



Converged Heterogeneous Advanced 5G Cloud-RAN Architecture for Intelligent and Secure Media Access

Project no. 671704

Research and Innovation Action

Co-funded by the Horizon 2020 Framework Programme of the European Union



Call identifier: H2020-ICT-2014-1

Topic: ICT-14-2014 - Advanced 5G Network Infrastructure for the Future Internet

Start date of project: July 1st, 2015 (30 months duration)

Deliverable D3.1

V-Security Management Plane Design and Definition

Due date: 01/01/2016

Submission date: 31/01/2016

Deliverable leader: i2CAT

Editors: Amaia Legarrea and Shuaib Siddiqui (i2CAT)

Dissemination Level

-
- | | |
|-------------------------------------|---|
| <input checked="" type="checkbox"/> | PU: Public |
| <input type="checkbox"/> | PP: Restricted to other programme participants (including the Commission Services) |
| <input type="checkbox"/> | RE: Restricted to a group specified by the consortium (including the Commission Services) |
| <input type="checkbox"/> | CO: Confidential, only for members of the consortium (including the Commission Services) |
-

List of Contributors

Participant	Short Name	Contributor
Fundació i2CAT	I2CAT	Amaia Legarrea, Shuaib Siddiqui, Eduard Escalona
Demokritos NCSR	NCSR	Eleni Trouva
APFutura	APFUTURA	Oriol Riba
Innoroute	INNO	Andreas Foglar, Marian Ulbricht
JCP-Connect	JCP-C	Yaning Liu
University of Essex	UESSEX	Mike Parker
Intracom	ICOM	Spiros Spirou
Ethernity	ETH	Eugene Zetserov

Change history

Version	Date	Partners	Description/Comments
0.0	13/10/2015	i2CAT	ToC definition
0.1	01/12/2015	NCSR, JCP-C, Innoroute, UEssex, APFutura	First Round of Contributions
0.2	12/01/2016	i2CAT	Reviewed first draft and provided comments
0.3	19/01/2016	i2CAT	i2CAT internal version
0.4	21/01/2016	i2CAT	Ready for internal review version
0.5	26/01/2016	i2CAT	Revision according to internal review feedback
1.0	29/01/2016	i2CAT	Final version ready for submission

Table of Contents

- Executive Summary..... 6**
- 1. Introduction 7**
- 2. Management systems of virtualised networks SOTA Survey 9**
 - 2.1. Software Defined Network Controllers.....9*
 - 2.2. NaaS Platforms10*
 - 2.2.1. OpenNaaS10
 - 2.2.2. OpenStack Neutron.....11
 - 2.2.3. OpenDaylight Virtual Tenant Network (VTN)12
 - 2.2.4. FlowVisor.....12
 - 2.2.5. OpenVirtex13
 - 2.2.6. OpenContrail.....14
 - 2.3. Cloud Environments14*
 - 2.3.1. OpenStack14
 - 2.3.2. Eucalyptus15
 - 2.3.3. Cloudstack.....15
 - 2.3.4. VMware vCloud Suite.....16
 - 2.4. NFV Orchestrators16*
 - 2.4.1. Open source projects16
 - 2.4.2. Project Based Implementations.....19
 - 2.4.3. Market Ready Orchestrator Solutions22
 - 2.5. CHARISMA Management Approach26*
- 3. Key drivers for CHARISMA management system.....27**
 - 3.1. Open Access27*
 - 3.1.1. Open access overview.....27
 - 3.1.2. CHARISMA vision for open access29
 - 3.1.3. CHARISMA open access requirements.....29
 - 3.1.4. CHARISMA open access services.....30
 - 3.1.5. Unified Control & Management32
 - 3.2. Security33*

- 3.3. *Low Latency*35
 - 3.3.1. *Low latency in content caching solution*.....36
- 4. Control, Management and Orchestration Plane High-level Design37**
 - 4.1. *Overview of the CMO plane of CHARISMA*37
 - 4.2. *Building blocks and functional design of the CMO plane*39
 - 4.2.1. *Management and Orchestration*39
 - 4.2.2. *Virtualised Network Functions*.....42
 - 4.2.3. *Virtualized Infrastructure*.....43
 - 4.3. *Management interfaces*44
 - 4.4. *NFV and the Network*46
- 5. Conclusions49**
- References50**
- Acronyms.....52**

Figures Summary

Figure 2-1: OpenNaaS Architecture and NaaS Resource Abstraction	11
Figure 2-2: OpenStack Neutron Architecture [13]	12
Figure 2-3: FlowVisor Architecture [15]	13
Figure 2-4: OpenVirtex Architecture [16]	13
Figure 2-5: Tacker Architecture [23].....	17
Figure 2-6: OpenMano scope diagram [24].....	19
Figure 2-7: High-level view of T-NOVA Orchestrator [25]	19
Figure 2-8: Mobile Cloud Networking Vision [27]	21
Figure 2-9: MCN Architectural Entities Relationship [27]	21
Figure 2-10: Cloudify Manager overall Architecture [28].....	22
Figure 2-11: CYAN Orchestrator Architecture [29].....	23
Figure 2-12: CYAN Multi-Vendor/Multi-Domain Framework [29]	24
Figure 2-13: HP NFV Building Blocks [30]	25
Figure 2-14: Cisco NFV Management and Orchestration Architecture [31]	25
Figure 3-1: Network value chain operation models.	28
Figure 3-2 Bitstream services in Open Access Networks	29
Figure 3-3: Hierarchical routing network	33
Figure 3-4: CHARISMA Aggregation Levels Architecture.....	35
Figure 4-1: The CHARISMA model for the access network	37
Figure 4-2: The high-level CHARISMA control and management plane.....	38
Figure 4-3: OpenStack: Functions description mapping.....	46
Figure 4-4: CHARISMA Open Access secure management interfaces.....	47

Executive Summary

This deliverable D3.1 “V-security management plane design and definition” represents the first full technical report emerging from the CHARISMA 5G-PPP project. The deliverable sets up the initial discussion regarding the control, management and orchestration plane of CHARISMA, as driven by the architecture definition and use case definition work carried out in WP1.

The document provides an introduction to the control, management and orchestration (CMO) plane design and architecture in the virtualized networking paradigm before presenting the state-of-the-art of the control and management methodologies of the related technologies. The state-of-the-art covers the pertinent technologies including software-defined network (SDN) controllers, Network-as-a-Service (NaaS) platforms, cloud environments, and network functions virtualisation (NFV) orchestration. The state-of-the-art covers only the related solutions, implementations or projects that have received community or industry wide attention.

The document also elaborates on the key drivers for the CHARISMA project including open access, security, and low latency. CHARISMA has designed open access into its architecture, focusing on converged wireless/wireline networking, inter-operability, simplified operation, administration & management (OAM), and modularity to underpin multi-vendor support as the main requirements. These requirements facilitate the provision of open access services and unified control and management aspects. The security driver for CHARISMA considers routing security, virtualized security (v-security), and trust sharing from an end-to-end service point of view. CHARISMA approaches the low-latency networking driver from multiple directions, including a hierarchical architecture approach, distributed content caching, and accelerated network hardware.

Based on the initial CHARISMA architecture, consisting of four levels of aggregation, and use case requirements, this deliverable D3.1 provides an overview of the initial architecture of control, management and orchestration plane. The architecture is aligned with the ETSI NFV architecture and consists of four groups of components: Virtualized Infrastructure (VI), Virtualized Network Functions (VNFs), Management and Orchestration (MANO), and Operations and Business Support Systems (OSS/BSS). The functional design of each of the components is also described including the management interfaces before the document ends with some concluding remarks.

This document is highly dependent upon the work carried out in WP1 of CHARISMA and hence, the high-level designs and solutions expressed hereby can be updated at a later stage by means of an extension to this deliverable.

1. Introduction

This document presents the initial design, both architectural and functional, of the Control, Management and Orchestration (CMO) plane being developed in the CHARISMA project. It also includes the definition of the building blocks and associated interfaces required to enable it. The work presented here represents the output of the first completed technical task of the project, T3.1 “CHARISMA open access secure management platform design”, which ended in month M6 of the project. It is worth mentioning that the CMO plane goes beyond the management of virtualised security (v-security), as suggested by the title of this deliverable, but also considers control and management of resources, both virtualized and physical, along with the orchestration of network services in the CHARISMA architecture.

The CHARISMA project has the ambitious objectives of creating an open access 5G network architecture, to allow virtualised slicing of network resources to different service providers (SPs) over the same common infrastructure, as a means to leverage down costs and achieve maximum efficient exploitation of available network resources. This represents a complex undertaking, since the traditional processes of network planning and operation become significantly more complex when infrastructure virtualisation becomes necessary. Software-defined networking (SDN) and network functions virtualisation (NFV) have emerged as major disruptive technologies in the communication networking paradigm, which enable the deployment of services as software functions running directly over the network commodity hardware. They are a mean to maximise efficient utilisation of network resources, achieve quicker operational functional changes, and manage the network over a quicker service provisioning time cycle. In addition, beyond SDN and NFV, achieving an Open Access infrastructure with competition between different network providers (NPs) (i.e., not just service providers) creates new complexity challenges, especially when the infrastructure owner needs to allocate virtual slices on top of the physical infrastructure, and lease the remote application programming interfaces (APIs) to offer access to the virtualised network resources. All this needs to be achieved whilst also maintaining the required security, both between end-users as well as creating the appropriate isolation between multiple tenants and access SPs and NPs, and also supporting the required quality of service and experience (QoS/E) and service level agreements (SLAs), e.g. with low latencies and high 5G data bandwidths.

This report describes the initial progress towards defining and specifying the different components and their logical interfaces for the CHARISMA open access secure management platform, so as to provide the required secure planning and operational phases. This means that the resulting CHARISMA management system needs to secure the virtual slice provisioning services enabling visibility up to Layer 7 classification, and securing the APIs (AuthN and AuthZ) with appropriate isolation and privacy.

This deliverable is organised as follows. After these short introductory remarks in Section 1, we provide a survey of the state-of-the-art in management systems of virtualised networks in Section 2, with particular regard to the SDN controllers, NaaS, Cloud environments, and NFV orchestration are also discussed in Section 2. Section 3 provides an overview of the key drivers underpinning the design of the CHARISMA management system, including key aspects for CHARISMA such as open access, security and low latency. In Section 4 the high-level design of the proposed CHARISMA control, management and orchestration plane

is provided, including the component building blocks, their functional design and their internal interfaces. Finally, some concluding remarks are offered in Section 5.

2. Management systems of virtualised networks

SOTA Survey

The following section presents a concise review of the current state-of-the-art (SOTA) technologies and industry/academic initiatives that are relevant to the CHARISMA Control, Management and Orchestration plane and SDN/NFV technologies.

2.1. Software Defined Network Controllers

A key abstraction of the SDN paradigm is the separation of the network control and forwarding planes. Conceptually, in SDN networks, resources are treated as a dynamic collection of arbitrarily connected forwarding devices with minimal intelligence. The control logic is implemented on top of a so-called SDN controller. The controller is a logically centralised entity which is responsible for a set of tasks, including the extraction and maintenance of a global view of the network topology and state, as well as the instantiation of forwarding logic appropriate to a given application scenario. In practice, the controller manages connections to all substrate switches using a southbound protocol such as OpenFlow, and installs, modifies and deletes forwarding entries into the forwarding tables of the connected switches by using protocol specific control messages.

The NOX controller [1] was the first widely available OpenFlow controller. NOX was originally developed by Nicira and released as open-source software. Due to its early availability and simplicity, NOX quickly became the de-facto reference design for OpenFlow controllers. As a result, it has been used to test new OpenFlow features, novel controller ideas and has been employed extensively in research and feasibility studies. NOX applications – called modules – are implemented using the C programming language. NOX is event based; each module essentially consists of a collection of call-back functions, triggered by the arrival of specific OpenFlow protocol messages. A spin-off of NOX called POX [2] enables the use of Python for programming modules. While NOX/POX is extremely versatile it is not primarily aimed for production use, as it is not optimised for performance and stability and lacks resilience features. Other controller frameworks aimed at deployment in production environments, include Beacon [3], Maestro [4] and FloodLight [5], all of which are implemented in Java. FloodLight is the open source basis for Big Switch's commercial OpenFlow controller.

OpenDayLight [6] is currently the newest and also largest SDN controller platform. It is backed by the Linux Foundation and developed by an industrial consortium, which includes Cisco, Juniper and IBM, among many others. OpenDayLight includes numerous functional modules, which are interconnected by a common service abstraction layer. Further, OpenDayLight provides a flexible northbound interface using Representation State Transfer APIs (REST APIs), and includes support for the OpenStack cloud platform.

Specifically, the current OpenDaylight release is built upon four "layers", i.e.:

- Technology-specific plug-ins, for managing SDN and non-SDN devices with various network configuration protocols;

- A Service Abstraction Layer, unifying the capabilities of the underlying technology-specific plug-ins;
- A core of basic network services, such as topology management, host tracking etc.;
- A set of northbound APIs (REST-based) for communicating with network management applications.

2.2. NaaS Platforms

Network service delivery and management models remain an area of on-going evolution. The network as a service concept represents an interesting service model from which CHARISMA can benefit. Network as a service is a business model for delivering network services virtually over any network on a pay-per-use basis. Despite not being a new concept, its development has been hindered by some of the same concerns that have affected also other cloud computing services, such as high availability or service level agreements. In essence, network becomes a utility, paid for just like any other utility (e.g., water, electricity or heat).

In this context, NaaS can be adopted as a new management and provisioning model, which needs to be evaluated in the context of the Network Management Architecture (NMA) principles, latest network management standards, and the trends in Next-Generation Operations Support System (NG-OSS).

NaaS capabilities are of particular relevance for the value they can provide to virtualization services, as they can extend the IaaS model from data centres into access and transport networks.

Below, NaaS related platforms are considered for its relevance and particular interest to CHARISMA.

2.2.1. OpenNaaS

The NaaS model has been instantiated in the OpenNaaS platform for easy prototyping and proof casing of its concepts. OpenNaaS [7], [8] is an open-source framework, which provides tools for managing the different resources present in any network infrastructure. The software platform was created in order to offer a neutral tool to the different stakeholders comprising an open access network (OAN). It allows them to contribute and benefit from a common NaaS software-oriented stack for both applications and services. It is based on a lightweight, abstracted, operational model, which is decoupled from actual vendors' specific details, and is flexible enough to accommodate different designs and orientations. In fact, the OpenNaaS framework provides tools to implement the logic of an SDN-like control and management plane on top of the lightweight abstracted model.

