UNISINOS

Programa de Pós-Graduação em

Computação Aplicada

Mestrado/Doutorado Acadêmico

Felipe Hauschild Grings

NASP - NETWORK SLICE AS A SERVICE PLATFORM FOR NEW-GENERATION NETWORKS BEYOND 5G

São Leopoldo, 2024

G867n Grings, Felipe Hauschild.

NASP: Network Slice as a Service Platform for new-generation networks beyond 5G / Felipe Hauschild Grings. – São Leopoldo, 2024.

88 f. : il. ; 30 cm.

Dissertação (mestrado) – Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Computação Aplicada, São Leopoldo, RS, 2024.

“Advisor: Prof. Dr. Cristiano Bonato Both; Co-advisor: Prof. Dr. Lucio Rene Prade”, Escola Politécnica.

1.Sistemas de comunicação móvel. 2.Sistemas de comunicação sem fio. 3.Internet das coisas. 4.Redes locais sem fio. 5.Protocolo de aplicação sem fio (Protocolo de rede de computador). I.Both, Cristiano Bonato. II.Prade, Lucio Rene. III.Título.

CDU 004.75.057.5
004.7

Catalogação na publicação:
Bibliotecária Carla Maria Goulart de Moraes – CRB 10/1252

Felipe Hauschild Grings

NASP - NETWORK SLICE AS A SERVICE PLATFORM FOR NEW-GENERATION NETWORKS BEYOND 5G

A thesis presented as a partial requirement to obtain the Master's degree from the Postgraduate Program in Applied Computation of the University of Vale do Rio dos Sinos — UNISINOS

Advisor:
Prof. Dr. Cristiano Bonato Both

Co-advisor:
Prof. Dr. Lucio Rene Prade

São Leopoldo
2024

To my father, the example of strength, integrity, and professionalism, who has been my guiding light and strongest supporter. His encouragement was instrumental in my decision to embrace this challenge, offering unwavering support at every turn. To my mother, a beacon of resilience and independence, who defied all odds and showed me the power of a woman undeterred by societal expectations. This work is dedicated to my parents, whose hard work and sacrifices paved the way for me to dream beyond limits and achieve what once seemed impossible.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to all those who have supported and contributed to the completion of this thesis.

First and foremost, I would like to thank my advisor, Dr. [Advisor's Name], for their continuous support, patience, and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis.

I would also like to extend my thanks to the members of my thesis committee, [Committee Member 1], [Committee Member 2], and [Committee Member 3], for their insightful comments and encouragement.

My sincere thanks also go to [Name of any specific individuals or organizations] for providing me with all the necessary facilities and resources needed for my research.

I am also grateful to my fellow lab mates and colleagues for the stimulating discussions, for the sleepless nights we were working together before deadlines, and for all the fun we have had in the last [number] years.

A special thanks to my family. Words cannot express how grateful I am to my [family members, e.g., parents, spouse, children] for all of the sacrifices that you've made on my behalf. Your prayers for me were what sustained me thus far.

Finally, I would like to thank [Your University/Institution] and the [Department/Faculty Name] for providing the environment and support for my research.

Thank you all.

ABSTRACT

The 5th generation of mobile telecommunications (5G) is rapidly adopted worldwide, accelerating the demands for highly flexible private networks. In this context, 5G has mobile network slicing as one of its main features, where the 3rd Generation Partnership Project (3GPP) defines three main use cases: massive Internet of Things (mIoT), enhanced Mobile BroadBand (eMBB), and Ultra Reliable Low Latency Communications (URLLC), along with their management functions. Moreover, the European Telecommunications Standards Institute (ETSI) defines standards for Zero-touch network & Service Management (ZSM) without human intervention. However, the technical documents of these institutes fail to define End-to-End (E2E) management and integration among different domains and subnet instances. This work presents a network slice as a service platform (NASP) agnostically to 3GPP and non-3GPP networks. A NASP architecture is based on the main components, namely: (i) onboard requests for new slices at the business level, fulfilling the translation for definitions of physical instances, distributions, and interfaces among domains; (ii) hierarchy orchestrator working among management functions; and (iii) communication interfaces with network controllers. These configurations are based on the technical documents of entities such as 3GPP, ETSI, and O-RAN, following the study of overlapping designs and gaps among the different views. The NASP prototype was developed based on the proposed architecture, bringing implementations and solutions for an agnostic platform and provider of an end-to-end Network Slice as a Service. The tests were analyzed using two use cases (3GPP and Non-3GPP) with four different scenarios, i.e., mIoT, URLLC, 3GPP Shared, and Non-3GPP. The results pointed out the platform's adaptability in serving different requests received by the Communication Service Management Function. Moreover, the evaluation showed the time to create a Network Slice Instance, where 68% is dedicated to the Core configuration. The tests also presented a 93% reduction in data session establishment time comparing the URLLC and Shared scenarios. Finally, the study presents the cost variation for operating the platform with the orchestration of 5 and 10 slices, presenting a variation of 112% between Edge and Central.

Keywords: NSaaS. VNFs. SMO. virtualization. SDN.

RESUMO

A 5ª geração de telecomunicações móveis (5G) é rapidamente adotada em todo o mundo, acelerando as demandas por redes privadas altamente flexíveis. Nesse contexto, o 5G tem como uma de suas principais características o fatiamento da rede móvel, onde o 3rd Generation Partnership Project (3GPP) define três principais casos de uso: Internet das Coisas massiva (mIoT), Banda Larga Móvel Aprimorada (eMBB) e Ultra Reliable Low Latency Communications (URLLC), juntamente com suas funções de gerenciamento. Além disso, o Instituto Europeu de Padrões de Telecomunicações (ETSI) define padrões para rede Zero touch & Service Management (ZSM) sem intervenção humana. No entanto, os documentos técnicos desses institutos falham em definir o gerenciamento End-to-End (E2E) e a integração entre diferentes domínios e instâncias de sub-redes. Este trabalho apresenta uma plataforma para prover fatias de rede como serviço (NASP) agnóstica para redes 3GPP e não-3GPP. Assim, é proposta uma arquitetura NASP com a definição dos principais componentes, sendo eles: (i) onboard de requisições de novas fatias no nível de negócio, cumprindo a tradução para definições de instâncias físicas, distribuições e interfaces entre domínios; (ii) orquestrador hierárquico atuando entre funções gerenciais; e (iii) interfaces de comunicação com controladores de rede. Essas configurações são baseadas em documentos técnicos de entidades como 3GPP, ETSI e O-RAN, seguindo o estudo de projetos sobrepostos e lacunas entre as diferentes visões. O protótipo do NASP foi desenvolvido analisando a arquitetura prospota, trazendo implementações e soluções para uma plataforma agnostica e provedora de Network Slice as a Service end-to-end. Os testes foram analizados utilizando dois casos de uso (3GPP e Non-3GPP) com quatro cenários diferentes, sendo eles: mIoT, URLLC, 3GPP Shared, e Non-3GPP. Os resultados mostram a adaptabilidade da plataforma em servir diferentes requisições NST, mostra também detalhes do tempo de criação de uma NSI, onde 68% é dedicado a configuração do Core. Os testes também apresentam uma redução de 93% no tempo de PDU Session estabilishment comparando os cenarios URLLC e Shared. Por fim, o estudo apresenta a variação de custos para operação da plataforma com a orquestração de 5 e 10 slices, apresentando uma variação de 112% entre Edge e Central.

Palavras-chave: NSaaS. VNFs. SMO. Virtualização. SDN.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

3GPP	*3rd Generation Partnership Project*
AF	*Application Function*
AMF	*Access and Mobility management Function*
AMPS	*Advanced Mobile Phone System*
ANDSP	*Access Network Discovery e Selection Policy*
ANN	*Artificial Neural Network*
API	*Application Programming Interface*
AUSF	*Authentication Server Function*
AS	*Access Stratum*
B5G	*Beyond fifth-generation*
BBU	*Base Band Unit*
BH	*Backhaul*
BSS	*Business Support System*
BW	*Bandwidth*
CI	*Continous Integration*
CD	*Continous Delivery*
CH	*Crosshaul*
CN	*Core Network*
CNN	*Convolutional Neural Networks*
CPU	*Central Process Unit*
CR	*Compute Resources*
CRI	*Container Runtime Interface*
CU	*Control Unit*
CUPS	*Control and User Plane Separation*
DRL	*Deep Reinforcement Learning*
DN	*Data Network*
DNN	*Deep Neural Network*
DRB	*Data Radio Bearer*
DU	*Data Unit*
DSG	*Disaggregation RAN Unit*
E2E	*End-to-End*
eMBB	*Enhanced Mobile Broadband*
EMS	*Element Management System*

EPC	*Evolved Packet Core*
ETSI	*European Telecommunications Standards Institute*
FFNN	*Feedforward Neural Networks*
FH	*Fronthaul*
gNB	*Next Generation NodeB*
GPRS	*General Packet Radio Service*
GSM	*Global System for Mobile*
GST	*Generic Network Slice Template*
GTP	*GPRS Tunneling Protocol*
HTTP	*Hypertext Transfer Protocol*
IaC	*Infrastructure as a Code*
IEEE	*Institute of Electrical and Electronic Engineers*
IP	*Internet Protocol*
IPSec	*Internet Protocol Security*
LTE	*Long Term Evolution*
MAC	*Media Access Control*
MH	*Midhaul*
MIMO	*Multiple Input and Multiple Output*
mIoT	*Massive Internet of Things*
MME	*Mobility Management Entity*
mMTC	*Massive Machine Type Communication*
ML	*Machine Learning*
MNO	*Mobile Network Operator*
MMTel	*Multimedia Telephony service*
N3IWD	*Non-3GPP Internet Working Function*
NF	*Network Functions*
NEST	*Network Slice Template*
NFV	*Network Functions Virtualization*
NFVO	*Network Functions Virtualization Orchestrator*
NG-RAN	*Next Generation Radio Access Network*
NGAP	*Next-Generation Application Protocol*
NO	*Network Operator*
NS	*Network Slice*
NSI	*Network Slice Instance*

NSSI	*Network Sub Slice Instance*
NSMF	*Network Slice Management Function*
NSSMF	*Network Sub Slice Management Function*
NSO	*Network Service Orchestration*
NWDAF	*Network Data Analytics Function*
ONOS	*Open Network Operating System*
OS	*Operating System*
OSS	*Operations Support System*
UE	*User Equipment*
PDCP	*Packet Data Control Protocol*
PDU	*Protocol Data Unit*
PEE	*Power, Energy and Environmental*
PFD	*Packet Flow Descriptions*
PGW	*Packet Gateway*
PHR	*Power Headroom Report*
PHY	*Physical Layer*
PNF	*Physical Network Functions*
QoE	*Quality of Experience*
QoS	*Quality of Service*
RAN	*Radio Access Network*
RFSP	*Radio Frequency Selection Priority*
REST	*Representational State Transfer*
RL	*Reinforced Learning*
RLC	*Radio Link Control*
RIC	*RAN Intelligent Controller*
RO	*Resource Orchestration*
RRC	*Radio Resource Control*
RRH	*Remote Radio Head*
RU	*Radio Unit*
SDAP	*Service Data Adaptation Protocol*
SDN	*Software Defined Network*
SDO	*Software Defined Orchestrations*
SGi	*Service Gateway internal*
SLA	*Service Level Agreement*

SLS	*Service Level Specification*
SMF	*Session Management Function*
SRQ	*Specific Research Question*
URLLC	*Ultra-Reliable Low-Latency Communication*
V2X	*Vehicle to everything*
VIM	*Virtual Infrastructure Manager*
VLAN	*Virtual Local Area Network*
VNF	*Virtual Network Function*
VNFM	*VNF Manager*
vNG-RAN	*Virtualized NG-RAN*
VM	*Virtual Machine*
WAN	*Wide Area Network*

CONTENTS

1 INTRODUCTION

Although the data traffic of mobile terminals is rapidly increasing, the consumer market of mobile broadband services will be saturated in North America, Europe, and East Asia (ZHOU et al., 2016a). Meanwhile, the growing popularity of Machine-Type Communication (MTC) terminals and applications of vertical enterprises poses an increasing demand for various services from mobile networks. However, legacy mobile networks are mostly designed to provide services for mobile broadband consumers and consist of a few adjustable parameters, such as priority and Quality of Service (QoS) for dedicated services. Therefore, mobile operators present a challenge to getting deeper into these emerging vertical services with different network design and development service requirements. For example, the dedicated network for a railway company involves coverage along the railway with high-speed mobility management. However, it exhibits an apparent difference from an electricity metering company, which only requires small-volume data transmission but massive connections at static positions. Some vehicle communication services are strictly delay-sensitive, while some video surveillance services require stable and immobile high bandwidth. Recently, Network Functions Virtualization (NFV) technology has been proposed to decouple the software and hardware of network elements to simplify service development (ZHOU et al., 2016a).

A study by the European Telecommunications Standards Institute (ETSI) shows that NFV and Software-Defined Networking (SDN) could shorten time to market and facilitate innovations in the technical field (e.g., saving maintenance cost, auto-scaling, enhancing system resilience) (ETSI, 2014). Nevertheless, the products and service types from operators are still limited. The concept of Network Slicing (NS) was proposed to allow the independent usage of a part of network resources by a group of mobile terminals with special requirements (NGUYEN; DO; KIM, 2016; BERTENYI et al., 2018). In this context, NS can enrich operators' products for vertical enterprises, provide service customization for emerging massive connections, and enhance the control given to enterprises and Mobile Virtual Network Operators (MVNOs).

NS aims to logically separate the network functions and resources within one network entity according to specific technical or commercial demands. Although NS is still nascent, similar techniques already exist. Among them, Institute of Electrical and Electronics Engineers (IEEE) 802.1Q, Virtual Local Area Networks (VLANs), which can be regarded as the ancestor of NS, provide a single broadcast domain to bring together a group of hosts possibly having no local and physical connectivity but sharing common interests (3GPP, 2019). Moreover, regarding the field of fixed networks, Internet Engineering Task Force (IETF) RFC 4026, which is also known as a Virtual Private Network (VPN), is another form of NS that could guarantee the QoS and security requirements for logically independent sessions (ETSI, 2016). However, in cellular networks, the realization of NS faces significant challenges since more parameters, such as mobility and authentication management in the control plane and session and charging management in the user plane, must be customized for a group of connections as a logical network.

Mobile operators can customize networks according to various mobile service requirements using NS integrating with NFV and SDN. Furthermore, these operators can lead to a more cost-effective way to build dedicated networks. The Third Generation Partnership Project (3GPP) initiated a technical study into NS to specify service and operational requirements (MEREDITH; ALESSI; PETRY, 2015). Vendors such as Ericsson, Huawei, Nokia, and ZTE have also published white papers about NS to introduce the realization of NS into 5G (3GPP, 2017). NS has been implemented in fixed networks to logically separate them, allowing slice owners to manage their networks (ALLIANCE; HATTACHI; ERFANIAN, 2015a). Another case is NS for emergency communications, which provides dedicated and priority resources to users for emergency communications, even in overwhelming scenarios (ZHOU et al., 2016b). To obtain such adaptability in the 5G network, 3GPP defined interfaces to connect the 5G network to the non-3GPP network, enabling the mobile network to integrate with any other network. Non-3GPP connections are crucial for the Internet of Things (IoT) ecosystem, providing diverse connectivity options for effective IoT deployment. Technologies like Wi-Fi, Bluetooth, and Zigbee enable seamless connectivity for IoT devices in environments where traditional cellular networks are impractical. This flexibility supports reliable, efficient communication in various settings, enhancing the overall IoT infrastructure and ensuring scalable growth. These technologies facilitate short-range communication, essential for the high-density, low-power needs of IoT devices, making them integral to the expansion and functionality of IoT networks (LINNARTZ et al., 2022). However, due to the scattered service models across Radio Access Networks (RANs), Core Networks (CNs), transport networks, and complex protocols in tens of 3GPP interfaces, the realization of mobile NS seriously needs to catch up to its counterpart in fixed networks.

Specifically, there are still some requirements to be an end-to-end service description of the mobile network for the northbound interface to deploy or manage a multi-vendor network slice across the domains with thousands of parameters, and operators steal problems with Management and Orchestration infrastructures as the way that is made nowadays (WYSZKOWSKI et al., 2024). 3GPP also initiated study Management and Orchestration (MANO) tools and architectures to coordinate the infrastructure efficiently. ETSI created the Zero-Touch and Service Management (ZSM) group in the same straight line to study zero human interaction network management and deployment. Finally, the challenges of the high costs and difficulties that come from the traditional way of building and operating mobile networks, such as inflexibility, static networks, and difficulty adapting (DEVLIC et al., 2017), need to be overcome.

1.1 Motivation

Despite the envisioned transformations, significant challenges are presumed to improve networks' adaptability and availability skills optimized for their respective requirements. Therefore, the main motivations of this work stem from the observation that there is a current lack of comprehensive guidelines that would help address the challenges related to the design organization and allocation process of network slices aligned with provisioning standards defined by entities, considering the concepts of automation and real-time network slice provisioning of ZSM on a Network Slice as a Service platform (WYSZKOWSKI et al., 2024). The design process is an essential stage of the slice lifecycle during the preparation phase. Specifically, it is conducted after obtaining a Network Slice Instance (NSI) allocation request from the Communication Service Consumer (CSC) in the ordering phase and directly precedes the NSI and Network Sub-Slice Instance (NSSI) operation phase at the Communication Service Provider's (CSP) premises, aiming for the execution flow to be carried out with minimal human interaction possible.

Designing an NSI as a service requires many details that are prerequisites for provisioning and operation. These details include deciding the internal structure, interfaces, and connections of an NSI topology, planning physical and virtual resources, deployment environments, operational configurations, end-to-end link definitions, and quality of metrics service to be monitored, thus integrating and interfacing among the three network domains (Core, RAN, and Transport) (WYSZKOWSKI et al., 2024). Currently, available specifications are limited to dictating SLA/SLS-based provisioning APIs and do not cover the mentioned design challenges. Precisely, standards do not specify which design models should be used or the design procedures that transform abstract SLA/SLS statements into an end-to-end slice integration and relation capable of meeting them (WYSZKOWSKI et al., 2024).

For the definition and construction of an NSaaS, the following challenges of existing network MANO solutions have motivated the adoption of the ZSM concept and further automation of Network Slice as a Service (NSaaS) (GIANNONE et al., 2019):

- Network Complexity: Massive Internet of Things (mIoT) connectivity, emerging services, and new 5G/6G technologies result in highly heterogeneous and complex mobile networks and significantly increase network orchestration and management complexity.

- New Business-Oriented Services: New services will be available in future networks, which should be quickly implemented to meet business opportunities. The NSaaS concept allows for the agile and straightforward deployment of new services and key-enabling technologies such as NS, NFV, and SDN.

- Performance Improvement: Diverse QoS requirements and the need to reduce operational costs and improve network performance trigger robust network operation and service management solutions.

- Revolution for Future Networks: Even though 5G networks are not fully available worldwide, numerous activities have been dedicated to developing future 6G wireless systems. New technologies, services, applications, and IoT connections will be available, making the future network very complex and complicated to manage efficiently by conventional MANO approaches.

Based on the analysis conducted in the related work and detailed in Chapter 3, the scientific community and industry have been trying to develop network slice orchestration as a service. However, predominantly, the works exhibit a decoupling between the architectures for end-to-end network slice definition based on SLA/SLS and the architectures for orchestration and management of end-to-end network slices as a service, resulting in a partitioning between them. From this perspective, the evaluation of the two topics is carried out separately.

For end-to-end network slice definitions, the literature presents Jiang et al. (JIANG; ANTON; DIETER SCHOTTEN, 2019) and Li et al. (LI et al., 2018), showcasing Q-learning algorithms to facilitate resource allocation in sub-slices. Abbas et al. (ABBAS et al., 2020) present a network slice framework for controlling RAN and Core resources. Lastly, (WYSZKOWSKI et al., 2024) discuss a standards organization tutorial for automating the Network Slice design process.

For development, orchestration, and management projects of end-to-end network slices, the literature presents Theodorou et al. (THEODOROU et al., 2021), who propose a zero-touch framework for automating service assurance of cross-domain network slices. Bega et al. (BEGA et al., 2020) introduce an AI-based framework for managing various phases of the network slice lifecycle. Moreover, projects such as ONAP (FOUNDATION, 2022), 5G- Growth (BARANDA et al., 2020), and 5G-Zorro (BREITGAND et al., 2021) offer several automation proposals toward the network slice instantiation but do not address the topic of end-to-end slice instantiation.

Based on the research conducted in the literature, a gap and, consequently, an opportunity for research development on the topic of end-to-end network slice definition and instantiation is noticed to: (i) defining computational and virtual resources of network slices based on SLA/SLS requests, (ii) linking end-to-end network slices by connecting and relating the three domains of a network (Core, RAN, and TN) and (iii) automating management and orchestration during the lifecycle of an end-to-end network slice. As presented in the context, Section 1.2 addresses the research study design developed through the research question from the gaps and opportunities presented

1.2 Research Question

We defined a research question to help guide this work's evolution, i.e., delimiting the scope and contributions of the proposal.

Research Question (RQ): How to orchestrate and integrate the standardized components to provide network slice as a service?

Hypothesis: *Orchestration and integration must be done from the definition of interfaces for the relationship among the various components and domains distributed in the network and the purpose of hierarchies of responsibilities to define the area of operation of each function within the orchestrator*

Based on the hypothesis, the presented study proposes an architecture of a platform for network slicing as a service named **Network Slice as a Service Platform (NASP)**. This architecture aims to resolve three major problems described above that are breakdown in five items where the architecture aims to (i) define the translation of business rules templates for the description of subnet instances; (ii) to propose a hierarchy of responsibilities for deploying slices; (iii) to use interfaces for management functions and infrastructure control agents; (iv) application of Continuous Integration and Continuous Delivery (CI/CD) concept and Infrastructure as a Code (IaC) strategies for integration between management and physical instances, and finally; (v) telemetry and observability flow. In this way, the management of end-to-end network slices must be planned, for example, the vertical and horizontal relationship among instances of sub-network slices. Finally, NASP must be transparent for 3GPP and non-3GPP networks, i.e., the platform must orchestrate and manage requests to create slices of any use cases, leaving it to the system to adapt and choose the subs networks.

1.3 Organization

This document is structured into five chapters. After the introduction presented in Chapter 1, the concepts related to this dissertation are presented in Chapter 2, introducing the area of mobile telecommunication, the functions of virtualized networks, the network slice, and other concepts that help in understanding this proposal. Followed by, Chapter 3 discusses the related work to present the state-of-the-art involving elasticity for the mobile telecommunications core. Chapter 4 presents the proposed model architecture, design decisions, and supplementation. Succeeded by, Chapter 5 demonstrates the prototype, describing technologies and decisions. Moreover, Chapter 6 shows the evaluation methodology and the NASP prototype. Next, Chapter 7 shows the results and analysis. Finally, in Chapter 8, the final considerations, emphasizing expected contributions, future work, and published works.

2 BACKGROUND

Mobile telecommunication is going through a transition, and new concepts of network programmability, network slicing, and network as a service are being applied to make this transition possible. In this context, a new generation of mobile telecommunications (5G) was defined (3GPP, 2019). This chapter presents the main concepts applied by the latest generation of mobile telecommunication.

2.1 5G Generation Mobile Networks

The current reality of networks and future projections of mobile networks caused the requirements of these networks to be redefined based on the new services and markets (AFOLABI et al., 2018). 3GPP, with Release 15 (3GPP, 2019) and improvement with Release 16 (3GPP, 2018a), describes these requirements given three classes of services, as follows:

Enhanced Mobile Broadband (eMBB): provides high connection and traffic density, mobility, and data rate. For each instance, the downlink must be at least 50 Mbps in open locations and at least 1 Gbps indoors (5G Local Area Network - 5GLAN), and half of these values are for uplink.

