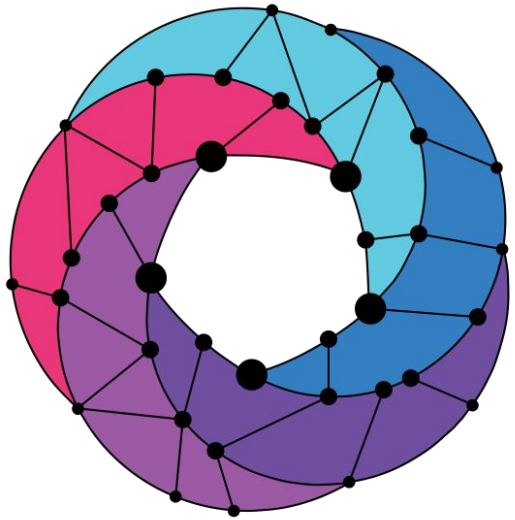


Open Source
MANO

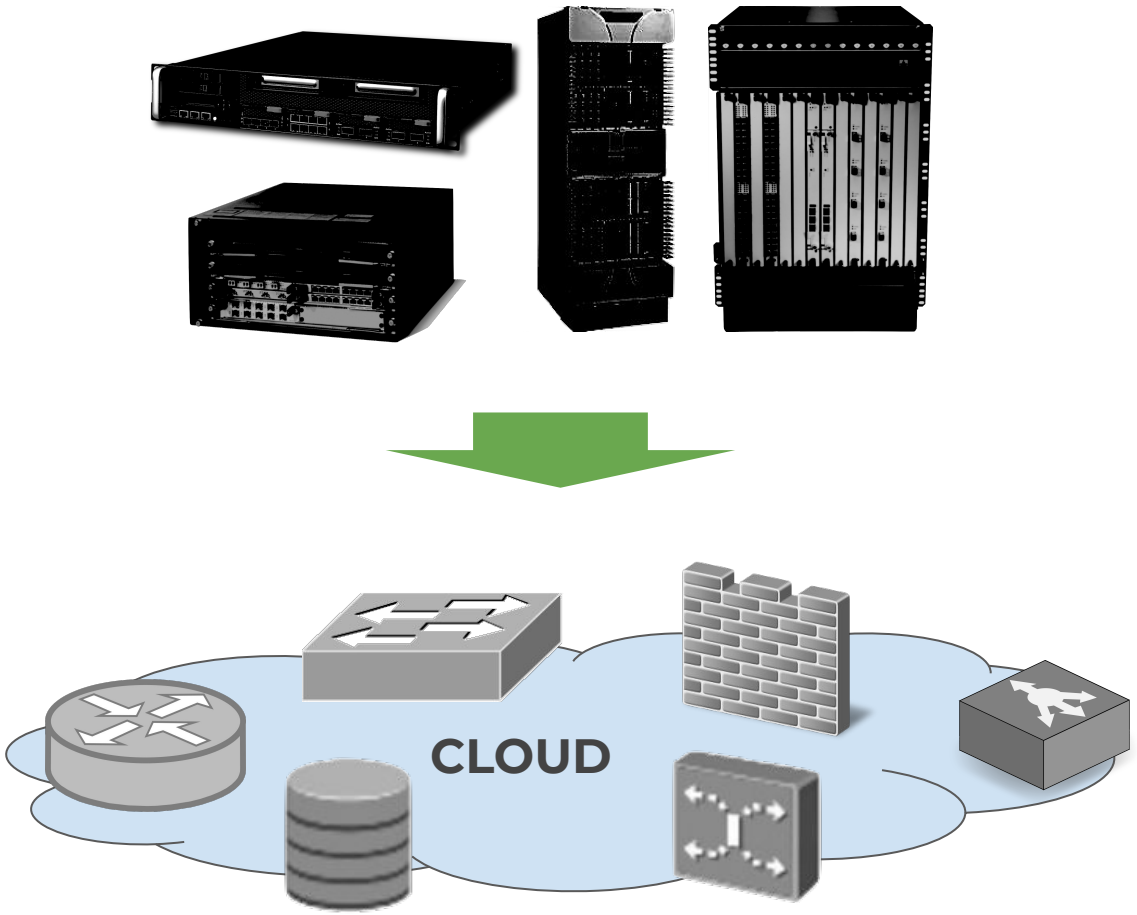
OSM#9 Hackfest
Hack 0: Introduction to NFV and OSM

ETSI

Open Source MANO

Introduction to ETSI NFV and standards

What is NFV trying to address?



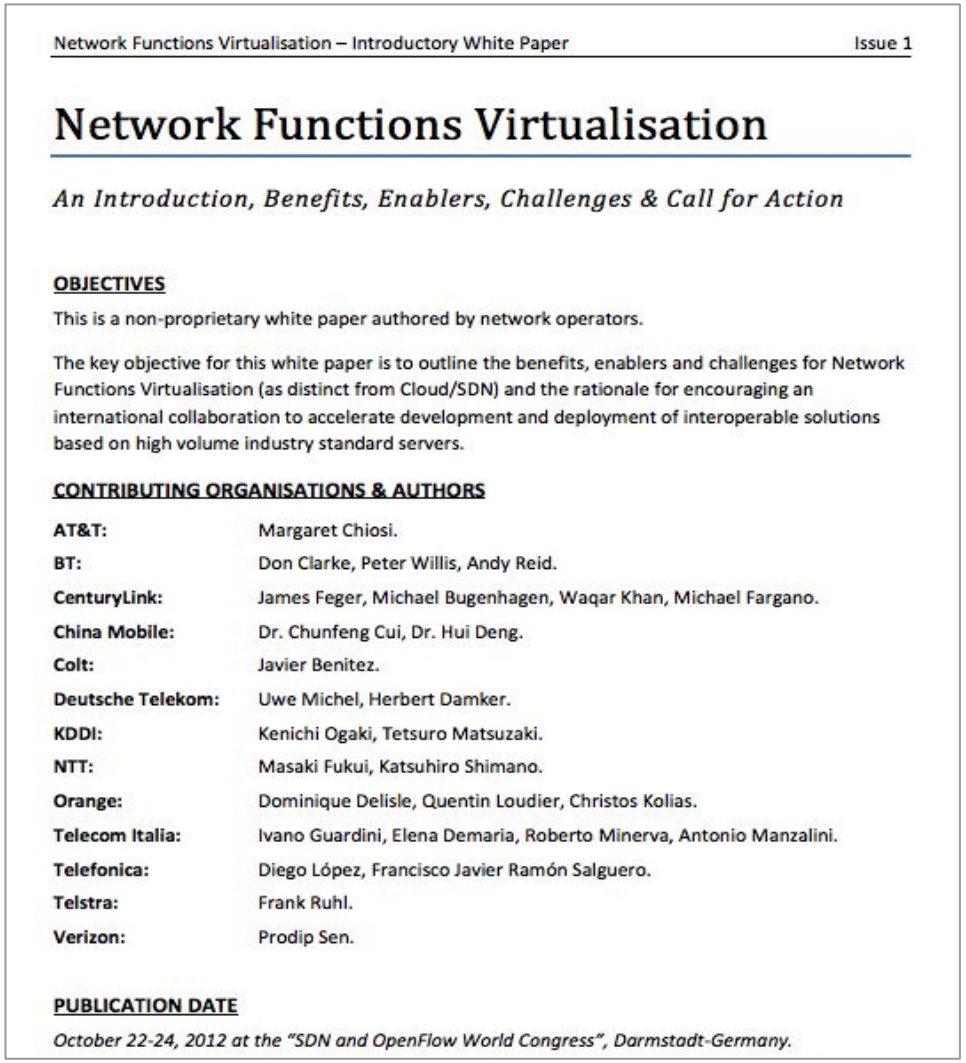
NFV proposes to **virtualize, over commodity hardware, network functions that typically run in dedicated appliances, why?**

- To make network operations more agile and cost-efficient.
- To increase independency of hardware vendors.
- To leverage all the advantages of the Cloud, for network functions.

The original idea triggered an industry movement

Open Source MANO

- Initial white paper was written in 2012 by the world's leading telecom network operators (Europe, América & Asia).
- This group evolved to the ETSI NFV ISG (Industry Specification Group), formed by 300+ companies.
- Their main motivation had to do with reducing TCO of building a network by using open solutions.



Network Functions Virtualisation – Introductory White Paper Issue 1

Network Functions Virtualisation

An Introduction, Benefits, Enablers, Challenges & Call for Action

OBJECTIVES

This is a non-proprietary white paper authored by network operators.

The key objective for this white paper is to outline the benefits, enablers and challenges for Network Functions Virtualisation (as distinct from Cloud/SDN) and the rationale for encouraging an international collaboration to accelerate development and deployment of interoperable solutions based on high volume industry standard servers.

CONTRIBUTING ORGANISATIONS & AUTHORS

AT&T:	Margaret Chiosi.
BT:	Don Clarke, Peter Willis, Andy Reid.
CenturyLink:	James Feger, Michael Bugenhagen, Waqar Khan, Michael Fargano.
China Mobile:	Dr. Chunfeng Cui, Dr. Hui Deng.
Colt:	Javier Benitez.
Deutsche Telekom:	Uwe Michel, Herbert Damker.
KDDI:	Kenichi Ogaki, Tetsuro Matsuzaki.
NTT:	Masaki Fukui, Katsuhiro Shimano.
Orange:	Dominique Delisle, Quentin Loudier, Christos Kolias.
Telecom Italia:	Ivano Guardini, Elena Demaria, Roberto Minerva, Antonio Manzalini.
Telefonica:	Diego López, Francisco Javier Ramón Salguero.
Telstra:	Frank Ruhl.
Verizon:	Prodip Sen.

PUBLICATION DATE

October 22-24, 2012 at the "SDN and OpenFlow World Congress", Darmstadt-Germany.

https://portal.etsi.org/nfv/nfv_white_paper.pdf

ETSI Publications

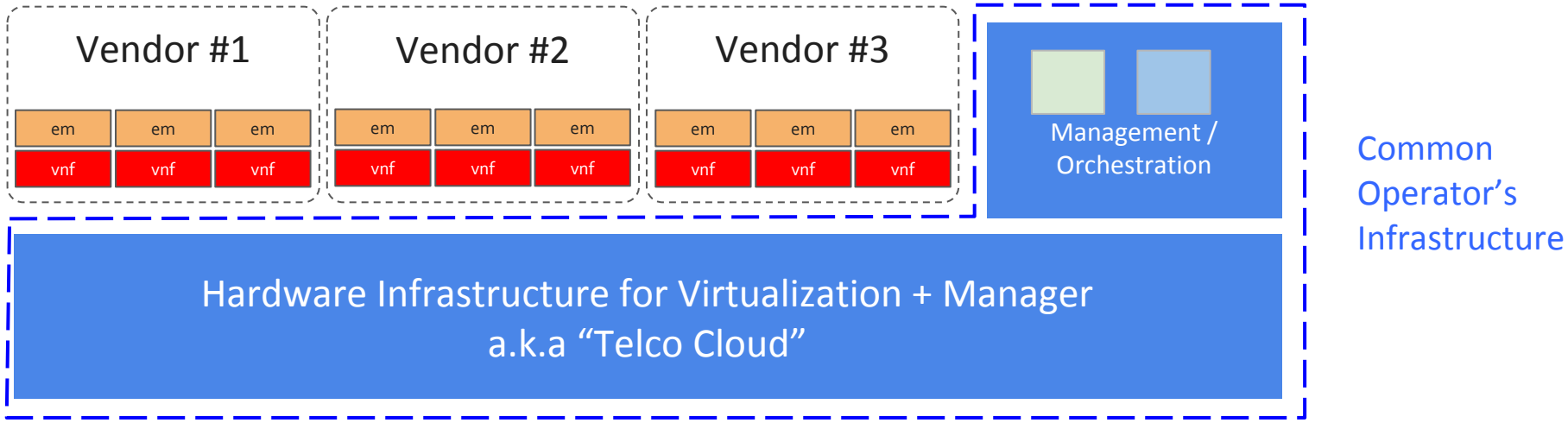
- Based on member's feedback, field experiences and proof of concepts, standard documents have evolved.
- 60+ publications exist today, including the following three main documents:
  - NFV Architectural Framework
    http://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.02.01_60/gs_NFV002v010201p.pdf
  - NFV Infrastructure Overview
    http://www.etsi.org/deliver/etsi_gs/NFV-INF/001_099/001/01.01.01_60/gs_NFV-INF001v010101p.pdf
  - NFV Management and Orchestration
    http://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.02.01_60/gs_NFV002v010201p.pdf

