POLITECNICO DI TORINO

POLITECNICO DI TORINO

Master Degree course in Communications and Computer Networks Engineering

Master Degree Thesis

5G Core Network Archtecture: Network Exposure Function

Supervisors
Carla Fabiana Chiasserini
Paolo Belloni

Candidate
Eusem Abazi

Academic Year 2018-2019

Abstract

The fifth generation wireless network (5G) has already begun its journey but making it a reality comes with several challenges along the way. A huge amount of work is underway from telecom operators for designing new technologies that are going to replace the current one (4G). With the arrival of 5G technology, also a wide range of use cases and business models will be enabled. Therefore, telecom networks need to support much greater throughput, lower latency and provide higher flexibility and scalability.
To achieve the 5G target goals, the concept of Network Function Virtualization (NFV) is introduced. NFV gives the opportunity to mobile operators to modernize their networks by creating a programmable platform, capable of being automated and exposed for operators to extract more value. To make real NFV, operators are using programmable servers in cloud able to run applications locally. One of those applications running on cloud is Network Exposure Function (NEF), a 5G Core Network Function. This thesis work is concentrated on this core network function, presenting its capabilities and features as a future function of new core networks on 5G technology.

This thesis work is developed in TeVA lab (Milano) a joint lab between Vodafone and Nokia. It presents a detailed way of Network Exposure Function feature based on the standardization and the NEF developed on lab environment. The main scope of the thesis is to work on lab environment for research and development purposes, setup and showcase NEF for 5G Vodafone Core Network and also enabling a number of 5G use cases for NEF.

The last part of the thesis is concentrated on the development of a new application able to exploit NEF exposure capabilities and test NEF features at the same time. This application can be used for testing purposes on the lab but at the same time is a good idea for marketing.

Contents

List of Figures

List of Acronyms

Abbreviation	Acronym
SBA	Service Based Architecture
NEF	Network Exposure Function
HSS	Home Subscriber Server
TAS	Telephony application Server
SDL	Shared Data layer
CMS	Container Management System
IMS	IP Multimedia Subsystem
DMZ	Demilitarized zone
CSS	Callight Server
TLS	Transport Layer Security
Auth	Authentication
LTE	Long Term Evolution
VoLTE	Voice over Long Term Evolution
eMBB	Enhanced Mobile Broadband
EPC	Evolved Packet Core
E-UTRAN	Evolved Universal Terrestrial Radio Acc
CN	Core Network
HSPA	High Speed Packet Access
GSM	Global System for Mobile Communicatio
IoT	Internet of Things
UMTS	Universal Mobile Telecommunications Sy
LTE	Long-Term Evolution
MNO	Mobile Network Operator
MIMO	Multiple Input Multiple Output
MME	Mobile Management Entity
OFMD	Orthogonal Frequency Division Multiple
WiMAX	
Worldwide Interoperability of Microwave Access GPRS	General Packet Radio Service
QoS	Quality of Service

Abbreviation	Acronym
RAN	Radio Access Network
MME	Mobilit Management Entity
eNodeB	Evolved Node B
S GW	Serving gateway
UE	User Equipment
UICC	Universal Integrated Circuit Card
PCRF	Policy Control and Charging Rules Function
UMTS	Universal Mobile Telecommunications System
USIM	Universal Subscriber Identity Module
UTRA	Universal Terrestrial Radio Access
VoA	Vodafone Offer Application
NFV	Network Function Virtualization
VNF	Virtualized Network Function
SCEF	Service Capability Exposure Function
API	Application porgramming interface
CD	Call Direction

Introduction

This thesis work aims to focus on the world of 5G technology and in particular to 5G core network functions, network exposure fucntion. In this thesis I have reported in a detailed way the what we actually know about this new 5G core network function.
This thesis was developed in Teva Lab Milano, a lab that serves for development and implementation of new mobile technologies during an Internship of 8 months. The main goal of the internship was to make teaching and research in relevance with the needs of modern bussines and also to give me opportunity me as a student to put my knowledge gained at Politecnico di Torino to practical use. From the other hand, the company had the benfit of having new ideas for enabling new bussines models from the reports and application developed that I have submitted to the company.
Baisically the main topic treated on this thesis are the introduction to mobile networks technologies and more particulary on VoLTE and 5G technologies, 5G Core Network Functions with the main function treated in this thesis that is network exposure function and the concpet of network function virualiation is introduced as well.
Serveral papers and standartisations has been used in order to have a more detailed overview of network exposure function. Among the chapter NEF features and specifications are being introduced along with benefits of having NEF in mobile operators network.
It is worth to mention the capability that NEF has been presenting during this thesis work in order to expose Vodafones assets like (voice, subscrier data etc) to third party application servrs and also the ability of NEF interaction with other network functions into the core networkis highlighted as hell.

The last part of the thesis talk about the development of my new application server able to interact and exhange assets with Vodafone Core Network and vice versa thanks to NEF. The application is called VodafoneOfferApp and is used for testing reason in Teva lab enviroment. VoA was an example of showing the NEF functionalites and features. A specific description of this app is done in the last chapter.

Chapter 1

Mobile Networks

1.1 Introduction

Global mobile data traffic is growing rapidly. Mobile networks are carrying more and more amount of traffic. We now have a mobile data traffic explosion on our hands, which can end up expanding growth into the next decade. The Cisco visual networking index [16], reports some of the global data traffic projections and growth trends.
Mobile data traffic has grown 17-fold over past 5 years. Mobile networks carried 686 petabytes per month in 2012. Global mobile data traffic reached 11.5 Exabyte per month at the end of 2017, up from 6.7 Exabyte per month at the end of 2016. (One Exabyte is equal to one billion gigabytes) [16].
Apart from traffic growth, the increasing number of connected devices imposes another challenge on the future mobile networks. In the future society, everything and everyone will be interconnected.
The 5G networks will enable growth in many industries and applications like for example connected cars, robots and tactile applications. It also should meet the requirements of three groups of use cases [17]:

- mMTC - Massive Machine Type Communications: A very large number of connected devices with disparate quality of service requirements. The objective o fthis category is to provide a response to the exponential increase in the density of connected objects,

- eMBB - Enhanced Mobile Broadband: ultra high-speed connection indoors and outdoors, with uniform quality of service, on the edges of a cell,

- uRLLC - Ultra-reliable and Low Latency Communications: this use case has stringent requirements for capablities such as latency and packet-loss, to ensure increased reactivity.

The demand for mobile networks is continuously increasing, driven by the need to deliver ultra-high definition video. However, even after this massive explosion of traffic, cellular networks have not backed away from accepting the challenge and presenting new innovations.

1.2 Historical Background

Every decade we have experienced a new generation of cellular network. The latest generation being introduced in 2011 is 4G and following this trend we expect to have the new 5G mobile generation standardised by 2020.
Before presenting 5G, it is worth to mention where it all started and mark where we are now. Herein below there is a roadmap of the evolution of mobile generations:

1G : refers to the first generation of wireless mobile telecommunications . All the wireless telecommunications were voice centric and the radio signals used where analogue.

2G: Migration from 1G to 2G means a movement from analogue system to the digital one. The North America standard IS 54 and European GSM were digital systems using TDMA multiplexing with three time slots in each 30 kHz channel, supporting three calls in the same spectrum. 2G technologies also introduced data services for mobile, starting with Short Message Services (SMS) and picture messages.

2.5G: Introduced high-capacity voice with limited data service. It also implemented packet switching along with the circuit switching in 2G. The code division multiple access (CDMA) system was adopted in America while in Europe we have an upgrade from GSM to GPRS and EDGE systems.

3G: It is an upgrade of 2G and 2.5G GPRS networks, for faster internet speed. It is the first international standard released from ITU, differently from the previous two generations. The new network technology called Universal Mobile Telecommunications System (UMTS) was introduced. This new network comes with a combination of 2G network aspects with some new technology and protocols to deliver a faster data rate.
The radio interface introduced is called W-CDMA (Wideband Code Division Multiple Access) using 5 MHz bandwidth. It operates in both frequency division TDD and FDD.In contrast to 2G, 3G systems have involved from voice-centric to data-centric systems.

4G: The movement from 3G to 4G means a shift from low data rates to high-speed data rates for mobile video. The definition of 4G has changed many times over the years. There are two 4G systems. WiMAX (Worldwide Interoperability for Microwave Access) system using OFDM multiplexing technique , developing from WiFi. The other system is LTE which is similar to WiMAX and the bandwidth that they use is both 20 MHz. The major cellular operators are more used to the

current developed LTE system.

5G: The fifth generation is at an early stage of deployment. It will provide super high-capacity and ultra-high-speed data. Compared with 4G networks, it must be able to provide a user experience data rate 10 times higher and a peak data rate 20 times higher that is currently available. Moving from 4G to 5G means to migrate from a single-discipline system to a multi-discipline one.

Technology & Feature	FIRST GENRATION (1G)	SECOND GENRATION (2G)	THIRD GENRATION (3G)	FOURTH GENRATION (4G)	FIFTH GENRATION (5G)
Start/Deployment	1980 – 1990	1990-2000	2000-2010	Now	Probably Soon(2020)
Data Rate	2kb/s	64kb/s	2 Mb/s	1Gb/s	< 1Gb/s
Technology	Analog Cellular Technology	Digital Cellular Technology	Broad bandwidth CDMA IP Technology	WiMax LTE, Wi-Fi	4G+WWWW
Service	Analog voice service No data service	Digital voice, SMS,MMS, Higher capacity packetized data	Integrated high quality audio, video and data	Dynamic Information access Wearable devices	Dynamic Information access Wearable devices with AI Capabilities
Multiplexing	FDMA	TDMA,CDMA	CDMA	CDMA	CDMA
Standards	MTS,AMTS,IMTS	GSM,GPRS,EDE	IMT-2000 HDSPA, HSUPA	Single unified Standard WiMax, LTE	Single unified Standard
Switching	Circuit	Circuit, Packet	Packet	All Packet	All Packet
Core Network	PSTN	PSTN	Packet N/W	Internet	Internet

Figure 1.1: Mobile Technology Comparison [2].

1.3 LTE and VoLTE technologies

1.3.1 LTE Technology

LTE stands for Long Term Evolution, a standard for wireless broadband communication based on GSM, EDGE, UMTS and HSPA technologies. It is widely known nowadays as Advance 4G or 4G LTE.
The standard of LTE is developed by 3GPP and is specified in its Release 8 document series. The main goal of LTE is to increase the speed of wireless data networks and the capacity of them. A further goal was the design of the network architecture to an IP-based system which compared to 3G architecture, reduces the latency.

Main advantages of LTE technology are set out below:

- **High throughput**: LTE guarantees high throughput due to the fact that high data rates can be achieved in Uplink and Downlink as well.

- **Low Latency**: Time to connect to the network is lowered to few milliseconds and the entrance into power saving states is done very quickly.

- **FDD and TDD**: On the same platform are used both Time Division Duplex and Frequency Division Duplex schema.
- **Better End-user experience**: The reduction of latency to 10 ms gives to user a better experience. Also, the optimization of signaling for connection establishment, mobility procedures and air interface are other factors that indicate the improvement of user experience.
- **Simple Architecture**: The running cost of products is lower due to the simple architecture.
- **Seamless Connection**: LTE is perfectly consistent and coherent to existing networks such as CDMA, WCDMA and GSM.

1.3.2 LTE Network Architecture

The high-level network architecture of LTE is composed of the following three main components [3]:

- The User Equipment (UE).
- The Evolved UMTS Terrestrial Radio Access Network (E-UTRAN).
- The Evolved Packet Core (EPC).

The evolved packet core communicates with packet data networks in the outside world such as the internet, private corporate networks or the IP multimedia subsystem. The interfaces between the different parts of the system are denoted Uu, S1 and SGi as shown below [3]:

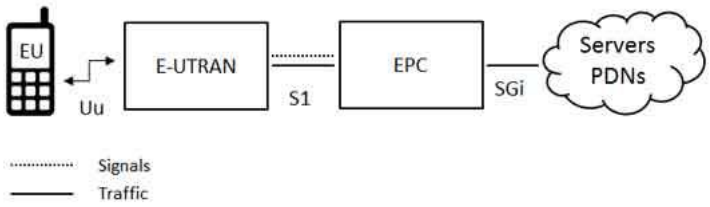

Figure 1.2: High-Level LTE Network Architecture [3].

1.3.3 The Evolved Packet Core (EPC) (The Core Network)

The overall architecture of 4G LTE has been illustrated on the figure below:
The functional elements of the 4G LTE network are [4]:

- **UE**: The 4G LTE device used by the subscriber, such as a smartphone, tablet, laptop or machine-embedded service.

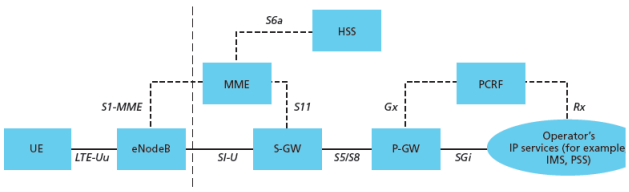

Figure 1.3: Evolved Packet Core [4].

- **Evolved Node B (eNodeB)**: A flat radio network architecture, unlike the layered method of 3G, which therefore simplifies the radio access network operation.

- **Serving Gateway (S-GW)**: All IP packets traverse the SGW, which is the local mobility anchor for bearers when the UE moves between different eNodeBs or hands over to legacy 2G or 3G network access.

- **PDN Gateway (P-GW)**: It provides IP address management, QoS enforcement and flow-based charging according to the policy rules it receives from the Policy Control and Charging Rules Function (PCRF). It provides the mobility anchor point for non3GPP technologies such as CDMA, WiMAX, WiFi and fixed broadband networks. The PGW also connects IP bearers to the PDNs.

- **PDN**: The PDN includes IP networks connected to the 4G LTE, such as for advanced communication services, content delivery networks and the Internet.

- **Moblity Management Entity (MME)**: The MME processes the signaling between the UE and the core network. Its roles include bearer management (establishment, maintenance and release) and connection management (establishing the data connection between the UE and the network).

- **PCRF**:The PCRF is responsible for policy control decision-making and for controlling flow-based charging. It instructs the network about enforcement of QoS policies based on information it receives from the subscriber policy repository, typically in the Home Subscriber Server (HSS) or PDNs (such as VoLTE or video bearer identification sent to the PCRF through the Rx interface).

- **HSS**: The HSS contains the UEs information, including QoS profiles, identification, authorization and provisioned services.