Ultra-Reliable and Low Latency Communications (URLLC): has low latency and high availability in the communication between services. For each instance, it must have 99.9999% reliability in the remote control for process automation with a minimum rate of 100 Mbps and an end-to-end latency of a maximum of 50ms (3GPP, 2021).

Massive Internet of Things (mIoT): considers many scenarios to support a high density of devices. For these cases, it is necessary to include operational aspects that enable various IoT devices (3GPP, 2021).

Schematically, the 5G system uses the same elements as the previous generations: User Equipment (UE), itself composed of a Mobile Station and a Universal Subscriber Identity Module (USIM), Radio Access Network (NG-RAN), and Core Network (5GC). However, the 5G system must be highly adaptable to support the abovementioned services. Therefore, a Service-Based Architecture (SBA) was designed and integrated with the 5G New Radio. This architecture separates the network functions into services at a granularity that guarantees single accountability (3GPP, 2018b).

2.1.1 Core Network

The 5GC architecture relies on an SBA framework. The elements of this architecture are based on Network Functions (NFs) rather than on traditional Network Entities. Any given NF via interfaces of the standard framework offers services to all authorized NFs and to any consumers permitted to use these provided services. In this context, the SBA framework offers modularity and reusability for 5GC. The 5GC is accessed by the Access and Mobility Management Function (AMF) in the control plane through UE and Next Generation Node B (gNB), representing a RAN. In addition, the User Plane Function (UPF) in the data plane handles the user data for communication on the Data Plane. The reference point between the access and the core networks is called the Next generation (NG). This reference point constitutes several interfaces (mainly N2 and N3). Figure 1 shows other functions of 5GC, described in the following.

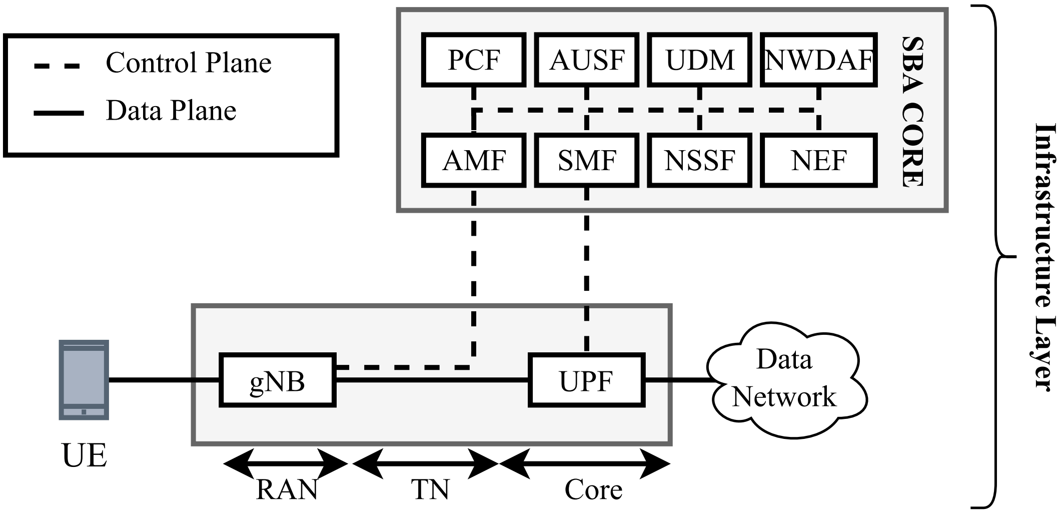


Figure 1 – 5G Core architecture.

Session Management Function (SMF): is responsible for session control, the definition of Internet Protocol (IP) addresses of UEs, selection and control of UPF, and traffic direction setting in UPF to route the data to the correct destination.

Network Repository Function (NRF): manages all NFs, including registration, deregistration, authorization, and discovery.

Network Exposure Function (NEF): externally and internally exposes data that other services can consume.

Unified Data Management (UDM): is responsible for data management (UEs, Policies, Sessions, etc.) in a unified way.

Network Data Analytic Function (NWDAF): collects data from the 5G core and provides analytics to support network automation, closed-loop operations, self-healing, experience improvement, and reporting.

Network Slice Selection Function (NSSF): is a dedicated network function for selecting Network Slices with specific characteristics to meet a well-defined network scenario.

Authentication Server Function (AUSF): enables services for the unified authentication of 3GPP and non-3GGP accesses.

Policy Control Function (PCF): is the service responsible for defining and delivering policies for network services.

Non-3GPP Interworking Function: This element of the 5G SBA (Service Based Architecture) is responsible for interworking between untrusted non-3GPP networks and the 5G Core.

The isolation and preparation for each environment considering tenant requirements and services provided by the core of mobile networks introduced the demand for the network slicing concept. In this context, NS is discussed in the following subsection.

2.1.2 5G Network Slicing

3GPP defines network slicing as a logical network that provides specific capabilities and characteristics of a network composed of three sub-networks: radio access, transport, and core. From an operator perspective, providing customer service in the 3GPP domain and non-3GPP domain, such as Service Gateway internal (SGi) Local Area Network (LAN) and fixed access network, is essential. End-to-end (E2E) network slicing is a logical network spanning 3GPP and non-3GPP domains. It is essential to provide customized E2E network service complying with agreed SLAs.

NS is a concept for running multiple logical customized networks on a standard shared infrastructure complying with agreed SLAs for different vertical industry customers and requested functionalities. NS needs an E2E architecture to be designed from an E2E perspective, spanning other technical domains (e.g., device, access network, core network, transport network, and network management system) and multiple vendors. In this context, NS contains distinct technical domains and Software Defined Orchestrations (SDOs) working in parallel to provide a slicing solution under their area of competence. As a result, the technical content is fragmented, i.e., it forms an E2E solution that requires significant work concerning cross-SDO and open-source project cooperation and coordination. NS was outlined as a vision for the 5G capability empowering value creation in NG-MN 5G (ALLIANCE; HATTACHI; ERFANIAN, 2015b). Therefore, it became one of the key features specified in 3GPP to be supported by the 5G system (3GPP, 2018b). Moreover, the ability to provide heterogeneous environments that guarantee all SLAs from each tenant is a crucial part of accomplishing the NS goal.

2.2 Network Slice Types

The NEtwork Slice Type (NEST) is a set of attributes that can characterize a network slice/service type. The values are assigned to express a given set of requirements to support network slice use cases. The NEST is an input to the network slice preparation performed by the Network Slice Provider (NSP), where every Provider deploying 5G networks will deploy a 5G Network Slice fitting to the use cases. Attributes are the smaller technical requirement descriptions of a NEST describing network performance, such as throughput, latency, and reliability to network functionalities and interfaces as specified in Global System for Mobile Communications Association (GSMA) NG.116 (GSMA, 2022a).

SLA is a commitment to provide network services between an operator and a consumer. The consumer declares communication service(s) requirements to the operator. These requirements are called SLS. To guarantee an SLS with each consumer, the network slice corresponding to each CSC reserves the appropriate amount of resources (e.g., radio resource, computer resource). After, the network slice deploys functions, such as UPF, at the right location, especially for low-latency communication (GSMA, 2022b).

A set of attributes characterizing a network slice/service type is specified as Service Profile in 3GPP TS28.541 (3GPP, 2018b). Moreover, the Generic network Slice Template (GST) in GSMA NG.116 (GSMA, 2018) expresses SLS, i.e., a set of service level requirements associated with SLA to be satisfied by a network slice. The Service Profile is assumed to be utilized in the 3GPP 5G System. From an E2E perspective, GST is specified as a standard set of attributes from the 3GPP and transport domain. The alignment of characteristics between the Service Profile and GST is proceeding. Table 1 shows an example of attribute details specifications for an eMBB slice (GSMA, 2022a).

Table 1 – eMBB Slice SLA Attributes.

Attribute	**Value**
Availability	99.999
Multimedia Telephony service (MMTel)	Supported
Session and Service Continuity (SSC)	Mode 1 support
Data Network (DN) access	{Direct access to Internet}, {IMS, Local traffic (no Internet access)}
Support data network	Internet DN and IMS
Slice quality of service	1, 2, 5, 6, 7, 8, and 9

Physical isolation is required to achieve some of the attributes and SLAs required for each template. Isolation is one of the key expectations of NS. A Network Slice Instance (NSI) may be wholly or partly, logically and physically, isolated from another network slice instance. Only VM-type virtualization technologies are explored for isolation at the virtualization level, but not containers, as with other types of virtualization technology. Isolation requirements on containers are a work in progress as current native solutions do not offer much, and additional support is required. ETSI NFV is working on the SEC023 GS (ETSI, 2019) specification dedicated to isolation requirements. NS allows the concurrent execution of multiple NSIs on top of a shared infrastructure, satisfying their service Key Performance Indicators (KPIs) while guaranteeing their independence. Using a single shared infrastructure makes isolation an essential requirement for NS.

Isolation can be defined as the ability of an NSP to ensure that congestion and lifecycle-related events (e.g., scaling in/out) on one NSI does not negatively impact other existing NSIs (3GPP, 2021). Moreover, isolation in NS is a multi-faceted problem with multiple dimensions that must be carefully addressed. The dimensions include performance, management, and security/privacy. Isolation in terms of performance means ensuring that service KPIs are always met on each NSI, regardless of the workloads or faults of other existing NSIs. Isolation concerning management means ensuring that individual NSIs can be managed as separate networks, with the possibility of the Network Slice Controller (NSC) retaining control of the slice. The following subsections provide more details on these isolation dimensions.

2.3 Network Slice as a Service

Network Slice as a Service (NSaaS) is the ultimate goal that represents a service delivery model that allows the operators to provision customized network slices to individual customers and eventually enables these customers to access some network slice management capabilities. It is up to the operator to decide which specific management capabilities are available to each customer, typically exposed through customer-facing Application Programming Interfaces (APIs) (e.g., TM Forum APIs) (GSMA, 2022a). 3GPP provided an API structure over network functions to distribute the management load and create the machine-to-machine interface communication, as shown in Figure 2, with RAN, transport, and Core components.

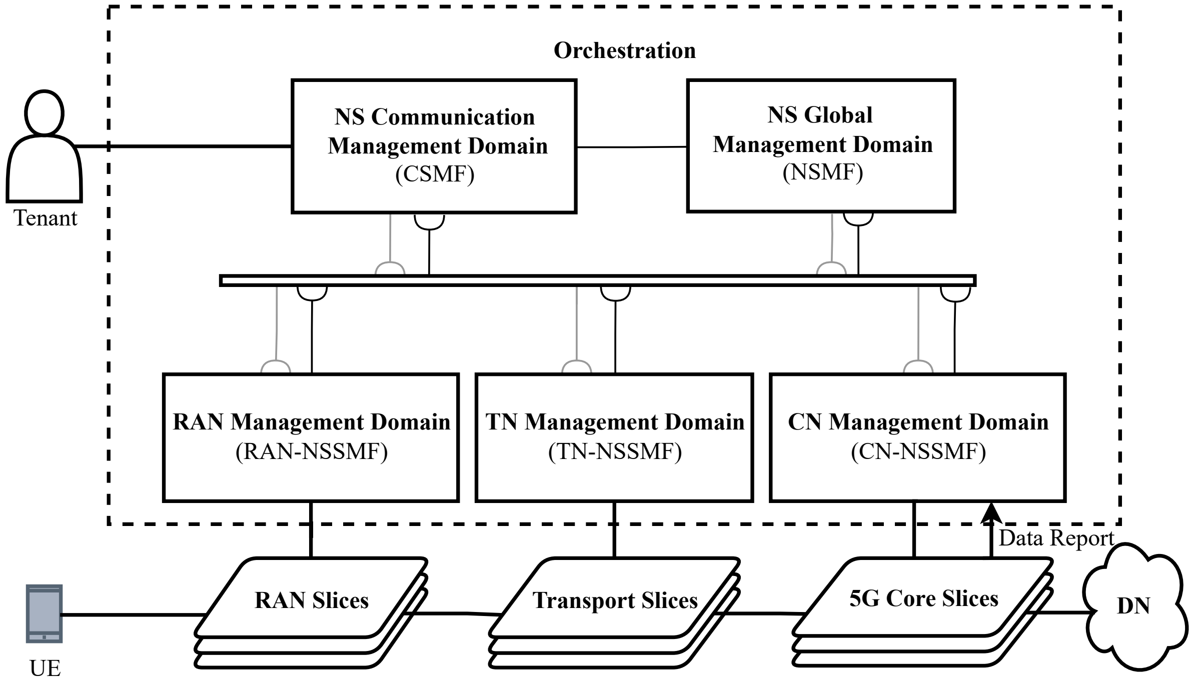

Orchestration
Tenant
NS Communication Management Domain (CSMF)
NS Global Management Domain (NSMF)
RAN Management Domain (RAN-NSSMF)
TN Management Domain (TN-NSSMF)
CN Management Domain (CN-NSSMF)
Data Report
UE
RAN Slices
Transport Slices
5G Core Slices
DN

Figure 2 – Network Slice Orchestration Architecture.

- **Communication Service Management Function (CSMF)**: translates the communication service-related requirement from Tenant to network slice-related conditions. Communicate with Network Slice Management Function (NSMF).

- **Network Slice Management Function (NSMF):** manages and orchestrates NSI. It derives network slice subnet-related requirements from network slice-related conditions communicating with the Network Slice Subnet Management Function (NSSMF) and Communication Service Management Function.

- **Network Slice Subnet Management Function (NSSMF):** manages and orchestrates NSSI communicating with the NSMF.

2.3.1 Management and orchestration

The management and orchestration process is performed over the entire lifecycle of an NSI. The process comprises three steps: Preparation, Operation, and Decommission. Figure 3 provides a high-level view of the management and orchestration process of an NSI. In each of the steps, the nature of isolated Network Slices aids in increasing the speed of the process, as there are fewer dependencies to consider. The three-step management and orchestration process are described as follows.

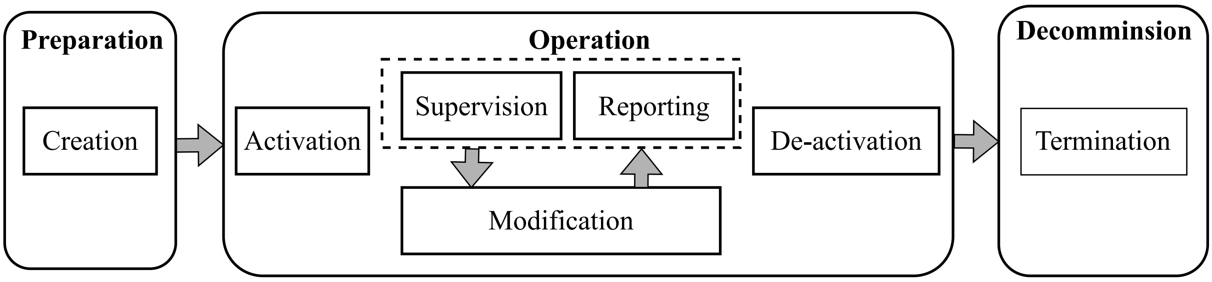


Figure 3 – Network Slice Lifecycle.

Preparation: Network Slice “blueprints” or “templates” are used for order creation and activation of an NSI. During NSI creation, all needed resources are allocated and configured to satisfy the Network Slice requirements (ROMMER et al., 2020).

Operation: includes the activation, supervision, performance reporting (e.g., for KPI monitoring), resource capacity planning, modification, and de-activation of an NSI. Activation makes the NSI ready to support communication services. The supervision and performance reporting include, e.g., monitoring, assurance, and reporting of the performance according to the KPIs agreed as part of SLAs for NSI. In this context, NSI modification could include, e.g., capacity or topology changes. The modification can consist of the creation or modification of NSI resources. Moreover, the NSI modification can be triggered by receiving new Network Slice requirements or as the result of supervision/reporting.

Finally, the de-activation includes actions that make the NSI inactive and stop the communication services (ROMMER et al., 2020).

Decommission: NSI provisioning in the decommission step includes decommissioning non-shared resources if required and removing the NSI-specific configuration from the shared resources. After the decommission step, NSI is terminated and does not exist anymore (ROMMER et al., 2020).

Each Network Slice can be a fully working network with all the functions and resources required for independent service. Tenants can be granted visibility with a NaaS business model of their Network Slice, modify it to suit their changing needs, or create new Network Slices for new business opportunities. To achieve this level of service and management, zero-touch automation is required to fully automate the provision of all resources.

2.3.2 Zero Touch Network & Service Management

The key deployment of 5G and network slicing has triggered the need for a radical change in how networks and services are managed and orchestrated, moving to a ZSM. In particular, there is a need to handle the increase in the overall complexity resulting from the transformation of networks into programmable, software-driven, service-based, and holistically managed architectures and the unprecedented operational agility required to support new business opportunities enabled by technology breakthroughs, such as NS (ETSI, 2021). These new deployments come with an extreme range of requirements, including massive infinite capacity, imperceptible latency, ultra-high reliability, personalized services with dramatic improvements in customer experience, global web-scale reach, and support for massive machine-to-machine communication (ETSI, 2021).

The complete end-to-end network and service management automation has become urgent in delivering services with agility and speed and ensuring the economic sustainability of the vast services offered by digital service providers. The ultimate automation target is to enable larger autonomous networks, which high-level policies and rules will drive. Without further human intervention, these networks will be capable of self-configuration, self-monitoring, self-healing, and self-optimization. All this requires a new horizontal and vertical end-to-end architecture designed for closed-loop automation and optimized for data-driven machine learning and artificial intelligence algorithms (ETSI, 2021).

This chapter discussed the theoretical background required to comprehend the basis of a mobile network and its future path. The next chapter will discuss the related work and most important projects currently in development.

3 RELATED WORK

This chapter presents related work to show state-of-the-art regarding slice creation and orchestration for 5G services. Section 3.1 presents the methodology for selecting articles for the basis of the work. Section 3.2 offers the related work to highlight the research carried out in the area. Section 3.3 describes the articles that solve specific questions and contributions unrelated to platforms and open-source tools. Finally, in Section 3.4, a comparison of the models proposed is presented to highlight the main points of interest for the development of this work.

3.1 Methodology and work selection

The article selection was based on SCOPUS and Google Scholar search engines, going through IEEE, ACM and government databases. The first step was to look for articles that mentioned Network Slice as a Service, Orchestration, and Management or ZSM in the 5G and Beyond 5G (B5G), reaching 4150 articles, through the following search string: (5G OR (3GPP AND "release 15") OR ("mobile" AND "telecom" AND "next generation")) AND (core OR SBA OR "Service Based Architecture") AND (ZSM OR NSaaS OR NaaS OR OSM). Once the initial base was set up, the duplicates were removed, reaching 3750 articles. We removed publications older than four years from the area to eliminate outdated research that no longer adheres to the status quo, getting 2390 articles. Subsequently, an analysis was carried out on the titles by keywords, maintaining or excluding articles based on two-word lists, reaching 315 articles. The summaries keeping the works that proposed a model were filtered, considering expressions: architecture, approach, or prototype for the 5G or B5G network, reaching 21 articles. Finally, we read these articles, and those that presented solutions that allowed the dynamics of 5G or B5G network services reached seven articles presented in Section 3.2.

3.2 Selected Projects

This section presents related work on 5G network slicing solutions that utilize virtualized infrastructures based on cloud computing concepts. All the works described below have some level of NS orchestration. For example, in the most basic scenario, the static configuration replicates the same pre-established network parameters in the initial state of a network slice's lifecycle. However, this literature review was directed to works that present the functions of the 5G with support for SBA, which natively has the NSSF function, as well as components for managing network slices, such as NSSMF and NSMF. Furthermore, projects with support to create models (*templates*) of slices of networks were discussed, allowing the modification of the execution state of these slices. Finally, the main analysis of the literature works refers to the dynamic orchestration of the network slices. Therefore, the concept of dynamic orchestration in the form *Partial* and *Total* was introduced to clarify and organize the works found in the literature that would consider this topic. *Partial* orchestration refers to the functionality available in cloud computing tools to auto-scale computing resources to share virtualized infrastructure with different services or specialized configurations within the 5G to support NS. The *Total* orchestration concerns solutions, allowing for the dynamic re-configuring of 5G in an integrated manner with the resources of the virtualized infrastructure (GRINGS et al., 2022) without any service downtime.

The main open-source cloud computing initiatives used by telecom operators are *Open Network Automation Platform* (ONAP) (FOUNDATION, 2022) and Open Source *Management and Orchestration* (MANO) (OSM) (ETSI, 2022). ONAP is a comprehensive platform from *The Linux Foundation*, comprising modules for orchestrating, managing, and automating services in real time. This platform is policy-driven for physical and virtual network functions, enabling rapid automation of new services and lifecycle management of these functions for 5G networks. OSM is an initiative of the *European Telecommunications Standards Institute* (ETSI), aiming to align development activities with the evolution of the ETSI *Network Function Virtualization* (NFV) standard, allowing operators and providers to have an ecosystem based on NFV architecture HAND. The two initiatives have features that consider the scope of NSs, such as 5G core support, the possibility of using models to create network slices, and support for self-scaling of computing resources using cloud computing features based on Kubernetes. However, although ONAP and OSM implement isolated *Virtual Network Functions* (VNFs), these platforms do not have an adaptive control over these functions, i.e., these tools do not allow the reconfiguration of the 5G core functions dynamically and integrated with the resources that manage the virtualized infrastructure. For example, to apply a new configuration to a slice of the network, it is necessary to terminate the lifecycle of that slice and start a new slice of the network, interrupting the service provided. Based on this characteristic, the network slice orchestration of these initiatives is classified as dynamic *Partial*, as seen in Table 3. The European 5G-TOURS (GARCIA-AVILES et al., 2020) project has the same characteristics as the ONAP and OSM initiatives.

Works such as 5GZORRO (BREITGAND et al., 2021) and 5Growth (BARANDA et al., 2020) present dynamic orchestration models *Partial*, with vertical and horizontal elasticity for NS, using the MANO architecture. Moreover, 5GZORRO and 5Growth explore the control of the NS lifecycle considering the three domains, i.e., access network, transport, and core, following the standards proposed by the 3GPP and ETSI entities. However, these works approach the 5G core in a unique and immutable way, limiting the management of resources made available by the SBA architecture proposed by 3GPP. In this context, the 5G-COMPLETE tool (GKATZIOS et al., 2020) also uniquely considers the 5G core. Such works fail to comply with certain service level agreements, affecting the QoS provided.

The literature on NS also presents works with a broad perspective. For example, the European 5G-CLARITY project (ORDONEZ-LUCENA et al., 2021) aims to develop a new management plan based on SDN principles, NFV, and the use of artificial intelligence algorithms to enable NSs in neutral equipment. Additionally, DYSOLVE (KUKKALLI et al., 2020) is a dynamic resource allocation proposal for dynamic 5G NS for a vehicular emergency scenario. This solution aims to cooperatively allocate radio and transport network resources to operators to optimize the cost of the network share, ensuring service availability. In these works, the orchestration of the virtualized infrastructure is present, i.e., self-scaling of the computational resources characterizing a *Partial* dynamic orchestration of the NS. However, due to the expanded perspective, these works do not consider the NS functionalities of the 5G core nor the slice models defined by the GSMA to assist in managing these network slices.

