4.

Network Function Virtualization (NFV)

- What is NFV ?
- How NFV contributes to 5G ?
- What are the main concepts of NFV ?
- What are the benefits/limitations of NFV ?
- NFV and Slicing

Concepts

NFV (Network Functions Virtualization)

NFV is a network architecture concept that uses virtualization technologies to manage and deploy network services.

VNF (Virtual Network Function)

A VNF is a specific implementation of a network function that runs in a virtualized environment.

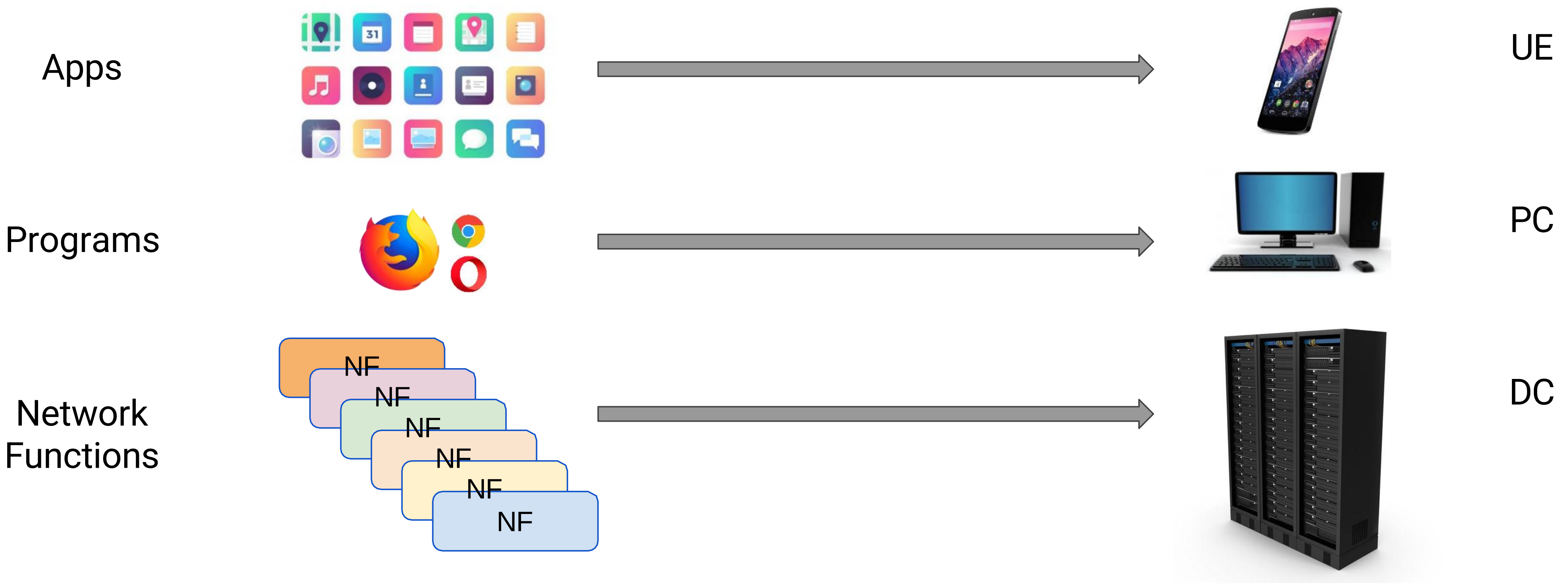
CNF (Cloud-native Network Function)

CNF refers to network functions that are designed specifically for cloud-native environments, utilizing microservices architecture and containerization.

