. Finally, future platforms are pushing size and power demands to a new extreme.

4.6.1 Suggested Hardware for xNodeB

For this reason design considerations to take into account when devising digital communication systems based on an SDR platform include the integration of the physical and network layers via a real-time protocol implementation of an embedded processor. Most communication systems are divided into logically separated layers in order to more readily facilitate the design of the communication system. However, it is imperative that each layer is properly designed due to the strong interdependence between all the layers. Ensuring that a sufficiently wide bandwidth radio front-end exists with agility over multiple subchannels and a scalable number of antennas for spatial processing. Many networks employing digital communication systems possess a centralized architecture for controlling the operations of the overall network. Knowing the radio network architecture is important since it will dictate what sort of operations are essential for one digital transceiver to communicate with another. The ability to perform controlled experiments in different environments is important for the sake of demonstrating the reliability of a particular SDR implementation, providing a sanity check capability. Reconfigurability and fast prototyping through a software design flow for algorithm and protocol description.

From the perspective of a radio system and its implementation, much of the system design will focus on the physical layer, but it can't be forgotten how the link layer may affect the physical. Nevertheless, given that the baseband processing is all conducted in software, it is possible for the communications system to implement the higher layers of the stack in software as well. Many communication standards have adopted this scheme, where the entire communication system across all the layers are implemented in software, although depending on data rate requirements, this can require significant computational capabilities on the part of the system to achieve real-time operation. All software implementation enable functional updates without hardware replacements. The trade-offs of which function or layer is done in fixed hardware versus flexible software is an engineering decision based on volume, cost, power, complexity and many other factors.

Table 4.2: Software Defined Radio Platforms for xNodeB

	Range (MHz)	Bandwidth (MHz)	Rx/Tx	Duplex	Tx Power (dBm)	Sampling (MSps)	Interface	Chipset (FPGA)	Gates (K)	Size (mm)	Price (USD)
USRP N210	-	-	-	-	15	100	1x 1Gb	DSP3400	6	Half RU	2,048
USRP N300	10 - 4400	100	2/2	Full	18	153.6	2x SFP+	Zynq7035	275	Half RU	6,850
USRP N310	10 - 6000	100	4/4	Full	18	153.6	2x SFP+	Zynq7100	444	Half RU	10,537
USRP N320	3 - 6000	200	2/2	Half	18	153.6	1xQSFP+	Zynq7100	444	Half RU	12,000
USRP N321	3 - 6000	200	2/2	Half	18	153.6	1xQSFP+	Zynq7100	444	Half RU	13,500
USRP X300	-	-	-	-	10	200	2x10Gb	XC7K325T	326	277x218x39	4,640
USRP X310	-	-	-	-	10	200	2x10Gb	XC7K410T	407	277x218x39	5,714
USRP E310	70 - 6000	56	2/2	Full	10	61.44	1x1Gb	Zynq7020	85	133x68.2x26	3,221
USRP E313	70 - 6000	56	2/2	Full	10	61.44	1x1Gb	Zynq7020	85	186x280x106	4,334
LimeSDR	0.1 - 3800	61.44	2/2	Full	10	61.44	USB 3.0	Cyclone 4	40	100x60	315
LimeSDR QPCle	0.1 - 3800	61.44	4/4	Full	10	61.44	PCIe x4	Cyclone 4	80	136.85x68.9	799
LimeNET Micro	0.1 - 3800	61.44	2/2	Full	10	61.44	USB 2.0	Cyclone 4	40	125x65	345
LimeNET Mini	0.1 - 3800	61.44	2/2	Full	10	61.44	1x 1Gb	Cyclone 4	40	110x105	2,799
LimeNET Base Station	0.1 - 3800	61.44	4/4	Full	10	61.44	2x 1Gb	Cyclone 4	40	265x270x260	21,250
LimeNET CrowdCell	0.1 - 3800	61.44	2/2	Full	10	61.44	1x 1Gb	Cyclone 4	40	N/A	N/A
SBX 782761	400 - 4400	40	1/1	Full	-	-	-	-	-	-	573
SBX 783351	400 - 4400	120	1/1	Full	-	-	-	-	-	-	688
CBX 782760	1200-6000	40	1/1	Full	-	-	-	-	-	-	573
CBX 783353	1200-6000	120	1/1	Full	-	-	-	-	-	-	688
UBX 783774-01	10 - 6000	40	1/1	Full	-	-	-	-	-	-	1,056
UBX 783775-01	10 - 6000	160	1/1	Full	-	-	-	-	-	-	1,291
TwinRX	10 - 6000	80	2/0	Half	-	-	-	-	-	-	2,890

¹ <https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/pt/cyclone-iv-product-table.pdf>.

² Amplification.

³ Max stated, may vary in MIMO configuration .

⁴ additional daughterboards required.

⁵ small cell form-factor.

The USRP N310 is a networked software defined radio (SDR) that provides reliability and fault-tolerance for deployment in large-scale and distributed wireless systems. It simplifies control and management of a network of radios by introducing the unique capability to remotely perform tasks such as updating software, rebooting, factory resetting, self-testing, host PC/ARM debugging and monitoring system health. The USRP N310 is one of the highest channel density devices in the SDR market, offering four RX and four TX channels in a half-wide RU form factor. The RF front end uses two AD9371 transceivers, the latest RFIC technology from Analog Devices. Each channel provides up to 100 MHz of instantaneous bandwidth and covers an extended frequency range from 10 MHz to 6 GHz. The open-source USRP Hardware Driver (UHD) API and RF Network-on-Chip (RFNoC) FPGA development framework reduces software development effort and integrates with a variety of industry-standard tools. The baseband processor uses the Xilinx Zynq-7100 SoC to deliver a large user-programmable FPGA for real-time and low-latency processing and a dual-core ARM CPU for stand-alone operation. Users can deploy applications directly on to the preinstalled embedded Linux operating system or stream samples to a host computer using high-speed interfaces such as 10 Gigabit Ethernet over the two SFP+ ports.

Additionally, the USRP N310 has a flexible synchronization architecture with support for traditional SDR synchronization methods such as clock reference, PPS time reference, and GPSDO, which enable implementation of high channel count MIMO systems. In addition, the USRP N310 uniquely features Ethernet-based synchronization using the open-source White Rabbit timing protocol which enables precise baseband synchronization over large distances in GPS-denied environments. All Ettus Research USRP SDRs and NI USRP SDRs are supported by the USRP Hardware Driver, which is published by NI under open-source licenses. This driver facilitates application development on USRP hardware and offers cross-platform support for multiple industry-standard development environments and frameworks. As dual-licensed software, the USRP Hardware Driver is available under the open-source GNU General Public License version 3 (GPLv3).

Figure 4-7: View of the N310 SDR platform. Source: [6]

(a) Isometric view of the N310



(b) Front side connectors



(c) Back side connectors



4.7 Bill of Materials (BOM)

Table 4.4: (BOM)

Parts list ^a	Quantity ^b	Cost
<hr/> Autopilot + Processing <hr/>		
Pixhawk 4 + PM07	1	215.00
Intel Aero board	1	399.00
Intel Aero enclosure	1	69.00
Inter Aero vision kit	1	149.00
USB OTG adapter	1	16.00
<hr/> TOTAL <hr/>		
 Wireless Communications <hr/>		
Wireless Communications		
LimeSDR Mini	2	318.00
High Gain Omni Antenna	2	50.00
LimeNET Base Station	1	21 250.00
<hr/> TOTAL <hr/>		
 Hardware Total		
 GRAND TOTAL <hr/>		

¹ Due to uRLLC and eMBB characteristics, the UC1 should be executed in a customised network slice. Until this service offering becomes available in the Greece portfolio, UC1 will be executed (with some limited performance and/or functionality) in eMBB and uRLLC network slices offered as a service to the CSC.

Chapter 5

Unmanned Aerial System Architecture and Implementation

5.1 System Overview

5.1.1 Intel Aero Compute Board

The Intel® Aero Compute Board is a purpose-built, unmanned aerial vehicle (UAV) developer kit powered by a quad-core Intel® Atom™ processor and geared toward UAV developers, researchers, and enthusiasts. It features abundant storage capabilities, 802.11ac Wi-Fi, support for multiple cameras including the Intel® RealSense™ camera (R200), industry standard interfaces, and reconfigurable I/O to facilitate connecting to a broad variety of drone hardware subsystems. The compute board ships with open-source Yocto Project for embedded Linux systems and offers sample applications and APIs for flight and vision interfaces, reducing hurdles for developers of sophisticated drone applications.

This development platform is intended to be modified by developers according to their professional judgment. Intel has not established operating limitations for the kit or tested any configurations other than the base configuration that is shipped as configured from the factory.

