Small cEllS coordinAtion for Multi-tenancy and Edge services

Grant Agreement No.671596

Topic: H2020-2014-ICT-14
Advanced 5G Network Infrastructure for the Future Internet
Research and Innovation Action

Deliverable D5.3

Techniques and optimization of VNF placement algorithms – Security issues

Document Number:	H2020-5GPPP-GA No.671596/*WP5/D5.3/30.09.2017*
Contractual Date of Delivery:	*30.09.2017*
Editor:	UPV/EHU
Work-package:	*WP5*
Distribution / Type:	Public (PU) / Report (R)
Version:	*1.0*
Total Number of Pages:	60
File:	*SESAME_Deliverable 5.3_v1.0_Final*

Abstract

This deliverable summarizes the work carried out in WP5 ("*Infrastructure Virtualisation and Management*") and, more specifically, compiles the work performed in the Task 5.3 about the optimisation of the SESAME VNF placement capabilities and the involved security issues. The higher complexity of the designed virtual infrastructure also implies added management and orchestration challenges.

One of the main aspects to "cover" in the management and orchestration plane is the placement of all the components of every NS in an optimal way. Each feasible placement policy for a set of network services has a direct effect on the service level experienced by the final users so that the performance of the NSs will be affected if the placement of their elements in the CESC cluster is not the optimal one. In consequence, the placement process is an important point in the lifecycle management, considering that it is a crucial decision to accomplish the 5G performance goals.

In addition to the VNF placement mechanism, and in order to cope with vulnerability issues and security challenges, this task has also provided a dossier of the most likely threats and attack with respect to confidentiality, privacy, integrity and availability of the CESC platform, along with their impact. Thereafter, a thorough vulnerability assessment has been performed to identify the weak system points and to propose strengthening methods.

As a result of the task, the obtained outputs are therefore twofold: an optimized VNF placement mechanism and the dossier of security issues.

Version History

Version	Date	Comments, Changes, Status	Authors, contributors, reviewers
0.1	16.05.2017	ToC and initial inputs	Bego Blanco – EHU
0.2	11.08.2017	Updated inputs on sections 1, 2 and 3	Bego Blanco – EHU Ianire Taboada – EHU Jose Oscar Fajardo – EHU Eva Ibarrola - EHU
0.3	11.08.2017	Added section 4	Elisa Jimeno - Atos
0.4	21.08.2017	Updated inputs on sections 1, 2 and 3	Bego Blanco –EHU Fidel Liberal – EHU Armando Ferro - EHU
0.5	23.08.2017	Added section 5	Konstantinos Kosmidis – UoB Haralambos Mouratidis - UoB
0.6	10.09.2017	Inputs on section 4 and 6. First complete draft ready for review	Bego Blanco – EHU Rubén Solozabal - EHU
0.7	18.09.2017	Review	Pouria Sayyad Khodashenas – i2CAT Irena Trajkovska - ZHAW
0.8	22.09.2017	Updated release	Bego Blanco - EHU
0.9	23.09.2017	Review	Athanassios Dardamanis - SMNET
1.0	26.09.2017	Final editorial and conceptual review. Document ready for submission to the European Commission	Ioannis Chochliouros - OTE

Contributors

First Name	Last Name	Partner	Email
Bego	Blanco	EHU	begona.blanco@ehu.eus
Ianire	Taboada	EHU	ianire.taboada@ehu.eus
Jose Oscar	Fajardo	EHU	joseoscar.fajardo@ehu.eus
Fidel	Liberal	EHU	fidel.liberal@ehu.eus
Rubén	Solozabal	EHU	ruben.solozabal@ehu.eus
Eva	Ibarrola	EHU	eva.ibarrola@ehu.eus
Armando	Ferro	EHU	armando.ferro@ehu.eus
Elisa	Jimeno	Atos	elisa.jimeno@atos.net
Konstantinos	Kosmidis	UoB	K.Kosmidis@brighton.ac.uk
Haralambos	Mouratidis	UoB	h.mouratidis@brighton.ac.uk
Pouria	Sayyad Khodashenas	i2CAT	pouria.khodashenas@i2cat.net
Shuaib	Siddiqui	i2CAT	shuaib.siddiqui@i2cat.net
Irena	Trajkovska	ZHAW	traj@zhaw.ch
Athanassios	Dardamanis	SMNET	adardamanis@smart.net.gr
Ioannis	Chochliouros	OTE	ichochliouros@oteresearch.gr

Glossary

Acronym	Explanation
3GPP	3rd Generation Partnership Project
5G	Fifth Generation of Mobile Communications
API	Application Programming Interface
ARM	Advanced RISC Machine
ARP	Address Resolution Protocol
ASLR	Address Space Layout Randomization
BF	Broadband Forum
CAPEC	Common Attack Pattern Enumeration and Classification
CE-RAN	Cloud-Enabled Radio Access Network
CESC	Cloud-enabled Small Cell
CESCM	Cloud Enabled Small Cell Manager
CLI	Command-Line Interface
CP	Connection Point
CPE	Customer Premises Equipment
CPU	Central Processing Unit
CVE	Common Vulnerabilities and Exposures
CWMP	CPE WAN Management protocol
DB	Database, Data Base
DC	Data Centre
DDoS	Distributed Denial of Service
DoS	Denial of Service
DSS	Decision Support System
EMS	Element Management System
EPC	Evolved Packet Core
ETSI	European Telecommunications Standards Institute
FAP	Femto Access Point
FCAPS	Fault, Configuration, Accounting, Performance and Security
GA	Grant Agreement
GS	Group Specification
GTP	GPRS Tunnelling Protocol
GUI	Graphical User Interface
H2020	Horizon 2020
HTTP	Hypertext Transfer Protocol
HW	Hardware
HWA	Hardware Accelerator
I/O, i/o	Input/Output
IaaS	Infrastructure-as-a-Service
ICT	Information and Communication Technology
IEEE	Institute of Electrical and Electronic Engineers
IETF	Internet Engineering Task force
ILP	Integer Linear Programming
IT	Information Technology
KPI	Key Performance Indicator
KVM	Kernel-based Virtual Machine
LAN	Local Area Network
Light DC	Light Data Centre
LXC	Linux Containers
MANO	Management and Orchestration

MEC	Mobile Edge Computing
MOCN	Multi-Operator Core Network
NCT	Network Connectivity Topology
NFP	Network Forwarding Path
NFV	Network Functions Virtualisation
NFVI	Network Functions Virtualisation Infrastructure
NFVO	NFV Orchestrator
NM	Network Management
NMS	Network Management System
NS	Network Service
NSD	NS Descriptor
OFDM	Orthogonal Frequency Division Multiplexing
ONF	Open Networking Foundation
OPNFV	Open Platform for NFV
OSS	Operations Support System
OVS	Open virtual Switch
PNF	Physical Network Function
PCI	Peripheral Component Interconnect
PPP	Public-Private Partnership
QoS	Quality of Service
QEMU	Quick Emulator
RAM	Random Access Memory
RAN	Radio Access Network
REST	Representational State Transfer
RIA	Research and Innovation Action
RISC	Reduced Instruction Set Computer
RO	Resource Orchestrator
RO	Robust Optimization
RRM	Radio Resource Management
RPC	Remote Procedure Call
RTC	Real Time Clock
RVC	Ruby vSphere Console
SC	Small Cell
SCaaS	Small Cells-as-a-Service
SCNO	Small Cell Network Operator
SDN	Software-defined Networking
SLA	Service Level Agreement
TOSCA	Topology and Orchestration Specification for Cloud Applications
TR	Technical Report
VDU	Visual Display Unit
VIM	Virtual Infrastructure Manager
VL	Virtual Link
VM	Virtual Machine
VNF	Virtual Network Function
VNFFG	VNF Forwarding Graph
VNFFGD	VNF Forwarding Graph Descriptor
VNFFP	VNF Forwarding Path
VNFD	VNF Descriptor
VNFM	VNF Manager
VSCNO	Virtual Small Cell Network Operator
WAN	Wide Area Network
WP	Work Package

Table of Contents

List of Figures

List of Tables

1. Introduction

This deliverable summarizes the work carried out in WP5 ("*Infrastructure Virtualisation and Management*") and, more specifically, it compiles the work performed in the Task 5.3 regarding the optimisation of the SESAME VNF placement capabilities and the involved security issues. The task focuses on VNF placement challenges in the virtualised execution infrastructure, i.e. the Light DC designed in WP4 ("*Light DC Design and Implementation*"), and the associated VIM developed in the previous tasks of WP5. These challenges include:

- Dynamic placement and elastic management of VNFs over virtual nodes.
- Virtual resource provisioning and efficient local and cross-CESC site placement algorithms to "meet" service level agreements.
- Adaptation of architectures for Cloud federation to the CESC, interoperation and network management (NM).
- Implementation of high impact sample VNFs (e.g., caching) for assessing the performance of the SESAME platform.
- APIs to remotely access the virtualised infrastructure.

In order to reach the previous objectives, the necessary steps in the service cycle will be tackled. The aim is to design and support the optimisation and bursting scenarios over the CESC network, by implementing the required functionalities for appropriately managing the virtualised resources.

As already defined in other work packages -*and more specifically in WP4*- SESAME proposes a distributed Cloud-Enabled RAN (CE-RAN) architecture. Designing a multi-tenant CE-RAN as an evolution of commercial Small Cells (SC) towards the so-called Cloud-Enabled Small Cells (CESC), poses extra challenges for service management and orchestration. This issue covers a wide range of subjects, including optimized resource placement, efficient lifecycle management, etc.

In this context, any tenant can deploy Network Services (NSs) to be run over virtualized infrastructures, by adding flexible and scalable hardware resource management capabilities. A NS is defined as a chain of Virtual Network Functions (VNF), while a VNF is devoted to running mobile edge service instances to execute virtualized service level functions within a CESC cluster; this implicates, *for example*: deep packet inspection, context aware routing, edge caching, transcoding unit, etc.

The use of a virtualized infrastructure has several advantages, but to make them valuable the performance of the deployed 5G service has to "meet" some performance indicators regarding, *for example*, latency or throughput. In addition, some VNFs are network-and computationally-intensive and may require some specific features to be provided by the NFVI to "meet" the performance goals. For this reason, the Light DC has been designed to include a variety of hardware (HW) appliances, such as hardware accelerators (HWA) and interfaces dedicated to networking and packet processing. These additional computational resources enable the reduction of the number of physical servers, network appliances and overall power consumption.

Obviously, the higher complexity of the designed virtual infrastructure also implies added management and orchestration challenges. Particularly regarding Task 5.3, one of the main aspects to cover in the management and orchestration plane is the placement of all the components of every NS, in an optimal way. Each feasible placement policy for a set of network services has a direct effect on the service level experienced by the final users. Additionally the performance of the NSs will be affected if the placement of their elements in the CESC cluster is not the optimal. In consequence, the placement process is an important point in the

management lifecycle, considering that it is a crucial decision to accomplish the 5G performance goals.

Apart from that, current trends towards green information and communication technologies target the improvement of energy efficiency in all the nodes, including the network nodes themselves [1]. In the scope of the SESAME project, **an energy-aware VNF placement solution for a cluster of cloud-enabled SCs in a 5G-deployment scenario involves both the micro-servers and SDN-enabled switches** to deploy the overall virtualized infrastructure needed. With these premises, we propose a VNF placement mechanism that minimizes power consumption of the deployed system subject to service delay constraints.

In addition to the VNF placement mechanism, and in order to cope with vulnerability issues and security challenges, this task has also provided a dossier of the most likely threats and attack with respect to confidentiality, privacy, integrity and availability of the CESC platform, along with their impact. Thereafter, a thorough vulnerability assessment has been performed to identify the weak system points and to propose strengthening methods.

As a result of the task, the obtained outputs are two-fold: (i) an optimized VNF placement mechanism and; (ii) the dossier of security issues.

1.1. Deliverable outline

The present deliverable covers the service management and orchestration functions and, particularly, the placement process of Network Services (NSs), which is defined as a VNF chain by the SESAME project. The document begins with a brief summary of the SESAME architecture and a review of related work about VNF placement alternatives that could be considered. Then, we mainly focus upon modelling the VNF placement problem and upon the SESAME-specific solution to the placement problem.

In particular, the document is organized in six (-6-) distinct sections:

- Section 1 offers a brief introductory overview and the outline of the deliverable.
- Section 2 defines the design of the SESAME VNF placement algorithm; it also describes the service-mapping problem and explains the model employed to solve it.
- Section 3 provides the output of the experiments with the designed algorithm and discusses the obtained results.
- Section 4 details the implementation issues for an automated management of the service lifecycle and provision.
- Section 5 analyses vulnerability issues and security challenges.
- Finally, Section 6 gathers the main conclusions of the task.

2. Design of the SESAME VNF placement algorithm

This multi-tenant radio-cloud environment poses extra challenges for service management and orchestration. The manager/orchestrator needs to simultaneously take into account both the radio status (i.e.: volume of traffic, geographical distribution of traffic, etc.) and the cloud capabilities (i.e.: available IT resources, Virtual Machine (VM) to VM communication requirements, etc.) for all the actions related to the service lifecycle management.

Service deployment is the first step in the service lifecycle management. By definition, it consists of two processes:

1) **Service Mapping:** It is a logical process to decide where (i.e., inside the CESC nodes) to deploy Network Service components (i.e., VNFs), over Light DC depending on parameters such as resource availability, agreed Service Level Agreement (SLA) with the tenant, traffic patterns and predictions, etc. A wide range of solutions might be adopted to solve the service mapping problem, ranging from accurate Integer Linear Programming[1] (ILP) formulations to heuristics. However, whilst accuracy is an important parameter, computational complexity is a major factor that determines the benefits of adopting one solution over the other.
2) **Deployment Mechanism:** It represents the actual interaction within virtual appliances such as NFV orchestrator (NFVO), Virtual Network Function Manager (VNFM) and Virtual Infrastructure Manager (VIM) to deploy services. This section highlights the service management and orchestration challenges that arise when handling service mapping on a multi-tenant distributed CE-RAN architecture, as introduced by the SESAME project.

To this end, this section "addresses" the problem of service mapping. We initially present the overall SESAME architecture with a focus upon the management level. Next, we define the type of NSs that SESAME envisions for a cloud-enabled radio environment. Then, the service mapping mechanism is further discussed. Finally, we describe our design of the VNF placement algorithm. The deployment mechanism is later described and discussed in Section 4.

2.1. SESAME overall architecture

The SESAME platform consists of a cluster of CESCs, which are devices that include both the processing power platform and the SC unit. CESCs can be deployed at low- and medium-scale venues and support multiple network operators (i.e.: multi-tenancy) and, further, network services and applications at the edge of the network. The proposed solution extends the Small Cell-as-a-Service (SCaaS) model, which facilities a third party provisioning of shared radio access capacity to mobile network operators in localized areas, with the provision of Mobile Edge Computing[2] (MEC) services. The architecture, as depicted in Figure 1, combines the current 3GPP framework for network management in RAN sharing scenarios and the ETSI NFV framework for managing virtualised network functions.

The CESC offers virtualised computing, storage and radio resources; the CESC cluster is considered as a cloud from the upper layers. With this architecture, the assignment of cloud "slices" is the feature that enables multi-tenancy. This way, the execution platform is used to

[1] For more informative details see, *inter-alia*: https://en.wikipedia.org/wiki/Integer_programming
[2] For more informative details see, *for example*: https://en.wikipedia.org/wiki/Mobile_edge_computing

support VNFs that implement the different features of the SCs as well as to support MEC services.

In our scope, a CESC consists of a 5G-enabled SC with its standard backhaul interface, standard management connection (TR069[3] interface for remote management) and with necessary modifications to the data model (TR196[4] data model) to allow Multi-Operator Core Network (MOCN) radio resource sharing. The CESC will be composed by a physical SC unit attached to an execution platform based on one of x86[5], ARMv8[6], MIPS64[7] architectures.

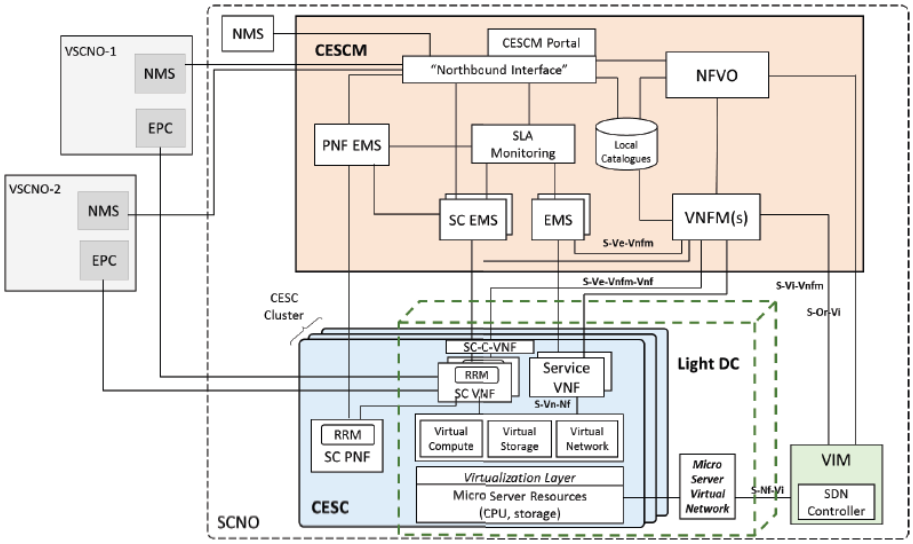


Figure 1: SESAME Overall Architecture

Edge cloud computing and networking are realised through the sharing of computation, storage and network resources of those micro-servers present in each CESC, which are grouped together as a cluster and form the Light DC. Moreover, the SESAME project proposes a micro-scale virtualized execution infrastructure in the form of a Light DC to enhance the virtualization capabilities of the SC deployment, providing high processing power at the network edge. The Light DC concept, which encompasses the micro-servers of the different CESCs in a cluster,

---

[3] TR-069 (Technical Report 069) is a technical specification that defines an application layer protocol for remote management of end-user devices. It was published by the Broadband Forum (BF) and entitled CPE WAN Management Protocol (CWMP). More informative details can be found, *for example*, at: https://en.wikipedia.org/wiki/TR-069

[4] TR-196 (Technical Report 196) is a BF technical specification. Its official title is "Femto Access Point Service Data Model." The purpose of this TR is to specify the Data Model for the Femto Access Point (FAP) for remote management purposes using the tr-069 CWMP.

[5] x86 is a family of backward compatible instruction set architectures based on the Intel 8086 CPU and its Intel 8088 variant. More relevant information can be found, *inter-alia*, at: https://en.wikipedia.org/wiki/X86.

[6] ARM is the industry's leading supplier of microprocessor technology, offering the widest range of microprocessor cores to address the performance, power and cost requirements for almost all application markets. Combining a vibrant ecosystem with over 1,000 partners delivering silicon, development tools and software, and with more than 90bn processors shipped, our technology is at the heart of a computing and connectivity revolution that is transforming the way people live and businesses operate. For more details also see: https://www.arm.com/products/processors.

[7] For more details see, *for example*: https://en.wikipedia.org/wiki/MIPS_architecture

provides a high manageable architecture optimized to reduce power consumption, cabling, space and cost. To achieve these requirements, it relies upon an infrastructure that aggregates and enables sharing of computing, networking and storage resources being available in each micro-server belonging to the CESC cluster.