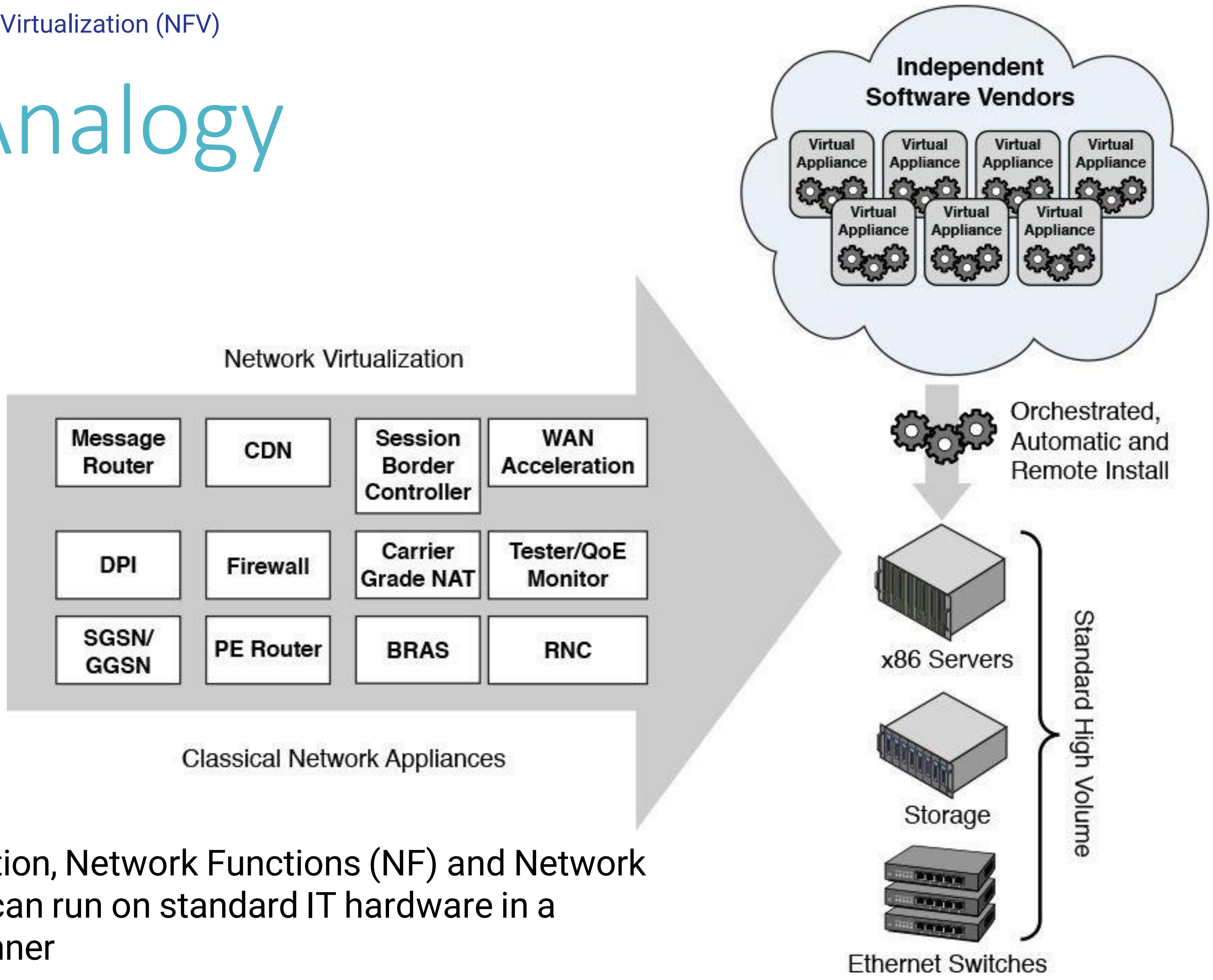
NFV Analogy

Software Applications
Resources **Sharing**
Multiple Vendors

Minimal Software (e.g. OS) + **Hardware**
Compute (CPU, RAM)
Storage (e.g. HDD/SSD)
Network (e.g. Ethernet NIC, Fiber Optic NIC)



NFV Analogy



With Virtualization, Network Functions (NF) and Network Services (NS) can run on standard IT hardware in a virtualized manner

Enablers for Network Functions Virtualisation

Cloud Computing

- **Cloud technologies are mainly virtualisation mechanisms**: hardware virtualisation by means of **hypervisors**, as well as the usage of **virtual ethernet switches** (e.g. OVS) for connecting traffic between virtual machines and physical interfaces

- For communication-oriented VNFs, **high-performance packet processing** is available through
  - high-speed multi-core CPUs with high I/O bandwidth
  - smart Ethernet NICs for load sharing and TCP offloading
  - routing packets directly to VM memory

- Cloud infrastructures provide methods to enhance resource availability and usage by means of **orchestration**, **automation** and **management mechanisms**

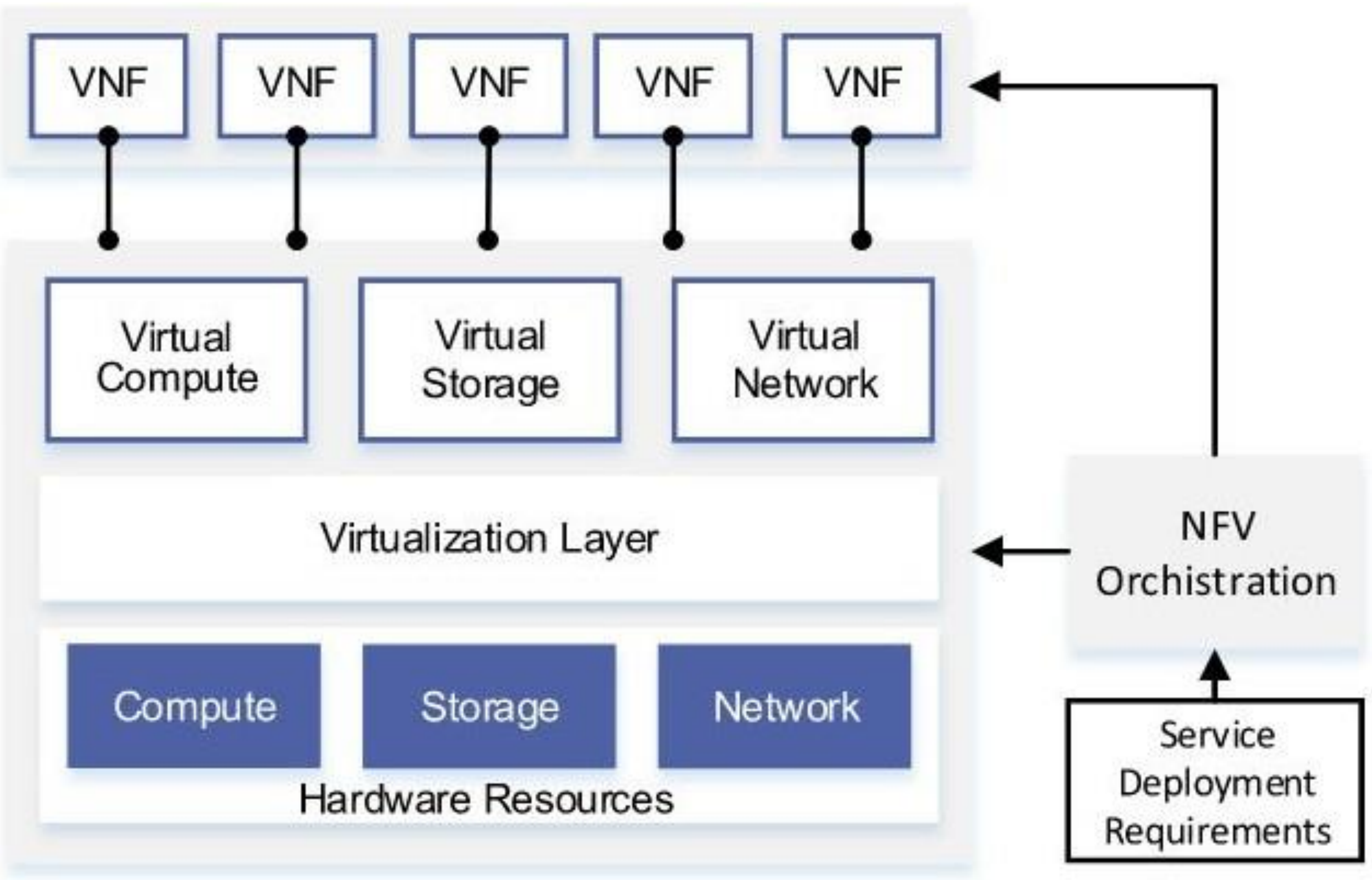
- **Open APIs** for management and data plane control, provide an additional degree of integration of NFV and cloud infrastructure