By default, Intel Aero board and Intel Aero Ready To Fly kit are delivered with a Yocto Project build already flashed. Yocto project is an open source set of tools for embedded professionals. The UEFI BIOS is maintained by InsydeH20.¹ This Yocto build is preconfigured and highly customized for Intel Aero. In parallel to this Intel supported Yocto image, full native Ubuntu with Intel drivers is available as user installation.

For our use-case a huge effort was dedicated in order to fine-tune the appropriate configuration of hardware and software as to not jeopardize the reliability of the system. More specifically, it was found that under no circumstances any other version of Ubuntu Linux 16.04.3 should not be installed, for the sole reason that the system would cease to boot entirely. Additionally, any updates could cause instabilities, including important security patches. For this reason, the software stack is considered highly insecure in terms of being up-to-date with current Linux distributions and should not be operated on public-facing infrastructures and is intended only for demonstration purposes.

Facing a choice between the two Linux distributions available for the platform, namely a pre-configured Yocto Project supplied by Intel with the necessary packages for conventional applications or the desktop version of Ubuntu 16.04, versatility was a deciding factor. Although more lightweight and use-case specific, Yocto does not come with a package manager, with the only option being a re-compilation of the system image using so-called "recipes", something that would be extremely time consuming and with variable results. For this reason, we opted for the commercially available Ubuntu distribution, placing additional care as to not diverge greatly from the basic configuration, apart from necessary packages for our use-case.

¹Since September 04, 2019, Intel has discontinued the Intel Aero Platform for UAVs. Forum-based support (Intel Community) for Intel Aero products is available until June 15, 2019. There are no further software updates planned for the Intel Aero Platform for UAVs. All documentation resources for the Intel Aero products will be available to the Intel Aero developer community until February 15, 2022. Files licensed under open source licenses will continue to be available in binary and source code on GitHub.

Figure 5-2: Downloading and setting correct kernel version.

```

cchassis -- aero@upatras-aero: ~ -- ssh aero@192.168.2.9 -- 80x24
aero@upatras-aero:~$ sudo apt-get update
Get:1 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]
Get:2 https://download.01.org/aero/deb xenial InRelease [2711 B]
Get:3 https://download.01.org/aero/deb xenial/main amd64 Packages [4834 B]
Hit:4 http://gr.archive.ubuntu.com/ubuntu xenial InRelease
Get:5 http://gr.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
Fetched 226 kB in 5s (37,9 kB/s)
Reading package lists... Done
N: Skipping acquire of configured file 'main/binary-i386/Packages' as repository
'https://download.01.org/aero/deb xenial InRelease' doesn't support architectur
e 'i386'
aero@upatras-aero:~$ sudo apt-get -y install aero-system
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  aero-ardupilot aero-bios aero-camera-streaming-daemon aero-fpga aero-init
  aero-mavlink-router aero-optical-flow aero-pxa aero-spi-xfer aero-systemd
  aero-utils firmware-atomisp i2c-tools linux-headers-4.4.76-aero-1.2
  linux-image-4.4.76-aero-1.2 python-serial read-edid
Suggested packages:
  libi2c-dev python-serial linux-firmware-image-4.4.76-aero-1.2 python-wxgtk3.0
  python-wxgtk
The following NEW packages will be installed:

```

```

GNU nano 2.5.3 File: /etc/default/grub
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
# info -f grub -n 'Simple configuration'

GRUB_DEFAULT="Advanced options for Ubuntu/Ubuntu, with Linux 4.4.76-aero-1.2"
#GRUB_HIDDEN_TIMEOUT=0
GRUB_HIDDEN_TIMEOUT_QUIET=true
GRUB_TIMEOUT=10
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""

# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that obtains
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefe,0x87abccdef,0xefefefef"

# Uncomment to disable graphical terminal (grub-pc only)

```

5.1.2 Flight Controller

Included in the Ready to Fly (RTF) kit, an STM32F427V autopilot board is responsible for hosting the Real Time Operating System (RTOS), under which the Autopilot Software Suite operates. The flight controller also hosts a variety of sensors responsible for vital vehicle operations such as a magnetometer and a compass, along with a plethora of connectivity for additional use-case specific modules. This specific module comes pre-installed on the aforementioned companion computer, attached via a proprietary connector. This raises some questions regarding the serviceability of the platform, since replacement parts are no longer available, but an external aftermarket flight controller can always be connected via a USB connection.

Regarding the autopilot, the two most popular stacks were under consideration for this implementation, namely PX4 and Ardupilot, with the former being the ultimate choice. PX4 consists of two main layers: the flight stack which is an estimation and flight control system, responsible for the basic functionalities of any autonomous vehicle such as balancing and positional awareness and the middleware, which is a general robotics layer that can support any type of autonomous robot that consists primarily of device drivers for embedded sensors and communication with the external world (companion computer, Ground Control Stations, etc.).

5.1.3 Cameras

Placed in the front-facing side of the quadrotor assembly, the Intel® RealSense™ R200 module implements a long range, stereovision 3D imaging system that implements a variety of capabilities to be used accordingly. More specifically, it can provide color, depth, and infrared video streams. Depth video streams are like color video streams, except each pixel has a value representing the distance away from the camera instead of color information. It consists of an infrared laser projection system, two infrared and a full HD color imaging sensors. The depth video stream is generated with stereo vision technology assisted by the Infrared laser projector and the two infrared imaging sensors. Color data is provided by the full HD color imaging sensor.

Due to the complexity of such device, additional effort has to be taken in order to ensure proper operation during the use-case scenario.

Initially, a custom kernel must be booted in order for the various camera modules to be recognized appropriately. Following, the legacy branch of the RealSense R200 public repository must be configured on the Ubuntu 16.04.3 OS, where after successful initialization, three distinct video devices should be listed using the command `sudo v4l2-ctl --list-devices`. To check if the camera drivers are correctly installed, list the video devices with `ls /dev/video*`

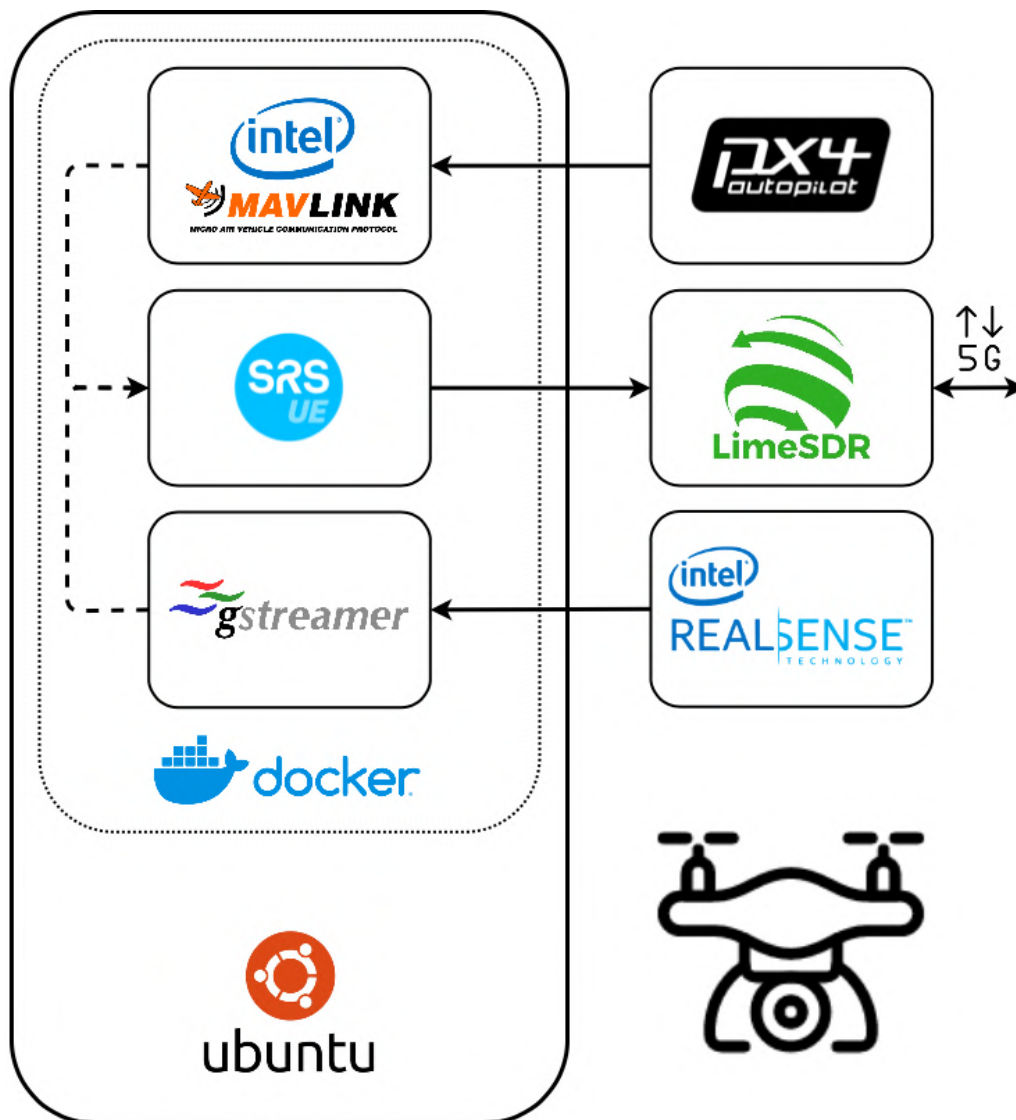
Third Party Cameras

It is also possible to connect another camera for further use case expansions. What is imperative though is ensuring compatibility with existing IO ports on the Intel Aero compute board (SPI, CAN and USB are available) and linux driver support (typically UVC, USB Video Class). The camera is recognized as a UVC device, as expected. As before, run `ls /dev/video*` to list the video devices. Use `sudo v4l2-ctl --list-devices` to get the details about the camera-number association.