http://www.etsi.org/standards-search

Benefits of a standard NFV architecture

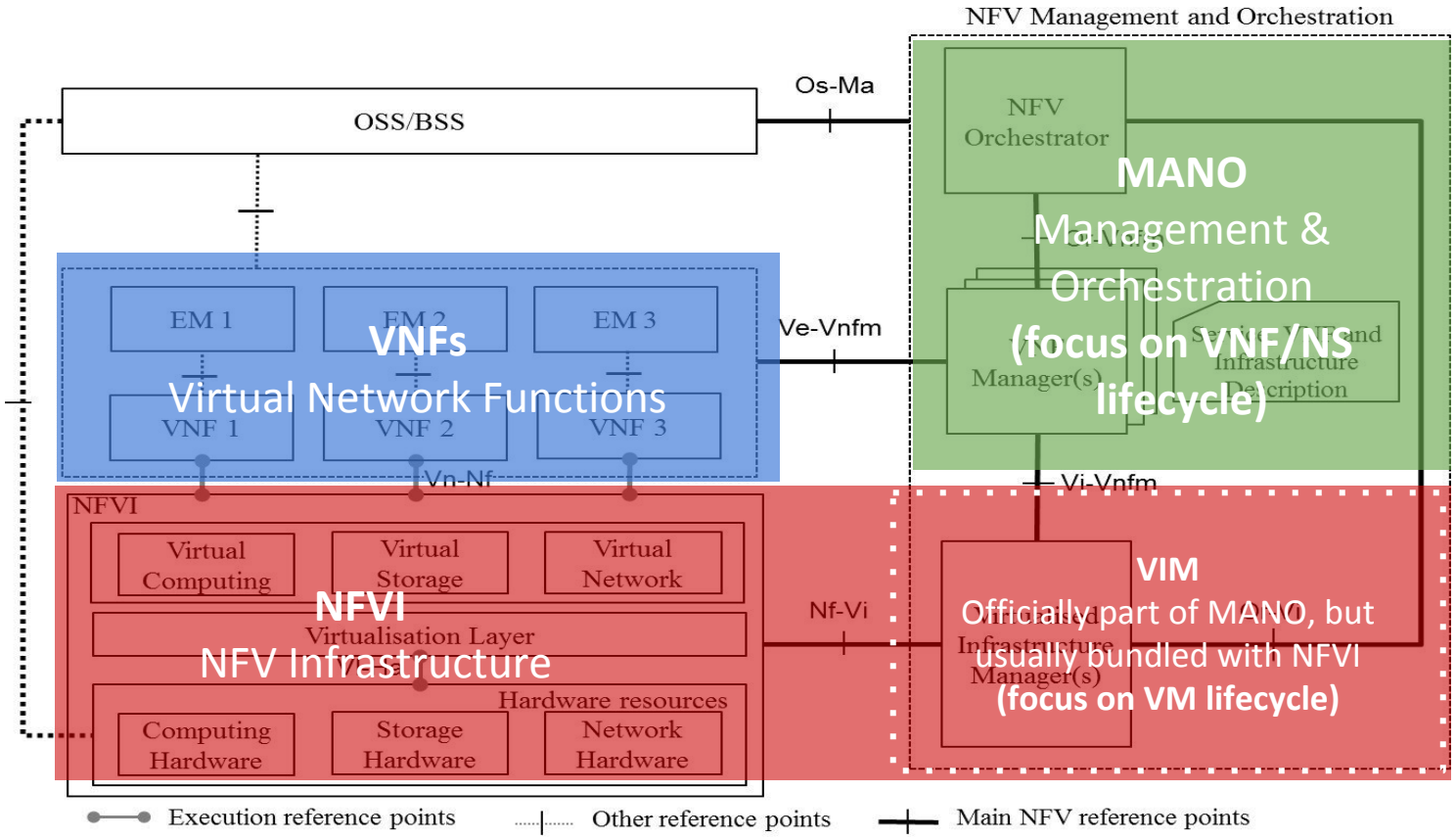
Open Source MANO

The ultimate goal is for operators to have a **unified and generic virtualization infrastructure**, compatible with any vendor's Virtual Networking Function (VNF), **this makes standardization a must**.



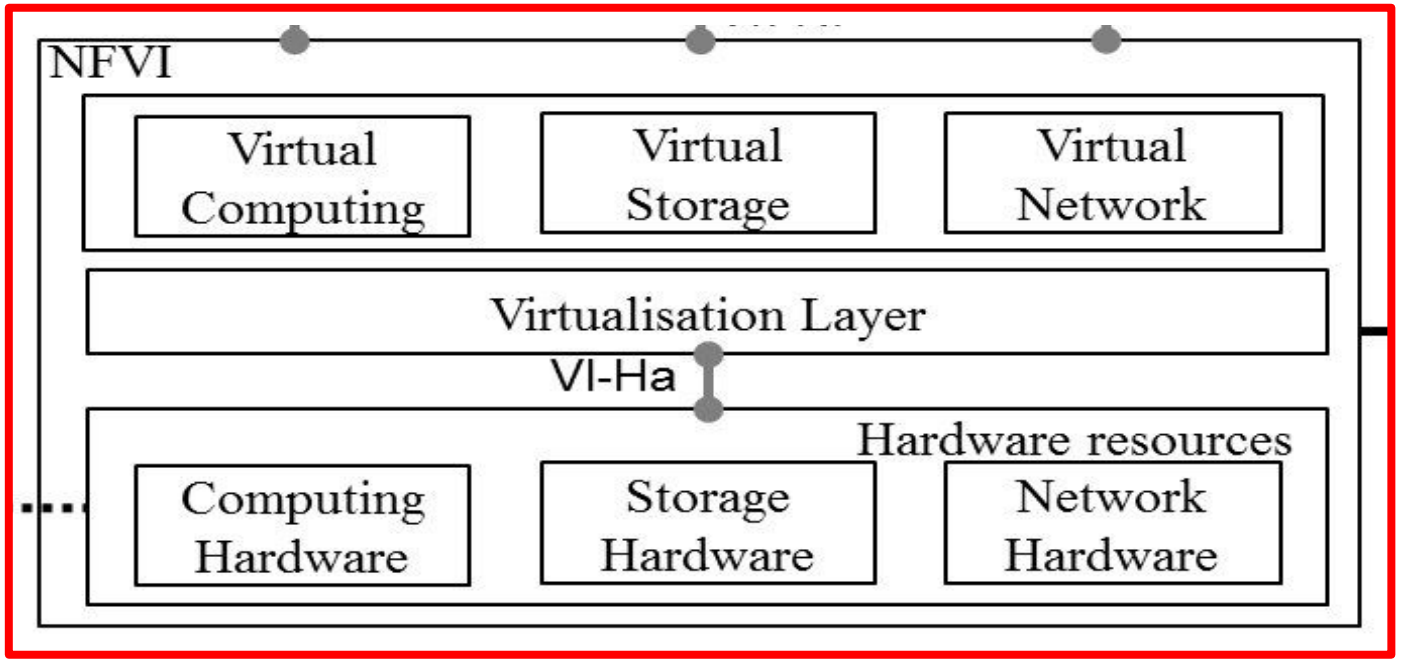
The ETSI NFV Architectural Framework

The standard architecture can be better understood in three blocks:



NFVI: NFV Infrastructure

NFVI goal is to provide a virtualization environment for VNFs, including virtual compute, storage and networking resources.



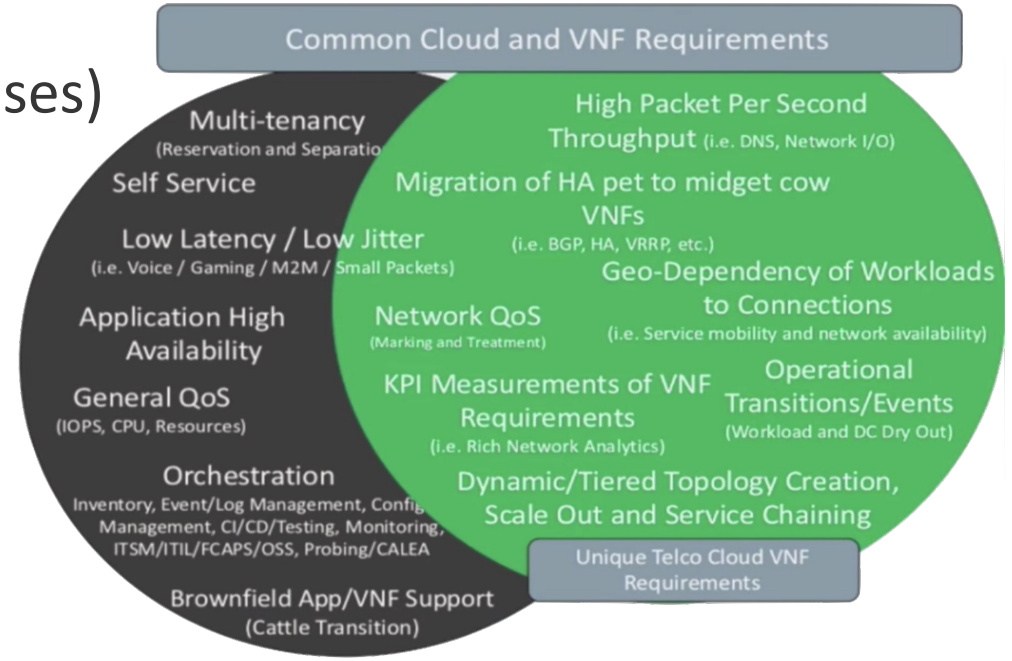
NFVI: NFV Infrastructure

VNF Special Requirements

Open Source MANO

VNFs, especially data-plane ones, usually have additional requirements than common cloud applications, including:

- **Minor latency** (disk I/O & network)
  → Faster hardware (More cores, SSD disks, faster buses)
  → Dataplane acceleration
- **Higher throughput or PPS**
  → Dataplane acceleration
  → EPA: Enhanced Platform Awareness
- **Geographical distribution**
  → multi-site cloud
- **Horizontal auto-scaling**
  → automated operations (orchestration)



OpenStack Austin 2016: Telco Cloud Requirements: What VNF's Are Asking For

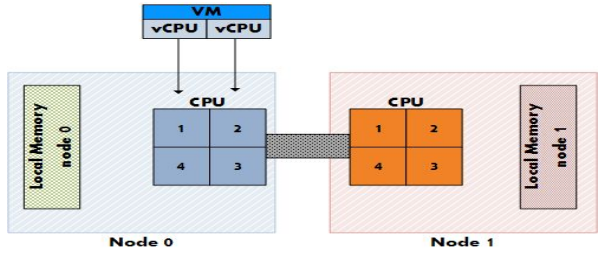
NFVI: NFV Infrastructure

VNF Special Requirements

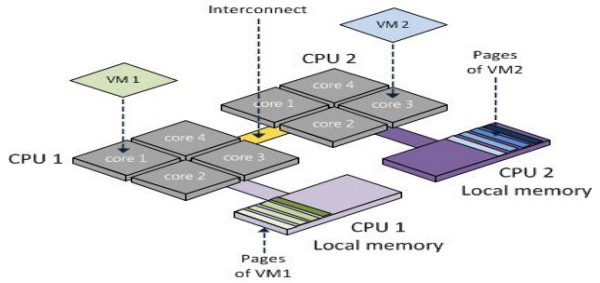
EPA covers the different approaches that can be taken at the NFVI layer to increase performance while maintaining a generic (COTS) infrastructure. VIM and MANO should be able to request them.

Huge Pages

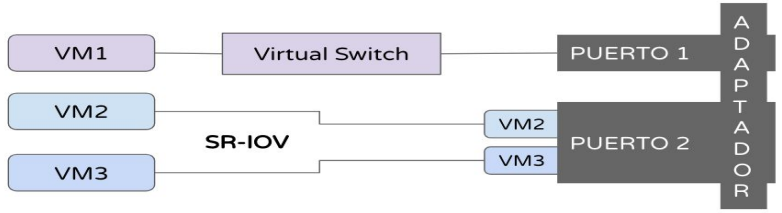
NUMA Topology Awareness



CPU Pinning

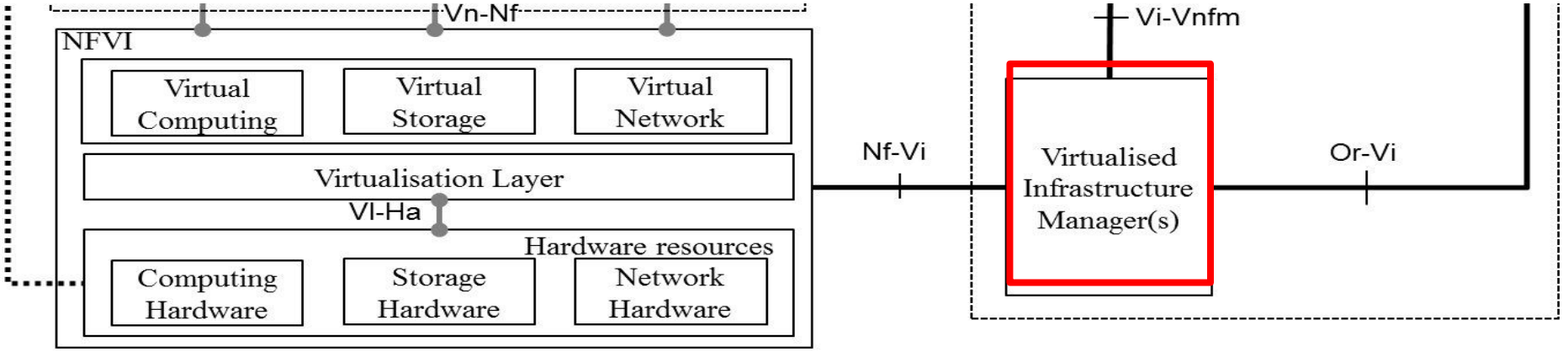


Data Plane assignment



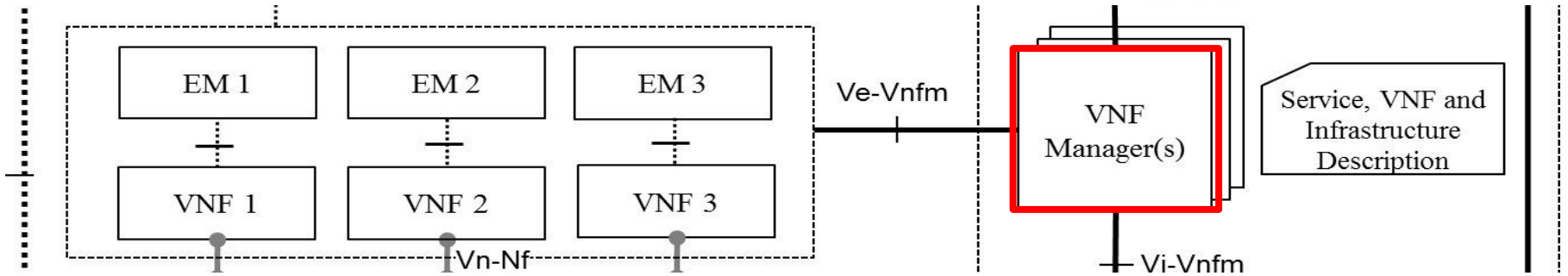
VIM: Virtualized Infrastructure Manager

The Virtualised Infrastructure Manager conceptually part of the ‘MANO Stack’, provides lifecycle management for virtualized resources (VMs, volumes, networking paths and connectivity, etc.)