Enablers for Network Functions Virtualisation

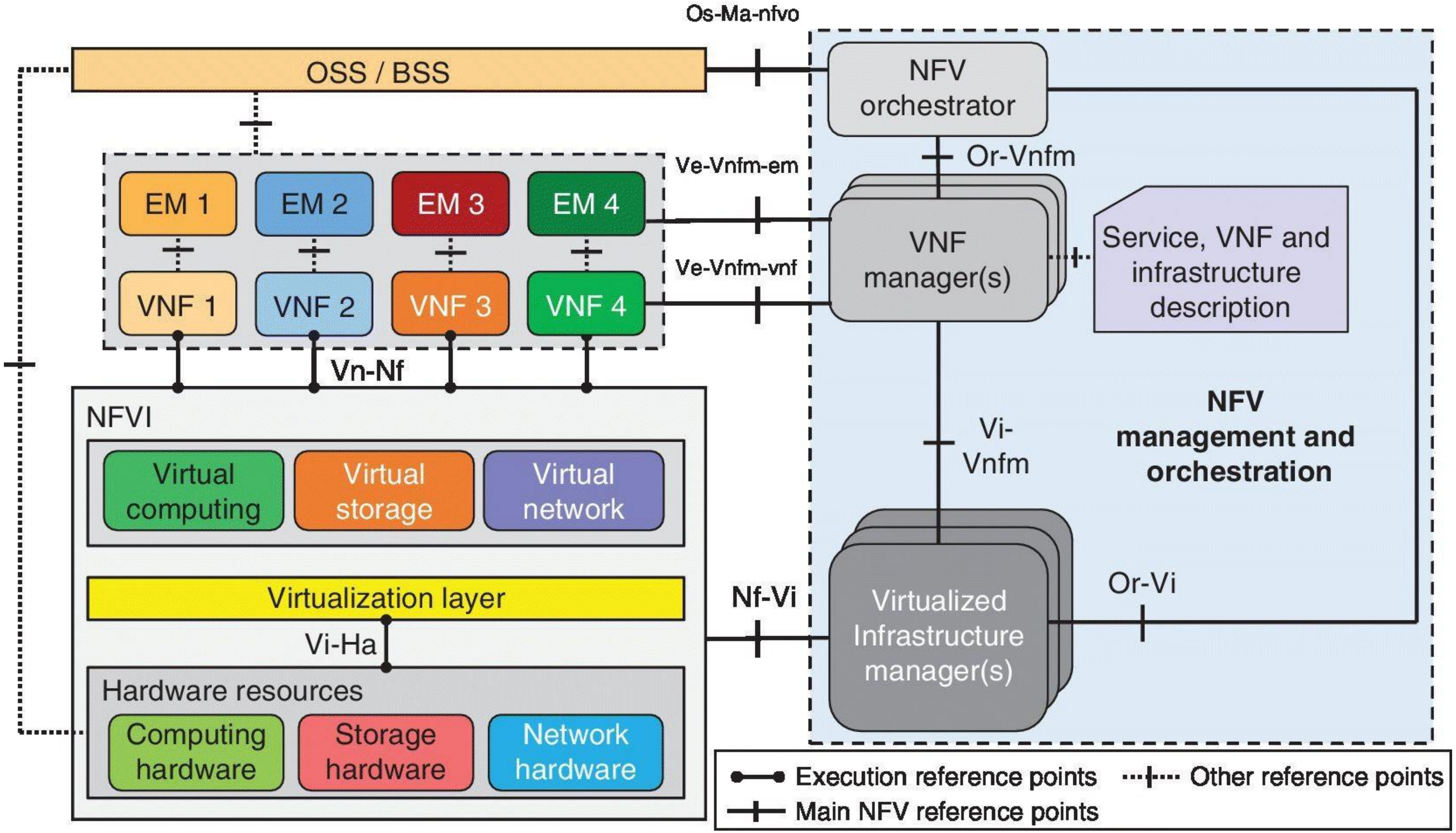
Industry Standard High Volume Servers & Commercial off-the-shelf (COTS)

- These are key elements in the economic case for NFV to leverages the economies of scale of the IT industry
- COTS hardware built using standardised IT components (e.g. x86 architecture, generic FPGA/ASIC)
- Standard data center requirements for both space and power

Simplified NFV Architecture



ETSI NFV Architecture [14]



ETSI NFV architecture

- **Virtualized network function (VNF)**: Virtualized instance of an NF traditionally implemented on a physical network appliance
- **Element management (EM)**: Component performing the typical network management functions (Fault, Configuration, Accounting, Performance and Security - FCAPS) requested by the running VNFs
- **NFV infrastructure (NFVI)**: Set of hardware/software components building up the environment in which VNFs are deployed, managed and executed. Can span across several locations (physical places where NFVI-PoPs are operated)
- **Virtualized infrastructure manager (VIM)**: Provides the functionalities to control and manage the interaction of a VNF with hardware resources under its authority, as well as their virtualization. Typical examples are cloud platforms (e.g., OpenStack) and SDN controllers (e.g., OpenDaylight)

ETSI NFV architecture

- **Resources**: Physical resources (e.g., computing, storage, and network) and Virtualization layer
- **NFV orchestrator (NFVO)**: Component in charge of orchestration and management of NFVI and software resources, and provisioning of network services on the NFVI
- **VNF manager (VNFM)**: Component responsible for VNF lifecycle management (e.g., instantiation, update, query, scaling, and termination). Can be 1-1 or 1-multi with VNFs.

ETSI NFV-MANO

- **Resources**: Physical resources (e.g., computing, storage, and network) and Virtualization layer
- **NFV orchestrator (NFVO)**: Component in charge of orchestration and management of NFVI and software resources, and provisioning of network services on the NFVI
- **VNF manager (VNFM)**: Component responsible for VNF lifecycle management (e.g., instantiation, update, query, scaling, and termination). Can be 1-1 or 1-multi with VNFs.