One initiative that stands out in the dynamic orchestration of network slices is the OpenSlice (TRANORIS, 2021) project. This open-source solution is based on OSM and allows users from different network slices to explore service specifications offered for cloud computing infrastructure. In addition, OpenSlice allows NFV developers to integrate and manage VNF artifacts and network services. In this way, the OpenSlice project is classified with a dynamic orchestration *Total* for NSs, as seen in Table 3. However, OpenSlice does not yet support specialized functions and features of the 5G core, restricting its applicability. Analyzing the NS literature, it is clear that no solutions support the reconfiguration of the 5G core dynamically after the initialization of a network slice to ensure compliance with service level agreements in a 5G network. The solution proposed supports these characteristics.

Table 2 – Summary of Related Work for Projects.

Works	Characteristics					
	5G Core SBA	Project Model	Architecture	Service Automation	E2E Service LC management	NSaaS
ONAP	H	H	H	M	M	L
OSM	M	H	H	M	M	L
5G-Tours	H	H	H	M	M	L
5GZORRO	H	M	M	L	L	L
5Growth	L	H	H	L	L	L
5G-COMPLETE	L	L	H	L	L	L
5G-Clarity	L	L	H	L	L	L
6G Flaship	H	M	M	H	H	L
Open Slice	H	H	H	M	M	L

*Maturity: *Low* (L) *Medium* (M) *High* (H)

The open-source projects described can help us understand the relationship and state-of-the-art for applied tools and platforms. In the next section, we describe related research that is not necessarily open-source projects but presents an available model for study.

3.3 Selected Articles

This section presents related work on 5G NS solutions. These solutions usually analyze specific research questions and provide a more detailed description of the evaluation and results. All the articles described in the following section are related to NS orchestration at some level.

Wyszkowski et al. (WYSZKOWSKI et al., 2024) proposed a complementing in an NS design based on SLA/SLS. Additionally, the authors proposed a generic taxonomy of building blocks for designing NSIs. Therefore, the author's contribution fills this void by providing original concepts and systematization's defending detailed definition when requesting an NSI. Finally, the author proposed a framework of defined design activities and automation methods to manage complex networks and explores the integration with algorithms and network infrastructures as future projects.

Jiang et al. (JIANG; ANTON; DIETER SCHOTTEN, 2019) present a unified framework with generality to integrate AI to conduct intelligent tasks for all network aspects, ranging from radio channels to signal processing, from resource allocation to NS orchestration, and from local control to end-to-end optimization, in line with the principles of ZSM. The intelligent slicing concept was introduced with the flexibility to instantiate, deploy, scale, reconfigure, and transfer AI functional modules on demand. Such slices can be deployed in an arbitrary network entity, facilitating problem-solving by selecting the best-optimized algorithm explicitly for this problem. Additionally, two example slices, i.e., neural network-based Multiple Input Multiple Output (MIMO) channel prediction and security anomaly detection in industrial networks, were illustrated to demonstrate the proposed framework.

In contrast to Jiang et al. (JIANG; ANTON; DIETER SCHOTTEN, 2019), which leverages the concept of intelligent slices to conduct different tasks with the flexibility to accommodate arbitrary AI algorithms, Bega et al. (BEGA et al., 2020) propose an AI-based framework for network slice management by introducing AI in the distinct phases of the slice life cycle to achieve the ZSM goal. The authors propose practical deep learning architectures that can help solve complex networking problems by considering three case studies: scheduling slice traffic at RAN, resource allocation to slices in the network core, and admission control of new slices. Furthermore, the authors conclude that AI has a clear potential to become a cardinal technology for future-generation zero-touch mobile networks and illustrate the typical high gain one can expect from integrating AI in NS.

Li et al. (LI et al., 2018) propose applying Q-learning and Deep Reinforcement Learning (DRL) schemes, which remove data preparation effort as in subslices, to solve resource allocation problems in NS scenarios (radio resource slicing and core slicing). Moreover, the authors perform extensive simulations to demonstrate that the proposed schemes significantly reduce the sum cost compared to other baselines. Finally, Li et al. also discuss all the possible challenges in applying RL for optimized NS to achieve a zero-touch environment.

The previous solutions deal with fundamental single-domain slicing challenges. Theodorou et al. (THEODOROU et al., 2021) propose a zero-touch framework for the automated service assurance of cross-domain network slices using Distributed Ledger Technologies (DLT) and AI-driven closed-loop automation techniques. The authors employ trained AI prediction models to forecast network slices' SLA violations supported by a decentralized marketplace for infrastructure resources and network services sharing among multiple providers (FERNAN-

DEZ et al., 2021). As a result, it triggers proactive mitigation actions such as slice extensions over infrastructure resources available in the marketplace (programmatically enabled via smart contracts) alongside the corresponding orchestration workflows. Moreover, the authors evaluate the accuracy of the proposed approach using an experimental prototype for service demand short-term predictions and validate the framework's ability to respond with timely preventive scaling actions.

Ferguson et al. (LARREA; FERGUSON; MARINA, 2023) proposed a framework named CoreKube that implements high availability and cloud-native mobile core system design focused on mobile network control plane which features truly stateless workers (processing units). In addition, the author's extensive stress tests compared with other single-slice core network solutions demonstrate that high availability, scaling, and resilience are key features to achieving 5G network goals. The authors do not explore multi-slice management and orchestration.

Along the same lines, Scotece et al. (SCOTECE et al., 2023) discuss the implementation of 5G network infrastructures using technologies such as NFV and SDN, emphasizing Kubernetes as an orchestrator. It aims to reduce operational costs by eliminating the complexity of traditional solutions like ETSI MANO. The article proposes 5G-Kube, a container-based deployment method using Kubernetes, demonstrating its feasibility in scenarios such as Industry 4.0 and Smart Cities. Finally, the authors concluded a significant cost reduction, faster failure recovery, and escalation time, proving that 5G Core orchestration based on Kubernetes is a good fit for 4.0 Industry and Smart Cities scenarios.

Pointing to E2E frameworks, Dalgitsis et al. (DALGITSIS et al., 2024) introduce a cloud-native orchestration framework designed to maintain network slice continuity for connected and automated vehicles across different network operators, utilizing advancements in C-V2X and edge computing. It employs virtualization, cloudification, and well-defined interfaces for slice federation, aligning with 5G/6G and O-RAN principles and ensuring compliance with GSMA efforts for Edge Federation. An experimental 5G platform was deployed to test the framework, conducting extensive experiments that demonstrated the impact of federation implementation and slice deployment strategies on network performance.

Finally, Abbas et al. (ABBAS et al., 2020) present an intent-based NS framework that can efficiently slice and control the RAN and core network resources. The system allows a user to provide high-level information in the form of a network slice intent, and in return, the proposed system deploys and configures the requested resources. Moreover, the authors apply Generative Adversarial Neural Networks to manage network resources and evaluate the proposed framework by creating several network slices, illustrating the performance improvement regarding bandwidth and latency.

This section discussed the related works and projects describing the topics and relations among each other. The section also discussed the common gaps found in the analyzed works. The next section goes deeper into the projects and papers relation to exploring a research opportunity.

Table 3 – Summary of Related Work for Articles.

Works	Characteristics					
	5G Core SBA	Project Model	Archi-tecture	Service Automation	E2E Service LC management	NSaaS
Ferguson et al., 2023	H	H	L	L	M	L
Wyszkowski et al., 2024	H	L	H	L	L	L
Jiang et al., 2019	H	H	L	H	M	L
Bega et al., 2020	M	H	L	L	H	L
Li et al., 2018	H	H	L	L	L	L
Fernandez et al., 2021	M	H	M	M	L	L
Theodorou et al., 2021	H	H	L	H	H	L
Scotece et al., 2023	H	M	M	H	L	L
Dalgitsi et al., 2024	H	H	M	M	H	L
Abbas et al., 2020	M	H	M	H	M	L

*Maturity: *Low* (L) *Medium* (M) *High* (H)

3.4 Research Opportunity

This section explores the gaps and open issues in the literature regarding end-to-end slice design solutions, automation of new slice instantiation, and finally, concluding with the provision of NSaaS. Firstly, both topics are evaluated through achievements in Open Source projects and published articles, as detailed in Sections 3.2 and 3.2. However, both topics analyze the same characteristics described in the tables, where "5G Core SBA" is related to the use of the 5G Core architecture with the principles of Service-Based Application (SBA), which is evaluated with parameters: Low (L) for articles that work using the 4G core; Medium (M) for articles that work with at least one function of the 5G core; and High (H) for works that present the complete 5G SBA core in their development. The "Project Model" parameter is related to the availability of the software project used for the work's development, where it was evaluated as Low for articles that do not provide, Medium for articles that provide part of the code, and High for articles that present a complete and accessible code repository. The "Architecture" parameter is used to evaluate the implementation architecture of a proposed end-to-end Slice, where the evaluation uses Low for works that do not present an architecture definition; Medium is used for works that present an architecture definition but do not consider the three domains; and High for projects that show the architecture along with the definition of slices that consider the three domains.

The analysis also evaluates automation and ZSM aspects of each project where "Service Automation" concerns the automation stage for creating an end-to-end slice considering ETSI ZSM specifications, i.e., Low is evaluated for works that do not present automation; Medium for works that present partial automation; and High when they present end-to-end automation with all CSP and CSC operations described and implemented. "E2E Service LC Management" concerns the automation of the lifecycle of a network slice considering all stages, i.e., Low is evaluated for works that only perform preparation and instantiation, Medium for works that deal with decommissioning, and High for works that perform Closed Loops and runtime management. Finally, "NSaaS" refers to the provision of network slices as a service, such as Low are works that do not perform such a task, Medium for works that perform in at least two domains, and High for works that implement NSaaS in all three domains.

In conclusion, considering the limitations exposed by the proposed works, it is possible to identify the (i) need for development and improvement of end-to-end network slice design based on the three domains, which in turn must be aligned with SLA/SLS requests, (ii) automation of slice instantiation processes, with ZSM characteristics, and finally, (iii) the definition of an architecture integrating the two problems and providing NSaaS. Therefore, these opportunities are hot and current, given that projects and works are focused on smaller integration solutions and highlight the gap in end-to-end integration.

4 NETWORK SLICE AS A SERVICE PLATFORM ARCHITECTURE

This chapter presents the proposed architecture for NSaaS Platform on New Generation Mobile Networks. Section 4.1 guides the decision-making process for the proposal preparation. Next, Section 4.2 introduces the architecture defined based on the 3GPP technical solutions, the business requests on-boarding, instances resource allocation interfaces, and, at last, monitoring and tracing loops.

4.1 Project Decisions

The project presents the architecture for a platform for NSaaS, considering the end-to-end NS, mapping the definitions of international entities and their relationship, and proposing techniques to supply the gaps among the technical documents. For this, several technical documents from entities such as 3GPP, ETSI, GSMA, and O-RAN Alliance were analyzed, where the most mature and convergent definitions were merged, forming the main blocks of this project. Therefore, the architecture maps the orchestration and management of a multi-slice network working with slicing as a service using the management definitions found in 3GPP. The main purposes can be defined in five topics: (i) translation of business templates to define instantiated network subslices; (ii) use of a hierarchy of responsibilities for the deployment of slices; (iii) definition of communication interfaces between management functions and control agents; (iv) Application of CI/CD concept and IaC strategies for integration between management and physical instances, and finally; (v) telemetry and observability flow.

The translation of business templates for defining instances of network subslices is the main part of onboarding a request for a new network slice. The GSMA template definitions were considered the first information a tenant received from the platform. The translation is performed with high-level descriptions and QoS attributes for choosing the best network NSSI definitions and their physical instances that best adapt to meet the requested needs. Mapping requests for a new slice ultimately generate between three and seven NSSI templates defining their physical instances and computational resources. Therefore, at the end of a network slice request onboarding, the definitions related to the initial request are delivered as output from the platform.

A hierarchy of responsibilities for deploying slices is necessary to create the horizontal relationship between NSSI and the vertical relationship between the slices and their NSSI. The hierarchy between templates was defined since NSSI are directly related to creating the link between the access and transport networks and the control between NSI and NSSI. The GSMA template processed by CSMF is the initial level, sending its attributes for NSMF, where the assignment of global identifiers of a slice (S-NSSAI) is made, and passing the definitions to their domains (RN NSSNF, TN NSSMF, and CN NSSMF), i.e., each function inherit the previous descriptions and assign with their specific attributes, such as input and output addresses, instances and physical resources. Finally, a hierarchy is composed of all definitions of a slice. The global management is performed in NSMF, and the specific management is performed in the subslice management functions. However, the technical documents do not specify the relationship and interfaces between physical instances' management and control functions. This definition of communication interfaces between management functions and control agents is carried out, and RN1, TN1, and CN1 interfaces are defined where the processor templates in the onboard stage are required for the controllers to instantiate and start the physical resources.

Since the physical resources required by the management functions are in code formats, practices such as IaC and CI/CD are needed for control and guarantee homogeneity in the same configuration, achieving the goal of Zero Touch Management by performing Machine-to-Machine (M2M) communication. Tests and monitoring during the infrastructure instantiation process are carried out, alerting the management functions about the process evolution. After instantiating the network slices, it is necessary to perform monitoring to collect information and be able to act if the initial requests are no longer fulfilled. Monitoring is based on observability principles with three main pillars: metrics, logs, and tracing. The information must be collected across all NSSIs for end-to-end network monitoring. Therefore, each domain collects its main information and exposes it to NSMF, where it finally has the global vision of NSI. 3GPP defines the NEF function for exposing information from NSI, so it is defined that all NSI have their NEF responsible for the interface with external components to collect the internal data of the slices.

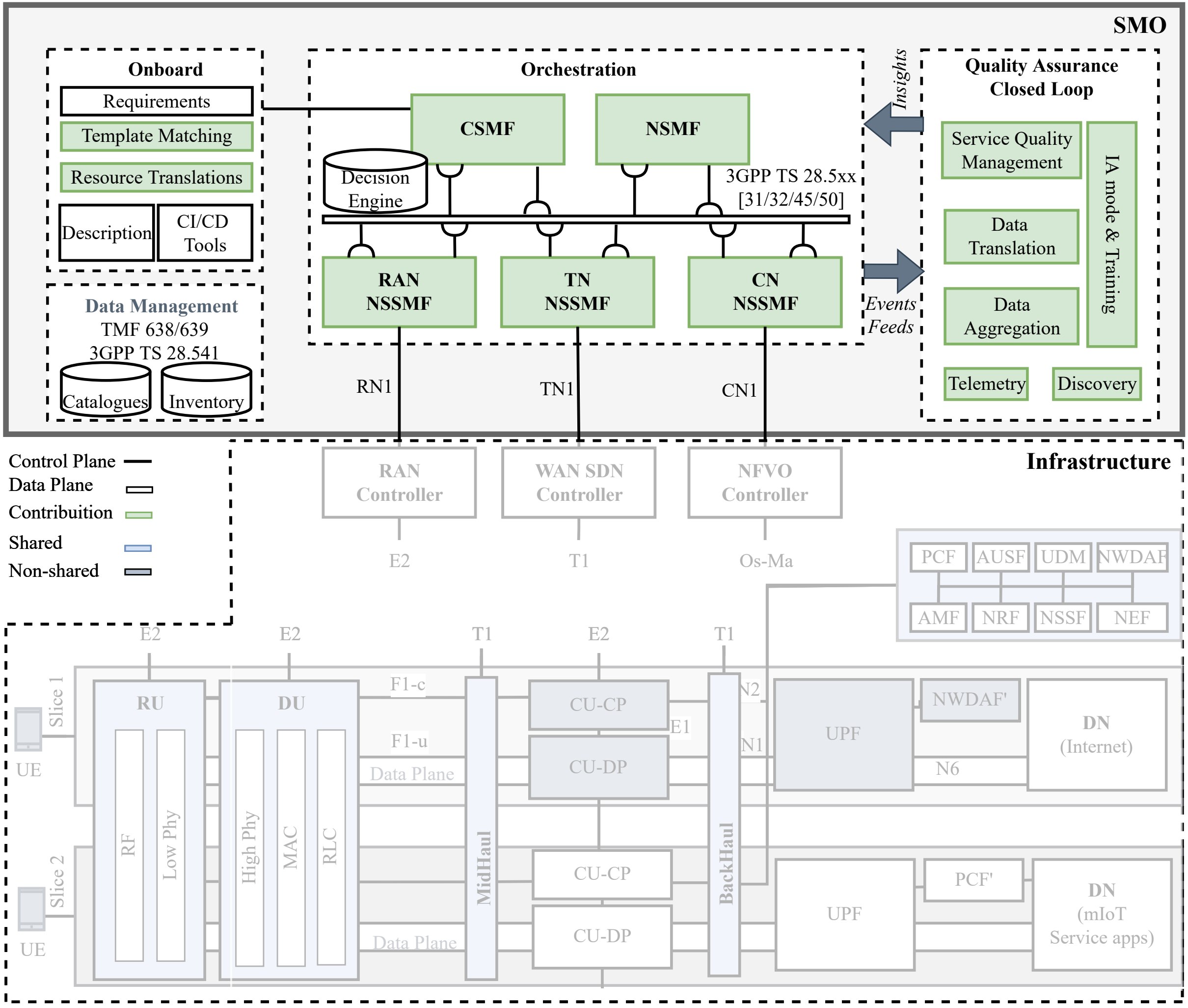
We cover the main gaps for integration between the entities technical documents using the definitions described above. In addition, the project definitions act between opened and overlapping entities definitions, thus clearing up the discussions in the next section. Section 4.2 develops the discussion on the definitions and lists the proposed general architecture, enabling a complete and end-to-end view of the NSaaS platform.

4.2 NASP Architecture

This section explores the proposed architecture of the NSaaS Platform to support and guide slice onboard. Three parts compose this section. Subsection 4.2.1 explains the strategy for tenant requests for Network Slice Templates and Network Slice Definition selection. Subsection 4.2.2 describes techniques for domain resource allocation, and Subsection 4.2.3 shows distributed architectural closed loops to watch, collect, analyze, and make decisions over each slice.

NASP manages three physical and distributed different domains, RAN, Transport, and Core, described by 3GPP and ETSI reference architectures as shown in Figure 4. The NASP architecture has four major components: (i) Onboard; (ii) Orchestration; (iii) Quality Assurance Closed Loops; (iv) Interfaces (RN1, TN1, CN1) to integrate with the controller as shown at the top of Figure 4. The NASP architecture describes the integration and components required on an NSaaS platform to onboard, deploy, and control the entire lifecycle of a network slice through all domains from a mobile network. Furthermore, the NASP architecture proposes a communication flow and interfaces to instantiate all infrastructure necessary to establish an E2E slice, as seen in the lower part of Figure 4. The architecture also proposed the responsibilities for each domain and decision level in each closed loop to pursue SLAs set at the onboard process. In this way, the architecture is based on the responsibilities for each domain and at the decision level defined in the integration process. Each macro functionality of the NASP architecture as a service is described in the following.

Figure 4 – NASP Architecture.



Tenant requests for a network slice are received as input in the Onboard functionality, which translates the requests to infrastructure resources, following the case models of use defined by GSMA. After specifying the use case, Onboard searches the domain-slicing of NSSI models in the catalogue and inventory database to create an NSI model. After that, Onboard outputs a Network Slice Definition consisting of virtual and physical descriptions for each domain.

3GPP defines the functionality of Orchestration and the interface with Onboard. For example, CSMF receives the resource allocation request using an interface that interacts with

NSSMFs for each domain. In addition, NSMF manages the network slice resource allocation request. The Orchestration functionality of the NASP architecture maps and defines the necessary interfaces to complete the resource allocation communication between the subnet managers (RAN NSSMF, TN NSSMF, and CN NSSF) and the domain controllers.

The main function of the Quality Assurance Closed Loop is related to the execution time of the lifecycle of a network slice. In this way, the Quality Assurance Closed Loop is responsible for monitoring, tracking, data analysis, and requesting actions to update a slice from a non-real-time point of view. All these responsibilities refer to guaranteeing the QoS of NS. In this context, the Quality Assurance Closed Loop is formed by six sub-functions: (i) Telemetry, (ii) Discovery, (iii) Data Aggregation, (iv) Data Translation, (v) IA Mode & Training, and (vi) Service Quality Management. These sub-functions work in an integrated way. For example, NSMF, as global slice management, is responsible for end-to-end decision-making about network slice quality. Quality Assurance Closed Loop functionalities provide decisions such as long-term forecasting, adaptability, slice management with non-real-time requirements, and physical and virtual allocation of network slices.

Network controllers in each domain work together to provide end-to-end slicing as a service. For example, the proposed resource allocation interface creates the association and interfaces necessary for an end-to-end allocation. Figure 4 shows the RN1, TN1, and CN1 interface between the domain network slice managers (RAN NSSMF, TN NSSMF, and CN NSSMF) and the domain infrastructure controllers (RAN Controller, WAN SDN Controller, and NFVO Controller). The controllers act in each physical or virtual infrastructure domain using their interfaces. For example, E2 is the RAN controller domain interface, T1 is the transport controller domain interface, and the Os-Ma interface is the main controller domain.

Figure 5 shows the hierarchy and responsibilities tree in the NASP architecture with four distinct levels, outlining the decision-making structure of a network slice. The first level, the highest one, represented by the CSMF, serves as the access point for platform users and is tasked with organizing requests according to SLA/SLS, integrating new NSIs, and exchanging data with the subsequent layer. Information exchange between these two levels encompasses aspects such as orchestration, management, and data analysis from an end-to-end perspective of the network slice. The second layer, represented by the NSMF, provides a comprehensive view of network slices and understands the subdivision of each subdomain. Data collected from each domain is processed and analyzed end-to-end and vice-versa. The third layer is divided into three domains: RN NSSMF, CN NSSMF, and TN NSSMF, related to RAN controllers, NFVO, and WAN SDN, respectively. This layer focuses on the specific view of its domain, albeit unaware of intrinsic details, except for communication interfaces and connections with related domains. Information processed at layer three is ultimately detailed regarding physical and virtualized resources. Lastly, layer four consists of numerous small fragments, where each physical instance resource is considered a component of an end-to-end slice.

The tiers are related to the descriptive files of each stage, where Tier 1 consists of business-level requirements, such as SLA and SLS, containing latency limits, bandwidth, packet loss, Max UE, max PDU sessions, etc. Tier 2, after querying infrastructure availability for the tiers below, complements the descriptive information of Tier 2 with the information received from Tier 1, composing infrastructure intent, such as communication interfaces describing IPs and ports for AMF, gNB, and SDN route intentions. Tier 3, responsible for its domain, represents its elements and responsibilities at the level of physical and virtual resources, such as CPU, RAM, VLAN, geolocation, replicas, etc. Finally, Tier 4 has its definitions of MAC, volumes, storage, etc., isolating each Tier's responsibilities and maintaining the architecture's organization.

Figure 5 – Responsibilities Levels.