MANO: VNF Manager (VNFM)

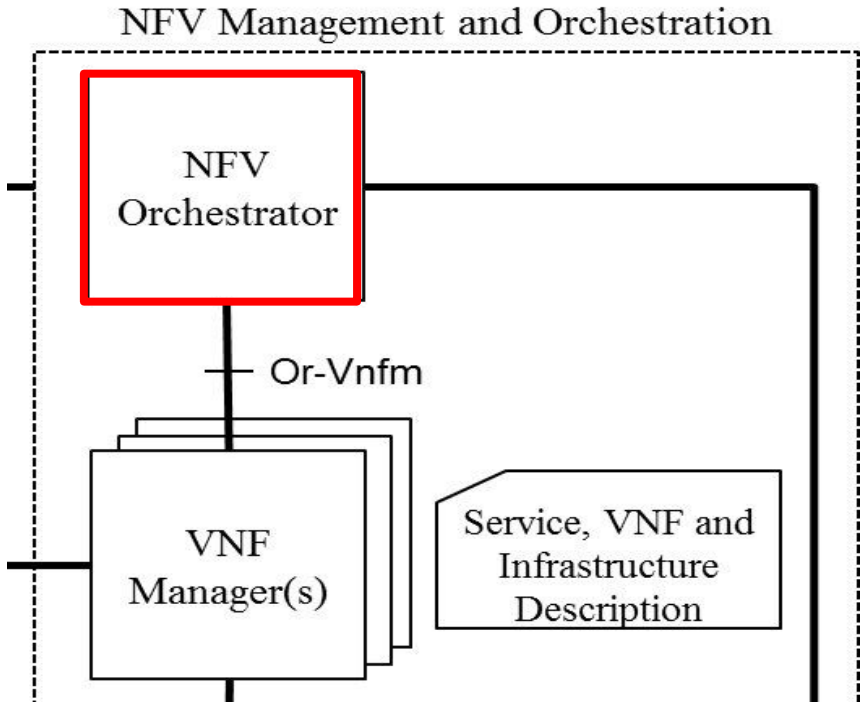
- The VNF Manager, also part of the 'MANO Stack', covers **lifecycle management for Virtual Network Functions** (VNFs), either directly or through their own Element Management System (EMS).



VNF Managers can be generic (current trend), or vendor-specific ones.

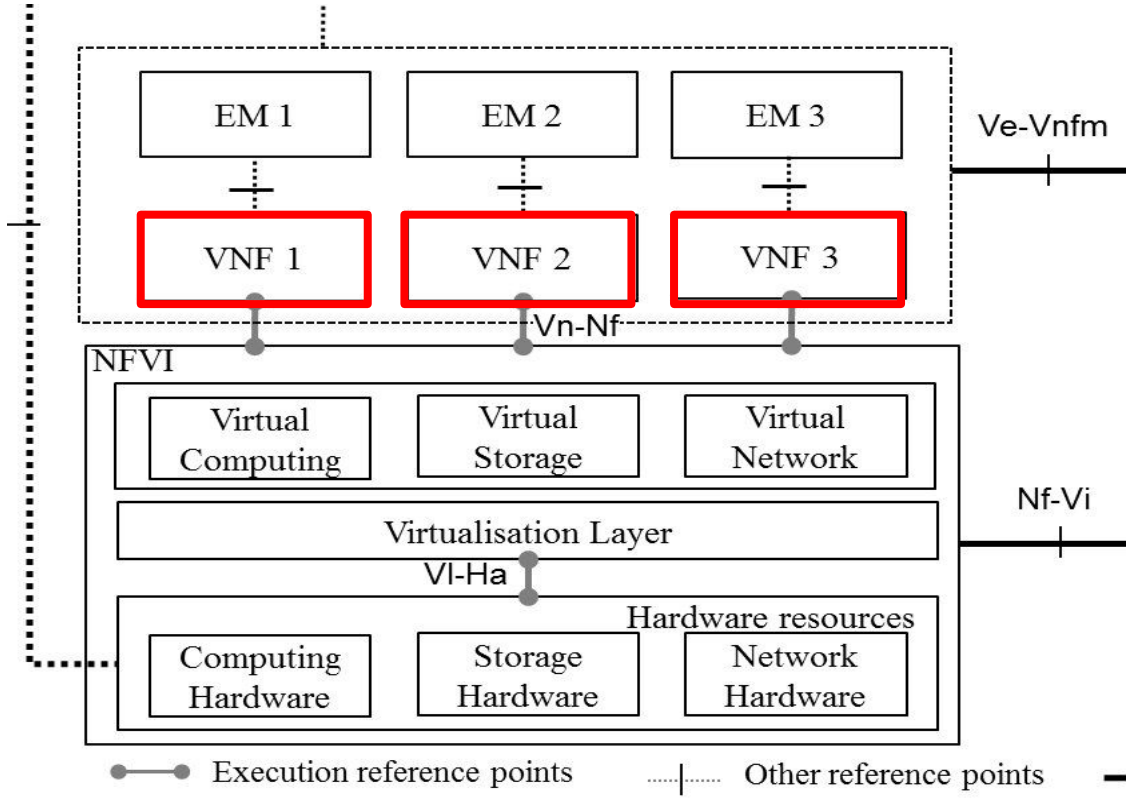
MANO: NFV Orchestrator (NFVO)

- The NFV Orchestrator, the higher entity in the ‘MANO Stack’, covers general resource orchestration and services lifecycle, which comprise multiple VNFs and define their roles (traffic paths, scaling decisions, and other service-related requirements)
- It can interact with a generic VNF Manager, or vendor-specific ones.



Virtual Network Functions (VNF)

- Finally, the VNFs, which are supported by the underlying NFVI, and managed by their own EM (internal, element manager) and the VNF Manager (external, 'context-aware' manager)
- They should be able to provide any networking function and interact with other VNFs.



VNF and Network Service descriptor files (VNFD / NSD)

Open Source MANO

One of the most important aspects of achieving a unified VNF catalogue, is having a standard way of describing VNFs and NSs.

- MANO solutions should give the possibility to describe VNFs through ‘descriptor files’
- The industry’s goal is a unified and standard descriptor file format across different platforms (ETSI SOL001/006)
- Both NS (comprised of VNFs) and VNFs should be described in a simple way.

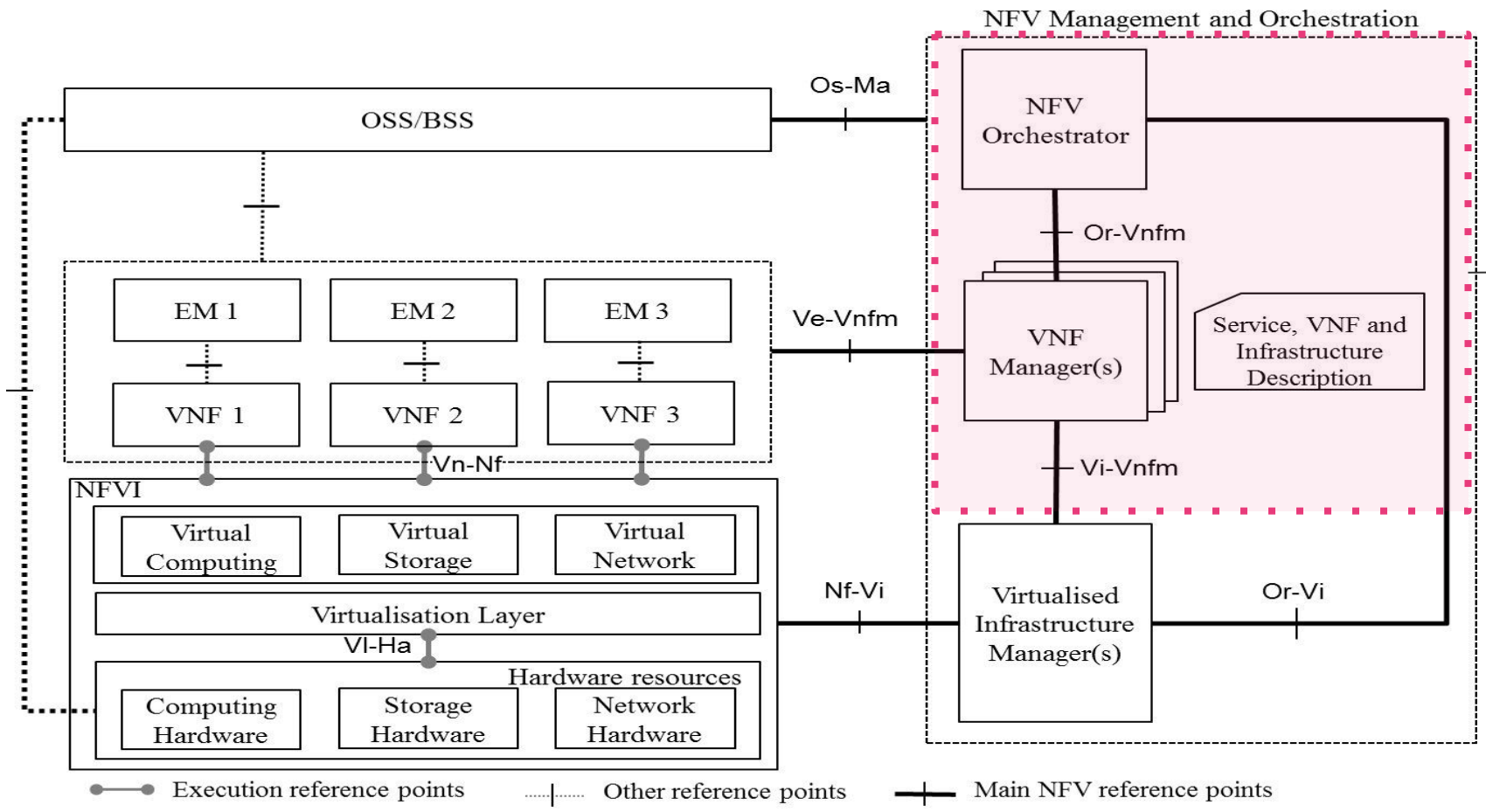
```
vnfd:vnfd-catalog:
    vnfd:vnfd:
    -   vnfd:connection-point:
        -   vnfd:name: eth0
            vnfd:type: VPORT
        vnfd:description: Generated by OSM pacakage generator
        vnfd:id: ubuntuvnf_vnfd
        vnfd:mgmt-interface:
            vnfd:cp: eth0
        vnfd:name: ubuntuvnf_vnfd
        vnfd:service-function-chain: UNAWARE
        vnfd:short-name: ubuntuvnf_vnfd
        vnfd:vdu:
        -   vnfd:cloud-init-file: cloud_init
            vnfd:count: '1'
            vnfd:description: ubuntuvnf_vnfd-VM
            vnfd:guest-epa:
                vnfd:cpu-pinning-policy: ANY
            vnfd:id: ubuntuvnf_vnfd-VM
            vnfd:image: ubuntu_admin
            vnfd:interface:
            -   rw-vnfd:floating-ip-needed: 'false'
                vnfd:external-connection-point-ref: eth0
```

The end product operators consume in an NFV world is a set of VNF Packages (which include the VNFD and other artifacts) that they can mix and match to describe their own Network Services.