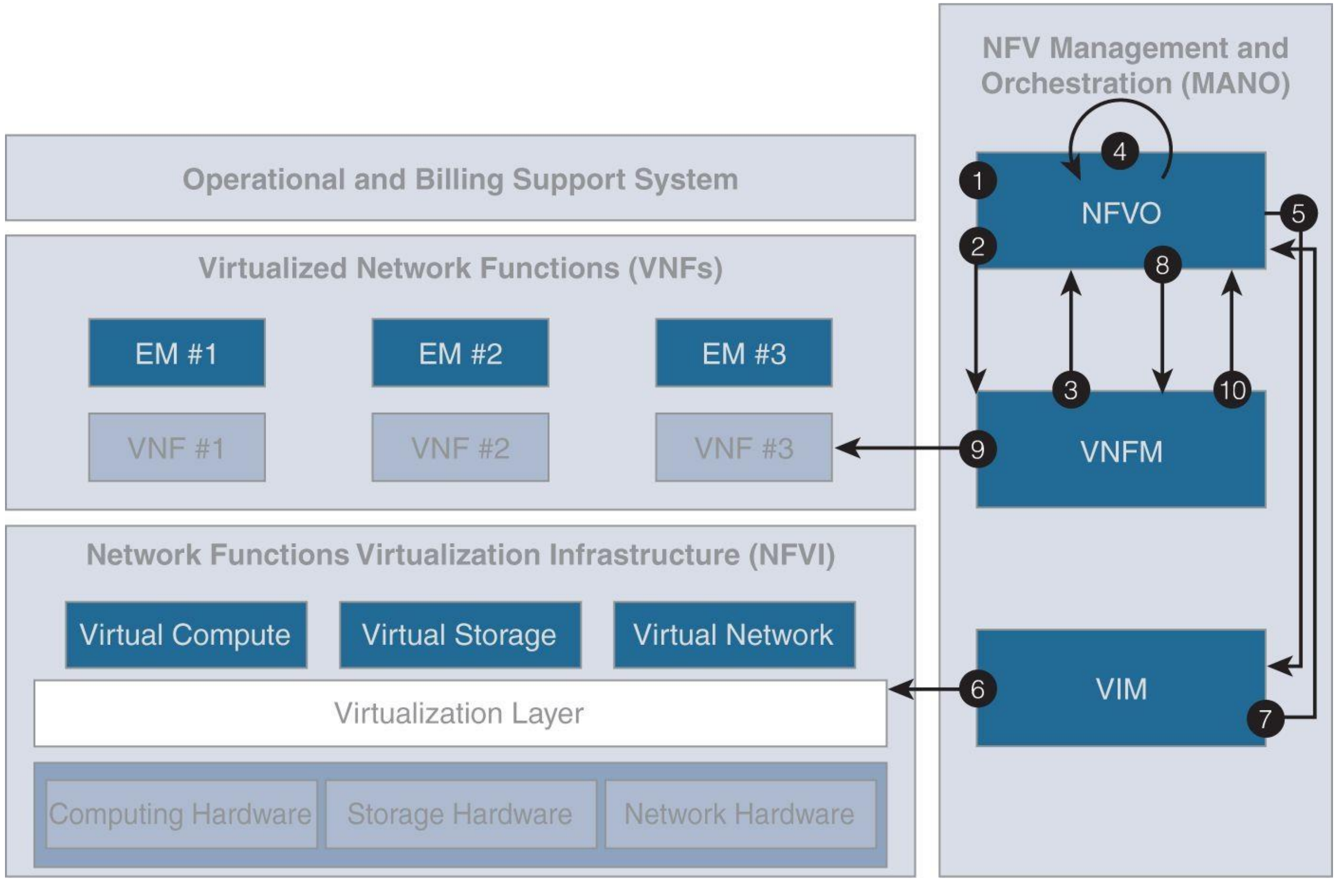
ETSI NFV-MANO

ETSI Open Source MANO (OSM) [15]

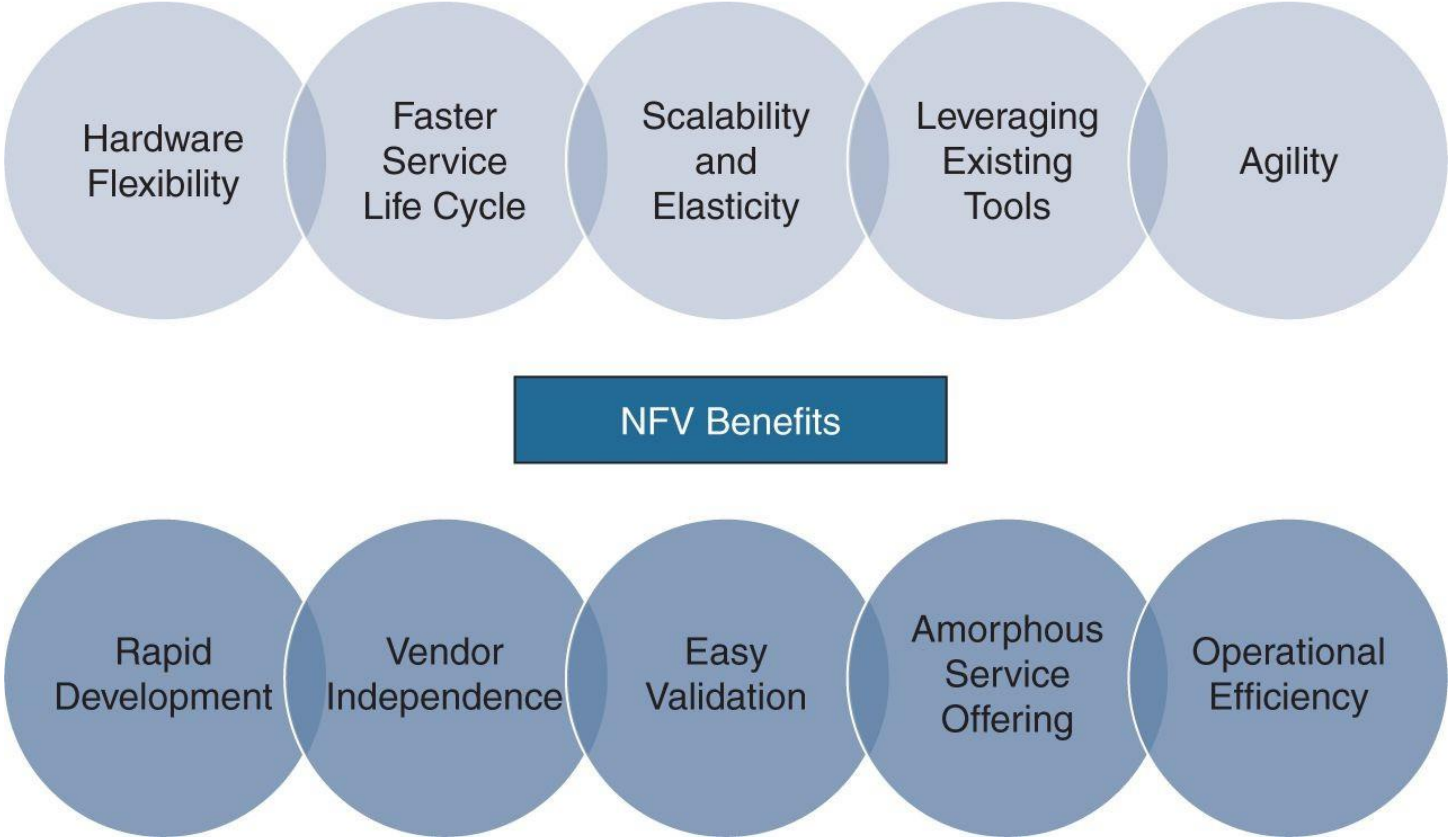
- Open Source Mano (OSM) initiative is launched in 2016 by ETSI
- OSM intends to develop an open-source NFV Management and Orchestration (MANO) software stack aligned with ETSI NFV specifications
- Facilitates the implementation of NFV architectures aligned to ETSI NFV specifications
- Ensures the interoperability among NFV implementations