The Light DC infrastructure also provides the backhaul and fronthaul resources for guaranteeing the requirements for connectivity in multi-operator (multi-tenancy) scenarios. The hypervisor computing virtualization extensions enable access of VMs to the HWAs, for providing an execution platform that can support the deployment of VNFs. Different types of VNFs can be deployed through the VIM, that is: for carrying out the virtualization of the SC; for running the cognitive/self-x management operations, and; for supporting computing needs for the mobile edge applications of the end-users. The combination of the proposed architecture allows achieving an adequate level of flexibility and scalability in the edge cloud infrastructure.

CESCM is the main management component in the architecture, covering the orchestration, management and configuration of NSs. The CESCM has a high-level knowledge of the virtual and physical resources available on the cloud-enabled RAN environment, including the radio access functionalities. The challenge SESAME aims to overcome (i.e., to provide services that involve both radio and virtualisation aspects at the network's edge), implies more complex management functionalities for such services. For that purpose, the platform is designed in such a way that the radio access management task (e.g. transmission power control, packet scheduling, handover and cell reselection thresholds, etc.) and NFV management responsibilities (e.g. VNF/service instantiation, lifecycle management, policy management, etc.) can be handled in an orchestrated and more centralized way. CESCM is composed of the following modules:

- The **CESCM Portal** is the graphical user interface (GUI) used by SC Network Operators (SCNOs) and Virtual SCNOs (VSCNOs) to interact with the platform capabilities. Such interaction is enabled by the Northbound Interface[8], which supports the data exchange of the tenant configuration parameters / requirements with the Orchestrator.
- The **NFVO** is the entity in charge of NS lifecycle management (creation, termination, monitoring, scaling, etc.) via coordination between ETSI MANO[9] elements, such as VIM, EMS and VNFM. NSs are defined in the form of NS descriptors (NSD), which contain VNF descriptors (VNFD) – defining required IT resources to deploy each VNF as well as specific configuration needed per VNF to offer its functionality – and connectivity between VNFs.
- The **VNFM** is the entity in charge of the lifecycle management of the VNFs, from deployment to termination, keeping track of their status to adjust their configuration if needed.
- The **EMS** is the entity in charge of the key functionalities as fault, configuration, accounting, performance and security (FCAPS) in specific per VNF/PNF understandable language. The EMS associated to radio functions also includes autonomous self-x functionalities to reconfigure the mobile network.
- The **SLA** component enhances service reliability providing monitoring mechanisms to evaluate the performance of NSs in the radio and cloud environments. It communicates with the NFVO, notifying faults in the system for it to perform the appropriate actions that assure the QoS guarantees of each service in a multi-tenant environment.

[8] For further information see, *for example*: *https://en.wikipedia.org/wiki/Northbound_interface*

[9] Also see: *ETSI GS NFV-MAN 001 V1.1.1 (2014-12): Network Functions Virtualisation (NFV); Management and Orchestration.*

As described in the ETSI, the VIM is the responsible for managing the virtualized infrastructure in order to enforce CESCM instructions into CESC cluster, i.e. the Light DC. It includes the catalogue of the allocated resources, forwarding graphs and chaining rules among VNFs and repository of resources, to provide optimized features. In the SESAME platform, the VIM is the software entity that monitors and manages the Network Function Virtualization Infrastructure (NFVI) (i.e. the Light DC) and performs the lifecycle management of the virtual units that will host the VNFs (VMs, lightweight technologies like Docker[10], etc.). This centralized administration of virtual resources across multiple localized infrastructures, so that instances can be administrated in a coordinated way; it also provides the flexibility and scalability needed to optimize and maximize the use of such resources. The VIM is enhanced with SDN component for the networking aspect. The controller takes into account the physically distributed SESAME NFVI and the stringent requirements in RAN performance metrics. Moreover, SESAME uses SDN for propagating the VNF chaining requests to the NFVI, in order to properly manage the networking resources within the Light DC.

2.2. SESAME NS definition

SESAME defines the Network Service as a collection of VNFs (including radio-related and service-related instances) required to deploy a complete 5G mobile service (e.g., innovative video content delivery) for the end-users of the VSCNO. Figure 2 illustrates a conceptual view of the ecosystem, which hosts three different VSCNOs over the shared NFVI (i.e. the Light DC).

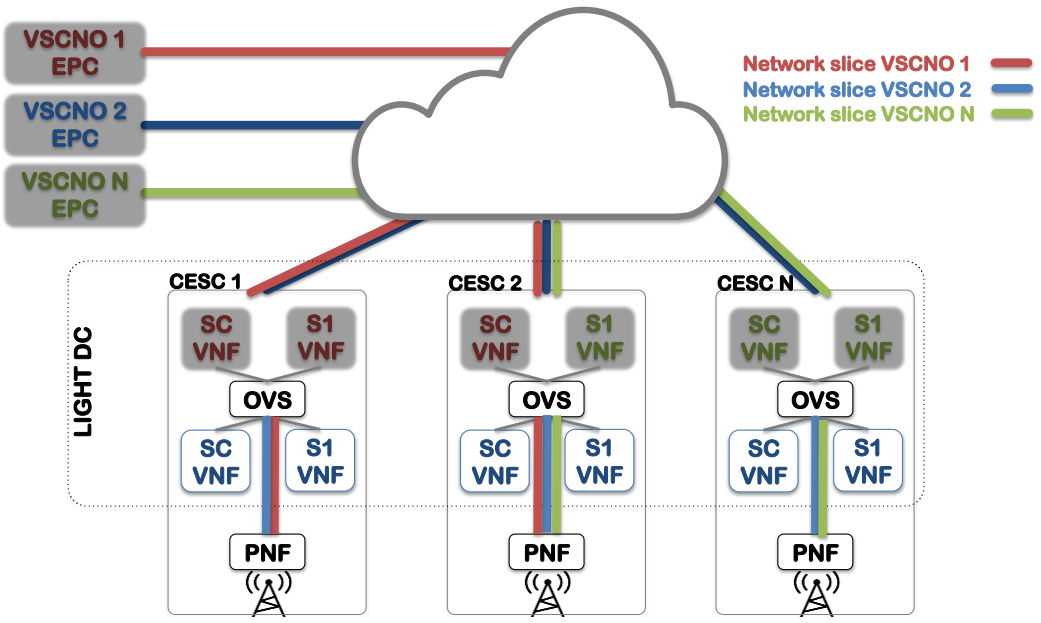


Figure 2: SESAME Edge Network Services

In order to form the multi-tenant scenario depicted in Figure 2, the NFVO needs to process the functional chaining of VNF requested by each VSCNO and to trigger its instantiation to the VIM that manages the NFVI. Therefore, a SESAME NS can be characterized through a series of radio-level and service-level KPIs that are captured in the SLA between the SCNO owning the NFVI and the interested VSCNO. These KPIs may include bare network parameters (e.g., aggregated

[10] For more details also see: *https://en.wikipedia.org/wiki/Docker_(software)*

uplink/downlink bitrate in a certain coverage area for a percentage of time) or more complex service-level KPIs (e.g., a number of supported video sessions with edge caching/edge transcoding features).

Every SESAME NS is defined through its associated NSD. As a matter of fact, upon the request of VSCNO on the CESCM Northbound interface or graphic user interface, NSD will be extracted/generated internally (more details will be presented later on). The complete deployment of a service consists of two main steps: 1) VNF instantiation and its relevant configurations, *and*; 2) enforcing the data flows between them. NFVO will coordinate this activity and via interaction with the other modules (e.g. VIM), it will handle the responsibility.

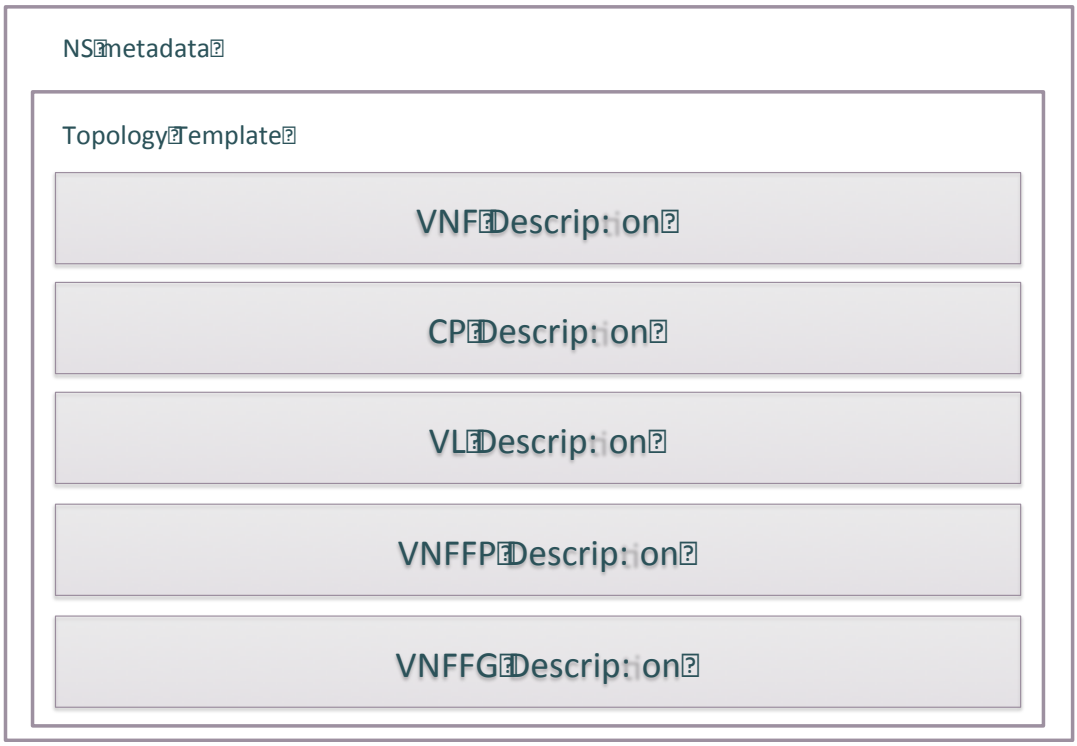


Figure 3: TOSCA-NFV Network Service template

Figure 3 depicts the TOSCA-NFV Network Service Template[11]. The main building blocks of a NSD are the VSCNO Network Connectivity Topology (NCT) and the VNF Forwarding Graph (VNFFG) descriptors. A VNF Forwarding Graph Descriptor (VNFFGD) is a deployment template that describes a topology of the Network Service or a portion of the Network Service, by referencing VNFs and Virtual Links (VLs) that connect them. The NCT determines the complete list of VNFs and their Connection Points (CPs), as well as the possible interconnections through a series of Virtual Links. In this sense, the VSCNO NCT can be seen as the virtual network slice assigned to that VSCNO in the CESC Cluster. The VNFFG describes the possible paths that data packets can follow within the VSCNO NCT. Generally speaking, a VNFFG determines the possible Network Forwarding Paths (NFPs) for a specific type of service. A NFP is an ordered list of CPs that determines a chain of VNFs from entry to exit.

[11] Further related information can also be found at: *http://docs.oasis-open.org/tosca/tosca-nfv/v1.0/tosca-nfv-v1.0.html*

Upon the reception of the deployment request (issued by NFVO based on the end-user requirements and pushed towards VIM), the VIM will "map" the logical request to the actual hardware by the instantiation of VNFs, which may run in different micro-servers. The distributed nature of the edge cloud introduces a novel challenge on the hardware resource allocation, which is the subject of this deliverable. At the end of the placement process, the created NS for the VSCNO can be observed as a separate underlying virtual LAN instantiated and ready-to-use. However, the actual data flows between VNFs need to be enforced (VNFs can be seen as the origin and destination of data flows). The SDN controller (integrated into the VIM) implements the forwarding rules necessary to move packets according to the VNFFG. In this way, the data packets will travel correctly from the first to the last VNF on the NS.

2.3. Service Mapping and Deployment Workflow

The concept of service mapping, (i. e., the placement process), refers to the mechanism that allocates the "softwarised" components (VNFs) of a NS onto the interconnected data-centres that conform the CESC Cluster. In other words, this implicates "where to instantiate the VNFs chain that composes each NS from every tenant".
Figure 4 shows an example of the placement process of a NS in the virtualized resources of a SESAME Light DC.

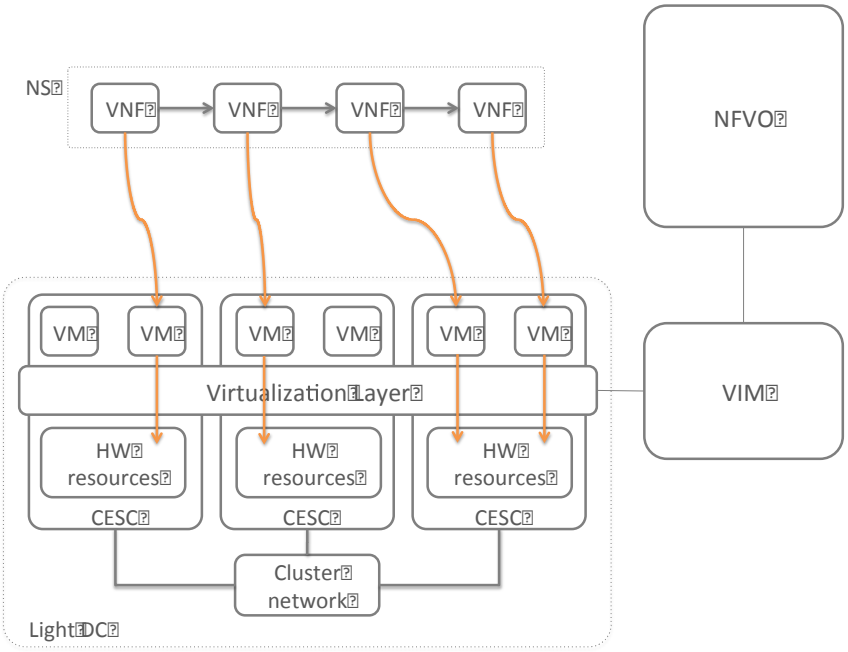


Figure 4: Placement process in the context of SESAME

At a logical level, and upon a new NS request from one of the tenants, the NFVO configures the ordered sequence of VNFs that provides the required network service.
Then, the NFVO maps the VNF chain to specific VNF instances and service chaining configurations in the Light DC. Finally, the VIM -which manages the virtual units that will host the VNFs- propagates mapping requests to the underlying physical resources that compose NFVI.

Figure 5 illustrates the scheme of the SESAME placement mechanism. As stated previously, the NS descriptor contains the logical VNF chain that composes a requested NS (as stored in the NS

Catalog), plus the service constraints imposed by the KPIs of the corresponding SLA agreed with the corresponding tenant. With this input, the placement algorithm "checks" the available virtualized resources in the NFVI catalog and the instantiation requirements for each VNF in the VNF catalogue.

This process is depicted in Figure 5; there, when a tenant needs to deploy a new service, it makes a request to the Service Orchestrator functional element, which is implemented as the NFVO in ETSI-MANO terminology. The orchestrator analyses its NS catalog to extract the correspondent service chain. The VNF chain extracted from the catalog is used as an "input" for the placement algorithm, together with the service constraints imposed by the KPI of the corresponding SLA between the VSCNO and SCNO.

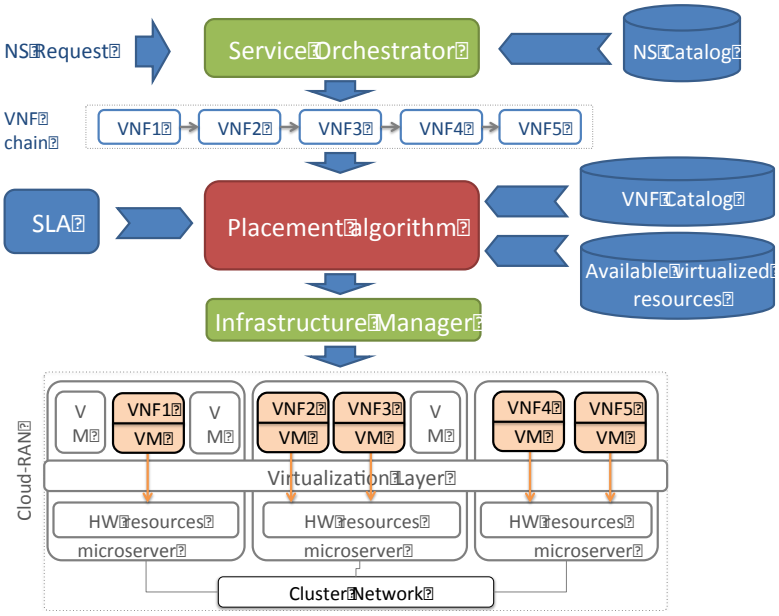


Figure 5: VNF placement process

In the SESAME VNF placement proposal, the SLA KPIs include the following network parameters: accepted latency for the NS; aggregated user bit rate, and; a robust protection parameter. This protection parameter is related to the uncertainty of the user traffic demand. We assume that the network service has a mean aggregated traffic demand that is used to size the necessary resources, in order to serve tenant's user. But along the service time, the user traffic demand varies around this mean value, thus affecting the amount of resources needed. The robust protection parameter establishes the expected peak traffic demand deviation, and is used to overestimate the allocation of virtualized resources in order to be prepared for eventual user peak demands.

Next, the placement algorithm inspects the VNF catalog for the parameterizations of the VNF instances needed to match the SLA, and it checks the available resources provided through the virtualized infrastructure. VNFs are characterized to use resources (i.e.; number of cores, RAM and storage) according to the aggregated bitrate (i.e., characterized aggregated traffic flows of a NS) served by the NS they belong to. Moreover, the service latency of each VNF also depends on the traffic load supported by the VNF instance. Lastly, virtualized resources may also include other hardware appliances, such as hardware accelerators (HWA), which can improve the performance of a VNF, in terms of latency. This way, these additional appliances help matching SLA with the tenant, but at a higher energetic cost. All the VNFs composing the service chains of

the requested network services must be allocated in the available virtualized resources, including a given number of CPUs, hardware appliances, RAM and storage space and network bandwidth. We assume that each core runs a single VM that is devoted to the execution of a single VNF instance, but one VNF instance can scale from one-to-many VMs upon the service requirements of the NS. On the other hand, the transmission latency along the network topology between two VNFs allocated in different micro-servers depends on the traffic load of the involved links and the size of the flow to be transmitted. With this information, the placement algorithm assigns each VNF instance to one -or more- VMs with the objective of minimizing the energy consumption of the whole system, while matching the latency constraints for the NS.