1.3.4 VoLTE Technology

Voice over LTE is a GSMA standard technology that manages to deliver services like voice and SMS over the Packet Switched network of LTE by taking advantages of the core network IP Multimedia Sub-System (IMS). This technology guarantees to mobile operators the interconnection between their LTE network and different type of devices connected to it.
As an all-IP method, VoLTE aims to provide the best user experience in terms of reliability, performance, roaming and interoperability.
On LTE networks the concept of bears is used to route the IP traffic from UT to PDN. Two dedicated bearers are used for the voice service: SIP signalling when the user register on the network and VoLTE bear which is established during the call.

1.3.5 VoLTE Architecture

The VoLTE logical architecture is based on the 3GPP defined architecture and principles for VoLTE UE, Long Term Evolution (LTE), Evolved Packet Core network (EPC), and the IMS Core Network. It consists of the following [5]:

- **VoLTE UE**:The VoLTE UE contains functionality to access the LTE RAN and the EPC to allow mobile broadband connectivity. An embedded IMS stack and VoLTE IMS application are required to access VoLTE services.
- **Radio Access Network**: The Evolved Universal Terrestrial Radio Access Network (E-UTRAN); this is often referred to as Long Term Evolution (LTE). LTE radio capabilities for FDD LTE only, TDD LTE only, or both FDD and TDD LTE are applicable for VoLTE.
- **Core Network**: The Evolved Packet Core (EPC).
- **IMS Core Network**: The IMS Core Network within the VoLTE architecture provides the service layer for providing Multimedia Telephony.
  The VoLTE logical architecture, including roaming and interconnect, is shown in Figure below:

1.3.6 LTE and NFV

Network Function Virtualization gives the opportunity to mobile operators to move the LTE core network to the cloud. More specifically, to replace the carrier grade LTE core network functions with software running in the cloud data centres.
In this way, network operators take advantages of scalability, dynamic load balancing and resources elasticity that the cloud offers.

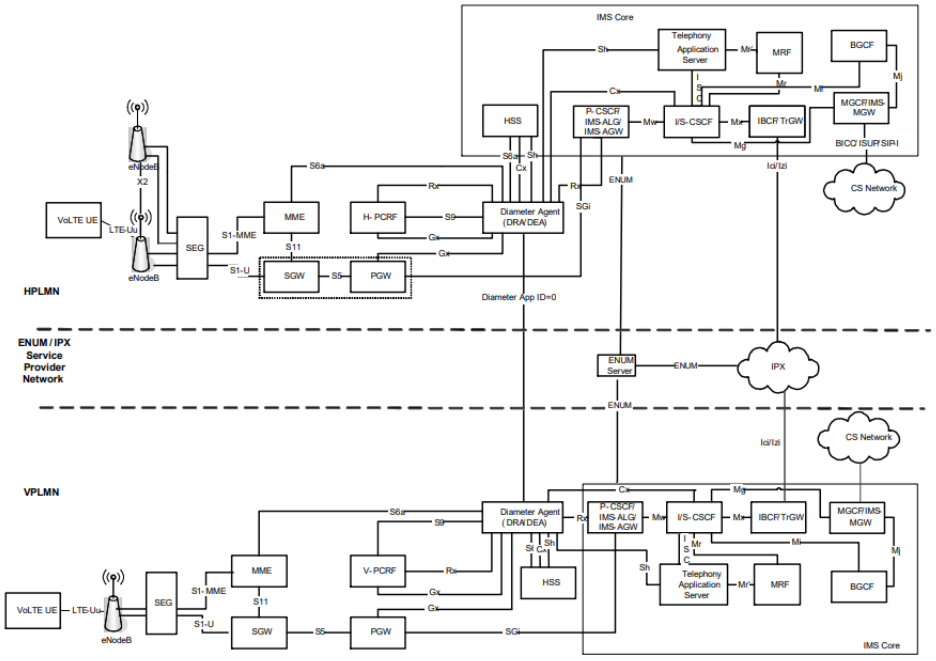

Figure 1.4: VoLTE Logical Architecture [5].

LTE NFV architecture virtualizes LTE core network functions (NF) over the cloud by moving them from the specific hardwares. The EPC network functions, virtualized on cloud handle control-plane and data-plane through separate protocols and network interfaces. For scalability and flexibility purposes, those functions are hosted on separate virtual machines (VMs).

LTE network operators want that the appropriate EPC VNFs are selected to serve their subscribers according to device geographical area (known as tracking area in LTE), and type of radio network (macro/micro base station) it uses. To achieve this, network operators configure a number of EPC VNFs and create a pool of these VNFs. The best available VNFs closer to the device and not heavily loaded are selected to serve the subscriber device during its registration procedure with LTE network.
This VNFs selection can be achieved either through stateful load balancer or through LTE standardized procedure. In the first approach, stateful load balancer sends a query to VNF pool database and gets the IP addresses of MME, SGW and PGW VNFs to serve the subscriber. This is a standard cloud based approach implemented in today public clouds [6].

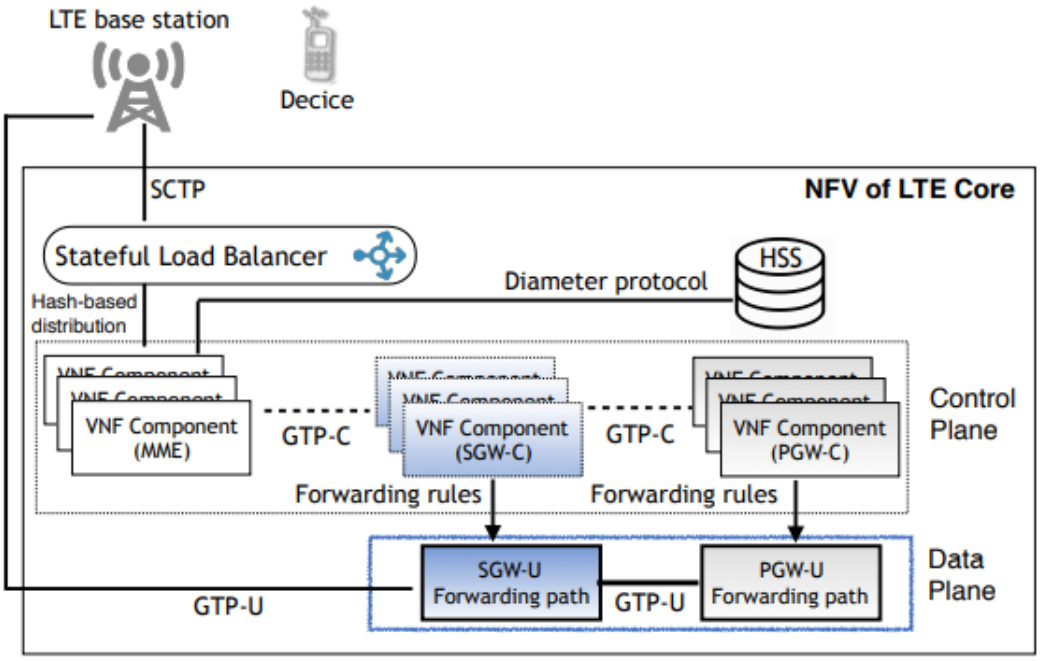

Figure 1.5: LTE over NFV [6].

1.4 5G Technology

The fifth generation networks 5G is at the moment under development and it is supposed to hit the market by 2020. Compared to the 4G LTE technology, 5G aims to reach high speed, lower power and low latency.

1.4.1 Why we need 5G?

The main reason that triggered the need of 5G are set out below [7]:

- Mobile data traffic is rising rapidly, mostly due to video streaming.
- With multiple devices, each user has a growing number of connections.
- Internet of Things will require networks that must handle billions more devices.
- With a growing number of mobiles and increased data traffic both mobiles and networks need to increase energy efficiency.
- Network operators are under pressure to reduce operational expenditure, as users get used to flat rate tariffs and don't wish to pay more.

- The mobile communication technology can enable new use cases (e.g. for ultra-low latency or high reliability cases) and new applications for the industry, opening up new revenue streams also for operators.

Figure 1.6: 5G diverse services illustration [7].

Comparison of key capabilities of IMT-Advanced (4th generation) with IMT-2020 (5th generation) according to ITU-R M.2083 [7]:

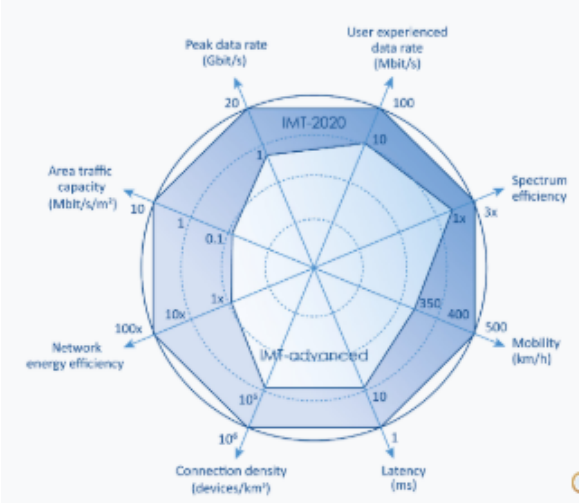

Figure 1.7: 5G diverse services illustration [7].

1.4.2 5G Standard Development

ITU-R has set up a project called IMT-2020 to define the next generation of mobile communication networks for 2020 and beyond with the following time plan [7]:

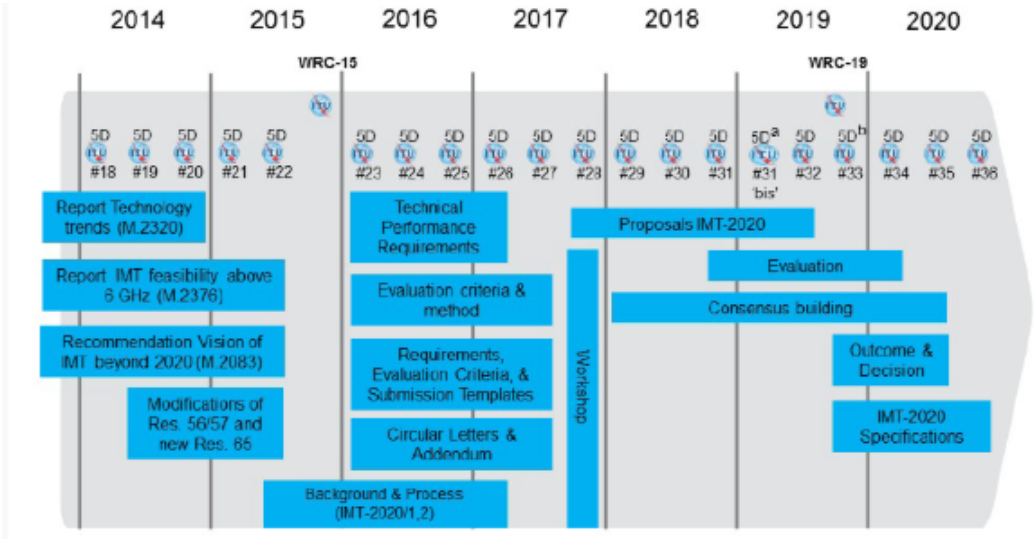

Figure 1.8: 5G diverse services illustration [7].

1.4.3 5G Core Network Architecture

1.4.4 Service Based Architecture

The 5G core is a mesh of interconnected services as shown in the figure below [18].

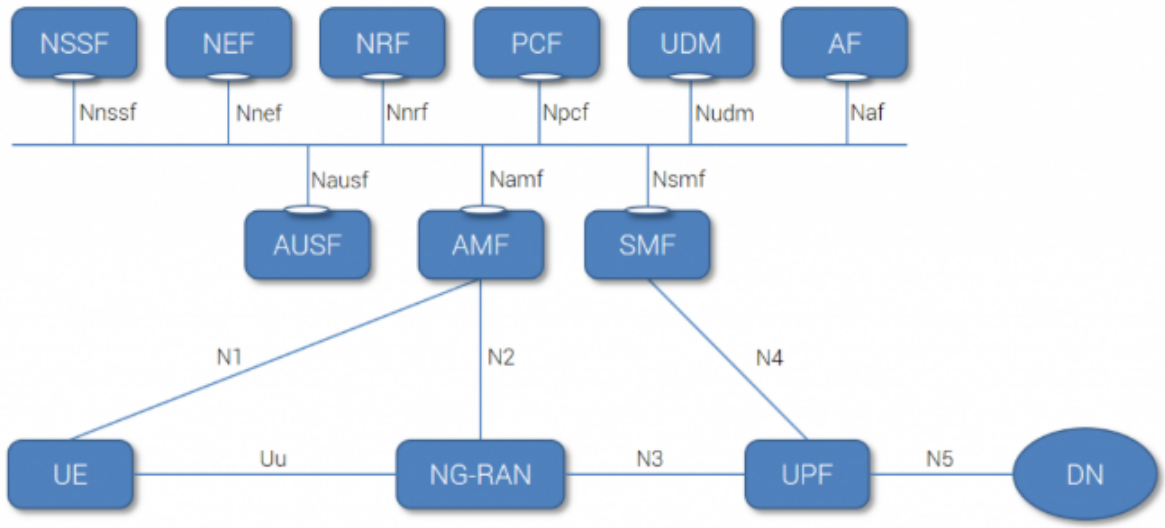

Figure 1.9: 5G Core Network (SBA) [18].

The upper part of the figure (5GC Control Plane), has a bus and service based interface exhibited by individual function. This creates a so called Service Based

Architecture (SBA), in which, one CP network function (e.g. SMF) allows any other authorized NFs to access its services. According to the figure, the NFs within 5GC Control Plane, shall only use service based interfaces for their interactions [18].

1.4.5 Network Functions NFs

Herein below it is the discussion about the Core Network functions and the functionalities that they support [18]:

- Access and Mobility Management function (AMF) supports: Termination of NAS signalling, NAS ciphering and ntegrity protection, registration management, connection management, mobility management, access authentication and authorization, security context management. (AMF has part of the MME functionality from EPC world).

- Session Management function (SMF) supports: session management (session establishment, modification, release), UE IP address allocation and management, DHCP functions, termination of NAS signalling related to session management, DL data notification, traffic steering configuration for UPF for proper traffic routing. (AMF has part of the MME and PGW functionality from EPC world)

- Session Management function (SMF) supports: session management (session establishment, modification, release), UE IP address allocation and management, DHCP functions, termination of NAS signalling related to session management, DL data notification, traffic steering configuration for UPF for proper traffic routing. (AMF has part of the MME and PGW functionality from EPC world)

- User plane function (UPF) supports: packet routing and forwarding, packet inspection, QoS handling, acts as external PDU session point of interconnect to Data Network (DN), and is an anchor point for intra and inter-RAT mobility. (UPF has part of the SGW and PGW functionality from EPC world)

- Policy Control Function (PCF) supports: unified policy framework, providing policy rules to CP functions, access subscription information for policy decisions in UDR.