5.1.4 Additional Software

Following proper setup of both the software and hardware aspects of the system, the next step is to ensure its key functionalities in order to operate within the use-case requirements. For this reason, two main software stacks need to be individually installed and configured, detailed below.

Figure 5-3: Vehicle-side overview of the software sub-systems.



MAVLink-Router

MAVLink is a packet implementation for communicating drone messages, a popular interface for flight controllers such as PX4, Ardupilot and more. This was mainly used for Ground Control Stations (GCS) such as QGroundControl and Mission Planner, mainly for the command and control link. MAVLink follows a modern hybrid publish-subscribe and point-to-point design pattern: Data streams are published as topics while configuration sub-protocols such as the mission protocol or parameter protocol are point-to-point with retransmission. MAVLink is a binary telemetry protocol designed for resource-constrained systems and bandwidth-constrained links. The low level implementation of this protocol allows packets to be sent over UDP, TCP and serial interfaces with the exact same message format. Messages can be received over a serial connection and be forwarded to one or many UDP/TCP/serial connections. Telemetry data streams are sent in a multicast design while protocol aspects that change the system configuration and require guaranteed delivery like the mission protocol or parameter protocol are point-to-point with retransmission. Most services use the client-server pattern, such that the GCS (client) initiates a request and the vehicle (server) responds with data. Many existing solutions that provide network routability of MAVLINK packets exist, but the aforementioned MAVProxy, written in Python is relatively slow-performing, thus unsuitable for real-time operation. Fortunately, alongside the commercial release of Intel Aero*, the company released a plethora of supporting applications, one of which is mavlink-router, written in C++, with its implementation seems more appropriate for low-latency scenarios.

Mavlink-router is an Intel open source project that routes MAVLink streams to specific endpoints, first implemented on the Intel Aero to handle communications with the companion computer. In addition to routing streams, mavlink-router has the ability to route MAVLink streams over a network. This can be done either manually, or through a configuration file.

Listing 5.1: Example of a typical configuration file.

```
[General - Mavlink-router serves on this TCP port]
```

```
TcpServerPort=5790
```

```
ReportStats=false
```

```
MavlinkDialect=auto
```

```
[TcpEndpoint Localhost]
```

```
Address = 127.0.0.1
```

```
Port = 25790
```

```
RetryTimeout=10
```

```
[UdpEndpoint Eavesdropping]
```

```
Mode = Eavesdropping
```

```
Address = 0.0.0.0
```

```
Port = 10000
```

```
[UdpEndpoint Endpoint]
```

```
Mode = Normal
```

```
Address = <ADDRESS>
```

```
Port = 11000
```

```
[TCPEndpoint LTE]
```

```
Address = XX.XX.XX.XX (replace with GCS ip)
```

```
Port = 5760
```

```
Log=$HOME/log/flight-stack
```

```
LogMode=while_armed
```


Mavlink-router's primary purpose as the name suggests, is to route MAVLink streams between endpoints. The normal usage of mavlink-router is to listen to a flight controller (usually on a serial or UDP port) and forward traffic to other endpoints. A UartEndpoint is defined to listen to a serial device, usually the flight controller. In this case, the flight controller is attached to the ttyS1 serial device of the OS. The mavlink stream can be routed to a UdpEndpoint or a TcpEndpoint of a specified address and port. In order to forward streams to Ground Control Stations the parameter `Mode = Normal` needs to be specified. A very important feature is that commands can be sent to the Flight Controller Unit from said endpoints, allowing for complete remote operation over IP. MAVLinks lastly, can be broadcasted over an entire network, by configuring the Bcast address of said network. This proves useful if a VPN (or a network slice) is used as a dedicated network for controlling the drone. `mavlink-router` also listens, by default, on port 5760 for TCP connections. Any connection there will also receive routed packets.

Lastly, mavlink-router provides logging functions in which enables logging on the companion computer without physically accessing the SD card from the FCU. Logging is simply enabled by adding the Log directory in the configuration file. You can also specify the LogMode in order to set the behavior of the logging. if log mode is set as `LogMode=whlie_armed` the log file is created only once the flight controller is armed and marks the file as read-only.

- I. Route MAVLink packets between endpoints The usual configuration is to have one "master" endpoint that is the flight stack (either on UART or UDP) and other components that can be on UDP or TCP or UART endpoints. This is not strictly required and other configurations are possible: mavlink-router mainly routes mavlink packets from one endpoint to the other endpoints without differentiating what they are.

- II. Multicast MAVLink packets to UDP endpoints

```
$ mavlink-routerd -e 192.168.7.1:14550 \
-e 127.0.0.1:14550 /dev/ttyS1:1500000
```

III. Route mavlinks packets from any interface (localhost) The mavlink-router can be used to route packets from localhost to an external interface. To route packets between SITL running on one computer (sending MAVLink traffic to localhost on UDP port 14550), and QGC running on another computer (e.g. at address 10.73.41.30) you could: `$mavlink-routerd -e 10.73.41.30:14550 127.0.0.1:14550`

IV. Test mavlink-router By using `examples/sender.py` and `examples/receiver.py` to simulate traffic of mavlink messages. First script sends mavlink ping messages to a target mavlink system-id, and second receives and responds to them.
`$ python examples/sender.py 127.0.0.1:3000 100 0` Will send mavlink pings to UDP port 3000. Those pings will have 100 as source system id and will have 0 as target system id (0 means broadcast). Receiver could be set as:
`$ python examples/receiver.py 127.0.0.1:4000 50` Where 50 is the receiver system id. Then, to route between those:

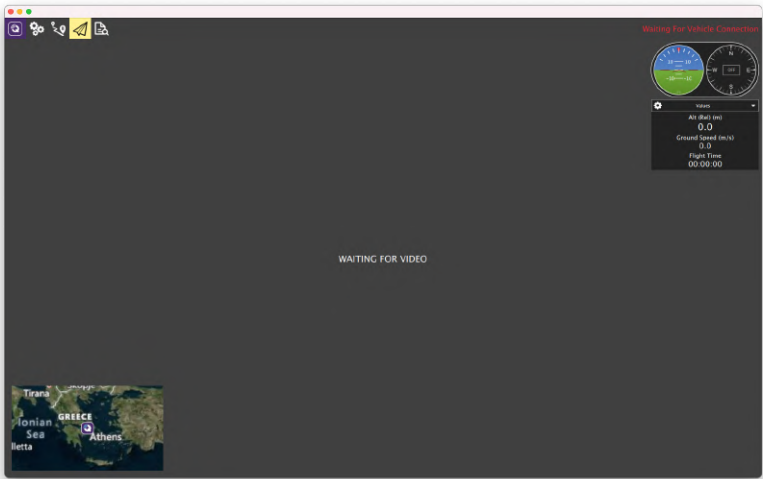
`$ mavlink-routerd -e 127.0.0.1:4000 0.0.0.0:3000` Note that it's possible to setup multiple senders and receivers to see mavlink-router in action.

V. Enable MAV_BROADCAST By enabling MAV_BROADCAST, heartbeats are periodically sent out on the local network. A remote computer can then connect by listening to the appropriate port (i.e. 14550 for QGroundControl).