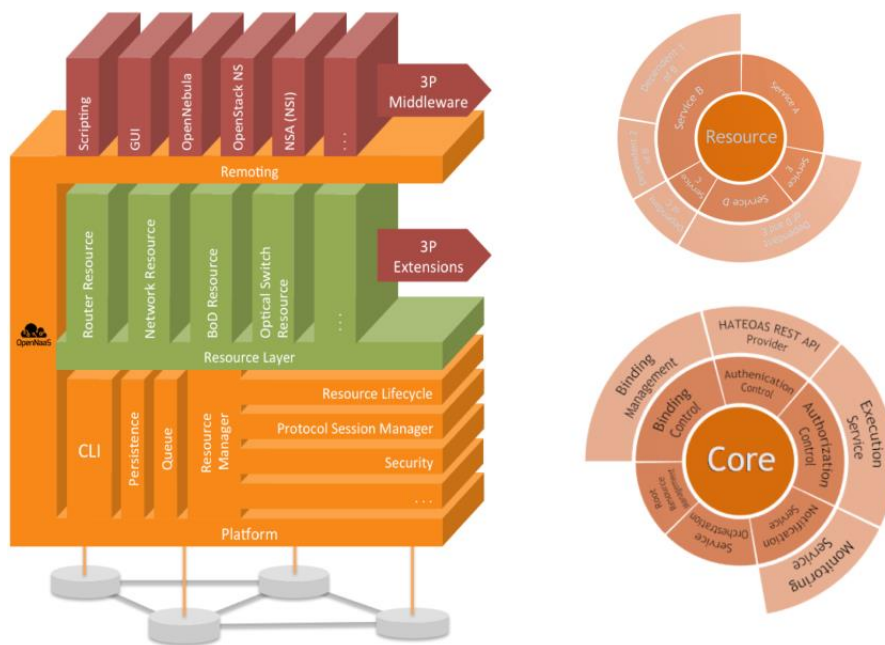


Figure 2-1: OpenNaaS Architecture and NaaS Resource Abstraction

Figure 2-1 depicts the layered architecture of the framework, with the platform layer, the resource abstraction layer with the NaaS manageable units, and the upper layer, where the network intelligence resides, as well as the integration of the framework with third-party components. Besides, the resource abstraction, the core platform concepts are also depicted. Different OpenNaaS deployment examples can be found in the following list of European projects extending the OpenNaaS framework: OFERTIE [9], CONTENT [10] and SODALES [11]. Furthermore, authors in [9] used OpenNaaS in order to build a first proof-of-concept pilot for VNF creation and management.

2.2.2. OpenStack Neutron

OpenStack Neutron [13], historically known as Quantum, is an OpenStack project focused on delivering Networking-as-a-Service (NaaS). Neutron provides a way for organisations to make it easier to deliver NaaS in the cloud and provides REST APIs to manage network connections for the resources managed by other OpenStack services.

Neutron provides native multi-tenancy support (isolation, abstraction and full control over virtual networks), letting tenants create multiple private networks and control the IP addressing on them, and exposes vendor-specific network virtualisation and SDN technologies.

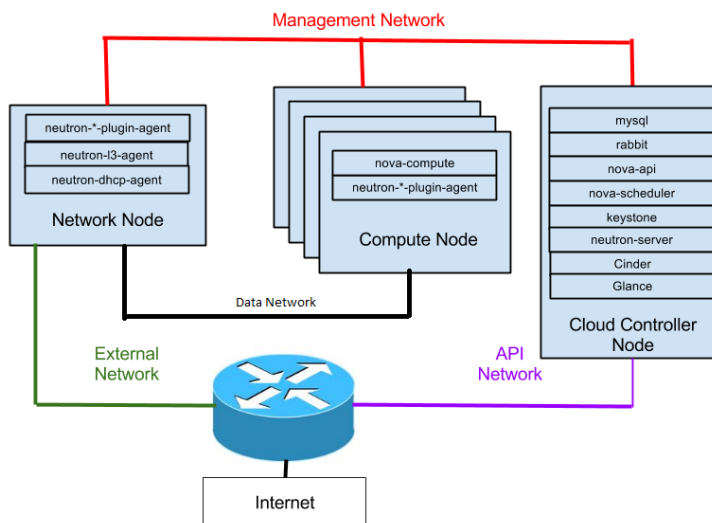


Figure 2-2: OpenStack Neutron Architecture [13]

Neutron includes a growing list of plugins that enable interoperability with various commercial and open source network technologies, including routers, switches, virtual switches and SDN controllers. Starting with the Folsom release, Neutron is a core and supported part of the OpenStack platform. However, it is a standalone and autonomous service that can evolve independently to OpenStack.

2.2.3. OpenDaylight Virtual Tenant Network (VTN)

OpenDaylight Virtual Tenant Network (VTN) [14] provides multi-tenant virtual networking on an SDN controller. VTN provides an abstraction that enables the complete separation of the logical plane from the physical plane of the network. This allows users to design and deploy virtual networks for their customers without needing to know the physical network topology or underlying operating characteristics. The VTN also allows the network designer to construct virtual networks using common L2/L3 network semantics.

VTN allows the users to define the network with the look and feel of a conventional L2/L3 network. Once the network is designed on VTN, it is automatically mapped onto the underlying physical network, and then configured on the individual switches leveraging an SDN control protocol. The definition of the logical plane makes it possible not only to hide the complexity of the underlying network, but also to better manage network resources. It achieves a reduction in the reconfiguration time of network services and minimises network configuration errors.

2.2.4. FlowVisor

FlowVisor [15] is the ON.LAB network slicer, which allows multiple tenants to share the same physical infrastructure. A tenant can be either a customer requiring his own isolated network slice; a sub-organisation that needs its own slice; or an experimenter who wants to control and manage some specific traffic from a subset of endpoints. FlowVisor acts as a transparent proxy between OpenFlow switches and various guest network operating systems. It supports network slicing and allows a tenant or an experimenter to control and manage some specific traffic from a subset of end points. This approach enables multiple experimenters to use a physical OpenFlow network without interfering with each other.

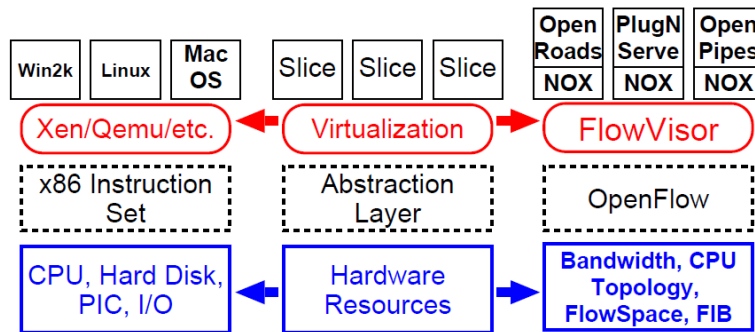


Figure 2-3: FlowVisor Architecture [15]

FlowVisor, originally developed at Stanford University, has been widely used in experimental research and education networks to support slicing where multiple experimenters get their own isolated slice of the infrastructure and control it using their own network OS and a set of control and management applications. FlowVisor has been deployed on the Stanford production network and sponsors, such as GENI, Internet2, NEC and Ericsson, have been contributing to it and using it in their research labs. The SDN research community considers FlowVisor an experimental technology, although Stanford University has run FlowVisor in its production network since 2009. FlowVisor lacks some of the basic network management interfaces that would make it enterprise-grade. For example, it currently does not support any CLI or Web-based administration console but requires users to make changes to the technology with configuration file updates.

2.2.5. OpenVirtex

OpenVirteX [16] is a network hypervisor that can create multiple virtual and programmable networks on top of a single physical infrastructure. Each tenant can use the full addressing space, specify their own topology, and deploy the network OS of their choice.

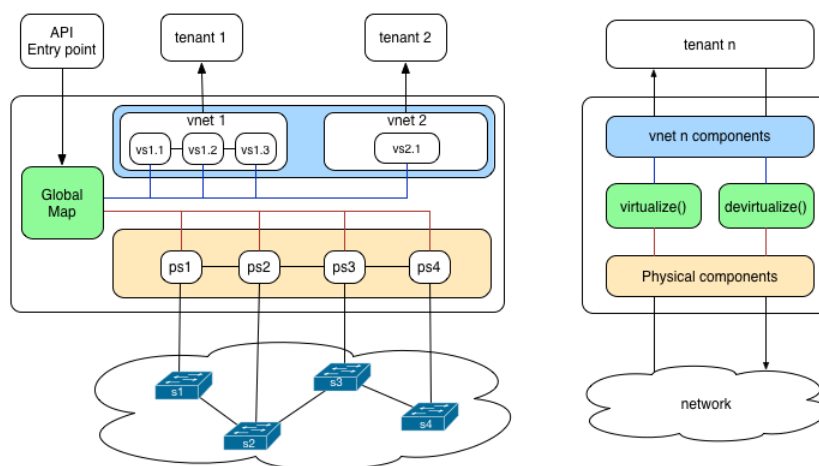


Figure 2-4: OpenVirtex Architecture [16]

OpenVirteX is actually a network hypervisor that enables operators to provide networks whose topologies, management schemes, and use cases are under the full control of their tenants. More specifically,

OpenVirteX builds on OpenFlow as the protocol and FlowVisor for design. In this respect they share some common properties i.e. act as proxies between tenants and the underlying physical infrastructure. Unlike FlowVisor however, OpenVirteX provides each tenant with a fully virtualised network featuring a tenant-specified topology and a full header space.

2.2.6. OpenContrail

Another framework that is attracting attention is led by Juniper Networks, named OpenContrail [17]. It is a modular project that provides an environment for network virtualisation and published northbound APIs. In particular, the network virtualisation is provided by means of a set of building blocks and high level policies; it integrates an SDN controller to support network programmability and automation, and a well-defined data model to describe the desired state of the network; an analytics engine is designed for very large scale ingestion and querying of structured and unstructured data. It also provides an extensive REST API to configure and gather operational and analytics data from the system.

2.3. Cloud Environments

Cloud management platforms are integrated tools that provide management of cloud environments. These tools incorporate self-service interfaces, provisioning of system images, enabling metering and billing, and providing some degree of workload optimisation through established policies. Through the self-service interface (e.g. based on OCCl) the user can request virtual infrastructure. This request is issued to a Cloud Controller, which provisions this virtual infrastructure somewhere on available resources within a Data Centre (DC). The Cloud Controller provides the central management system for cloud deployments.

The most popular cloud management platforms include open source solutions such as OpenStack, CloudStack and Eucalyptus and commercial solutions from Microsoft and VMware. This section provides an overview of some of these solutions, on the Cloud Controller component. In the CHARISMA context, the Cloud Controller is a key component that can deliver end-to-end provisioning of virtual infrastructure, to enable full control over it and to provide a detailed and real-time view of the infrastructure load.

2.3.1. OpenStack

OpenStack [18] is a cloud OS that controls large pools of compute, storage and networking resources throughout a DC, all managed through a dashboard that gives administrators control while empowering their users to provision resources through a web interface. As an open source solution, OpenStack is developed and supported by a global collaboration of developers and cloud computing technologists, and seeks to deliver solutions for all types of clouds by being simple to implement, scalable and being feature rich. The technology consists of a series of interrelated projects delivering various components for a cloud infrastructure solution. All OpenStack source code is available under an Apache 2.0 license.

OpenStack has a modular design that enables integration with legacy and third party technologies. It is built on a shared-nothing, messaging-based architecture with modular components, each of which manages a different service; these services, together instantiate an Infrastructure-as-a-Service (IaaS) Cloud. The primary component of the cloud operating environment is the Nova compute service. Nova compute

orchestrates the creation and deletion of compute/VM instances. Nova is designed to operate as much as possible as hypervisor-agnostic. It works with open source libraries such as libvirt. Similar to other OpenStack components, Nova is based on a modular architectural design where services can be co-resident on a single host or, more commonly, on multiple hosts.

The core components of Nova include the following:

- The nova-api accepts and responds to end-user compute API calls. It also initiates most of the orchestration activities (such as running an instance) as well as enforcing some policies;
- The nova-compute process is primarily a worker daemon that creates and terminates VM instances via hypervisor APIs (XenAPI for XenServer/XCP, libvirt for KVM or QEMU, VMWareAPI for vSphere, etc);
- The nova-scheduler process keeps a queue of VM instance requests and for each request it determines where the VM instance should run (specifically, which compute node it should run on).

The Nova service itself does not come with a hypervisor, but manages multiple hypervisors, such as KVM or ESXi. Nova orchestrates these hypervisors via APIs and drivers. For example, Hyper-V is managed directly by Nova and KVM is managed via libvirt, while Xen and vSphere can be managed directly or through management tools such as libvirt and vCenter for vSphere, respectively.

2.3.2. Eucalyptus

Eucalyptus (Elastic Utility Computing Architecture Linking Your Programs To Useful Systems) [19] is an open-source Cloud that provides on-demand computing instances and shares the same APIs as Amazon's EC2 cloud. Eucalyptus was designed as a highly-modular framework in order to enable extensibility with minimal effort (Eucalyptus Systems, Inc, 2014). The Cloud Controller (CLC) in Eucalyptus acts as the Cloud entry-point by exposing and managing the virtualised resources. The CLC offers a series of web services oriented towards resources, data and interfaces (EC2-compatible and Query interfaces). In addition to handling incoming requests, the CLC acts as the administrative interface for cloud management and performs high-level resource scheduling and system accounting. The CLC accepts user API requests from command-line interfaces like euca2ools or GUI-based tools like the Eucalyptus Management Console and manages the underlying compute, storage, and network resources.

2.3.3. Cloudstack

Apache CloudStack [20] is open source software designed to deploy and manage large networks of virtual machines, as a highly available, highly scalable Infrastructure-as-a-Service (IaaS) cloud computing platform. CloudStack is used by a number of service providers (e.g. BT) to offer public cloud services, and by many companies to provide an on-premises (private) cloud offering, or as part of a hybrid cloud solution. CloudStack is a turnkey solution that includes the entire "stack" of features most organisations want with an IaaS cloud: compute orchestration, Network-as-a-Service, user and account management, a full and open native API, resource accounting and a high quality User Interface (UI).

CloudStack is a framework that allows pooling of computing resources in order to offer IaaS cloud services that can be used to provide IT infrastructure such as compute nodes (hosts), networks and storage as a service to the end users on demand. CloudStack Management Server is the main component of the framework, consisting of managing resources such as hosts, storage devices and IP addresses. The Management Server runs on a dedicated host in a Tomcat container and requires a MySQL database for persistence. The Management Server controls allocation of VMs to hosts and assigns storage and IP addresses to VM instances. This component also controls or collaborates with the hypervisor layers on the physical hosts over the management network and thus controls the IT infrastructure.

2.3.4. VMware vCloud Suite

VMware's vCloud Suite [21] - is a comprehensive, integrated cloud platform for building and managing cloud environments. Tools for cloud management are delivered through VMware vCenter Server, a centralised and extensible platform for managing virtual infrastructure. The tools included in the vCenter Server framework support: configuration of ESX servers and VMs, performance monitoring throughout the entire infrastructure; use of, events and alerts. The objects in the virtual infrastructure can be securely managed with roles and permissions.

2.4. NFV Orchestrators

2.4.1. Open source projects

2.4.1.1. OpenBaton

OpenBaton [22] is an ETSI NFV compliant Network Function Virtualization Orchestrator (NFVO). OpenBaton was part of the OpenSDNCore (www.opensdncore.org) project started almost three years ago by Fraunhofer FOKUS with the objective of providing a compliant implementation of the ETSI NFV specification.

OpenBaton is easily extensible. It integrates with OpenStack and provides a plugin mechanism for supporting additional VIM types. It supports Network Service management either using a generic VNFM or interoperating with VNF-specific VNFM. It uses different mechanisms (REST or PUB/SUB) for interoperating with the VNFMs. Its initial focus was to provide the main functionalities for provisioning and managing Network Services, however in its future releases new additional features will provide mechanisms for increasing the automation in NS management. Those new features will include auto-scaling, fault management, TOSCA, etc.

The orchestrator implements the key functionalities of the MANO architecture. Specifically, it:

- Currently uses the OpenStack as first integrated NFV PoP VIM;
- Maintains an overview on the infrastructure, supporting dynamic registration of NFV PoPs;
- Receives virtual network function packages from the different users including VNF images and virtual network functions descriptors (VNFDs);

- Deploys on-demand the VNFs on top of an infrastructure consisting of multiple data centre instances (NFV PoPs);
- Deploys in parallel multiple slices one for each tenant, consisting of one or multiple VNFs.

Through this functionality, the orchestrator provides a multi-tenant environment distributed on top of multiple cloud instances.