End-to-End Flow in the ETSI NFV Framework [13]

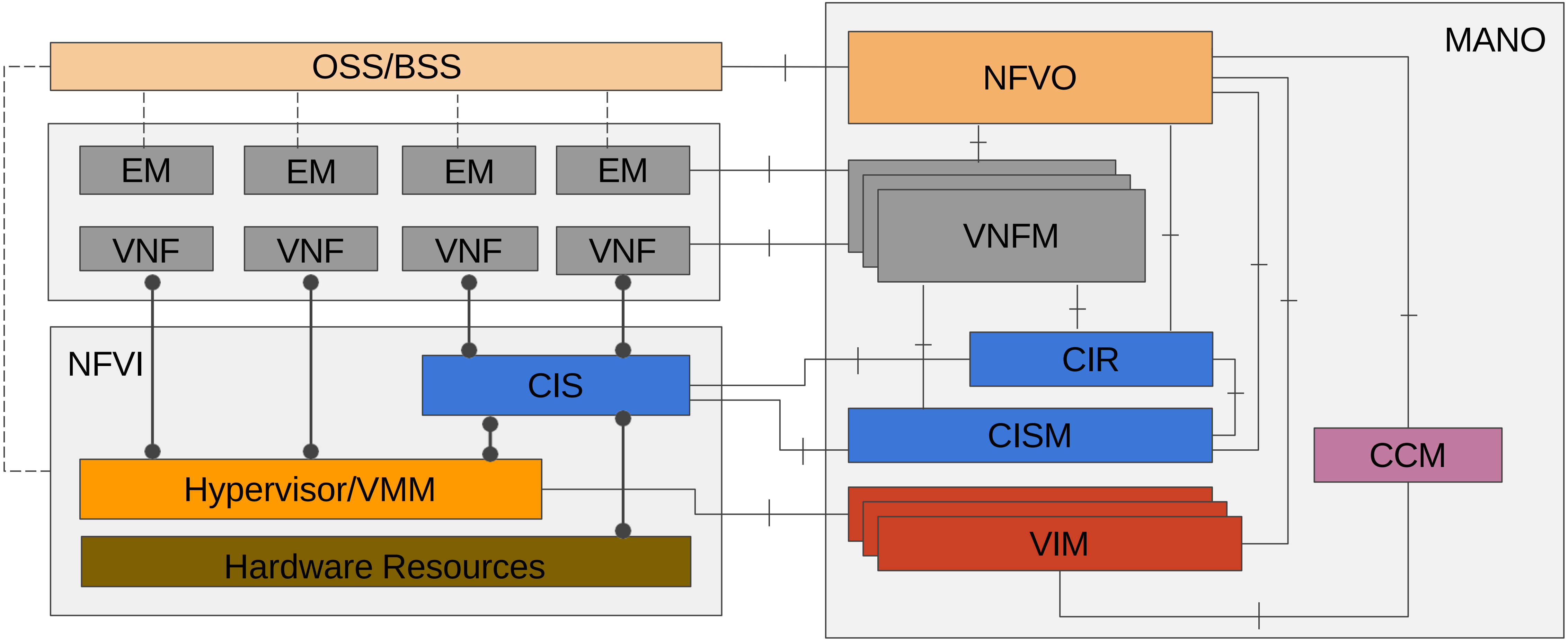
- **Step 1.** The full view of the end-of-end topology is visible to the NFVO
- **Step 2.** The NFVO instantiates the required VNFs and communicate this to the VNFM
- **Step 3.** VNFM determines the number of VMs needed as well as the resources that each of these will need and reverts back to NFVO with this requirement to be able to fulfill the VNF creation
- **Step 4.** Because NFVO has information about the hardware resources, it validates if there are enough resources available for the VMs to be created. The NFVO now needs to initiate a request to have these VMs created
- **Step 5.** NFVO sends request to VIM to create the VMs and allocate the necessary resources to those VMs
- **Step 6.** VIM asks the virtualization layer to create these VMs.
- **Step 7.** Once the VMs are successfully created, VIM acknowledges this back to NFVO
- **Step 8.** NFVO notifies VNFM that the VMs it needs are available to bring up the VNFs
- **Step 9.** VNFM now configures the VNFs with any specific parameters.
- **Step 10.** Upon successful configuration of the VNFs, VNFM communicates to NFVO that the VNFs are ready, configured, and available to use