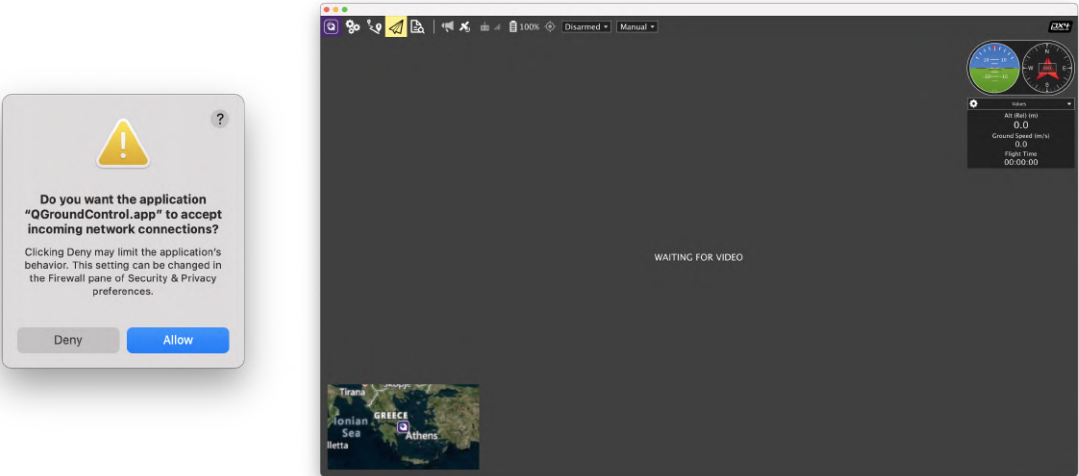
VI. Route packets from Localhost

After saving, restarting the router service is required with `sudo systemctl restart mavlink-router`. Launching QGroundControl on the user side should automatically receive telemetry feed from the drone.

Figure 5-4: Default Ground Control Station software state.



[Bottom row displays the request to allow traffic through the MAVLink port and the resulting information.]



Intel Camera Streaming Daemon

Camera streaming daemon gets installed with aero-system by default. It is responsible for detecting installed camera devices and proposing RTSP feeds. No further manual configuration is required.

Listing 5.2: Check that CSD is running

```
$systemctl status csd
csd.service - Camera Streaming Daemon
Loaded: loaded (/lib/systemd/system/csd.service; disabled; vendor
       preset: enabled)
Active: active (running) <output ommited>
```

Camera Streaming Daemon will stream the video feeds over the network with RTSP, a standard protocol. RTSP video can be received by QGroundControl and typical video players on most platforms.

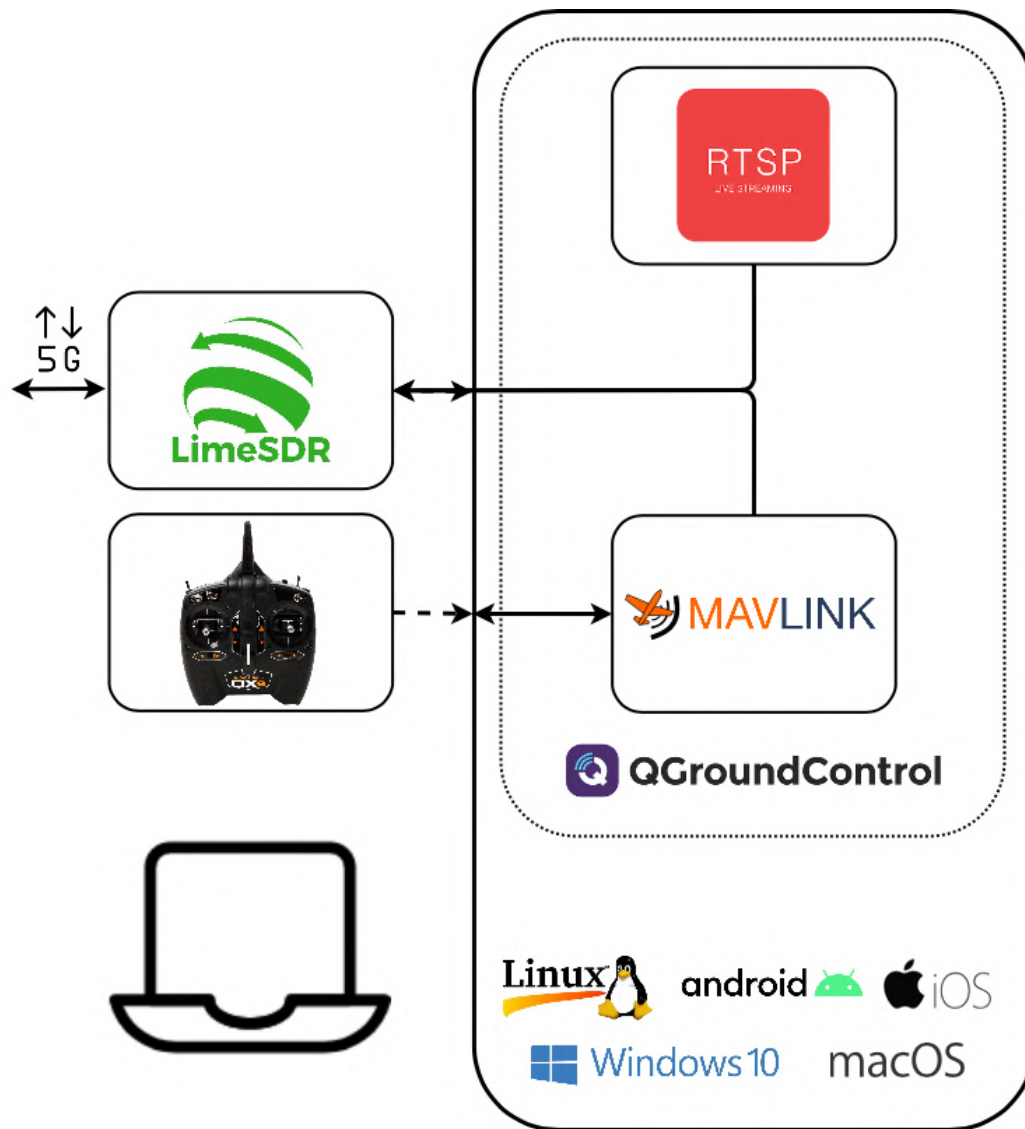
Intel Aero is proposing by default 5 RTSP video feeds:

```
RealSense R200, HD camera: rtsp://192.168.8.1:8554/video13
RealSense R200, depth sensor: rtsp://192.168.8.1:8554/rsdepth
RealSense R200, infrared first camera: rtsp://192.168.8.1:8554/rsir
RealSense R200, infrared second camera: rtsp://192.168.8.1:8554/rsir2
Bottom facing global shutter: rtsp://192.168.8.1:8554/bottom
```

Figure 5-5: RTSP video stream of the front-facing camera.



Figure 5-6: User-side overview of the software sub-systems.



QGroundControl

QGroundControl provides full flight control and vehicle setup for PX4 or ArduPilot powered vehicles. It provides easy and straightforward usage for beginners, while still delivering high end feature support for experienced users. The MAVLink "microservices" define higher-level protocols that MAVLink systems can adopt in order to better inter-operate and are used to exchange many types of data, including: parameters, missions, trajectories, images and other files.

Figure 5-7: Result of proper calibration of all sensors to arm motors.

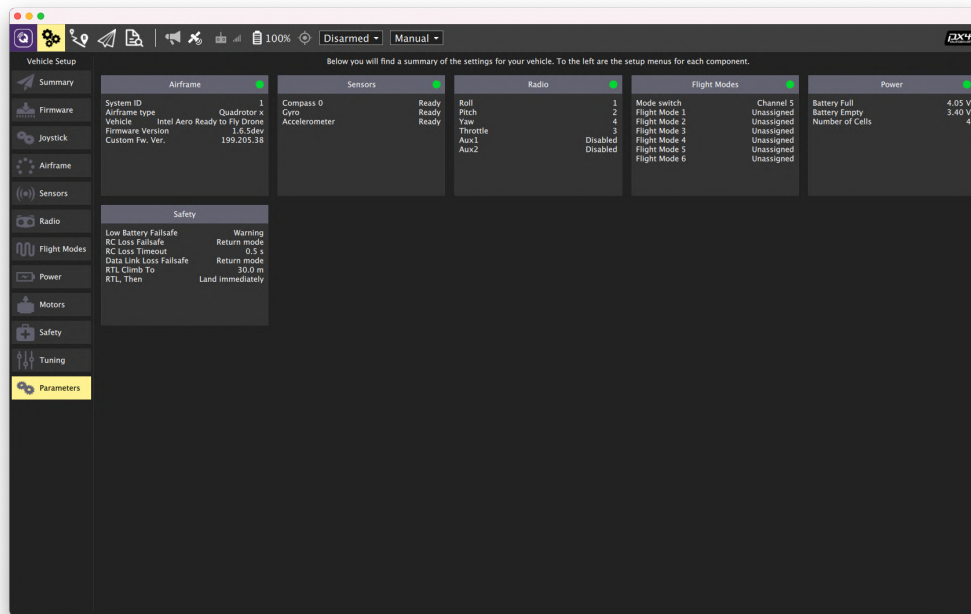


Figure 5-8: Verifying wireless controller adapter is recognized as input.