The Generic VNFM in OpenBaton represents an implementation of the MANO function. It is tightly coupled with the OpenBaton EMS, which runs as a software within the deployed Network Functions. It can be easily extended to support the management of third party VNFs. The OpenBaton VNFM can execute the following operations:

- Is instantiated on demand by the NFVO;
- Request to the NFVO the allocation of specific resources for the virtual network instance;
- Can request from the NFVO the instantiation, modification, starting and stopping of the virtual services (or directly to the VIM);
- Instructs the generic OpenBaton EMS to save and to execute specific configuration scripts within the virtual machine instances.

2.4.1.2. **Tacker**

Tacker [23] is a new OpenStack project that aims at building an open NFV orchestrator with a general purpose VNF manager to deploy and operate virtual network functions on an NFV Platform. It is based on the ETSI MANO Architectural Framework and aims at providing a full functional stack to orchestrate VNFs end-to-end.

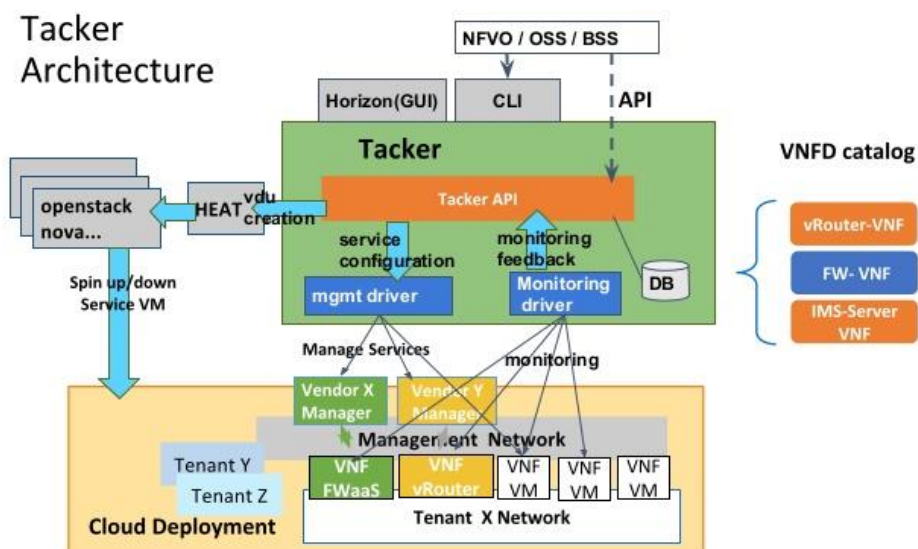


Figure 2-5: Tacker Architecture [23]

Tacker offers features like a VNF Catalogue, basic VNFM Life Cycle Management, a VNF Configuration Management Framework, and a VNF KPI Health Monitoring Framework. The VNF Catalogue makes use of

TOSCA for VNF meta-data definition and OpenStack Glance to store and manage the VNF images. The Tacker VNFM Life Cycle Management takes care of instantiation and termination of VMs, self-healing and auto-scaling, and VNF image updates. It also takes care of interfaces to vendor specific element management systems. OpenStack Tacker is under heavy development. As of the writing of this report, several crucial features, such as service function chaining and VNF decomposition, are still missing and are under discussion.

2.4.1.3. *OpenMano*

OpenMano [24] is an open source project that provides a practical implementation of the reference architecture for Management & Orchestration under standardization at ETSI's NFV ISG.

The OpenMano software consists of three major components, each one of them covering one layer of requirements associated to the MANO architecture of the ETSI NFV ISG:

- **openvim**: reference implementation of an NFV VIM. It interfaces with the compute nodes in the NFV Infrastructure and an SDN controller in order to provide computing and networking capabilities and to deploy virtual machines. It offers northbound interfacing, based on REST (i.e. openvim API), where enhanced cloud services are offered including the creation, deletion, and management of images, flavours, instances, and networks. The implementation of this component follows the recommendations in ETSI-PER001, NFV Performance and Portability Best Practices.
- **openmano**: reference implementation of an NFV Orchestrator. It interfaces with an NFV VIM through its API and offers a northbound interface based on REST (openmano API), where NFV services are offered including the creation and deletion of VNF templates, VNF instances, network service templates and network service instances.
- **openmano-gui**: web UI to interact with openmano server, through its northbound API, in a user-friendly way.

Figure 2-6 below depicts the relationship of the different software components of the OpenMano platform with the ETSI NFV reference architecture. In the right side of the image, in blue boxes, the openmano components are depicted. It can be seen how *openmano* performs both Orchestration and VNF Management, while *openvim* is devoted to virtualized infrastructure management.

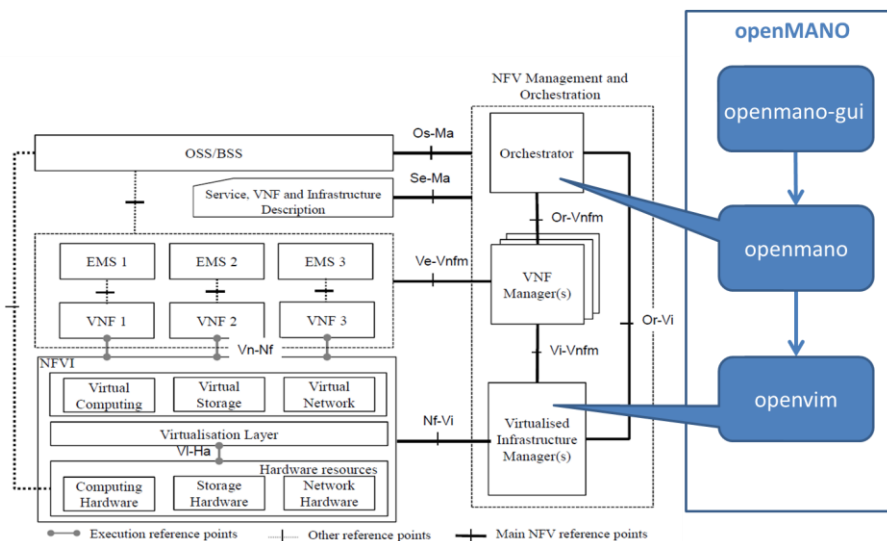


Figure 2-6: OpenMano scope diagram [24]

2.4.2. Project Based Implementations

2.4.2.1. T-NOVA Orchestrator (TENOR)

FP7 T-NOVA [25] specifically focuses on the aspects of Network Functions Virtualisation (NFV), which aimed to introduce a novel enabling framework, allowing operators not only to deploy Virtualized Network Functions (VNFs) for their own needs, but also to offer them to their customers, as value-added services. Virtual network appliances (gateways, proxies, firewalls, transcoders, analyzers etc.) can be provided on-demand as-a-Service, eliminating the need to acquire, install and maintain specialized hardware at customers' premises.

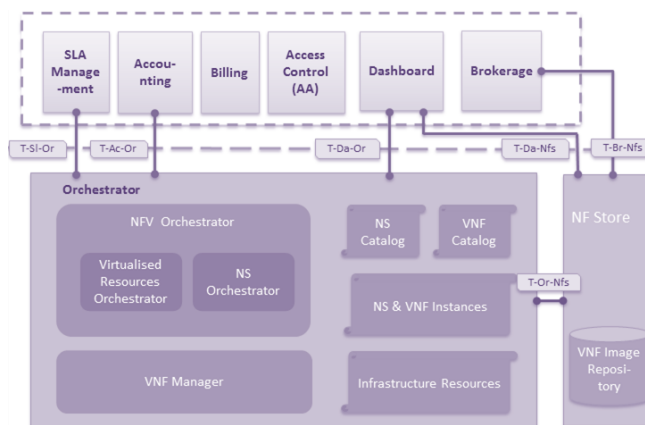


Figure 2-7: High-level view of T-NOVA Orchestrator [25]

For these purposes, T-NOVA designs and implements a management/orchestration platform named TENOR for the automated provision, configuration, monitoring and optimization of Network Functions-as-a-Service (NFaaS) over virtualised Network/IT infrastructures. In other words, T-NOVA combines IT/cloud virtualisation and Network-as-a-Service concepts to offer a complete end-to-end Cloud Network service. The T-NOVA orchestrator architecture is depicted in Figure 2-7.

Furthermore, in order to facilitate the involvement of diverse actors in the NFV scene and attract new market entrants, T-NOVA established a “NFV Marketplace”, in which network services and functions by several developers can be published and brokered/traded. Via the Marketplace, customers can browse and select the services and virtual appliances, which best match their needs, as well as negotiate the associated SLAs and be charged under various billing models. A novel business case for NFV is thus introduced and promoted.

2.4.2.2. **UNIFY Orchestrator**

UNIFY (Unifying Cloud and Carrier Networks) project [26] has the goal of increasing the potential of virtualization and automation across the whole networking and cloud infrastructure. The project is focused on enablers of a unified production environment and develops an automated, dynamic service creation platform, leveraging a fine-granular service chaining architecture. UNIFY proposes a service abstraction model and a service creation language to enable dynamic and automatic placement of networking, computing and storage components across the infrastructure. UNIFY orchestrator includes optimization algorithms to ensure optimal placement of elementary service components across the infrastructure. Moreover, UNIFY has the intent to research new management technologies and develop a Service Provider DevOps concept to address the dynamicity and agility of new services. Therefore, the UNIFY consortium is researching, developing and evaluating the means to orchestrate, verify and observe end-to-end service delivery from home and enterprise networks through aggregation and core networks to data centres.

UNIFY focuses on enabling virtualization and automation across the whole networking and cloud infrastructure in a unified manner. It envisions full network and service virtualization to enable rich and flexible services and operational efficiency.

Through the design of universal hardware architectures UNIFY aims at improving the intelligence and flexibility of the network and so open up opportunities for new converged fixed and mobile end-user service offerings, while also enabling advanced programmability and efficient virtualization, providing means to reduce the cost of new service creation and operation.

2.4.2.3. **MCN Orchestrator**

The Mobile Cloud Networking (MCN) [27] project targets the integration between the Cloud and Telco worlds, allowing operators to benefit from the principles of virtualization.

The project focuses, in particular, on mobile operators. For this reason, the main target is to fully cloudify the whole components of a mobile network operation, namely:

- Access (RAN - Radio Access Network);
- Core (EPC – Evolved Packet Core);
- Services (IMS – IP Multimedia Subsystem, CDN – Content Delivery Networks, DSS – Digital Signage);
- Operational Support Systems (OSS) (Provisioning, Monitoring, SLA Management);
- Business Support Systems (BSS) (CRM – Customer Relationship Management, Charging, Billing).

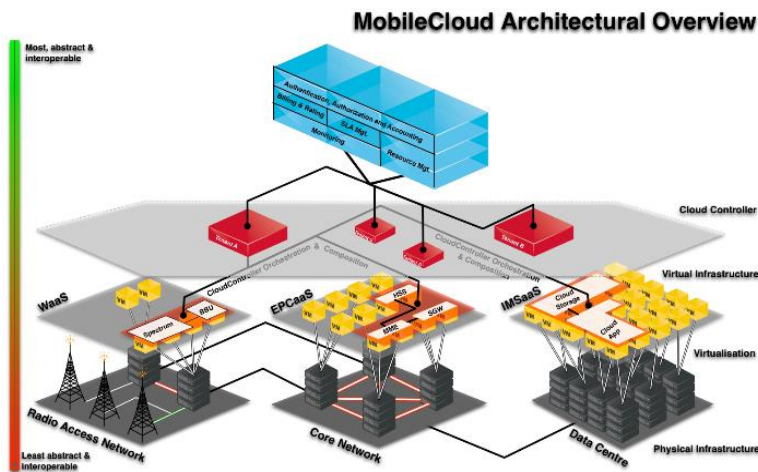


Figure 2-8: Mobile Cloud Networking Vision [27]

Beyond virtualization, the project explores the “as-a-Service” (XaaS) concept, where the functions are provided as a full operational service. That means the customer of the service does not need to worry about implementation, deployment and dimensioning details of it. This approach allows the easy creation of “end-to-end (e2e)” services by composition of basic services. As an example, the creation of an MVNOaaS service can be considered as the composition of RANaaS+EPCaaS+IMSaaS.

Mobile Cloud Networking overall architecture design (MCN Consortium, 2013) is mainly governed by service oriented design principles. Every service in MCN has the same provisioning and lifecycle management pattern and architecturally follows the global MCN reference architecture.

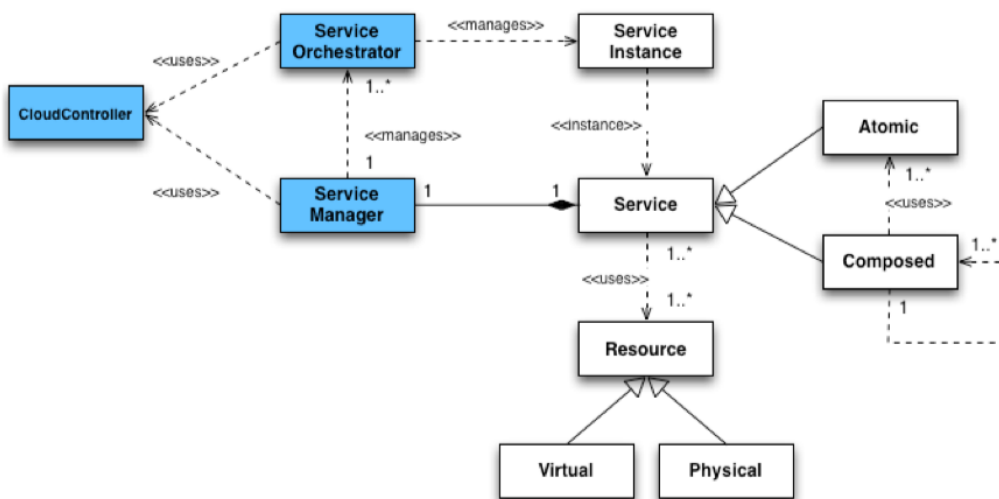


Figure 2-9: MCN Architectural Entities Relationship [27]

A brief description of key components follows:

- **Service Manager:** provides an external interface to the Enterprise End User (EEU) and is responsible for managing service orchestrators, it has business and technical management functions.

- **Service Orchestrator:** it oversees the end-to-end orchestration of a service instance. It is responsible for managing the Service Instance and in particular its components (SIC), once it is created and running.
- **Cloud Controller:** provide the signalling and management interfaces to enable the common (northbound), and technology-specific (southbound) control planes. It provides both atomic and support services required for realizing SO needs. The main MCN architectural entities that interact most with the Cloud Controller are the SM and SO.

Service Orchestrator implementation in MCN project is service specific as it depends on the domain knowledge of the respective NF it is implementing. Some of the prominent NFs being virtualized and managed as a service in MCN are – EPC, IMS, DNS, OSS/BSS (RCB), AAA, CDN, etc.

2.4.3. Market Ready Orchestrator Solutions

2.4.3.1. Cloudify

Cloudify [28] provides the full end-to-end lifecycle of NFV orchestration – from installation and deployment through monitoring of KPIs, with auto-healing and auto-scaling based on metrics, all through a simple TOSCA-based YAML blueprint. This significantly simplifies the complexities involved with exposing networking elements to the user by abstracting the networking piece of the deployment into an application blueprint, all while enabling the hardening of security to match the application topology exactly.