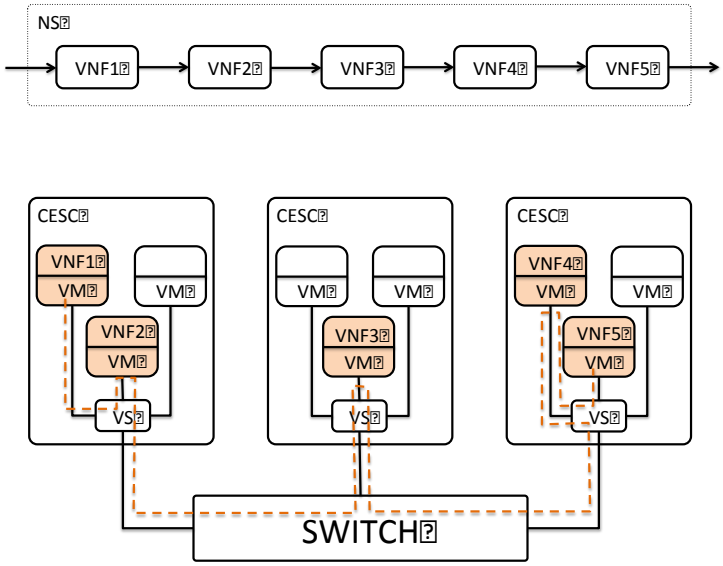


Figure 6: Assignment of VNFs to VMs

Finally, the Infrastructure Manager functional element (or the VIM in ETSI-MANO terminology) receives the outcome of the placement decision and maps each VNF to the assigned resources (as in Figure 6). It is important to point out that a VNF chain includes radio-related and service-related instances. The placement algorithm introduced in this work considers a scenario with several tenants, each of them requesting one or more NSs, which share the available resources.

Considering the aforementioned communication particularities of 5G, Figure 7 shows an example of a simple complete placement result of a multi-tenant environment. This simple example considers a distributed edge cloud, shared by two tenants. Tenant 1 offers two network services and Tenant 2 offers a single network service. All NSs are associated to their correspondent latency constraint. In the example, two NSs of the same tenant share the same radio VNF-SC VNF (SCVNF). In this way, the common radio operations are performed in a coherent way between the different edge services of a tenant, and the data paths of the specific service instances per tenant are split. The SCVNF of each tenant will be shared by all de NSs of this tenant, which means that the use of resources of the SCVNF must then consider the sum of the aggregated flows of the NSs of the same tenant.

The distinctive features of 5G communications have important implications over the placement algorithm. Meanwhile, every tenant allocates a unique SCVNF instance at the edge cloud, in order to support the control plane and the data plane operations of all the SCs associated to

that tenant. Moreover, since the SCVNF manages the uplink and downlink traffic between the mobile users and the core network, it leads to circular forwarding paths when including edge service instances. The example also shows how some VNFs need more than one VM to execute the associated functionalities, and how one VNF uses a HWA to "meet" the latency constraint of the server.

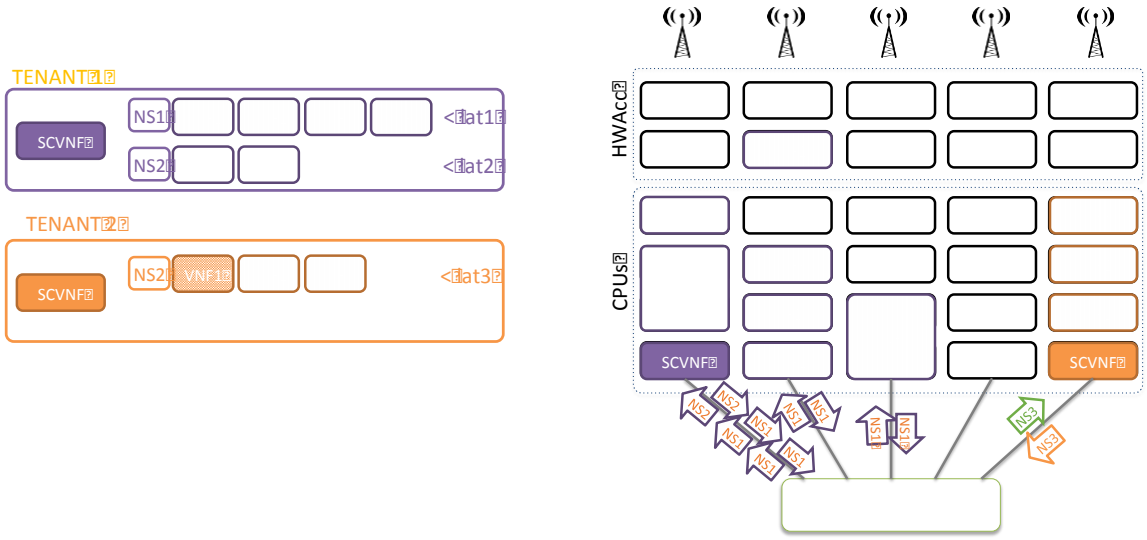


Figure 7: Multi-tenant placement example

The success of the approach depends on the efficiency of the VNF placement algorithm to allocate the virtual functions in the available resources, given a set of service constraints.

Next subsections describe the SESAME approach to design the placement algorithm and finally present the service mapping problem modelling as a VNF placement optimization problem that minimizes power consumption subject to meet the service delay constraints.

2.4. Design of the placement algorithm

In order to "address" the VNF placement problem in the scenarios considered in the SESAME project, we propose a placement mechanism that minimizes power consumption subject to latency constraints. The VNF placement problem discussed above basically consists in a discrete combinatorial optimization problem. In other words, service mapping in NFV is a NP-hard[12] optimization problem [8] that can be solved by using exact or heuristic optimization strategies depending on the instance size. Existing literature provides a wide variety of VNF placement approaches, ranging from accurate Integer Linear Programming (ILP) formulations to heuristic algorithms.

However, most existing solutions are more focused on the use of NFV over big centralized data centres, rather than over small edge clouds devoted to provide radio access and edge networking services. In addition, the analysed studies rarely consider uncertainty in the definition of the problem.

[12] Also see: *https://en.wikipedia.org/wiki/NP-hardness*

In consequence, the main contribution of T5.3 is the design of an energy-aware VNF placement expert system for a Cloud-Enabled Radio Access Network (CE-RAN) environment, deployed with 5G small cells. We solve the discrete optimization problem of resource assignment combined with robust optimization techniques (RO) [10] to develop an energy-aware VNF placement policy. Thus, the novelty of our approach is the use of RO, which allows the generation of suboptimal placement policies that enable the introduction of uncertainty in the problem. In particular, we introduce service demand uncertainty in order to capture the variable nature of traffic flow size and per-service job execution burden.

Along the following subsections, we provide a brief review of the most relevant works we analysed related to VNF placement and service mapping. Then, we introduce the VNF placement model describing the modelling process of each component of the model. Next, we describe the consecutive iterations made in the design of the algorithm to reach the definitive version. Finally, we present in detail the mathematical model built to solve the VNF placement problem in the SESAME context.

2.4.1. Related Work

Nowadays, mobile operators see network virtualisation as the "key" towards enabling flexible 5G network architecture. For this reason, the problem of VNF placement has been attracting increasing attention during the last years ([2]-[16]). Nonetheless, the optimal placement of virtual elements to physical resources aimed at minimizing power consumption latency-constrained for the 5G scenario under study is a complex issue.

VNF placement is a resource allocation discrete or combinatorial optimization problem, which can be considered equivalent to a knapsack problem[13]. In consequence, it can be solved by using exact methods -or heuristics- depending on the number of VNF instances and the complexity of the resources in competition. As typically provided in the literature (see [3]-[5]), an ILP approach optimally solves the placement problem in reasonable time when the number of service instances is low. In reference to heuristics-based placement proposals, the variety of tools employed for solving the problem is huge; as examples, [15] uses a greedy algorithm[14] whereas [7] a binary search heuristic.

Even though some of the aforementioned relevant works deal with energy-related optimization taking into account delay bounds (for instance, the objective in [3] and [4] is minimizing the number of CPUs constrained by latency), those solutions do not usually consider optimal/suboptimal power consumption itself. However, it is worth to mention the few works we have found that deal with power-based VNF placement proposals ([8]-[12]) for our case study. The work in [8] aims at minimizing power consumption allowing users to "meet" delay requirements by using a genetic algorithm approach[15], while the works in [9] and [10] propose heuristic algorithms that give power reduction improvement. Besides, [11] presents a suboptimal placement solution that combines traffic and energy cost optimization derived by means of a Markov approximation[16] with matching theory. Apart from that, research works in [13]-[15] go a step forward, and cope with the migration problem of active micro-servers needed in low traffic conditions, to turnoff -or suspend- micro-servers so as to achieve energy savings.

[13] For further information also see, *for example*: https://en.wikipedia.org/wiki/Knapsack_problem
[14] For more related information also see, *inter-alia*: https://en.wikipedia.org/wiki/Greedy_algorithm
[15] For more related information also see, *inter-alia*: https://en.wikipedia.org/wiki/Genetic_algorithm
[16] Also see: https://en.wikipedia.org/wiki/Markov_chain_approximation_method

Moreover, a common assumption of the different optimization models proposed to solve different versions of the VNFs placement problem is that input data is perfectly known, which is difficult in practice. Unfortunately, small deviations in input data values may usually lead to situations where a previously found optimal solution is not even feasible any more and, *consequently*, the presence of uncertain data may produce useless placement solutions. In order to cope with uncertain input parameters, RO [22] is applied to solve optimization problems. Indeed, RO has been successfully employed in different contexts, such as in resource allocation in OFDM networks [15] with uncertainty in channel state information, in cloud resource provisioning without perfect knowledge of demands and price [12], and even in the virtual network embedding problem on a physical network substrate in recent works ([18]-[21]) that usually deal with uncertain service demands.

Nevertheless, to the best of our knowledge, except the work presented in [12], there is no VNF placement proposal based on RO. Therefore, to achieve our goal of designing an energy-aware placement solution by using RO approach that considers service demand uncertainty, the work in [12] has been relevant. We develop a novel power-based placement solution with uncertainty in VNFs CPU demands, based on RO theory. To this aim, we take into account the physical servers and network resources along with their energy efficiency. Nonetheless, we focus upon the core network part whereas the scope of the project is oriented to combined deployment of radio / service network functions at the edge. Additionally, the placement problem considered in this project takes into account a heterogeneous edge cloud, made up of a cluster of inter-connected SCs.

2.4.2. Description of the components of the VNF placement problem

After we presented the VNF placement problem in the study, in this section it is discussed a placement solution to the problem, by using the Robust Optimization approach. The addressed VNF placement problem focuses on where each VNF -which conforms a NS- is located over the provided infrastructure. Hence, our intelligent placement algorithm will determine where each VM of a SC is located. To that end, *first*, we describe the model of the problem that is composed of: the network service model, the VNF model, the micro-server model, the switch model and the latency and power consumption calculation model.

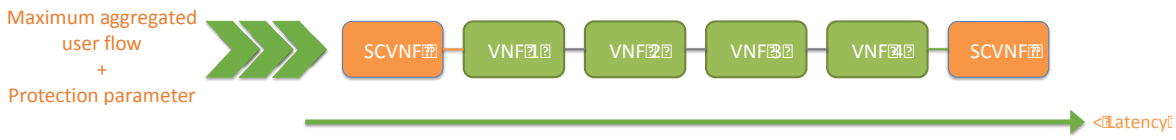


Figure 8: Network Service Model

In our proposal, a Network Service is characterized by five parameters:

- The first one, *already introduced*, is the *VNF chain* as an ordered set of virtual functions that make-up a NS.
- The second one is the *tenant that offers the NS*. This is a crucial aspect since each NS must begin and end with a special VNF (the SCVNF) to manage the GTP tunnel of the mobile user communication. The specification of the tenant that offers each NS determines the SCVNF (belonging to that single tenant) that must be used, to process the data flow of each end-user. This SESAME-specific feature is introduced in the model to enable multi-tenancy and it is one of the distinctive characteristics of the proposed placement mechanism.

- The third parameter is the *maximum user aggregated flow* that the NS is dimensioned to serve. This aggregated maximum flow is affected by the fourth parameter, which is the protection parameter.
- This *protection parameter* is related to the uncertainty of the user traffic demand. We assume that the network service has a mean aggregated traffic demand that is used to size the necessary resources in order to serve tenant's user. But along the service time, the user traffic demand fluctuates around this mean value, thus affecting the amount of resources needed. The robust protection parameter is therefore related to the expected peak traffic demand deviation and will be used to overestimate the allocation of virtualized resources, in order to be prepared for eventual user peak demands.
- Finally, the fifth modelling parameter of a NS is the *maximum accepted latency* that must be accomplished. This service latency is used by the model to constrain the solution space.

After describing the NS model, the next step is the modelling of the main components of the network service, that is the VNFs. Basically, the main objective of the VNF model is to "derive" the use of resources in each VNF of every NS and their processing latency. In consequence, VNFs are characterized by the use of resources (i.e. the number of cores, RAM and storage) according to the aggregated bitrate (i.e. characterized aggregated traffic flows of a NS) served by the NS they belong to. Moreover, the service latency of each VNF also depends on the traffic load supported by the VNF instance.

Table 1, *below*, shows an example of VNF modelling where the resource consumption is subject to the maximum aggregated traffic flow the VNF must support.

	N Mbps	2*N Mbps	3*N Mbps	4*N Mbps	5*N Mbps	6*N Mbps
Nº of CPUS cores	1	1	1	1	2	2
CPU usage	55%	60%	70%	90%	65%	70%
RAM	1 GB	1 GB	2 GB	2GB	4 GB	4 GB
Storage	10 GB	10 GB	10 GB	10 GB	10 GB	10 GB
Service latency	10 ms	15 ms	25 ms	40 ms	20 ms	25 ms

Table 1: Example of VNF modelling

In the context of a CESC cluster introduced by SESAME project, each NS has to be deployed in the Light DC. This means that all the VNFs composing the service chains of the requested network services must be allocated in the available virtualized resources, including a given number of CPUs, hardware appliances, RAM and storage space and network bandwidth. In consequence, the modelling of the virtualized infrastructure becomes essential to set-up the constraints related to the available resources as input of the optimisation algorithm.

Figure 9 depicts the representation of the micro-server model with its components. We assume that each core "runs" a single VM that is devoted to the execution of a single VNF instance, but one VNF instance can scale from one to many VMs upon the service requirements of the NS.

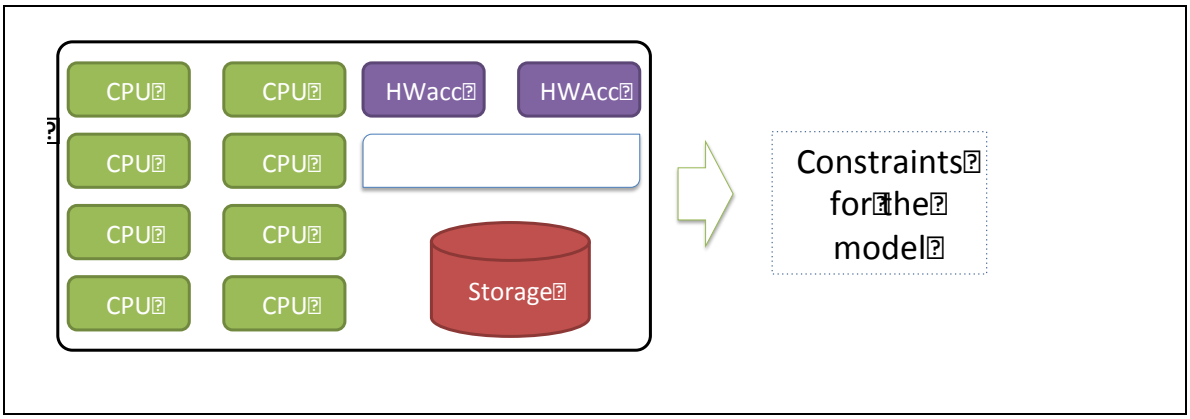


Figure 9: Micro-server Model

In addition, the virtualised infrastructure represents an important part of the power consumption of the whole system. We model the power consumption of the CPUs that compose each micro-server to be dependent of the usage percentage, as modelled for every VNF subject to the maximum aggregated flow to be supported. Table 2, which follows, shows an example of a CPU power model. The energy consumption within each micro-server is then modelled as the sum of the individual consumption of each CPU.

CPU usage	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Power consumption (W)	5	30	40	55	70	85	100	120	130	139	145

Table 2: Example of CPU power model

Besides, virtualised resources also include other hardware appliances, such as hardware accelerators (HWA) that can improve the performance of a VNF, in terms of latency. This way, the additional appliances help matching SLA with the tenant at a higher energy cost. As shown in Figure 10, the use of hardware acceleration involves two main implications. *On the one hand*, if there is a VNF that is dimensioned to use a certain number of cores and the optimization algorithm decides that it will be executed by employing HWA, then the VNF instance will only use one core and the HWA appliance. *On the other hand*, the use of HWA implies not only higher cost of the infrastructure but also higher power consumption (up to N times the power consumption of the regular execution without HWA).

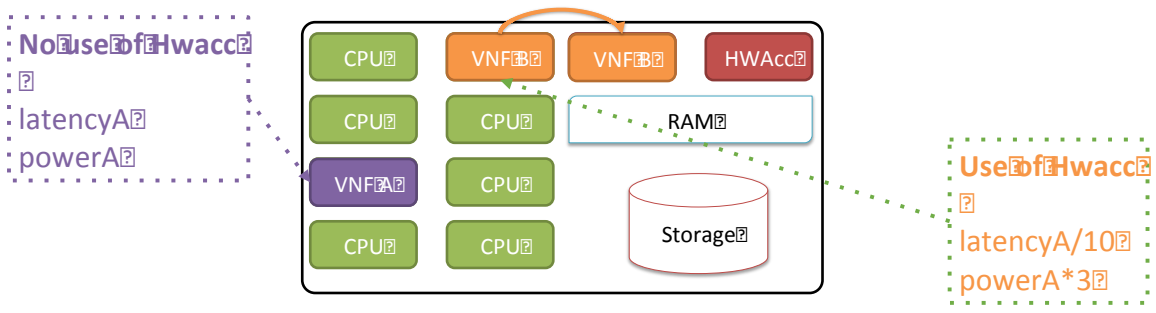


Figure 10: Use of HW accelerators

Finally, the last piece of the problem to be modelled is the network topology. Both the latency introduced by the connection of VNFs of the same chain allocated in different SCs, and the power consumption of the network elements depend on the sum of aggregated flows that traverse those network elements. In particular, the transmission latency along the network topology between two VNFs allocated in different micro-servers depends on the traffic load of the involved links and the size of the flow to be transmitted. In addition, the power consumption of a network switch is modelled as the sum of the power related to every active port and the entire traffic load that traverses the switch. Figure 11 illustrates the described model for a network switch and Table 3 shows an example of the power consumption model

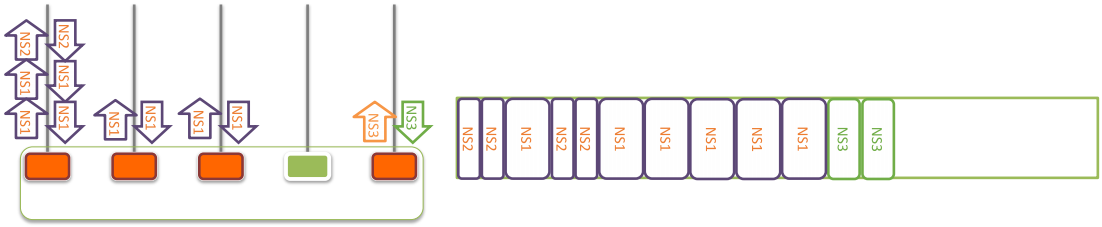


Figure 11: Switch model

With this information, the placement algorithm assigns each VNF instance to one or more VMs with the objective of minimizing the energy consumption of the whole system while matching the latency constraints for the NS and considering the available resources. The latency of each NS deployed for every tenant is the sum of the processing latencies of the VNFs that compose the service chain, plus the latencies introduced by the transmission of the user flows through the VNFs allocated in different CESCs. In addition, the total power consumption of the system is calculated as the sum of the consumptions of the CPUs allocated to the VMs that run the VNFs, the power consumption of the SC chassis and the power consumption of the switch.