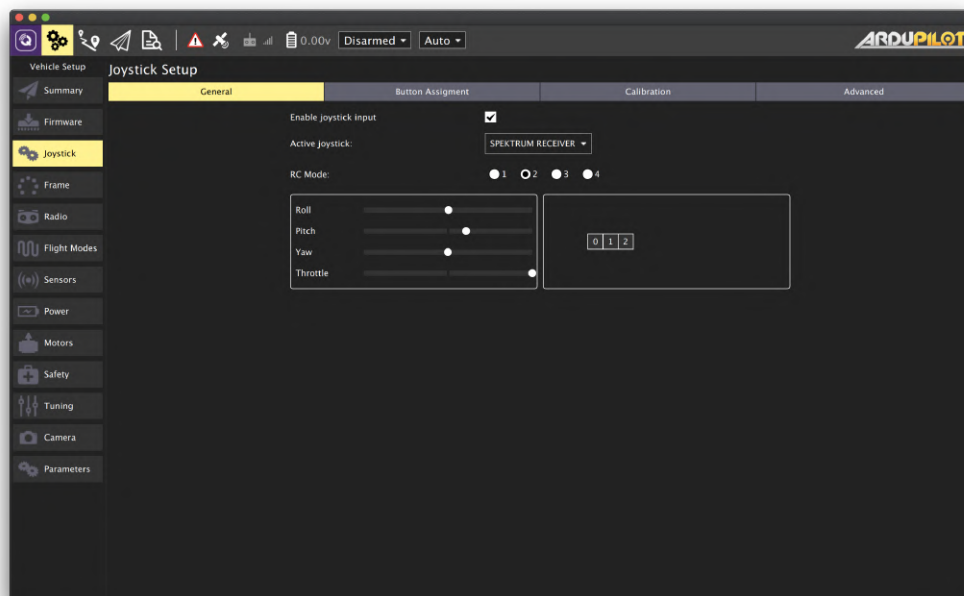
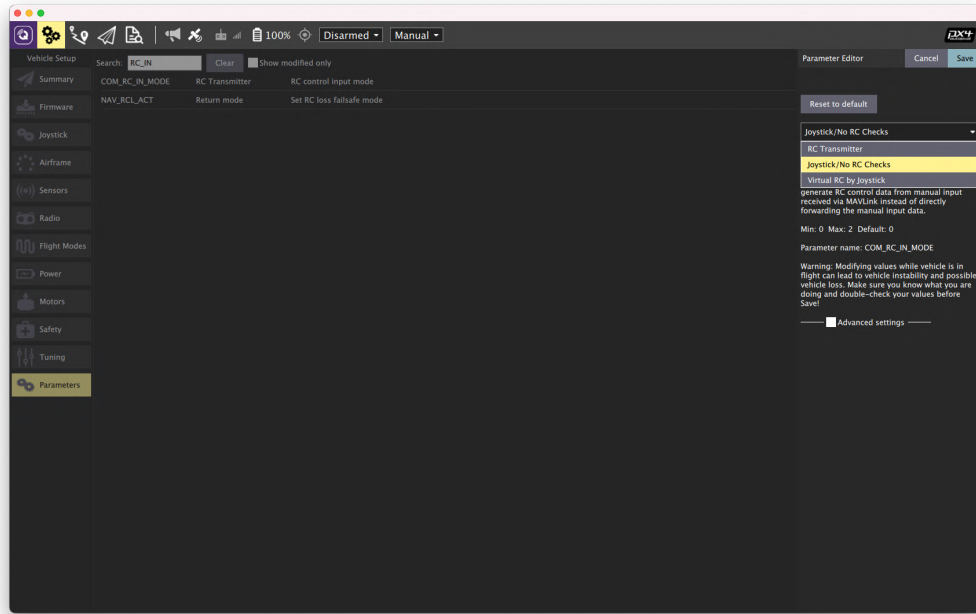


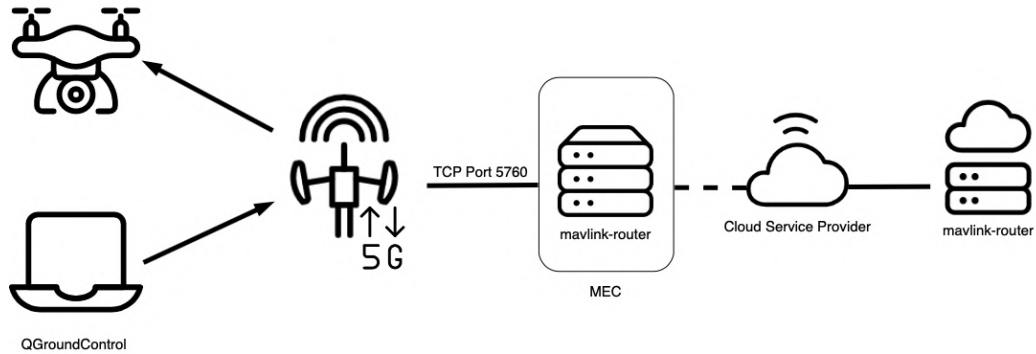
Figure 5-9: Setting parameter COM_RC_IN_MODE to accept Joystick input.



By default, QGC communicates directly with Aero when connected to the same subnet. However, when the drone is on a cellular connection, a direct connection is non-trivial as both the drone and PC can be behind multiple layers of NATs/firewall. In the options described below, a cloud server is used to help route packets between the drone and PC. Alternatively, the prospective MEC of the gNodeB would be a prime candidate for hosting said instances for greatly reduced latency, as seen on Figure 5-10.

On the Intel Aero Ready to Fly Drone (aka Aero RTF), mavlink-router runs locally to handle routing packets between the flight controller and different IP endpoints. But we can also deploy another instance of mavlink-router in the cloud to handle routing just IP traffic. Both the drone and QGC are then connected to this cloud based mavlink-router and communication is established. One benefit of using this method is flight logs are stored automatically in the cloud, so a copy of the logs can always be retrieved. This method also scales well with one-to-many use cases.

Figure 5-10: MAVLink Router implementation



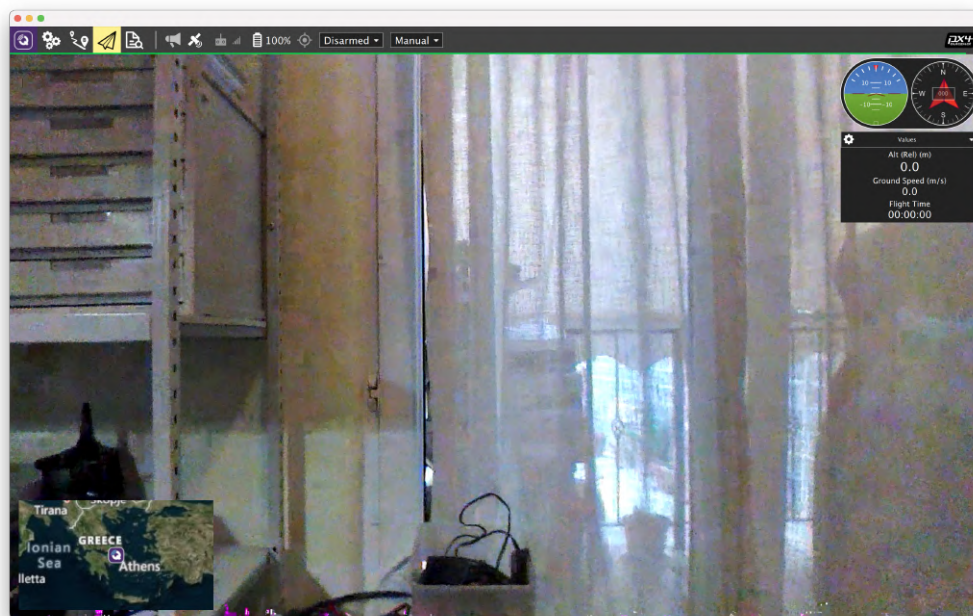
Listing 5.3: Check that MAVLink Router is running

#Check if the system is listening for MAVLink messages on port 5760:

```
$netstat -l|grep 5760|
```

```
tcp        0      0  *:5760                 :::*                    LISTEN
```

Figure 5-11: Complete view of the Ground Control Station software.