Open Source
MANO

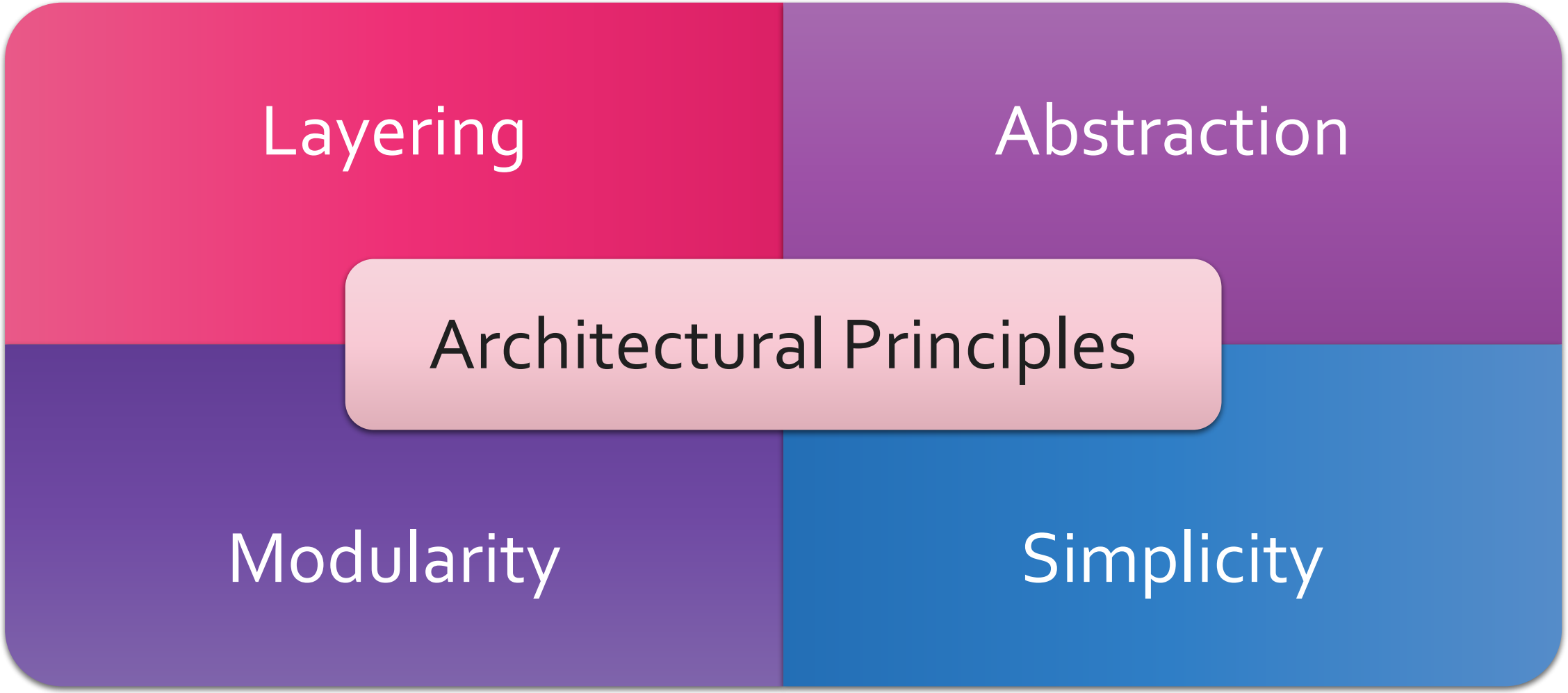
Introduction to **Open Source MANO**

The Open Source MANO Project



We are here!
Open Source MANO is an ETSI-hosted project developing an Open Source NFV Management and Orchestration (MANO) software stack aligned with ETSI NFV.

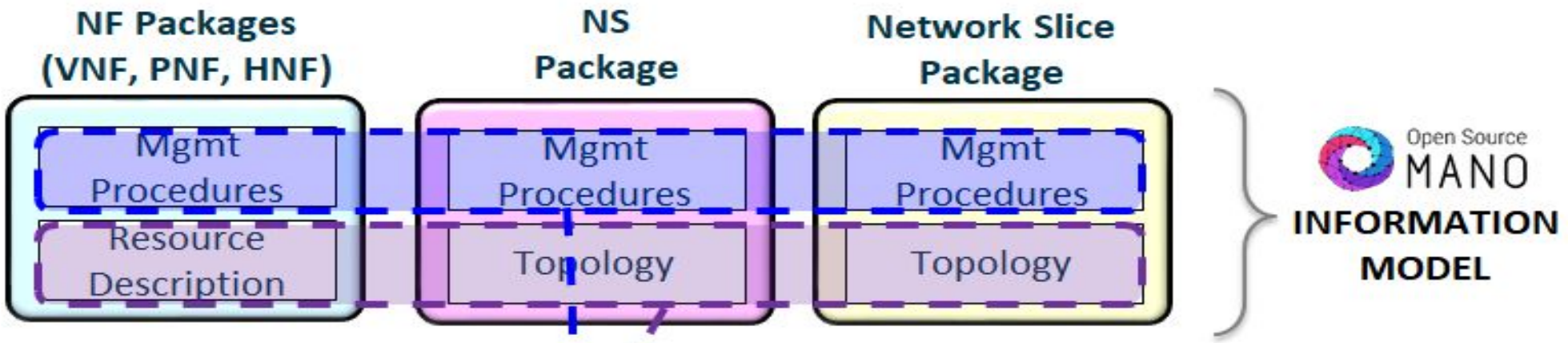
OSM Architectural Principles



OSM's approach aims to minimize integration efforts

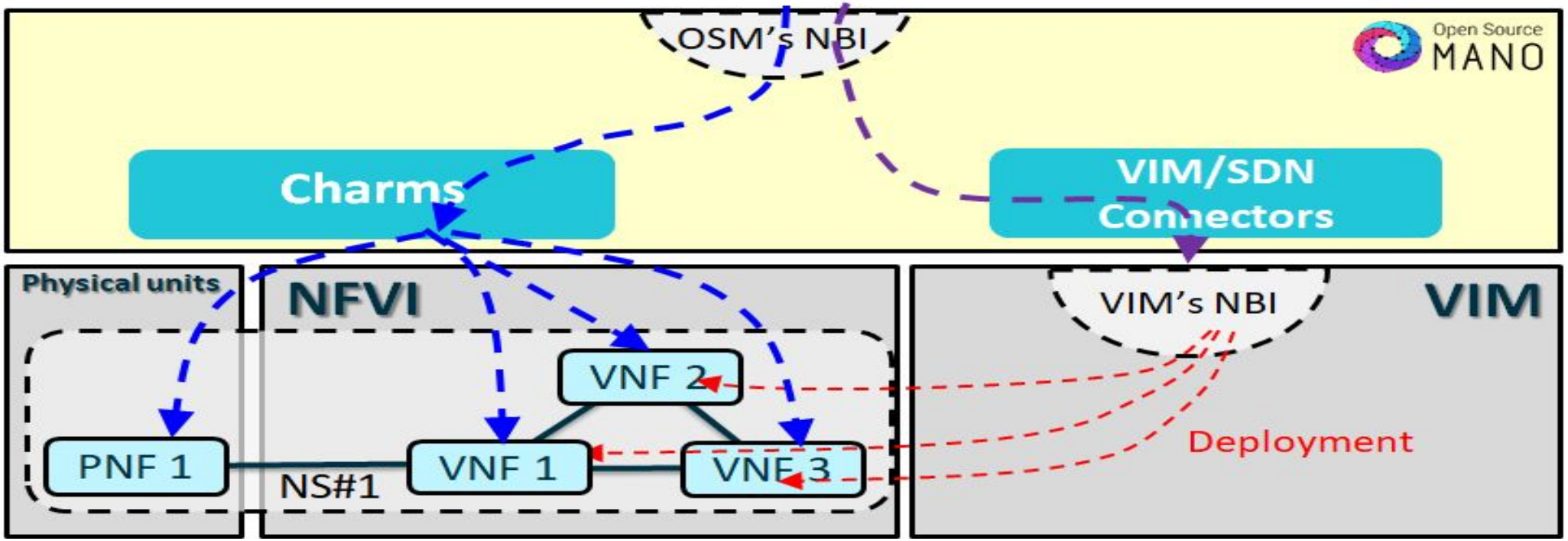
1. A well-known Information Model (IM), aligned with ETSI NFV, that is capable of modelling and automating the full lifecycle of Network Functions:

- VNFD (VNF Descriptor) → VNFR (VNF Record)
- NSD (Network Service Descriptor) → NSR (Network Service Record)
- NST (Network Slice Template) → NSI (Network Slice Instance)



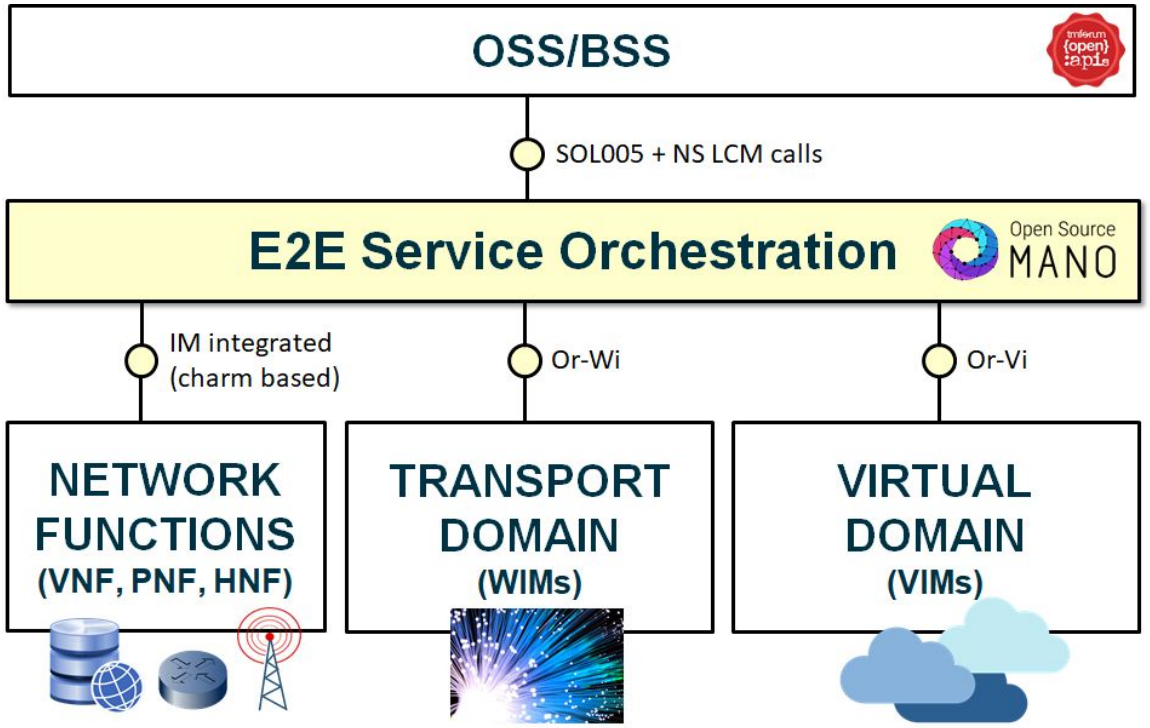
OSM's approach aims to minimize integration efforts

2. A unified northbound interface (NBI), based on NFV SOL005



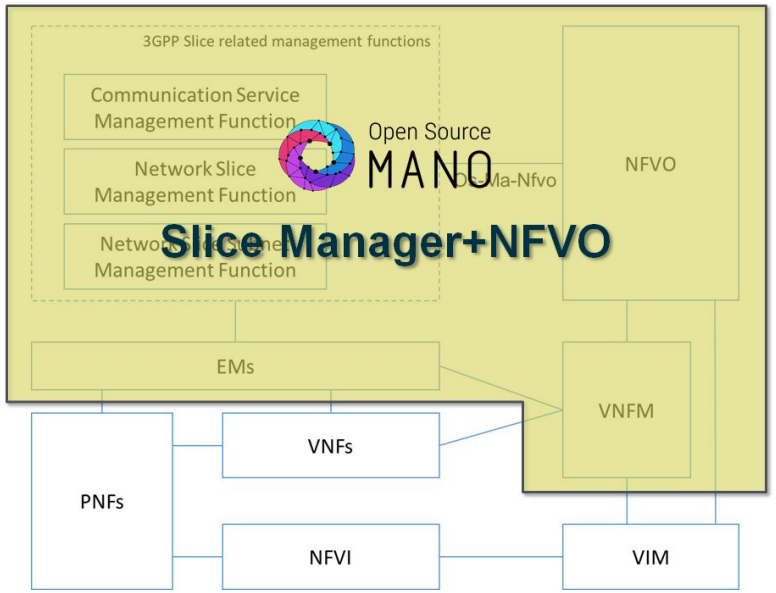
OSM's approach aims to minimize integration efforts

3. The extended concept of "Network Service" in OSM, so that an NS can span across the different domains identified and therefore control the full lifecycle of an NS interacting with VNFs, PNFs and HNFs.