CPU usage	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Power consumption (W)	0	1	2	3	4	5	7	10	12	14	15

Table 3: Example of switch power model

2.4.3. Development of the algorithm

To solve the problem we consider a Robust Optimization approach using constraint programing to develop the algorithm and then to solve the problem. With the aim of developing the placement algorithm, we split the problem in different iterations that simplify the discussed original problem, such that every iteration introduces more complexity to the model.

In the *first iteration*, we develop a simple placement model, as follows: each VNF uses a fixed amount of resources per user and the latency of each link is also fixed. This means that the model does not introduce variability on the demand load. All the links accept aggregated flows until the bandwidth limit is reached. We previously described that each NS has to start and end with a SCVNF but, in this iteration, we do not make any distinction between SC-VNF and service VNFs. In addition, the model does not include HW accelerators.

The *next iteration* introduces variability in the load in order to insert more complexity to the placement algorithm. Hence, VNFs are modelled to use resources (e.g., %CPU) according to the

traffic load (aggregated bitrate server). With the aim of introducing uncertainty to the problem we use Robust Optimization. In order to do that, we define a deviation/protection parameter. The aggregated user bitrate that is used to calculate the power consumption and the latency, is now the nominal value that includes the deviation. In this iteration, the NS model is conformed like a Circular VNF chain; each chain starting and ending with a SC-VNF that is not shared among NSs of the same tenant. In this iteration, we introduce the use of HW accelerators.

Finally, in the *third iteration* all the NSs belonging to the same tenant use a common SC-VNF. For this reason, the use of resources of the SC-VNFs must then consider the sum of the aggregated flows of the NSs of the same tenant. Note that SC-VNFs do not use HW accelerators. In addition, we introduce variability on the link latency with the load of the link and the size of the flow.

2.4.4. Problem modelling

As stated before, the VNF placement problem addressed in this deliverable focuses on the assignment of the VNFs composing each NS to the provided 5G network infrastructure. Therefore, it constitutes a discrete -or combinatorial- optimization problem, where the minimization of energy consumption constitutes the optimality criterion. Table 4 reports and summarizes the mathematical notations used for the model indexes, sets, input parameters and decision variables of VNF placement problem that is defined in Table 5.

We consider a set X of small cells that compose the CE-RAN and a set Y_x of cores included in the micro-server of the SC x. Additionally, we also consider a set of I network services offered by the CE-RAN operator to the different tenants and a set J_i of VNFs that form the functional chain of the NS i. Finally, R represents the set of different resources available in the micro-servers for the execution of the VNFs (storage, RAM, hardware accelerators, etc.). Hence, our constraint based expert system must determine in which VM y of a SC x is each VNF j of a NS i located.

Let Π denote the set of all possible power-aware and latency- and resource-constrained VNF placement policies, which decide the mapping of each VNF of available NSs to available VMs in the provided SCs. We formulate our VNF placement optimization problem aimed at minimizing power consumption with delay requirements as (1).

In reference to the energy part, we define the total power consumption P^{π} as the sum of the energy demand of all the allocated VNFs $P^{\pi}_{i\ j}$ (5). Then, for each VNF instance, the power consumption is expressed in (6) as the sum of the power consumption of VNF j of NS i due to the CPU use of VM y of SC x, plus the power consumption of the physical switch due to the forwarded traffic of the VNF, plus the power consumption of SC x because of being switched on.

The power consumption of the cores that run the VMs depends on the aggregated user traffic served by the correspondent VNF, and this relationship may be non-linear. It must be also understood that when a VNF instance scales to a higher number of cores, the individual load of the involved CPUs decreases, thus leading to lower power consumption per core. Additionally, the energy consumption of the networking devices (e.g., an Ethernet switch) depends on the aggregated flow that traverses the device to forward data packets between SCs and the number of active ports. Finally, note that the power spent because of the SC being active is constant, not affecting the optimization results. Besides, we express per-NS delay in (7), as the sum of per-VNF delay, and the per-VNF delay in (8) as the sum of the processing delay introduced by the execution of the VNF instance, plus the path delay introduced by the traffic going from one VNF to the next one in the chain defined in the correspondent NS.

Indexes	
π	Index for VNF placement policy
x	Index for CESC microserver
y	Index for core number in each microserver
i	Index for NS
j	Index for VNF in a Network Service chain
r	Index for physical resources in CESC microserver
Sets	
Π	Set of possible VNF placement policies
X	Set of CESC microservers
Y	Set of cores composing a CESC microserver
I	Set of offered NSs
J_i	Set of VNFs in NS i
R	Set of resources available in each CESC microserver
Input parameters	
T_i	Contracted maximum aggregated user traffic for Network Service i
D_{max_i}	Maximum latency allowed for NS i
M_{rx}	Amount of resource r available in CESC microserver x
Psc_x	Power consumption of SC x because of being switched ON
$Pcpu_{ij}^{xy}$	Power consumption of VM y of SC x due to CPU use of VNF j of NS i
$Pswitch_{ij}$	Power consumption in the physical switch due to CPU use of VNF j of NS i
Decision variables	
P	Total power consumption
P_{ij}	Power consumption of VNF j of NS i
m_{rx}	Consumption of resource r in SC microserver x
tl_x	Traffic of the link that connects SC microserver x with the switch
A_{ij}^{xy}	Binary variable that expresses the assigment of VNF j of NS i to VM y in SC x
L_{ij}	Binary variable to express the use of the network switch to forward the flow of VNF j of NS i to the next VNF.
d_i	Delay of NS i
d_{ij}	Delay of VNF j of NS i
dp_{ij}^{xy}	Processing delay in the CPU of VNF j of NS i in VM y of SC x
$dl_{ij}^{x_1y_1,x_2y_2}$	For NS i path delay between VNF j (located in VM y_1 of SC x_1) and VNF $j+1$ (located in VM y_2 of SC x_2)
$d_{vs,xy}$	Link delay between the virtual switch in SC x and the VM y in SC x
$d_{vs_x,s}$	Link delay between the virtual switch in SC x and the physical switch
d_x^{vs}	Delay in the virtual switch of SC x
d_s	Delay in the physical switch

Table 4: Summary of model indexes, sets, parameters and decision variables

On the one hand, the processing delay in the CPU grows with the supported user traffic load. *On the other hand*, we consider that the path/link delay is composed of several link delays from/to VMs to/from the switches plus the delay in the switches.

Hence, depending on the adopted placement policy π, each NS will follow a different path through the network. In such a way, if the VNFs belonging to a NS are placed in the same SC, the service chain traverses virtual/internal links and the virtual switch of the selected SC; whereas if the VNFs of a NS are located in different SCs external links and the physical switch is also used. Therefore, generally, $dl_{ij}^{x1y1,x2y2} = d_{x1y1,vs} + d_{x1}^{vs} + d_{vs(x1),s} + d_s + d_{s,vs(x2)} + d_{x2}^{vs} + d_{vs,x2y2}$; but for a given VNF if the next VNF in its chain holds in the same SC, its link delay is only affected because of passing through the virtual switch. Note that for the last VNF in the chain the link delay is null, being the sum referring to the link delay for that j is null.

In order to guarantee that each VM holds only a single VNF, constraint (9) must be satisfied. Nevertheless, we consider that each VNF of a NS can be instantiated as a VM that is executed over several cores of the same SC. Apart from that, the first VNF and the last VNF in a service chain are the same (the SC VNF) in order to perform the specific functions that allow the split between control and data plane required in 5G communications. Therefore, it implies that Axy = AxJi .

The energy-aware VNF placement optimization problem

$$\min_{\pi \in \Pi} P^{\pi} \tag{1}$$

s.t.

$$d_i^{\pi} < D_{max_i} \qquad \forall i \in \mathcal{I}, \pi \in \Pi \tag{2}$$

$$m_{rx}^{\pi} < M_{rx} \qquad \forall r \in \mathcal{R}, x \in \mathcal{X}, \pi \in \Pi \tag{3}$$

$$tl_x^{\pi} < BW_x \qquad \forall x \in \mathcal{X}, \pi \in \Pi \tag{4}$$

$$P^{\pi} = \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}_i} P_{ij}^{\pi} \tag{5}$$

$$P_{ij}^{\pi} = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}_x} Pcpu_{ij}^{xy} A_{ij}^{xy} + \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}_i} Pswitch_{ij} L_{ij} + \sum_{x \in \mathcal{X}} Psc_x A_{ij}^{xy} \tag{6}$$

$$d_i^{\pi} = \sum_{j \in \mathcal{J}_i} d_{ij}^{\pi} \tag{7}$$

$$d_{ij}^{\pi} = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}_x} dp_{ij}^{xy} A_{ij}^{xy} + \sum_{x_1 \in \mathcal{X}} \sum_{y_1 \in \mathcal{Y}_{x1}} \sum_{x_2 \in \mathcal{X}} \sum_{y_2 \in \mathcal{Y}_{x2}} dl_{ij}^{x_1 y_1, x_2 y_2} A_{ij}^{x_1 y_1} A_{ij+1}^{x_2 y_2} \tag{8}$$

$$\sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}_i} A_{ij}^{xy} = 1 \qquad \forall x \in \mathcal{X}, y \in \mathcal{Y}_x \tag{9}$$

$$m_{rx}^{\pi} = \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}_i} \sum_{y \in \mathcal{Y}} m_{ij}^{r} A_{ij}^{xy} \tag{10}$$

$$tl_x^{\pi} = \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}_i} \sum_{y \in \mathcal{Y}} T_i A_{ij}^{xy} A_{ij+1}^{x'y} \tag{11}$$

Table 5: Placement problem modelling

Finally, the model must also include the constraints referred to the available resources. In this sense, equation (3) states that the consumption of the resource r due to the placement of VNFs in the micro-server of the SC x, m_{rx}^{π}, must not exceed the total amount of resource M_{rx} , where m_{rx}^{π} is defined in (10). Similarly, equation (4) constrains the traffic through the network links tl_x^{π} to the maximum capacity of the links BW_x, as the sum of the traffic flows of the network services that must be forwarded from a VNF in one SC to the next VNF in another SC.

2.4.5. Robust Constraint-Based Solution

As previously mentioned, previous solution and other related works assume perfect knowledge on input parameters leading to the formulation of a deterministic optimization problem. Unfortunately, in the real world, the estimated values of the input parameters may differ due to biased data, unrealistic assumptions or numerical errors, thus affecting the obtained optimal

solution and its performance. This potential deviation of the nominal or expected input may lead to the violation of the problem constraints and, *therefore*, make the obtained solution suboptimal or even unfeasible.

The uncertainty of modelling parameters has been traditionally handled through stochastic programming and sensitivity analysis, but Robust Optimization techniques have recently appeared as a powerful tool to manage the impact of uncertain input sets.

RO seeks an uncertainty-immunized solution with an acceptable performance under the realization of the uncertain inputs, becoming a conservative, worst-case oriented methodology. The main tools of RO are uncertainty sets and a robust counterpart problem [23]. The uncertainty of the input data is described later in this section, while the robust counterpart problem is the deterministic model previously detailed in Section 2.4.4.

Here we propose a robust constraint-based placement solution that deals with service demand uncertainty. With this RO approach, we consider a trade-off between the problem optimization and the protection from deviations caused by input parameters uncertainty. This uncertainty in service demand has an impact on several components on the formulation previously presented in subsection 4.1. *On the one hand*, we consider that a VM that is running a specific VNF requires an expected use of CPU and, thus, certain constant power consumption due to CPU. Nevertheless, the uncertainty in traffic demand may cause uncertainty in the CPU requirement/use and, consequently, on its energy consumption and pro- cessing delays. *On the other hand*, both energy consumption and delays in switches and links depend on that demand uncertainty as well.

In order to obtain a proposal for the placement problem with uncertain service demand, we employ the Soyster [22] method used in the framework of RO. That technique is also known as the worst-case approach, where the solutions are achieved by using the most extreme expected values of the uncertain variables. Therefore, that kind of resolution guarantees feasible solutions for any value of the uncertain service demand.

In reference to the modelling of the uncertain traffic demand per NS_i, we consider an expected traffic demand, $E[T_i] = T_i$, plus a term due to uncertainty in the service demand, $UT_{max\ i}$. We assume that the random variable $UT_{max\ i}$ is symmetrically distributed between $[-\Delta T_i, +\Delta T_i] \cdot T_i$ and with mean zero. Furthermore, the sum of deviations of the uncertain service demand should not exceed the worst-case maximum value $T_{max\ i}$, fulfilling (12). In this way, the value of $T_{max\ i}$ controls the trade-off between the robustness and the impact on the optimization; higher $T_{max\ i}$ leads to worse objective function values, but protecting from more parameter deviations.

$$\frac{UT_i}{\Delta T_i} \leq T_{max_i} \qquad \forall\, i \qquad (12)$$

3. Evaluation of the algorithm performance

3.1. Scenario description

Our evaluation scenario is depicted in Figure 12. We consider a 10 SC deployment, e.g. to cover a football stadium for periodical flash events. The SCs are connected in a star topology through an Ethernet switch.

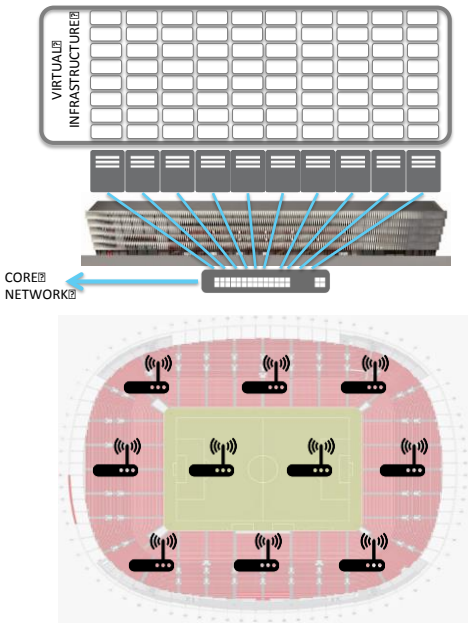
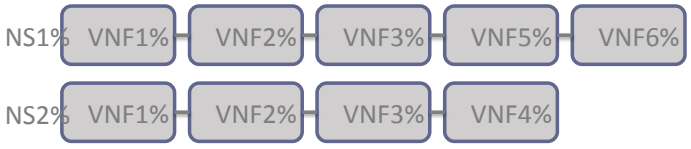


Figure 12: Evaluation scenario

Each SC is composed of 8 cores, 16 GB RAM, 100 GB of storage and 1 HWA (e.g. GPU). We also consider 2 NSs that can be offered at 3 service flavours (bronze, silver and gold) that determine the latency constraint for each NS and the nominal aggregate user bit rate served, as shown in Table 6.



	SERVICE'FLAVOUR		
	BRONZE'	SILVER'	GOLD'
NS1'	Max%at%%150%ms% Max%low%%130%Mbps%	Max%at%%120%ms% Max%low%%130%Mbps%	Max%at%%100%ms% Max%low%%130%Mbps%
NS2'	Max%at%%110%ms% Max%low%%90%Mbps%	Max%at%%90%ms% Max%low%%90%Mbps%	Max%at%%80%ms% Max%low%%90%Mbps%

Table 6: Description of NSs and service levels

Table 7 shows the VNF modelling of the VNFs that compose the proposed NSs for the user traffic demand that will be employed in the evaluation experiments. The user traffic values include the nominal bit rates of services NS1 and NS2 (130 Mbps and 90 Mbps, *respectively*), as well as the values when a 20% protection parameter is applied to the nominal values (156 Mbps and 108 Mbps, *respectively*) in order to study the effect of robust optimization techniques on the placement decision. As a final remark, the placement behaviour of VNFs can be altered by the use of HWA. The assignment of a HWA to a VNF implies that the service latency of the correspondent instance is reduced 3 times, at a cost of incrementing the power consumption by 10. It is important to note that when the instance needs more than 1 CPU to execute without the assistance of a HWA, then, if the placement algorithm assigns it to a HWA, the VNF will need a single core.

		VNF1	VNF2	VNF3	VNF4	VNF5	VNF6
Nº of cores per instance	90 Mbps	1	2	1	1	1	1
	108 Mbps	1	2	2	1	1	1
	130 Mbps	1	2	2	1	1	1
	156Mbps	1	2	2	1	1	1
CPU usage	90 Mbps	30%	20%	90%	50%	50%	20%
	108 Mbps	40%	30%	60%	60%	60%	30%
	130 Mbps	40%	30%	60%	60%	60%	30%
	156Mbps	40%	30%	90%	40%	20%	30%
Service latency	90 Mbps	10 ms	15 ms	30 ms	35 ms	20 ms	25 ms
	108 Mbps	10 ms	15 ms	35 ms	40 ms	25 ms	30 ms
	130 Mbps	10 ms	15 ms	35 ms	40 ms	25 ms	30 ms
	156Mbps	12 ms	16 ms	40 ms	50 ms	30 ms	35 ms

Table 7: VNF characterisation for the service flavours of the evaluation scenario

3.2. Simulation results

As stated before, the evaluation scenario includes 3 tenants that hire a combination of network services /flavours to the SC operator. In this way, in the following, first we are going to evaluate the behaviour of the placement algorithm for the three service levels and, next, we will show the effect of applying RO techniques to those results.

3.2.1. Bronze flavour scenario results with 0% robust protection

Figure 13 shows the placement outcome when all the network services operate at bronze flavour. Each tenant has been assigned a different colour in order to identify the VNFs that belong to its NS, and different tones distinguish NSs of the same tenant. As a result of the placement algorithm, all the NSs are allocated in the available virtualized resources with an energy consumption cost of 3712 W, meeting all the latency constraints. The figure displays the assignment of VNFs to the SCs, highlighting some placement particularities. As already explained

in previously, the VNF chains of the same tenant share their SCVNF. In our example, Tenant 1 and Tenant 2 have both two NSs that are served with a single SCVNF. On the other hand, each CPU executes a single VM, but some VNF instances need two cores to manage the user demand. In connection with this behaviour, the VNF scalability feature is also visible in Figure 13, by observing the placement of, *for example*, VNF3 for the different NSs. NS1, with a higher aggregated user demand needs two CPUs to instantiate VNF3, but since NS2 operates at a lower user demand, the instantiation of VNF3 needs just a single core. Note, as well, that bronze service flavour does not require HW acceleration to meet the latency constraints of the NSs. Finally, it can also be observed that the algorithm tries to group as many VNFs as possible in each server, in order to reduce the data traffic across the switch, minimize its load and, *therefore*, its energy waste.