Benefits of Network Functions Virtualization

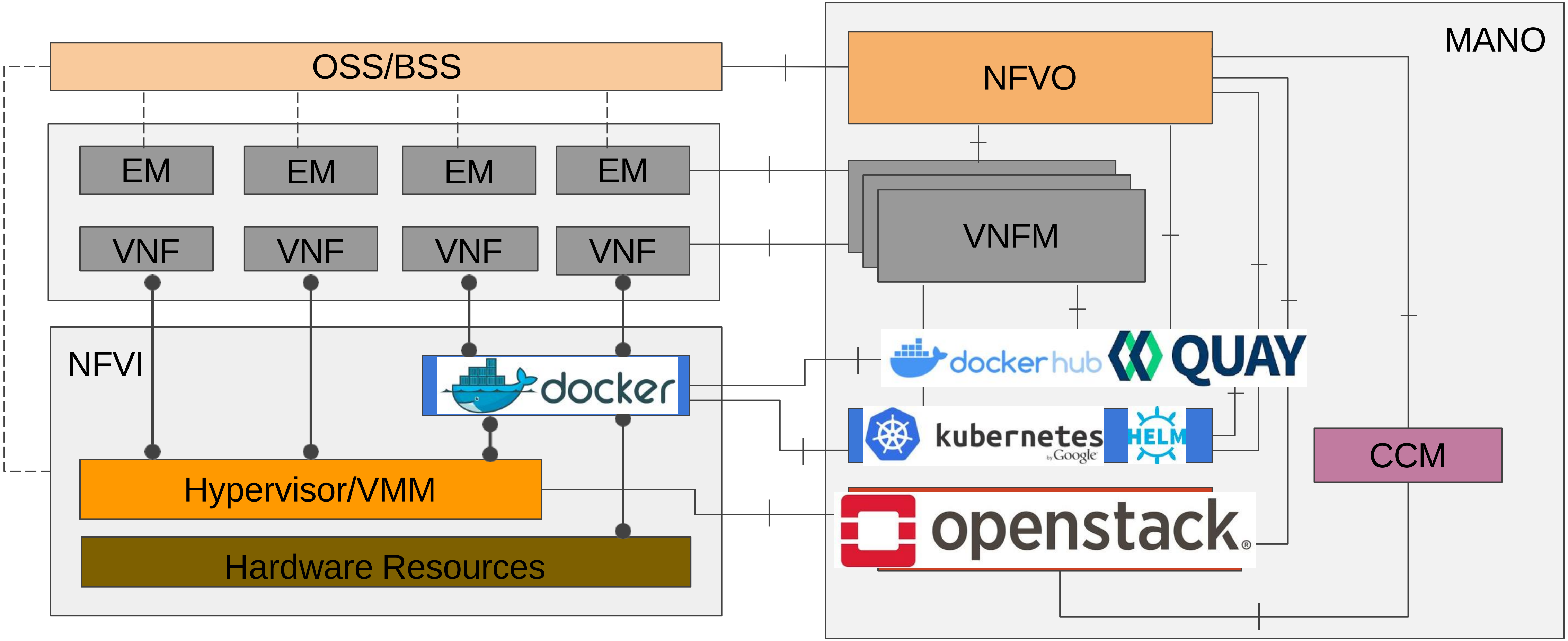


OS Container introduction in ETSI NFV Release 4



Container Infrastructure Service Management (CISM)
Container Infrastructure Service (CIS)
Container Image Repository (CIR)
Container Cluster Management (CCM)

Open Source to Build & Run NFV Infrastructure



Container Infrastructure Service Management (CISM)
Container Infrastructure Service (CIS)

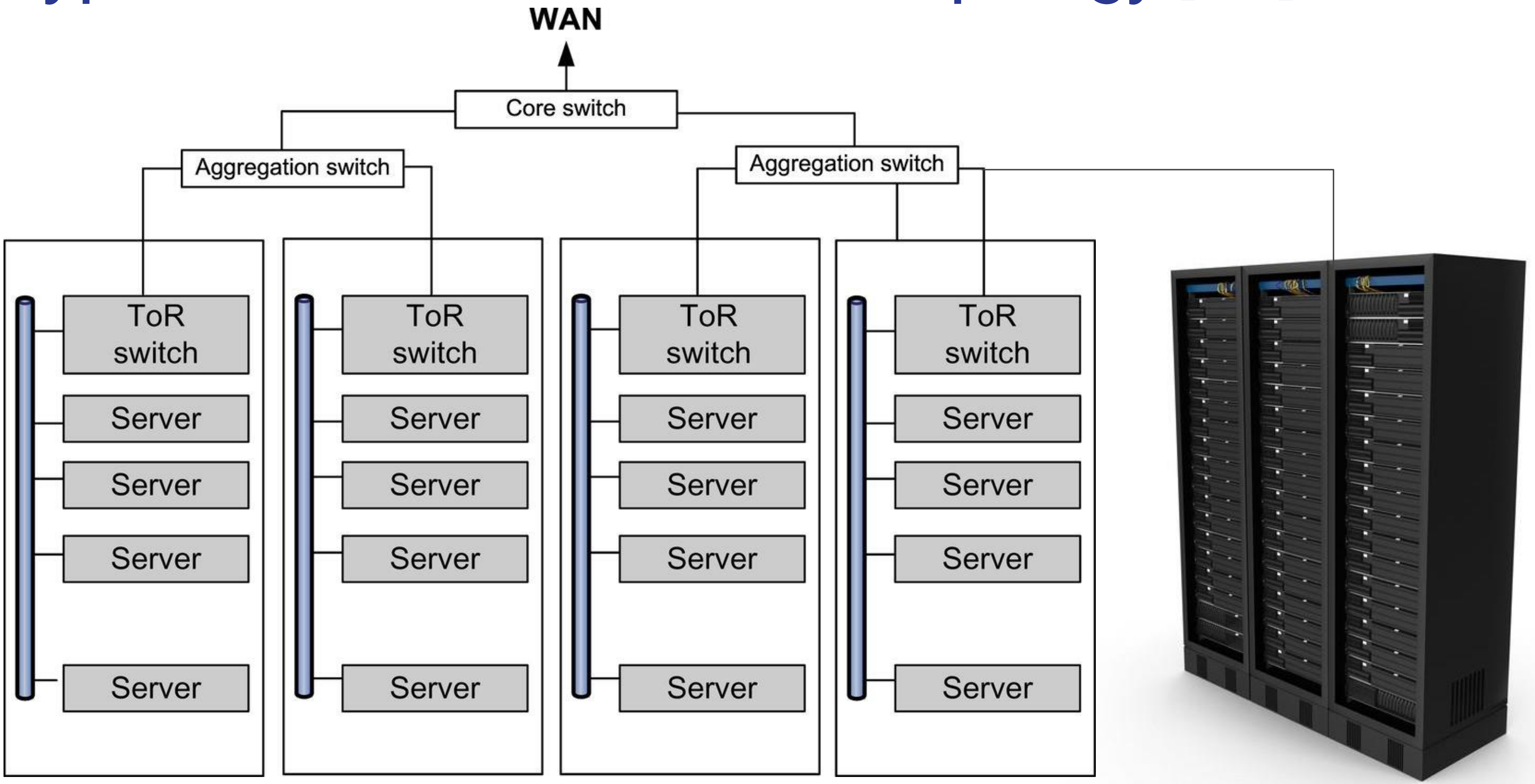
Container Image Repository (CIR)
Container Cluster Management (CCM)

4.

Software Defined Networking (SDN)

- What is SDN?
- SDN Implementation and Protocols
- SDN Use-Cases for Different Networking Domains (SDN for DC, SD-WAN)
- Service Function Chaining
- Performance Consideration (DPDK, SRIOV, VPP)

Typical data center network topology [12]



Typical data center network topology [12]