- Authentication Server Function (AUSF) acts as an authentication server.

- Unified Data Management (UDM) supports: generation of Authentication and Key Agreement (AKA) credentials, user identification handling, access authorization, subscription management.

- Application Function (AF) supports: application influence on traffic routing, accessing NEF, interaction with policy framework for policy control.
- Network Exposure function (NEF) supports: exposure of capabilities and events, secure provision of information from external application to 3GPP network, translation of internal and external information.
- NF Repository function (NRF) supports: service discovery function, maintains NF profile and available NF instances.
- Network Slice Selection Function (NSSF) supports: selecting of the Network Slice instances to serve the UE, determining the allowed NSSAI, determining the AMF set to be used to serve the UE.

Chapter 2

Teva Lab

2.1 Introduction

Teva is the lab environment where this thesis porject is being developed. It is a joint test between Vodafone and Nokia and is mainly devoted for experimental research and development purposes.
Teva supports voice calls over 2G, 3G, 4G and VoLTE technologies as per implementation design, preparing in this way the environment for 5G network deployment.

2.2 TeVA envirnomnet and target goals

The primary goal in the lab was to physically implement the Network Exposure Function between the web servers and Vodafones network (TAS) (highlighted on the figure below). NEF will allow Vodafones Network to deliver an improved quality of experience for the internet application and will also securely expose Vodafones call management API to application developers.

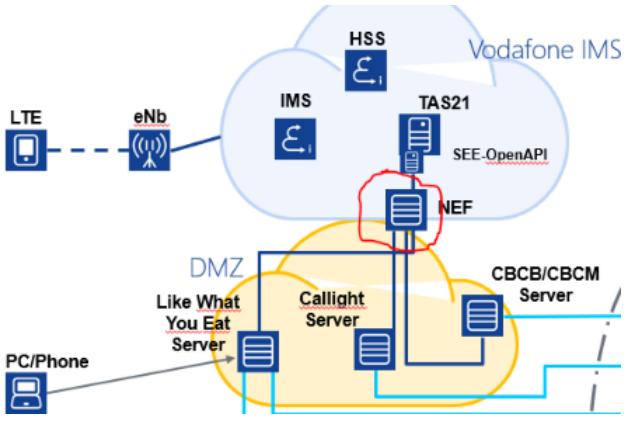

Figure 2.1: Network Exposure Functionn [8].

2.2.1 DMZ zone

In the figure below are represented web servers for each use case scenario that are already implement on the lab and are going to be explained below. All web servers are located on the cloud and everything is virtualized. Actually, what we really have for each server is a virtual machine having some good Web Servers capabilities on top of it.
On the existing demo for the use cases, we already have that the virtual machines and the rest of Vodafones network, on the same cloud. Our aim is to create a DMZ zone for web servers.

DMZ zone is a physical and or logical subnetwork that contains and exposes the Vodafones external services to an untrusted network, usually to a larger network such as the Internet. The purpose of a DMZ is to add an additional layer of security to Vodafones network: external world can access only what is exposed in the DMZ, while the rest of the Vodafones network is "firewalled" [19].

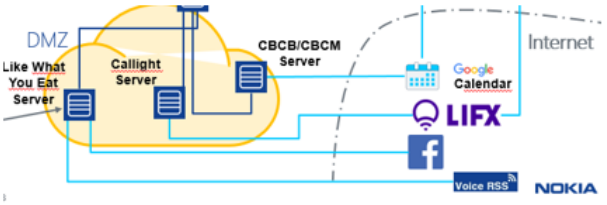

Figure 2.2: Network Exposure Functionn [8].

To put it more concretely, external world of the Vodafones network in the picture is the place where fb, google calendar, lifx are located. The external word (internet) does not see any API but just Vodafone Servers capabilities retrieving information or exchanging information.
Also, to be mention is the fact that everything that is happing on the virtual machines is hidden from the google calendar, fb or lifx (external word).
To summarize, for security reasons the purpose of this DMZ zone is to move out from the Vodafones private cloud, the cloud of virtual machines. DMZ cloud, still can run on Vodafones network but not on the IMS network or cloud, but somewhere else separately.

2.2.2 NEF USE Cases

Several new use cases for the NEF are proposed on lab to work with:

- TLS Termination, Certificate Management

- AutheN / AuthZ (application validation)
- Transaction logging, audits and alerting
- Analytics
- Monitoring
- Transformation / Inspection
- Call logging
- SLA / Threat Protection

2.2.3 Callight Use Case

By exploiting NEF capabilities the following use case is already implemented on Teva lab. This use case implies the interaction of 5G core network with external web servers for IoT. Herein below all the details about this use case. Smart home lights can blink,change color, turn on or off, etc. when receiving a phone call or text message. Different effects can be applied for different callers as shown on the figure below [8].

Figure 2.3: Callight Colors [8].

- User A registers for the service.
- In the Callight web app, User A can associate MSISDN numbers with certain RGB colors.
- User B is calling User A.
- Nokia TAS sends a Call Event Notification to Callight application server and it sends a notification to the Lifx Smart Light with the same color data User A was specifying to User Bs number.
- Lifix Smart Light starts to blink in color.

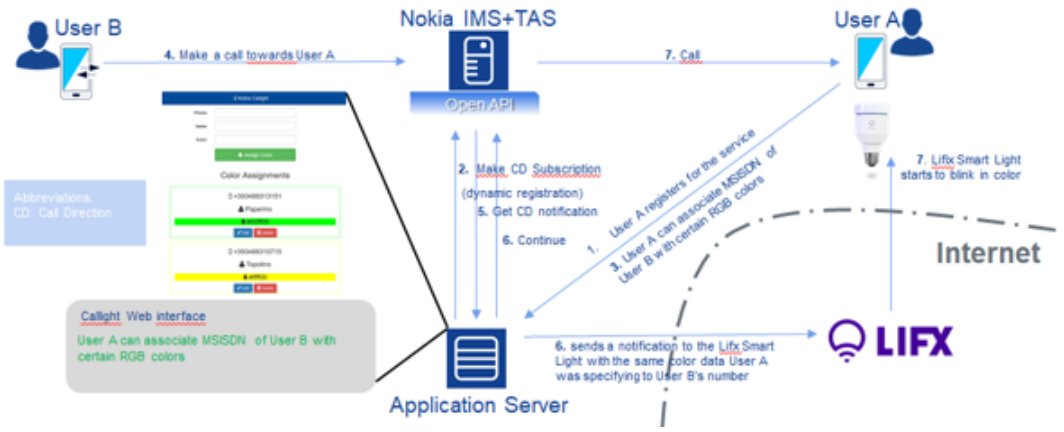

Figure 2.4: Callight Use Case [8].

2.2.4 Call Direction(CD) API

- **Usage**: Take control of a call and make intelligent decisions on call disposition based on other mash-up data [8].
- **Description**: An application can subscribe to one or more valid call events for an IMS subscriber. When a call direction subscription filter is matched for an incoming or outgoing call, the Telephony Application Server (Nokia TAS) suspends its call and sends the matched event via POST to the specified destination URL (of 3rd Party Application Server). The Nokia TAS waits for response to determine the call disposition treatment.
- **Events**: CalledNumber, NoAnswer, Busy, Disconnected, NotReachable
- **Why use it?**: especially useful when mashing up data from other sources so as to change the calling pattern/experience
- **Examples**: Block all calls to child during school time and play a message to the caller.
  Play user my last Twitter post while I answer the call.
  Mashup connected car data to change the calling pattern while driving

2.2.5 Container Managment System

Container Management System is a management platform which allows you to install on top of it different collections of microservices. This management system creates the opportunity to deploy the Network Exposure Fucntion (NEF) which is a collection of microservices, so different piece of software that makes available the

network element NEF. In principal NCMS is a delivery for NOKIA that deals with any other microservices.

This container system is composed by different nodes which practically are just virtual machines. As shown on the picture above we have the Deployment Node and the Control Node as part of O&M network. The deployment node is the machine responsible for creating and installing all the other nodes on the system while for controlling operations we have the Control Node. The nodes that are part of O&M control the Internal Network that is made by Worker nodes and Edge Nodes. The worker node is the real processing machine where the application runs while the Edge node is more front-end, load balancer node and is the only machine that has external interface.

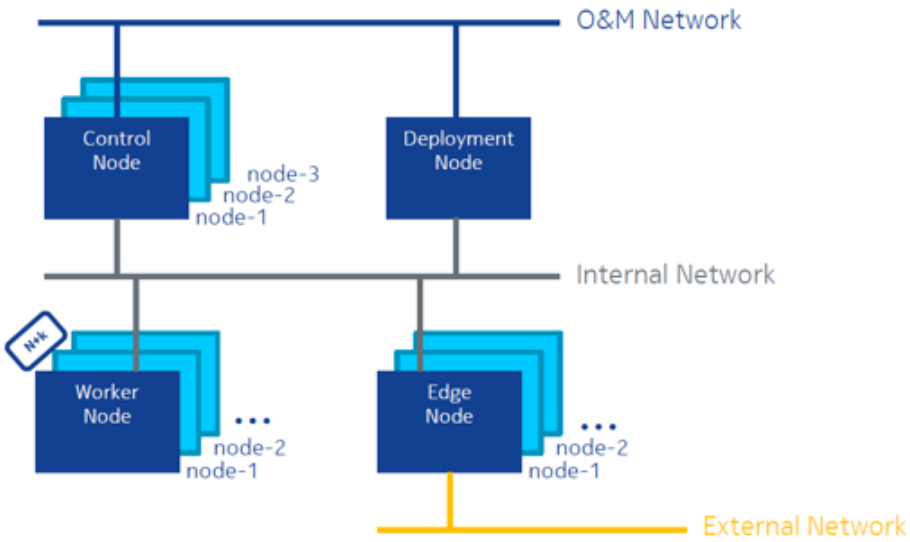

Figure 2.5: Container Managment System [8].

2.3 Virtualized NEF

2.3.1 A Telecom Application

:In the telecommunication world is a single or multiple node application responsible for a well-defined task in the telecommunication network. A Telecom Application uses standardized interfaces to connect to other network elements and implements standardized functions. On top of the standardized functions a telecom application can have vendor specific functionality and can use vendor specific interfaces. One Telecom Application can integrate one or more standard functions [10]

2.3.2 Microservices

:Solve the challenges of monolithic systems by being as modular as possible. In the simplest form, they help build an application as a suite of small services, each running in its own process and are independently deployable. These services may be written in different programming languages and may use different data storage techniques. While this results in the development of systems that are scalable and flexible, it needs a dynamic makeover. Microservices are often connecta via APIs, and can leverage many of the same tools and solutions that have grown in the RESTful and web service ecosystem [20].

2.3.3 NFV Specifications

Network functions virtualization (NFV) provides a new way to create, distribute, and operate networking services. It is the process of separating the network functions from hardware appliances so they can run in software on standardized hardware. These functions (such as firewall, deep packet inspection, and intrusion prevention) become virtual network functions [9].

Figure 2.5 illustrates the high-level NFV framework. Three main working domains are identified in NFV [9]:

- Virtualised Network Functions, as software implementation of a network function which is capable of running over NFVI.

- NFV Infrastructure (NFVI), including the physical resources and how these can be virtualised. NFVI supports the execution of the VNFs.

- NFV Management and Orchestration, which covers the orchestration and lifecycle of management of physical and software resources that support the infrastructure virtualisation, and the lifecycle of management of VNFs.

2.3.4 NFV in Teva Lab

Figure 2.6 illustrates the virtualized Network Functions on the LAB environment located into the cloud. To create the cloud is used a HP G9 chassis and on top of that is installed the open stack. The deployment of open stack is based on VMware (ESXi) and KVM hypervisor. OpenStack Compute supports many hypervisors. Most installations use only one hypervisor. However, you can use ComputeFilter and ImagePropertiesFilter to schedule different hypervisors within the same installation.

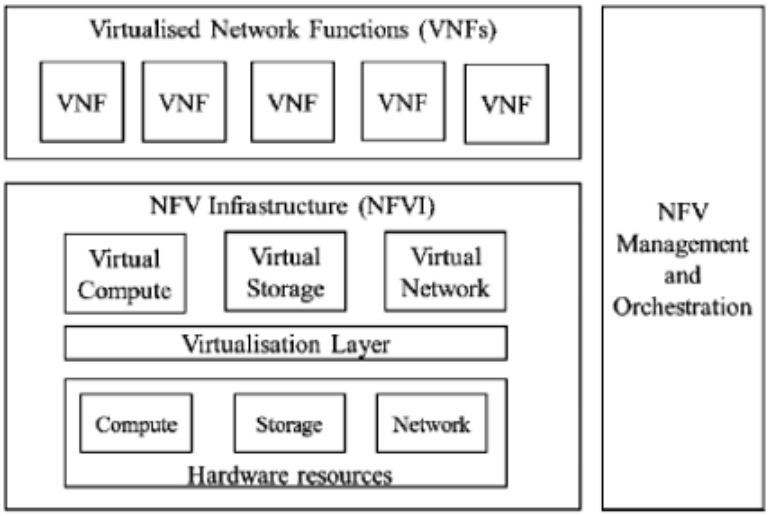

Figure 2.6: High-level NFV framework [9].

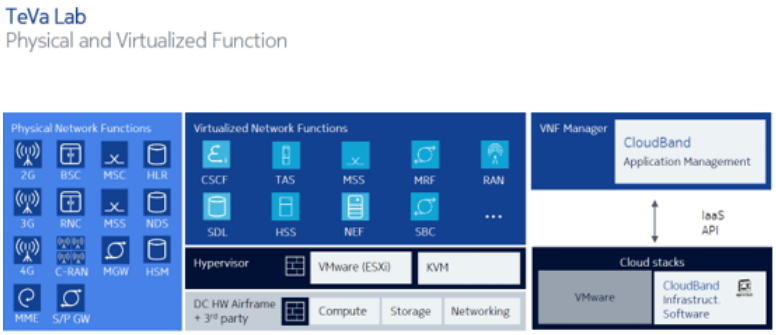

Figure 2.7: NFV in Teva Lab [8].

2.3.5 What is a hypervisor (VMware (ESXi) ?

The hypervisor is a software layer between the host machine (physical server) and guest machines (virtual machines). The hypervisor interacts with the hardware and system resources to provide an interface to share available resources with the guest operating system. Hypervisor is the key to enable the virtualization. It acts as a platform for the VMs to be created on. It manages the sharing of physical resources on the virtual.