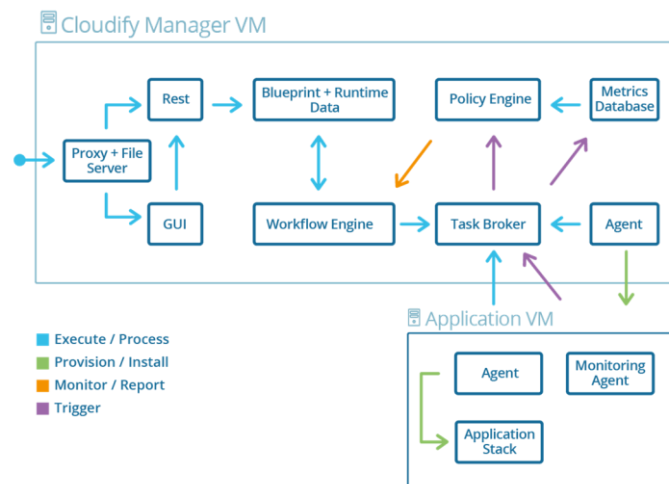


Figure 2-10: Cloudify Manager overall Architecture [28]

Cloudify brings together a variety of tools that are used throughout the different stages of the application lifecycle:

- Configuration management tools, such as Chef, Puppet, Fabric and Docker;
- Infrastructure orchestration tools, such as OpenStack Heat;
- Logging and monitoring tools, such as logstash and Elasticsearch;
- Real-time analytics tools, such as Reimann.io.

Most automation tools are focused on the installation and configuration phases of application management, while in reality much of the management takes place after the application has been deployed.

Cloudify was redesigned to eliminate the boundaries between orchestration and monitoring, providing a mechanism to automatically react to monitored events with the appropriate corrective measures. The building blocks of custom workflows, through a workflow engine, and the TOSCA modelling language enable the automation of any process and any stack, including monitoring and custom policies for automated triggering of such corrective measures to provide auto-healing and auto-scaling capabilities.

2.4.3.2. Cyan NFV Orchestrator

Cyan NFV Orchestrator [29] manages, automates and orchestrates services that leverage physical and/or virtual resources across the telco cloud and the WAN. It relies on a flexible architecture, which integrates NFV, Cloud and/or Multi-Domain Service Orchestration capabilities based on an open system that ensures interoperability with different OSS platforms, cloud management systems, SDN controllers, network elements and VNFs.

NFV capabilities of Cyan NFV Orchestrator, namely Planet Orchestrate, comply with ETSI’s NFV ISG Management and Orchestration (MANO) framework, providing intelligent multi-vendor NFV Orchestration (NFVO) capabilities to manage and automate the processes associated with on-boarding and removing VNFs, as well as service chaining, along with an intuitive HTML user interface for managing all service resources. It also offers support for multiple cloud management systems such as OpenStack and VMware, simplifying the adoption of SDN and NFV.

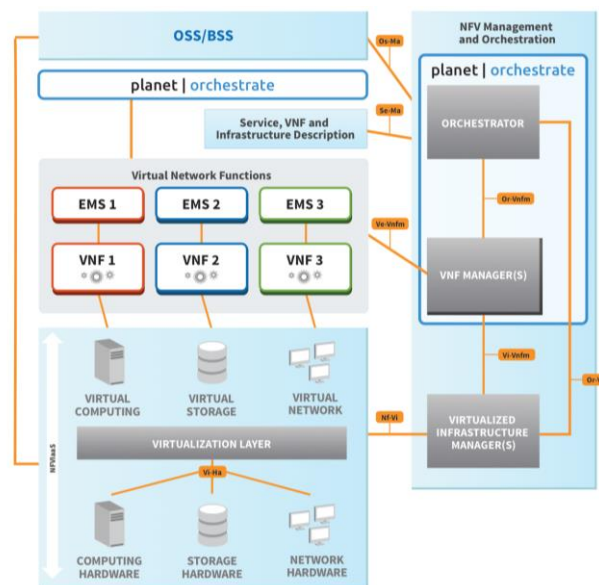


Figure 2-11: CYAN Orchestrator Architecture [29]

Planet Orchestrate incorporates service-centric information templates to create a highly programmable system that is agnostic to the type of resources being orchestrated. Focused on enabling operators to leverage the agility, flexibility and dynamism offered by SDN and NFV architectures, it supports a degree of automation not available via conventional management systems.

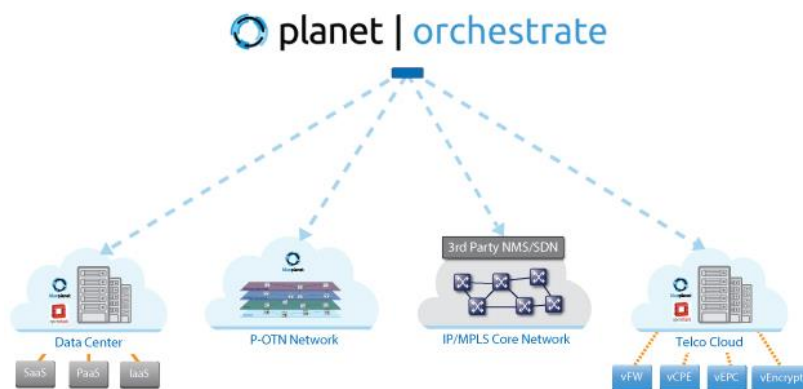


Figure 2-12: CYAN Multi-Vendor/Multi-Domain Framework [29]

Planet Orchestrate is optimized for multi-vendor, multi-domain environments. Support for open APIs enables interoperability and interaction with the broadest array of software applications, as well as both physical and virtual resources.

Planet Orchestrate seamlessly provisions and manages “service chains” comprised of physical network elements and SDN/NFV-enabled virtual components across multiple domains. Using APIs, Planet Orchestrate can be integrated with SDN controllers such as Cyan’s Planet Operate, as well as 3rd party controllers, element/network management systems, and cloud management platforms. This breaks down management silos, allowing network operators to orchestrate services from end-to-end.

2.4.3.3. HP NFV Director

OpenNFV [30] is a comprehensive project launched by HP, built around a proposed open reference architecture, encompassing a service portfolio, and enforced by an ecosystem of ISVs, NEPs and application developers.

HP architecture is aligned with the ETSI model, and HP has a number of active contributors in the NFV ISG. OpenNFV main components are a NFV Infrastructure and a NFV Orchestrator module, in turn based on HP Converged Infrastructure and HP Converged Cloud propositions. It also capitalizes on the SDN role, and on HP’s SDN technology assets. It is a modular architecture, vendor agnostic, and allow a modularized approach to NFV take-up. Figure 2-13 below shows the open model behind the OpenNFV architecture.

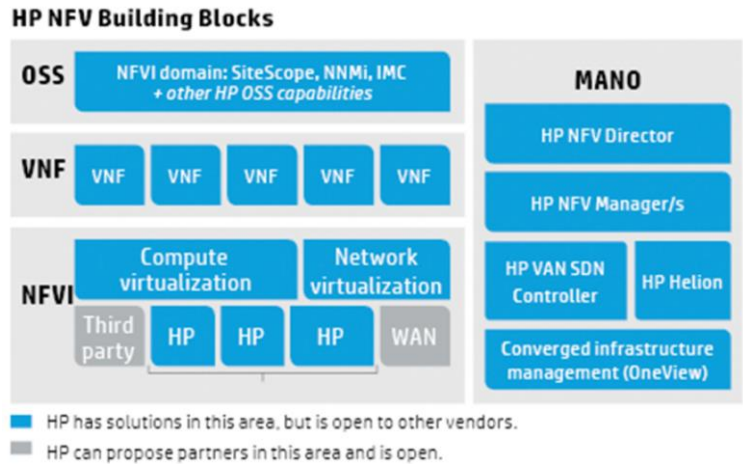


Figure 2-13: HP NFV Building Blocks [30]

The virtualized NFV infrastructure layer includes compute, storage and networking plus their element management components. It is a logical infrastructure layer, designed to support both physical and virtualized resources at the same time. The NFVI layer offers pre-integrated modules supporting virtualization and performance enhancement technologies like SR-IOV and DPDK, to ensure that the hardware horsepower is suitable to seamlessly virtualize network functions without performance degradation. Multiple hypervisors can be supported, like VMware ESX, Microsoft Hyper-V, and Linux KVM.

On the network virtualization side, the architecture can seamlessly support automated provisioning of both physical and virtual networks. HP physical switches also support L2/L3 forwarding, bridging the gap between legacy and SDN-controlled networks. A comprehensive heterogeneous network management platform is also available. The NFV Orchestrator is hypervisor-agnostic, so it can support different solutions of both proprietary and open source kinds.

2.4.3.4. **CISCO Orchestrator**

Cisco’s NFV orchestration solutions [31] include Cisco Evolved Services Platform (ESP) and Cisco Evolved Programmable Network (EPN), which are aligned with the Cisco Open Network Strategy. The NFV management and orchestration solution includes the Services Orchestrator, the VNF Manager, and the SDN Controller.

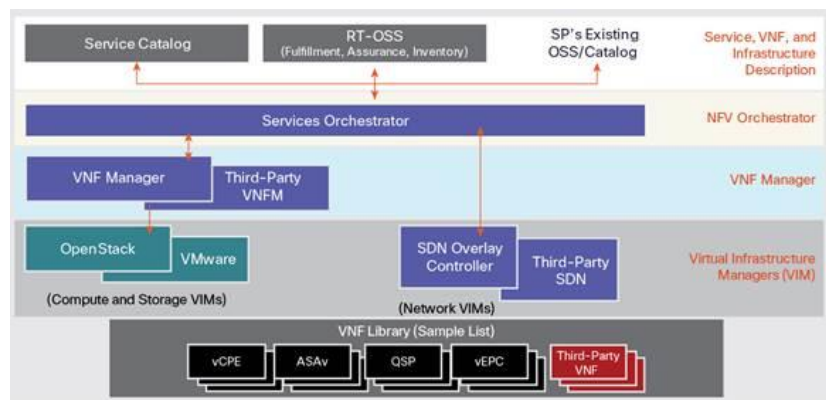


Figure 2-14: Cisco NFV Management and Orchestration Architecture [31]

The Services Orchestrator operates at the resource-facing services layer (RFS) and is responsible for providing the overall lifecycle management at the network service level. The Services Orchestrator provides a standards-based northbound interface for transparent integration with systems that operate at the customer-facing services (CFS) layer such as the service catalogue or RT-OSS.

The VNF Manager provides scalable, automated VNF lifecycle management, including the creation, provisioning, and monitoring of both Cisco and third party VNFs. Unlike many other vendor solutions, the Cisco implementation is agentless, helping optimize the overall performance in VNF management.

The SDN Controller is responsible for connecting the virtualized services (a VNF or a set of chained VNFs) to the service provider VPNs, the Internet, or both. It is designed around open standards and APIs and uses a holistic systems-based approach to managing multivendor and multitenant data centres.

2.5. CHARISMA Management Approach

The envisioned architecture for the CHARISMA management platform leverages on SDN, NFV, and Network Slicing principles, aspects and results. Thus, it is useful to analyse the related work and prior-art, which is studied, debated and showcased in several commercial or open source initiatives and research projects. Within section 2, we discussed the current SoTA in the fields of SDN controllers, NaaS platforms, cloud environments and NFV orchestrators. It is out of the scope of CHARISMA project to research and develop another proposal on these areas. Existing solutions from the surveyed ones will be considered, and extended to meet CHARISMA objectives on need basis, that have been tested as they are already deployed and in use.

CHARISMA will implement a holistic management platform, combining cloud, network slicing, SDN, and NFV technologies while focusing on security aspects. Although newly introduced technologies as SDN and NFV offer the potential for improving security, they also introduce new vulnerabilities and a period of uncertainty, as functions and equipment go from being rooted in established network approaches to being much more dynamic. CHARISMA will provide the necessary mechanisms and controls to address the complex security challenges of today's heterogeneous networks, including both physical and virtual elements. It combines a set of assets such as security policy management, decision control for threat detection, virtualization isolation, identity and access management and proactive traffic and resource monitoring to deliver security in current 5G networks. The details of the CHARISMA CMO plane are presented in Section 4. In the next section, we highlight the main drivers behind CHARISMA CMO plane.

3. Key drivers for CHARISMA management system

The Control, Management and Orchestration (CMO) plane of CHARISMA is conceived to provide the adequate platform to deliver an intelligent hierarchical routing architecture combining the open access, low-latency and end-to-end security concepts. The CHARISMA architecture that is currently being defined in WP1 has three main drivers:

- An open access platform where Virtual Network Operators (VNOs) can share the same network infrastructure but different virtual slices, considering the converged wireless/wireline advanced 5G networks.
- A Secured system.
- Providing a very low-latency (<1ms) platform.

CHARISMA aims for a flexible and programmable CMO plane enabled by a platform based on SDN and NFV concepts, to offer VNOs the specified functionality with additional cost savings. Within this context, CHARISMA will provide a cloud infrastructure platform targeting the development of a 5G network infrastructure where the SDN and NFV paradigms will help the implementation of a virtualised secured open access solution that will allow different service providers to share the same common infrastructure so that operational and potentially capital costs are leveraged down and an efficient usage of the available resources is implemented.

In this section, we will go through these three areas that are motivating the CHARISMA CMO plane: open access, security and a low latency platform.

3.1. Open Access

Open access is a wide concept with different interpretations and operation models depending on the incumbents' approach and the legislation of the different countries in EU. Before describing how CHARISMA approaches this issue, it is important to explain the different alternatives in order to horizontally segment the telecom network value chain and understand how the architecture of CHARISMA fits into open access networking.

The FP7 – ICT – 318600 SODALES [11] and FP7 – ICT– GA 249025 OASE [12] projects have already studied the implications of the different open access models. CHARISMA takes them as a starting point and defines the different operational alternatives that will be developed as part of CHARISMA.

3.1.1. Open access overview

Typically, telecommunications networks have three main layers, each layer with a separated functionality, and each comprising differentiated devices carrying out specific tasks for each of them and with different business logics and revenue models. Those are:

- Network Infrastructure.

- Data bitstream transport.
- End customer service and content delivery.

Although each of these three layers can be segmented into sub-layers (e.g., for instance, civil infrastructure, ducts and fibre optic cables, optical distribution frames, etc. as network infrastructure subsections), no further segmentation will be done from the CHARISMA network slicing point of view. Figure 3-1 presents the typical telecom value chain and describes the different operational models.

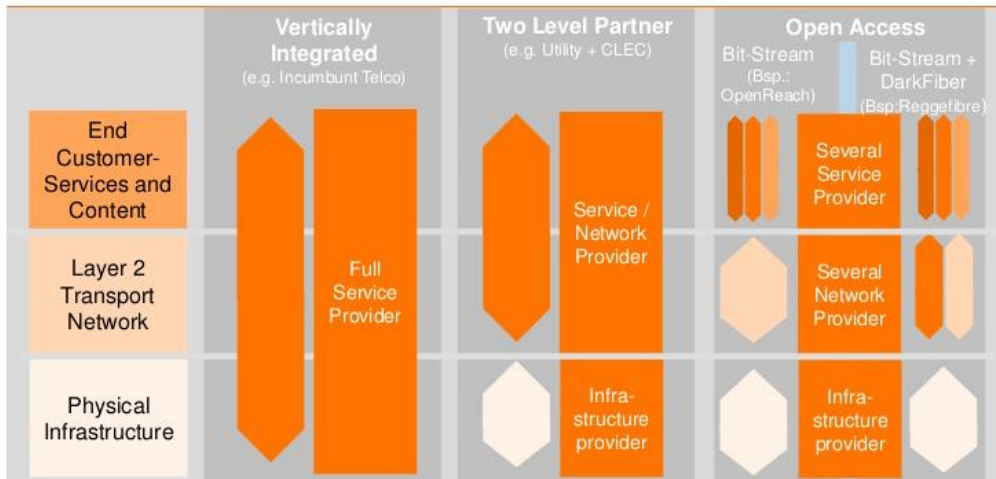


Figure 3-1: Network value chain operation models.