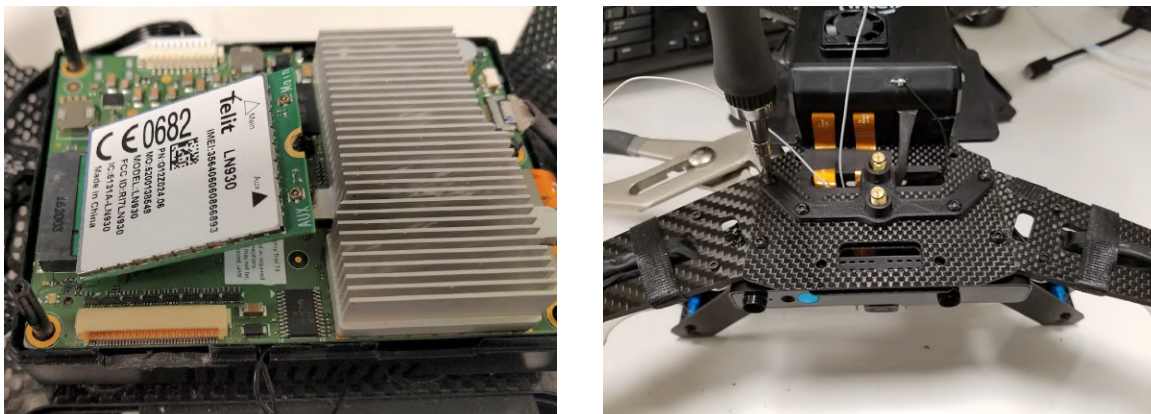
5.2 Networking

By default, Intel Aero is networking through WiFi, either as a client or host. This is especially useful during development periods and short-range tests. Throughout the duration of this work, WiFi connectivity was used for testing purposes, with cellular connectivity roll-out planned for later stages. Regardless, few things are expected to differ in terms of configuration and overall setup.

The Intel Aero Compute Board enables LTE modem devices to be installed into the M.2 interface by implementing a modem management software in the updated BIOS, especially convenient for our intended use-case, as any future M.2 form-factor modems can be swapped out, thus allowing easier connectivity upgrades and testing.

- I. The modem will be installed into the M.2 connector which is located on the top side of the Aero Compute Board, adjacent to the 80 pin I/O Connector.
- II. When installing the modem, two antennas are required for proper operation. The WiFi antennas could be re-purposed for use with the LTE modem, but since both functions are required, we will be adding antennas which will extend the range of the drone's cellular radio.
- III. Modem Manager should automatically detect the installed LTE modem and enumerate it as Modem 0 with `$mmcli -m 0`. An APN must be set appropriately to access the network.

Figure 5-12: Installing the modem and placing the additional antenna mounts.



5.3 Docker and Container Networking

There are many benefits for abstracting from a typical desktop OS installation in our specific use-case. A major one would be the compartmentalization of each of the processes that is to be run isolated from each other, as to ensure proper resource allocation, alleviating at the same time dependency problems previously mentioned. In this way, by updating only the docker runtime environment, each process can be install, updated and run independently. An added benefit of this setups would be of course its evolution to interact with the next-generation capabilities of 5G Core (or any next-gen network for that matter). By this, each initialized instance would be able to request a specific type of network slice according to its use, all the while providing added benefits of load balancing and fault tolerance that are eclipsed from regular OS or VM implementations.

Following, are the three main docker container images that will be supplied with the finalized version of this drone platform, each geared towards a specific purpose in the overall use-case scenario.

5.3.1 srsue

srsLTE is a free and open-source LTE software suite developed by SRS The srsUE, srsENB and srsEPC applications include example configuration files that should be modified to meet the system configuration. Running the command `sudo srsue` and by using the default configuration, creates a virtual network interface named "tun_srsue" on the container with an IP in the network 172.16.0.x. Assuming the UE has been assigned an IP, we may now theoretically exchange traffic over the cellular link.

By default, containers are attached to a Docker network with a default route. This means everyone has internet access through the virtualized Docker network. In our use-case, the default-route parameter should be modified to point towards the `srsue` container, which will handle the software stack that establishes the cellular connection either through a conventional modem or software-defined radio device. In

order to make containers access the internet through the `srsue` instead, we simply:

- Configure network address translation at the UE

```
docker exec virtual-srsepc iptables -t nat \  
-A POSTROUTING -s 172.16.0.0/24 -o eth0 -j MASQUERADE
```

This will masquerade all forwarded traffic from local containers (matched by source IP address) leaving the UE's `eth0` (Docker) interface.

- Configure microservices containers to route traffic via the `srsue` by default

```
docker exec virtual-srsue ip route replace default via 172.16.0.1
```

Now we should have network access through the onboard UE software stack.

Additionally, the library currently supports the Ettus Universal Hardware Driver (UHD) and the bladeRF driver. Thus, any hardware supported by UHD or bladeRF can be used, assuming allocation to the appropriate container. There is no sampling rate conversion, therefore the hardware should support 30.72 MHz clock in order to work correctly with LTE sampling frequencies and decode signals from live LTE base stations.

5.3.2 gstreamer

Due to poor streaming quality and performance by using conventional network feed streaming applications such as VLC, another solution would need to be implemented in order to further reduce the latency of the video transmission. For this reason, gstreamer, an open-source multimedia framework, is able to provide the granularity that's required in our use-case. Examples of that include -but are not limited- to enabling H.264 encoding, target framerate, resolution and aspect ratio, as well as accepting multiple inputs at the same time. By configuring the appropriate default gateway setting on the networking parameters of this container and exposing its relevant network ports, we should be able to override using the system's bridge network and route the rtsp stream through `srsue`.

5.3.3 mavlink-router

As previously mentioned, mavlink-router's sole purpose is to provide an encapsulation method for MAVLink packets to traverse the IP network. Additionally, its versatility in terms of addressing also provides a strategic advantage in terms of flexibility, as it can route streams to and from multiple devices independently at the same time. For this reason, by creating a basic container image and pointing the ttyS1 serial device to it, each running image will be able to handle separately traffic between the flight controller and ground station end points. Same as before, we can route the connectivity of the MAVLink messages through the software stack of **srsue**.

Chapter 6

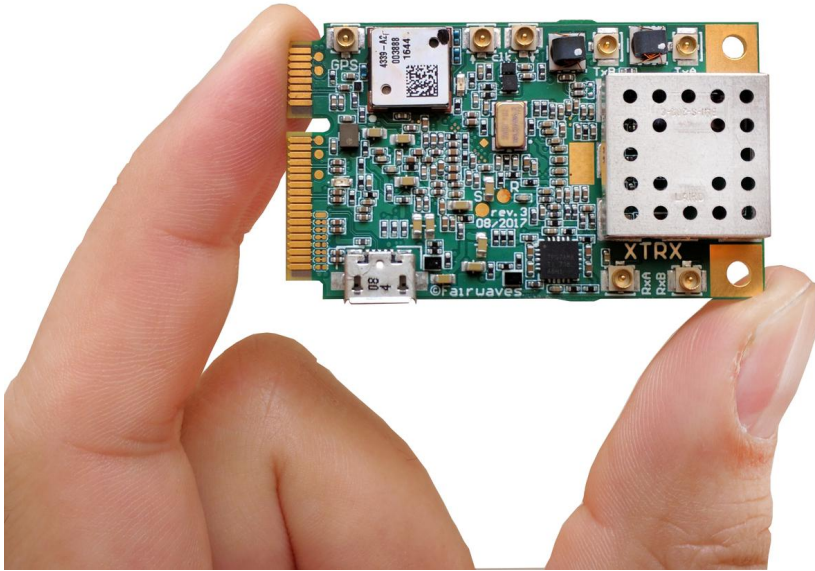
Future Work and Development

It should be apparent at this point that the drone platform described in detail in the previous chapters will be able to carry out the use-case demonstration it was intended for. During the development of this thesis, the use case scenario was validated to be fully operable through a wireless IP network, with acceptable delays in the video and control data transmission. What is also true is that there is plenty of room for improvements left in order to augment its capabilities and bring it closer to the next generation of interconnected aerial platforms of the future. Three key points have been selected to demonstrate some more or less obvious points for development - in the form of hardware or software - that would enhance the drone's capabilities and further improve its overall performance.

6.1 Miniaturized Software Defined Radio platforms

One area that the microelectronics industry has significantly influenced over the past half century is the digital communication systems sector, where microprocessor systems have been increasingly employed in the implementation of digital transceivers, yielding more versatile, powerful, and portable communication system platforms.