2.3.6 KVM hypervisors

KVM (for Kernel-based Virtual Machine) is a full virtualization solution for Linux on x86 hardware containing virtualization extensions (Intel VT or AMD-V). It

consists of a loadable kernel module, kvm.ko, that provides the core virtualization infrastructure and a processor specific module, kvm-intel.ko or kvm-amd.ko. [21] Using KVM, one can run multiple virtual machines running unmodified Linux or Windows images. Each virtual machine has private virtualized hardware: a network card, disk, graphics adapter, etc. [21]

Thanks to hypervisor and openstack technology we can create several virtual machines into a single physical machine. On top of these virtual machines we can run several containers, each of them containing a specific network function application (TAS, NEF, HSS etc).

2.3.7 Orchestrating Containers

In a containerized architecture, the different services that build an application are packaged into separate containers and deployed across a cluster of physical or virtual machines. But this rise the need for container orchestration a tool that automates the deployment, management, scaling, networking, and availability of container based applications [22].

2.3.8 VNF Manager and Orchestration

Network functions virtualization (NFV) defines standards for compute, storage, and networking resources that can be used to build virtualized network functions. The virtual network functions manager (VNFM) is a key component of the NFV management and organization (MANO) architectural framework [9].

The VNFM is a key component of NFV-MANO, the standards of which allow for interoperability of software-defined networking elements that use network function virtualization. The VNFM is responsible for the lifecycle management of VNFs.

VNFM operations include [9]:

- Instantiation of VNFs
- Scaling of VNFs
- Updating and/or upgrading VNFs
- Termination of VNFs

2.3.9 Kubernets for Orchestration in Teva

Kubernetes supervise and take care the life cycle management of containers. It is important to mention that Kubernetes does not take care of the state of the

application. It manages only the container state, in case the application dies restarts the container for example. In case the application state is needed to be considered than that should be handled inside the application. Kubernetes also allows to set affinity and anti- affinity rules, which mean that it is possible to define that containers to be or not to be placed onto one container host. Kubernetes also includes support for Rocket containers.

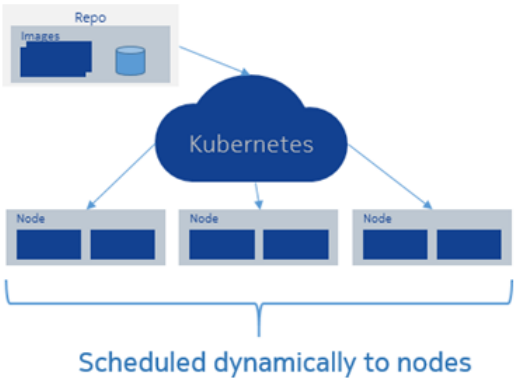

Figure 2.8: Scheduled Dynamically to nodes [10].

2.3.10 Container Technoloy

Containers technology allows large number of applications to run into the same HW environment but also gives performance benefits about starting and stopping the application. Telecom applications requires strong performance and high availability.
Container technology ensure an environment for application isolation. This isolation is important from usage and resource access point of view. This is especially important when two applications are running on the same hardware environment and they are competing for the same resources. This resources can be: CPU time, memory etc.
Hypervisor based offers a great isolation when the application is running on the top of the guest operating system through the host operating system. But more isolation means more overhead. Reducing this overhead is the major advantage of container based technology and allows the application to start much faster 100 times then Hypervised based technology.
Docker is a container tool, it ease container creation but also introduce the notion of container image and image repository. By doing this, container commoditization tools make not just the SW delivery but also the application delivery much faster. (Continuous application delivery is also a benefit of reduced time of SW delivery).
Container based technology offer a great level of flexibility by allowing to containerize either a whole system or just parts of the system together with the system. This

is important from the portability point of view, which means that the container can be lightweight(share the OS machine kernel and therefore do not require an OS per application) to contain only application.

2.3.11 Docker Container

Docker developers realized that not just container creation is difficult but also it is not easy to persistently save a container content like in case of virtual machines. The below figure shows the basic difference between Linux containers (left side) and Docker containers (right side), which is a Linux container created by Docker [10].

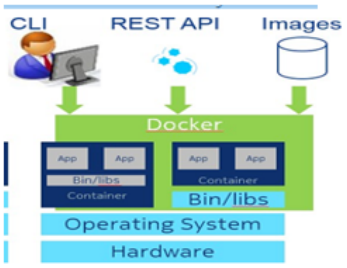

Figure 2.9: Docker Container [10].

As we can observe from the 1 Docker added a command line interface to container creation, a REST API and most importantly an image repository to store modified or preinstalled container images [10].

These images conceptually are very similar to VM images as they store a snapshot of the installed application but the main difference is that when a user modifies this image and wants to store the modified image, then only the modification is stored and not the entire modified image. This is important from that perspective that an image modification can be used quickly not just on the place of modification but pushing it to a central repository is much easier because the total size of the content. committed to repository is much smaller than committing the full modified image. . A Docker container holds everything that is needed for an application in order to run. [10].

Chapter 3

Network Exposure Function

3.1 NEF in 3GPP Specifications

Capability exposure, making 5G Core Network functionalities available to 3rd parties such as service providers and vertical industries, is provided by the Network Exposure Function (NEF). The interface provided by the NEF to 3rd parties can be regarded as one of the essential elements through which 5G communicates more closely towards vertical industries than mobile networks of earlier generations.

3.2 Where did NEF come from?

3.2.1 SCEF

Service Capabilities Exposure Function. It is designed to provide a means to securely expose the services and capabilities provided on 3GPP network interfaces [23].

The need to optimize 4G networks for Internet of things, has resulted in an effort to define enhancements for the MTC, Machine Type Communication. Most of enhancements are provided to radio network but some features are defined also for the Evolved Packet Core. One of MTC enhancement for the EPC is SCEF as a network function that enables small amounts of data to be transported within the signalling messages of the Control Plane. In this way, the traffic generated by IoT devices does not require the establishment of User Plane connection.

The SCEF plays the role of the mediator between the third party and the 3GPP InP facilitating the following operations [11]:

- Authentication/authorization and secure access of third parties to the 3GPP network ensuring that the InP is under control of the exposed services.

- Charging based on offered service and quality provision.
- QoS provision and SLA monitoring, allowing third parties to request and set service priorities in a dynamic manner
- Provision of user context information, e.g., real-time user location, user connection properties, average data rate, etc., and network status changes to third parties
- Admission control regarding predictable communication patterns, e.g. considering the time window and traffic volume, pre-schedule communication timing, etc.

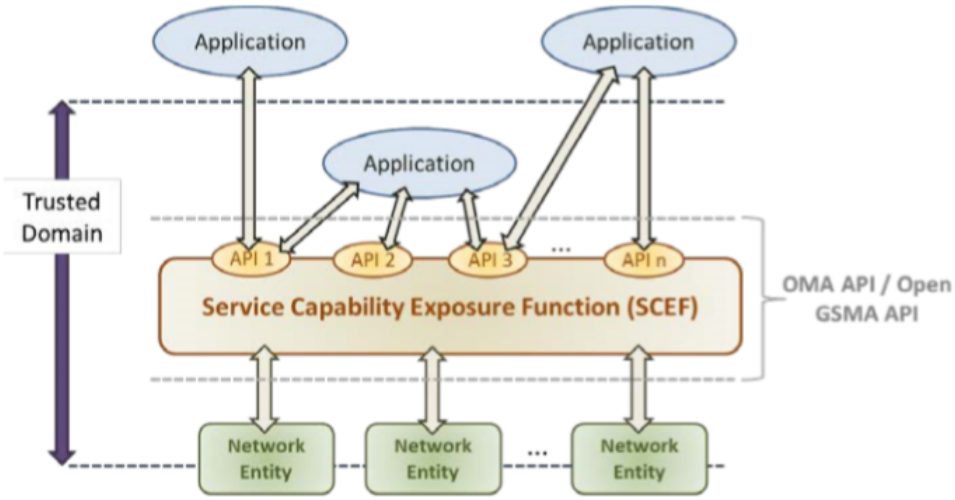

Figure 3.1: Service Capability Exposure Function (SCEF) architecture [11].

3.2.2 NEF and SCEF

NEF: The Network Exposure Function has appeared in the 5G standards as an intelligent, service-aware «border gateway» that will enable the external AFs to communicate with the 5G Network Functions in a secure manner.

So, what do the SCEF (for transporting small amounts of data in the Control Plane signalling messages) and the NEF (to guarantee security at the network border) have in common? Other than the fact they have the term «Exposure» in their names? Well, a lot in fact. The 4G SCEF not only enables Non-IP Data Delivery (NIDD), it also has functions for border security, while the 5G NEF will enable so-called Small Data Delivery on top of its security border functionalities.

Conclusion? Even though SCEF descriptions in the 4G standards and those of the NEF in 5G documents do not seem extremely alike, the NEF in effect is a newer and better SCEF [23].

3.3 NEF Restful Architecture

For the purpose of exposure of functionality to 3rd parties as well as other types of system internal communication 3GPP chose to make use of the widely established REST architecture design paradigm, which describes the design of distributed applications and more specifically of Application Programming Interfaces (APIs).

The APIs offered to 3rd parties, also known as «northbound APIs» are only applicable to a single interface of the 5G system, whilst the NEF is one of man Network functions within the completely redesigned 5G Core Network. 5G service exposure by NEF as decided on 3GPP specifications should be based on RESTfulAPIs as shown in the figure below [12]:

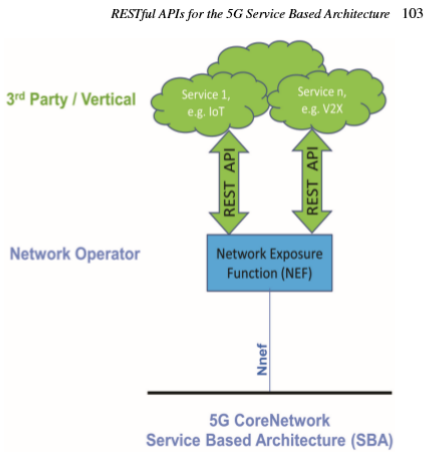

Figure 3.2: RESTful APIs for the 5G Service Based Architecure [12].

3.3.1 NEF Reference Model

The NEF Northbound interface resides between the NEF and the AF as depicted in the figure 1. An AF can get services from multiple NEFs, and NEF can provide service to multiple AFs [13].

The Application Function (AF) may interact with the 3GPP Core Network via the NEF in order to access network capabilities.

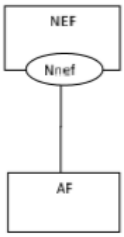

Figure 3.3: Reference Architecture for Nnef Service [13].

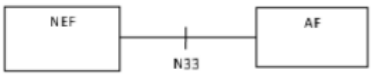

Figure 3.4: SBI representation [13].

3.4 NEF functional elements

The Network Exposure Function (NEF) is a functional element that supports the following functionalities [13]:

- The NEF shall securely expose network capabilities and events provided by 3GPP NFs to AF.
- The NEF shall provide a means for the AF to securely provide information to 3GPP network and may authenticate, authorize and assist in throttling the AF.
- The NEF shall be able to translate the information received from the AF to the one sent to internal 3GPP NFs, and vice versa.
- The NEF shall support to expose information (collected from other 3GPP NFs) to the AF.
- The NEF may support a PFD Function which allows the AF to provision PFD(s) and may store and retrieve PFD(s) in the UDR. The NEF further provisions PFD(s) to the SMF.

3.5 NEF Services

The following NF services are specified for NEF [14]:

Service Name	Description	Reference in TS 23.502 [3]
Nnef_EventExposure	Provides support for event exposure	5.2.6.2
Nnef_PFDManagement	Provides support for PFDs management	5.2.6.3
Nnef_ParameterProvisio n	Provides support to provision information which can be used for the UE in 5GS	5.2.6.4
Nnef_Trigger	Provides support for device triggering	5.2.6.5
Nnef_BDTPNegotiation	Provides support for negotiation about the transfer policies for the future background data transfer	5.2.6.6
Nnef_ TrafficInfluence	Provide the ability to influence traffic routing.	5.2.6.7.2

Figure 3.5: NF Services provided by NEF [14].

3.5.1 Nnef_EventExposure service

3.5.2 Nnef_EventExposure_Subscribe operation [1]

- **Service operation name**: `Nnef_EventExposure_Subscribe`
- **Description**: the consumer subscribes to receive an event, or if the event is already defined in NEF, then the subscription is updated.
- **Inputes(required)**:(Set of) Event ID(s) as specified in `Npcf_PolicyAuthorization_Notify` service operation, target of event reporting (GPSI or External Group Identifier), Event Reporting Information.
- **Inputs(optional)**: Event Filter, Subscription Correlation ID (in case of modification of the event subscription).
- **Outputs(required)**: When the subscription is accepted: Subscription Correlation ID.

3.5.3 Nnef_PFDManagement service

3.5.4 Nnef_PFDManagement_Fetch service operation [1]

- **Service operation name**:`Nnef_PFDManagement_Fetch`
- **Description**: Provides the PFDs for Application Identifier to the NF Consumer.
- **Inputs(required)**: Application Identifier(s).
- **Outputs(required)**: Application Identifier, PFDs.

3.5.5 Nnef_ParameterProvision service

This service is for allowing external party to provision of information which can be used for the UE in 5GS [1].

3.5.6 `Nnef_ParameterProvision_Update` service operation

- **Service operation name**: `Nnef_ParameterProvision_Update`
- **Description**: the consumer updates the UE related information (e.g., Expected UE Behaviour).
- **Inputs(required)**: GPSI, AF ID, Transaction Reference ID.
- **Inputs(optional)**: Any combination of the Expected UE Behaviour parameters.
- **Outputs(required)**: Operation execution result indication.
- **Outputs(optional)**: Transaction specific parameters, if available.

3.5.7 `Nnef_Trigger` service

- `Nnef_Trigger_Delivery request service operation`: the consumer requests that a trigger be sent to an application on a UE and subscribes to be notified about result of the trigger delivery attempt [1].
- `Nnef_Trigger_DeliveryNotify service operation`: NEF reports the status of the trigger delivery to the consumer (failure or success).

3.5.8 `Nnef_TrafficInfluence` service

- `Nnef_TrafficInfluence_Create operation`: Authorize the request and forward the request for traffic influence [1].
- `Nnef_TrafficInfluence_Update operation`: Authorize the request and forward the request to update the traffic influence.
- `Nnef_TrafficInfluence_Delete operation`: Authorize the request and forward the request to delete(s) request for traffic influence.