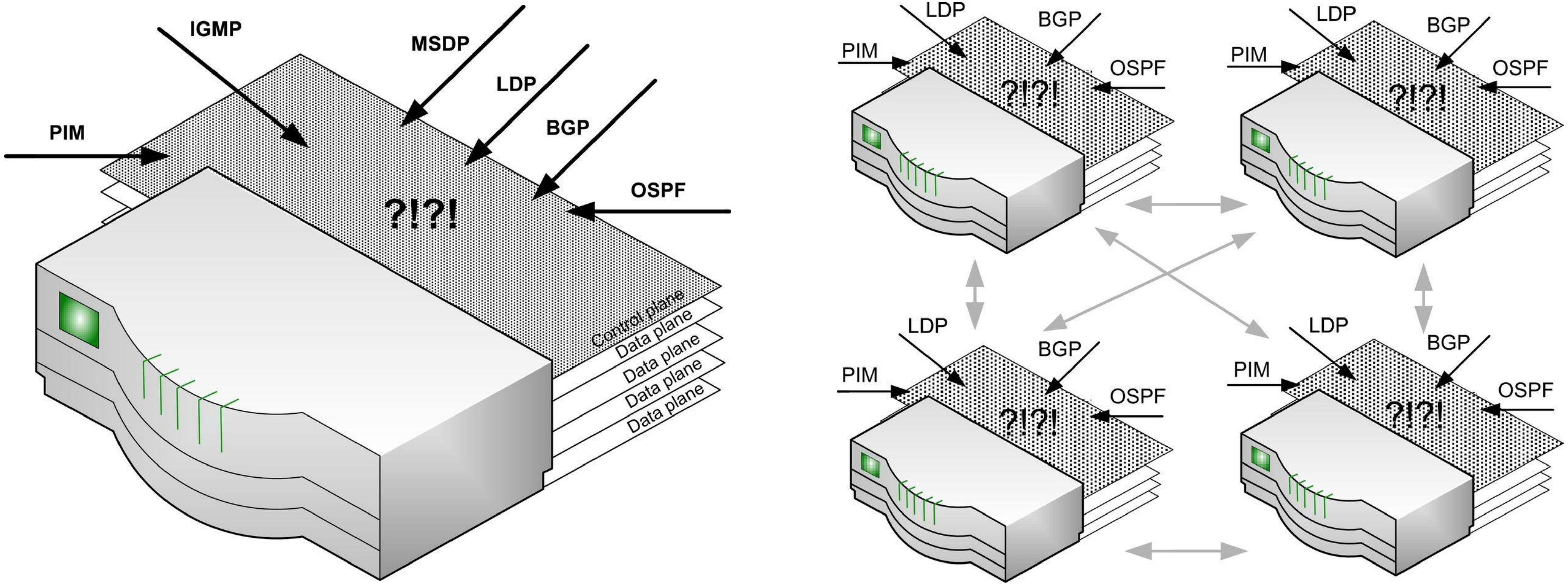
If we consider

- 120,000 Physical Server per DC
- 20 VM per server

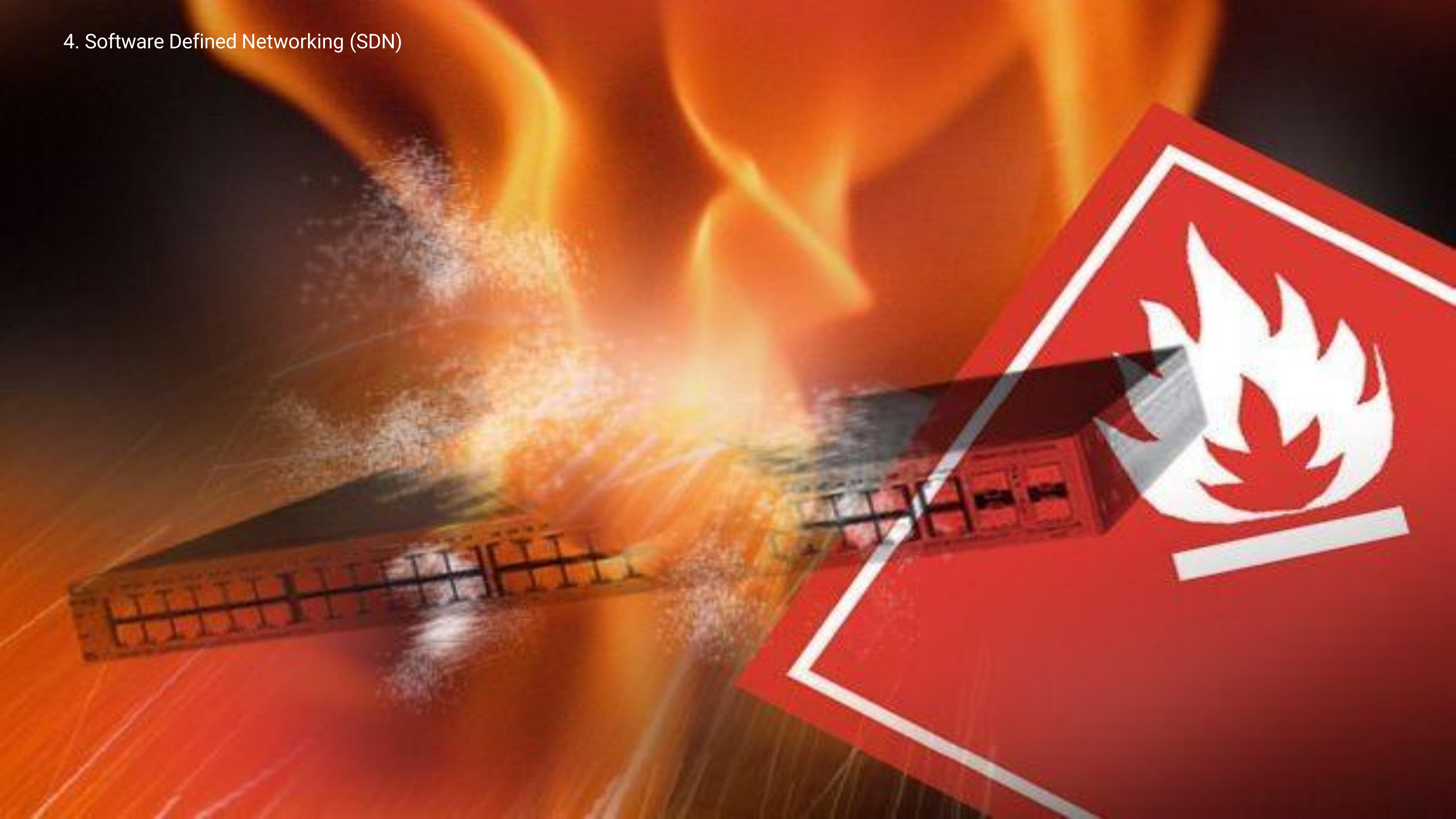
Total number of VMs per DC

2,400,000

Overhead of dynamic distributed route computation [12]