Figure 6-1: XTRX Pro Board for Scale. Source: [8]



Devices like the XTRX SDR shown in Figure 6-1 integrate the full RF, analog and digital signal chain onto a single CMOS device. Devices like the LMS7002M focus on very low power consumption small form factor SDR applications such as UAV data links and handheld communication systems. This opens up a new design potential for a different suite of applications where using wideband waveforms or occupying noncontiguous spectrum can now be implemented in a much smaller form factor. The XTRX offers a 2x2 MIMO configuration with a stable clock accurate enough for cellular standards, an onboard GPS disciplined oscillator (GPSDO) and a SIM card reader. When inserted into an appropriate Mini PCIe slot, it appears as a USB SIM card reader, able to interface with a wide variety of single board computers such as the one used in our usecase.

6.2 Leveraging new streaming protocols

Traditional streaming protocols, such as RTSP and RTMP, support low-latency streaming. But they aren't supported on all endpoints and work best for streaming to a small audience from a dedicated media server. These protocols achieve this by transmitting the data using a firehose approach rather than requiring local caching.

WebRTC is a combination of standards, protocols, and JavaScript APIs created by the World Wide Web Consortium (W3C) that enables real-time communications, allowing browser applications to make calls without any plugins and designed primarily with latency in mind. Different bodies such as the Internet Engineering Task Force, created to standardize the used protocols with browser APIs have been working on this implementation. Users connecting via Chrome, Firefox, or Safari can communicate directly through their browsers — enabling sub-500 millisecond latency to more than 85% of all installed browsers globally for real-time communications on the internet.

Table 6.1: Comparison Between Video Streaming Protocols in ms. Source: [9]

	Establishment	Reception	Client to Client	Client to Web
RTSP	2304 ms	2161 ms	37.807 ms	-
WebRTC	1835 ms	1709ms	8.072 ms	5.112 ms
Skype	-	-	-	11.646 ms

In a relevant work [9], two video streaming platforms implementing the most commonly used video streaming protocols, RTSP and WebRTC, have been implemented in order to conclude which offers the best results in the scope of web real-time video streaming applications. The analysis includes different parameters, namely the connection establishment and stream reception time, critical factors in applications such as our usecase. The results of the experiments discussed suggest that the use of the WebRTC protocol offers a vastly improved performance over the others, as can be seen in Table 5-11.

6.3 Going cloud native

Cloud-native today is almost synonymous with containers orchestrated by Kubernetes. It wasn't always thus. It's perfectly possible to build something that is cloud-native in all respects other than running in containers – i.e. dynamically orchestratable stateless microservices running in VMs – and production deployments have demonstrated many of the benefits, particularly with regard to simple, rapid scaling of the system and the automation of lifecycle management operations such as software upgrades.

Drawing a parallelism to the ETSI NFV architecture, it essentially covers the Virtual Infrastructure Manager (VIM) and VNF Manager (VNFM) roles. In its VIM role, Kubernetes schedules container-based workloads and manages their network connectivity, including a kind of Load Balancer as a Service, making it easy to deploy scale-out microservices. In its VNFM role, Kubernetes can monitor the health of each container instance and restart any failed instance. It can also monitor the relative load on a set of container instances that are providing some specific micro-service and can scale out (or scale in) by spinning up new containers or spinning down existing ones. In this sense, Kubernetes acts as a Generic VNFM. For some types of workloads, especially stateful ones such as databases or state stores, Kubernetes native functionality for lifecycle management is not sufficient. In NFV terms, it comprises a standardized way of building Specific VNFMs. But the beauty of Kubernetes is that it can provide a common environment across all types of cloud infrastructures. In this way, Kubernetes allows us to achieve a unified strategy for NFV that is entirely compelling.

Acronyms

3GPP 3rd Generation Partnership Project. 7, 18, 25, 28, 38, 42, 49

5G-VINNI 5G Verticals Innovation Infrastructure. 11, 16, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 59, 60, 61, 62, 63

5G-PPP 5G Infrastructure Public Private Partnership. 7, 15, 16, 17, 22, 30

AF Application Function. 19, 20, 28

AN Access Network. 39

API Application Programming Interface. 27, 59, 61

APN Access Point Name. 29

AR Augmented Reality. 11, 66

ASQ Assured Service Quality. 59

BBU Baseband Unit. 22

CIM Cooperative Information Manager. 54, 61

CN Core Network. 28, 50

CP Control Plane. 23, 54, 55, 59

CPRI Common Public Radio Interface. 22

CSC Communication Service Customer. 21

CSP Communication Service Provider. 30, 32, 34, 36

CU Central Unit. 28

DC Data Center. 37, 38, 39, 42

DU Distributed Unit. 28

E2E end-to-end. 11, 16, 19, 21, 25, 28, 29, 32, 38, 46, 54, 55, 56, 60, 61, 62

eMBB Enhanced Mobile Broadband. 7, 13, 18, 30, 31, 63, 64, 66

eNB eNodeB. 28

EPC Evolved Packet Core. 51

ETSI European Telecommunications Standards Institute. 7, 22, 23, 54

FWA Fixed Wireless Access. 48

gNB gNodeB. 28, 52, 53

IaaS Infrastructure as a Service. 35

ICT Information and Communication Technology. 15, 16

IoT Internet of Things. 34, 52, 63, 65

ISG Industry Specification Group. 22, 23

ITU International Telecommunication Union. 30

KPI Key Performance Indicator. 16, 17, 21, 30, 34, 47, 48, 63

LTE Long Term Evolution. 22, 28, 48, 50, 51

MANO Management And Network Orchestration. 24, 38, 43, 48, 53, 54, 55, 56, 59,

MEC Multi-access Edge Computing. 19, 66

mIoT Massive Internet of Things. 7, 34

MNOs Mobile Network Operators. 20, 52

MP Management Plane. 21

MPLS Multiprotocol Label Switching. 43, 54, 58, 59

NB-IoT Narrowband IoT. 48

NF Network Function. 19, 21, 22, 23, 24, 29, 39, 53, 54

NFV Network Functions Virtualization. 22, 23, 24, 38, 48, 53, 54, 55, 56, 59

NFVI Network Function Virtualization Infrastructure. 24, 53, 54, 61

NMS Network Management System. 24

NR New Radio. 28, 34, 50, 51

NSA Non-Standalone. 49, 50, 51

NSaaS Network Slice as a Service. 29, 36

NSI Network Slice Instance. 21, 27, 28, 37, 38, 39, 42, 43, 65

NSSAI Network Slice Selection Assistance Information. 29

NSSI Network Sub-Slice Instance. 21, 28, 38, 39, 42

OS Operating System. 40, 41, 42

OSM Open Source MANO. 43

PaaS Platform as a Service. 35

PLMNs Public Land Mobile Networks. 19

PNF Physical Network Function. 23, 24, 37, 39, 48, 54

PoP Points of Presence. 24

PRB Physical Resource Block. 37, 44

PtP Point-to-Point. 52

QAM Quadrature Amplitude Modulation. 51

QoS Quality of Service. 29, 51, 58, 59

RAN Radio Access Network. 8, 19, 22, 27, 28, 29, 37, 38, 42, 43, 44, 48, 49, 50, 51, 52, 54