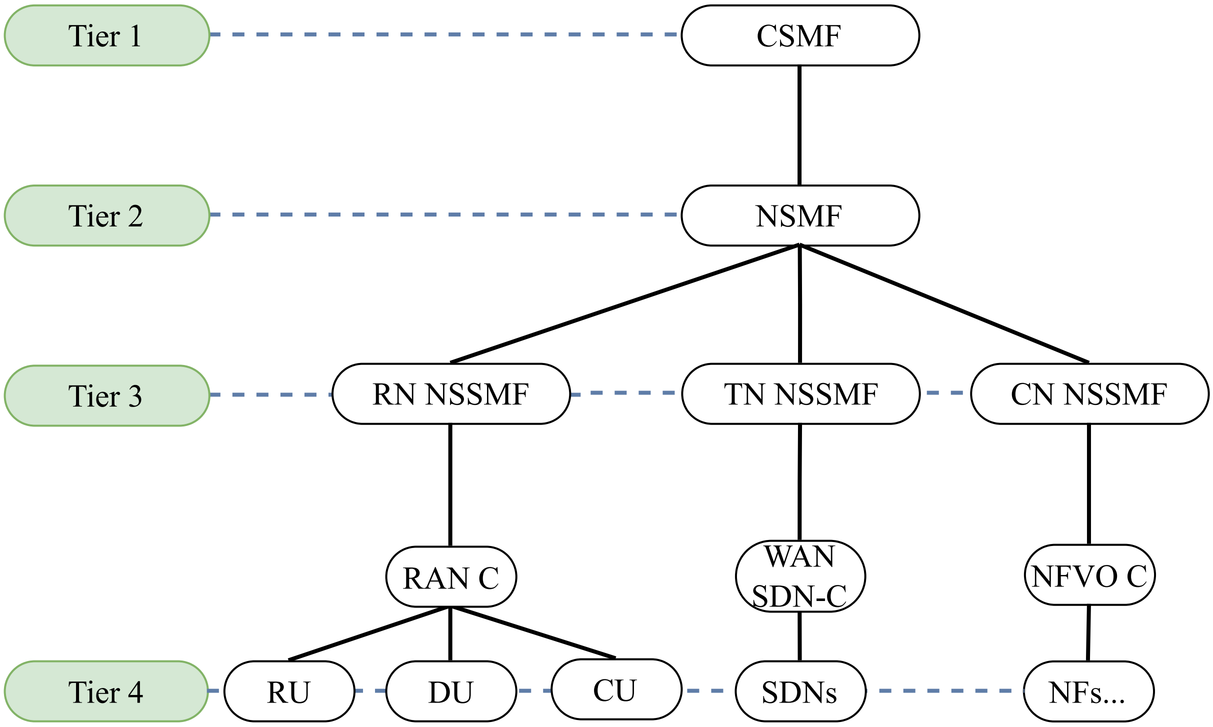
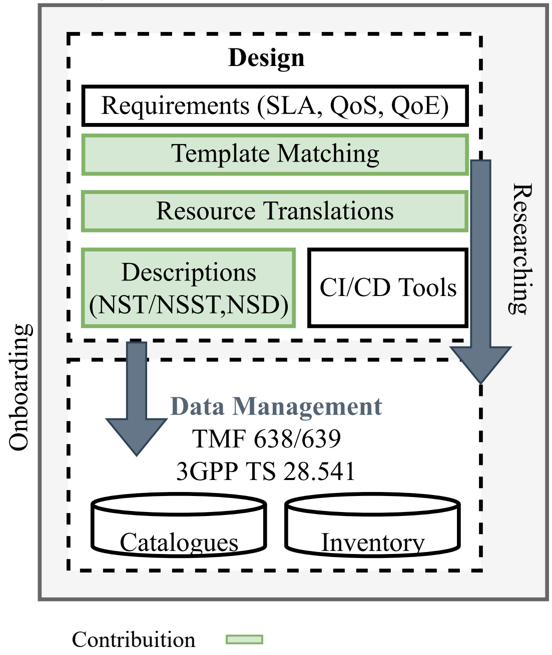


Figure 5 also shows the relationship between the NASP architecture, the responsibilities, and the decisions made by each layer. The NASP architecture integrates slice managers and slice instances once managers have no attributions to control and create virtual and physical resources, concluding a required extra layer to complete the E2E Network Slice platform. The hierarchy is necessary for designing an end-to-end slice where vertical and horizontal relationships are crucial. An end-to-end slice must be designed from a request received at Tier 1 and detailed and projected down to the description of the components at Tier 4. Additionally, it is horizontally related, describing all communication interfaces among domains. As a slice is altered at runtime, interfaces may or may not change.

4.2.1 Tenant Request and Slice Onboard

The network slice tenant requests onboard propose a resource translation over GSMA templates. GSMA has defined Network Slice Templates for multiple 5G use cases. The NASP architecture receives the required business scenario from the tenant and starts the onboard, as shown in Figure 6. Once the use case is specified, the proposed components, Template Matching, and Resource Translation process the input use case scenario from the tenant and research for NSSI templates in the Catalogues and Inventory database to build an NSI template as a group of NSSI templates already cataloged in the database. Finally, the onboard outputs a Network Slice Definition composed of virtual and physical descriptions for each domain.

Figure 6 – Business slice onboard.

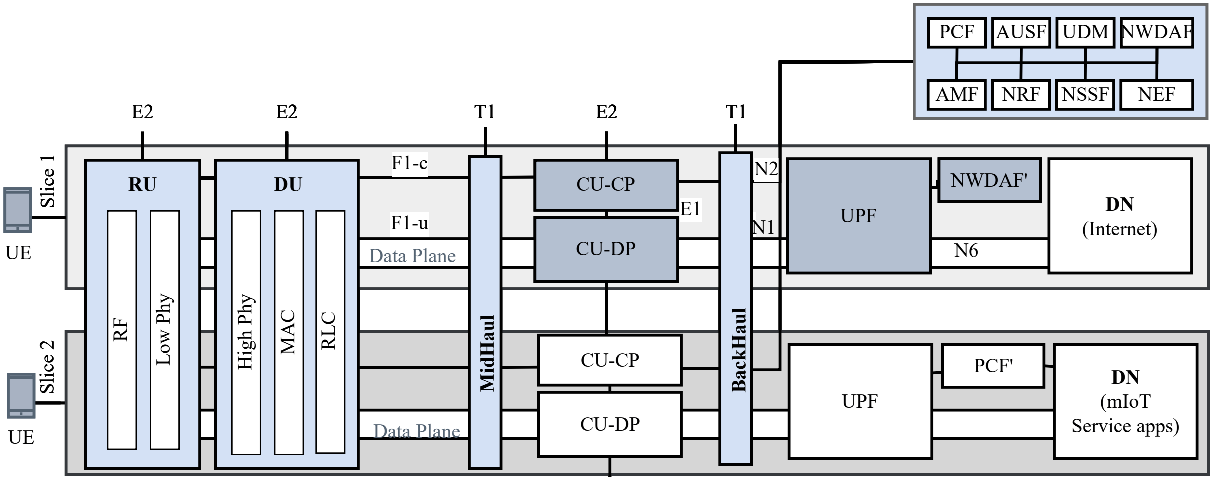


An onboard result is a Network Slice Template composed of Network Sub Slice Templates. An NST contains at least one NSST for each domain that describes domain infrastructure characteristics such as components, distribution, and domain characteristics such as VNFs interfaces, access ports, and endpoints. A Network Slice Descriptor maps slices technical attributes values, as described:

- GSMA Template Attributes.
- 3GPP Slice Attributes.
- 3GPP Network Attributes.

Figure 7 shows an example of the sub-slice instances in an E2E slice composed of shared and non-shared resources. The illustration shows a Slice 1 with a RAN as a shared disaggregated RU and CU between Slice 1 and Slice 2 and a non-shared CU for the control plane and data plane, a Transport Network as a shared MidHaul and BackHaul and a Core Network as non-shared UPF and Network Data Analytics Function (NWDAF) and a shared default Core Network Functions.

Figure 7 – Sub Slice Instances.



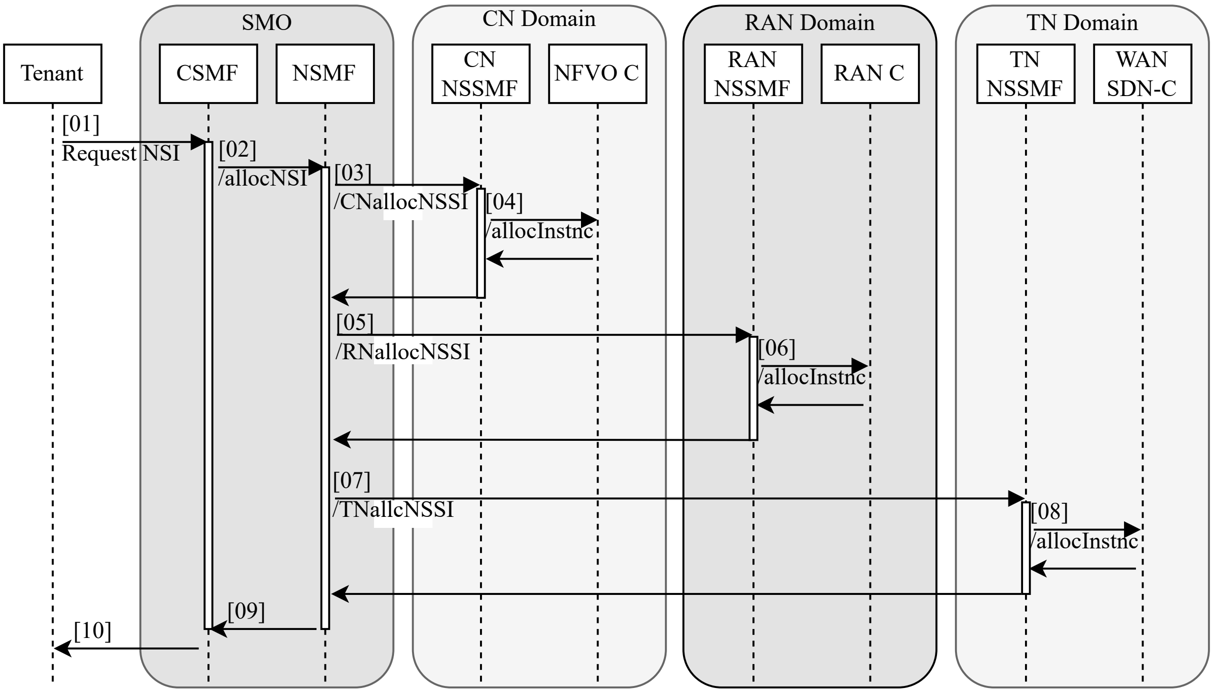
The process of onboarding network slice tenants in the NASP architecture involves the translation of resource requests using GSMA-defined templates and network slice designs. Upon receiving the tenant's business scenario, the architecture initiates the onboarding process, leveraging components such as Template Matching and Resource Translation. These components analyze the tenant's use case scenario and search for corresponding NSSI templates in the Catalogues and Inventory database. The resulting Network Slice Template comprises Network Sub-Slice Templates, each describing domain infrastructure characteristics and domain-specific attributes. These attributes are mapped from GSMA Template Attributes, 3GPP Slice Attributes, and 3GPP Network Attributes, i.e., UE density, a Maximum number of UEs and PDU Sessions, and NASP proposed attributes, such as Shared, N3GPP Support, and Exposed. An example of sub-slice instances within an end-to-end slice, including shared and non-shared resources across various domains where they describe all inter-domain interfaces to isolate and connect resources, illustrating the versatility and adaptability of the onboarding process in meeting diverse network slice requirements, where the next step is the resource allocation.

4.2.2 Resource Allocation Interface

3GPP has defined a resource allocation request interface from CSMF to NSSMF. The NASP architecture maps and defines the interfaces required to complete the resource allocation communication between subslices managers and domain controllers. The NSMF function manages the network slice resource allocation request from a global technical point of view, distributing the specific requests to each domain manager. NASP creates the relationships and interfaces required for an E2E allocation. Figure 8 shows the request allocation flow from Tenants to an

E2E slice allocation.

Figure 8 – Sequence diagram presents the steps proposed by the NASP architecture.



Based on architecture components, Figure 8 shows the diagram with the sequence of events proposed by the system. Initially, from the Tenant input, the use case mapping scenarios, the NST and NSST selections, NSD definition, and NSI request until complete E2E instantiating and tenant response. Serial requests are needed because of dependencies between different parts of the network, the core network first gives the interface info needed to connect with the RAN. Once the RAN gets this info, it sets up its interfaces and then provides the details needed to create links in the transport network. This step-by-step process ensures that each part of the network is correctly configured before moving on to the next, as described below:

01. The tenant requests a new mobile use case for Network Slice allocation and instantiation to NASP.

02. CSMF reads the use case scenario and translates it to Network Slice Definitions using the database catalog and inventory after requesting an NSI allocation to NSMF.

03. NSMF distributes NSI into CN NSSI and requests CN NSSI Allocation to the NFVO Controller.

04. NFVO Controller executes NSSI allocation and responds to instance allocation information.

05. NSMF distributes and defines slice attributes using the access end-points received from the NFVO Controller, requesting NSI to be added to RAN NSSI and RAN NSSI Allocation to RAN Controller.

06. RAN Controller executes NSSI allocation and responds to instance allocation information.

07. NSMF distributes NSI into TN NSSI and requests TN NSSI Allocation to the WAN SND Controller. item [08.] The WAN SND Controller executes NSSI allocation, connecting endpoints from RAN and CN and responding to instance allocation information.

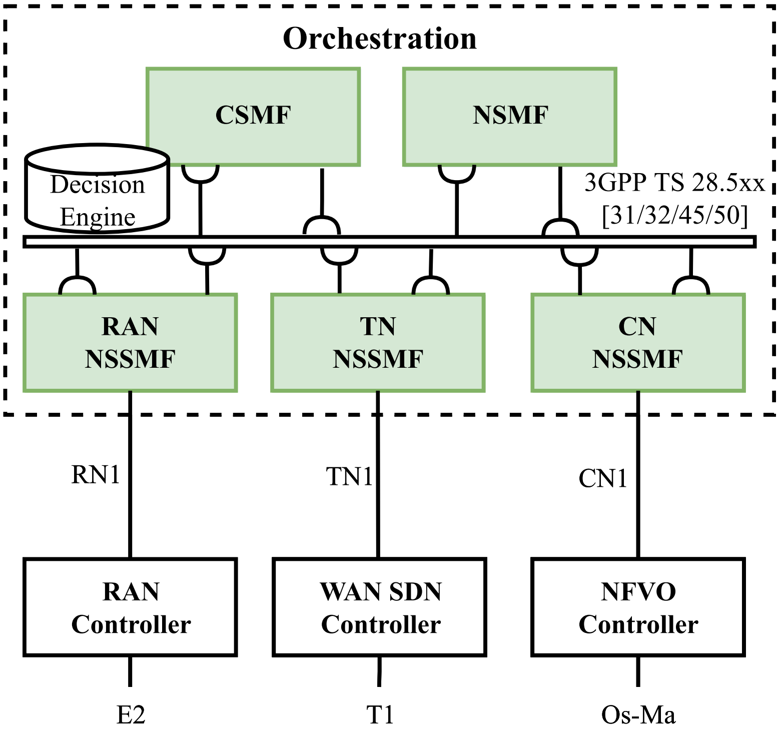
09. NSMF checks domain responses and stores instance information in the database, responding to CSMF allocation status.

10. CSMF responds to tenant allocation status.

The order presented in Figure 9 follows the sequence of information dependencies for the deployment project of an NSI. After the request and transformation of a request passing through the CSMF and NSMF, the instance allocation interfaces are requested. The first domain to be requested must be the Core domain, as it contains the access information of the AMF of the slice in question. Once the communication information, such as IP and port, of the AMF is made available, it is possible to proceed with the configuration request of the RAN with the respective RU, CU, and DU, as these components require the connection endpoint with the Core. After both interfaces are configured, it is possible to establish the connection from the transport network, redirecting outgoing packets from the RAN to the Core's input and vice versa, passing through the route intention requested by the controller of the transport network domain.

Figure 9 shows the interfaces between Domain Network Slice Managers and Domain Infrastructure Controllers where E2 interfaces the RAN Controller domain, T1 interfaces the Transport Controller domain, and Os-Ma interfaces to the Core Controller domain. Moreover, NSMF receives the request from CSMF, completes slice attributes required for the NSI instance, and sends the information to the domain's NSSMF with a data structure. The described structure approaches network slice allocation and instantiation within the NASP architecture. Beginning with the tenant's request and culminating in the instantiation of an end-to-end slice, each step is orchestrated to ensure integration across domains. Through the coordination of components such as CSMF, NSMF, NFVO Controller, RAN Controller, and WAN SND Controller, the architecture manages resource allocation and configuration, enabling the functionalities of an expected network slice, where the next Quality Assurance is the next step.

Figure 9 – Components Interface Diagram.



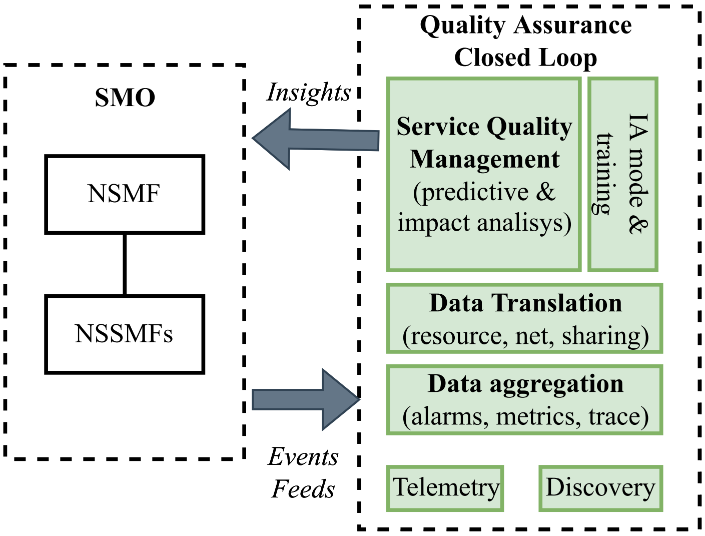
4.2.3 Quality Assurance Closed Loop

The Quality Assurance Closed Loops are the principal function of the run-time slice step in the lifecycle. The component is responsible for slice monitoring, tracing, data analyses, decision-making, and updating action requests. Figure 10 shows the architecture with the related components. Each layer should have its Quality Assurance responsibilities based on the layer scope. NSMF, as global slice management, is responsible for global decision-making over slice quality. Decisions include future long-term prediction and adaptability, non-real-time slice management, and slice physical and virtual distribution. However, NSSMFs are responsible for domain slice qualities. The proposed component, Quality Assurance Closed Loop, presented in Figure 10, has six elements: (i) Telemetry and discovery; (ii) Data aggregation; (iii) Data translation; (iv) Service Quality Management; (v) AI prediction and impact analyses. The components are explained below from bottom to top.

(*i*) **Telemetry and discovery** - It acts as consuming and discovering data exposure from network domains. Each subslice component should have a metrics exposure access point to be consumed by the respective NSSMF. Most analyses come from Core Network processed by NWDAF and exposed by NEF.

(*ii*) **Data aggregation** - It aims to relate the data collected from the components below, tracing and monitoring the required information for the specific layer.

Figure 10 – Quality Assurance component.


SMO
NSMF
NSSMFs
Insights
Events Feeds
Quality Assurance Closed Loop
Service Quality Management (predictive & impact analisys)
IA mode & training
Data Translation (resource, net, sharing)
Data aggregation (alarms, metrics, trace)
Telemetry
Discovery

(*iii*) **Data Translation** - The business requirements must be translated into infrasctuture requirements. The translation occurs in the opposite direction in these components since the collected data are originally from infrastructure resources.

(*iv*) **Telemetry and discovery** - It acts as consuming and discovering data exposure from network domains. Each subslice component should have a metrics exposure access point to be consumed by respective NSSMF.

(*v*) **AI model & training** - As described by 3GPP, we have an NWDAF responsible for data analytics that exposes useful information for each slice. The AI model component from Quality Assurance has the same purpose but focuses on a global overview, where it could use inter-domain and inter-slice data to process.

(*vi*) **Service Quality and Management** - centralizes and determines the actions of the architecture, being responsible for making decisions and consequently defining the modifications and amount of the virtualized functions necessary to supply the network demand in non-RT requirements. Among the attributions are receiving the input of the network topology, evaluating the initial resource distribution over subslices, analyzing the metrics and restrictions imposed by the templates, and exposing the data in dashboards for a User Interface view to comprehend the actual state of NSI easily.

The components of the NASP architecture are responsible for the network slice run-time step contribution to pursue the requirements requested by tenants at the slice onboard step. We can understand each component of NASP architecture, responsibility, and interface related to building the platform for E2E Network Slice. This section presented the NASP architecture in this context, discussing the main design decisions and detailing the blocks with their responsibilities and internal and external interfaces.

The proposed decisions have taken advantage of gaps and overlaps in the definitions of international entities to present an architecture proposal for the E2E platform. Board decisions were evaluated considering simplified ways to translate slice requests, facing the complexity of implementing the proposal. Hierarchy definitions were crucial for the E2E mapping, vertical and horizontal, as seen in Figure 7, and the complexity of the relationship among all NSSIs. After defining the NSSI, the output interfaces were added for communication between the platform and the controllers of the physical instances. Finally, the relationship between CI/CD and IaC strategies with the instantiation flow designed for the platform. The AI techniques used for quality control have yet to be defined. However, it is the next step in the research. In the next chapter, the evaluation methodology is presented, containing the analyzed metrics, the prototype developed, and the case studies planned for implementation, discussing all the necessary steps for developing the work.

5 NASP PROTOTYPE

This chapter presents the prototype developed based on the proposed Network Slice as a Service Platform on New Generation Mobile Networks. Section 5.1 guides the Technology Tools chosen to develop and explains why those technologies were selected. Next, Section 5.2 describes the network configuration for NFs, domain, and interdomain communications. Finally, 5.3 discusses the platform flow, describing NSST, NST, and NSI flows from design to deployment and RT configuration.

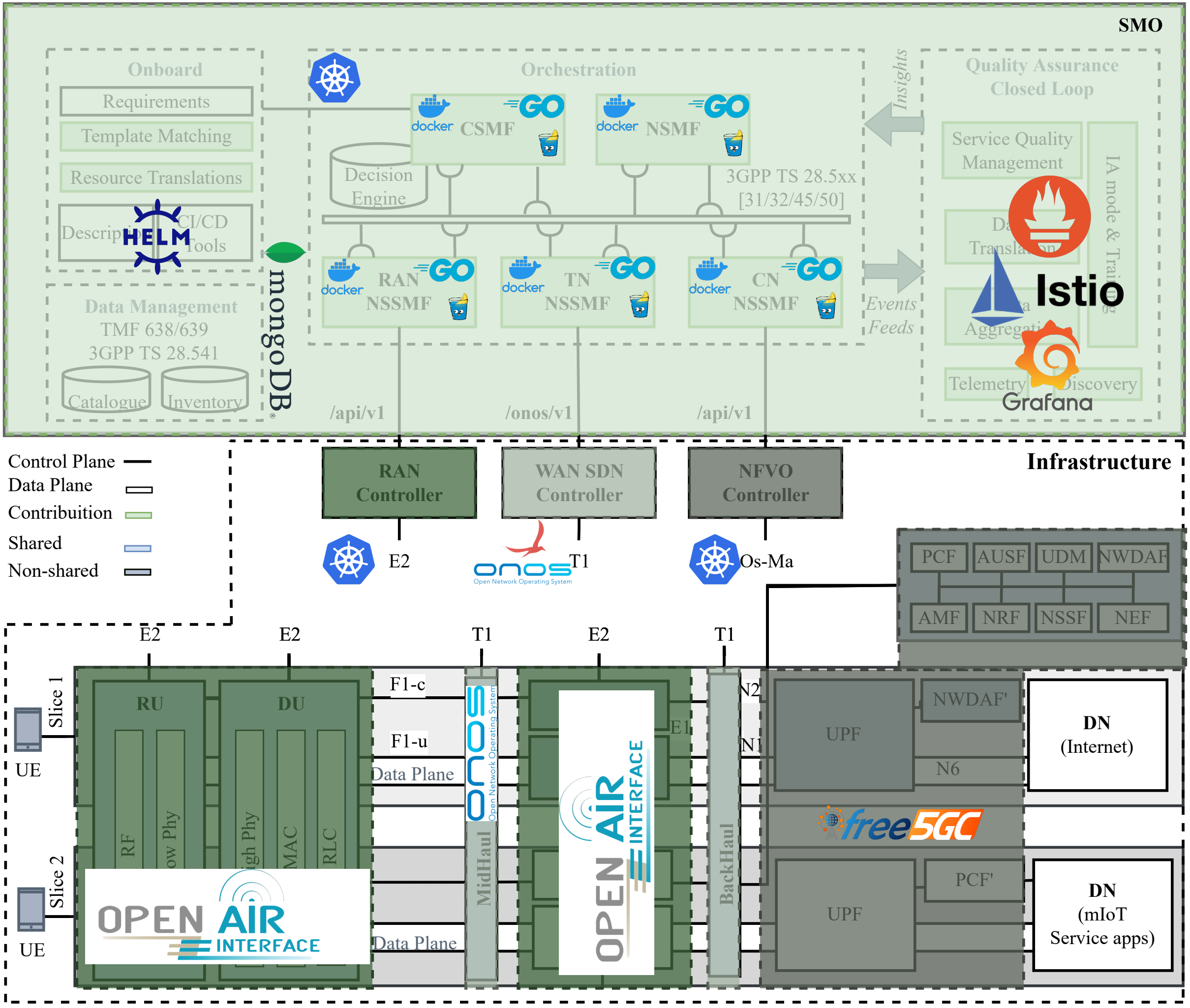
5.1 Technology Tools

The prototype aims to create an experimental environment according to the NASP architecture, focusing on developing the dynamic-aware orchestrator to position virtualized radio functions. In this sense, a Kubernetes platform was used to develop the solution aligned with the SMO block of the O-RAN architecture. Kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services, bringing configuration and automation gains. In addition to having a vast ecosystem of fast-growing (AUTHORS, 2023).