3.6 NEF on Lab Environment

TeVA is an open and evolving lab environment that supports 5G technology research and validation, vertical industry product development and testing opportunities. The focus of work on this lab is on 5G, offering support and providing infrastructure for Vodafone and verticals industries to perform all tests and simulations needed for new mobile network generation.

TeVA is exactly the place where NEF as a telecom application in running on cloud. NEF as a test product on lab has its own specifications, trying in this way to approach as much as possible to NEF on 3GPP specifications.

3.6.1 Service capability exposure

Nokia Network Exposure Function provides you with a robust platform for creating new services by consolidating APIs and presenting unified access to the API framework for both your developers and 3rd party developers [15].

- Secure exposure of network services (voice, data connectivity, charging, subscriber data, etc.) towards 3rd party application over APIs.
- Developer environment and SDK for operator and community.
- Service mashup for creating end-to-end offering by combining any of the network assets into your application.
- Integration layer that connects your application to operator's network.

The API framework is realized through a plug-in concept that makes it possible to create complex API-based services. You can add additional APIs over time, according to the needs. For those operators with multi-vendor cores, Network Exposure Function can expose APIs from other vendors network functions, by enabling the use of 3rd party plugins.

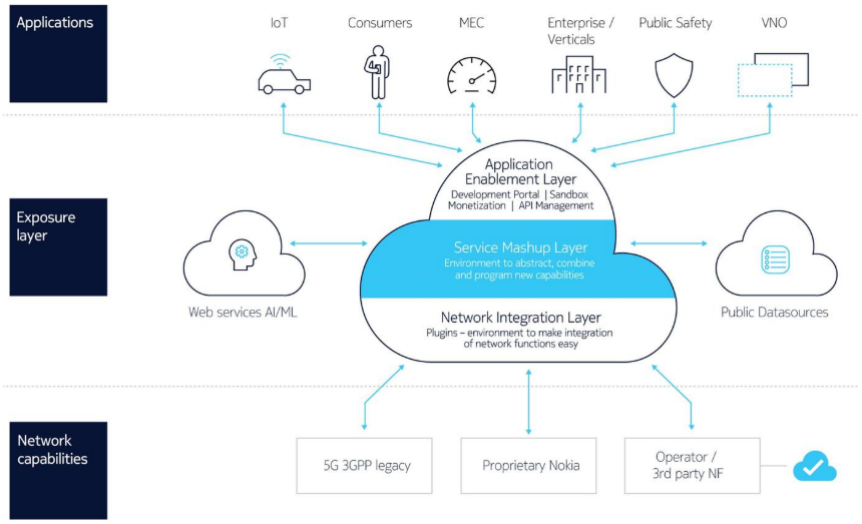

Figure 3.6: Network Exposure [15].

3.6.2 NEF Functionalites Overview

NEF exposes secure, scalable, simplified, and SLA compliant APIs towards external apps seamlessly [8].

- Enables new services and business models.
- Pre-5G app enablement (Call Mgmt. APIs).
- ntegrates NFs/AFs via plugins; providing an extensible platform for custom integration.
- Supports value added apps, e.g. edge computing, specialized application servers.
- Industry leading APIM/API Gateway in partnership with CA Technologies.

3.6.3 NEF Features

Network Exposure function is a key ingredient of 5GC Service Based Architecture which adapts and transforms telecom protocols as RESTful APIs. Some exposure capabilities of NEF are highlighted below [8]:

- 3GPP R15 (and beyond) NEF capability exposure.
  - Monitoring (e.g AMF and UDM events).
  - Provisioning (e.g. UE data)
  - Policy and Charging
  - Core network internal capabilities for analytics.
- Advanced capability exposure
  - Edge computing
  - IoT management
  - On-demand end-to-end (RAN and Core) slice management.
  - Call Management/TAS, and other APIs, e.g. Digital Assistant.
  - API composition and orchestration
- API composition and orchestration

As can be shown on the fig. below NEF has some important elements. It is worth mentioning the API Gateway and the API portal [8].

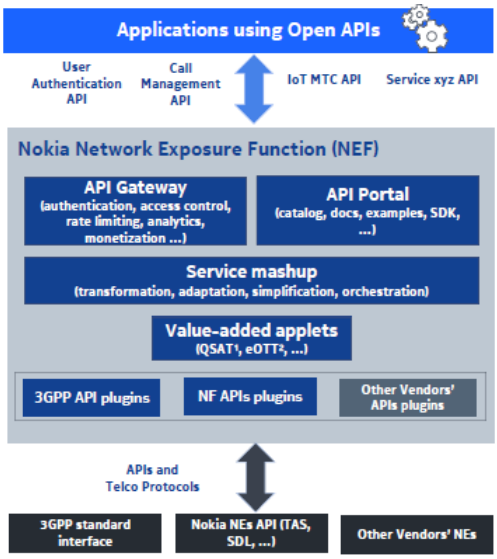

Figure 3.7: NEF Overview [8].

3.6.4 API Gateway

The API gateway is an important element of NEF. The main role of it is to enforce policies and access control between the external parties and the APIs [8].

As the entrance of NEF all request would go through api gateway to specific service.

The figure below demonstrates an end-to-end APP-to-API flow example passing through API Gateway.

The main policies like dynamic routing, SLA enforcment, transformation, threat proteaction etc. and also the access control part can be seen briefly on the figure.

3.6.5 API Portal

The API Portal has an informative role for the providers and 3rd parties. The portal describes what APIs are available for usage, listing them all and providing description for every method [8].

The documentation on the portal should also provide the authentication and authorization mechanism, uses cases that describe the business context and live real implementations.

The portal also describes API lifecycle information, eligibility to be the consumer of API, pricing information, API health insights (real time monitoring) etc.

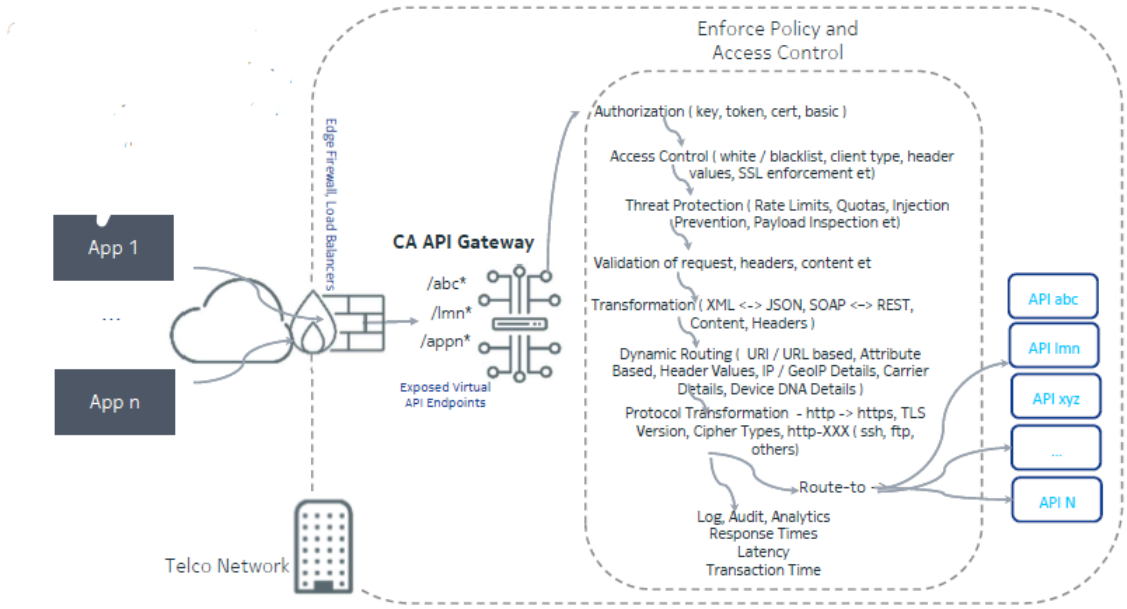

Figure 3.8: Gateway API Managment [8].

All the feature of the API portal are shown briefly on the picture below.

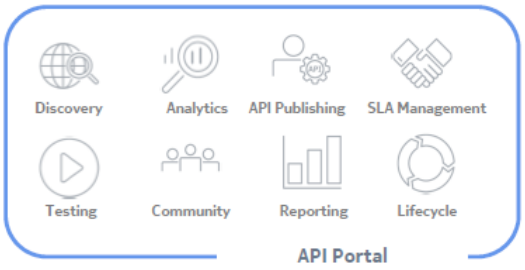

Figure 3.9: API portal features [8].

3.6.6 NEF Release

NEF release used on the lab for this project has the following listed features [8]:

Architecture: Microservices architecture with NEF functions such as API GW and Plugins decomposed into separate Docker Containers.

Api Integration: Secure exposure of Call Management and Roaming/ Devices info APIs via authentication and policy enforcement.

- Call Event Notification API
- Call Direction Notification API
- Call Control API
- User Interaction API
- Device Information API
- Romaing Information API

DEV portal: Developer portal for developers to understand and experiment APIs (Swagger API spec), and develop applications using available SDKs/tools.

OA&M: API diagnostics and usage analytics for Call Management and Roaming/Device Info APIs.

Security: HTTPS/TLS and basic OAuth.

Chapter 4

Deployment Stage (VoA)

4.1 Introduction

Network Exposure Function is based on 5G specifications released by 3GPP standardization, part of 5G Core network function and aims to securely exposes operators assets to application developers. NEF is completely conceptualized and developed in lab environment to expose Vodafones 5G API towards 3rd-party applications and also to enable new service for testing and research purposes.

In order to fully exploit and test NEF exposure capabilities, a new use case is enabled on TeVA lab. It is about the development of a new application (server) that is called Vodafone Offer Application (VoA). All the subscribers of this application will e able to get new services thanks to the ability of NEF to expose Vodafones assets to VoApp. 5G API like CallDirection, Roaming and Device information are exposed by NEF, giving in this way the opportunity to VoA to retrieve information about incoming calls, roaming and devices info about a specific customer.

4.2 VoA Overview

This application is created on python programming language for testing purposes. The idea behind it is explained below:

VOA is an application that is created for all Vodafones customers that will be able to get discounts on different products on Vodafone Shops and on different touristic points on the country where they are roaming.
Every customer that has chosen Vodafone as its mobile operator has the right to download this application and subscribe on it by simply inserting his phone number (msisdn). When the application takes the phone number as an input by the user, it is able to send http/https post requests to two different interfaces (APIs) based

on this phone number.
The first request consists on getting information about the subscriber if it is in roaming or not and if yes, in which country it is. The second request consists on getting information about the specific device that the subscriber is using.

All the requests are towards the HSS (Home Subscriber Server) server that is the function which keeps user profile data like roaming information, device information etc. Note that all the requests pass through the NEF (Network Exposure Function). All requests that come from different applications towards HSS or other servers (ex. TAS) are forwarded only by NEF which is the only network function of the Vodafone Cloud that has communication with the external trusted applications.

Furthermore, VoApp offers to its subscribers the option "Do-not-disturb" to reject/accept a phone call through the App while they are in roaming outside Italy. This option is made possible through the capability of NEF to expose CallDirectionAPI from TAS to VoApp. More information about this option of the App will be provided later on this chapter.

A better view of working lab environment is shown on the figure below:

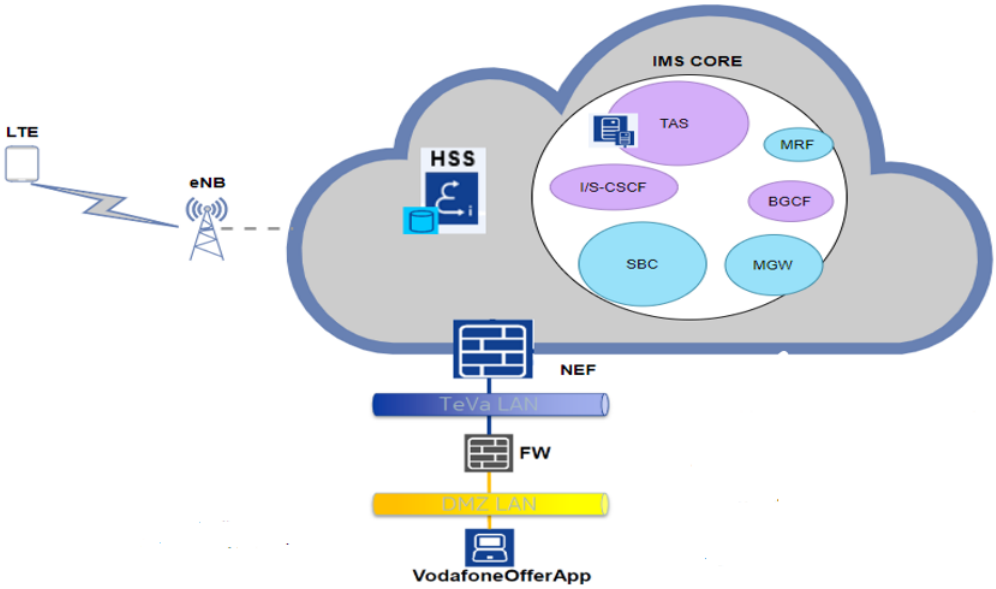

Figure 4.1: Lab environment - VoA.

4.3 NEF Use Cases

4.3.1 NEF-VoA: HTTP communication

The first use case of the application is done by using the HTTP protocol for communication between the client (VOA app) and in this case the NEF. Fig.2 shows the schematic view of the use case:

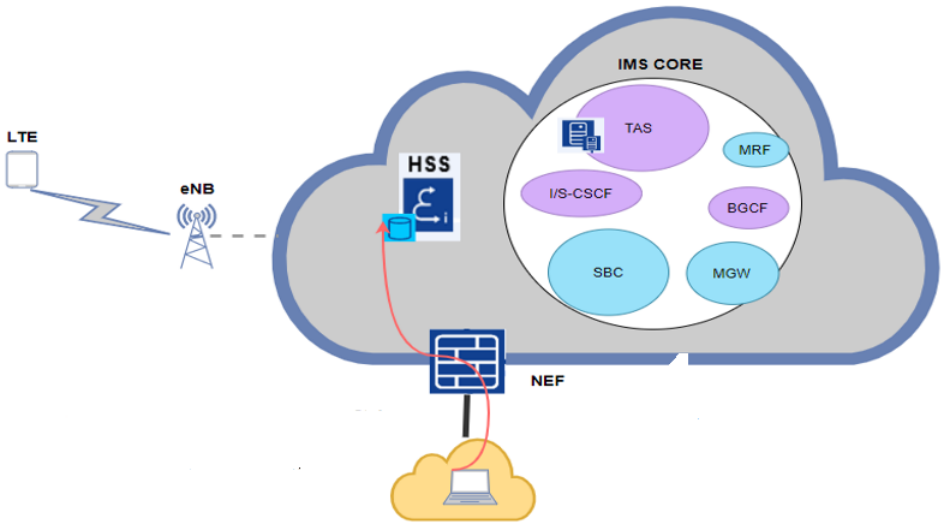

Figure 4.2: HTTP POST Request.