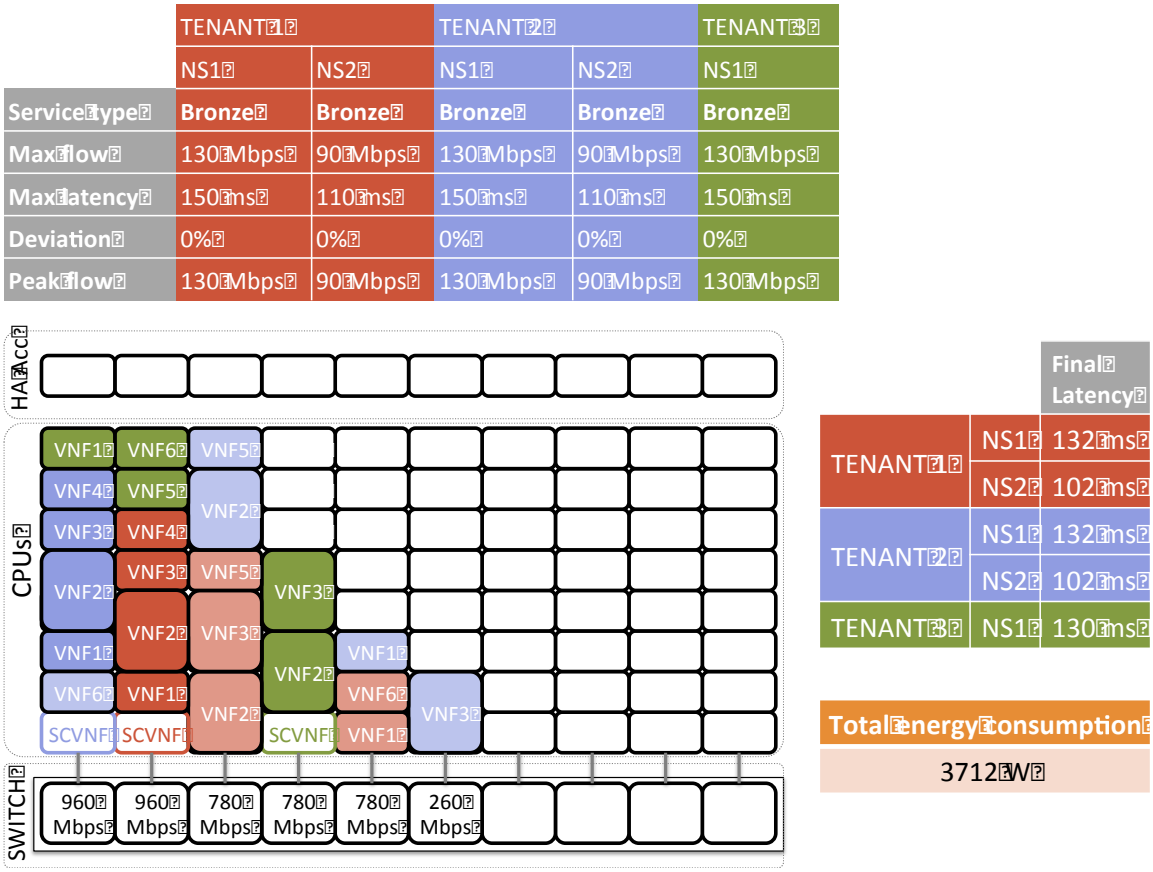


Figure 13: Placement results of bronze setting with 0% robust protection

The following evaluation experiment analyses the behaviour of the placement algorithms when the latency constraints become stricter with silver service flavour. The further limitation of maximum latency values forces the algorithm to make use of HW acceleration units. Figure 10 shows that all the NSs must use HW acceleration in one of the VNFs composing the service chain. Obviously, the resulting total latency of the NSs decreases to "meet" the service level requirements but, *in return*, the global power consumption grows up to 4547 W, a 22% more. Apart from that, the placement results illustrate the same behaviour as the previous example regarding the allocation of VNFs in the available resources, scaling features and multi-tenancy.

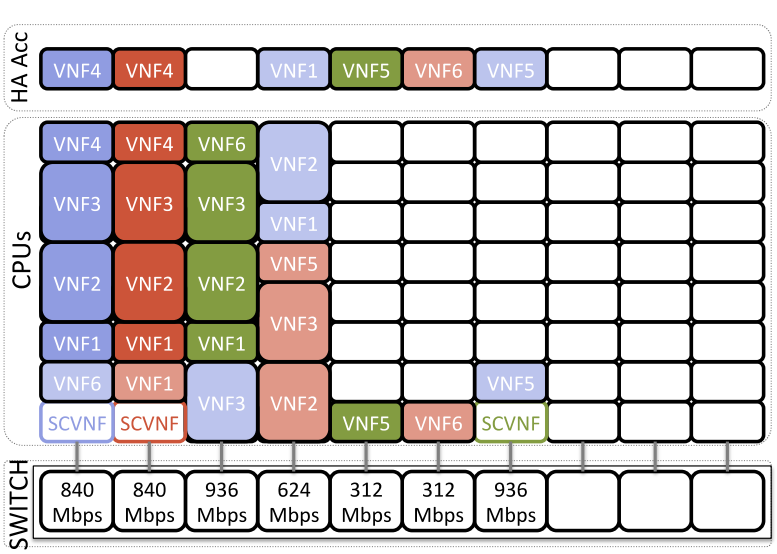
The aforementioned evaluation examples work with nominal aggregated user traffic to execute the optimization algorithm. However, a perfectly known user traffic demand is seldom available. The usual case is the existence of a certain level of uncertainty in the behaviour of the user demand. Robust optimization techniques allow the placement algorithm to include a defined level of uncertainty and protect the resultant system against the adverse effects of demand

peaks. The following evaluation examples include a robust protection parameter and analyse its impact on the global power consumption.

3.2.2. Silver flavour scenario results with 20% robust protection

This section analyses the placement results of the evaluation scenario exposed in the previous section, but now introducing a 20% robust protection parameter. This protection parameter implies an increase of the 20% of the aggregated user traffic as an input of the algorithm, in order to be prepared to eventual demand peaks during the service activity. As a result of the traffic increase, Figure 14 shows that VNF3 of NS2 must scale and needs two CPUs for each instance. The traffic increase also affects the processing latency of the VNFs causing a modification in the use of HW accelerators.

	TENANT 1		TENANT 2		TENANT 3
	NS1	NS2	NS1	NS2	NS1
Service'type'	Silver'	Silver'	Silver'	Silver'	Silver'
Max'flow'	130 Mbps	90 Mbps	130 Mbps	90 Mbps	130 Mbps
Max'latency'	150 ms	110 ms	150 ms	110 ms	150 ms
Devia4on'	20%	20%	20%	20%	20%
Peak'flow'	156 Mbps	108 Mbps	156 Mbps	108 Mbps	156 Mbps



		Final' Latency'
TENANT 1	NS1	116 ms
	NS2	79 ms
TENANT 2	NS1	110 ms
	NS2	79 ms
TENANT 3	NS1	118 ms

Total'energy'consump4on'
5047 W

Figure 14: Placement results of silver setting with 20% robust protection

At the same time, the different use of the HW acceleration units has as consequence that some VNFs that required 2 CPUs for their execution in the previous evaluation scenario now employ a single core in combination with the HWA. Therefore, in some cases, the higher power consumption that is implicit to the use of HWAs, may be balanced with a decrease on the number of CPUs required for the VNF. In addition, the higher aggregated user traffic involves a more intense use of the network switch. These factors have an impact on the power consumption of the components of the network, resulting in a global energy consumption of 5047 W, which suppose an increment of 11% compared to the unprotected placement.

3.2.3. Gold flavour scenario results with 20% robust protection

Finally, the last evaluation scenario imposes even more strict latency constraints to the NSs to be placed and includes a 20% robust protection parameter. Figure 15 shows the results of the placement algorithm with this setting. Again, the reduction of latency values forces the increment of the use of HW acceleration.

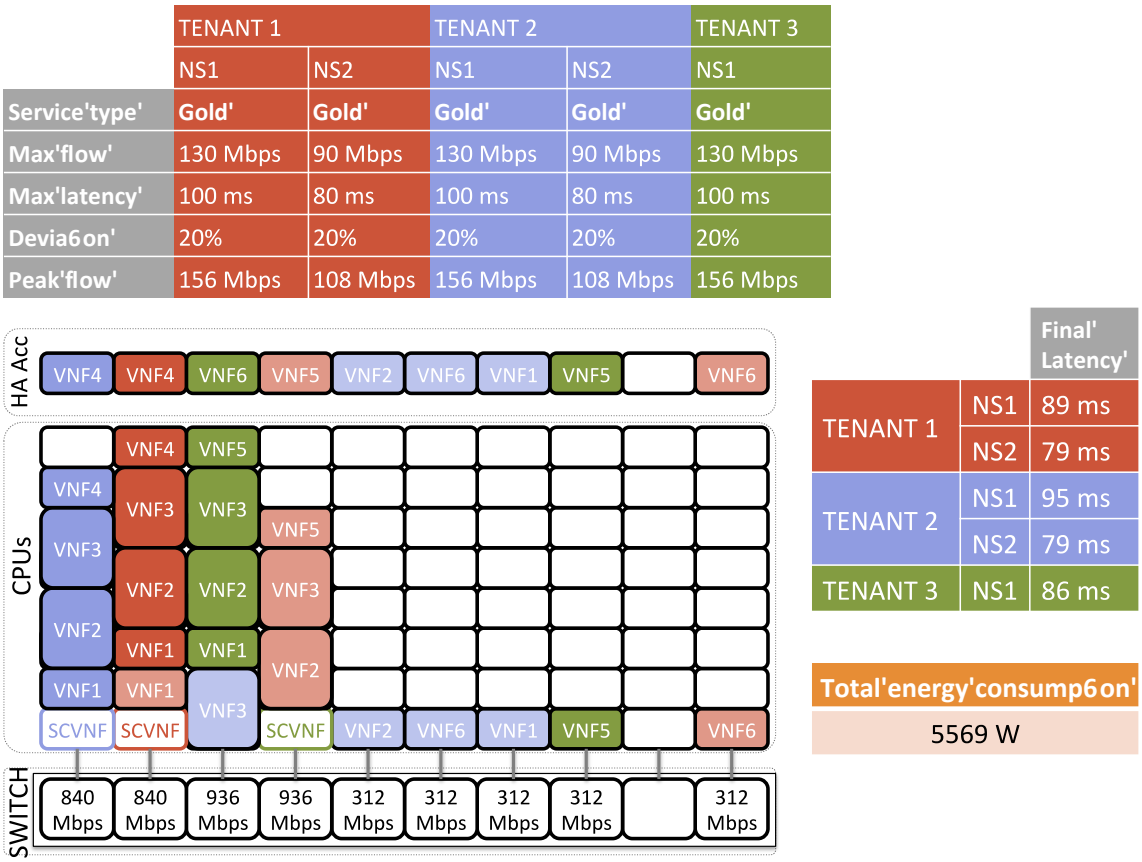


Figure 15: Placement results of gold flavour setting with 20% robust protection

In this case, the main the consequence is the spreading of the VNFs across the SCs to use the single HWA available in each cell. The use of HWAs, the increase of power consumption due to higher user aggregated traffic and the higher network traffic traversing the switch cause the global energy.

3.2.4. Conclusions

As a final remark, Figure 16 shows the comparison of energy consumption and use of resources (cores, active CESCs and HWA) related to the robust protection level in a wider collection of scenarios. In particular, the charts of Figure 16 compare the results of the placement algorithm for the three service levels described in the previous sections (bronze, silver and gold) combined with three robust protection levels (0% or no protection, 10% and 20%).

Obviously, the main trend is that both power and resource consumption increase with higher values of protection, but the associated cost can be assumed in favour of reducing the impact of unexpected user demand peaks.

However, there are some cases in which the protection parameter has not so adverse implications. For example, it can be observed how the number of used cores in the silver service level with a 10% protection level is lower than in the case of no robust protection.

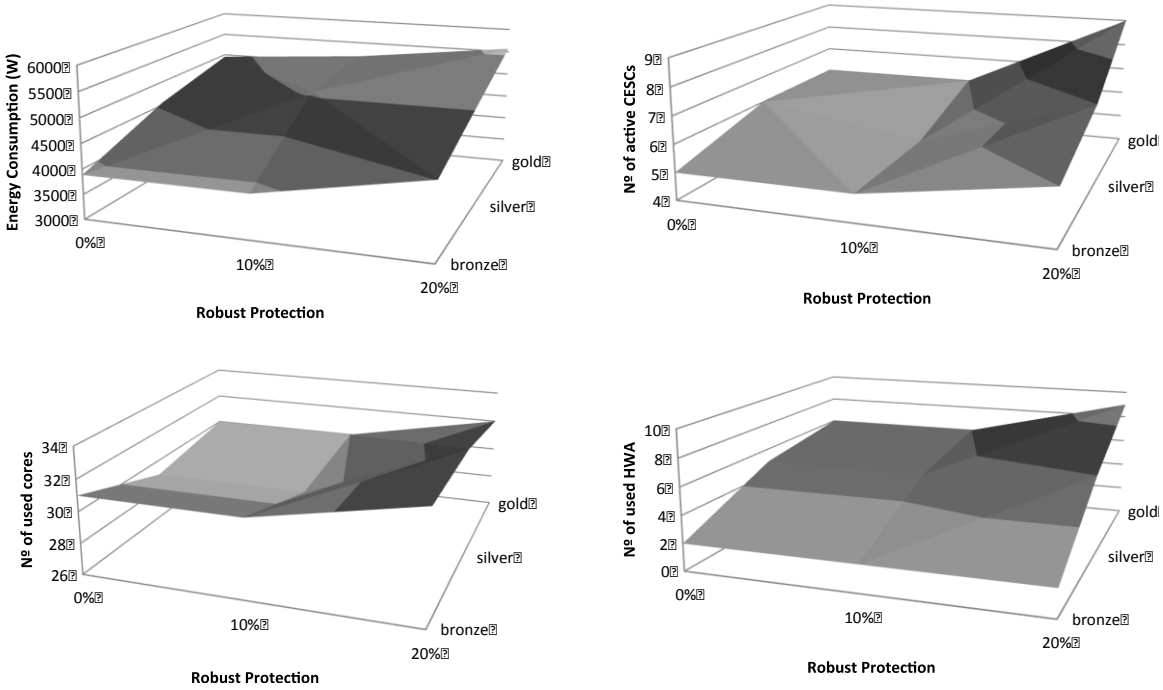


Figure 16: Comparison of results regarding to service flavour and protection level

The reason of this behaviour is how the algorithm uses HWA appliances in this case. The increase of the user traffic demand in the protected services implies the scaling-up of some VNFs that now must use more cores for the operation of the NSs. Then, the placement algorithm decides to use HW accelerators for those specific VNFs, compensating in this way the higher energy cost of HWAs with a lower use of CPU cores. With all these considerations that can be extracted from a deeper analysis of robust placement results, robust optimization can be applied to decision-making problems, which does not end just with the optimization problem.

The proposed steps in the proposed approach can be used in any placement decision-making context and its goal is to provide a "tool" that helps system administrator to evaluate the possible impact of any decision on the performance of the deployment underneath.

4. Implementation approach

Within this scenario, one of the main challenges to be overcome is the automated management of the service lifecycle and provision. This can be done by exploiting the capabilities of the general purpose software tool used to manage the general hardware to provide these new services. OpenStack software[17] is used as the Virtual Infrastructure Manager (VIM) to provide Infrastructure-as-a-Service (IaaS) capabilities. Coming from the Cloud perspective, this implies that OpenStack is designed to provide a transparent view of the set of available computing nodes (i.e. nodes running the OpenStack Compute software), to the tenant.

To this end, the OpenStack project has designed Nova[18], as a tool to provide personalized and elastic services of compute resources. Compute resources are represented by host compute; the term of "host" means a physical node that has a nova-compute service, on which VMs can be provided. The platform provides on-demand access to compute resources by provision and manage of VMs. In the event of a new VM -or VNF- deployment request, the tenant does not need to know -in fact typically does not care- in which node of the whole infrastructure the VM is going to be launched as long as it covers the hardware demands.

4.1. Location of the module/ Infrastructure as a Service

OpenStack is a cloud operating system (open source software) that allows managing and deploying a cloud Infrastructure as a service platform. OpenStack is highly configurable, formed of several components that can independently be customized proving several ways to use OpenStack. Figure 17 depicts the independent modules that compose the platform.

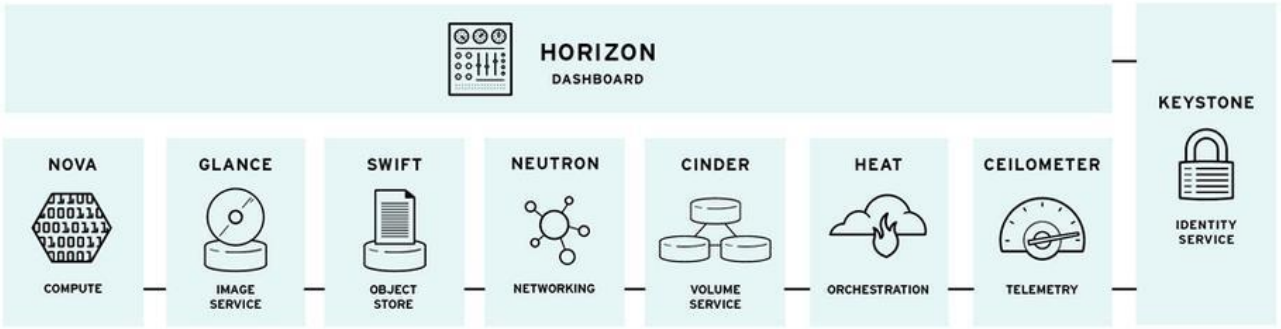


Figure 17: OpenStack projects

This architecture aims to benefit public and private clouds, by minimizing the total cost of ownership. For that purpose, one "key" aspect is to contain hardware expenses by managing resources efficiently, meeting user service level requirements. In the SESAME context, a multi-tenancy scenario where compute nodes are scattered across a distributed network has been proposed. This approach can be performed by Nova compute nodes located on top of CESC micro-servers.

Nova is comprised of multiple server processes, each performing different functions. The user interface is a REST API, while internally Nova components communicate via an RPC message passing mechanism. The following diagram shows the key components of Nova architecture.

[17] For further details also see: https://www.openstack.org/software/
[18] For further details also see: https://docs.openstack.org/nova/latest/

The modules are custom-written Python[19] daemons[20] on top of Linux servers to provide the services. Nova is organised as Web Server Gateways Interface to receive and response calls, and worker daemons that performed the orchestrations tasks. Two additional pieces, messaging queue and the database, that facilitate the asynchronous orchestration of complex tasks.