The NASP prototype uses Open Source tools to control different domains and manage the infrastructure. The work uses Docker and Container Runtime Interface (CRI) as the container run-time for virtualization and containerization, and for container orchestration, we are choosing Kubernetes running over Ubuntu 20.04 machines. Moreover, we define to work with Open Network Operating System (ONOS) for SDN infrastructure controlling a Mininet virtual environment. We also use OpenAir interface Radio Access modules for 3GPP use cases for RAN control, and the Core network runs over the Free5gc project.

Figure 11 shows SMO functions are containerized NVFs running over a Kubernetes container orchestrator exposing Hypertext Transfer Protocol (HTTP) services over Ingress Controller and Load Balancer provided by Kubernetes. Core NSSMF function interface with Core controller will use Kube-API-server API provided by Kubernetes over HTTP Representational State Transfer (REST) API. Kubernetes is a complete REST Service-based Architecture, so once authentication keys are changed, all infrastructure management is available in an M2M communication. Transport NSSMF function interface with WAN SDN Controller is projected to use ONOS REST API provided by the ONOS project controlling a Mininet virtualized infrastructure. The API contains manipulation interfaces for devices, network topology, hosts, groups, components, and applications, where once the infrastructure is deployed, NASP throw ONOS can request intents depending on Slice requirements. Finally, RAN NSSMF interfaces with Kube-API-server and My5G-RANTester. Where, my5G-RANTester is a tool designed to emulate the control and data planes of User Equipment (UE) and gNodeB (gNB), the 5G base station. It aims to implement the NGAP and NAS protocols as defined by 3GPP Release 15 (R15) and beyond. With my5G-RANTester, users can study various functionalities of a 5G core network, ensuring adherence to 3GPP standards. A feature of my5G-RANTester is its scalability, allowing it to mimic the behavior of numerous UEs accessing a 5G core network. Templates, instance descriptions, and Infrastructure codes are designed to be stored as documents in a document-oriented database, once NST and NSST are JSON-formatted documents, in this case, MongoDB. Infrastructure codes are using Helm manifests as a majority. Sometimes, the use of Helm cannot be stored as a manifest. In that case, raw YAMLs were projected to be used as infrastructure descriptors. Tiers distinguish templates and Instance descriptions as tags on document descriptions. For monitoring, alerting, and observability, the project was based on Prometheus as a metric collector, Istio as traffic tracing, and Grafana to expose all the data and configure alerting webhooks.

Figure 11 – Proposed Prototype.



The functions were developed in Python on version 3.9 (latest document writing date) with Flask as an HTTP server framework to provide HTTP request handlers embedded in the application, containing Docker images stored in Docker-hub as an image repository. As an application front-end framework, CSMF is defined as using Django with a single-page application concept to deliver the fastest user-end application response time. Moreover, we use Bash on version 5.9 to control the Linux infrastructure under Kubernetes, manage the communication among domains, and manipulate Linux packages, networks, and firewall tools. After developing and integrating the modules with the tools, it is possible to carry out E2E communication and request new network slices. The M2M communication flow was described in Chapter 4 and identified the tools APIs as described in this subsection. However, the prototype requires an infrastructure to run and orchestrate, as described in the next section.

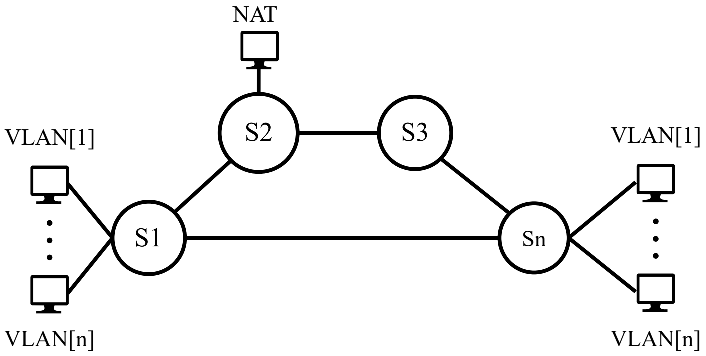
5.2 Network Configurations

In the 5G network architecture context, operating E2E network slices requires addressing several challenges. One of these challenges relates to the relationship among the different domains involved in creating, designing, and managing these slices, ensuring integration throughout the entire process as described in Chapter 3. In addition, the simultaneous exposure of multiple slices represents another demand. Another crucial aspect to address is the redirection of packets in an isolated manner by slices, ensuring segregation and appropriate routing of network traffic among different slices without compromising the QoS or the security of the network as a whole.

For the connection among the domains of a slice, it is necessary to redirect packets among the NSSIs. Considering that each POD within Kubernetes runs a Container, and each Container is a combination of Linux Namespaces, Linux CGroups, and IP Tables rules, the construction of the transport network was chosen to be based on packet routing using VLANs, where each POD has its IP Tables rules rewritten, redirecting external traffic to the IP address received from the transport network controller. Figure 12 presents the topology used where S1, S2, S3, and Sn are the virtualized Switches by the Mininet network representing a real topology with network traffic data collected and described in Chapter 6. VLAN[1] connected to S1 represents the first available VLAN connected to a physical port of the Switch, ranging from 1 to n, which receives an IP, and all received traffic is forwarded to the specific port. After completing the route and arriving at Sn, it is sent to VLAN[N] and redirected to the original address. The routes are divided between Long Path and Short Path, with their final lower latency, with S1, S2, S3, and Sn being the long route and S1, Sn being the short route. NAT represents the NAT Server applied to the topology for translating IPs allocated to the VLANs. For example, the traffic from a POD is sent to VLAN[1], known by ONOS and Mininet, where the route was previously defined by the ONOS controller, and upon reaching the final internal address, it is redirected to the new original POD.

Network slices, which are vital for the efficient management of resources, have been implemented in Kubernetes. However, as real devices do not operate within this platform, there arises a need to manage these slices in a way that exposes their access points beyond the cluster's internal network while complying with the standards set by 3GPP. This deployment presents a

Figure 12 – Transport Network Topology.



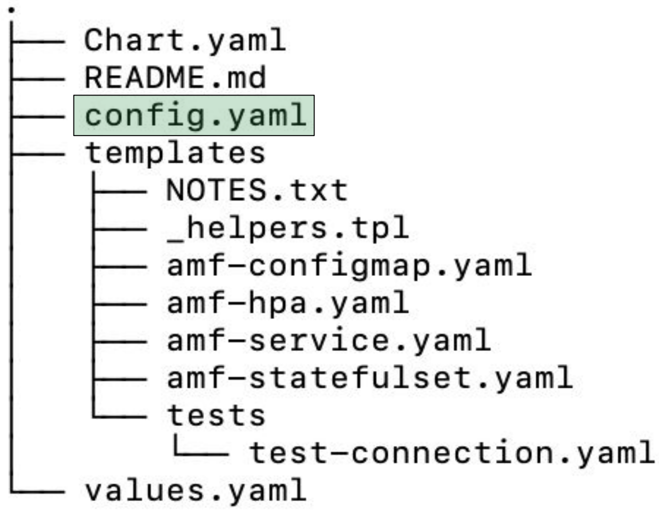
challenge due to the common use of Kubernetes and its load balancing for microservices, which operate at the application level. To overcome this problem, a slice allocation configuration was developed, where each new request directs public IPs to the internal network configurations of the cluster. This deployment allows load balancing of the resources instantiated in the cluster while maintaining essential definitions such as network slice admission access points and port sharing, such as 38412 as stipulated by 3GPP, for the admission of new UEs in the same physical instance, shared among different network slices. This approach ensures the efficient operability of network slices aligned with the industry standard requirements. However, since the architecture includes a user interface for the platform, it is necessary to define the user flow as described below.

5.3 Platform Flow

This section presents deeper into and illustrates, with examples, the requests and processes discussed in the previous chapter. It focuses in detail on the practical implementation of the following steps: the creation of an NSST, the development of an NST, the execution of a request for the instantiation of an NSI, as well as communication among different domains and the identification and resolution of existing gaps in network controllers for the implementation of the NASP. Finally, the section concludes by examining closed loops, which are essential for the continuous automation and optimization of the platform.

Each domain has its specific way of creating NSSTs, with RAN and Core using a pre-defined structure of Helm Charts and TN adopting a unique descriptive structure made of lists. Figure 13 shows the file structure organization for RAN and Core domains, including standard Helm files such as *Chart.yaml*, *README.md*, and *NOTES.txt*. Inside the templates folder, some files detail the proposed structure for the NSST, encompassing Kubernetes network configurations, high availability, and configurations for PODs and containers. The *values.yaml* file contains internal Kubernetes configurations that can be customized as needed. Meanwhile, the *config.yaml* file holds specific configuration information for the NF, including dnnList, snssaiList, plmnId, mcc, mnc, and mcsi, among other essential configuration files. These and other settings are mapped and adjustable by the NASP at the time of installation, allowing the NSSTs to be tailored to the specific needs of each deployment. This process facilitates the establishment of isolated resources, such as NFs and transport routes, ensuring efficiency and efficacy in managing network resources.

Figure 13 – NSST directory tree.



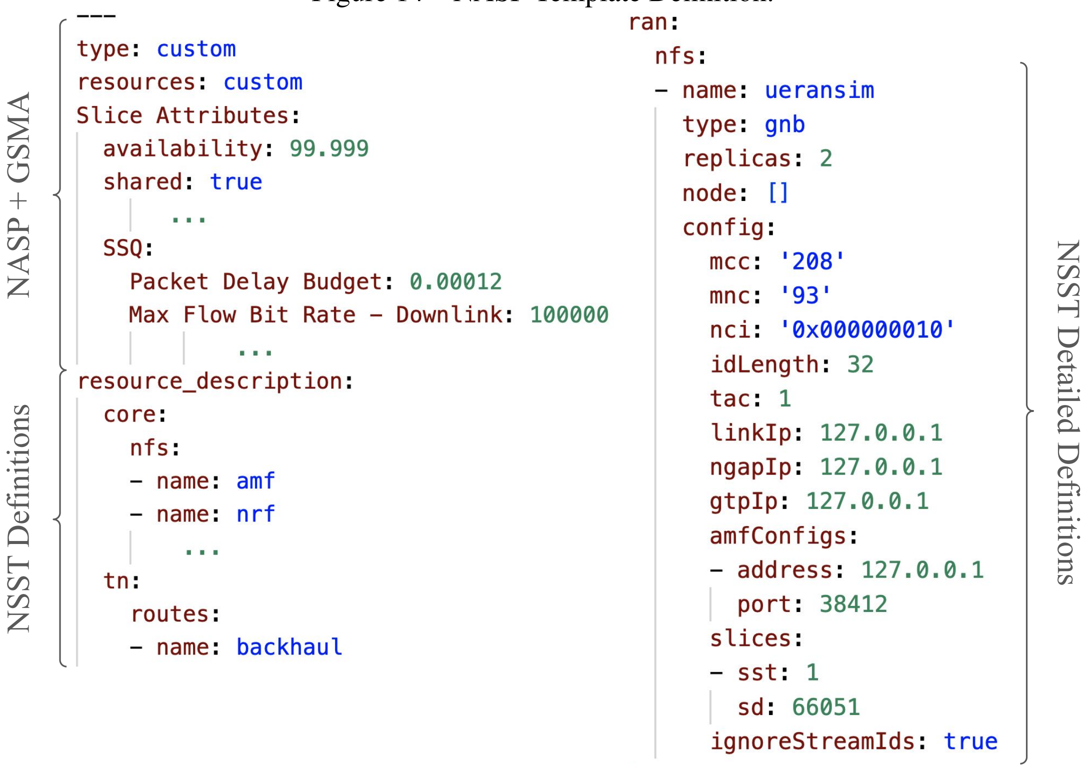
```
.
├── Chart.yaml
├── README.md
├── config.yaml
├── templates
│   ├── NOTES.txt
│   ├── _helpers.tpl
│   ├── amf-configmap.yaml
│   ├── amf-hpa.yaml
│   ├── amf-service.yaml
│   ├── amf-statefulset.yaml
│   └── tests
│       └── test-connection.yaml
└── values.yaml
```

After establishing the NSSTs, the next step is the creation of an NST. This creation begins with a request from the CSMF, in which the user specifies the desired characteristics for their network slice based on the templates provided by the GSMA. With the request in hand, the NASP analyzes the provided requirements. Utilizing direct functions, such as defining the maximum number of UEs, latency, UE density, and whether the slice will be shared or not, the NASP structures the NST. This process involves selecting the appropriate NSSTs that meet the established requirements.

During the formulation of the NST, the NASP also defines major aspects such as the involved NFs, the locations for hosting the resources (edge or central), the necessary interfaces for communication between different network components, and the specific characteristics of the transport route that best align with the slice's requirements. The NST in NASP is designed to support GSMA standard definitions such as mIoT, but it is also engineered to accommodate specific definitions, such as descriptions of the resources needed within each slice, ranging from NFs to the location of each NF, as presented in Figure 14. Therefore, the definition of the NST becomes a detailed process prepared for the request of design and deployment of an NSI.

The deployment of an NSI follows the same request format presented in Figure 14. Highlighted in NASP+GSMA are the essential standard definitions described by the GSMA entity, with the addition of necessary attributes for the request using the NASP platform. The NSST Definitions element offers the possibility for more detailed requests by presenting the NSST composition of the slice, where, separated by domain, the user can select each specific NF.

Figure 14 – NASP Template Definition.



```yaml
---
type: custom
resources: custom
Slice Attributes:
  availability: 99.999
  shared: true
    ...
  SSQ:
    Packet Delay Budget: 0.00012
    Max Flow Bit Rate - Downlink: 100000
        ...
resource_description:
  core:
    nfs:
    - name: amf
    - name: nrf
      ...
  tn:
    routes:
    - name: backhaul
```

```yaml
ran:
  nfs:
  - name: ueransim
    type: gnb
    replicas: 2
    node: []
    config:
      mcc: '208'
      mnc: '93'
      nci: '0x000000010'
      idLength: 32
      tac: 1
      linkIp: 127.0.0.1
      ngapIp: 127.0.0.1
      gtpIp: 127.0.0.1
      amfConfigs:
      - address: 127.0.0.1
        port: 38412
      slices:
      - sst: 1
        sd: 66051
      ignoreStreamIds: true
```

NSST Detailed Definitions add the possibility of detailed definition, bringing the internal configurations of each NSST, such as interfaces, identifiers, and general settings, as well as intentions of point-to-point routes for the TN case.

To integrate the TN domain, creating an intermediary software that acts between the NASP and ONOS was necessary. The challenge encountered was that ONOS, despite offering functionalities for network configuration, does not provide an option that meets the specific requirements of this project in terms of detailing and customization. The APIs provided by ONOS are divided into two categories: one quite abstract, which simplifies the creation of routes from point A to point B without allowing specifications such as latency or bandwidth; and another extremely detailed, requiring manual configuration and in-depth network knowledge, connecting equipment piece by piece until the complete route is formed.

Given this limitation, an intermediary API was developed to interpret the specific demands of a 5G network and determine the best possible route within the available options. Subsequently, this API communicates with ONOS, detailing all the necessary point-to-point connections to establish the transport route for the requested network slice. Therefore, the middleware ensures that the NASP platform can manage the transport network with the required level of detail without the need for complex technical knowledge about the transport network structure directly on the platform.

Figure 15-a depicts the complete cycle of creating an NST and requesting an NSI to provide a more precise representation of the aforementioned slice federation phases. The NASP

NST definition flowchart begins with an operator's request, where it is then checked if the sent template was a custom template, with the selection of NSSTs and their computational resources previously established. If so, the NASP interprets the file, checks the availability of the requested NSSTs, and declares the new NST following the initial request information. Moreover, the tool analyzes each variable, checking whether functions will be shared and exposed and whether there will be support for Non-3GPP, latency requirements, availability, and PDU sessions. For each variable, the tool performs predefined actions, such as the selection of N3IWF NSSTs, or the allocation of public IPs for communication external to the cluster, as well as the allocation of computational resources for functions such as UPF and the configuration of surplus UPFs for high network availability and redundancy. After the combination of NSSTs and the definition of computational resources for each NSST, the creation of the NST with the set of NSSTs is finalized.

Figure 15 – NASP NST/NSI Flowchart.

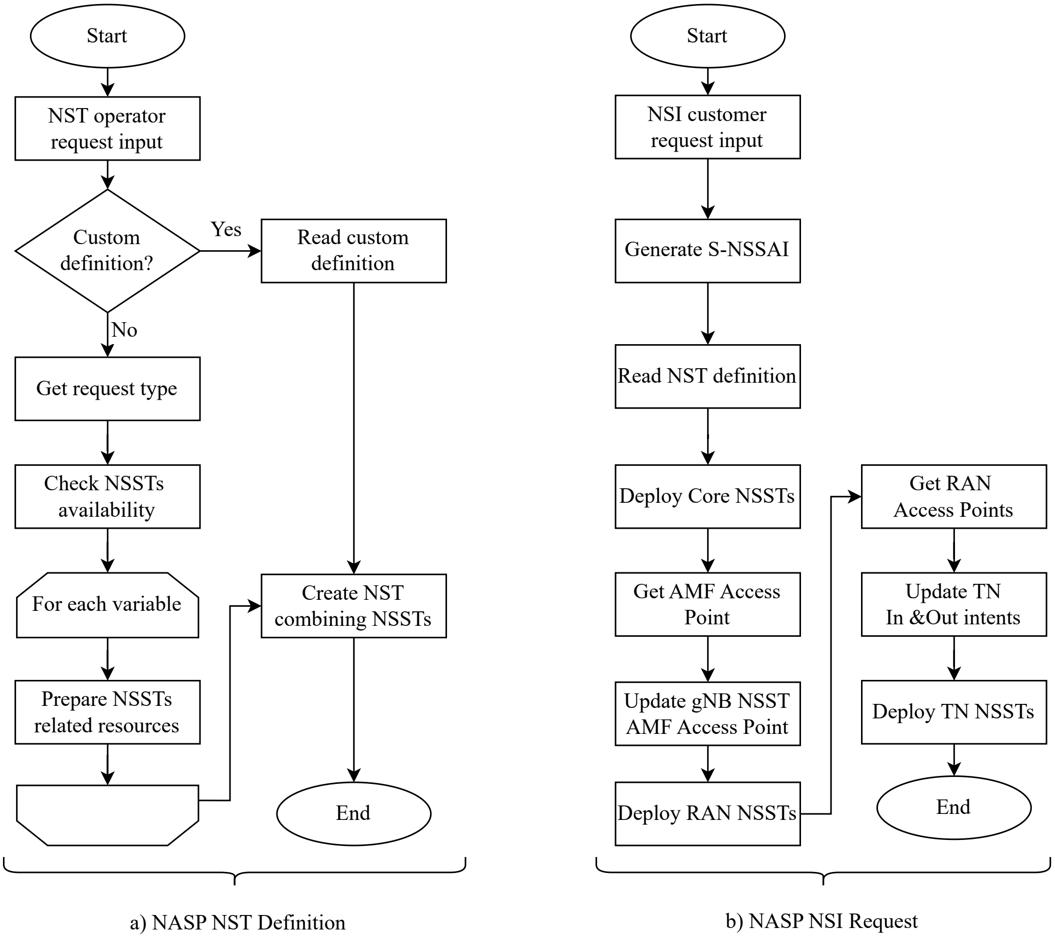


Figure 15-b also presents the request and creation of an NSI. The flow begins with the NSI request from the tenant, where the tool generates an S-NSSAI based on the NST template and reads the NST definitions to deploy the Core domain, where the return of this deployment will be the access points for connection with the gNB, such as AMF IP. Subsequently, the platform updates the gNB definitions to carry out the deployment connecting to the declared AMF. Finally, by interconnecting the communication interfaces between the RAN and Core domains, the TN domain carries out the requests for route intentions, mapping the declared interfaces of each domain.

This chapter detailed the essential definitions and implementations for the prototype's development, covering the template structures and required flows for designing and deploying an E2E network slice. With this foundation in place, the next chapter focuses on describing the evaluation methodology and the environments used and presenting the selected use cases to test the proposed architecture and tool.

6 EVALUATION METHODOLOGY

The methodology used in this work for evaluating the performance of experiments follows the concepts of Jain et al. (JAIN, 1990). It aims to obtain greater statistical accuracy for the responses collected at a lower cost. Therefore, it is understandable that the methodology planning provides independent variables (factors). In this way, it is feasible to define the most probable values these variables can assume (levels). In this case, realizing the effect manipulation causes on the response variable (dependent variable). This chapter is composed of three sections. First, the computational infrastructure is presented in Section 6.1. Next, evaluation metrics are detailed in Section 6.2 and, finally, Section 6.3 describes the scenarios and study cases.

6.1 Infrastructure

The infrastructure used in this study plays a vital role in data collection, processing, and analysis. It was designed to support the specific technical requirements of the use cases explained in Section 6.3, enabling replicability and the simulation of a real distributed topology. For this work, three main configurations were considered for constructing the environment: machines, cluster configurations, and network topology.

The study used a set of VMs allocated in the public cloud service, Digital Ocean, using general-purpose computational resources to meet the different applications used for prototype validation. The tools utilized included NASP, Kubernetes cluster, Mininet, and ONOS. The machines were two VMs with Intel(R) Xeon(R) CPU E5-2650 v3 @ 2.30GHz with 4GB RAM ECC. Moreover, we used three VMs with Intel(R) Xeon(R) CPU E5-2650 v2 @ 2.6GHz with 8GB RAM ECC. Additionally, two VMs were dedicated to running ONAP and Mininet with ONOS, and three VMs were dedicated to running the Kubernetes Cluster, which varies according to the required load. The rules for scaling up and down were established at 80% and 20%, respectively. All machines were allocated in New York - USA, using Ubuntu 22.04 as the operating system with Kernel version 5.15.0-92-generic. The Kubernetes Cluster runs on version v1.28.3 with tools Calico version v3.26.3 as the standard CNI, CoreDNS version v1.10.1, and Multus version v4.0.2 as the secondary CNI responsible for auxiliary interfaces in containers deployed for Core, RAN, and N3IWF. The environment also utilizes the Helm tool version v3.9.3 for deploying templates.