The three models, shown in Figure 3-1, represent different implementations of the way the Network and Service Providers manage and operate the infrastructure in order to deliver services over their network or over that of an infrastructure provider with whom they have a contractual agreement.

CHARISMA’s vision is that the structural separation shown in the right part of the graph is key to achieve the following:

- Reduce CAPEX and OPEX of next generation networks.
- Optimize costs and investments, allowing a faster network implementation with reduced investment requirements.
- Foster competition, which will have a positive impact on final users, in that they will see a reduction in the cost of their communications services.

There is an alternative to the open access model, which is called the dark fibre approach model, where ISPs rent fibres and cables and deploy their own equipment. This alternative is not seen as the optimum, because in reality, only the cost of the civil infrastructure is shared and ultimately, an overlay cost of network devices deployment is inherited, which means higher CAPEX and increased OPEX to operate, and manage the network, and with increased power consumption as well.

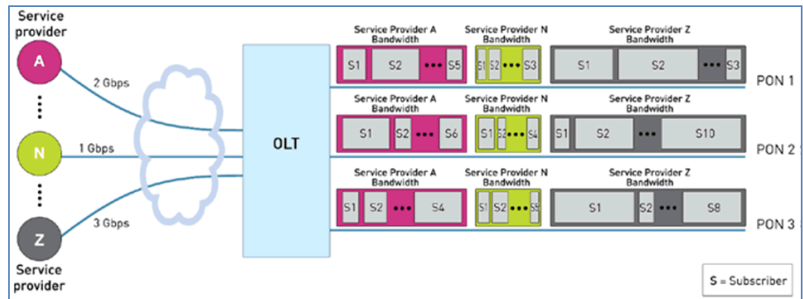


Figure 3-2 Bitstream services in Open Access Networks

CHARISMA’s proposal for the optimum operational model would be to offer end-to-end bitstream services, as indicated in Figure 3-2. The network segmentation at the bitstream level optimizes active equipment utilization and power consumption, CAPEX and OPEX, and at the same time offers universal access to any ISP offering services to the final customers interconnected by the CHARISMA network.

3.1.2. CHARISMA vision for open access

Typically, a classical open access approach would offer bitstream services with guaranteed bandwidth and no specifications about jitter, latency and OAM parameters. In addition, open access networks do not normally deal with convergent services for radio backhauling.

The innovative aspects that CHARISMA proposes from the open access point of view are:

- Advanced OAM for ISPs L2 end-to-end (E2E) control and management;
- Hardware abstraction layer for multi-vendor compatibility over a converged wireless/wireline network.

These objectives will allow CHARISMA to implement an innovative open access approach, mainly focused on reducing the latency, providing an optimized network and a reduced power consumption.

3.1.3. CHARISMA open access requirements

Converged open access

CHARISMA proposes a converged system to deliver 5G services, offering a common infrastructure to all 5G ISPs in order to optimize deployment costs.

Open Access Networks seamlessly provide the service provisioning to network subscribers independently from the ISP they belong to. Also, this requirement needs to be compatible with legacy technologies so that CHARISMA becomes a unique platform for the delivery of final radio services.

Inter-operability

CHARISMA is an interoperable system that can be shared among different operators for the delivery of connectivity and applications using a common and shared radio/fixed access infrastructure.

Simplified operation, administration and management (OAM)

CHARISMA simplifies OAM and allows ISPs to control OAM parameters proposing a centralised control, management and service orchestration platform that guarantees the SLAs and service synchronization, together with provision and fault management from a unique system that is able to control and manage all the different slices and devices of the network. There is an aspect that will focus on the development of this centralised, simplified system dealing with different parts of the converged network, which is the resiliency over this network that it is out of the scope of CHARISMA.

Modular structure to support a multi-vendor scenario

The implementation of the management and control plane will be open in order to support equipment from different vendors. Interoperability in the access network is nowadays one of the major barriers that the infrastructure owners face when deploying new networks. The software implementation of CHARISMA will be modular and object-oriented in order to guarantee that the addition of a new device from a new vendor will only require the development of specific hardware driver for that piece of equipment. The SDN/NFV techniques that are to be implemented by CHARISMA control, management and orchestration plane will help on the integration of new modules and new vendor devices into the architecture.

3.1.4. CHARISMA open access services

CHARISMA open access services for network operators will require a set of technical requirements that are explained below. This section has been segmented taking into consideration the different operation layers that constitute the telecommunications networks:

- Dark fibre requirements.
- Wavelength services.
- Metro Ethernet services.
- OAM and sync services.

These requirements are discussed in greater depth below.

Dark fibre requirements

Even though CHARISMA does not encourage dark fibre services as such, given the fact that these services develop an overlay of networks sharing the same physical infrastructure, we aim to explain the requirements from a high-level point of view. The dark fibre services provide a point-to-point connection from the CO to the final user location by splicing the fibre cables deployed to assure signal continuity.

A correct and clear identification of the fibre path is essential in order to manage the outside network efficiently. Outside meaning external network, the part of the telecommunications network that is not sitting in a building, telephony exchange or similar. An inventory of the physical connections will need to be implemented to keep track of the external network connectivity.

As far as the fibre specifications are concerned, typical fibre specifications for the external network are ITU-T G652 single mode fibre and ITU-T G657a bend-insensitive, single-mode optical fibre, is proposed for the final-drop segment in order to allow bending of cables when inside the end-user premises.

Wavelength services

Wavelength services provide guaranteed end-to-end isolated capacity and are offered by assigning a dedicated wavelength from the CO to the final customer. In order to prevent the access of the SP to the physical infrastructure, demarcation points will be implemented at the CO and end-user premises. The SP will inject the services into the network by means of an uncoloured interface, where CHARISMA equipment will transform it to WDM over the CHARISMA infrastructure.

This is a layer-1 transport service. This strategy implies that even if a transceiver from the SP fails, the entire WDM segment is prevented from failing since CHARISMA is in complete charge of the optical layer.

This requirement forces the system to define the capacity of the wavelength service depending on the different devices required according to the deployed data-rates (10G/40G etc.).

At the moment of writing this document, it is not clear whether CHARISMA specifications will offer a L1 service and segmentation at the wavelength level, here we provide the information as a reference only. This will be defined at a later stage within WP1 of CHARISMA.

Metro Ethernet services

One of the objectives of CHARISMA is to provide the platform to be able to offer convergent bitstream services for 5G radio access. Thus, layer-2 segmentation and Metro Ethernet functionalities will be implemented in order to have the ability to aggregate several customers belonging to the same bitstream service assigned to a specific RSP (retail service provider).

Depending on the SP needs, traffic will be delivered at the CO aggregated by service, while differentiating the final customer (or group of customers).

The network operator may use an Ethernet access network, such as CHARISMA, to reach the end-user through a network infrastructure and to connect to the peering operator service endpoint. The SP must then orchestrate the services between the end-user service endpoints at each subscriber site. The end-to-end lifecycle service orchestration between operator networks is part of the CHARISMA WP3 objectives at L2 (metro-Ethernet level). This functionality will be performed by the SP with the appropriate business relationship with the end-user. The SP could be one of the network operators or a third party who does not own nor operates the physical network infrastructure. SPs can also orchestrate and bundle network as a service with other network-based services, e.g., firewall, intrusion prevention, and cloud services. All these services will be explored and implemented by CHARISMA at a later stage.

OAM and sync services

OAM and sync services will be implemented in CHARISMA encouraging the usage of standard protocols, mechanisms and procedures for monitoring. It will investigate the status of Virtual Connections (VCs), Operator Virtual Connections (OVCs), and External Interfaces across a defined OAM Domain, where that domain will be a part of the network driven from the use cases that are being defined by WP1 at the moment of writing this document. OAM uses the protocols [IEEE 802.1ag] and [ITU-T Y.1731] in order to determine the status of and troubleshoot connectivity across a particular domain, one of the steps forward that CHARISMA will develop is to offer to the SP the ability to access OAM parameters. This will enable assured end-to-end maintenance on a service level for each specific bitstream service.

Further to this, provided the 5G convergence between radio/fixed services, sync services are a key requirement in order to easily recover the clock at the base station. SynE and ITU-T 1588 sync services will be supported over the CHARISMA network. The parameters that are expected to be covered by CHARISMA and hence stated here as requirements include: bandwidth management, jitter and QoS.

3.1.5. Unified Control & Management

The control, management and orchestration plane in CHARISMA will provide the required functions to allow network providers to configure network parameters and to offer tailored infrastructure slices to SPs. Additionally, it will also implement the processes that are required for all the OAM of the network services deployed on top of the CO, IRRH and Virtual CPE, including monitoring and troubleshooting. This is a very important requirement when CHARISMA aspires to be a platform for SPs that share a common infrastructure managed from a centralised system.

The functional requirements that constrain the specification of the technical functionalities to be supported by the control and management platform have to be identified prior to the design of its modules. It is important to achieve a comprehensive and complete requirement specification at the initial phase of the project to ensure that the design supports the different business models and technical details of the IRRH components implemented in WP2.

The requirements of the control and management system from an open access point of view will target the provisioning of the network services enabled by the IRRH. These requirements are the following.

- Isolation and Virtualisation support: A key objective of CHARISMA is to explore and implement virtualization mechanisms to support multi-tenancy for service providers and guarantee isolation between virtualized resources.
- Network and management service: Each participant of the network service to be provided requires control and management mechanisms for applying the policies and functionalities associated with its own role. The CHARISMA control and management platform will offer a platform with tailored control and management functions depending on the infrastructure provider or service provider point of view.
- Monitoring and recovery: Monitoring is an important requirement that the control and management system has to support to enable the detection of alarms and the recovery from failures for every SP.
- Heterogeneity support: The resource abstraction implemented for CHARISMA will allow the homogeneous control of heterogeneous resources that have similar characteristics, which will be the base for supporting virtualization mechanisms for each SP.
- Pre-configured or on-demand provisioning: The on-demand provisioning and configuration of the IRRH components are basic for a dynamic and efficient usage of the resources.

- **Scalability:** The design of the control and management plane has to be developed to guarantee high performance, even when several virtualization instances have to be controlled, allowing multiple service providers to seamlessly coexist in a common physical infrastructure.
- **Dynamic re-configurability:** The platform has to enable the possibility of reconfiguring virtualized and provisioned resources while minimizing the effect on already operative services.
- **AAI (authentication, authorization infrastructure) mechanisms support:** Security is an important requirement in the CHARISMA infrastructure, which involves the support of authentication and authorization mechanisms for enabling or disabling the access to certain slices of the network resources for service providers.

3.2. Security

The CHARISMA security idea is based on the concepts of routing security, v-security, and trust sharing.

Routing Security

One of the main threats in networks today is a certain portion of insecurity that stems from the fact that routing paths are not fixed and in practice are not predictable. The routing protocols, such as BGP, are based, among others, on cost metrics. Just by offering a very low cost route, packets can be “legally” deviated to paths where data could potentially be intercepted.

A routing concept with deterministic paths could be an incentive for increased security, and this can be achieved by source routing, where the source determines the complete path.

In the case of hierarchical routing (see Figure 3-3, taken from the CHARISMA Technical Annex) the routing path stays as local as possible. For example, a packet sent to the neighbour just goes up to the first hop. Packets sent to destinations within the same country stay automatically within the country network.

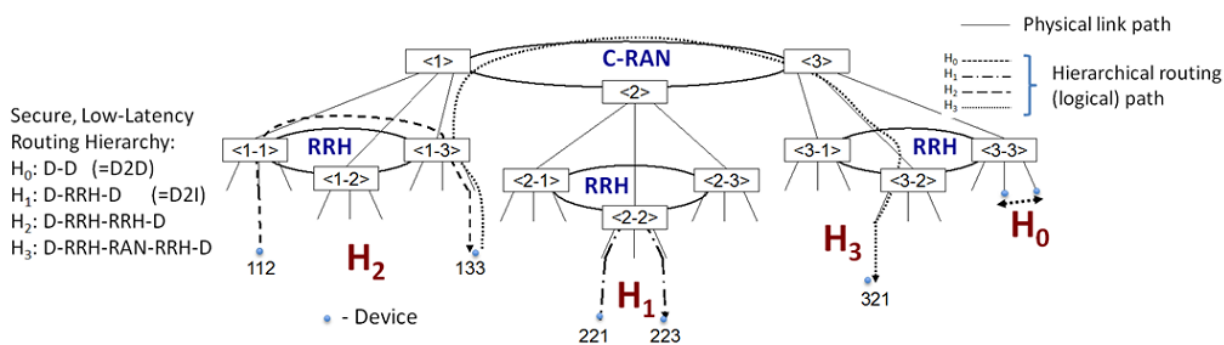


Figure 3-3: Hierarchical routing network

The hierarchical routing allows source addresses to be checked by the network. At the network entry and in the upstream direction the source address of the packet is checked and in case of non-conformity packets would be discarded. Faking a false source address will not be possible. A receiver can rely on the correctness of the source address and can use this information as an additional criterion for the firewall. For example,

packets coming from a remote office could be accepted without further check (whitelisting). Conversely, packets from unwanted sites could be discarded immediately (blacklisting).

However, the routing security is not perfect, since it only assures layer 3 security. Packets could be intercepted at layers 1 or 2, e.g. by sniffing fibre optic links, or at higher layers, e.g. by placing Trojans in the protocol stack of the endpoints. In the routing layer itself the routers could be “hacked”. This can be impeded by several means:

- Place routers in secure places: locked cabinets with access control, central office building with secure entry etc.;
- Harden router against false configuration: panel configuration only by authorized personnel, e.g. with RFID identification;
- Disable remote configuration by design: default routing path implemented in hardwired logic. Use of open routing code;
- The integrity of all soft-changeable configuration and code is verified in iterative timeslots. The integrity of soft- and hardware functions is checked mutually.

Note that the hard-coded, permanent routing mode is complemented by dynamic flow based forwarding, initiated by SDN methods. For this purpose, the routers contain a configurable flow table. The setup process to establish dynamic flow forwarding may benefit from the constant and reliable routing mode.

Note that routing security is orthogonal to existing security means, such as end-to-end encryption (VPN).

Routing security increases the security of public key infrastructure (PKI). When a public key is sent by email, it must be assured that the email is not a fake. With the hierarchical verified routing concept, the public key can be sent directly from peer to peer.

V-Security

The v-Security concept bases the abstraction of common security functions like Firewalls, DPI, IDS implemented via functional add-ins through NFV technology in which CHARISMA WP3 has based the design of its architecture. This virtual security function can be logically distributed over the network to identify and prevent possible intrusions. The virtual abstraction layer separates several service providers as an additional security aspect. The SPs will not have access to the hardware, such that all actions are virtualised and controlled by the network infrastructure owner. The security management system of the Internet infrastructure provider prevents unauthorized actions of the services providers.

Trust sharing

CHARISMA will provide a comprehensive authorisation and authentication concept based on Pretty Good Privacy (PGP) and web of trust (WOT) techniques. Every user, every virtual and hardware component has to be authenticated. This gives the end-user the possibility to trust used APs and prevent fake AP attacks. CHARISMA also plans to explore the MACsec [41] for authentication and encryption for MAC layer security.