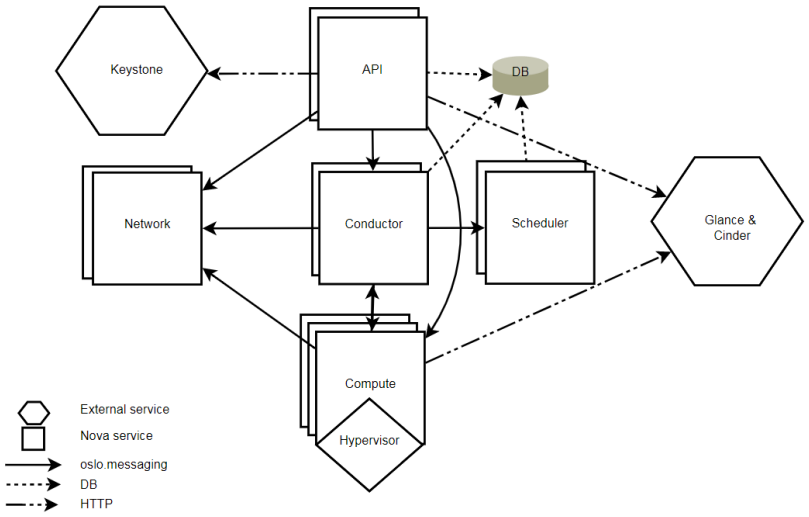


Figure 18: Nova Logical Architecture

The sub-modules of the nova application and function are described as follows (Figure 18):

- Nova-Conductor: This is the Remote Procedure Call (RPC) server; it acts as a broker, handles the requests (deployment, updates…) from a compute node, and performs database query and statistics.
- Nova-API: It is the service front API that accepts and responds to end-user compute and volume.
- Nova-Network: It handles networking aspect, when the Neutron project[21] is not used by the OpenStack deployment.
- Nova-Scheduler: This decides the right host for placing VMs.
- Nova-compute is the process in which VMs are created and terminate, by the route passing through the hypervisor.

The following section will describe how implementing service function chaining mechanism, add extra features and slightly modify the original architecture of Nova to satisfy resource requirements and placement policies of a VM.

4.2. Initial placement

The perspective studied in SESAME, *however*, differs from the traditional Cloud environment. Because of the virtual infrastructure, it is composed of a set of CESCs distributed over a specific area, the localization of end-users connected to the platform, and therefore the CESCs that have

[19] For further details also see: *https://www.python.org/*
[20] Also see: *https://stackoverflow.com/questions/473620/how-do-you-create-a-daemon-in-python*
[21] For more details see: *https://wiki.openstack.org/wiki/Neutron*

more active users, it becomes relevant, e.g. to reduce delay by placing VNFs as closest as possible to end-users, to redistribute the location of existing VNFs to reduce energy consumption.

Due to the low-level requirements that are considered to deploy the VNFs in terms of information about the status of the infrastructure, there has been decided to enhance OpenStack's Nova scheduler algorithm with the logic needed for the SESAME scenario.

Nova scheduler module takes two steps to determine the appropriate host, (i.e. compute node), in which to deploy a VM [24]. First, it applies a series of filters to the set of available compute nodes to eliminate those hosts that do not comply with the requirements of the VM, for example in terms of available memory or due to lack of specific hardware. The nodes that result as output of this process are then weighted based on configurable metrics, yielding an ordered list of candidate hosts depending on suitability (Figure 19).

4.3. Extended OpenStack architecture

In the context of SESAME, a placement algorithm has been developed to appropriately manage the virtualized resources to "meet" agreements of service, as a customization of OpenStack compute scheduler. OpenStack provides a heavily customized and hooks/plugins/custom for a compute scheduler. The algorithm has been developed as compute-node filter Scheduler, which supports filtering and weighting to take the decisions on where the new instance should be placed. The algorithm has been developed in a way that it is plugged it in trough configuration.

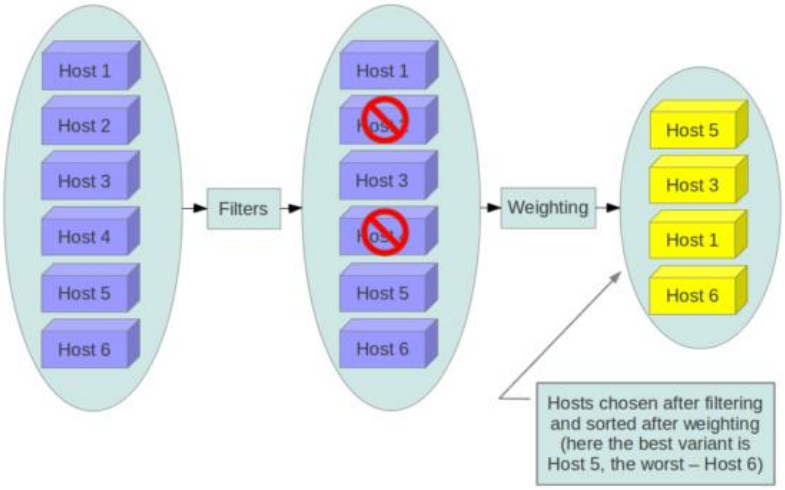


Figure 19: Filtering workflow

Following the design of OpenStack architecture, in order to extend the functionality of the stock filters, the implementation needs to inherit from the abstract class BaseHostFilter, where the host_passess method has to be implemented; this method returns, in binary, the hosts that has passed the filter based on two parameters (host_state and filter_properties dictionary), the filter will reject the host not considered.

```
--scheduler.driver=nova.scheduler.FilterScheduler
--filter_scheduler.available_filters=nova.scheduler.filters.all_filters
--filter_scheduler.available_filters=myfilter.MyFilter
--filter_scheduler.enabled_filters=RamFilter,ComputeFilter,MyFilter
```

Figure 20: Nova Configuration

A two-steps configuration is needed to import the custom filter, first the implemented code needs to be place inside the nova/scheduler/filters catalog, and etc/nova/nova.conf needs to be updated with the new configuration; this *".ini*" file is copied in every compute node. This file contains the configuration options that will run the nova-* services (Figure 20).

This configuration allows nova to use the FilterScheduler, default and customized filters, for the scheduler driver. The RamFilter, ComputeFilter, and custom filter are used by default when no filters are specified in the request. The order the filters has been defined in the configuration parameters, defines the iteration over the filter scheduler to endorse the possible hosts, this procedure sift out based on criteria of reducing the non-valid hosts for placement.

After the filter scheduler is configured, and upon a new request for a resource, it first applies filters to determine which hosts are eligible for consideration when dispatching a resource. Filters are binary: either a host is accepted by the filter or it is rejected. Hosts that are accepted by the filter are then processed by a different algorithm to decide which hosts to use for that request. The scheduler iterates over the available hosts, by repeated filtering and weighing, to provide a list of accepted hosts (Figure 21).

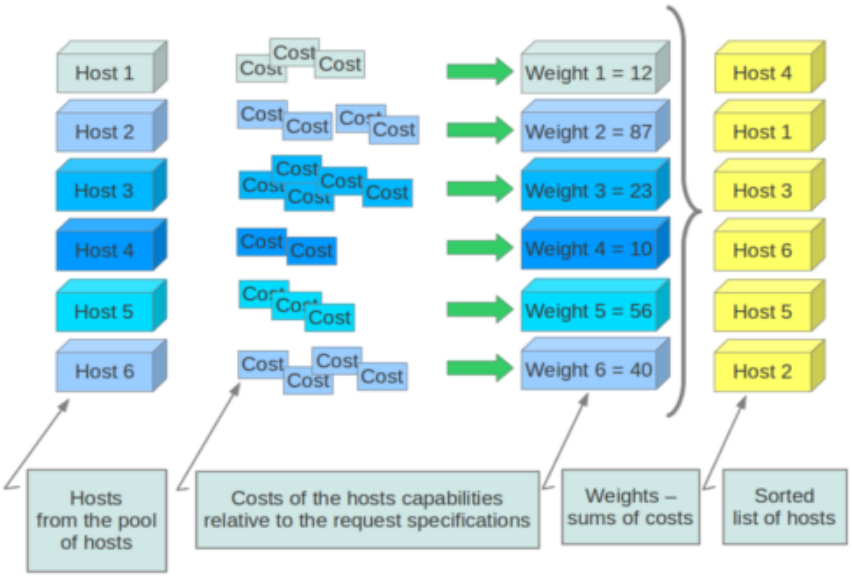


Figure 21: Weighting hosts

In accordance with SESAME Nova Scheduler, it is not possible to determine the optimal placement for each VM composing a NS. The entity responsible for the execution of the

placement algorithm needs to access the deployed NS and the current state of the NFV Infrastructure in order to infer the optimal placement for the VMs.

To perform the solution, two new elements need to be introduced in the architecture. A Placement module in collaboration with the MANO and a new scheduler filter, which acts as a client, must request (from the proposed placement module) the host where it is most optimal to deploy the VMs (of the requested NS).

The proposed placement module inside TeNOR[22] (Figure 22) receives the NS instantiation requested by each tenant. TeNOR has the knowledge of the status of each NS deployed in the NFVI. The placement module has to communicate with the NS manager and the VNF manager to know the actual status of NS and VNF deployed. In conjunction with the knowledge of the NFV Infrastructure, the descriptions contain the design constrains for the placement algorithm.

With this knowledge the placement module infers the placement decision. The proposed algorithm is deployed as a service inside TeNOR, however the solution needs to be consumed by the Nova Scheduler. For this communication, a server/client scheme is proposed.

This service would receive requests from Nova Scheduler. Therefore, a new scheduling weighting filter is needed in order to program the request and obtain the outcome from the placement algorithm. This filter has no intelligence, only interacts with the proposed placement server.

VIM has no knowledge of the correspondence between the NS and the requested VMs it corresponds to. The client implemented inside Nova Scheduler is not able to request the location for a specific NS. Therefore and to aim synchronization, the service is blocked during request service time; in other words, a server can have only one request at a time. When the request is finished, the server is unblocked and it can continue to serve a new request. The placement module must correlate the request of a chain of VNFs with the requests of VM placement that would come from Nova Scheduler.

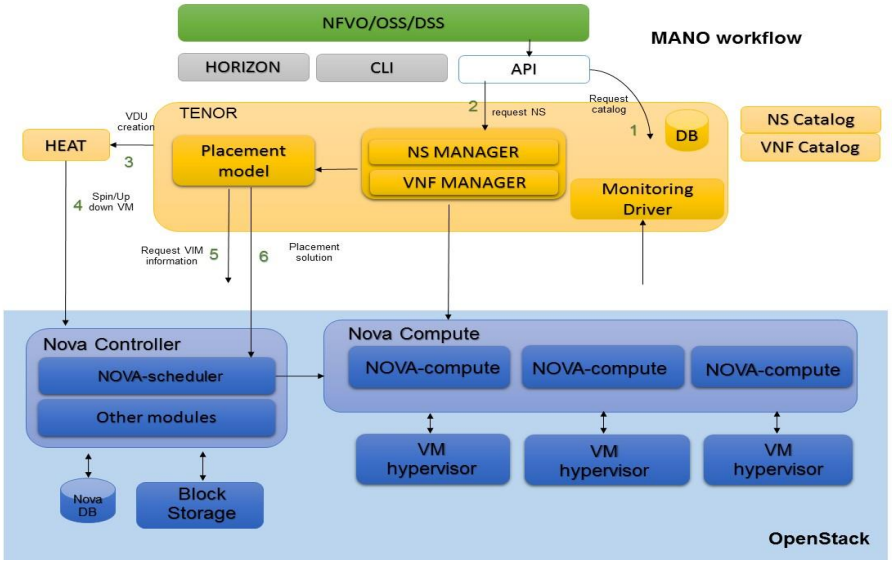


Figure 22: Placement model architecture

[22] TeNOR is the NFV Orchestrator from the FP7-TNOVA project, enabling Network Functions Virtualization as a Service over Virtualised Infrastructures.
For more details see: *https://github.com/T-NOVA/TeNOR/wiki*

5. SESAME Security Considerations

5.1. Introduction

The SESAME security analysis has been carried out at two different levels, that is: the requirements level and threat level. For the former, the Secure Tropos methodology and its automated tool (SecTro[23]) have been used to analyse security requirements related to SESAME. For the latter, the SESAME potential security threats were identified by using the STRIDE threat modelling[24]. The identified threats were then analysed using the Secure Tropos methodology and were documented also through the creation of a dossier[25] that includes all the relevant information.

5.2. Security Requirements Analysis

As mentioned above, for the analysis of the high-level security requirements of SESAME, we used the Secure Tropos security requirements engineering methodology and its SecTro tool. Secure Tropos is based on the Tropos methodology[26], which uses the concepts of: *actor* (entity that has strategic goals and intentionality); *goal* (an actor's strategic interest); *soft-goal* (goal without clear criteria whether it is satisfied or not); *task* (it represents the way of doing something); *resource* (it represents a physical or informational entity, without intentionality) and; *social dependencies* (indicate that one actor depends on another in order to attain some goals, execute some tasks, or deliver a resource).

Secure Tropos supports the modelling and analysis of security requirements by using a number of relevant concepts. In particular, a *security constraint* is defined as a restriction related to security issues, such as privacy, integrity and availability, which can influence the analysis and design of the information system under development by restricting some alternative design solutions, by conflicting with some of the requirements of the system or by refining some of the system's objectives. Secure Tropos uses the term *secure entity* to describe any goal and plan related to the security of the system. A *secure goal* represents the strategic interests of an actor with respect to security. Secure goals are mainly introduced in order to achieve security constraints that are imposed on an actor or exist in the system. However, a secure goal does not particularly define how the security constraints can be achieved, since alternatives can be considered. The precise definition of how the secure goal can be achieved is given by a secure plan. A *secure plan* is defined as a plan that represents a particular way for satisfying a secure goal. A *secure dependency* introduces security constraint(s) that must be fulfilled for the dependency to be satisfied. Both the depender and the dependee must agree for the fulfilment of the security constraint in order for the secure dependency to be valid. That means the depender expects from the dependee to satisfy the security constraint(s) and also that the dependee will make an effort to deliver the dependum by satisfying the security constraint(s).

The process in Secure Tropos is one of analysing the security needs of the stakeholders and the system in terms of security constraints imposed on the stakeholders and the system, identifying secure entities that guarantee the satisfaction of the security constraints, and assigning capabilities to the system to help towards the satisfaction of the secure entities.

---

[23] See: http://www.sense-brighton.eu/research/sectro-tool/
[24] For further details see: https://msdn.microsoft.com/en-us/library/ee823878(v=cs.20).aspx
[25] See: http://www.sense-brighton.eu/research/dossier-attacks/
[26] For more related information see: http://www.troposproject.eu/node/93

5.3. Security Requirements identified using Secure Tropos

In order to investigate the security challenges of SESAME -and 5G networks in general-, one must look into the security issues of its elements and their interaction. It is also equally important to consider the SESAME architecture "as a whole" and not its individual components, as looking only at individual component. Figure 23 illustrates a model that provides a partial view of the SESAME high-level security analysis, which focuses on three critical resources (i.e.: VNF Manager, NFV Orchestrator and Radio Access Manager).

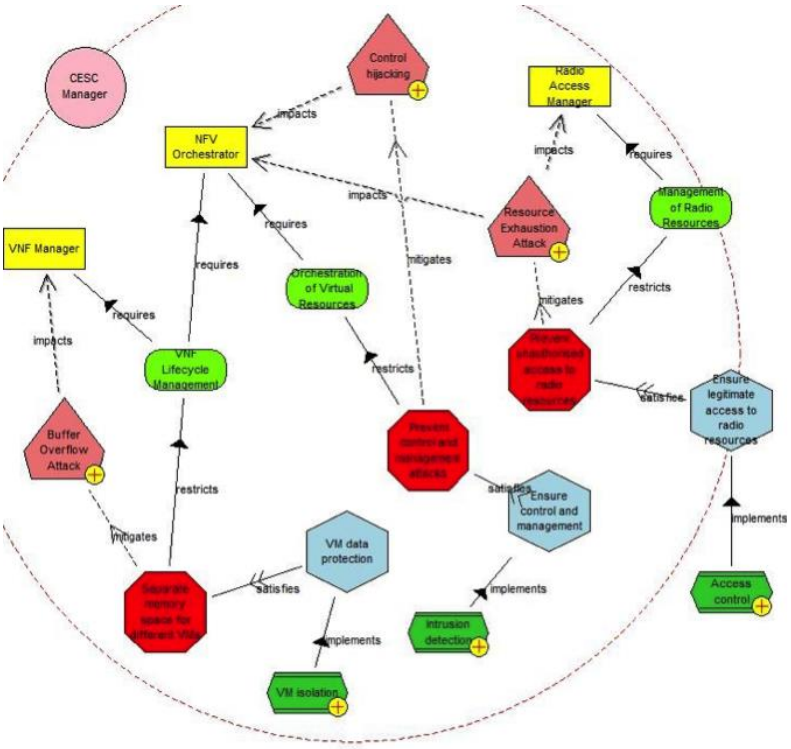


Figure 23: Partial Requirements Model

According to the Secure Tropos methodology, an actor's goal may be restricted by one -or more- security constraints. In this example, the security requirements (constraints) for the aforementioned goals are: *Separate memory space for different VMs; Prevent control and management attacks*, and; *Prevent unauthorized access to radio resources*. The security constraints have to be satisfied by the following three security objectives: *VM data protection*, *Ensure control and management protection*, and; *Ensure legitimate access to radio resources*. The corresponding security mechanisms to implement these objectives are *VM isolation; Intrusion detection*, and; *Access control*. We also consider the following security threats on the critical resources: *Buffer overflow attack; Resource exhaustion attack*, and; *Control hijacking*. The *Buffer overflow attack* threat can be mitigated by the *Separate memory space for different VMs* security constraint. The *Resource exhaustion attack* can be mitigated by the *Prevent unauthorized access to radio resources* security constraint. Finally, the *Control hijacking* can be mitigated by the *Prevent control and management attacks* security constraint.

Particular attention must be paid on protecting user's data and confidentiality. Some data or code, including various configuration settings and security policies, can be altered. This is a particularly important issue in a virtualised multitenant environment. It must be taken into account that some tenants could be malicious. Hence, adequate data and VM isolation for different tenants must be ensured. This could be done, for example at the database level or at the hardware level. Also, in some cases, sensitive information of a tenant may be leaked and made available to the adversary or to a malicious tenant.

Another important security consideration from our analysis is the unauthorized access to SC radio resources (physical or virtual) and MEC resources. Given the increasing trend of outsourcing data and applications, an adequate security solution must ensure that a security requirement to only allow authorized entities to gain access in the system is crucial. Also, the insider dimension should be considered and appropriate mechanisms must be put in place for preventing service providers from misusing tenants' data.

Particular attention must also be paid on appropriate encryption methods. Weaknesses or improper use of cryptographic mechanisms may lead to security breaches in authentication processes and data confidentiality. Also, the generation of cryptographic keys should not rely on "weak" random number generators. Other security problems may arise due to communication protocols that use weak cryptographic primitives. Hence, it should be ensured that the cryptographic security controls are in place. To ensure appropriate levels of protection, new multi-domain and multi-service trust models need to be considered.

Control hijacking attacks are expected to be a serious threat in SDN-*based* 5G networks. By exploiting the SDN controller implementation weaknesses, the adversary may try to divert the control flows to a controlled device. Then, the captured messages can be discarded preventing the data plane entities from proper operation. In a more advanced case, the captured messages may be manipulated with a special purpose code and sent into the network.

5.4. Threat Analysis

The previous requirements identification has provided us with a first high-level understanding of the various considerations that must be provided with respect to Security in SESAME. As a next step, we have made use of the STRIDE Threat Model to gain a better understanding of lower level threats. In performing threat modelling, our objective is to identify potential vulnerabilities related to SESAME and then identify countermeasures to prevent -or mitigate- the effects of, threats to the system. In the context of this work, we define "threat" as a potential -or actual- undesirable event that may be malicious (such as DoS attack) or incidental (failure of a Storage Device).

STRIDE is a classification scheme for characterizing known threats according to the kinds of exploit that are used (or motivation of the attacker). The STRIDE acronym is formed from the first letter of each of the following categories and a model developed by Microsoft. It provides a mnemonic for security threats in six categories.