For our numerical evaluation, we considered a network architecture consisting of five virtual machines operating in a cloud-native infrastructure, each serving as a Kubernetes Node. To assess network performance under various latency and cost conditions, we used three types of cloud sites: Regional (or Central), Metropolitan (or Edge), and Internal (or Extreme Low Latency Edge). The Central cloud corre- sponds to larger data centers with ample resources, while the Metropolitan cloud represents smaller data centers situated closer to end-users, facilitating low latency. The Internal cloud, depicted in our study by specialized and compact facilities at the edge of mobile networks, is a data center within the user operator's network, ensuring the lowest achievable latency

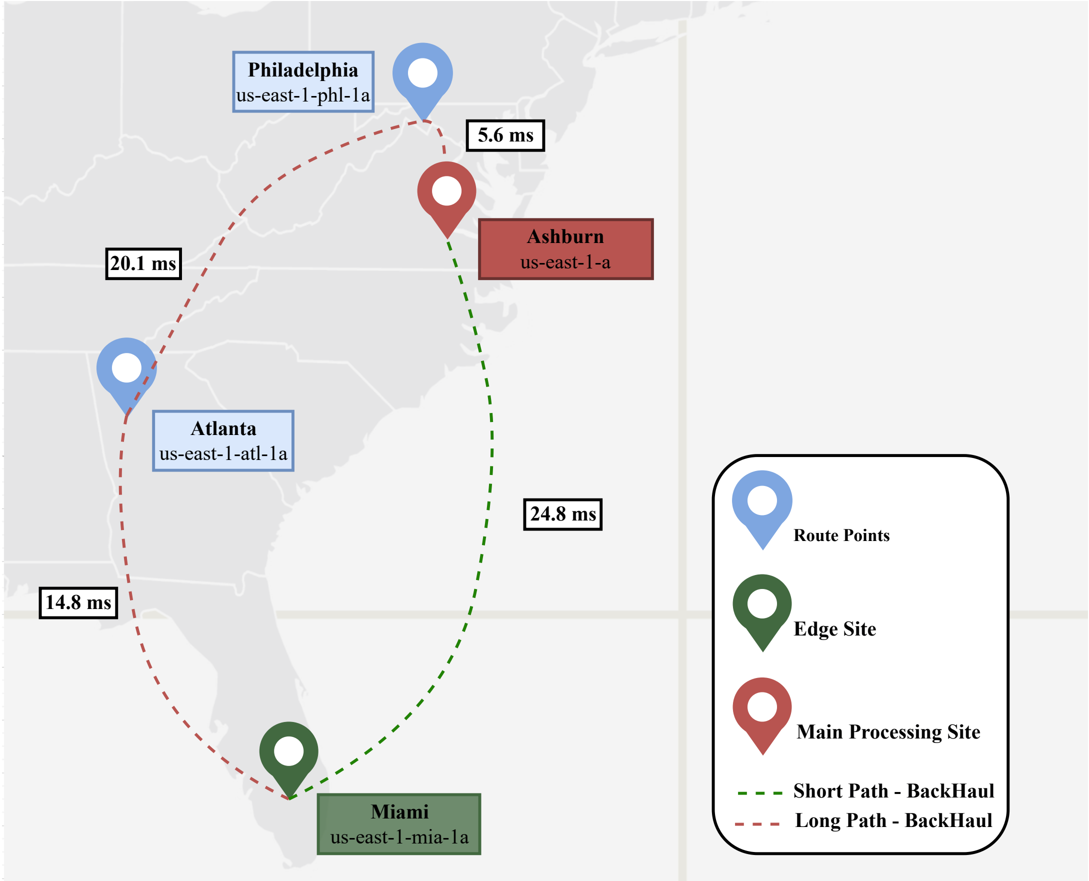
For our numerical evaluation, we considered a network architecture consisting of a cloud-native infrastructure. To assess network performance under various latency and cost conditions, we used three types of cloud sites: Regional (or Central), Metropolitan (or Edge), and Internal (or Extreme Low Latency Edge). The Central cloud corresponds to larger data centers with ample resources, while the Metropolitan cloud represents smaller data centers situated closer to end-users, facilitating low latency. The Internal cloud, the edge of mobile networks, is a data center within the user operator's network, ensuring the lowest achievable latency. A cost analysis of the VM instances was conducted on the Internal site, focusing on three specific types of instances. It was observed that the Internal site offers a limited range of instances compared to the more extensive options available on the Central and Metropolitan sites. Table 4 details the attributes and costs of each cloud instance. The analysis prioritized the most cost-effective options for the Internal site. Instances from the Metropolitan and Central sites, comparable to Internal options, were selected to ensure comparison consistency. The selection of cloud computing resources involves a delicate balance between performance optimization and cost efficiency, reflected in the adopted approach. Each cloud instance, classified by Level, offers different amounts of vCPU, RAM, and Storage. A careful evaluation of these characteristics of application requirements allows for the optimization of resource allocation, reduction of operational costs, and overall efficiency enhancement of computational workloads.

Table 4 – Description of the Cloud Instances.

Type	Size	vCPU	RAM (GB)	Storage (GB)	Price/Month
Internal	medium	2	4	200	$70,88
Internal	xlarge	4	16	200	$193,52
Internal	2xlarge	8	64	200	$526,40
Metropolitan	medium	2	4	200	$67,96
Metropolitan	xlarge	4	16	200	$117,60
Metropolitan	2xlarge	8	64	200	$181,84
Central	medium	2	4	200	$46,37
Central	xlarge	4	16	200	$76,74
Central	2xlarge	8	64	200	$137,47

For the TN domain, Mininet, version 2.3.0, was used to emulate the real topology under study, and the ONOS tool, version 3.0.0, was for network control. ONOS runs encapsulated in a Container controlled by Docker. An analysis of the long-distance network topology was conducted to assess latency in cloud networks by deploying VMs in three metropolitan regions. Each internal site is linked to a main Central site. This Central location is in a designated region with a wide range of Metropolitan and Internal sites. Network delay samples were collected from Metropolitan regions containing both types of sites. The farthest Metropolitan area from the Central location and the closest were chosen, as illustrated in Figure 16. Latency data were acquired from a variety of reliable sources (GET STARTED WITH AWS WAVELENGTH - AWS WAVELENGTH — DOCS.AWS.AMAZON.COM, 2024).

Figure 16 – Long Range Cloud Network Topology.



Latency measurements were collected at a frequency of one value per second for 72 hours, starting at 10:00 AM on February 3rd, 2024, and ending at 10:00 AM on February 6th, 2024. All data were aggregated into a single 24-hour interval to identify daily patterns and estimate the average trend of a typical day, disregarding specific day information and focusing only on the hour. Subsequently, a cubic non-linear regression was applied to fit these regression models. A synthetic dataset was successfully generated using this model, presenting average values that represent a 24-hour cycle for each link. Furthermore, Table 5 presents the latency information numerically.

Table 5 – Latency per link for cloud sites.

Source	Destination	Avarage Latency (ms)	Standard Deviation (ms)
us-east-1-mia-1a	us-east-1-a	24.8	4
us-east-1-mia-1a	us-east-1-atl-1a	14.8	3
us-east-1-atl-1a	us-east-1-phl-1a	20.1	3
us-east-1-phl-1a	us-east-1-a	5.6	1
gNB	us-east-1-mia-1a	2	<1

Given the established computational resources and the network topology replicated in an emulated manner, it is possible to proceed to the work evaluation metrics. The next section presents five metrics that evaluate the three study goals described in Chapter 1.

6.2 Evaluation Metrics

Various critical aspects must be analyzed to comprehensively evaluate a network automation platform such as NASP, each with its specific evaluation methodology. These aspects include (i) the instantiation time of an E2E network slice, (ii) scalability, (iii) flexibility and customization of a slice request, (iv) cost efficiency, and (v) UE connection latency.

The first aspect, the implementation time of an E2E network slice, is crucial for understanding the agility of the platform, which covers the items (i), (iii), and (iv), described as goals in Chapter 4. This total time can be broken down into three main components: the design time, which covers the planning and design of the slice; the configuration time, required to adjust the infrastructure and resources; and the activation time, which is the period until the slice is operational and available for use.

Regarding scalability, the platform's ability to support growth in the number of network slices or users without performance degradation is fundamental. The growth rate supported by the platform is calculated by considering the increase in the number of users or slices over time, providing a quantitative measure of scalability. A high supported growth rate indicates a robust platform capable of efficiently adapting to increasing demands.

Flexibility and customization are desirable qualities in any network automation platform, reflecting its ability to adapt to different needs and requirements, covering items (ii) and (v) described in Chapter 4. Evaluating these characteristics involves counting the configurations or customizations available and measuring the time required to implement changes. A highly flexible platform allows for a wide range of quick adjustments, facilitating the customization of services as needed.

In evaluating network slices' cost efficiency, attention turns to the ratio between the performance achieved and the financial investment employed. This analysis focuses on the performance of the network slices in terms of their data transmission and processing capabilities, in addition to the operational cost associated with maintaining these specific slices. High-cost efficiency indicates that the network slice is optimizing its performance concerning the capital invested, a fundamental aspect of ensuring effective budget and financial resources management.

Lastly, the platform's performance impact is assessed through UE connection latency. This evaluation considers not only the direct latency experienced by users during data transmission but also the efficiency of the network infrastructure in managing and routing data packets. Platforms that demonstrate lower UE connection latency through more efficient network management and optimized data routing contribute positively to user experience, aligning with broader performance and service quality objectives. These evaluation methodologies together provide a detailed framework for the platform's performance and efficacy, allowing for a thorough analysis of its suitability and potential to meet network automation needs. The next session describes the use cases utilized.

6.3 Study Case

The study cases for evaluating the NASP architecture are divided into two cases, namely: (i) 3GPP architectures mapping and allocation of the template declared by the GSMA of mIoT with broken resources such as radio, different TN topologies and shared resources in the Core; (ii) Non-3GPP access networks using Non-trusted connections over the AMF and UPF communication interfaces. For this, one physically distributed Kubernetes Cluster, four virtualized SDN WAN, and emulated gNB with an emulated UE. Each study case represents different scenarios, where the 3GPP case has (i) a Full centralized slice, (ii) a Full edge slice, (iii) UPF edge and centralized Core, and (iv) a shared slice. Non-3GPP has the following scenario (i) Full centralized slice, as described in Table 6.

Four distinct scenarios for network slice configurations are introduced, each tailored to meet specific operational requirements and goals. The "mIoT" scenario features all network functions as isolated and dedicated, strategically located in the Cloud region to capitalize on lower operational costs and abundant resources. This scenario is configured using the Long Path route as Backhaul TN NSSI. Besides, the "URLLC" scenario aims to achieve minimal latency by allocating the main functions for PDU session establishment and resignation and the UPF as dedicated resources at the Edge location for lower latency in UE admission and bandwidth. The Edge location has a minimal latency when accessing RAN resources, such as lower than 1 ms. The scenario is also configured with a Short Path route as Backhaul focused on low latency over the TN domain. In the third scenario, called Shared, only specific functions, such as UPF, AMF, and SMF, are isolated. The shared NFs are allocated in the Cloud data center as well as the specific NFs. UPF, AMF, and SMF are instantiated in run time, and shared functions are reconfigured in run time to serve the new S-NSSAI. The scenario also uses the Longe Path route as TN NSSI. The final scenario, called "Non-3GPP," has a very similar infrastructure configuration to the mIoT scenario, except for the difference in the access function. The Non-3GPP scenario deploys an N3IWF NF to receive the UE traffic.

Table 6 – Slice Deployment Scenarios.

Scenario Name & ID	Shared NFs	Slice-specific NFs	Edge Deployments	Central Deployments	Transport Route
mIoT (1)	-	All 5G NFs	-	All 5G NFs	Long Path
URLLC (2)	-	All 5G CP NFs	UPF,AMF, SMF	5G CP NFs	Short Path
Shared (3)	5G CP NFs	AMF, SMF UPF	-	All 5G CP NFs	Long Path
Non-3GPP (4)	-	All 5G NFs + N3IWF	-	All 5G CP NFs	Long Path

6.3.1 3GPP O-RAN

The first case study to be evaluated is the O-RAN to explore the complete 3GPP 5G network stack, managing the three domains and their breaks as described in Chapter 2 dynamic management at run-time providing NSaaS is one of the gaps in the literature, along with E2E management. Therefore, the case study represented in Figure 17, where the instances are managed at run-time, becomes an important research environment.

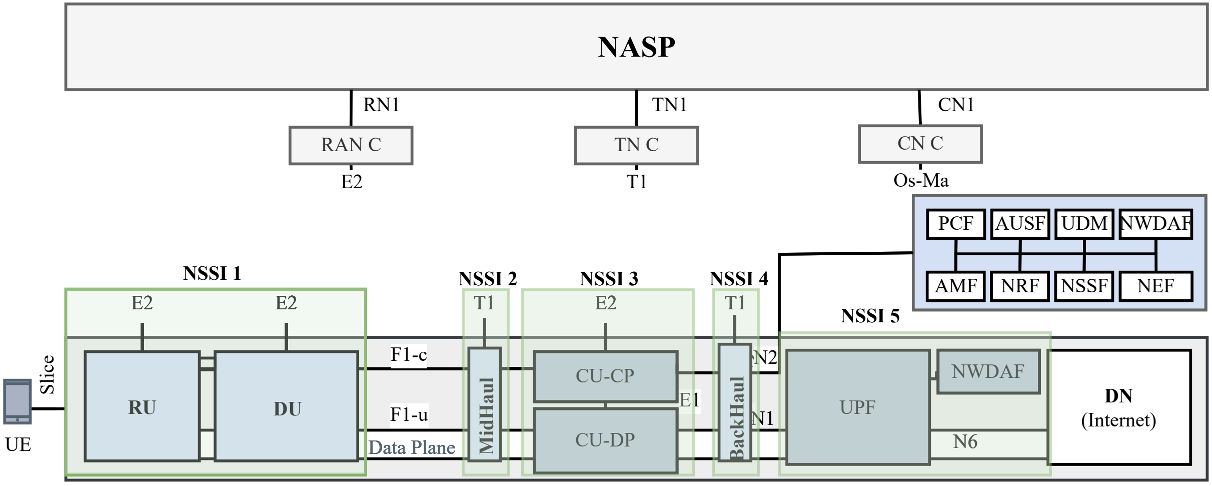
The NG-RAN simulation initially is integrated with the My5G-RANTester, on the entire RAN disaggregation protocol stack. First, however, nodes must be available for instantiation and able to request at the RAN Controller interface. Next, the node is virtualized in a container for the prototype to orchestrate Kubernetes. Therefore, according to the characteristics of Kubernetes, initially, the multi-tasking infrastructure or O-Cloud (from the O-RAN architecture) is developed in the cluster format and subdivided into three distinct computational resources (CRs). Finally, the Kubernetes platform validates the prototype once the orchestrator is customized within the scaling function and the multi-tasking infrastructure's global vision.

Figure 17 presents the platform in a single block, highlighting only the communication interfaces. Therefore, it is possible to visualize the relationship between the instances and infrastructure with the NASP, highlighting its immutability in the face of different scenarios, as presented with the case study for non-3GPP networks.

6.3.2 Non 3GPP

The second case study to be evaluated is the non-3GPP network to explore the adaptability of the proposed NASP architecture. The 3GPP standard does not define the advanced trust level for a non-3GPP network, but we can infer that behavior similar to that of a 3GPP (PENTTINEN,

Figure 17 – O-RAN study case.



2021) network is expected. Figure 18 illustrates the main components of reliable non-3GPP access.

Figure 18 – Non 3GPP study case.

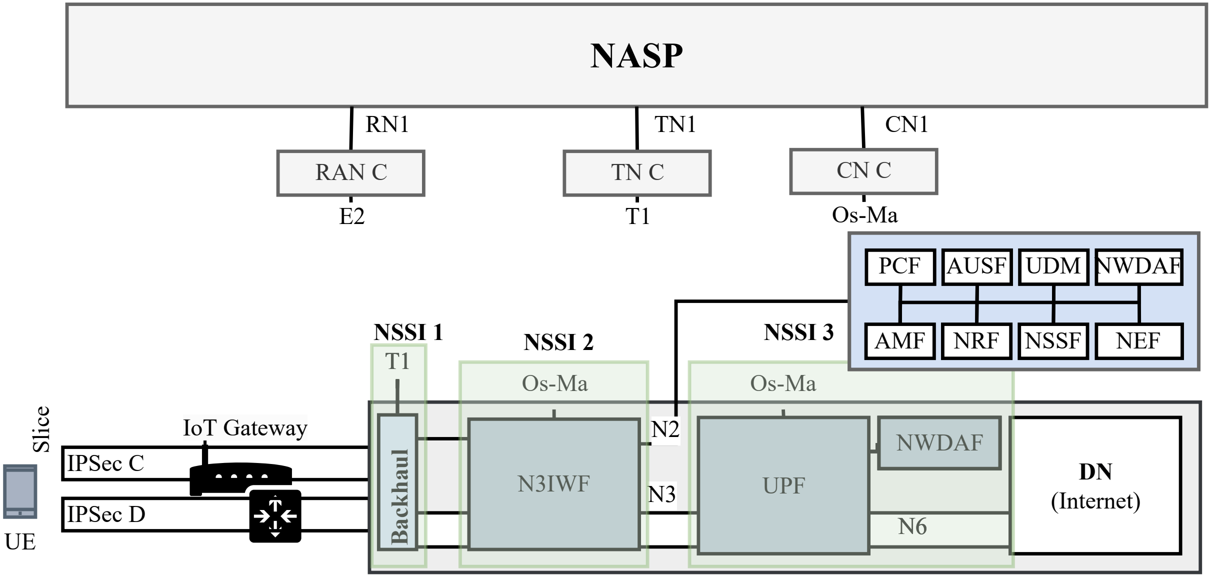


Figure 18 shows an NSI composed of three NSSIs, two of which are from the core domain and one from the transport domain. NSSI 1 represents the TN interfacing between the IoT Gateway and the non-3GPP Inter-Working Function (N3IWF) function. For non-3GPP connections, Internet Protocol Security (IPSec) tunnels are defined for the control and data planes. The N3IWF function must receive these tunnels and finally be integrated into the 3GPP network. This representation highlights the adaptability of the NASP architecture with the same interfaces, i.e., it is possible to integrate with different final network architectures and controllers. In the figure, it is possible to see that the RAN domain is useless, but given the definition of the use case in the initial request, the orchestrated platform must develop and manage the resources with the same transparency as any 3GPP network.

7 RESULTS

This chapter presents the results of the NASP solution. The chapter is divided into two sections. Section 7.1 shows the experimental evaluation based on NASP Slices design and deployment. Section 7.2 presents the experimental evaluation based on NASP Slices lifecycle management.

7.1 NASP Design and Deployment

The performance evaluation of the NASP architecture for designing and implementing the slices requested by the platform providing NSaaS was organized into three distinct analyses and a conclusion with final considerations. The first evaluation examines the relationship among the deployment time, configuration time, and the total time until availability for a UE connection in the requested slice. The second investigation explores the steps for instantiating a network slice considering the mIoT scenario, presenting the time for NASP configurations, deployment, and configuration of the NFs. The third analysis presents the relationship between the number of deployments and re-configurations required to realize a new slice or reconfigure an existing slice. Finally, the associations between the three results are discussed from a global perspective of the NASP architecture.

This initial analysis reveals the research outcomes, focusing on the interrelationship between the two principal objectives studied: the design of E2E network slices and their deployment automation to provide these slices as a service. Figure 19 illustrates the total time required from the request to the actual provisioning of an E2E network slice, characterized as NSI, based on the four scenarios detailed in the preceding chapter. This overview seeks to highlight the efficiency and feasibility of the proposed solutions for the implementation and automated management of network slices, emphasizing the platform's request flexibility and adaptability.

Figure 19 breaks down the process of setting up an NSI, from request to implementation. This process, called NASP Design, includes phases such as setting up S-NSSAI identifiers, selecting and sizing the right NFs, and allocating the necessary computer resources. Furthermore, the figure shows the deployment time for specific NSSIs in the CN, RAN, and TN domains. Highlighting it also indicates the time for NFs to self-configure after the E2E link stabilizes. This last period reflects the time required for functions to connect and configure each other within the 5G Core's SBA, emphasizing the complexity and sophistication involved in orchestrating and managing network slices in 5G networks

Figure 19 shows a Slice deployment time for different scenarios where the total time in divided into five steps, NASP Design, Core deployment, RAN deployment, TN deployment and slice auto-configuration. The Figure also shows the total time of scenarios mIoT, URLLC, Shared and Non-3GPP where it was 52.05, 53.41, 22.36, and 50.16 seconds, respectively, showing a 58% reduction between the longest and shortest E2E slice deployment. This graphic also

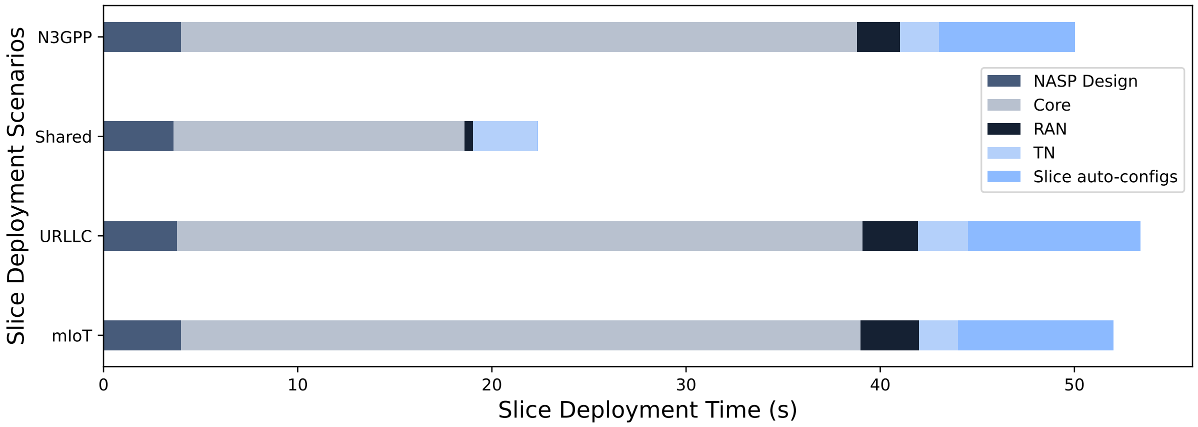


Figure 19 – Slice Deployment Time for Different Scenarios.

reveals the efficiency and variability of process times across different scenarios. Shared Scenario emerges as the most efficient, boasting the shortest total time and the highest efficiency per step, with an average time of 5.59 seconds across its four steps. This efficiency is notable compared to the other scenarios, which have a longer total time and more steps, each with average times ranging from 10.0033 to 10.682 seconds. The examination of step times across scenarios, where comparable, uncovers a significant variability in the time taken for Step 2, highlighting the optimization in the deployment of fewer NFs and reconfiguring the necessary NFs. Conversely, Step 1 shows low variability, indicating a standardized process across scenarios where an optimization affects all scenarios.

Figure 19 clearly shows that the Shared scenario had the fastest network slice allocation process due to shared functions. This reduced the need to implement many functions from scratch and reconfigure others for a new slice, speeding up the setup significantly. Furthermore, the figure reveals that the allocation times for the other scenarios were quite similar, especially the URLL scenario, where the use of functions at the Edge made its total allocation time slightly longer compared to Non-3GPP and mIoT scenarios. The average time to set up the NASP was 3.8 seconds for any scenario, while the time to set up the CN varied depending on the number of NFs required. The setup and deployment steps in the RAN showed little difference between the 3GPP and Non-3GPP cases, indicating that N3IWF can be considered part of the RAN, with all setup steps following a similar pattern for both.