In this test the customer that has subscribed on the application has the following phone number (msisdn): 393488310996. This user is roaming in Germany and is using a phone model Nokia 8.

On the figure below we have the trace of all steps of communications between VOA app and NEF. The trace is obtained by capturing traffic on the NEF interface using WireShark.

1. VOA sends an http post request towards NEF (path: /graphql) for getting roaming information. This request is shown on the figure below:

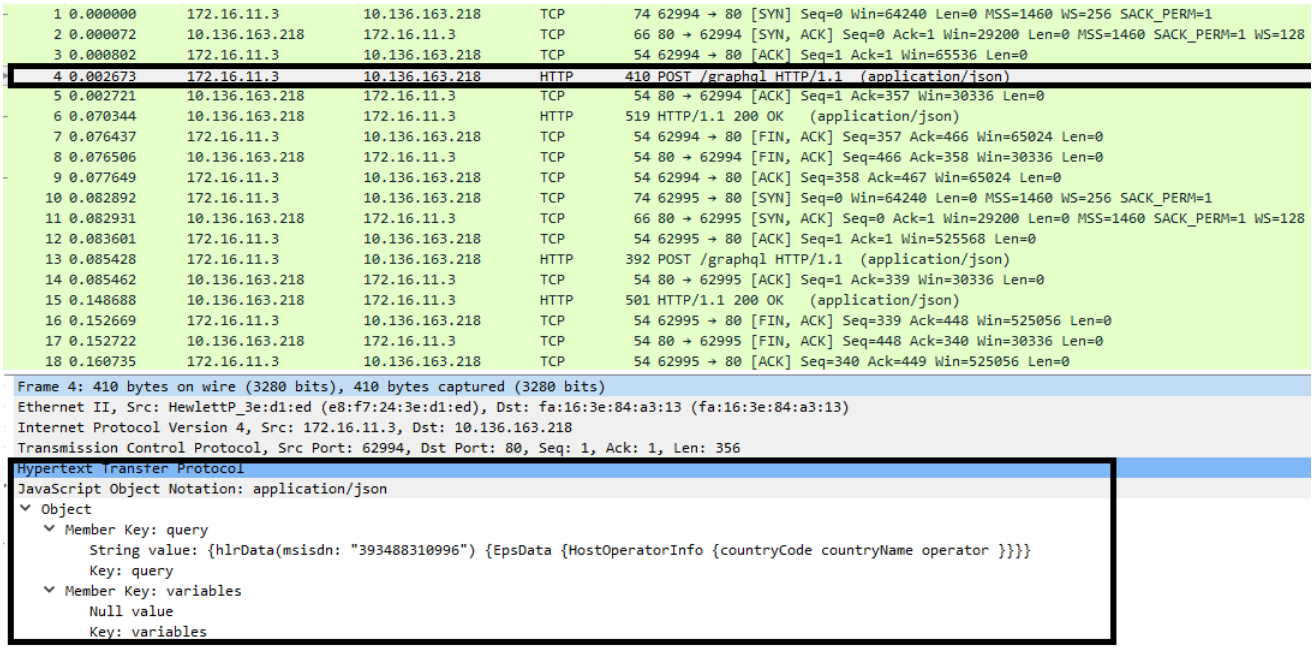

Figure 4.3: HTTP POST Request.

2. NEF replies with a response (HTTP/1.1 200 OK) about the roaming request. The response is shown on the figure below:

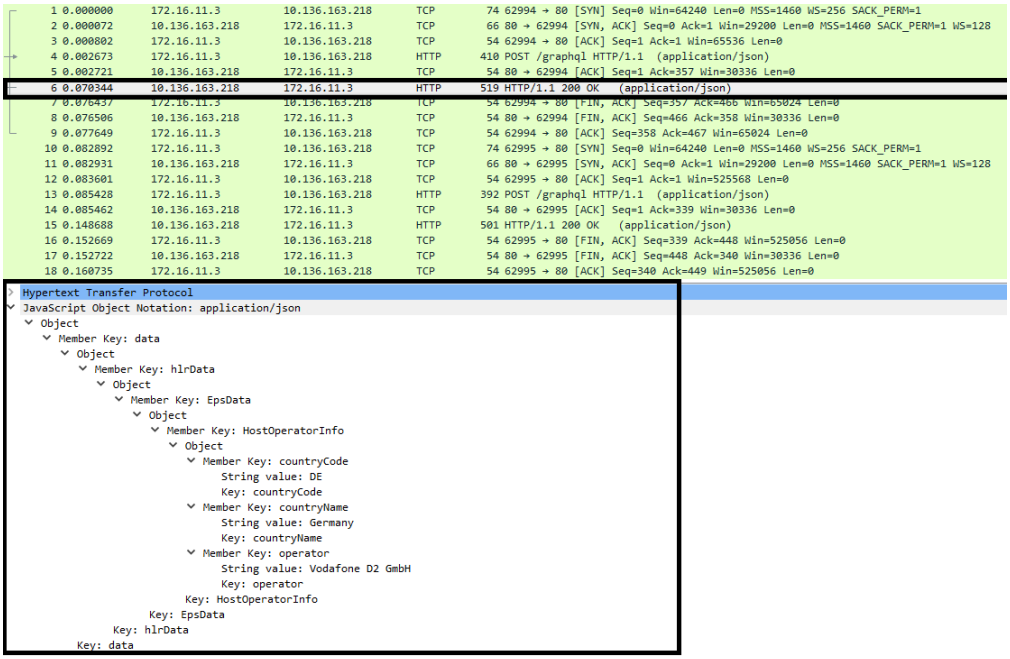

Figure 4.4: NEF response on wireshark demostration.

As can be seen by the response the customer is roaming in Germany.

3. VOA sends the second http post request towards the NEF (path: /graphql) for getting information about the device of the customer. This request in shown on the figure below:

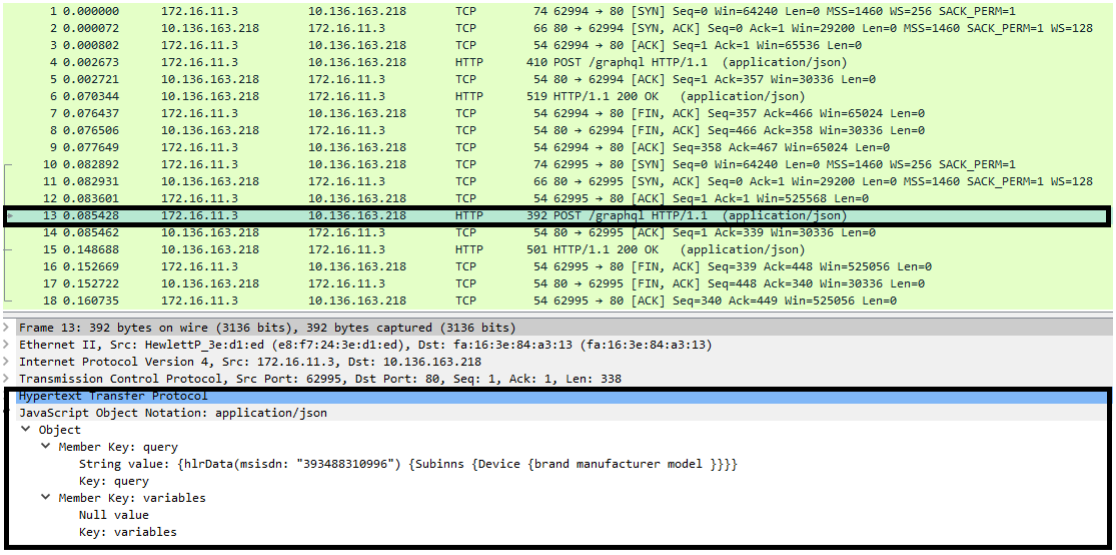

Figure 4.5: Roaming Information Requested.

4. NEF replies with a response (HTTP/1.1 200 OK) about the device information request. The response can is shown on the figure below:

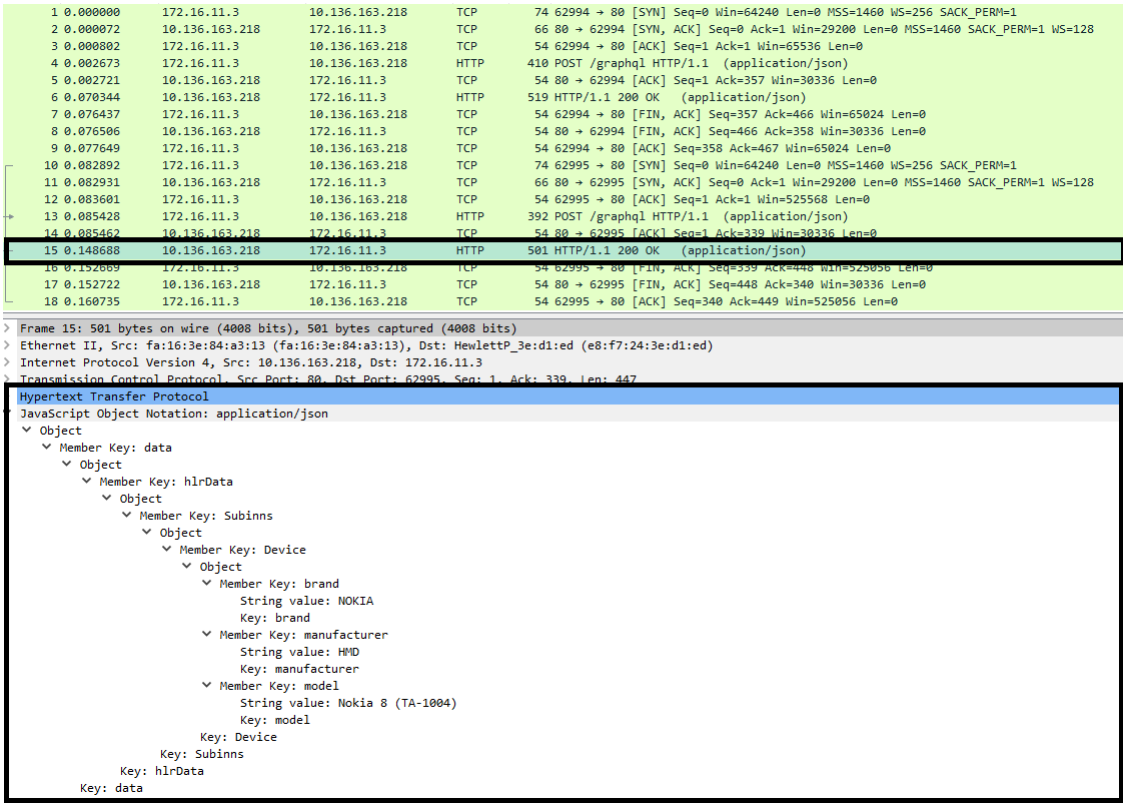

Figure 4.6: Roaming Information Retrieved.

As can be seen the phone model that the customer is using is Nokia 8 (TA-1004).

5. The output that the user see on his application is shown on the figure below:

As can be seen below, based on the information that the application received from the NEF (HSS), it is able to process this information and to come with some offers from the customer.

First offers are referring based on the location that customer is declaring. Note that the offers about the touristic places, museums/restaurants etc are refer to them which have already an agreement with Vodafone to offer discounts for the customer that have already installed VOA application.

```
Enter phone number:     393488310996
------------------------------------------------------------
Requesting roaming information for user with MSISDN: 393488310996.
Host Operator Information:
        Country Code:   DE
        Country Name:   Germany
        Operator:               Vodafone D2 GmbH

USER IS ROAMING OUTSIDE OF ITALY!

Get the best roaming packets offers with VodOfferApp in DE:
Pkt_1Gb
Pkt_2Gb
Pkt_3Gb

List of restaurants in DE that offer discounts with VodOfferAPP:
Rest_X
Rest_Y
Rest_Z
Rest_W

List of museums in DE that offer discounts with VodOfferApp:
Museum_X
Museum_Y
Museum_Z
Museum_W
------------------------------------------------
Requesting device information for user with MSISDN: 393488310996.
Device Information:
        Brand:                  NOKIA
        Manufacturer:   HMD
        Model:                  NOKIA 8 (TA-1004)

Upgrade to NOKIA model 10 and get 20% discount with VodOfferApp.
------------------------------------------------------------
```

Figure 4.7: The output for the customer.

Application Python code Demonstration:

```
url = http://nef51.tscc.lab-mi.vas.omnitel.it/graphql  #http
headers = {'Content-type': 'application/json'}


#Roaming request passed to NEF from the application based on the phone number given as an input.
roaming_req = {
                "query": "{hlrData(msisdn: \"phone_number\") {EpsData {HostOperatorInfo {countryCode countryName operator }}}}",
                "variables": None
                   }            #json format


#Device info request to NEF form the application based on the phone number given as an input.
device_req = {
                "query": "{hlrData(msisdn: \"phone_number\") {Subinns {Device {brand manufacturer model }}}}",
                "variables": None
                   }            #json format
```

Figure 4.8: Roaming and Device request: Application Code.

4.3.2 NEF-HSS: HTTPS Communicatiion

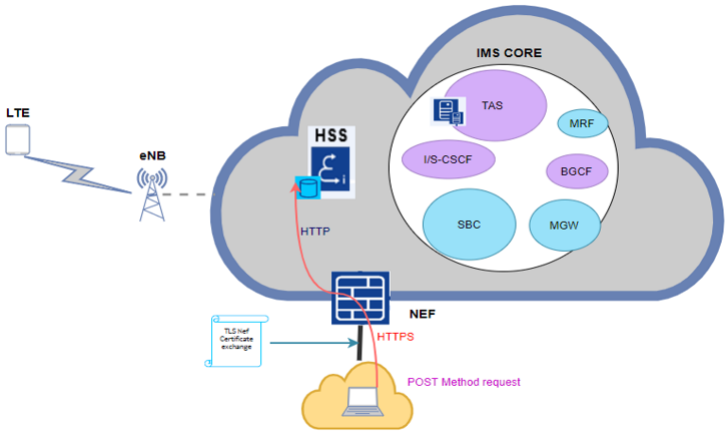

Figure 4.9: TLS Termination.