The threat categories are:

- **Spoofing of user identity**. An example of identity spoofing is illegally accessing and then using another user's authentication information, such as username and password.
- **Data Tampering**. Data tampering involves the malicious modification of data. Examples include unauthorized changes made to persistent data, such as that held in a database, and the alteration of data as it flows between two computers over an open network, such as the Internet.
- **Repudiation**. Repudiation threats are associated with users who deny performing an action without other parties having any way to prove otherwise - for example, a user performs an illegal operation in a system that lacks the ability to trace the prohibited operations. Nonrepudiation refers to the ability of a system to counter repudiation threats. For example, a user who purchases an item might have to sign for the item upon receipt. The vendor can then use the signed receipt as evidence that the user did receive the package.
- **Information disclosure** (privacy breach or data leak). Information disclosure threats involve the exposure of information to individuals who are not supposed to have access

to it—for example, the ability of users to read a file that they were not granted access to, or the ability of an intruder to read data in transit between two computers.

- **Denial of service** (DoS). Denials of service (DoS) attacks deny service to valid users - for example, by making a Web server temporarily unavailable or unusable. You must protect against certain types of DoS threats simply to improve system availability and reliability.
- **Elevation of privilege**. In this type of threat, an unprivileged user gains privileged access and thereby has sufficient access to compromise or destroy the entire system. Elevation of privilege threats include those situations in which an attacker has effectively penetrated all system defences and become part of the trusted system itself, a dangerous situation indeed.

5.5. SESAME Potential Targets

As part of the information-gathering portion of the threat analysis, we identified SESAME's components that we need to explore for high-level threats and vulnerabilities. These are:

- **Virtualized Network Functions.** VNFs could be the source of an attack being a malware developed and programmed to perform attacks or it can be the target of an attack as a software component and part of an infrastructure.
- **Virtualization layer:** Adversaries could exploit vulnerabilities existing in hypervisors. For instance, to escape from the virtual computing, network or storage to the host's physical compute, network or storage resources. This vulnerability gives the necessary time to an adversary to compromise the confidentiality, integrity and/or availability of VNFs resources.
- **Communications with and within NFV MANO:** Adversaries may try to intercept or modify the traffic that transmits between the NFVI and the NFV MANO, as well as traffic within the NFV MANO.
- **Orchestrator and/or VNF manager:** Adversaries may try to take advantage of vulnerabilities in order to exploit these two components to interrupt the lifecycle management of the Network Services or of individual VNFs.
- **Virtualized Infrastructure Manager (VIM):** The VIM is responsible for the management of the NFVI compute, network and storage resources used by the VNFs. So VIM can be compromised causing Denial of Service (DoS), tampering, spoofing, privilege escalation or bypassing hypervisor isolation.

Some research has been suggested to improve the security of NFV conditions and status. For example, CloudBand [26] and integration of policy enforcement [27] are related actions. Most of the efforts to develop NFV focused on management and orchestration such as OpenMANO [25] and T-NOVA [38]. The ETSI establishes in [29] the threat landscape of NFV as being the centre of the threats to generic virtualization and generic networking. NFV is an implementation of Cloud computing for networking, therefore there are several incidents that have been performed against NFV systems and hypervisors. Potential areas of concerns for NFV are also established in [29]. These attacks can be categorized depending on the components previously listed:

Virtualized Network Functions: Denial of Service (DoS) attacks and incidents are a significant and important threat to NFV environments. There are several DoS incidents on providers and their services hosted in the Cloud, like Bitbucket[27], a web-*based* hosting service company hosted

[27] See: *https://bitbucket.org/*

by Amazon that was victim of massive DDoS (Distributed DoS) attacks [30]. The danger of DDoS is even more considerable given the background of NFV, because it could also affect unfocused and non-targeted services that are hosted on the same physical host. VNFs are software components providing network functions, so they are likely to be vulnerable to software programming bugs: it could be possible to bypass firewall restrictions or to take advantage of a buffer overflow to execute arbitrary code. CVE-2012-2663[28] for iptables and CVE-2006-5276[29] for Snort[30] are examples of such vulnerabilities and they give a first perception of threats that target typical NFs that are deployed in NFV infrastructures.

Virtualization layer. A respective amount of attacks can be accomplished on the virtualization layer:

- *Code execution on the physical host*. Wojtczuk [31] presents several attacks against common hypervisors QEMU[31]-KVM[32], Virtualbox[33], Xen[34]) that allow code execution on the host from a compromised or malicious Virtual Machine (VM). The first one lets the adversary to acquire code execution privileges in Xen's para virtualization domain by making Xen run a VM with a filesystem polluted in such a way that it can cause CVE-2007-5497[35]. The second one allows code execution on the host by using a use-after-free vulnerability[36] in QEMU-KVM, triggered by requesting a PCI unplug action on the virtual RTC (Real Time Clock) that was not hot plugged-in. Finally, the third attack uses a buffer overflow vulnerability that has to do with the emulation of the e1000 router to gain execution of arbitrary code.
- *Return-oriented-programming-based attacks*. Riddle and Chang [32] introduce an attack on the Xen hypervisor[37] that with the usage of return-oriented-programming[38] permits and lets the adversary to escalate their VMs to a privileged state. In the context of NFV, these attacks could be used to read -or modify- the memory of, take control of, or deny resources to VNFs co-resident with a malicious VNF meaning, something that disrupts the action of several Network Services, or even deploy more malicious VNFs.
- *Resource monopolization*. Riddle and Chang [32] present two attacks to hook resources from other VMs: Monopolization of CPU: It is possible either to use up to 98% of the physical host's CPU, if the VMs are running over a Xen hypervisor, having as a result to oppose and forbid the CPU to other VMs, or to decide whether 2 VMs can be the starting point of another attack, by taking advantage of Xen's credit scheduler.
- *I/O performance-based attacks*. If the adversary has knowledge of the scheduling characteristics of the hypervisor, attackers can utilize that knowledge in order to overload I/O resources, having as a result in slowing down co-resident VNFs.
- *Data theft*. Riddle and Chang [32] explain that if the target VM is co-resident with the attackers' malicious VM and is infected with malware, then the attacker can use

[28] For further information also see: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-2663
[29] For further information also see: http://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2006-5276
[30] For more information about Snort also consider: https://www.snort.org/
[31] Also see: https://www.qemu.org/
[32] For more details about QEMU-KVM also see, *inter-alia*: https://wiki.qemu.org/Features/KVM
[33] Also see: https://www.virtualbox.org/wiki/VirtualBox
[34] For more details also see, *inter-alia*: https://en.wikipedia.org/wiki/Xen
[35] For further information also see: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-5497
[36] See: http://www.webopedia.com/TERM/U/use-after-free.html
[37] Further information can be found at: https://www.xenproject.org/developers/teams/hypervisor.html
[38] See: https://en.wikipedia.org/wiki/Return-oriented_programming

memory bus or cache contention to stealthily steal data (e.g. cryptographic keys), from the target VM.

- *VM monitoring evasion*. VM rollback attack is introduced by Riddle and Chang [32]: if the hypervisor is already compromised, then the adversary may execute a VM from an older snapshot without the VM administrator having knowledge of it, giving them the opportunity to bypass security systems. For example, in a brute force password attack, when the VM brings up a security alert, the compromised hypervisor returns to the previous snapshot, and the adversaries can process their attacks. A technique presented as the hypervisor introspection is introduced by Wang et *al.* [33]. This attack passively monitors VMs, the hypervisor needs to suspend the VM to get a consistent view of the hardware state. By determining the frequency at which the hypervisor pauses the VM for inspection, adversaries can perform operations between the monitoring checks. This allows them, for example, to stealthily exfiltration data where in NFV is network traffic or to maintain and hide the existence of a back-door shell inside a VM.

Orchestrator and/or VNF manager: Using ephemeral storage to steal data (CVE-2013-7130[39]): The create_images_and_backing method in libvirt driver in OpenStack Compute (Nova), when using KVM live block migration, does not properly create all expected files, which allows attackers to obtain snapshot root disk contents of other users via ephemeral storage. In an NFV over OpenStack environment this could be used to steal cryptographic keys from other VNFs thus enabling, for instance, data modification, impersonation or eavesdropping.

Virtualized Infrastructure Manager: Privilege escalation (CVE-2014-3790[40]): Ruby vSphere Console (RVC) in VMware vCenter Server appliance (centralized management and operation, resource provisioning and performance evaluation of VMs in a distributed virtual data center) allows remote authenticated users to execute arbitrary commands as root by escaping from a chroot jail[41], meaning that they can gain control over the infrastructure domain managed by the VIM. Concerning NFV identified in [29], among the incidents introduced, DDoS attacks can be related to secure crash, performance isolation and availability of management support infrastructure. Vulnerability CVE-2014-3790 and resource monopolization incidents can be related to performance isolation, while data interception and stealing can be related to private keys within cloned images.

NFV is an implementation of Cloud Computing for networking, so NFV has same or similar threats:

Distributed Denial of Service: Not having unlimited resources enables DoS incidents, and the fact that it is difficult to tell the difference between normal traffic and malicious traffic. Solutions have been proposed to take countermeasures against DDoS. A suggestion came from Joshi et al. [34], where he proposed Cloud Trace ack[42]. His solution is a back-propagation neural network trained to detect malicious traffic. It becomes possible to take countermeasures against DoS incidents, by using techniques like selective blackholing [35], when malicious traffic can be classified and separated from normal traffic. There are many vulnerabilities in hypervisors whose exploitation allows these. Some of them are code execution on the host, privilege escalation, isolation breaking. Previous years, more than 50 CVEs concerning VMWare's VSphere, QEMU, KVM, Xen, Hyper-V[43], LXC[44] and Docker[45] using a Common Vulnerability

[39] For further information also see: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-7130
[40] For further information also see: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-3790
[41] See: https://unix.stackexchange.com/questions/105/chroot-jail-what-is-it-and-how-do-i-use-it
[42] Also see the context in: A. Sharma, and S. Singh (2015): DDOS Attack Detection and Prevention with Cloud Trace Back. *International Journal of Innovative Computer Science & Engineering, 2*(3), pp.30-35.
[43] For further details also see: https://en.wikipedia.org/wiki/Hyper-V
[44] For further details also see: https://en.wikipedia.org/wiki/LXC
[45] For further details also see: https://www.docker.com/

Scoring System[46] scored greater than 7 (out of 10). Some of these security incidents exploit buffer overflows, which can be counter measured with techniques such as ASLR[47] or canaries[48]..

Side-channel attacks: In side-channel attacks, adversaries deduce information in an accidental manner, for example by estimating the frequency at which a VM is paused. Some countermeasures against this type of attack, is to completely remove or to minimize the information released by the side channel (TEMPEST [36]), or to insert some kind of noise to the channel.

5.6. SESAME Security Dossier

Based on the above security requirements and attack analysis, we have identified a number of threats that are relevant to the SESAME architecture. We have documented those on the SESAME Security Dossier, which can be found in *http://www.sense-brighton.eu/research/dossier-attacks/.* The dossier includes a total of 44 threats. To support better understanding of those threats, we have enhanced the dossier with a clear alignment between the STRIDE threat categories, potential SESAME targets/assets, and a proposal for countermeasures. Moreover, we have provided information regarding the classification and enumeration of an attack, using the CAPEC (Common Attack Pattern Enumeration and Classification) database and the corresponding vulnerability information using the Common Vulnerabilities and Exposures (CVE) database. Such information provides a complete analysis of the relevant threats. Below we provide an example from each of the main STRIDE categories.

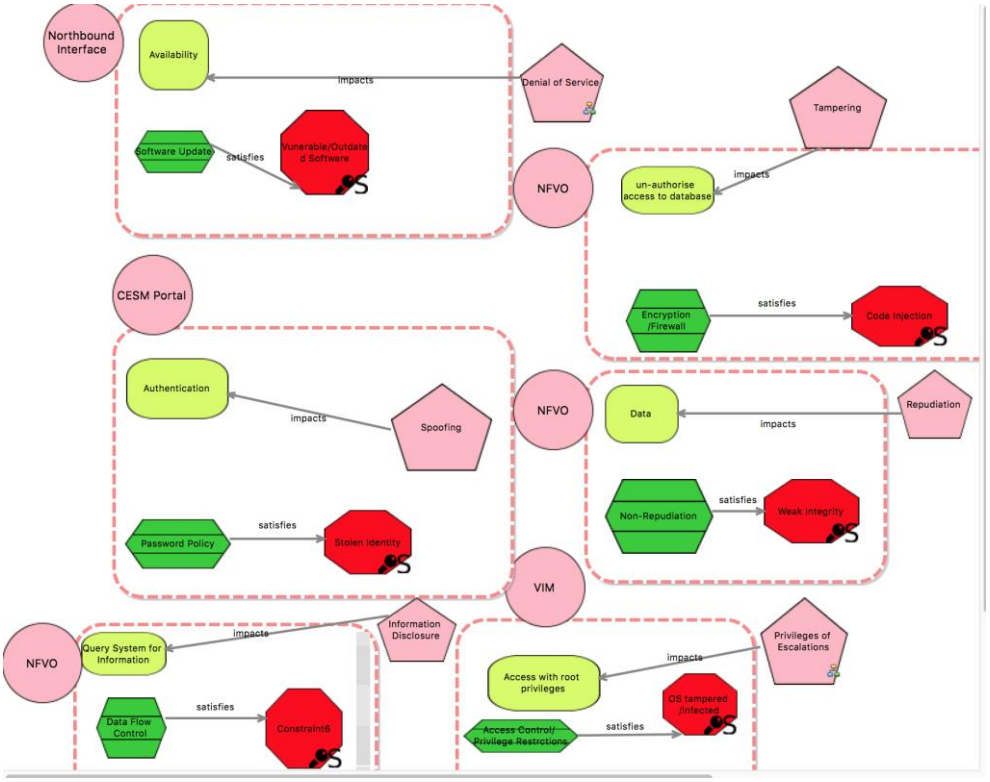


Figure 24: Abstract Model of STRIDE Cases proposed

[46] Also see: *https://en.wikipedia.org/wiki/Common_Vulnerability_Scoring_System*
[47] For further information see: *https://en.wikipedia.org/wiki/Address_space_layout_randomization*
[48] For further information see: *https://canary.is/*

Moreover, for each of threats we have used the SecTro tool to create a model of the threat. This is important as it has enabled us to directly link the various threats with the security requirements we have identified from our SecTro analysis and better understand the impact of those threats to the security requirements.

The models explained are made with Secure Tropos tool. Firstly, there is an abstract presentation of the threat model proposed and then a more specific approach of the analysis for each STRIDE category. In the abstract model based on the cases proposed below, there is an actor-component, a threat, a goal, a constraint and a secure mechanism. Threat has a goal for the actor and actor has a constraint that demands a secure mechanism.

To support understanding of the above figure, the SecTro notation is provided below[49]:

Actor. "Actors" are active entities that carry out actions to achieve goals by exercising its know-how. The term actor refers generically to any unit to which intentional dependencies can be ascribed. An actor interacts with other actors not only through actions or information flows, but also relate to each other at an intentional level. Actors depend on each other to achieve goals, perform tasks, and furnish resources. While each actor has strategic goals to pursue, they are achieved through a network of intentional dependencies.

Goal. A "Goal" is a condition or state of affairs to be achieved. An actor can choose freely among different ways to achieve a goal. Represents and intentional desire of an actor, the specifics of HOW the goal is to be satisfied is not described by the goal.

Constraint. "Constraint" is a restriction on an actor's function. There are two types of Constraints, namely Security and Privacy. Additionally, a Constraint is related to an Objective e.g. Confidentiality, Integrity, Authentication.

Mechanism. A "Mechanism" represents a system mechanism that supports the satisfaction of a security objective. It can be any of two types, Security or Privacy.

Threat. "Threat" represents a circumstance that has the potential to cause damage to the system.

[49] This is a partial illustration of the SecTro notation, for a complete one please go to *http://www.sense-brighton.eu/research/sectro-tool/*

Attacker. An "Attacker" is a malicious actor who tries to endanger the security of the system through attacking its resources, goals and plans.

Vulnerability. "Vulnerability" is a weakness of the system or the organisation.

Attack Method. An "Attack Method" is a method by which a Threat is realised.

5.6.1. STRIDE Threat category: Spoofing

STRIDE CATEGORY: Spoofing
SESAME Target: CESCM Portal
Threat: Stolen tenant identity
Potential Countermeasure Category: Authentication
Example of Countermeasure: Multi-factor Authentication
Domain of Attack: Supply Chain
Severity: Medium
CAPEC ID: 151: Identity Spoofing
CVE Vulnerability Exploited: "1999-0667: The ARP protocol allows any host to spoof ARP replies and poison the ARP cache to conduct IP address spoofing or a denial of service. OAuth makes no attempt to verify the authenticity of the authorization server. A hostile party could take advantage of this by intercepting the client's requests and returning misleading or otherwise incorrect responses. This could be achieved using DNS or Address Resolution Protocol (ARP) spoofing.

Figure 25 presents an example model for spoofing of the CESCM Portal. The adversary uses a method to attack the component abusing the vulnerability, depicted as red circle, of this system that affects the Authentication goal. A relevant security mechanisms is presented («Password Policy»).

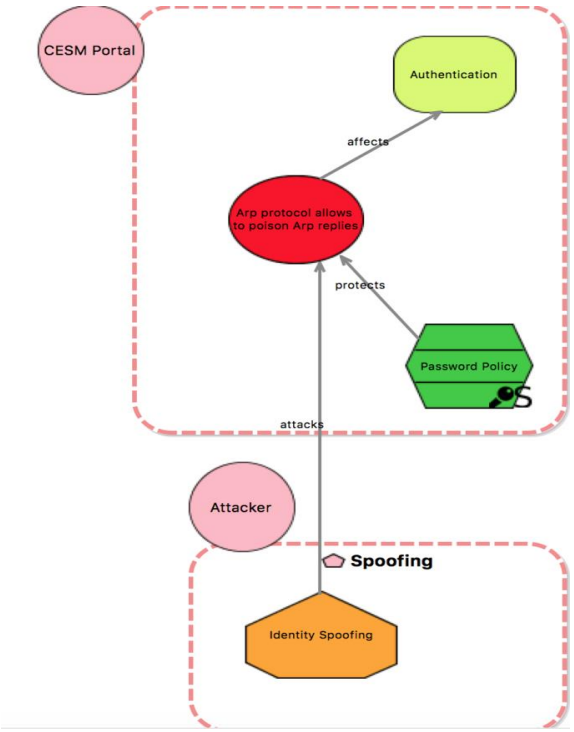


Figure 25: Spoofing

5.6.2. STRIDE Threat category: Tampering

STRIDE CATEGORY: Tampering
SESAME Target: NFVO Coordinator
Threat: un-authorised change on database
Potential Countermeasure Category: Data Confidentiality
Example of Countermeasure: Encryption / Firewall
Domain of Attack: Supply Chain
Severity: High
CAPEC ID: 242: Code Injection
CVE Vulnerability Exploited: 2013-1892: Remote Code Injection Vulnerability

Figure 26 illustrates a partial model for Tampering of the NFVO. The adversary uses a method to attack the component abusing the «Remote Code Injection» vulnerability, which has an impact on the Data confidentiality goal. Encryption/firewalls are proposed as potential security mechanisms.