3.3. Low Latency

Another key driver of the CHARISMA architecture is the demand for a very low-latency networking. Indeed, one of the 5G key performance indicators (KPIs) is to target a 1 ms latency time. However, achieving this 1ms target represents a technical challenge, e.g. assuming an access network incorporating a long-reach passive optical network (LR-PON) with a 100-km distance between the OLT and ONT, implies a minimum round-trip delay for the light signal itself of 1ms alone, independent of any additional processing and routing times. Hence, within the CHARISMA project we are addressing the low-latency aspect in a variety of ways. The first one is our adoption of a hierarchical architecture approach, where data is always routed, where possible, at the lowest common aggregation point or flexibility node. This means that for device-to-device (D2D) communications, data is routed directly between the devices, whereas routing at the next lowest CHARISMA aggregation level (CAL), e.g. at CAL0, means the data is routed between devices via the local (e.g. home or access) gateway. See Figure 3-4 for reference and the CAL (CHARISMA aggregation level) visual explanation. For example, devices within a micro-cell, routing is via the CAL1 level; within a macro-cell, it is via the CAL2 level, e.g. at the macro base station or active remote node; and finally for non-local routing, this is performed at the CAL3 level, or at the central office. In addition, positioning of local caching at the various CAL levels, allows data to be accessed ever closer to the end-user, so as to reduce the latency between a data request and the reception of the data.

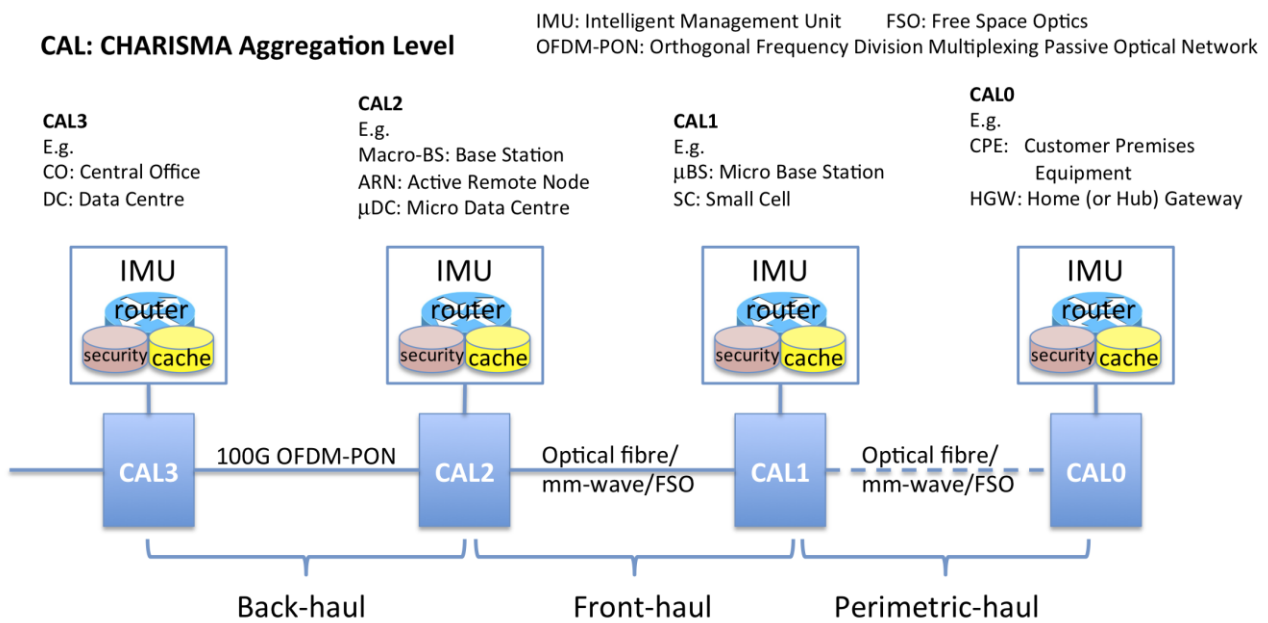


Figure 3-4: CHARISMA Aggregation Levels Architecture

Finally, through the use of the accelerated network interface card (NIC), and the TrustNode hierarchical router, these will combine to provide a very low-latency 5G networking architecture. The management of network and compute resources (e.g. caching, routing, security, and processing) at the different CALs or network flexibility points, requires an appropriate CHARISMA control and management plane which is being developed within WP3 of the project. The following section provides a high-level explanation of the

implementation of the Content Caching base solution and traffic handling capabilities that will provide a compound 5G solution for the CHARISMA system.

3.3.1. Low latency in content caching solution

To meet the growing data demands of mobile users, commercial LTE networks have been deployed to significantly increase the bandwidth and reduce the latency in the backhaul network. However, the service latency is highly depending on the distance from the DC to the mobile network operator connecting point to the Internet. This service latency cannot satisfy the QoS of current applications, especially for latency-sensitive applications like audio and video services.

Mobile network caching is a cost-effective solution to improve the service latency and reduce the mobile backhaul traffic by replicating popular and frequently demanded content in the IP-based 3G/4G network elements closer to mobile users. Mobile network caching is getting a lot of attraction from researchers and companies. New infrastructures, cache framework and cache algorithms in mobile network have been studied [32], [33], [34], [35] and [36] . Caching solutions for 3G or LTE networks have also been developed by companies like Allot Communications, PeerApp, Altobridge, etc.

In this project, a caching architecture will be converged in the 5G network to achieve the following goals: to improve the service latency, reduce traffic exchanges and operational costs while improving QoE for end users. A collaborative hierarchical caching system is expected to provide such benefits.

Within the context of CHARISMA WP3, an SDN-based caching controller will be developed to further ameliorate the caching efficiency by optimising the data flow and the content caching. SDN and NFV are architectural enablers for content caching solutions in CHARISMA. SDN allows decoupling of the control and the data planes by moving the control intelligence to a centralized function, called the controller. In turn, the controller has a vision of the whole network and can seamlessly manage the traffic. In practice, the SDN controller can be used to manage/control content replicas by keeping track of the location and availability of content in distributed locations. Moreover, the SDN controller can also intelligently relocate content and reroute traffic in the converged network according to users' needs, thus avoiding undesired effects and inefficiencies, such as the thrombosing effect. Moreover, NFV can further help in the deployment of a unified service delivery. By instantiating network functions on demand, the same equipment can be configured to support typical fixed network or mobile network functions according to the needs, maximizing the resource sharing between networks.

4. Control, Management and Orchestration Plane High-level Design

4.1. Overview of the CMO plane of CHARISMA

CHARISMA aims to virtualize the access network so that its control and management becomes homogeneous, flexible and cost-efficient. A second goal, aided by virtualization, is to enable the sharing of the access network among multiple network operators as has already been explained in section 3. However, these goals should be achieved without compromising operational and application security and application latency. In fact, it is desirable to strengthen security and decrease latency, when possible. As mentioned earlier, the CMO plane, described in this section, not only encompasses the v-security management but also enables control, management and orchestration of the physical and virtual resources of the CHARISMA architecture from end-to-end network service point of view.

In the context of CHARISMA, the access network is defined in WP1 as shown in Figure 4-1 (which is a more abstracted version of Figure 3-4.) It has 4 levels of aggregation, or CALs (CHARISMA Aggregation Level), from the Customer Premises Equipment (CPE) at CAL0 (it can also be a User Equipment (UE)) to the Optical Line Termination (OLT) at CAL3. Intermediate CALs host the (micro-, macro-, etc.) Base Stations (BS). The links between CALs can be either wired or wireless. The CAL0-CAL1 link in particular can be both wired (e.g., FTTx) and wireless (e.g., cellular) with vertical handover between the two. This access network model stems from current practice and is sufficiently generic to stay applicable for 5G access networks.

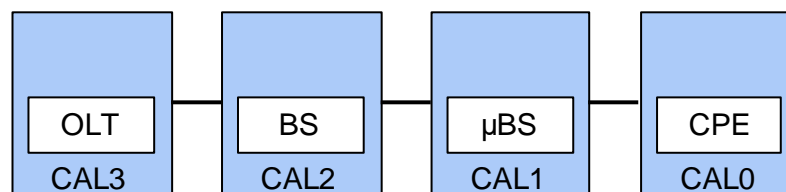


Figure 4-1: The CHARISMA model for the access network

Having defined the access network, and before proceeding with the description of the high-level design of the control, management and orchestration plane (CMO plane), it is beneficial to define the terms “control”, “management” and “orchestration”:

- *Control* adjusts the behaviour of a system in a changing environment in order to fulfil a given policy. Control is thus focused on a short time scale and is typically realized in a control loop. An example is when an unknown type of packet arrives at an SDN Switch, the packet is forwarded to the SDN Controller, which calculates a new flow rule, based on a pre-set policy, and then sets this new rule to the SDN Switch for future reference.

- *Management* governs the long-term behaviour of a system by setting operating parameters at deployment and then at significant changes of the environment that will last for the foreseeable future. Management includes the collection of operating data (analytics) in order to support long-term decision-making. An example of management is when setting the IP and port of the SDN Controller to an SDN Switch during deployment or topology updates.
- *Orchestration* is responsible for on-boarding of new network services (NS) and virtual network function (VNF) packages; NS lifecycle management; global resource management; validation and authorization of network functions virtualization infrastructure (NFVI) resource requests.

The high-level design of the CHARISMA control and management plane is shown in Figure 4-2. It closely follows the ETSI NFV architecture [37] as the latter is a standard that has been developed internationally over several years and is geared towards virtualization and multi-tenancy. Moreover, the ETSI NFV architecture comes with background work on security [38] and performance [39].

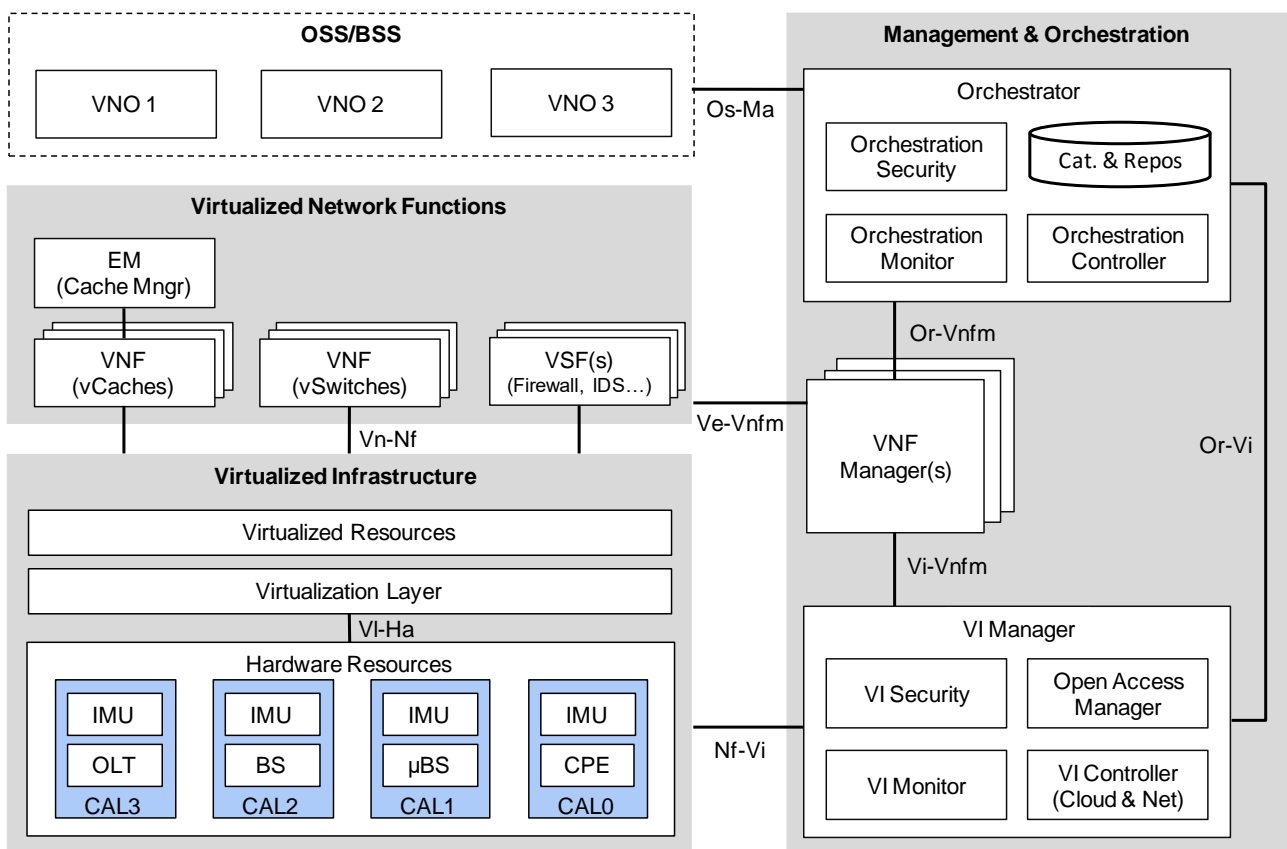


Figure 4-2: The high-level CHARISMA control and management plane

The architecture consists of four groups of components: Virtualized Infrastructure (VI), Virtualized Network Functions (VNFs), Management and Orchestration (MANO), and Operations and Business Support Systems (OSS/BSS). CHARISMA will focus on the first three groups in an effort to enable multiple Virtual Network Operations (VNOs) at the OSS/BSS who will be sharing the hardware resources at the VI.

The Virtualized Infrastructure group virtualizes the hardware resources (computing, storage, and network) via e.g., a hypervisor at the Virtualization Layer, which pools the resources and exposes them for consumption by VNFs. The hardware resources constitute the CHARISMA access network, with the notable addition of an Intelligent Management Unit (IMU) at each CAL. The IMU models computing and storage resources that are either spare within access network equipment, e.g., BSs, or introduced with COTS hardware, e.g., servers.

The Virtualized Network Functions group comprises software components that implement network functions destined to run on the VI (and finally on the IMUs). CHARISMA will work specifically on implementing VNFs for caching, switching, and security. However, any other network function, e.g., CDN, would be able to run on the VI.

The Management and Orchestration group includes components for the combination of VNFs into graphs implementing network services, the lifecycle management of VNFs, the coordination of allocating VNFs to virtualized resources, the homogenized control and management of the hardware resources, and the slicing of resources for supporting multi-tenancy. MANO operates under the policy set by the owner of the hardware infrastructure and communicates with the OSS/BSS of VNO to report status and possibly to receive requirements.

The following sections describe the components, its function and interfaces of the CHARISMA control and management plane in more detail.

4.2. Building blocks and functional design of the CMO plane

As presented in Figure 4-2, the CHARISMA management plane has three logical differentiated groups: Management and Orchestration, Virtualized Network Functions and Virtualized Infrastructure. In the following sections, we present the building blocks of each group and describe their main functionalities.

4.2.1. Management and Orchestration

The Management and Orchestration (MANO) group has been divided in three subgroups, the Orchestrator, the VNF Manager and the Virtualised Infrastructure Manager (VIM).

4.2.1.1. Orchestrator

The establishment of an end-to-end CHARISMA service requires the knowledge of the available resources, the planning of the service and the interaction with the VIM managing the infrastructure assets, which will be involved. All these operations need to be carried out by a higher-level management entity, which jointly orchestrates the underlying infrastructure. This is the role of the Orchestrator, which will incorporate functional modules dealing with orchestration resources and network services as well as VNF management.

The main function of the Orchestrator is to manage the virtualized Network Services (NS) lifecycle procedures. Since the NSs are composed by Virtual Network Functions (VNFs) and/or Virtual Security Functions (VSFs), the Orchestrator is able to decompose each NS into the constituent VNFs. Nevertheless, although the Orchestrator has the knowledge of the VNFs/VSFs that compose the NS, it delegates their

lifecycle management to a dedicated entity, the VNF Manager (VNFM). Furthermore, besides orchestrating the virtualized service level operations, therefore abstracting the service specificities from the business/operational level, the Orchestrator also manages the virtualized infrastructure resource level operations. Hence, it coordinates the resource allocation to specific NSs and VNFs according to the availability of the virtualized infrastructures, also known as DCs.