Note that only NEF and applications communicate in HTTPS. Internally in the cloud, NEF communicate HTTP with other functions.

1. The first test is made without installing the certificate when using the application. Herein below on figure 8 there is a trace captured with Wireshark in order to see what happens:

1	0.000000	172.16.11.3	10.136.163.218	TCP	74 55782 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000083	10.136.163.218	172.16.11.3	TCP	66 443 → 55782 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.000771	172.16.11.3	10.136.163.218	TCP	54 55782 → 443 [ACK] Seq=1 Ack=1 Win=65536 Len=0
4	0.076998	172.16.11.3	10.136.163.218	TLSv1.2	295 Client Hello
5	0.077063	10.136.163.218	172.16.11.3	TCP	54 443 → 55782 [ACK] Seq=1 Ack=242 Win=30336 Len=0
6	0.078518	10.136.163.218	172.16.11.3	TLSv1.2	1436 Server Hello, Certificate, Server Key Exchange, Server Hello Done
7	0.082869	172.16.11.3	10.136.163.218	TLSv1.2	61 Alert (Level: Fatal, Description: Unknown CA)
8	0.083041	10.136.163.218	172.16.11.3	TCP	54 443 → 55782 [FIN, ACK] Seq=1383 Ack=249 Win=30336 Len=0
9	0.083777	172.16.11.3	10.136.163.218	TCP	54 [TCP Previous segment not captured] 55782 → 443 [ACK] Seq=250 Ack=1384 Win=64256 Len=0
10	0.083817	172.16.11.3	10.136.163.218	TCP	54 [TCP Out-Of-Order] 55782 → 443 [FIN, ACK] Seq=249 Ack=1383 Win=64256 Len=0
11	0.083835	10.136.163.218	172.16.11.3	TCP	54 443 → 55782 [ACK] Seq=1384 Ack=250 Win=30336 Len=0
12	0.083853	172.16.11.3	10.136.163.218	TCP	54 55782 → 443 [RST, ACK] Seq=250 Ack=1384 Win=0 Len=0

Figure 4.10: No certificate used.

   As can be seen the application sends a ClientHello message which contains all the information the NEF needs to connect via HSS. Since the application (HSS) does not recognize this certificate because it does not find it installed on the host where application runs, the application refuses the connection. In the figure above, we see the response from the application Unknown CA.

2. The second test is done with the certificate already installed on the host device of the application. Once the connection is established both the Application and NEF use the agreed algorithm and keys to securely send messages to each other. The TLS connection is set up by a handshake on three main phases:

(a) **Hello**: VOA sends ClientHello message which contains all the information the NEF needs to connect via SSL. The NEF responds with a ServerHello, which contains similar information required by the app, including a decision based on the app preference about cipher suit and SSL version.

(b) **Certificate Exchange**: After the connection is established NEF has to prove the identity of the application and this is achieved using a SSL certificate. The certificate contains various data like validity date, owner name, certificates public key etc. The application checks either to trust or not this certificate and if it is also trusted by one of the several Certificate Authorities (CAs).

(c) **Key Exchange**: The encryption of the data is done using a single key algorithm. The application generates a random key and encrypts it during the Hello phase, and the NEFs public key (found on SSL certificate). Then this encrypted key is send to the NEF, where is decrypted using the NEFs private key and in this way the handshake part is completed.

On the figure below are showed from the Wireshark capture all the steps that are described above for the certificate exchange. NOTE that TLS and SSL are the same thing. TLS is more popular acronym used nowadays.

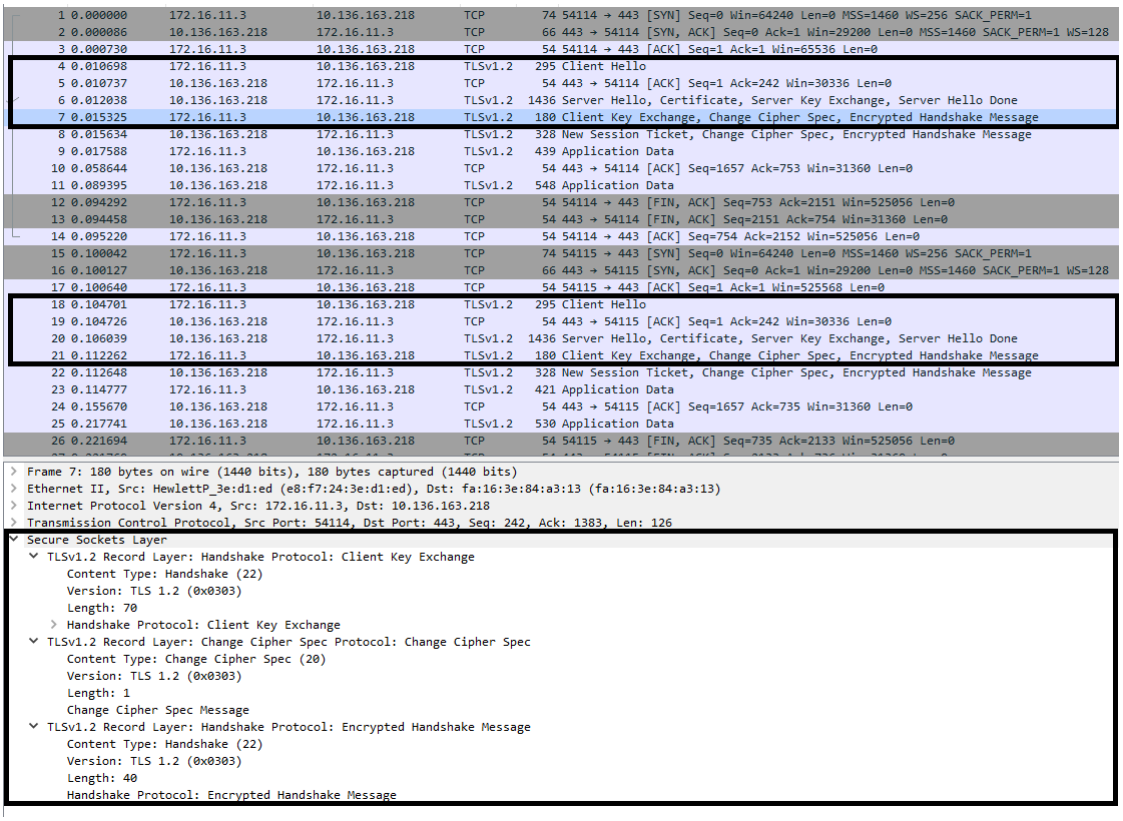

Figure 4.11: HTTPS Connection.

Above are highlighted the certificate exchange between the application and NEF for both requests (roaming and device info) made by the application. Also, more detailed info about the client key exchange for the certificate is highlighted.

3. At this point HTTP requests and response can now be sent between NEF and the application by forming a plaintext message and then encrypting and sending it. Only the application can decrypt the message that comes from NEF. As can be seen clearly even from the Wireshark trace, we cannot see anymore the request and the response from NEF (it is encrypted) and vice versa. In this way we can avoid Man In the Middle Attackers that tries to read and modify any requests that they may intercept. On the figure 10 below we can observe it clearly:

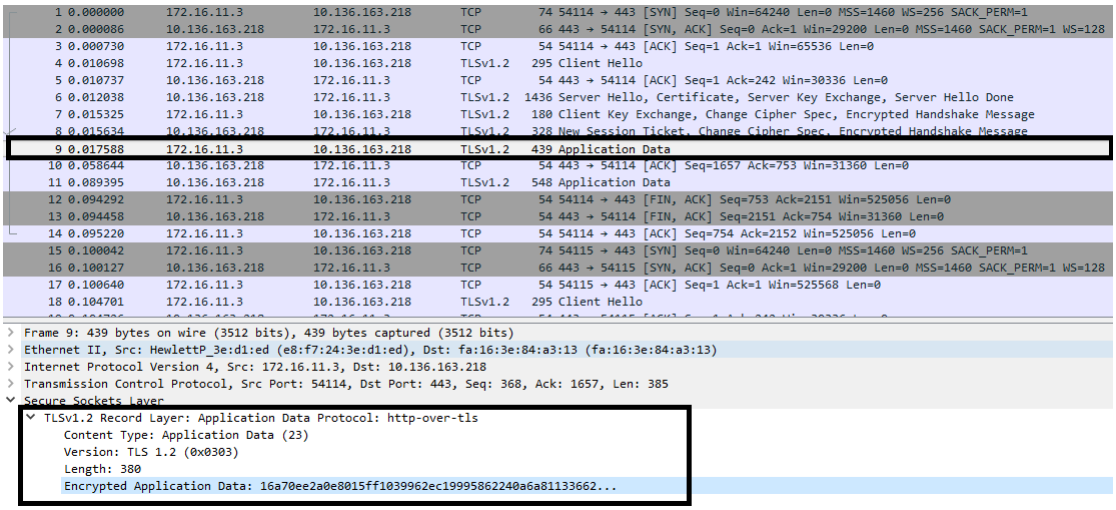

Figure 4.12: Application Data Encrypted.

Application code demonstration

```
url = https://nef51.tscc.lab-mi.vas.omnitel.it/graphql   #HTTPS secure

 # NEF certificate file passed as a parameter in order to make it recognizable from the SSL on application part.
cafile = '/usr/local/share/ca-certificates/Espoo_Root_CA.pem'


 #The string response that we will get form NEF by passing on the request the path, headers and the certificate.
 response = requests.post(url, json=device_req, headers=headers, verify = cafile)
```

Figure 4.13: Certificate relate code.

4.3.3 Basic Authentication

The purpose of this test is to check if the NEF is able to support some authentication feature. During the test is realised that NEF is able to support HTTPS Basic Authentication. In this test NEF has requested authentication information (username and password) from the VOA application. The client passed this information to the NEF in an Authorization header.

The trace captured through Wireshark is shown below but is important to notice that the protocol used is HTTPS for security reasons and the authentication information passed from the application to NEF cannot be seen from Wireshark since this information is encrypted.

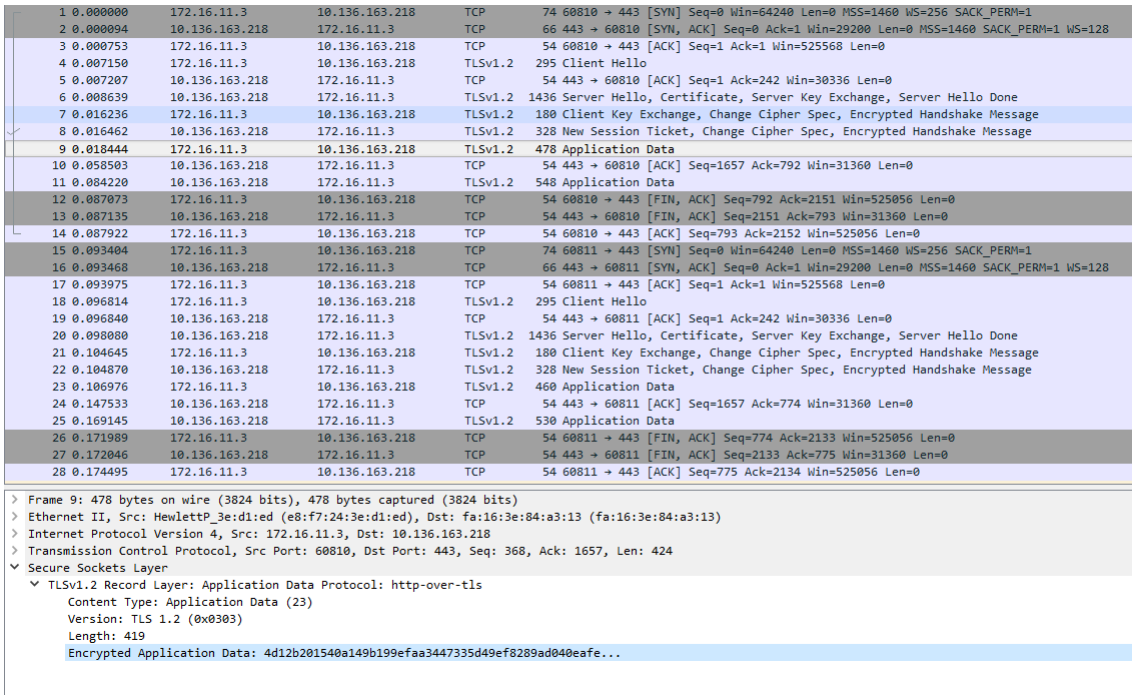

Figure 4.14: Basic authentication.

Application code demonstration

```
 url = https://nef51.tscc.lab-mi.vas.omnitel.it/graphql  #path

cafile = '/usr/local/share/ca-certificates/Espoo_Root_CA.pem' #Certificate file

response = requests.post(url, json=roaming_req, headers=headers, verify = cafile, auth=HTTPBasicAuth('admin', 'admin'))   #request and all the pa

auth=HTTPBasicAuth('admin', 'admin')
```

Figure 4.15: Authentication related code.

Authentication information is passed by the application to NEF as can be seen above from the code of the application. Username: admin and Password: admin.

4.3.4 NEF-HSS Communication

As mentioned even before, the communication between NEF and SDL is made possible through the APIs. Since here we are in a secure zone (inside the Vodafone cloud), the communication NEF-SDL is realised over HTTP.

The API request towards SDL and the response that comes from SDL are not encrypted. The encryption of the data is made by NEF before forwarding them to

the application always in case the communication NEF-App is HTTP.

In the figure below there is a trace capture with Wireshark on the interface of HSS in order to see the internal communication between NEF and HSS.