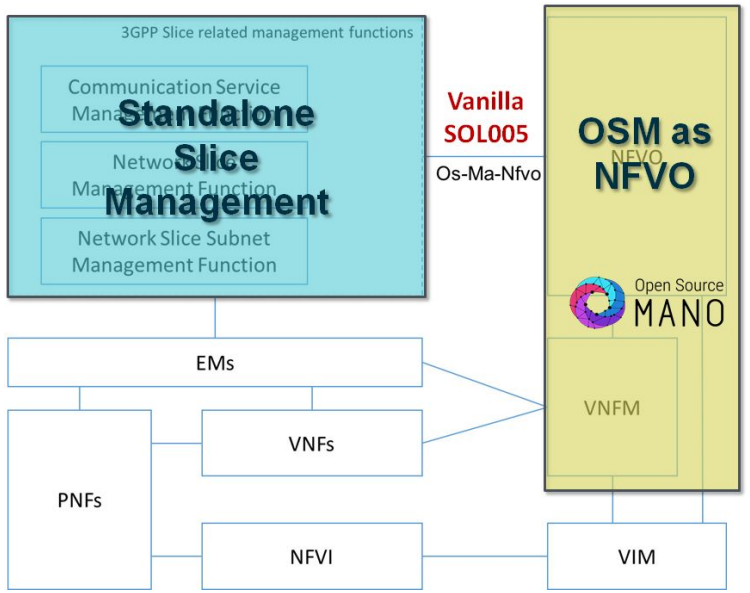


OSM's approach aims to minimize integration efforts

4. The lifecycle management of Network Slices, assuming if required the role of Slice Manager, or integrating with an external Slice Manager



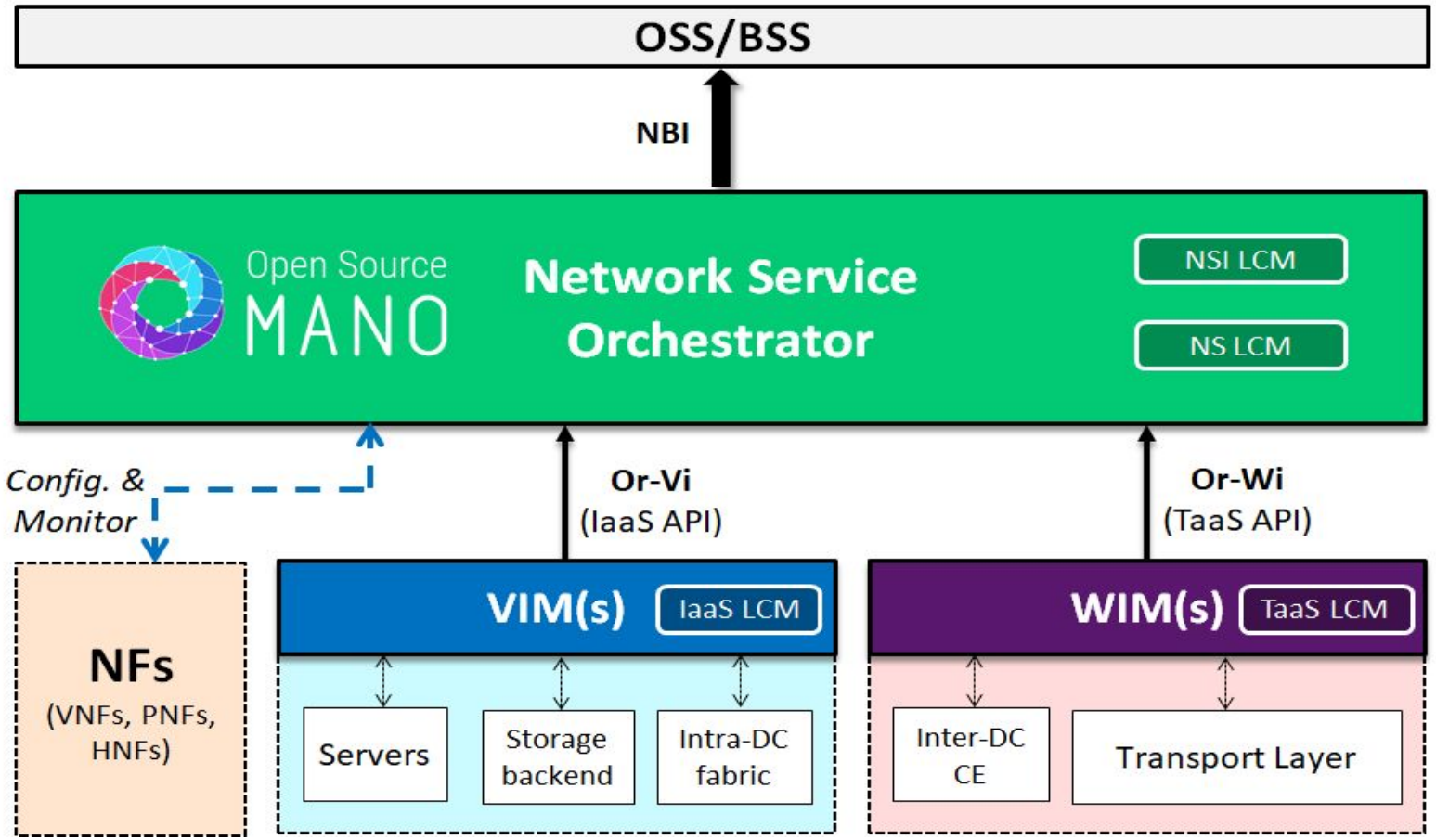
Full E2E Management
(Integrated Modelling)



Standalone Management
(Vanilla NFV/3GPP)

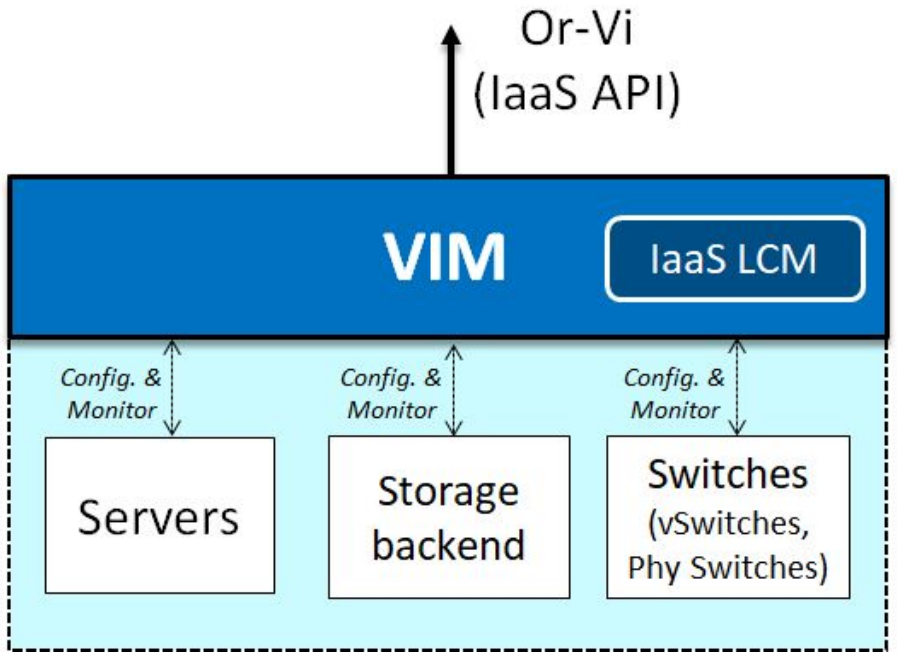
OSM Service Platform View

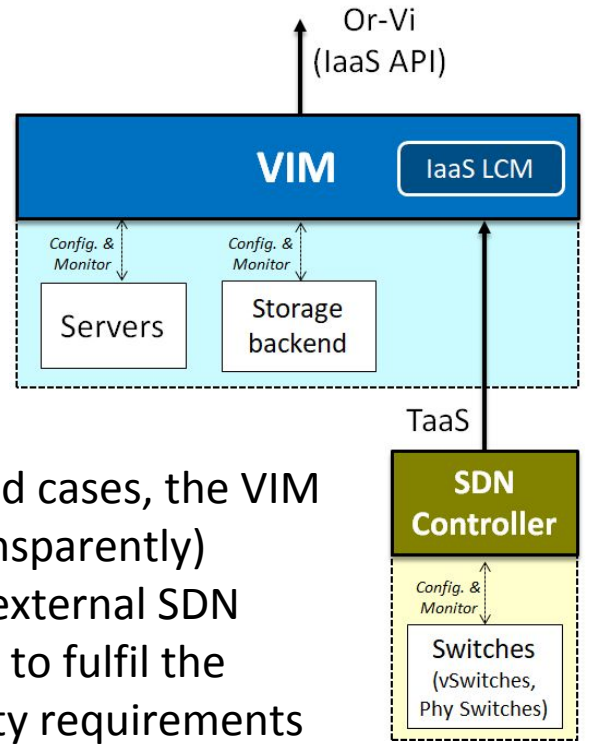
OSM implements an End-to-end Network Service Orchestrator (NSO)



OSM Integration with VIM layer

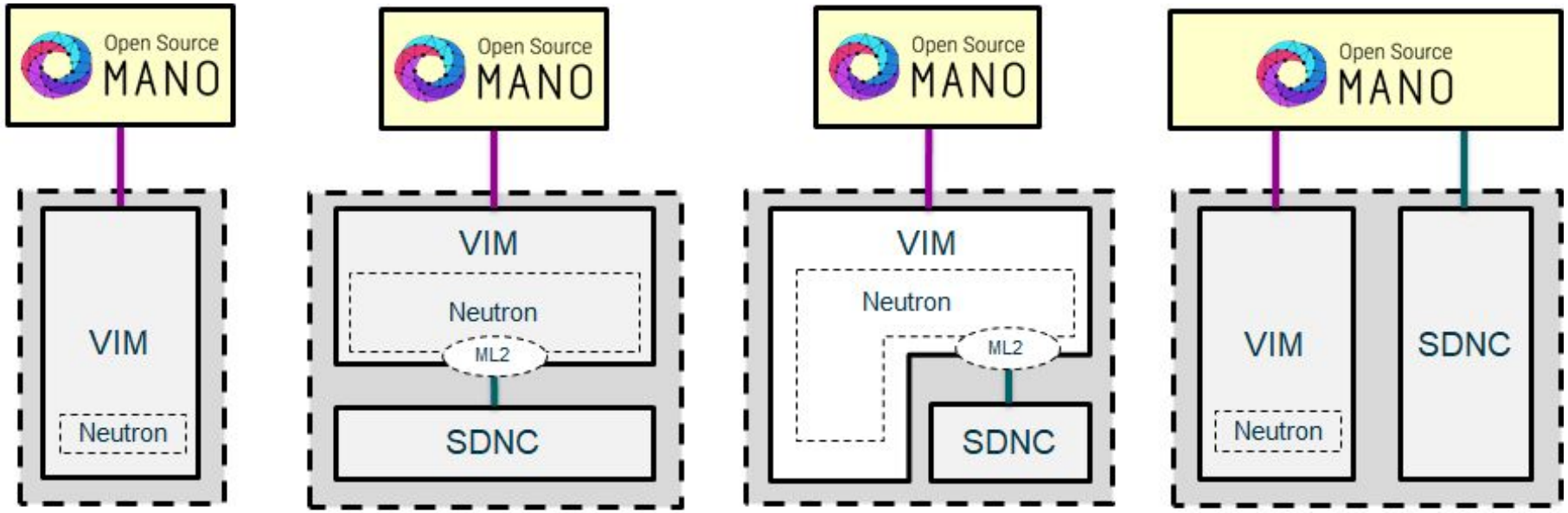
VIM layer (including K8-clusters) are managed to provide the VNF's connectivity, either in VM or Container formats





In advanced cases, the VIM might (transparently) control al external SDN Controller, to fulfil the connectivity requirements

OSM Integration with SDN Controller



CASE #1: Vanilla
- Overlay: Native
- No underlay

CASE #2: VIM + all SDN
- Overlay: SDNC
- Underlay: if available, via SDNC

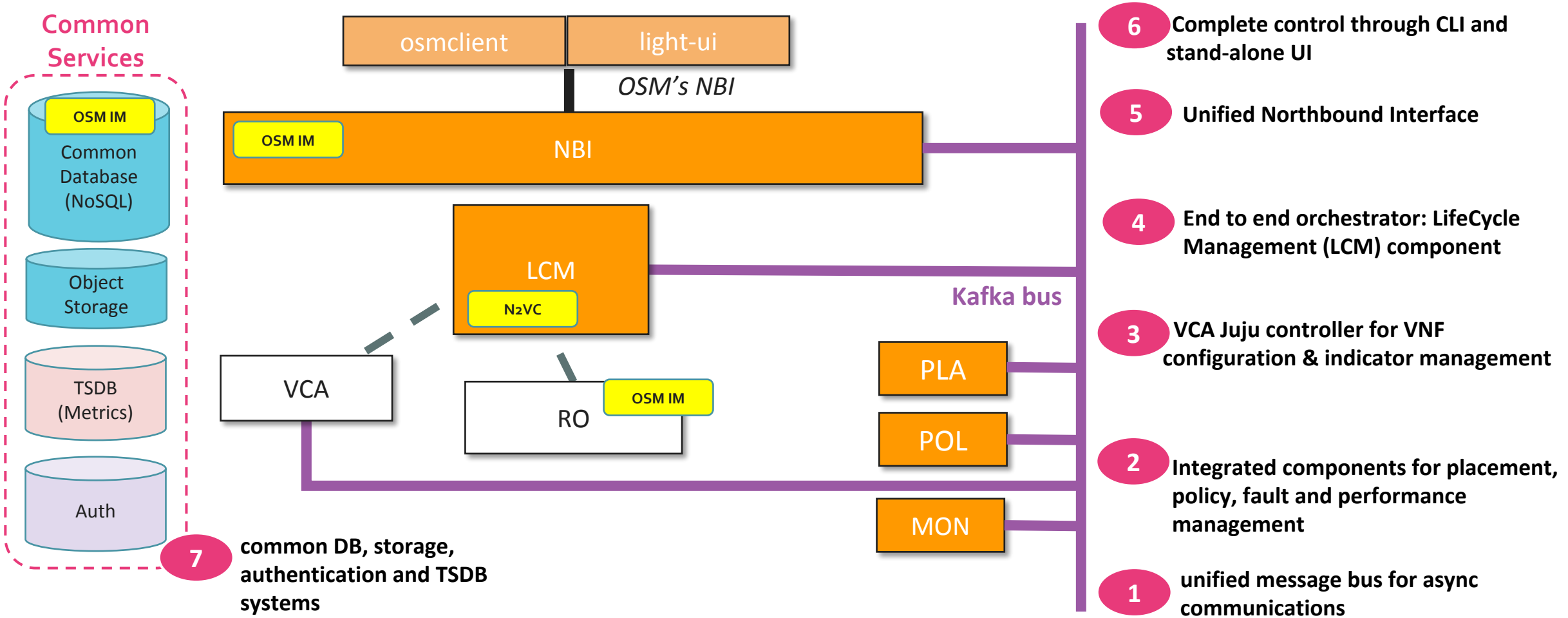
CASE #3: VIM with partial SDN
- Overlay: Native
- Underlay: SDNC