In order to fulfil its role, the Orchestrator should maintain internal catalogues and repositories containing information about underlying resources, available and established Network Services, available VNFs and deployed instances of both, as well as infrastructure resources. At the same time, a repository for hosting the VNF images and associated metadata from where the VNF images are retrieved for deployment as instances into the infrastructure will be implemented. In particular, the following repositories and catalogues required for the orchestration have been specified:

- The **Infrastructure Resources Repository** that holds information about available/reserved/allocated NFVI resources as abstracted by the Virtualized Infrastructure Management (VIM) across the infrastructure of the operator, thus, providing adequate information useful for resources reservation, allocation and monitoring purposes.
- The **VNF/VSF Catalogue** represents the repository of all available on-boarded VNFs and VSFs, supporting the creation and management of the VNF/VSF Descriptor (VNFD/VNSD). The information contained in the VNFD is defined by ETSI. It is important to clarify that the VNF/VSF Catalogue contains a list of the available VNFs/VSFs, which can be included in an NS, not the deployed VNFs themselves. In a similar way, a similar catalogue can be used –if necessary- in order to contain specific Network Services (i.e. combination of VNFs/PNFs) that are described by the respective NS Descriptors.
- **NS Catalogue** that represents the repository of all the on-boarded NSs in order to support the NS lifecycle management. The NS Catalogue includes descriptions of the network services (NS Descriptors (NSDs)), along with other required information such as SLAs, deployment flavours, references to the virtual links (Virtual Link Descriptors) and the constituent VNFs/VSFs (VNF Forwarding Graph Descriptors).
- The **VNF/VSF Instances Repository** that contains information of all service instances which have been actually deployed. The repository is frequently updated, to reflect the status and the lifecycle of the deployed virtualised services.
- The **VNF/VSF Image Store** that contains the VNF/VSF images and their associated metadata.

Orchestration Controller

The Orchestration Controller is the core decision-making component, meaning the “kernel” of the Orchestrator. The Orchestration Controller instantiates VNFs, which are part of network services using the VNF templates in the corresponding catalogues and manages the whole network service lifecycle. For this purpose, it communicates with the VNF Manager and the Infrastructure Resources for the control of VNF instances and the (re-)allocation of virtualised resources. This task includes the control of the network assets, for virtual network establishment and QoS provision.

Orchestration Security

The Orchestration Security entity is responsible for the policy management and the decision making related to security. It receives security rules set by VNOs, based on events received from the monitoring component and information received from deployed network services, it can detect possible attacks and security threatening conditions. Depending on the security policies set by the VNOs, the security entity can take decisions on counter-measures that are appropriate to address a particular threat. Examples of such decisions are the configuration, termination or migration of an already deployed service and the deployment of security related services (e.g. a firewall) that respond to the specific attack.

Orchestration Monitor

The Monitoring component within the Orchestration layer comprises of functionality that performs metrics acquisition from the VIM layer. Based on these metrics the Orchestrator can derive decisions and perform changes to the network services that are already deployed (for example, the scaling of a particular VNF if monitoring data received indicate that is required) or even -if required- instantiate and deploy new services.

4.2.1.2. VNF Manager

The VNF Manager (VNFM) is responsible for the lifecycle management of VNF instances and the FCAPS management of the internal VNF operations. Each VNF instance is assumed to have an associated VNF Manager. A VNF manager may be assigned the management of a single VNF instance, or the management of multiple VNF instances of the same type or of different types. Operations carried out by the VNF Manager are VNF instantiation, configuration, scaling and termination. The VNFM is also responsible for the overall coordination and adaptation role for configuration and event reporting between NFVI and the traditional vendor specific Element or Network Management System (E/NMS) and provides an interface to it.

4.2.1.3. Virtualized Infrastructure Manager (VIM)

Open Access Manager

The Open Access Manager performs virtualization at the control and management plane level. This entity is responsible for creating isolated slices over the CHARISMA physical infrastructure. These virtual networks are offered towards the different VNOs. The actual service is decoupled from the infrastructure where it is provisioned, following the Open Access Network model. Another responsibility of the Open Access Manager block is to provide isolation between different virtual networks. Using isolation techniques, interference between services provisioned by different service providers can be avoided.

VI Controller

The VI Controller consists of two separate controllers, the network controller and the cloud controller.

The network controller is a logically centralised entity which is responsible for a set of tasks, including the extraction and maintenance of a global view of the network topology and state, as well as the instantiation of forwarding logic appropriate to a given application scenario. In practice, the controller manages connections to all substrate switches using a southbound protocol such as OpenFlow, and installs, modifies

and deletes forwarding entries into the forwarding tables of the connected switches by using protocol specific control messages.

The cloud controller provides the central management system for cloud deployments. It includes the software components required for building and managing cloud computing platforms for public and private clouds. The cloud controller supports the deployment and lifecycle management in cooperation with the Orchestration of the VNFs deployed on Virtual Machines (VMs) within the cloud infrastructure. It also provides a common virtualisation layer across different platforms making the VNFs independent of the actual underlying physical infrastructure. It achieves so by managing the virtual compute environment, including VMs, virtual switches and top-of-rack switches in DCs.

VI Monitor

The VI Monitor component is mediator/processing entity at the VIM level responsible for collecting, consolidating, processing metrics and communicating them to the Orchestrator. An intelligent aggregation procedure is required to achieve the grouping/aggregation of various metrics from various parts of the infrastructure as well as alarms, and the dynamic identification of information that is of actual value to the Orchestrator. The VI Monitor aggregates metrics by polling the cloud and network controllers and by receiving additional information from monitoring agents running in the VNFs, consolidates these metrics, produces events/alarms if appropriate and communicates them to the Orchestrator.

VI Security

The VI Security entity is responsible for the common authentication, authorization and accounting (AAA) management functions. Identity management and access management are implemented within this component, such as the access to the physical equipment and the management of access and operations over the different virtual networks created.

4.2.2. Virtualised Network Functions

The Virtualised Network Functions (VNFs) group consists of software components that will be running on top of the CHARISMA virtualized infrastructure. As we have already mentioned, VNFs implement common network functions such as gateways, proxies, firewalls and transcoders, traditionally carried out by specialised hardware devices and are deployed on top of commodity IT infrastructure. Our focus within the project will be the development of VNFs that are implementing caching, switching and security functions.

A VNF is composed by one or more VNF Components (VNFCs) that are interconnected through Virtual Network Links. The VNF details (e.g. deployment rules, scaling policies, and performance metrics) are described in the VNF Descriptor. A VNF Descriptor (VNFD) is a deployment template, which describes a VNF in terms of deployment and operational behaviour requirements. The VNFD also contains connectivity, interface and KPIs requirements that can be used by the Orchestrator functional blocks to establish appropriate Virtual Links between VNFC instances, or between a VNF instance and the endpoint interface to other Network Functions.

4.2.3. Virtualized Infrastructure

The Virtualised Infrastructure group includes the physical and virtual resources (commodity servers, VMs, storage systems, switches, routers etc.) and the virtualization layer required in order to be able to accommodate Virtual Network Functions (VNFs) as workloads.

4.2.3.1. *Hardware Resources*

The hardware resources include the devices comprising the CHARISMA access network and DC elements such as COTS servers that are virtualization-capable to support NFV.

The CHARISMA access network consists of Optical Line Termination (OLT), Base Station (BS), micro-Base Station (μ BS), Customer Premises Equipment (CPE), possibly User Equipment (UE) and wired or wireless links that interconnect these components. The devices within the CHARISMA access network may or may not have virtualization capabilities. In case of virtualised components, they can provide resources for accommodating IMUs.

Commodity servers are added in the architecture to provide IMUs when the resources within the CHARISMA access network devices are not enough. Servers incorporate compute, memory and networking resources. The compute resources consist of computing and storage equipment (standard high-volume servers with or without specialized hardware accelerations and storage infrastructure). For standard-scale data centre implementations, servers based on the x86 architecture are a common choice. The network resources include all networking elements, such as SDN and non-SDN switches and routers that interconnect all the compute/storage infrastructure of the compute domain. Within the hardware resources we can include specialized network interfaces cards with features for hardware-assisted virtualization, such as DPDK (Data Plane Development Kit) support and SR-IOV (Single-Root I/O Virtualisation) that seems promising for the enhancement of VNF performance.

4.2.3.2. *Virtualised Resources*

Virtualization is the ability to simulate a hardware device, such as a server, storage device, memory or network resource, in software terms. All of the functionality is separated from the hardware and simulated as a “virtual instance” with the ability to operate similar to the traditional hardware solution. The hardware resources within CHARISMA architecture that are virtualisation-capable are abstracted to virtualised resources to support IMUs. These virtualised resources are eventually used by the VNFs. Virtualised resources in CHARISMA include virtual compute, virtual memory and virtual networking resources.

4.2.3.3. *Virtualization Layer*

The Virtualization Layer, also commonly known as hypervisor, is responsible for the abstraction of the physical compute and storage resources (possibly aggregated across multiple physical elements) and their assignment/allocation to VNFs. The hypervisor domain mediates the resources of the computer domain to the virtual machines of the software appliances. Hypervisors as developed for public and enterprise cloud requirements place great value on the abstraction they provide from the actual hardware such that they can achieve very high levels of portability of virtual machines. In essence, the hypervisor can emulate every

piece of the hardware platform even in some cases, completely emulating a CPU instruction set such that the VM believes it is running on a completely different CPU architecture from the actual CPU on which it is running. Such emulation, however, has a significant performance cost. The number of actual CPU cycles needed to emulate virtual CPU cycle can be large. The hypervisor commonly exposes a northbound interface for the interaction with the management layer.

4.3. Management interfaces

The interfaces between the building blocks described previously in sections 4.1 and 4.2, shown in Figure 4-2 are described in detail in this section.

4.3.1.1. *Virtualization Layer - Hardware Resources - (VI-Ha)*

This reference point interfaces the virtualization layer to hardware resources to create an execution environment for VNFs, and collect relevant hardware resource state information for managing the VNFs without being dependent on any hardware platform.

4.3.1.2. *VNF - NFV Infrastructure (Vn-Nf)*

This reference point represents the execution environment provided by the NFVI to the VNF. It does not assume any specific control protocol. It is in the scope of NFV in order to guarantee hardware independent lifecycle, performance and portability requirements of the VNF.

4.3.1.3. *Orchestrator - VNF Manager (Or-Vnfm)*

This reference point is used for:

- Resource related requests, e.g. authorization, validation, reservation, allocation, by the VNF Manager(s).
- Sending configuration information to the VNF manager, so that the VNF can be configured appropriately to function within the VNF Forwarding Graph in the network service.
- Collecting state information of the VNF necessary for network service lifecycle management.

4.3.1.4. *Virtualized Infrastructure Manager - VNF Manager (Vi-Vnfm)*

This reference point is used for:

- Resource allocation requests by the VNF Manager.
- Virtualized hardware resource configuration and state information (e.g. events) exchange.

4.3.1.5. *Orchestrator - Virtualized Infrastructure Manager (Or-Vi)*

This reference point is used for:

- Resource reservation and/or allocation requests by the Orchestrator
- Virtualized hardware resource configuration and state information (e.g. events) exchange.

4.3.1.6. **NFVI - Virtualized Infrastructure Manager (Nf-Vi)**

This reference point is used for:

- Specific assignment of virtualized resources in response to resource allocation requests.
- Forwarding of virtualized resources state information.
- Hardware resource configuration and state information (e.g. events) exchange.

4.3.1.7. **OSS/BSS - NFV Management and Orchestration (Os-Ma)**

This reference point is used for:

- Requests for network service lifecycle management.
- Requests for VNF lifecycle management.
- Forwarding of NFV related state information.
- Policy management exchanges.
- Data analytics exchanges.
- Forwarding of NFV related accounting and usage records.
- NFVI capacity and inventory information exchanges.

4.3.1.8. **VNF/EMS - VNF Manager (Ve-Vnfm)**

This reference point is used for:

- Requests for VNF lifecycle management.
- Exchanging configuration information.
- Exchanging state information necessary for network service lifecycle management.

4.3.1.9. **Service, VNF and Infrastructure Description - NFV Management and Orchestration (Se-Ma)**

This reference point is used for retrieving information regarding the VNF deployment template, VNF Forwarding Graph, service-related information, and NFV infrastructure information models. The information provided is used by NFV management and orchestration.

4.3.1.10. **OpenStack Mapping**

In the graph below, the OpenStack by OPNFV, which is mapped in according to our architecture, is shown. All modules are the APIs for Network, Storage, Monitoring, Authentication, etc. In **black** are the Modules (APIs) responsible for Functions, which are in **red**.

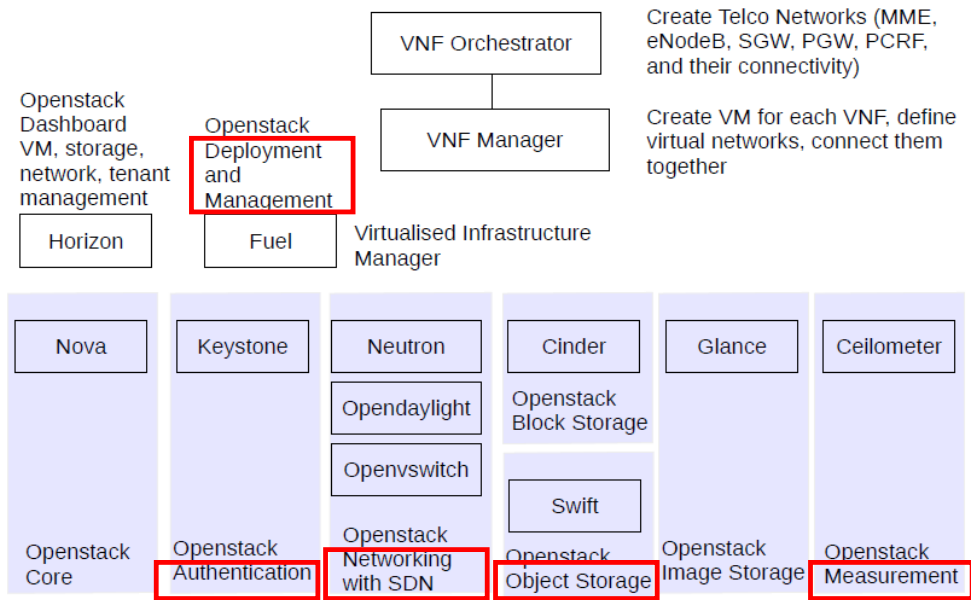


Figure 4-3: OpenStack: Functions description mapping

4.4. NFV and the Network

Nowadays there is an increasing interest in several activities that aim to implement the NFV environment into telecommunications networks. OPNFV (ETSI aligned from an architectural point of view) is a cloud project whose main objective is to produce such a cloud platform that is suitable for running virtual Telco applications on it. Many companies are joining OPNFV, mainly because it is being supported and pushed by big Telecom companies like Ericsson, Huawei, Orange, and important IT companies like Redhat, Intel and others. OPNFV has many sub projects, each project dealing with a specific part of the telecom environment, based on the concepts defined in ETSI for NFV (Network Function Virtualization) [40].

The objective of OPNFV is to create a standard approach (API), and standard software infrastructure for how developers can innovate and create solutions around NFV. With a standardized approach that takes advantage of much the innovation coming from a number of industries including the cloud and data centre industries, the community is able to bring a set of solutions to the market that are interoperable and best in class of what the appealers behind NFV, which is vendor choice and best of breed solutions in the market. The CHARISMA project can leverage from OPNFV, firstly, because it is aligned with our main Architecture/API concept (and ETSI), and secondly because it is running environment.