When analyzing the steps, Figure 20 shows the detailed duration of each step at the domain level and a deep dive detailing each step within the domain. Based on the previous analysis, it was observed that the variation in preparation time for each domain was considerably high. Therefore, a detailed study was chosen to observe the deployment of each NSSI. The details of the characteristics are divided into NSSI deployment and NASP platform processing steps.

The analysis of the NASP platform, as illustrated in Figure 20, reveals a significantly higher volume of NFs in the Core domain compared to other domains, accompanied by an increase in the average time required to deploy these functions. This phenomenon can be attributed to the minimum time needed for internal communication through the Helm tool, which facilitates interaction between the NASP and the Core domain controller. A similar observation applies to the RAN domain, where, on the other hand, the process is noticeably more agile in the TN domain. In the case of TN, NASP interacts directly with the prototype's specific controller and subsequently with the ONOS controller to update the SDN settings related to emulation. This procedure optimizes communication and deployment, significantly accelerating the process, even when it is necessary to establish various route intentions to configure a complete path that supports E2E communication among domains.

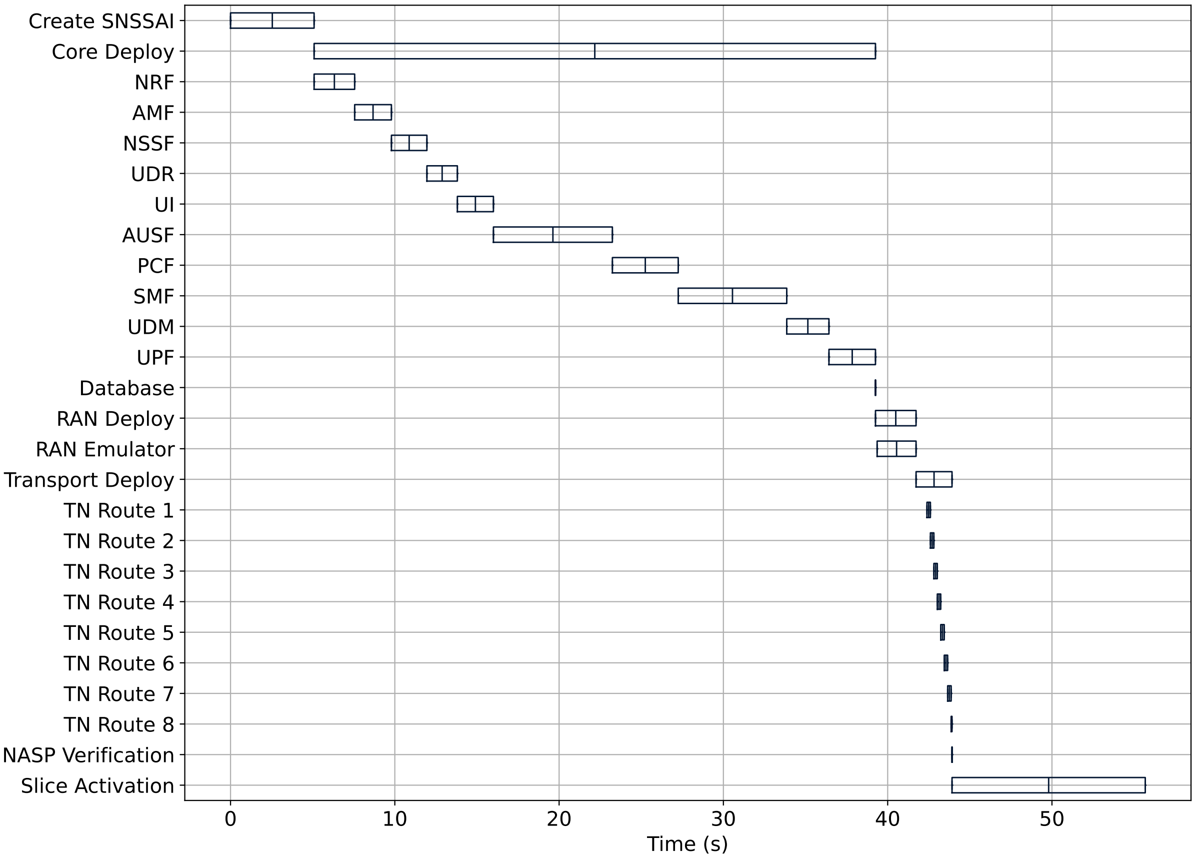


Figure 20 – Slice Deployment Steps Time.

Another analysis is shown in Figure 21, indicating the relationship between the number of new NFs and reconfigurations in the four scenarios presented. It is observed that while new deployments offer isolation, better performance in resource management, minimizing failure possibilities, and improving management capabilities, they demand additional resources, as shown in the graphic.

Isolated NSSIs require a larger number of deployed functions to cover basic functionalities. On the other hand, shared NSSIs allow for NF reconfigurations, minimizing the need for resources, but they could lead to service interruptions, configuration errors, and increased complexity in configuration and management.

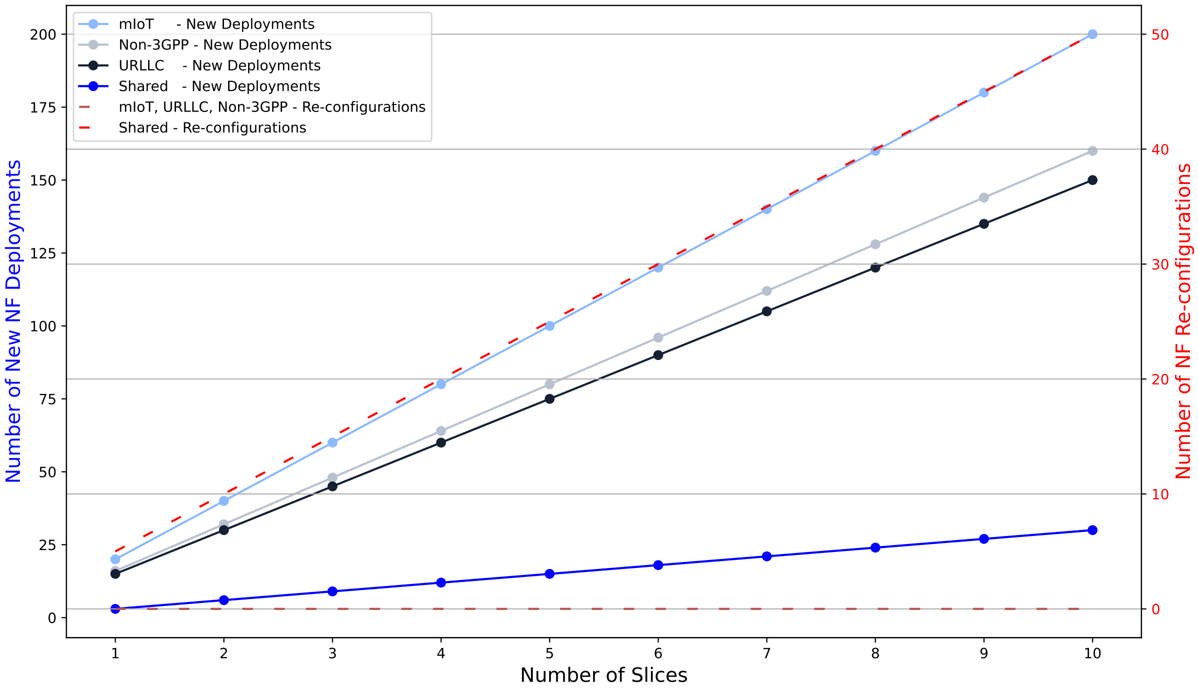


Figure 21 – Deployment Over Time.

7.2 Slice performance and lifecycle management

The performance evaluation of the NASP architecture for slice performance and lifecycle management was organized into three distinct analyses and a conclusion with final considerations. The first assessment focuses on latency variation when connecting a UE device to a previously active network slice, considering four distinct scenarios. The second analysis investigates the availability and adaptability of active network slices, examining modifications that can be made in real time. The third analysis investigates the behavior of CRs employed by the NASP solution. Finally, a cost analysis is conducted, considering data collected during the research and publicly available prices for cloud computing services.

The first analysis investigates the behavior of new UE connection requests to the available slice. Therefore, it evaluates the response time behavior between a connection request and the stabilization of the data plane communication tunnel. The collection was carried out using the My5G-RANTester tool with the test configuration for new UE connections, and the data from the action performed are exported at the end.

For the performance analysis of a slice, Figure 22 presents the histogram as a result of the test load carried out. The sample was collected using ten test batteries with 20 runs for each scenario, totaling 200 connections per scenario. For better reliability of the analyzed data, outliers with a variation of $\pm 3\sigma$ were removed. As can be seen, the connection latency in the URLLC scenario is lower than the other scenarios due to the proximity of the AMF, SMF, and UPF functions and the low latency route selected, using an exclusive path in the TN, thus making the PDU Sessions Establishment request faster. Scenarios mIoT and Non-3GPP, despite being different use cases with exclusive access functions (3GPP and Non-3GPP), had very similar behavior. Finally, the Shared scenario, which is the most resource-abundant but with the highest latency, showed the longest connection time and PDU session establishment.

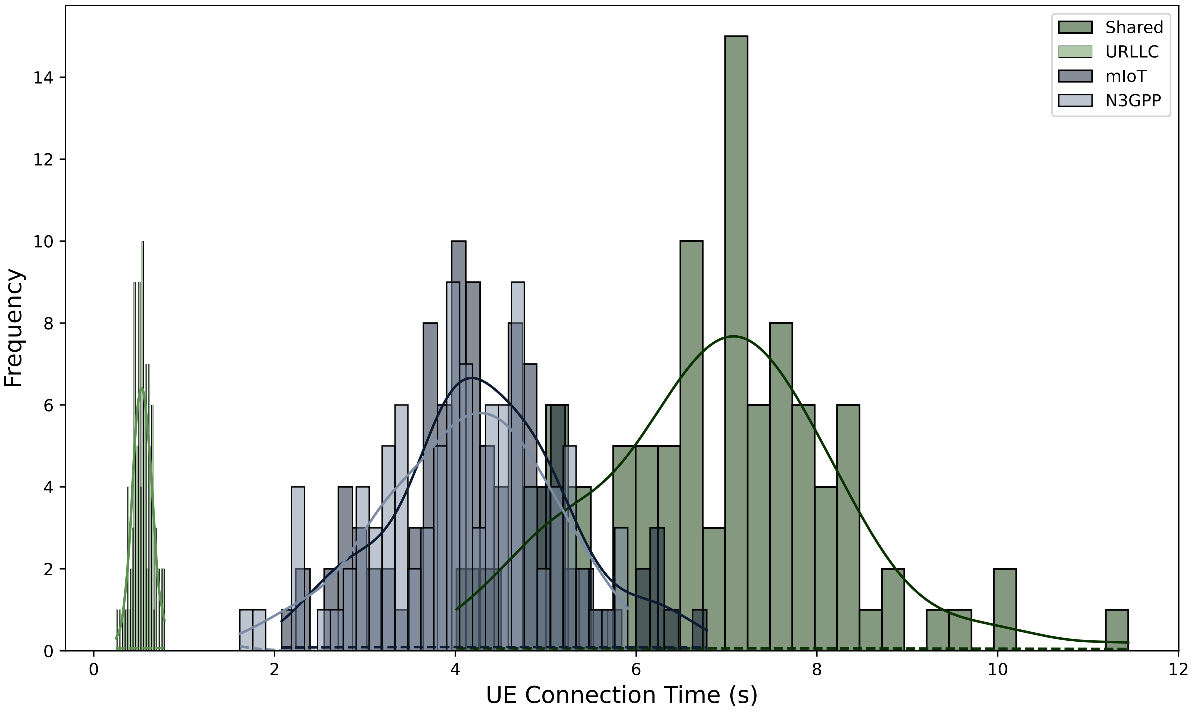


Figure 22 – UE Connection Time for Different Scenarios.

Following the UE connection time, test batteries to generate requests for UE registrations in the 5G core, select specific network slices and establish connections between gNB and the 5G core. We automatically changed the configurations of the network slices supported by the 5G core through partial dynamic and NASP orchestration among the bursts of registration requests. Moreover, the messages were recorded in log formats, and the results represented an average of 20 repetitions and a data read period of 0.5 s without showing variation between executions. These repetitions presented statistically significant values, as they reached a 95% confidence level.

The analysis refers to a network slice's availability (0 or 1) during its reconfiguration process at run-time. This process concerns the request, processing, and response of reconfiguration requests of the network functions provided by the 5G core and changes in the virtualized infrastructure. Figure 23 shows the beginning and end of this reconfiguration process over a time window. NASP orchestration did not present any interruption in service provision during the reconfiguration process of 9s. This behavior occurs due to the abstraction layer created by Kubernetes (K8S) on the gNB communication and 5G core. K8S uses a single IP informed to all gNBs and has all re-configurations performed at run-time, i.e., closing and starting new Stream Control Transmission Protocol (SCTP) connections with AMFs. The reconfiguration time results from the allocation of AMFs and the core reconfiguration related to the queue of requests processed during the change of AMFs.

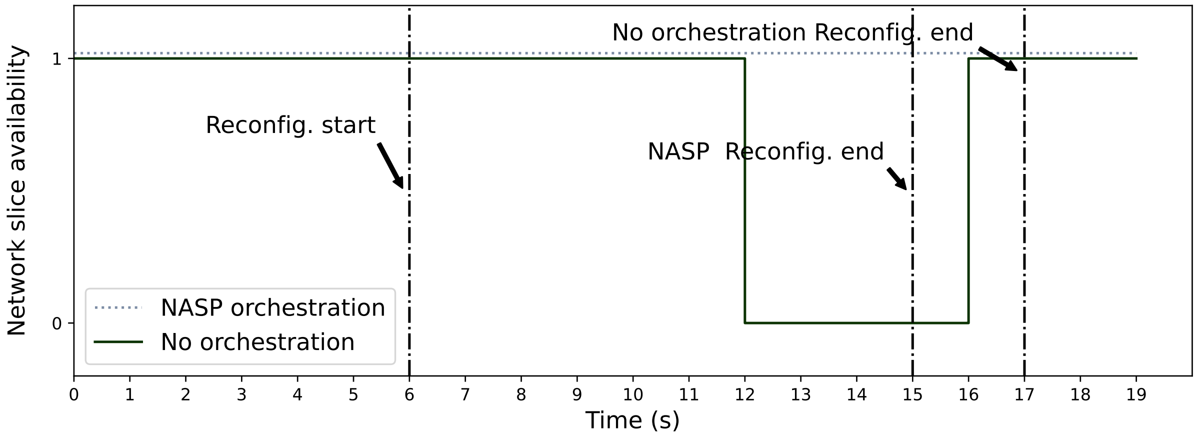


Figure 23 – Availability of network slices.

Unavailability occurs between seconds 12 and 16, and the reconfiguration time occurs in 11s in the partial dynamic orchestration. Processing requests are correctly terminated, but this unavailability is observed during the destruction period, the configuration of the new network slice, and the instantiation of the 5G core network functions. This behavior occurs because the destruction of AMF to update its settings is impossible due to a new instance requesting the same IP from the Next Generation Application Protocol (NGAP) protocol subnet, managed within K8S. The controller proposed in the NSSMF Core impacts the QoS during reconfiguration steps, guaranteeing the service available in the network slicing reconfiguration process.

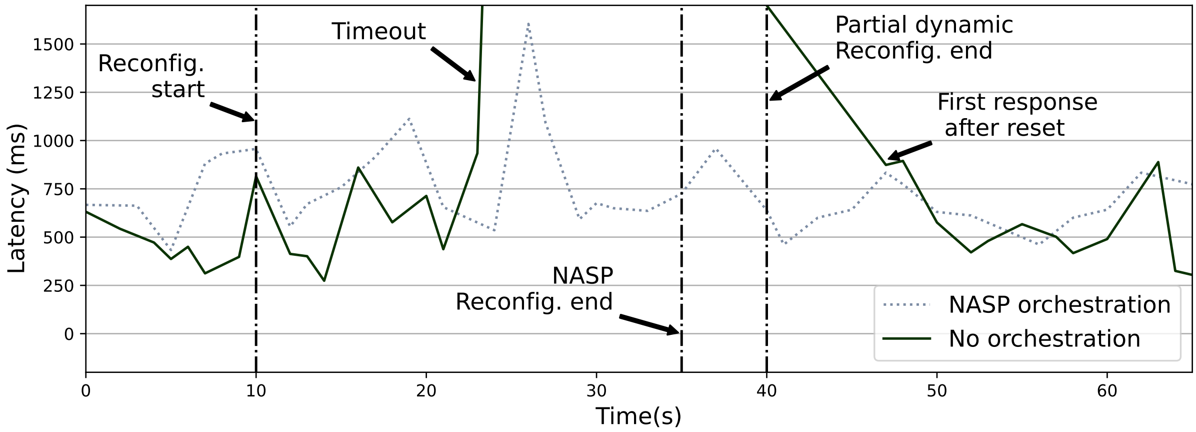


Figure 24 – Latency of the reconfiguration process.

An analysis of the latency presented for UE registration before, during, and after the reconfiguration process of a network slice. Figure 24 shows the latency of the reconfiguration process with a data read period of 0.5s, considering the NASP orchestration. In this case, the latency peak was 1.5s, returning to the default of approximately 600ms, as observed before the reconfiguration. This high latency results from the low CRs available and the communication among multiple networks and subnetworks used in the evaluation scenario. Figure 24 also shows the latency for the partial dynamic orchestration. After starting the reconfiguration process, at 10s, we can see a significant increase in latency. Observed no responses to registration requests from the second 23 up to the 47, i.e., due to timeout messages and the service was unavailable. After the reconfiguration, the latency stabilized at around 700 ms.

The analysis refers to the adaptability of the network slices used in NASP orchestration. Figure 25 presents a network slice 1 configured to simultaneously accept up to seven UEs to provide a high-quality service for these connected UEs. When the number of UEs is reached in network slice 1, the NSSMF Core controller triggers the network reconfiguration, stopping the availability of network slice 1 for new requests for UE records and starting the availability process in network slice 2. The controller proposed guarantees the fulfillment of the service level agreements established at run-time without interfering with other available services.

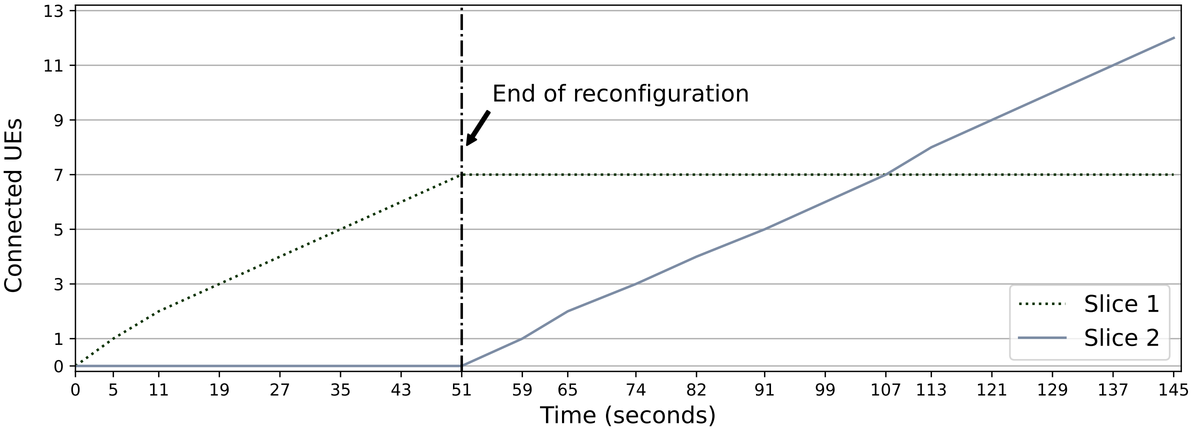


Figure 25 – Adaptability of network slices.

For the next analysis, the average consumption of computational resources was observed, considering infrastructure, platform, and slices in a control network load. For this, ten test rounds were conducted, in which each battery requested ten NSIs where each NSI would receive 40 UE connections performing the PDU Session Establishment and PDU Release process per minute, in which data were collected from all machines simultaneously with a collection frequency of once per second throughout the process.

Figure 26 shows the variation of CPU and RAM combined among all resources used in the testing environment, including K8s Cluster, Mininet, and ONOS, among others. The figure displays the moments from the start of an NSI request, which occurred at 10 s, to the conclusion, which happened at 117 s, with all slices made available to the network with a data read period of 0.5s. The graphic also shows the resource lines (CPU, RAM) over time, with the fluctuation around the main lines being their standard deviations with standard error. In the figure, it is possible to see the increase in RC with a small plateau in CPU due to the interval between requests made by automation. Moreover, it is possible to see that 600 MB of RAM and 1.2 vCPU consumption are the average values over a lifecycle.

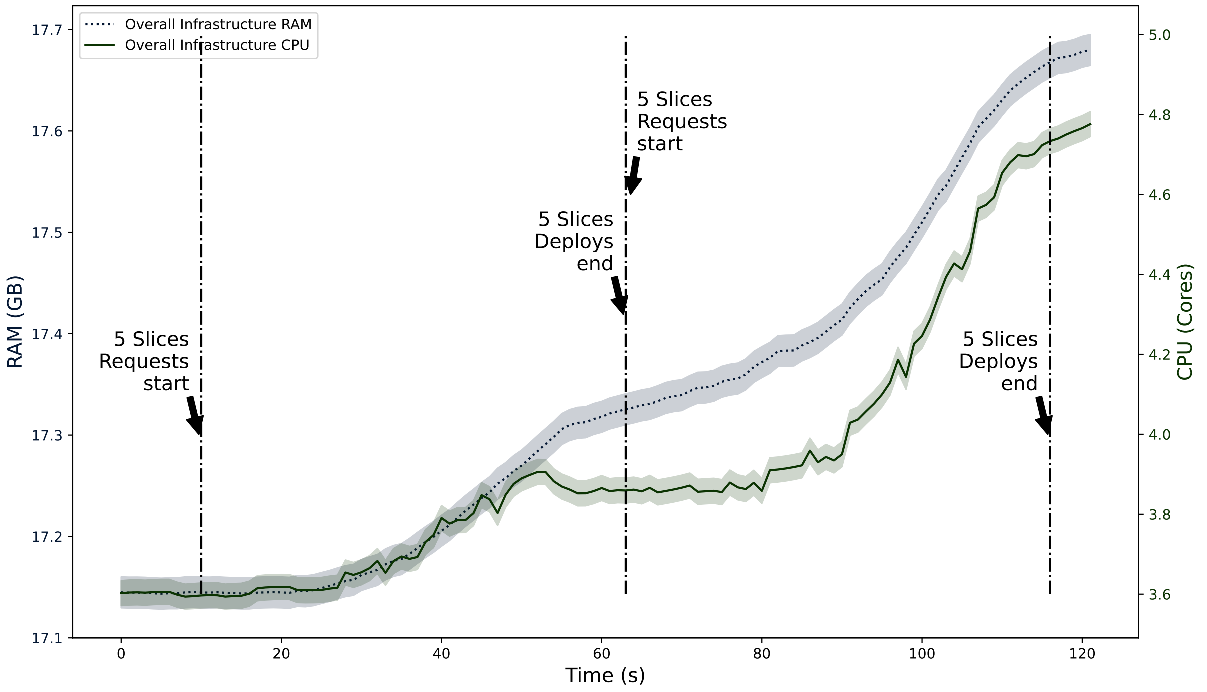


Figure 26 – Trends in RAM and CPU Usage Over Time.