CASE #4: SDN Assist
- Overlay: VIM native
- Underlay: SDNC, via OSM

SDN Assist

Allows OSM to control SDN connectivity, even when not possible by the VIM (eg: PCI Passthrough, SR-IOV)

OSM Architectural view

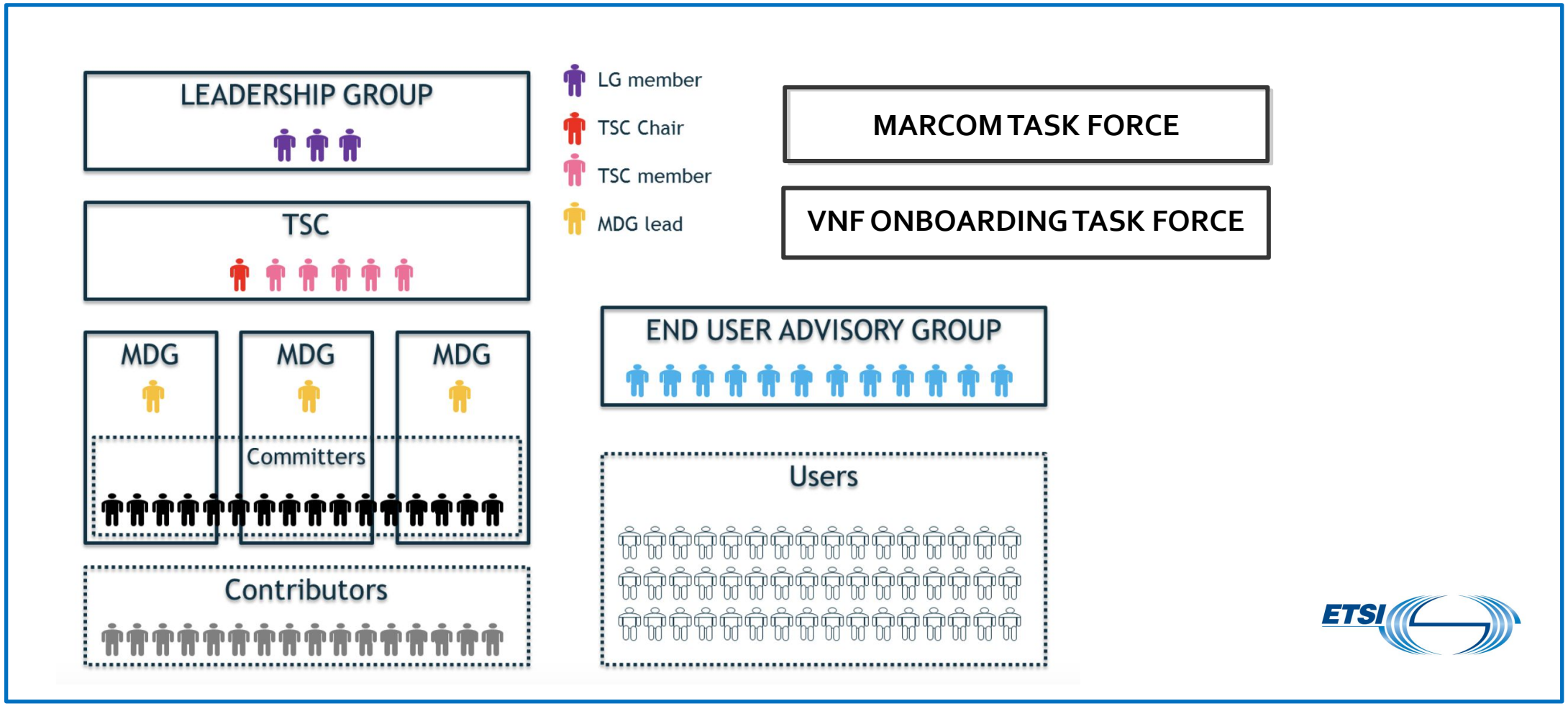


What makes OSM Awesome?

It has a large and diverse community! Around **140** members!

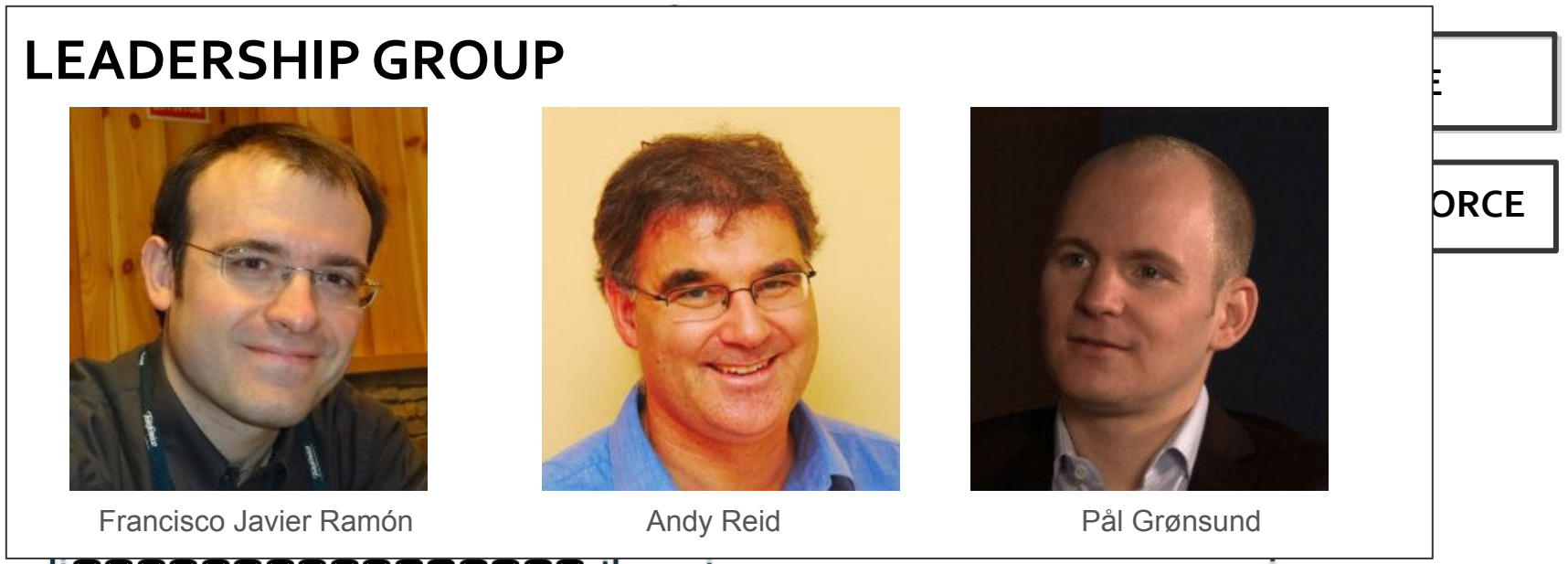
What makes OSM Awesome?

It is well organized for producing production-ready upstream code



What makes OSM Awesome?

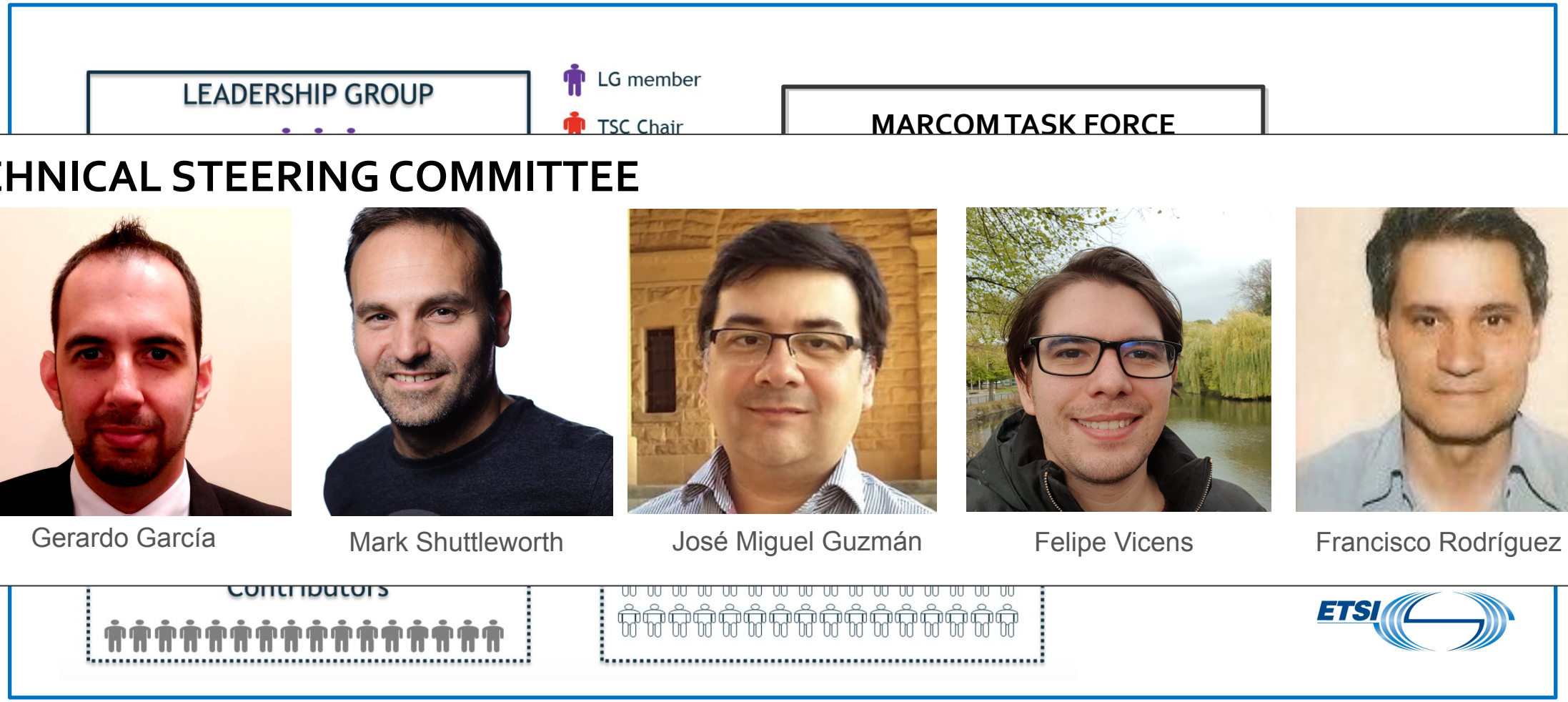
It is well organized for producing production-ready upstream code



What makes OSM Awesome?

Open Source MANO

It is well organized for producing production-ready upstream code



What makes OSM Awesome?

Open Source MANO

It is well organized for producing production-ready upstream code

DEVOPS MDL

Mark Beierl

N2VC MDL

David García

RO MDL

Alfonso Tierno

SA MDL

Subhankar Pal

UI TFL

Francesco Lombardo

MDG MDG MDG

Committers

Contributors

Users

ETSI TE

Silvia Almagia

What makes OSM Awesome?

END USER ADVISORY GROUP

Antonio Marsico

MARCOM TASKFORCE

Mona Hrapkowicz

VNF ONBOARDING TASKFORCE

Gianpietro Lavado

What makes OSM Awesome?

Release SEVEN, launched in December 2019, adds new features ready for production environments:

- Ability to provide real-time feedback in CLI and GUI upon request
- RobotFramework for building automated tests
- Migration of componentes to Python3
- osmclient package creation and validation tool
- Automated monitoring dashboards for system and NFs
- Enhanced VNF Management through full charm support
- Deployment of OSM over kubernetes infrastructure
- **Deployment of containerized NFs over Kubernetes**

What makes OSM Awesome?