A network service can be viewed architecturally as a forwarding graph of Network Functions (NFs) interconnected by supporting network infrastructure. These network functions can be implemented in a single operator network or interwork between different operator networks. The underlying network function behaviour contributes to the behaviour of the higher-level service. Hence, the network service behaviour is a combination of the behaviour of its constituent functional blocks, which can include individual NFs, NF Sets, NF Forwarding Graphs, and/or the infrastructure network. The end points and the

network functions of the network service are represented as nodes and correspond to devices, applications, and/or physical server applications. An NF Forwarding Graph can have network function nodes connected by logical links that can be unidirectional, bidirectional, multicast and/or broadcast. A simple example of a forwarding graph is a chain of network functions. An example of such an end-to-end network service can include a smartphone, a wireless network, a firewall, a load balancer and a set of CDN servers. The NFV area of activity is within the operator owned resources. Therefore, a customer-owned device, e.g. a mobile phone is outside the scope as an operator cannot exercise its authority on it. However, virtualization and network-hosting of customer functions is possible and is in the scope of NFV (e.g. see use cases Virtual Network Platform-as-a-Service (VNPaaS) and Virtualization of the Home Environment. All those functions have the Vn-Nf interface with P-Infrastructure (P – Physical) or can work on top of vSwitch to get full networking connectivity and benefit from vSwitch functions.

ENET task is to improve the connectivity, and decrease latency by HW forwarding offload, as shown in the Figure 4-4. The logical position of the NIC (Network Adapter Card) is at Vn-Nf API.

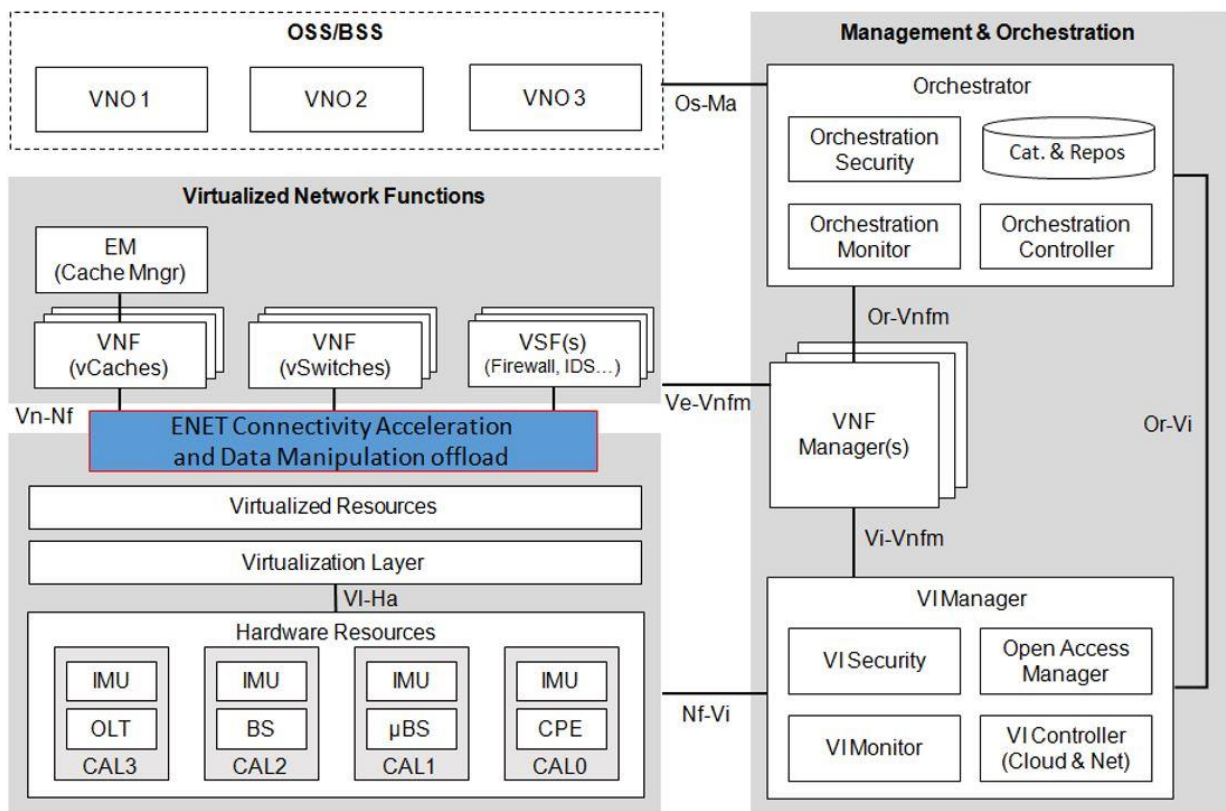


Figure 4-4: CHARISMA Open Access secure management interfaces

NFV emphasizes the fact that the exact physical deployment of a VNF instance on the infrastructure is not visible from the E2E service perspective, with the exception of guaranteeing specific policy constraints (e.g. location awareness required to implement a virtualized CDN cache node (see the use case Virtualization of CDNs (vCDN), or to ensure redundant infrastructures that are in different locations). This enables a VNF instance to be implemented on different physical resources, e.g. compute resources and hypervisors,

and/or be geographically dispersed as long as its overall end-to-end service performance and other policy constraints are met. In any case, VNF instances and their supporting infrastructure need to be visible for configuration, diagnostic and troubleshooting purpose. The ENET solution also has to take into consideration methods use by VNF/VI Managers to provide full monitoring granularity per function and per flow in each function.

5. Conclusions

Throughout this document, a high level explanation has been provided for the control, management and orchestration plane of CHARISMA. An initial design and architecture concept based on the virtualized network paradigm has been presented, which also includes Software Defined Networking (SDN) and Network Function Virtualization (NFV) technologies as drivers for the achievement of the CHARISMA secure CMO plane.

In this first deliverable of WP3, we have provided a description of the state-of-the-art of SDN and NFV technologies that form the background context for our CHARISMA CMO architecture design. We have also described the key drivers for the CHARISMA project in general that will inherently influence the CMO work being carried out in this WP: Open Access, Security, and Low latency.

The secure CMO architecture adopted for the CHARISMA network consists of four groups: Virtualized Infrastructure, Virtualized Network Functions, Management and Orchestration, and Operations and Business Support Systems. These have been defined in detail in section 4, which is the key section of this document. In particular, the hierarchical approach of the CHARISMA architecture via the different aggregation levels (CALs) each with its own intelligent management unit (IMU) and how it influences the resulting CMO architecture has also been described.

The work and high level design that has been proposed in this deliverable will be refined and influenced by the work carried out in other WPs and it will follow-up from discussions currently happening in other WPs, especially WP1 and WP2. This document will be supported by an internal document, planned for M12 that will extend this deliverable to include an updated design with workflows and interfaces (internal and external) specification.

References

- [1] Controller, NOX Openflow. [Online] <https://github.com/noxrepo/nox> .
- [2] Controller, POX Openflow. [Online] <https://github.com/noxrepo/pox> .
- [3] Beacon [Online] <https://openflow.stanford.edu/display/Beacon/Home>.
- [4] Maestro Platform -A scalable control platform written in Java which. [Online] <http://code.google.com/p/maestro-platform>.
- [5] Floodlight, Project Floodlight - Open Source Software for Building Software-Defined Networks Project. [Online] www.projectfloodlight.org/floodlight.
- [6] OpenDaylight Project. [Online] <https://www.opendaylight.org/>
- [7] Network-as-a-Service, OpenNaaS – Open Platform for. [Online] <http://opennaas.org>.
- [8] Ferrer Riera, J.; Escalona, E.; Batalle, J.; Garcia-Espin, J.A.; Figuerola, S., "Management of Virtual Infrastructures through OpenNaaS," in *Smart Communications in Network Technologies (SaCoNeT), 2013 International Conference on* , vol.02, no., pp.1-5, 17-19 June 2013.
- [9] OFERTIE. [Online] <http://www.ofertie.org/> .
- [10] CONTENT [Online] <http://content-fp7.eu/> .
- [11] SODALES [Online] <http://www.fp7-sodales.eu/> .
- [12] OASE. [Online] <http://www.ict-oase.eu/>
- [13] Neutron, OpenStack. [Online] <http://wiki.openstack.org/wiki/Neutron>.
- [14] OpenDaylight Virtual Tenant Network. [Online] https://wiki.opendaylight.org/view/OpenDaylight_Virtual_Tenant_Network_%28VTN%29:Main.
- [15] ON.LAB. Flowvisor. [Online] <https://github.com/OPENNETWORKINGLAB/flowvisor/wiki> .
- [16] OpenVirtex. Programmable virtual networks. [Online] <http://ovx.onlab.us/>.
- [17] Juniper Networks. OpenContrail. OpenContrail. [Online] 2014. <http://opencontrail.org/>.
- [18] OpenStack. Open source software for building. OpenStack. [Online] 2014. www.openstack.org.
- [19] Eucalyptus Systems, Inc. Eucalyptus. Eucalyptus. [Online] 2014. www.eucalyptus.com.
- [20] The Apache Software Foundation. Apache CloudStack™ - Open Source Cloud Computing™. Apache CloudStack. [Online] 2014. <http://cloudstack.apache.org/>.
- [21] vmware. vCloud Suite. vmware.com. [Online] 2014. <http://www.vmware.com/products/vcloud-suite>.
- [22] OpenBaton Project. [Online] <http://openbaton.github.io/>
- [23] OpenStack Tacker Project. [Online] <https://wiki.openstack.org/wiki/Tacker>
- [24] OpenMano Project. [Online] <https://github.com/nfvlabs/openmano>
- [25] T-NOVA project. Specification of the Network Function Framework and T-NOVA Marketplace. 2014. D2.41. 72. Use System Cases and Requirements. 2014. D2.1. 73. Overall System Architecture and Interfaces. 2014. D2.21.

- [26] UNIFY Project. [Online] <https://www.fp7-unify.eu/>.
- [27] MCN Project. [Online] <http://www.mobile-cloud-networking.eu>.
- [28] Cloudify. [Online] <http://getcloudify.org/>.
- [29] Cyan NFV Orchestartor – BluePlanet. [Online] <http://www.blueplanet.com/products/nfv-orchestration.html>.
- [30] HP OpenNFV. [Online] <http://www8.hp.com/us/en/cloud/nfv-architecture.html> .
- [31] Cisco White Paper: “Cisco Visual Networking Index: Forecast and Methodology, 2012-2017,” 29, May 2013.
- [32] Malandrino, F.; Casetti, C.; Chiasserini, C., "Content Discovery and Caching in Mobile Networks with Infrastructure," IEEE Transactions on Computers, vol.61, no.10, pp.1507,1520, Oct. 2012
- [33] Hazem Gomaa, Geoffrey G. Messier, Robert Davies, Carey Williamson, "Peer-Assisted Caching for Scalable Media Streaming in Wireless Backhaul Networks," IEEE Global Telecommunications Conference (GLOBECOM 2010), pp.1-5, Dec. 2010
- [34] Hasti Ahlehagh and Sujit Dey, "Video caching in Radio Access Network: Impact on delay and capacity," IEEE Wireless Communications and Networking Conference (WCNC), pp.2276-2281, April 2012.
- [35] Negin Golrezaei, Karthikeyan Shanmugam, Alexandros G. Dimakis, Andreas F. Molisch, and Giuseppe Caire, "FemtoCaching: Wireless video content delivery through distributed caching helpers," Proceedings IEEE INFOCOM, pp.1107-1115, March 2012.
- [36] Shinae Woo, Eunyoung Jeong, Shinjo Park, Jongmin Lee, Sunghwan Ihm, and KyoungSoo Park. "Comparison of caching strategies in modern cellular backhaul networks". In Proceeding of the 11th annual international conference on Mobile systems, applications, and services (MobiSys '13). New York, NY, USA, 2013.
- [37] Network Functions Virtualisation (NFV); Architectural Framework, ETSI Standard GS NFV 002, 2014.
- [38] *Network Functions Virtualisation (NFV); NFV Performance & Portability Best Practises*, ETSI Standard [GS NFV-PER 001](#), 2014.
- [39] Network Functions Virtualisation (NFV); NFV Security; Problem Statement, ETSI Standard GS NFV-SEC 001, 2014.
- [40] ETSI GS NFV 003: "Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV"
- [41] Media Access Control (MAC) Security. [Online] <http://www.ieee802.org/1/pages/802.1ae.html> .

Acronyms

Acronym	Definition
5G	5th Generation of Wireless Network
AAA	Authentication Authorization Accounting
API	Application Programming Interface
AuthN	Authentication
AuthZ	Authorization
BGP	Border Gateway Protocol
BS	Base Station
BSS	Business Support System
CAL	CHARISMA Aggregation Level
CAPEX	Capital Expenditures
CDN	Content Distribution Network
CFS	Customer Facing Services
CLC	Cloud Controller
CLI	Command Line Interface
CMO	Control Management Orchestration
CO	Central Office
CRM	Customer Relationship Management
D2D	Device to device connectivity
DC	Data Centre
DNS	Domain Name Service
DPDK	Data Plane Development Kit
DPI	Deep Packet Inspection
EEU	Enterprise End User
EMS	Element Management System
EPC	Evolved Packet Core
EPCaaS	EPC as a Service
EPN	Evolved Programmable Network
ESP	Evolved Services Platform
ESX	Elastic Sky X
ETSI	European Telecommunications Standards Institute
EU	European Union
FTTx	Fibre to the X
GPG	GNU Privacy Guard
HTML	Hyper Text Markup Language
IDS	Intrusion Detection Systems
IMS	IP Multimedia Subsystem
IMSaaS	IMS as a Service
IMU	Intelligent Management Unit
IP	Internet Protocol

IRRH	i-Remote Radio Head
ISG	Industry Specification Groups
ISP	Internet Service Provider
IT	Information Technology
KVM	Kernel-based Virtual Machine
KPI	Key Performance Indicator
LR-PON	Long Reach Passive Optical Network
LTE	Long Term Evolution
MANO	Management and Orchestration
MCN	Mobile Cloud Networking
NaaS	Network as a Service
NFaaS	Network Function as a Service
NFV	Network Function Virtualization
NFVI	Network Function Virtualization Infrastructure
NFVO	Network Function Virtualization Orchestrator
NIC	Network Interface Card
NP	Network Provider
NSD	Network Service Descriptors
OAN	Open Access Network
OCCI	Open Cloud Computing Interface
OLT	Optical Line Termination
ONT	Optical Network Termination
OpenNaas	Open Network as a Service
OPEX	Operational Expenditures
OS	Operating System
OSS	Operations Support System
OVC	Operator Virtual Connections
PKI	Public Key Infrastructure
PoP	Point of Presence
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
RANaaS	RAN as a Service
REST	Representational Stateless Transfer
RFS	Resource Facing Services
SDN	Software Defined Networks
SIC	Service Instance Component
SLA	Service Level Agreement
SOTA	State of the Art
SP	Service Provider
SR-IOV	Single Root I/O Virtualization
TOSCA	Topology and Orchestration Specification for Cloud Application
UI	User Interface

v-CPE	virtual Customer Premises Equipment
VI	Virtualized Infrastructure
VIM	Virtualized Infrastructure Manager
VM	Virtual Machine
VNF	Virtual Network Function
VNFD	VNF Descriptor
VNFM	VNF Manager
VNO	Virtual Network Operator
v-security	Virtualized Security
VTN	Virtual Tenant Network
WDM	Wavelength Division Multiplexing
WOT	Web of Trust
YAML	YAML Ain't Markup Language

<END OF DOCUMENT>