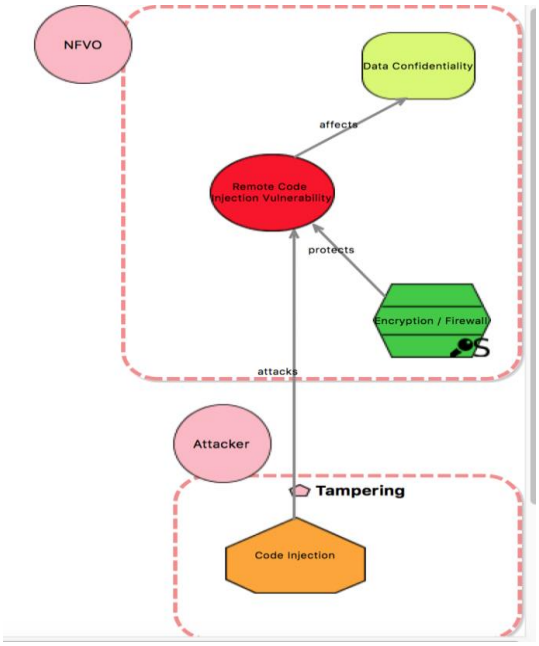


Figure 26: Tampering Case

5.6.3. STRIDE Threat Category: Repudiation

STRIDE CATEGORY: Repudiation
SESAME Target: NFVO Coordinator
Threat: insufficient auditing / weak protection for audit data
Potential Countermeasure Category: Non-Repudiation
Example of Countermeasure: Integrity monitoring for configuration files
Domain of Attack: Communications / Software
Severity: Very High
CAPEC ID: 75: Manipulating Writeable Configuration Files
CVE Vulnerability Exploited: 1999-0019: Delete or create a file via rpc.statd, due to invalid information.

Figure 27 illustrates the partial SecTro model for the Repudiation of NFVO Coordinator. The adversary uses an attack method to manipulate the configurable files of the system, which can exploit a specific vulnerability. A potential security mechanism to mitigate this is provided through the integrity monitoring security mechanism.

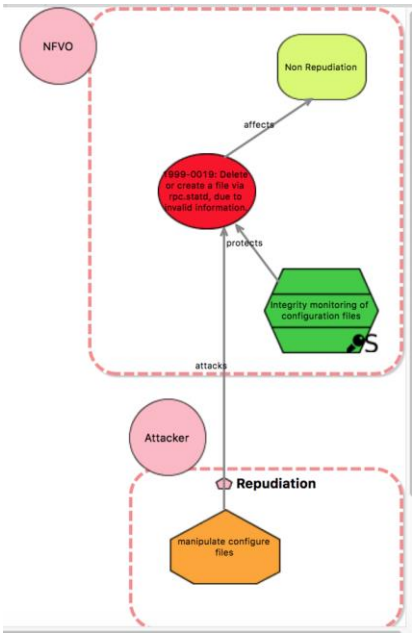


Figure 27: Repudiation of the NFVO Coordinator

5.6.4. STRIDE Threat Category: Information Disclosure

STRIDE CATEGORY: Information Disclosure
SESAME Target: NFVO Coordinator
Threat: Query System for Information
Potential Countermeasure Category: Data Confidentiality
Example of Countermeasure: Data Flow Control
Domain of Attack: Communications
Severity: Low
CAPEC ID: 191: Read Sensitive Strings within an Executable
CVE Vulnerability Exploited: 1999-0154: IIS 2.0 and 3.0 allows remote attackers to read the source code for ASP pages by appending a. (dot) to the end of the URL. 1999-0129: Sendmail allows local users to write to a file and gain group permissions via a. forward or: include: file.

Figure 28 provides an example illustration of the Information disclosure of NFVO Coordinator. The adversary uses a method to attack a potential vulnerability of the system. Data Flow control is provided as an example security mechanism.

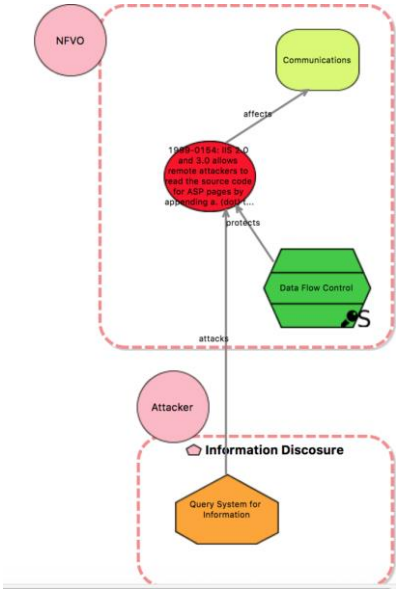


Figure 28: Information Disclosure

5.6.5. STRIDE Threat Category: Denial of Service

STRIDE CATEGORY: Denial of Service
SESAME Target: NFVO Coordinator
Threat: out-dated version of software used
Potential Countermeasure Category: Availability
Example of Countermeasure: Software Update
Domain of Attack: Software
Severity: High
CAPEC ID: 343: Denial of Service
CVE Vulnerability Exploited: 2017-1000357: OpenDaylight, Denial of Service attack when the switch rejects to receive packets from the controller

Figure 29 provides an example illustration of the Denial of Service of Northbound Interface. The adversary uses as Amplification to attack an out-of-date software vulnerability. Software update is provided as an example of a security mechanism.

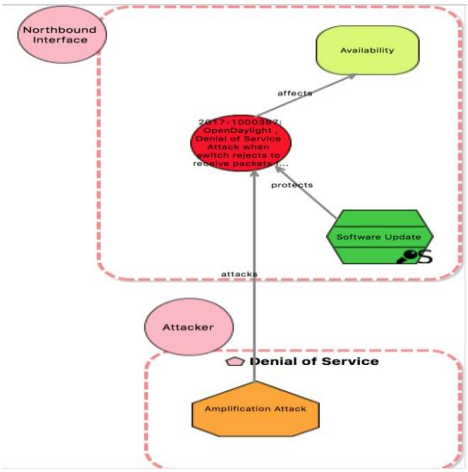


Figure 29: Denial of Service

5.6.6. STRIDE Threat Category: Elevation of Privilege

STRIDE CATEGORY: Elevation of Privilege
SESAME Target: NFVO Coordinator
Threat: Access to the NFVO with root privileges
Potential Countermeasure Category: Access Control
Example of Countermeasure: Privilege Restrictions
Domain of Attack: Software
Severity: High
CAPEC ID: 233: Privilege Escalation
CVE Vulnerability Exploited: 2011-4127: privilege escalation from QEMU / KVM guests

Figure 30 provides an example illustration of the Elevation of privileges of VIM. The adversary uses a Horizontal Privilege Escalation Attack to explore elevation of privilege vulnerability. Access Control and/or Privilege restrictions are examples of generic potential mechanisms to mitigate the attack.

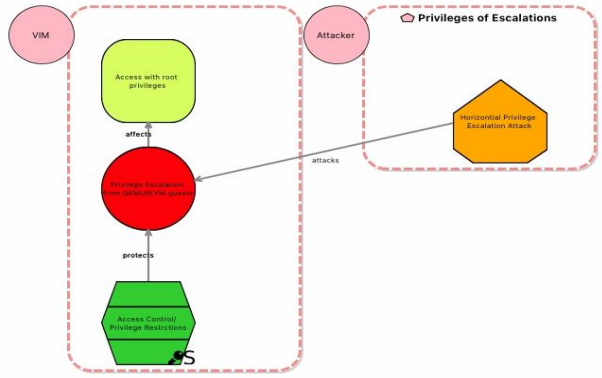


Figure 30: Elevation of Privilege

5.7. Conclusion

This section provided information related to our security analysis of the SESAME security and in particular the Network Functions Virtualisation. The literature does not provide any evidence of attacks that specifically target an NFV infrastructure, so there was a search for CVE Identifiers and attacks related to Cloud computing and hypervisors and an overall security analysis of the SESAME components. The outputs of the work have been recorded in the form of a dossier.

6. Conclusions

This deliverable compiles the activities related to VNF placement performed in Task 5.3 ("*VNF Placement Algorithms and Development*") in the context of the SESAME WP5. These activities include both the analysis and design of an intelligent VNF placement mechanism and also the study of vulnerability issues and security challenges.

Section 2 focuses on the VNF placement problem description and modelling. There, we analyse the challenges of service mapping on the virtualised execution infrastructure and the associated VIM, by covering aspects such as virtual resource provisioning and efficient local and cross-CESC site placement algorithms to "meet" agreements of service. We propose a constraint-based expert system to support the decision-making process of VNF chain placement in distributed edge cloud networks. The placement algorithm applies robust optimization techniques to the minimization of the global power consumption of the system, subject to the bitrate and latency constraints of the network services. The power consumption of the components of the system depends on the usage percentage, which is a consequence of the aggregated user traffic demand. The model assumes that the relationships between the user traffic demand, the power consumption and the use of the resources may be non-linear. Besides, the placement algorithm considers the 5G communication particularities of both control and data plane and allows multi-tenancy. Finally, the algorithm also includes VNF scaling features and the use of hardware acceleration.

In Section 3, we evaluate the performance of the placement algorithm. The results show that the placement algorithm succeeds in allocating the VNF that compose the NSs of different tenants in the available resources, thus meeting the latency constraints and minimizing the energy consumption of the whole system. The presented evaluation examples demonstrate the features of the constraint-based expert system and illustrate the impact of the selected service flavour and robust protection on the global power consumption and, *therefore*, the exploitation cost of the 5G distributed edge cloud.

Implementation issues are addressed in Section 4. This section proposes the enhancement of OpenStack's Nova scheduler algorithm with the logic needed for the SESAME scenario.

Section 5 presents a security analysis of the SESAME, across two levels: Security Requirements and Security Threats. The outputs of that analysis have been documented in the form of a security dossier.

Finally, Section 6 briefly gathers the final conclusions of the deliverable.

This document concludes the tasks of WP5 as an input for WP7, responsible for the integration of all the task outputs.

7. References

[1] I. Chih-Lin, C. Rowell, S. Han, Z. Xu, G. Li and Z. Pan (2014): Toward green and soft: a 5G perspective. *IEEE Communications Magazine*, *52*(2), pp.66-73.

[2] J. Gil Herrera, J.F. Botero (2016): Resource Allocation in NFV: A Comprehensive Survey, *IEEE Transactions on Network and Service Management 13*, pp.518–532.

[3] B. Addis, D. Belabed, M. Bouet and S. Secci (2015): Virtual network functions placement and routing optimization. In Proceedings of the *IEEE 4th International Conference on Cloud Networking (CloudNet*), pp.171-177. Niagara Falls, Ontario, Canada, October 05-07, 2015.

[4] H. Moens and F.D. Turck (2014): VNF-P: A Model for Efficient Placement of Virtualized Network Functions. In Proceedings of the *IEEE 10th International Conference on Network and Service Management (CNSM) and Workshop*, pp.418-423. Rio de Janeiro, Brasil, November 17-21, 2014.

[5] A. Gupta, M Farhan Habib, P. Chowdhury, M. Tornatore, and B. Mukherjee (2015): On service chaining using Virtual Network Functions in Network-enabled Cloud systems. In Proceedings of the *IEEE 2015 International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pp.1-3. Kolkata, India, December 15-18, 2015.

[6] M. Bouet, J. Leguay, T. Combe, and V. Conan (2014): Cost-based placement of vDPI functions in NFV infrastructures. *International Journal of Network Management*, *25*(6), pp.490-506.

[7] M.C. Luizelli, L.R. Bays, L.S. Buriol, M.P. Barcellos, L.P. Gaspary (2015): Piecing together the NFV provisioning puzzle: Efficient placement and chaining of virtual network functions. In Proceedings of the *IFIP/IEEE 2015 International Symposium on Integrated Network Management (IM)*, pp.98-106. Ottawa, Ontario, Canada, May 11-15, 2015.

[8] S. Kim, Y. Han, and S. Park (2016): An Energy-Aware Service Function Chaining and Reconfigura- tion Algorithm in NFV. In Proceedings of IEEE International Workshops on Foundations and Applications of Self* Systems, pp.54-59. Augsburg, Germany, September 12-16, 2016.

[9] V. Eramo, A. Tosti, and E. Miucci (2016): Server resource dimensioning and routing of service function chain in NFV network architectures. *Journal of Electrical and Computer Engineering 2016,* Article ID 7139852.
Available at: http://dx.doi.org/10.1155/2016/7139852.

[10] K. Hida, S.-I. Kuribayashi (2016): Virtual Routing Function Allocation Method for Minimizing Total Network Power Consumption. *World Academy of Science, Engineering and Technology, International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering 10*(8), pp.1018-1023.

[11] C. Pham, N.H. Tran, S. Ren, W. Saad, and C.S. Hong (2017): Traffic-aware and Energy-efficient vNF Placement for Service Chaining: Joint Sampling and Matching Approach. *IEEE Transactions on Services Computing 99*, p.1.

[12] A. Marotta, and A. Kassler (2016): A Power Efficient and Robust Virtual Network Functions Placement Problem. In *Proceedings of the 2016 28th International Teletraffic Congress (ITC 28),* vol.1, pp. 330-338. Würzburg, Germany, September 12-16, 2016.

[13] C. Pham, H.D. Tran, S.-Il. Moon, K. Thar, C.S. Hong (2015): A general and practical consolidation framework in CloudNFV. In Proceedings of the 2015 International Conference on Information Networking (ICOIN), pp.295-300. Cambodia, January 12-14, 2015.

[14] V. Eramo, M. Ammar, and F.G. Lavacca (2017): Migration Energy Aware Reconfigurations of Virtual Network Function Instances in NFV Architectures. *IEEE Access, 5*, pp.4927-4938.

[15] N. El Khoury, S. Ayoubi, and C. Assi (2016): Energy-Aware Placement and Scheduling of Network Traffic Flows with Deadlines on Virtual Network Functions. In *Proceedings of the 2016 5th IEEE International Conference on Cloud Networking (Cloud- net)*, pp.89-94. Pisa, Italy, October 03-05, 2016.

[16] V. Eramo, E. Miucci, M. Ammar, and F.G. Lavacca (2017): An approach for service function chain routing and virtual function network instance migration in network function virtualization architectures. *IEEE/ACM Transactions on Networking 25*(4), pp.2008-2025.

[17] V. Jumba, S. Parsaeefard, M. Derakhshani, and T. Le-Ngoc (2015): Energy-Efficient Robust Resource Provisioning in Virtualized Wireless Networks. In Proceedings of the 2015 IEEE International Conference on Ubiquitous Wireless Broadband (ICUWB), pp.1-5. Montreal, Quebec, Canada, October 04-07, 2015.

[18] S. Coniglio, A. Koster, and M. Tieves (2016): Data Uncertainty in Virtual Network Embedding: Robust Optimization and Protection Levels. J*ournal of Network and Systems Management, 24*(3), pp.681–710.

[19] I. Takouna, K. Sachs, and C. Meinel (2014): Multiperiod robust optimization for proactive resource provisioning in virtualized data centers. *The Journal of Supercomputing, 70*(3) pp.1514-1536.

[20] G. Chochlidakis, and V. Friderikos (2015): Robust virtual network embedding for mobile networks. In *Proceedings of the 2015 IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC),* pp.1867-1871. Hong Kong, China, August 30 – September 02, 2015.

[21] S. Coniglio, A.M.C.A. Koster, and M. Tieves (2015): Virtual network embedding under uncertainty: exact and heuristic approaches. In *Proceedings of the 2015 11th International Conference on the Design of Reliable Communication Networks (DRCN)*, pp.1-8. Kansas City, MO, US, March 24-27, 2015.

[22] A. Ben-Tal, L. El Ghaoui, A. Nemirovski (2009): *Robust Optimization*. Princeton, New Jersey: Princeton University Press.

[23] D. Bertsimas and M. Sim (2004). The price of robustness*. Operations Research, 52*, pp.35-53.

[24] OpenStack Nova Scheduler. Available at: https://docs.openstack.org/nova/latest/user/filter-scheduler.html. Accessed on September 2017.

[25] D.R. Lopezj (2015): OpenMANO - the Dataplane-Ready Open Source NFV MANO Stack. In Proceedings of the IETF Meeting of March 2015. Available at: https://webcache.googleusercontent.com/search?q=cache:iCAW4RSC4-gJ:https://www.ietf.org/proceedings/92/slides/slides-92-nfvrg-7.pdf+&cd=1&hl=el&ct=clnk&gl=gr

[26] "Providing Security in NFV: Challenges and Opportunities – Strategic White Paper," http://www.tmcnet.com/redir/?u=1011422 , 2014.

[27] C. Basile, A. Lioy, C. Pitscheider, F. Valenza, and M. Vallini (2015): A novel approach for integrating security policy enforcement with dynamic network virtualization. In *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft 2015)*, London, UK, April 13-17, 2015.

[28] T-NOVA: Network Functions-as-a-Service (NFaaS) over Virtualized Infrastructures. Available at: http://www.t-nova.eu.

[29] ETSI (2014): ETSI-ISG-NFV: Network Functions Virtualisation (NFV); NFV Security; Problem Statement.http://www.etsi.org/deliver/etsi_gs/NFVSEC/001_099/001/01.01.01_60/gs_NFV-SEC001v010101p.pdf.

[30] T.-S. Chou (2013, June): "Security Threats on Cloud Computing Vulnerabilities. *International Journal of Computer Science and Information Technology*, *5*(3), pp.79-88.

[31] R. Wojtczuk (2014, June): Poacher turned gamekeeper: Lessons learned from eight years of breaking hypervisors. In *Black Hat USA 2014*, Mandalay Bay, La Vegas, Nevada, August 002-07, 2014.

[32] A.R. Riddle and S.M. Chung (2015): "A Survey on the Security of Hypervisors in Cloud Computing. In *Proceedings of the 2015 IEEE 35th International Conference on Distributed Computing Systems Workshops (ICDSSW)*, pp.100-104. Colombus, Ohio, US, June 29 - July, 02, 2015.

[33] G. Wang, Z.J. Estrada, C. Pham, Z. Kalbarczyk, and R.K. Iyer (2015): Hypervisor Introspection: A Technique for Evading Passive Virtual Machine Monitoring. In *Proceedings of the 9th USENIX Workshop on Offensive Technologies (WOOT'15)*, pp.1-8. Washington, DC, US, August 10-11, 2015.

[34] B. Joshi, A. Santhana Vijayan, and B. Kumar Joshi (2012): Securing Cloud Computing Environment Against DDoS Attacks. In *Proceedings of the 2012 International Conference on Computer Communication and Informatics (ICCCI-2012)*, pp.1-5. Coimbatore, India, January 10-12, 2012.

[35] T. Jayawardena, and L.E. Morales (2008, October): *Distributed denial-of-service attack mitigation by selective black-holing in ip networks*. http://www.google.sr/patents/US20060031575

[36] National Security Agency (NSA): *TEMPEST Certification Program*. https://www.nsa.gov/applications/ia/tempest.

[37] F. Reynaud, F.-X. Aguessy, O. Bettan, M. Bouet and V. Conan (2016): Attacks against Network Functions Virtualization and Software-Defined Networking: State-of-the-art. In *Proceedings of the 2016 IEEE NetSoft Conference and Workshops (NetSoft)*. Seoul, South Korea, June 06-10, 2016.