And because other people say that OSM Rocks!

Table III: OSM vs ONAP resource footprint comparison.

Resource	OSM-4	ONAP-B
vCPU	2	88
Memory(GB)	8	176
Storage(GB)	40	1760
IP Addresses	1 static	20 Floating 3 static

Scan & Download

This work has been submitted to the IEEE for possible publication.

On the Challenges and KPIs for Benchmarking Open-Source NFV MANO Systems: OSM vs ONAP

Girma M. Yilma, Faqir Zarrar Yousaf, Vincenzo Sciancalepore, Xavier Costa-Perez
NEC Laboratories Europe GmbH, Germany
Email: {girma.yilma|zarrar.yousaf|vincenzo.sciancalepore|xavier.costa}@neclab.eu

***Abstract*—NFV management and orchestration (MANO) systems are being developed to meet the agile and flexible management requirements of virtualized network services in the 5G era and beyond. In this regard, ETSI ISG NFV has specified a standard NFV MANO system that is being used as a reference by MANO system vendors as well as open-source MANO projects. However, in the absence of MANO specific KPIs, it is difficult for users to make an informed decision on the choice of the MANO system better suited to meet their needs. Given the absence of any formal MANO specific KPIs on the basis of which a performance of a MANO system can be quantified, benchmarked and compared, users are left with simply comparing the claimed feature set. It is thus the motivation of this paper to highlight the challenges of testing and validating MANO systems in general, and propose MANO specific KPIs. Based on the proposed KPIs, we analyze and compare the performance of the two most popular open-source MANO projects, namely ONAP and OSM, using a complex open-source vCPE VNF and identify the features/performance gaps. In addition, we also provide a sketch of a test-jig that has been designed for benchmarking MANO systems.**

I. Introduction

AGILITY and flexibility for the management of the network resources and services represents one of the key innovations of 5G networks to support carrier-grade operations for different verticals with diverse service requirements at reduced CAPEX/OPEX costs. In this context, Network Function Virtualization (NFV) has been widely accepted as a technology enabler for addressing the challenging requirements of 5G networks [1]. The key concept of NFV is the decoupling of the network functions from the underlying hardware platforms, while the network functions are realized as a virtualized entity commonly referred to as Virtualized Network Functions (VNFs). VNFs can embody less complex network functions such as Firewall (vFW), load balancer (vLB) to more complex functions such as Evolved Packet Core (vEPC), Customer Premises Equipment (vCPE) to name a few. End-to-end Network Services (NS) are composed by chaining relevant VNFs over Virtual Links (VL).

The introduction of NFV technology has great implications on the network management systems where they need to be extended to provide Life Cycle Management (LCM) of VNFs, NSs and VLs beyond the traditional FCAPS (Fault, Configuration, Accounting, Performance, Security) management services. The LCM actions include operations such as on-boarding, instantiation, scaling in/out/up/down, migration, update/upgrade, etc of a VNF and its associated components. In this regard the ETSI ISG NFV has proposed NFV Management and Orchestration (MANO) system [2] and has specified interfaces and operations on its various reference points to support different functional features in its various specification documents. Fig. 1 provides a high level overview of the ETSI NFV MANO system functional blocks and the various interfaces defined on the reference points. The ETSI NFV MANO framework is also serving as a reference to other independent MANO projects that are being undertaken either by vendors or by open source communities. The latter is gaining a lot of prominence and attention from operators due to the diverse efforts that are being expended towards developing open source MANO platforms.

A. Problem Statement

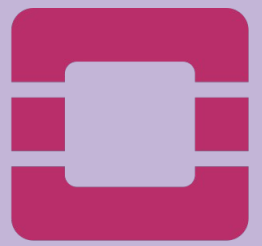
Open source MANO projects such as ONAP [3], OSM [4], Open Baton [5], Cloudify [6], OPNFV [7], are under different stages of steady development. All are competing to make their mark in the operators' infrastructure but, owing to the complex nature of the NFV MANO system itself, no project to date can claim to support the entire LCM spectrum of the NFV assets or be ready for field operations. More prominent among these projects are Open Network Automation Platform (ONAP) and Open Source MANO (OSM), which have gained a lot of attention from the operators' community, especially because of the patronage of some big operators behind the development of ONAP and OSM. For instance ONAP, which is being developed under the umbrella of the Linux Foundation, is mainly supported by AT&T, whereas OSM is driven by Telefonica and is being developed under the mandate of the newly formed ETSI Open Source Group (OSG).

Both ONAP and OSM are under different stages of their releases but they are far from being complete or stable. Both are aiming to provide an integrated NFV MANO framework, but they are following very different directions in terms of architecture and implementation. *There are still gaps between what is being claimed and what features and functionalities are actually supported.* There are ambiguities in terms of their deployment footprint as well as operational efficiency for providing carrier-grade management to NFV services. Owing to the fact that these are relatively latest developments, there is very much less information and experience available in terms of the functional and operational capabilities of these platforms and technology readiness level (TRL).

Moreover, carrying out ... MANO system ...

Open Source
MANO

OpenStack & Kubernetes primer



openstack®

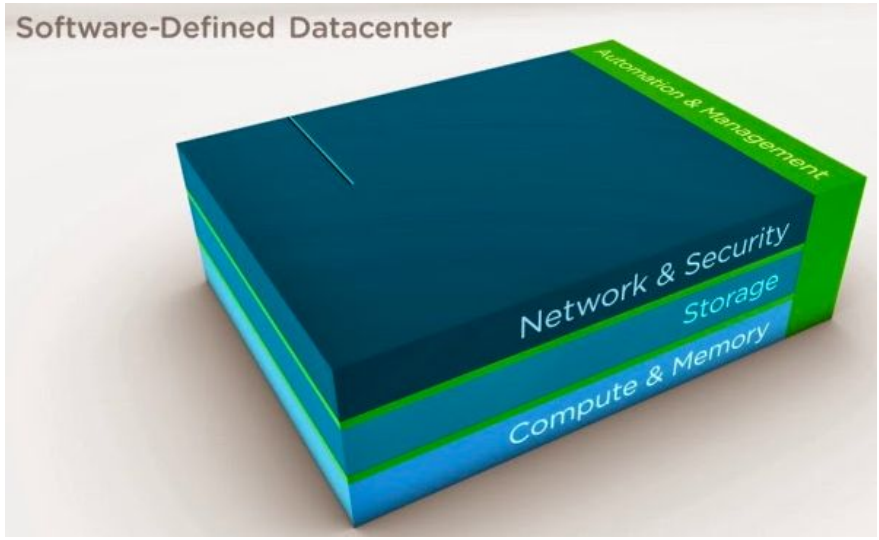
kubernetes

The Software-Defined DataCenter

Software defined data centers are composed of completely virtualized infrastructure, that can be easily managed using software.
When talking about SDDC, we are mainly referencing **virtualization of compute, storage and networking**, being that all the infrastructure is totally programmable.

This new data center building paradigm focuses on:

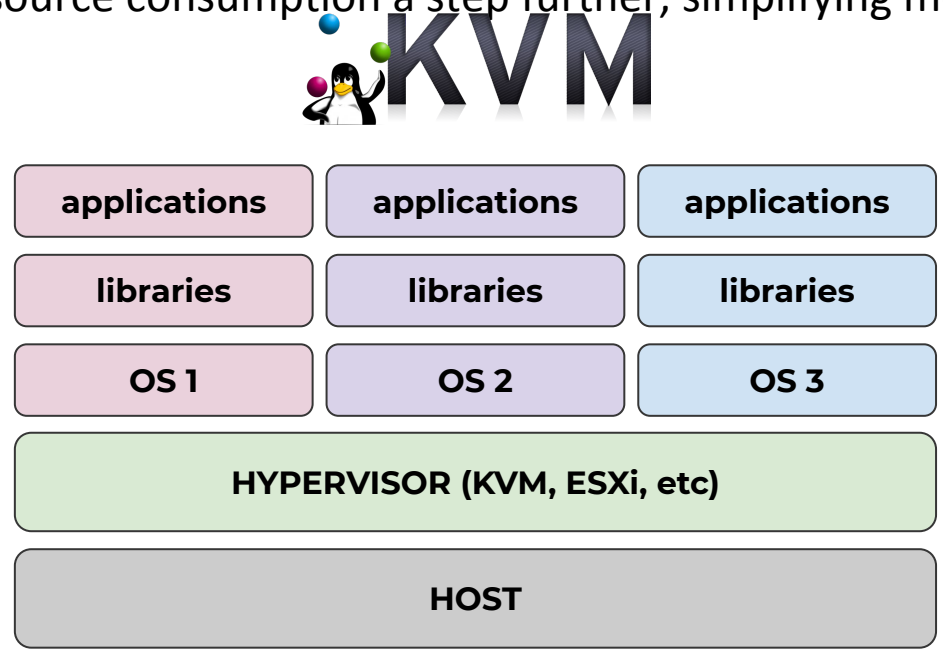
- Using the full potential of hyper scalable architectures
- More agility and faster application provisioning
- Cost reduction
- Automated management (elastic and programmable)

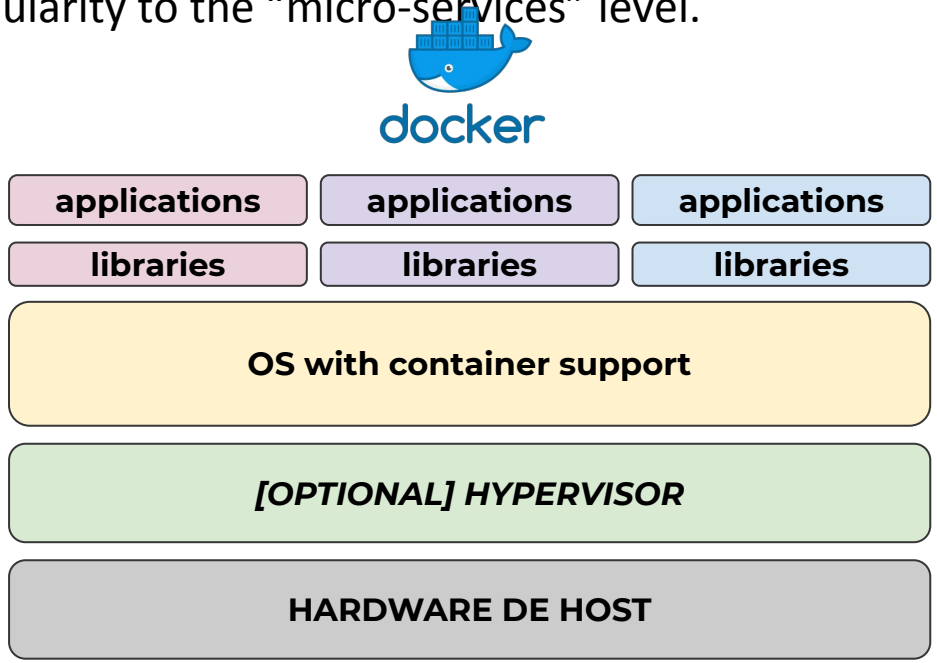


Virtual Machines and Containers

With hypervisors, like KVM, we are capable of partitioning a physical compute node into multiple **“virtual machines”** that use their own Operating System to share the physical resources, providing efficiency on the host resources consumption.

“Containers”, made popular by Docker, produce instances that share a single Operating System while only adding the libraries they need to run a lightweight application. This concept increases application mobility and takes the efficiency on host resource consumption a step further, simplifying modularity to the “micro-services” level.