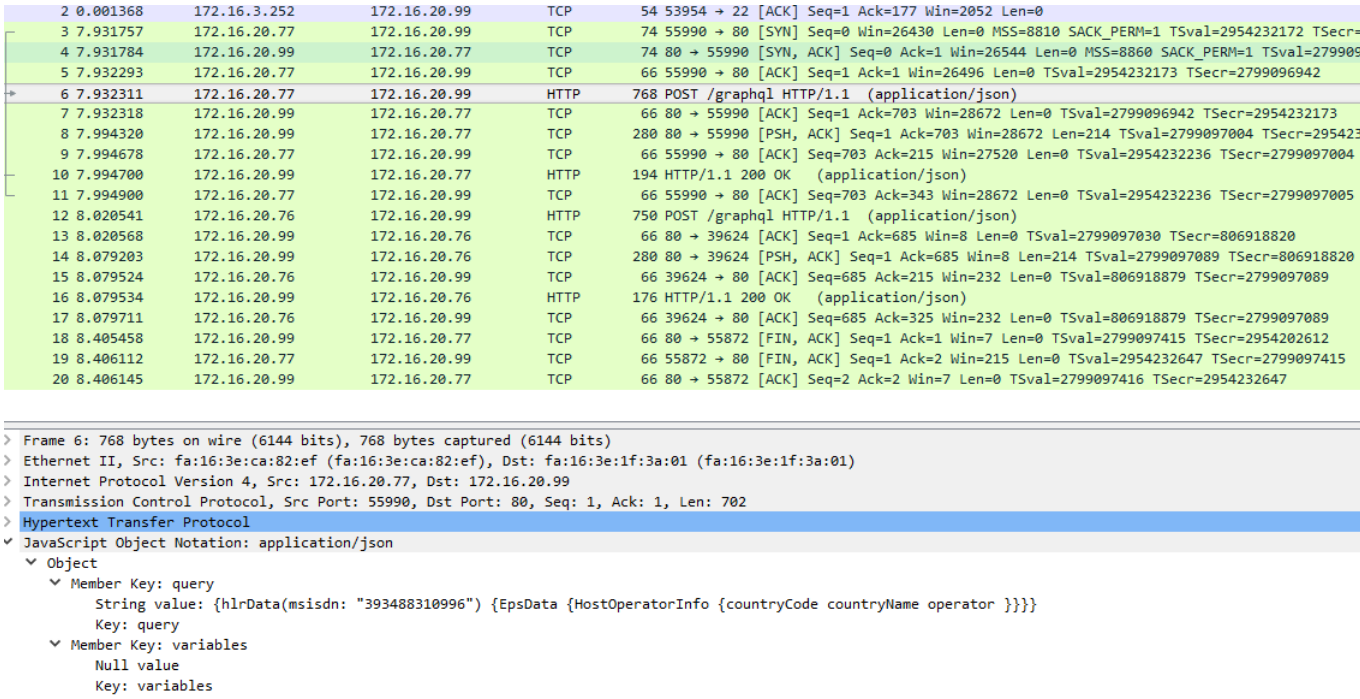

Figure 4.16: NEF-HSS communication.

4.3.5 TAS Scenario

4.3.6 VoA, "Do Not Disturb" option

As described in the beginning VoAapp offer another option "Do-not-disturb". Once the application knows from HSS that you are in roaming, it offers you the opportunity to reject/accept the call through the app before the call is routed to your phone device. This kind of feature is made possible thanks to the capability of NEF to expose callDirectionAPI to VoApp.
The role of TAS (Telephony Application Server) is important in this scenario. TAS get notified about the app subscription and when a filter is matched, it suspends the call until a decision is made by the application.

4.3.7 TAS Framework

TAS is based on OMA Call Notification API specifications. It can handle Subscriptions for Call Direction API and can send notifications about the subscribed

call events. Through Call Direction API third-parties applications can subscribe dynamically to call events. (Called Number, Answer, No Answer, Busy, Not Reachable and Disconnected).

The role of TAS is to notify the AS about such events and to suspend the call processing, allowing in this way the AS to perform any needed operation before taking the decision about the call.

4.3.8 Restful Methodology

Any functionality that is provided by TAS is defined as a resource. Every resource is being identified by an URL. The AS has to issue an HTTP request to that URL in order to access a specific resource. The operation which an AS wants to execute is defined by an HTTP POST, GET, PUT, and DELETE followed with a resource. TAS can also act as a server and send HTTP requests to application servers, e.i when TAS notifies an application that an incoming call has been received for a given subscriber.

4.3.9 VoA subscription to Tas for a call event

4.3.10 Notes about subscirption

Dynamic subscription of VoApp is necessary through TAS. Static subscription is optional via HSS. [8]

- In case of call direction subscription, TAS allows only one address registration at a time.
- The criteria can contain as many call events as subscriber application would like to have.
- Several application servers can register for the same event.
- The order of call event notifications may be the same as the order of event subscriptions (subscription order is kept with best effort).
- TAS supports both calling and called party address in the call direction subscription.
- TASvsupports all of the OMA defined events (CalledNumber, Busy, NoAnswer,Answer, NotReachable and Disconnected). Since there is no indication in event notification whether it is sent for "called" or "calling" event, VoApp server should use different notify urls in event subscription if such information is important

If an event happened with the calling party, and notification was sent about it to application server, then TAS supports the following actions from third party VoA server [8]:

- Route
- Continue
- EndCall
- AS could request an announcment playing for the calling party
- AS could request a digit collection

4.3.11 Subscription to a call event for an IMS user

The following resources are used for the subscription [8]:

- To subscribe to call direction notifications, create a new resource under taseAPI/-callnotification/v1/subscriptions/callDirection with resource identifier taseAPI/-callnotification/v1/subscriptions/callDirection/sub s?id=212..
- To cancel a subscription to call direction notifications, delete the resource created during subscription with the resource id: taseAPI/callnotification/v1/subscriptions/callDi s?id=212..

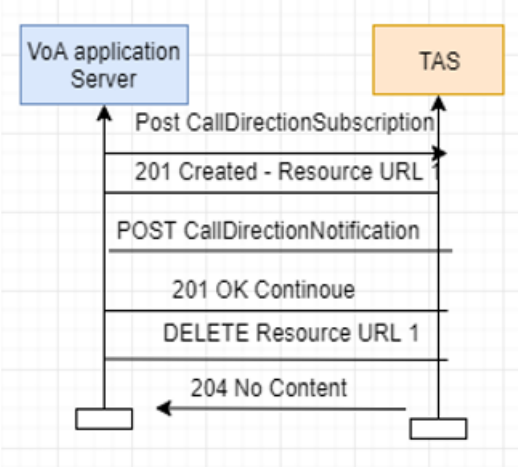

Figure 4.17: Subscribe to call directin notification.

4.3.12 VoA WorkFlow

The following steps outline the flow for VoA:

- User B register for the service
- Make CD subscription notification
- Retreive Roaming and Device Info, desplay info to User B
- User A calls User B
- Get Call Direction Notification
- App gives permession for the call
- TAS assign the call to User B

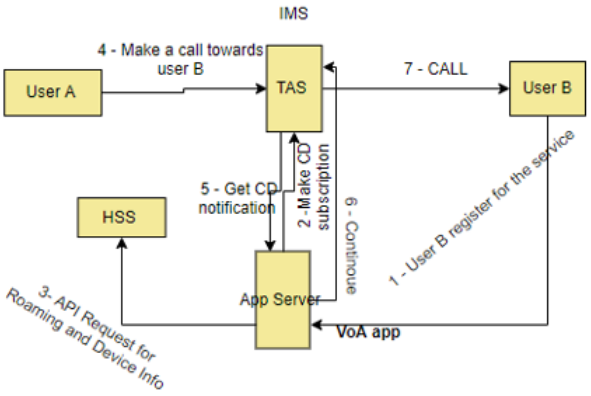

Figure 4.18: VoA data & call work flow.

4.3.13 Application code demonstration

On the figure below is demonstrated the python fucntion code part of VoA application that does the subscription to TAS.To make the subscription is used the "CallDirectionSubscription" API, so an HTTP post request is sent by the application through NEF to TAS.

```
def subscribe_to_tas(phone_number=''):
    global subs_url

    url = "https://nef51.tscc.lab-mi.vas.omnitel.it/tasseeAPI/callnotification/v1/subscription/callDirection/"
    cafile = '/usr/local/share/ca-certificates/Espoo_Root_CA.pem'

    headers = {
        'Accept': "application/json",
        'Content-type': 'application/json',
        'User-Agent': 'Jersey/2.5',  # (HttpUrlConnection 1.7.0_85)
        'Host': '127.0.0.1:8888',  # to be changed to the IP of the computer inside the Nokia/Vodafone network
        'Connection': 'keep-alive',
        'Content-Length': '318'  # may not be necessary or different -- also maybe it should be just number and not strin
    }

    subs_req = {
        "CallDirectionSubscription": {
            "CallbackReference": {
                "NotifyURL": "http://0.0.0.0/vodofferapp/api/callEvent"  # to be changed to the IP of the computer inside
                # if necessary add callbackData and notificationFormat
            },
            "Filter": {
                "Address": ["tel:+{}".format(phone_number)],  # change the phone number accordingly
                "Criteria": ["CalledNumber"],
                "AddressDirection": "Called"
            },
         "client Correlator": "demo-server-semi"
        }
    }

    response = requests.post(url, json=subs_req, headers=headers, verify=cafile, auth=HTTPBasicAuth('admin', 'admin'))
    json_response = json.loads(response.text)

    if response.status_code == 201:
        print "Application successfully subscribed to TAS SEE!"
        resource_url = str(json_response['callDirectionSubscription']['ResourceURL'])
        subs_url = resource_url.split("/")[-1]

    elif response.status_code == 404:
        print "ERROR!"
```

Figure 4.19: Subscription to TAS, function code.

After the VoA has done the subscription, it receives an url that contains the subscription ID. When the event satisfies the specified criteria occurs, the TAS notifies VoA server by using POST to the specified endpoint.

On the figure below is shown the server part code inside the application. It is the passive part of code. It acts like a serve and it waits for an HTTP post request by TAS server that in this case acts like a client in order to take commands in order to make the decision about the call.

```
@app.route("/vodofferapp/api/callEvent", methods=["POST"])
def call_event():
    incoming_req = request.json

    notification_type = incoming_req['callEventNotification']['notificationType']
    call_event = incoming_req['callEventNotification']['eventDescription']['callEvent']
    calling_participant = incoming_req['callEventNotification']['callingParticipant']
    called_participant = incoming_req['callEventNotification']['calledParticipant']
    call_session_identifier = incoming_req['callEventNotification']['callSessionIdentifier']

    print incoming_req
```

Figure 4.20: VoA server code, waiting for TAS notification.

Below is demonstrated the part of the code when VoA response with a decision to TAS, on which action must be performed. In this case, to the call is given OK to continue and the call will be routed.

```
if call_event == 'CalledNumber':
    response = {
        'action': {
            'actionToPerform': 'Continue'
        }
    }

    print jsonify(response)
    return jsonify(response), 200
else:
    return "Error", 404
```

Figure 4.21: VoA gives OK to continue the call.

4.4 NEF file log

Another feature of NEF is logging. Log API usage is based on the following rules:

- "Service" is used to identify API sever like TAS, HSS
- "Routes" means specific API path, this can be used ot select a specific API
- "Cousnumer" means who is accessing the API service, can be identified by username

Conclusions

This thesis project presented an overview of NEF Network Exposure Function, a 5G core network function which has been standardized in 3GPP Release. Firstly, we gave an overview of NEF standardization and then its services and capabilities that are exposed by Northbound APIs, along with examples of APIs are used by an application server. Lately we have demonstrated that the implementation of NEF as microservice on the containers located on cloud and orchestrated by Kubernets. The successfully deployment of NEF for 5G voice and date in a standalone architecture, more specifically on top of VoLTE is shown as well.
The main goal of the thesis was to exploit the NEF capabilities in order to enable new use cases and also to test all features that NEF provides to third parties servers (applications). For this reason, a new use case was enabled on TeVA LAB that implied the development of a new application(server) that is called VodafoneOfferApp (VoA). VoA customized promotion based on current country location and terminal device model and also gives the opportunity to subscribes to have the option do not disturb to reject, accept or suspend the call through the application before the call is routed as a voice call to the subscriber device. It is also worth to mention that this kind of application was made available thanks to ability of NEF expose assets of the Vodafone mobile core network to VoA application server and vice versa. In addition to this, to make this application functional the following APIs are used: Roaming, Device Info and CallDirection API and the following NEF functionalities are exploited: TLS termination with certificate, authentication, access lists and routing based on the path.
Furthermore, VoA app helped us to understand how NEF works, what features can support and how it interacts with other functions on the Vodafone Cores Network. Consequently, we arrived into the conclusions that NEF is able to translate the information received from the app servers to the one sent to internal core network functions, and vice versa. Also, NEF showed the capability of exposing callSubscirption, Roaming and Device Info APIs over HTTPS using certificate exchange between VoA server and NEF. In addition, NEF guarantees a basic authentication and the ristriction access to some services, guaranteeing in this way the security exposure of APIs and Vodafone Core Network Assets.Finally, NEF was able to route the different requests to different functions of core network based on the path.

New NEF versions that approaches more the 3GPP standartds/Vodafone Network are going to be released and new use cases that implies NEF are on the way.

Bibliography

[1] ETSI, *5G: Procedures for the 5G System , 3GPP TS 23.502 version 15.2.0 Release*, 06.

[2] A. T. S. Mohammad Javed, "Transformation of mobile communication network from 1g to 4g and 5g," 04.

[3] Tutorialspoint, *https://www.tutorialspoint.com/lte*.

[4] "Voice over lte: The new mobile voice," *ALCATEL-LUCENT WHITE PAPER*.

[5] GSMA, *VoLTE Service Descriptio and Implementation Guidlines*, 2014.

[6] M. T. Raza and S. Lu, "vepc-sec: Securing lte network functions virtualization on public cloud," 2018.

[7] Etsi, *https://www.etsi.org/technologies/5g*.

[8] Nokia, "Nokia networks documentations,"

[9] ETSI, *Network Function Virtualisation, Architectural Framework*, 2010.

[10] K. F. Csaba Rotter, "`Using linux containers in telecom applications.`,"

[11] V. S. Konstantinos Samdanis, Xavier Costa-Perez, "From network sharing to multi-tenancy: The 5g network slice broker,"

[12] G. Mayer, "Restful apis for the 5g services based architecture," 05.

[13] ETSI, *5G Systems: Network Exposure Function Northbound APIs, 3GPP TS 29.522 version 15.0.0 Release*, 07.

[14] ETSI, *5G: System Architecture for the 5G System , 3GPP TS 23.501 version 15.2.0 Release*, 06.

[15] Nokia, *https://www.nokia.com/networks/products/network-exposure-function/*.

[16] C. V. N. Index, "Forecast and trends, 2017-2022 white paper," 2017.

[17] Arcep, "5g: Issues & challenges," 03.

[18] Grandmetric, *https://www.grandmetric.com/2018/03/02/5g-core-network-functions/*.

[19] wikipedia, *https://www.wikipedia.org*.

[20] smartbear, *https://smartbear.com/solutions/microservices/*.

[21] KVM, *https://linux-kvm/page/*.

[22] Infoworld, *https://www.infoworld.com/article/3268073/what-is-kubernetes-your-nea*

[23] ip solutions, *https://apistraining.com/news/5g-nef/*.