RF Radio Frequency. 37

RU Radio Unit. 22, 28

SA Standalone. 29, 49, 51

SBA Service Based Architecture. 19

SBC Single Board Computer. 11, 74

SCS Sub-Carrier Spacing. 50

SDN Software Defined Networking. 59

SD-WAN Software-defined Wide Area Network. 53, 58, 59

SDOs Standards Developing Organisations. 17, 25

SLA Service Level Agreement. 25, 26, 35, 36

TN Transport Network. 37, 38, 42, 43

TR Technical Report. 18, 22, 50

TS Technical Specification. 18, 20

UE User Equipment. 19, 28, 37

UP User Plane. 30

URLLC Ultra-Reliable Low Latency Communication. 7, 13, 32, 33, 64

V2X Vehicle-to-Everything. 48

VDU Virtual Deployment Unit. 13, 39, 40, 41, 42

VIM Virtualised Infrastructure Manager. 43, 61

VLAN Virtual Local Area Network. 43

VM Virtual Machine. 11, 41, 42

VNF Virtual Network Function. 21, 22, 23, 24, 39, 41, 42, 43, 47, 48, 53, 54, 61

VNFM Virtual Network Function Manager. 24

VPN Virtual Private Network. 29, 58, 59, 61

Bibliography

- [1] J. Ordonez-Lucena, P. Ameigeiras, D. López, J. Ramos-Muñoz, J. Lorca, and J. Folgueira, “Network slicing for 5g with sdn/nfv: Concepts, architectures, and challenges,” *IEEE Communications Magazine*, vol. 55, pp. 80–87, 2017.
- [2] Telecom Infra Project, “Telecom Infra Project, Creating Ecosystems for End-to-End Network Slicing,” Telecom Infra Project, White Paper, 2018. [Online]. Available: https://cdn.brandfolder.io/D8DI15S7/as/qcraft-9d31vs-exu2kb/E2ENS-Phase1_-_White_Paper_-_Telecom_Infra_Project.pdf
- [3] A. Gallego, A. Elmokashfi, A. J. Gonzalez, C. Guerrero, C. Politis, C. Tranoris, C. Kalogiros, D. Warren, D. R. Lopez, D. Kritharidis, F. Michelinakis, G. D. Stamoulis, J. Rodrigues, J. Carapinha, J. Ordonez-Lucena, K. Mahmood, K. Liolis, I. Telecom, M. Pauls, M. Xie, P. Grønsund, P. Muschamp, P. Gupta, R. Trivisonno, T. Briedigkeit, V. Theodorou, W. Y. Poe, and Y. Wang, “D1.2 Design of network slicing and supporting systems,” p. 65, Mar. 2019. [Online]. Available: <https://bscw.5g-ppp.eu/pub/bscw.cgi/229119>
- [4] [Online]. Available: <https://www.5g-vinni.eu/facility-site/>
- [5] 5G PPP Architecture Working Group, “5G PPP 5G Architecture,” 5G PPP Architecture Working Group, White Paper, Jan. 2018. [Online]. Available: <https://5g-ppp.eu/wp-content/uploads/2018/01/5G-PPP-5G-Architecture-White-Paper-Jan-2018-v2.0.pdf>
- [6] [Online]. Available: <https://www.ettus.com/all-products/usrp-n310/>
- [7] Intel Corporation, “Intel Aero Compute Board Hardware Features and Usage,” Intel Corporation, Technical Guide, 2017. [Online]. Available: <https://www.intel.com/content/dam/support/us/en/documents/drones/development-drones/intel-aero-compute-board-guide.pdf>
- [8] [Online]. Available: <https://xtrx.io>
- [9] J. M.-G. P. C.-G. Iván Santos-González, Alexandra Rivero-García, “Implementation and analysis of real-time streaming protocols,” *Sensors* 2017, vol. 41, no. 7, pp. 73+, Apr. 2017.

- [10] A. Bedeschi, "EC Launches 5G PPP Phase 3 With Three New Projects," European Conference on Networks and Communications, Ljubljana, Jun. 2018. [Online]. Available: <https://5g-ppp.eu/ec-launches-5g-ppp-phase-3-with-three-new-projects/>
- [11] C. Willcock, "Assessing the 5G research and development investment Leverage Factor," *The European 5G Annual Journal 2020*, p. 101, May 2020.
- [12] E. Commission, "Horizon 2020 - Work Programme 2018-2020: Information and Communication Technologies," European Commission, p. 64, Jun. 2018. [Online]. Available: https://ec.europa.eu/research/participants/data/ref/h2020/wp/2018-2020/main/h2020-wp1820-leit-ict_en.pdf
- [13] [Online]. Available: <https://5g-ppp.eu/5g-vinni/>
- [14] A. Neal, "Study on new services and markets technology enablers," 3rd Generation Partnership Project, VODAFONE Group Plc, Technical report, Sep. 2016.
- [15] D. Chandramouli and T. Sun, "System architecture for the 5G System (5GS)," 3rd Generation Partnership Project, Nokia Germany and China Mobile US Research Center, Technical specification, Mar. 2015.
- [16] P. Hedman and T. Sun, "Procedures for the 5G System (5GS)," 3rd Generation Partnership Project, Ericsson LM and China Mobile US Research Center, Technical specification, Dec. 2016.
- [17] M. Schramm, "Policy and charging control framework for the 5G System (5GS); Stage 2," 3rd Generation Partnership Project, HUAWEI TECHNOLOGIES Co. Ltd., Technical specification, Sep. 2017.
- [18] L. Zou, "Management and orchestration; Concepts, use cases and requirements," 3rd Generation Partnership Project, HUAWEI TECHNOLOGIES Co. Ltd., Technical specification, Jun. 2017.
- [19] J. Groenendijk, "Management and orchestration; Architecture framework," 3rd Generation Partnership Project, Ericsson LM, Technical Specification, Jan. 2018.
- [20] X. Sun, "Management and orchestration; 5G end to end Key Performance Indicators (KPI)," 3rd Generation Partnership Project, China Mobile Research Inst., Technical specification, Jan. 2018.
- [21] A. Umesh, "Study on new radio access technology: Radio access architecture and interfaces," 3rd Generation Partnership Project, NTT DOCOMO INC., Technical report, Mar. 2016.
- [22] 5G PPP Architecture Working Group, "View on 5G Architecture, Revision 2.0," 5G PPP Architecture Working Group, Technical Report, Dec. 2017. [Online]. Available: <https://5g-ppp.eu/wp-content/uploads/2018/01/5G-PPP-5G-Architecture-White-Paper-Jan-2018-v2.0.pdf>

- [23] O. Adamuz-Hinojosa, J. Ordonez-Lucena, P. Ameigeiras, J. J. Ramos-Munoz, D. Lopez, and J. Folgueira, "Automated network service scaling in nfv: Concepts, mechanisms and scaling workflow," *IEEE Communications Magazine*, vol. 56, no. 7, pp. 162–169, 2018.
- [24] European Telecommunications Standards Institute, "Network Functions Virtualisation (NFV); Management and Orchestration," European Telecommunications Standards Institute, Group Specification, Dec. 2014. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_NFV-MAN001v010101p.pdf
- [25] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing & softwarization: A survey," *IEEE Communications Surveys and Tutorials*, vol. 20, no. 3, pp. 2429–2453, 2018.
- [26] ITU-R, "Framework and overall objectives of the future development of IMT for 2020 and beyond," Sep. 2015. [Online]. Available: https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.2083-0-201509-I!!PDF-E.pdf
- [27] A. Li, "Service requirements for the 5G system," 3rd Generation Partnership Project, VODAFONE Group Plc, Technical specification, Oct. 2016.
- [28] A. Sultan, "Release description; Release 15," European Telecommunications Standards Institute, 3rd Generation Partnership Project, Technical Report, Mar. 2018.
- [29] O. Sallent, J. Perez-Romero, R. Ferrus, and R. Agusti, "On radio access network slicing from a radio resource management perspective," *IEEE Wireless Communications*, vol. 24, no. 5, pp. 166–174, 2017.
- [30] K. Takeda, "Study on New Radio (NR) access technology," 3rd Generation Partnership Project, NTT DOCOMO INC., Technical report, Jul. 2018.
- [31] E. Haleplidis, K. Pentikousis, S. Denazis, J. H. Salim, D. Meyer, and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology," RFC 7426, Jan. 2015. [Online]. Available: <https://rfc-editor.org/rfc/rfc7426.txt>
- [32] D. Zhang and D. Ionescu, "Qos performance analysis in deployment of diffserv-aware mpls traffic engineering," in *Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007)*, vol. 3, 2007, pp. 963–967.
- [33] O. Michel and E. Keller, "Sdn in wide-area networks: A survey," in *2017 Fourth International Conference on Software Defined Systems (SDS)*, 2017, pp. 37–42.
- [34] E. K. Ali, M. Manel, and Y. Habib, "An efficient mpls-based source routing scheme in software-defined wide area networks (sd-wan)," in *2017*

IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA), 2017, pp. 1205–1211.

- [35] B. P. Gero, M. Dramitinos, C. J. Bernardos, A. Ramos, L. M. Contreras, R. Szabo, R. Guerzoni, B. Sonkoly, A. Galis, D. Morris, B. Martini, J. Melian, H. Lonsethagen, D. Daino, G. Biczok, M. D. Girolamo, F. Tusa, G. Darzanos, I. Papafili, A. Sgambelluri, O. Dugeon, G. D. Stamoulis, M. A. Carnero, R. Canto, F. Valera, A. Garcia, H. Woesner, J. Kim, and F. Ubaldi, “Deliverable 2.1 5GEx Initial System Requirements and Architecture,” p. 210, Dec. 2016. [Online]. Available: https://bscw.5g-ppp.eu/pub/bscw.cgi/d139873/5GEx_D2.1v1.1_public_version.pdf
- [36] [Online]. Available: <https://5ginfire.eu/>
- [37] D. R. López, J. R. Martínez, B. N. Dorado, I. V. Fernández, A. P. Silva, D. Gomes, E. Sousa, and F. de Oliveira Silva, “D4.2 - Intermediate Report on the MANO Platform,” p. 43, Jan. 2018. [Online]. Available: https://bscw.5g-ppp.eu/pub/bscw.cgi/d139873/5GEx_D2.1v1.1_public_version.pdf