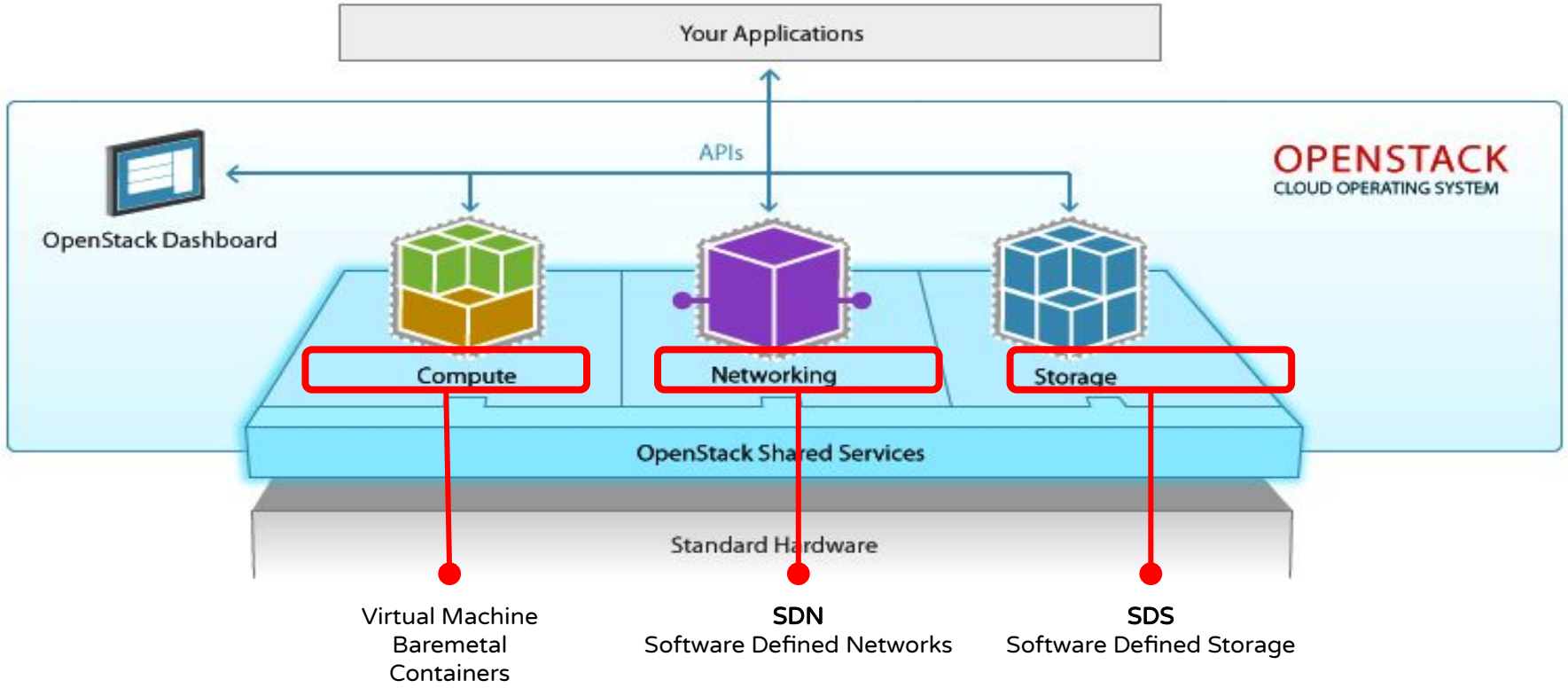




OpenStack

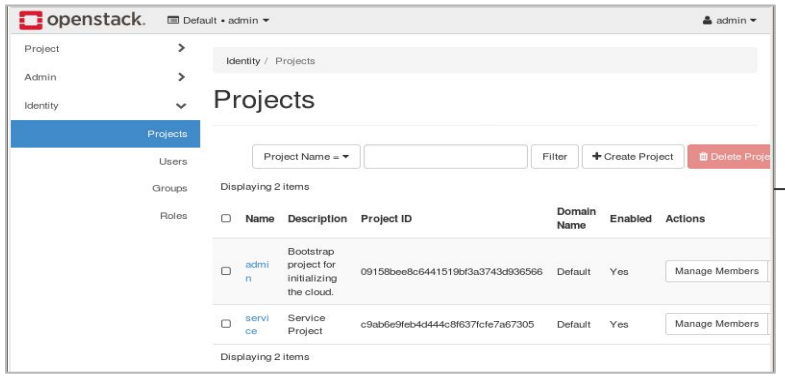
OpenStack is an **operating system for the cloud** which controls large amounts of computing, storage and networking resources of a data center in a centralized and simple way.

It provides the essential to build an Infrastructure-as-a-Service platform (IaaS).



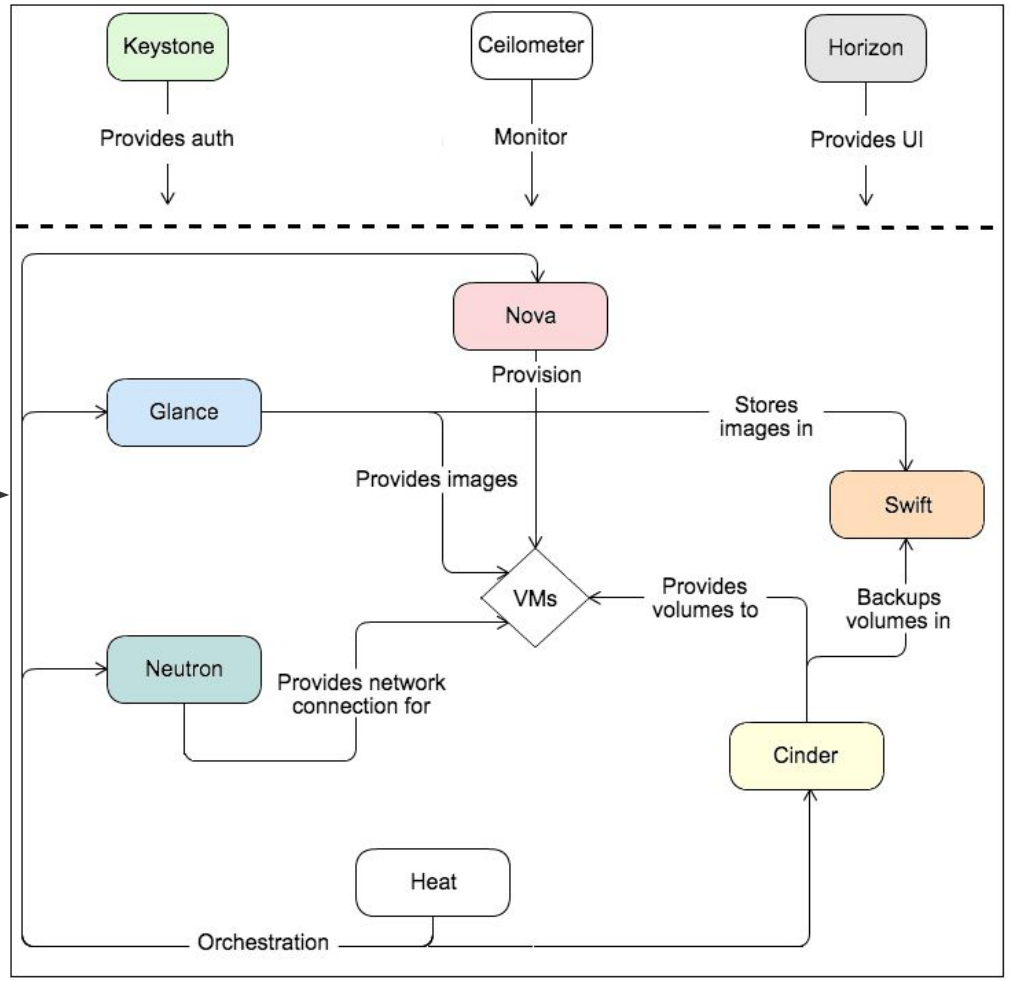
OpenStack

Even though it is capable of providing containers, **it highly specializes in automating the lifecycle and resources of virtual machines.**



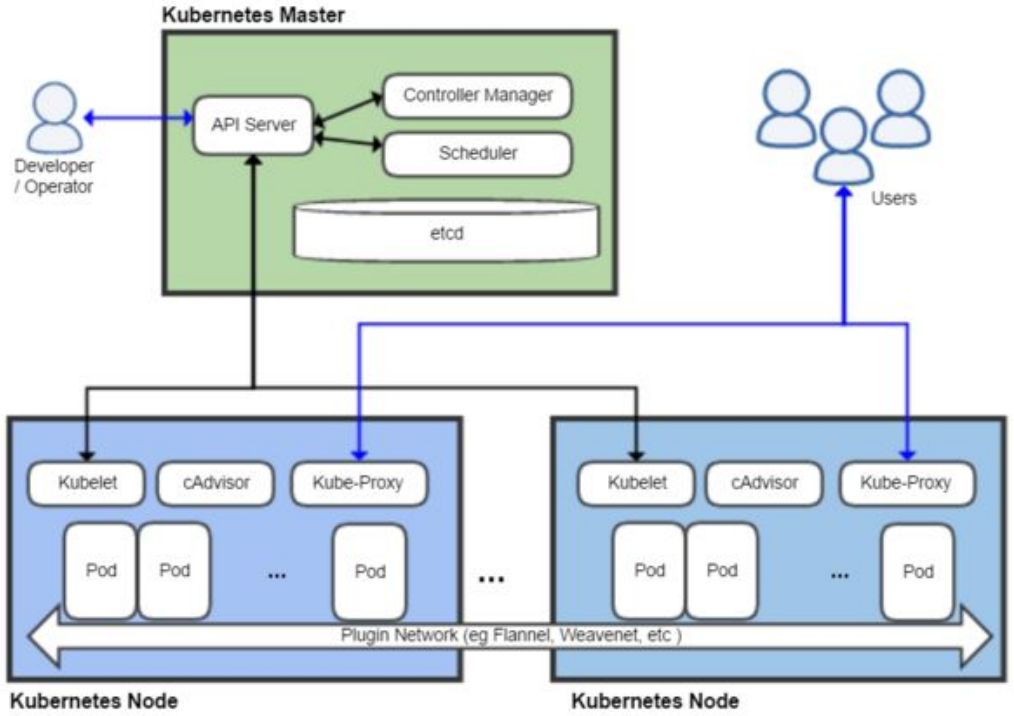
Open Source MANO

REST API



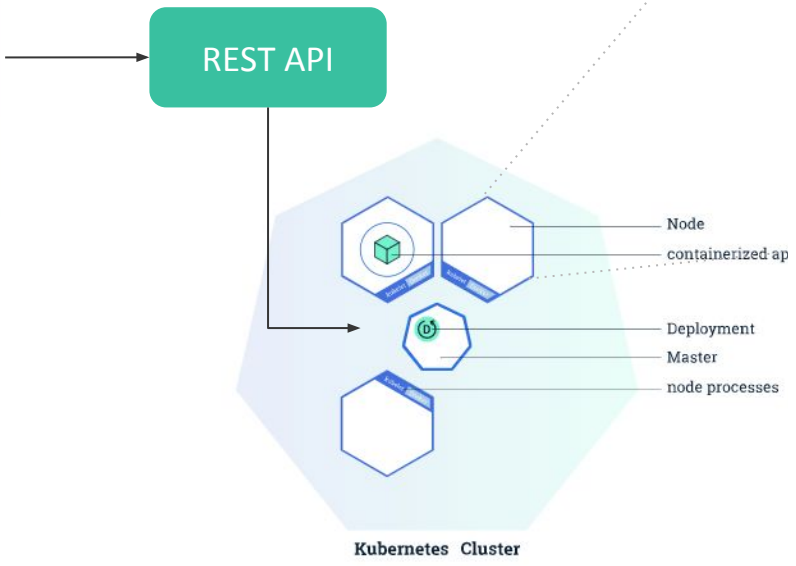
Kubernetes

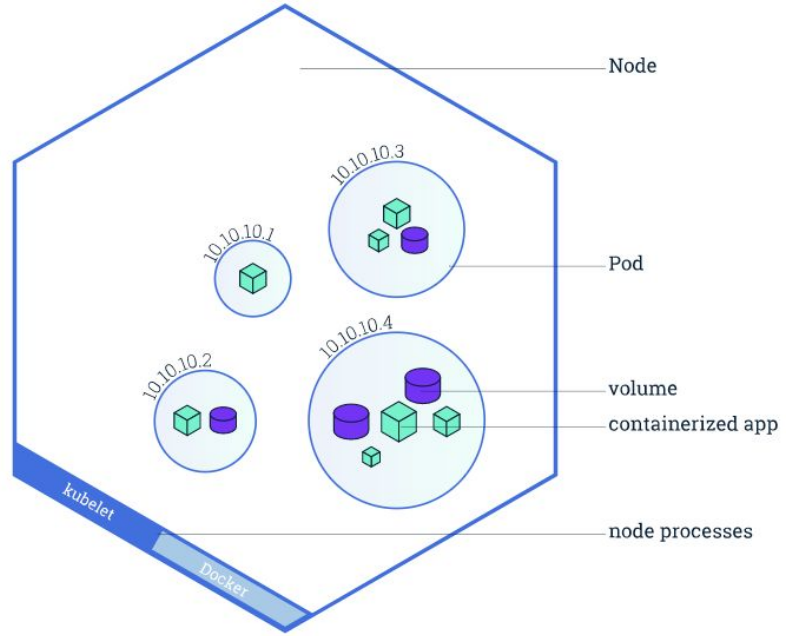
“Kubernetes” is a Container Orchestration Engine (CoE) that manages the life cycle of docker containers, including how they are grouped together to form applications (via “pods”), how they interconnect (via “overlay” networks), how the services are balanced, protected and scaled.

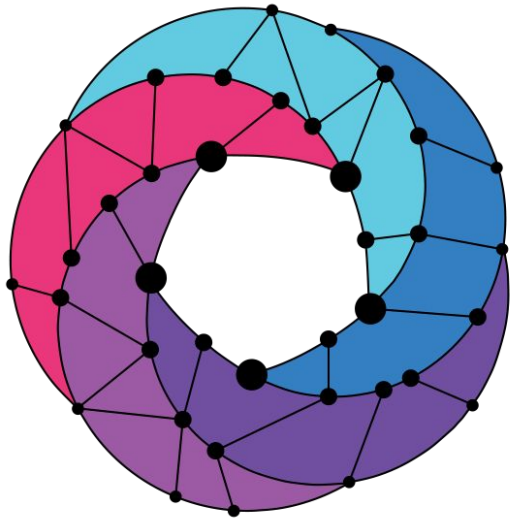


Kubernetes

Even though it is capable of providing virtual machines, **it highly specializes in automating the lifecycle and resources of containers**.







Open Source
MANO

Thank you!

ETSI