In line with our approach, we carried out similar tests by setting up ten network slices connected to five UEs, each pushing a traffic of 200 Mbps, to examine the escalating load impact on our infrastructure, mainly focusing on the data plane. However, for reasons yet to be determined, the bandwidth when utilizing the data plane was capped at 30 Mbps per UE, leading to an unexpectedly low load for our data plane stress analysis. We conducted comparable experiments using the UEs within the same virtual environment but through the default interface, where we achieved the anticipated bandwidth limits of 200 Mbps. Further testing and analysis are required to pinpoint the cause of this discrepancy.

Finally, the last analysis deals with the evaluation cost related to the test of computational resource consumption used above. To carry out the study, it was necessary to find the relationship between CPU and RAM resources and the operating cost of these resources in their respective environments (Edge, metropolitan, or cloud). For this purpose, a linear regression was performed using the values from Table 4, resulting in three different equations, one for each environment. Afterward, the consumption data collected in the previous evaluation was used to predict the price for each environment.

$$Edge = 39.42 * CPU + 3.65 * RAM - 22.56 \tag{7.1}$$

$$Metropolitan = 33.58 * CPU + -1.46 * RAM + 6.63 \tag{7.2}$$

$$Cloud = 15.187 * CPU + RAM + 15.996 \tag{7.3}$$

As shown in Figure 27, it is possible to analyze the variation cost between different environments, considering the deployment of all core functions together. An exponential variation can be noticed in the cost of a slice deployed in the Cloud environment compared to the Edge environment. The final cost of five slices in the Cloud, Metropolitan, and Edge environments are $91.90, $110.89, and $192.73, respectively. There is an increase in cost between Cloud and Edge of $109.71. The final cost for ten network slices in the Cloud, Metropolitan, and Edge environments are $106.20, $141.19, and $230.23, respectively. In this case, the increased cost of operating slices between the Cloud and Edge environments is $116.78.

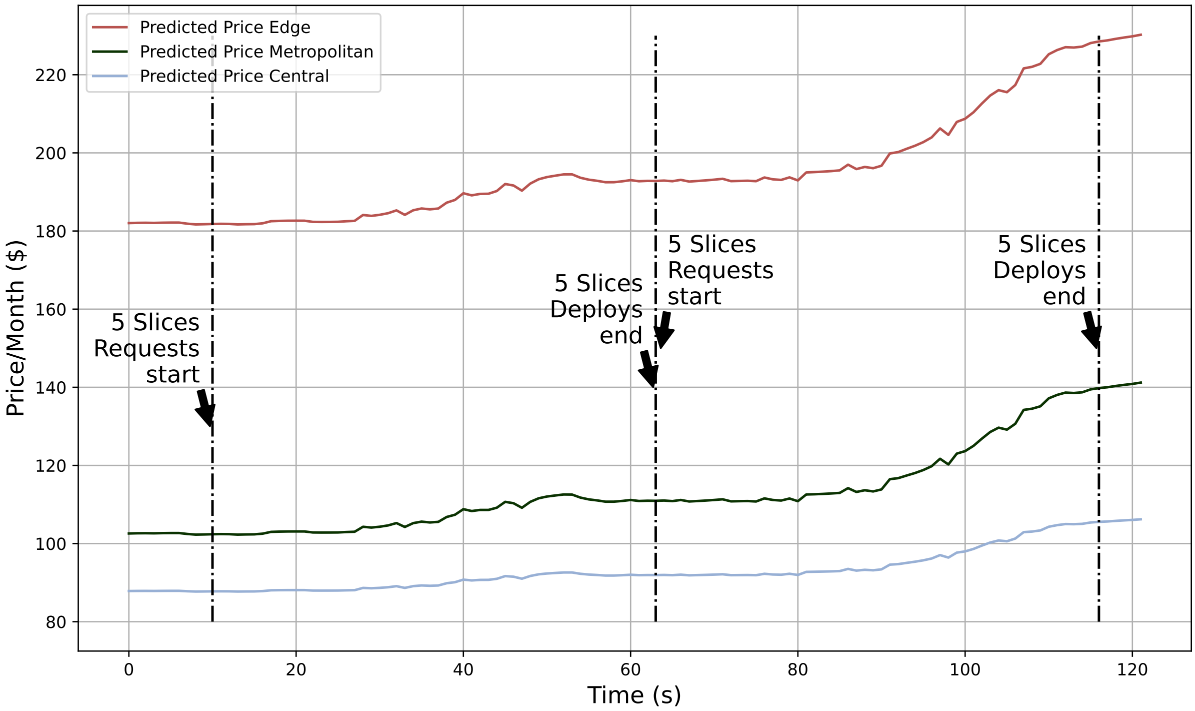


Figure 27 – Predicted Price/Month vs. vCPU RAM.

7.3 Final Considerations

This study aimed to investigate an NSaaS architecture to analyze the slice design, automation techniques for deployment, and management over E2E network slices. The obtained results demonstrate the architecture's adaptability across different scenarios and environments, achieving the design of NSIs upon request of a combination of GSMA, 3GPP, and ETSI templates. The results also detailed the time spent on the instantiation of an NSI, where it's possible to see the relationship of time spent between domain for slice instantiation and configuration. Furthermore, the results provided insights into managing shared and isolated slices and their deployment and reconfiguration costs. Upon analyzing the connection times and the use of resources for the management and execution of the slices, it was possible to observe the difference among the chosen environments for analysis.

Considering the design and instantiation evaluation, the instantiation behavior of the NFs used for the network Core is significantly important for the final time of NSI instantiation. Although it varies depending on the environment, the result was similar for all of them, where the Core has the longest implementation time. The isolation technique used for NFs, considering each as an NSSF, brought several orchestration and management advantages to the platform but significantly impacted the domain's instantiation time when requesting a new slice.

The second analysis presented the performance data of the slice and the lifecycle management considering runtime changes, as well as the CRs used. A significant result highlighted was the connection time of a UE when comparing scenarios, where the connection time is relatively higher for scenarios that utilized centralized resources with higher latency transport routes. Interesting results are also shown on the management view of NSI in real time, presenting adaptability and the operation of load balancers and a design aimed at high availability, where the reconfiguration of NFs, such as AMF, does not impact UEs using the network. Although the work does not investigate deeply into the data plane, the tests also perform measurements showing the ideal environment and the results obtained from the used tools, thus being an initial test with the mapping of potential gaps for the enhancement of the platform and its resources like domain controllers and the NFs themselves.

8 CONCLUSION

The evolution proposed by 5G mobile networks drives digital transformation in different segments of society. This transformation is led by the standards organizations ITU-T, 3GPP, ETSI, and O-RAN, which drive significant changes in architectures, concepts, and technologies. For example, the main specification of 3GPP, the Release (3GPP, 2021), addresses evolutions in the entire chain of mobile networks. In particular, the slice as a service, which evolves into templates and SLAs, experiences significant architectural transformations and immersion in virtualization concepts. Such changes open space for numerous research opportunities.

Despite the envisioned transformations, significant challenges are presumed to improve the adaptability and availability skills of networks optimized for their respective requirements. Therefore, the main motivations of this work refer from the observation that there is currently, there is a lack of comprehensive guidelines that would help address the challenges related to the organization of the design and allocation process of network slices aligned with provisioning standards defined by entities, considering automation and real-time network slice provisioning of ZSM on a Network Slice as a Service platform.

Based on the research conducted in the literature, a gap was observed and, consequently, an opportunity for research development on the topic of definition and instantiation of E2E network slices about (i) definition of resources, computational and virtual slices of network based on SLA/SLS requests, (ii) linking of E2E network slices connecting and relating the three domains of a network (Core, RAN, TN), and (iii) automation of management and orchestration during the lifecycle of an E2E network slice. This work led us to the research question: **SQR1: How to orchestrate and integrate the standardized components to provide network slice as a service?**

The NASP architecture explored the orchestration of Network Slice Instances with the proposition of techniques to translate business templates to technical instance definitions. Furthermore, meeting the specified requirements for the breakdown of GSMA template definitions, with the latency requirement being one of the most challenging. The work also presented a prototype in an experimental environment for project validation and detailed two specific case studies and evaluation metrics covering the three topic goals.

The prototype was developed to quantify the efficacy of the NASP architecture, allowing for a detailed analysis of two fundamental issues. The first is the design process, which involves applying a combination of GSMA templates and others to create NSIs and establishing a link between all domains involved for a complete E2E network slice configuration. The second concerns automation strategies and the management of the flow of requests necessary to enable the instantiation and ongoing management of these slices. To this end, the prototype was built using Python 3, incorporating specific libraries for communication with domain controllers, such as K8s and ONOS, and implementing an HTTP server that makes its interface available through the HTTP REST protocol. The system also makes use of Linux tools like iptables net-tools, namespaces, and CGROUPS, executing modifications in namespaces routing rules and in the creation of VLANs in K8s, allowing for the proper segregation and interconnection of the domains of the E2E network slices.

Two case studies were adopted for the practical analysis, subdivided into four distinct scenarios. This division was based on the functionalities and specificities of each requested NST, demonstrating the adaptability and agnosticism of the tool in the face of varied demands. The experiments investigated two main research topics: the design and translation of SLA-based requests, encompassing E2E network slices, and the automation process of deploying and managing slices in real time. In this context, the deployment time for each slice request in the proposed scenarios, detailing the steps for the instantiation of an E2E network slice and the relationship between the deployment and reconfiguration of network slices were evaluated. Additionally, the connection time for a UE within the operating slice, the adaptability and real time management of the slices, the CRs employed for the instantiation of ten slices, and finally, the costs associated with different scenarios were analyzed.

The study's results validated the adaptability and effectiveness of the NASP solution, highlighting that approximately 66% of the total instantiation time was dedicated to the Core domain, reflecting the strategy of managing each NF as an individual NSST. The comparison between isolated and shared slices revealed a balance between resource consumption and adaptability, with isolated slices offering greater flexibility at the cost of more intensive resource use, while shared slices showed resource efficiency with greater ease of allocation. Performance tests indicated significant optimization in the time to establish PDU sessions in edge scenarios, using NFs with lower latency routes. The resource consumption for operating ten simultaneous network slices was manageable, with a continuous flow of PDU sessions requiring 600 Mb of RAM and 1.4 vCPU. The cost analysis highlighted an average increase of 102% for operations in edge environments compared to cloud scenarios, underlining cost considerations in the choice of resource locations. The NASP platform is a viable and efficient solution for advancing the NSaaS architecture, adequately balancing adaptability, resource efficiency, and cost considerations.

8.1 Contributions

NASP is an NSaaS solution for E2E network slices, focusing on solving design issues and automating the instantiation and management of NFs. It seeks an architectural solution that can be applied through a platform aligned with the directives of GSMA, ETSI, and 3GPP entities. It contributes the following specific features:

1. Design and develop an NSaaS platform to allocate, active, and deallocate slice instances in run-time.

2. Design and development of integration between three major mobile network institutions

to merge all required definitions and create an E2E-integrated architecture.

3. Translate GSMA business-level templates to NST and NSST templates.

4. Develop an E2E Slice allocation to Non3GPP applications.

Compared to the related work in Chapter 3, the NASP solution stands out as the only platform solution for providing and managing E2E network slices covering the RAN, Core, and Transport domains. It integrates with requests through GSMA templates that are translated into NSIs.

8.2 Limitations

In this section, possible limitations identified during the development of the NASP solution are listed. First, a limitation of the solution is the factor of the access network emulation, where the platform has limited knowledge about the access network due to the abstractions and emulations carried out in the prototype's development. Another limitation is due to the emulation used in the access network, where the performance tests were limited to a bandwidth of 40 Mb/s, not being sufficient for stress and load tests, where the performance, resilience, and adaptability of the network from the perspective of the data plane would be analyzed. The non-3GPP scenario also presented limitations considering the integration between the network and UE outside the cluster environment due to simulator permissions issues. Finally, due to the use of different tools in development, the solution's instability did not allow for long-duration tests, considering weeks of stress and load on the solution. Tools such as GTP-5G often required reinstallation for proper functioning and communication with the UPF used from free5GC.

8.3 Future Works

Based on the previously presented limitations, some of these can be viewed as opportunities for future work for the NASP solution, as detailed below:

1. Integration with Distributed O-RAN Models: Improve NASP to make it a capable platform for integration with distributed RAN, addressing the gap in experimental analyses on orchestration and management of E2E slices.

2. Optimization of Data Plane Usage: Optimize resources to enable stress and load testing on the data plane.

3. Enhancement of GSMA Template Translation Techniques: Introduce algorithms to find the best definitions according to SLA requirements at the time of slice request.

4. Enhancement of Closed Loop Analyses: Collect and analyze specific data of network slices in real time on the data and control planes, making decisions based on the collected values.

8.4 Publications

During this research, articles were developed addressing issues related to total dynamic slice allocation to different conferences evaluated by Qualis CC, such as the A1-scored IEEE Global Communications Conference (GLOBECOM), A4-scored Brazilian Symposium on Computer Networks and Distributed Systems (SBRC), and B4 scored the Brazilian Symposium of Telecommunication, and Signal Processing (SBrT). The submitted articles are listed below:

- GRINGS, F. et al. Full dynamic orchestration in 5G core network slicing over a cloud-native platform, 2022. Submitted and accepted to the IEEE Global Communications Conference (GLOBECOM), pp. 2885-2890.
- GRINGS, F. et al. Orquestração dinâmica total de fatiamento de rede no núcleo 5G sobre plataforma nativa de computação em nuvem. In: XL Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, Porto Alegre, RS, Brasil. Anais. SBC, 2022. pp. 349–362.
- LIMA, H. et al. Controle de Admissão para Network Slicing Ciente de Recursos de Rede e de Processamento. In: Brazilian Symposium of Telecommunication end Signal Processing, 2022.
- MACEDO, C. et al. Improved support for UAV-based computer vision applications in Search and Rescue operations via RAN Intelligent Controllers. In: Brazilian Symposium of Telecommunication end Signal Processing.

REFERENCES

3GPP. **Study on New Radio Access Technology; Radio Access Architecture and Interfaces (Release 14)**. [S.l.]: $3^r d$ Generation Partnership Project (3GPP), 2017. Technical Recommendation (TR). (38.801).

3GPP. **Service Requirements for the 5G System, document TS 21.916 v16.0.0**. [S.l.]: $3^r d$ Generation Partnership Project (3GPP), 2018. Technical Specification (TS). (21.916).

3GPP. **System Architecture for the 5G (Release 15)**. [S.l.]: 3^{rd} Generation Partnership Project (3GPP), 2018. Technical Recommendation (TR). (23.501).

3GPP. **Technical Specification Group Services and System Aspects (Release 15)**. [S.l.]: 3^r ^{d} Generation Partnership Project (3GPP), 2019. Release 15 Description. (21.915).

3GPP. **System Architecture for the 5G (Release 17)**. [S.l.]: $3^{r\ d}$ Generation Partnership Project (3GPP), 2021. Management and orchestration; 5G performance measurements. (28.552).

ABBAS, K. et al. IBNSlicing: intent-based network slicing framework for 5G networks using deep learning. **2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS)**, [S.l.], p. 19–24, 2020.

AFOLABI, I. et al. Network slicing and softwarization: a survey on principles, enabling technologies, and solutions. **IEEE Communications Surveys & Tutorials**, [S.l.], v. 20, n. 3, p. 2429–2453, 2018.

ALLIANCE, B. N.; HATTACHI, R. E.; ERFANIAN, J. NGMN 5G White Paper. , [S.l.], 2015.

ALLIANCE, B. N.; HATTACHI, R. E.; ERFANIAN, J. NGMN 5G White Paper. , [S.l.], 2015.

AUTHORS, T. K. **Production-grade container orchestration**. [S.l.]: The Linux Foundation, 2023. (2021-12-15).

BARANDA, J. et al. Scaling Composite NFV-Network Services. In: TWENTY-FIRST INTERNATIONAL SYMPOSIUM ON THEORY, ALGORITHMIC FOUNDATIONS, AND PROTOCOL DESIGN FOR MOBILE NETWORKS AND MOBILE COMPUTING, 2020. **Proceedings. . .** [S.l.: s.n.], 2020. p. 307–308.

BEGA, D. et al. Network slicing meets artificial intelligence: an ai-based framework for slice management. **IEEE Communications Magazine**, [S.l.], v. 58, n. 6, p. 32–38, 2020.

BERTENYI, B. et al. Ng radio access network (ng-ran). **Journal of ICT Standardization**, [S.l.], v. 6, n. 1, p. 59–76, 2018.

BREITGAND, D. et al. Dynamic Slice Scaling Mechanisms for 5G Multi-domain Environments. In: IEEE INTERNATIONAL CONFERENCE ON NETWORK SOFTWARIZATION, 2021. **Anais. . .** [S.l.: s.n.], 2021. p. 56–62.

DALGITSIS, M. et al. Cloud-native orchestration framework for network slice federation across administrative domains in 5G/6G mobile networks. **IEEE Trans. Veh. Technol.**, [S.l.], p. 1–14, 2024.

DEVLIC, A. et al. Nesmo: network slicing management and orchestration framework. In: IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS WORKSHOPS (ICC WORKSHOPS), 2017., 2017. **Anais...** [S.l.: s.n.], 2017. p. 1202–1208.

ETSI. . [S.l.]: 3^{rd} Generation Partnership Project (3GPP), 2019. Network Functions Virtualisation (NFV). (NFV-SEC023).

ETSI. **Zero-touch network and Service Management (ZSM); Closed-Loop Automation**. [S.l.]: European Telecommunications Standards Institute, 2021. (ETSI GS ZSM 009-1).

ETSI. **Open Source Management and Orchestration**. 2022.

ETSI, G. N.-E. **Network functions virtualisation (nfv); virtualisation technologies; report on the application of different virtualisation technologies in the nfv framework**. [S.l.: s.n.], 2016.

ETSI, N. **Network functions virtualisation (nfv); terminology for main concepts in nfv**. [S.l.: s.n.], 2014.

FERNANDEZ, A. et al. Multi-Party Collaboration in 5G Networks via DLT-Enabled Marketplaces: a pragmatic approach. In: JOINT EUROPEAN CONFERENCE ON NETWORKS AND COMMUNICATIONS & 6G SUMMIT (EUCNC/6G SUMMIT), 2021., 2021. **Anais...** [S.l.: s.n.], 2021. p. 550–555.

FOUNDATION, T. L. **Open Network Automation Platform**. 2022.

GARCIA-AVILES, G. et al. Experimenting with open source tools to deploy a multi-service and multi-slice mobile network. **Computer Communications**, [S.l.], v. 150, p. 1–12, 2020.

GET started with AWS Wavelength - AWS Wavelength — docs.aws.amazon.com. [Accessed 15-02-2024],
`https://docs.aws.amazon.com/wavelength/latest/developerguide/get-started-wa`

GIANNONE, F. et al. Impact of virtualization technologies on virtualized ran midhaul latency budget: a quantitative experimental evaluation. **IEEE Communications Letters**, [S.l.], v. 23, n. 4, p. 604–607, 2019.

GKATZIOS, N. et al. Optimized placement of virtualized resources for 5G services exploiting live migration. **Photonic Network Communications**, [S.l.], v. 40, p. 233–244, 2020.

GRINGS, F. H. et al. Full dynamic orchestration in 5G core network slicing over a cloud-native platform. In: GLOBECOM 2022 - 2022 IEEE GLOBAL COMMUNICATIONS CONFERENCE, 2022. **Anais...** IEEE, 2022.

GSMA. **Migration from physical to virtual network functions – best practices and lessons learned**. 2018.

GSMA. **Generic Network Slice Template**. [S.l.]: GSM Association, 2022. (GSMA NG.116).

GSMA. **E2E Network Slicing Architecture**. [S.l.]: GSM Association, 2022. (GSMA NG.127).

JAIN, R. **The art of computer systems performance analysis**: techniques for experimental design, measurement, simulation, and modeling. [S.l.]: John Wiley & Sons, 1990.

JIANG, W.; ANTON, S. D.; DIETER SCHOTTEN, H. Intelligence slicing: a unified framework to integrate artificial intelligence into 5g networks. In: IFIP WIRELESS AND MOBILE NETWORKING CONFERENCE (WMNC), 2019., 2019. **Anais...** [S.l.: s.n.], 2019. p. 227–232.

KUKKALLI, H. et al. Evaluation of Multi-operator dynamic 5G Network Slicing for Vehicular Emergency Scenarios. In: IFIP NETWORKING CONFERENCE, 2020. **Anais...** [S.l.: s.n.], 2020. p. 761–766.

LARREA, J.; FERGUSON, A. E.; MARINA, M. K. **Corekube**: an efficient, autoscaling and resilient mobile core system. [S.l.]: ACM, 2023.

LI, J. et al. Deep reinforcement learning based computation offloading and resource allocation for mec. **2018 IEEE Wireless Communications and Networking Conference (WCNC)**, [S.l.], p. 1–6, 2018.

LINNARTZ, J. P. M. G. et al. ELIoT: enhancing LiFi for next-generation internet of things. **EURASIP J. Wirel. Commun. Netw.**, [S.l.], v. 2022, n. 1, Sept. 2022.

MEREDITH, S. E.; ALESSI, S. M.; PETRY, N. M. Smartphone applications to reduce alcohol consumption and help patients with alcohol use disorder: a state-of-the-art review. **Advanced health care technologies**, [S.l.], v. 1, p. 47, 12 2015.

NGUYEN, V. G.; DO, T. X.; KIM, Y. H. Sdn and virtualization-based lte mobile network architectures: a comprehensive survey. **Wireless Personal Communications**, [S.l.], v. 86, p. 1401–1438, 2 2016.

ORDONEZ-LUCENA et al. On the Rollout of Network Slicing in Carrier Networks: a technology radar. **Sensors**, [S.l.], v. 21, n. 23, 2021.

PENTTINEN, J. T. Support of Trusted Access Network. In: _____. **5G Second Phase Explained: the 3gpp release 16 enhancements**. [S.l.]: John Wiley & Sons, 2021. p. 219–219.

ROMMER, S. et al. Chapter 11 - network slicing. In: ROMMER, S. et al. (Ed.). **5g core networks**. [S.l.]: Academic Press, 2020. p. 247–264.

SCOTECE, D. et al. 5G-Kube: complex telco core infrastructure deployment made low-cost. **IEEE Commun. Mag.**, [S.l.], p. 1–7, 2023.

THEODOROU, V. et al. Blockchain-based zero touch service assurance in cross-domain network slicing. In: JOINT EUROPEAN CONFERENCE ON NETWORKS AND COMMUNICATIONS & 6G SUMMIT (EUCNC/6G SUMMIT), 2021., 2021. **Anais...** [S.l.: s.n.], 2021. p. 395–400.

TRANORIS, C. Openslice: an opensource oss for delivering network slice as a service. **arXiv, disponível em https://arxiv.org/pdf/2102.03290.pdf**, [S.l.], 2021.

WYSZKOWSKI, P. et al. **Comprehensive tutorial on the organization of a standards-aligned network slice/subnet design process and opportunities for its automation**. [S.l.]: Institute of Electrical and Electronics Engineers (IEEE), 2024. 1–1 p.

ZHOU, X. et al. Network slicing as a service: enabling enterprises' own software-defined cellular networks. **IEEE Communications Magazine**, [S.l.], v. 54, p. 146–153, 7 2016.

ZHOU, X. et al. Network slicing as a service: enabling enterprises’ own software-defined cellular networks. **IEEE Communications Magazine**, [S.l.], v. 54, n. 7, p. 146–153, 2016.