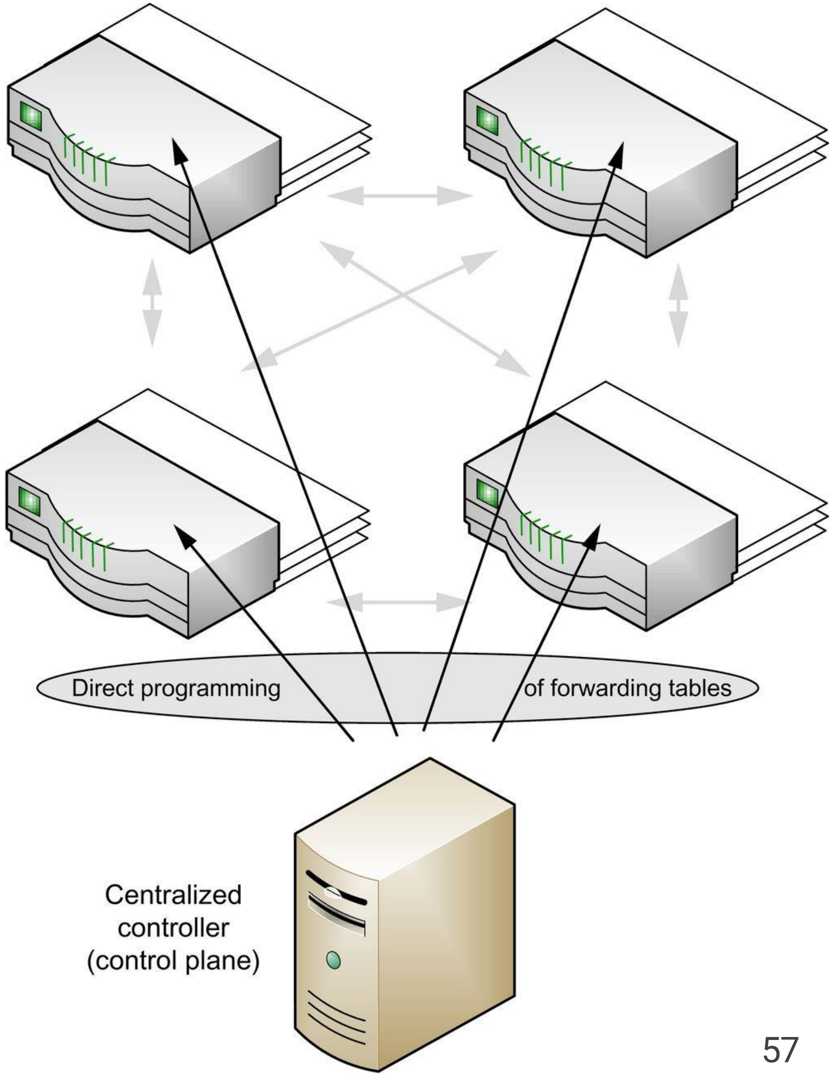
4. Software Defined Networking (SDN)



SDN Definition

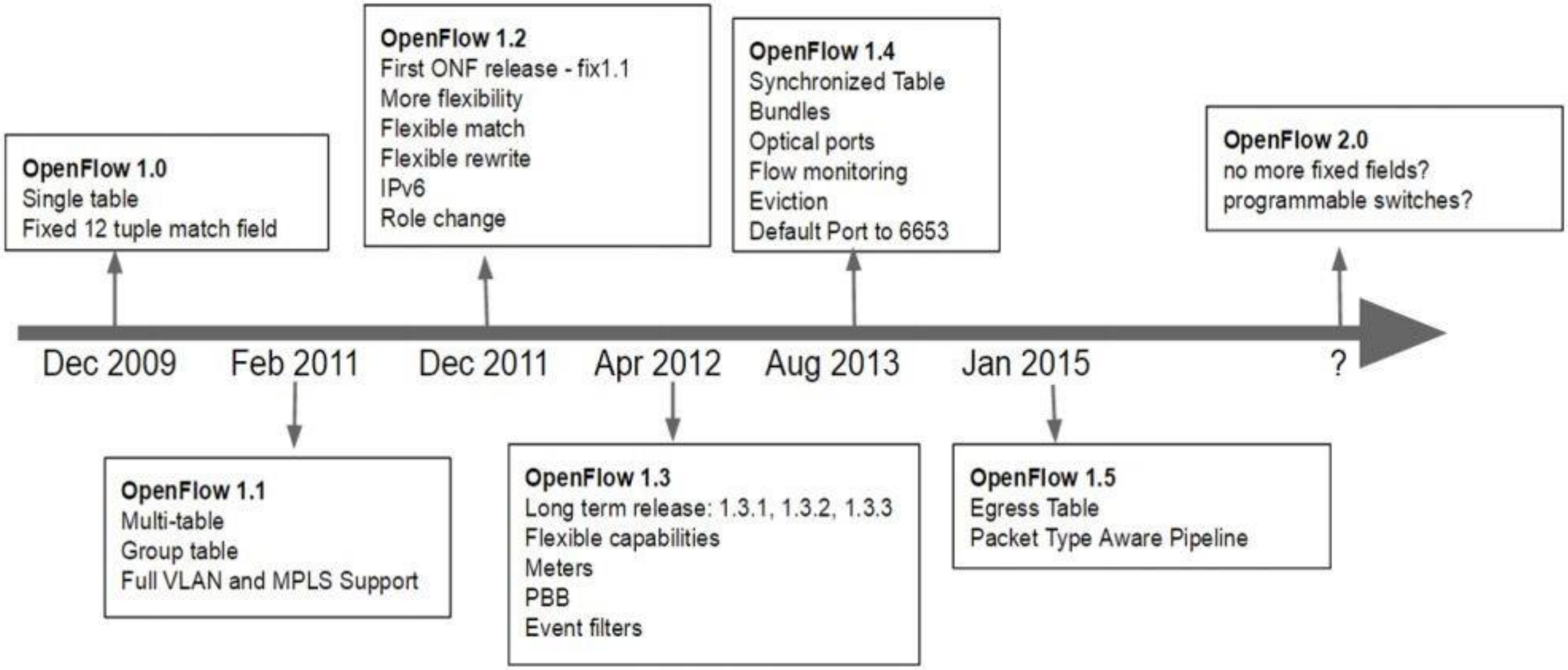
The **physical separation** of the network **control plane** from the **forwarding/data plane**, and where a control plane controls several devices

- **Directly programmable**: Network control is directly programmable because it is decoupled from forwarding functions
- **Agile**: Abstracting control from forwarding lets administrators dynamically adjust network-wide traffic flow to meet changing needs
- **Centrally managed (originally)**: Network intelligence is (logically) centralized in software-based SDN controllers that maintain a global view of the network
- **Programmatically configured**: SDN lets network managers configure, manage, secure, and optimize network resources very quickly via dynamic, automated SDN programs
- **Open standards-based and vendor-neutral**: When implemented through open standards, SDN simplifies network design and operation because instructions are provided by SDN controllers instead of multiple, vendor-specific devices and protocols.

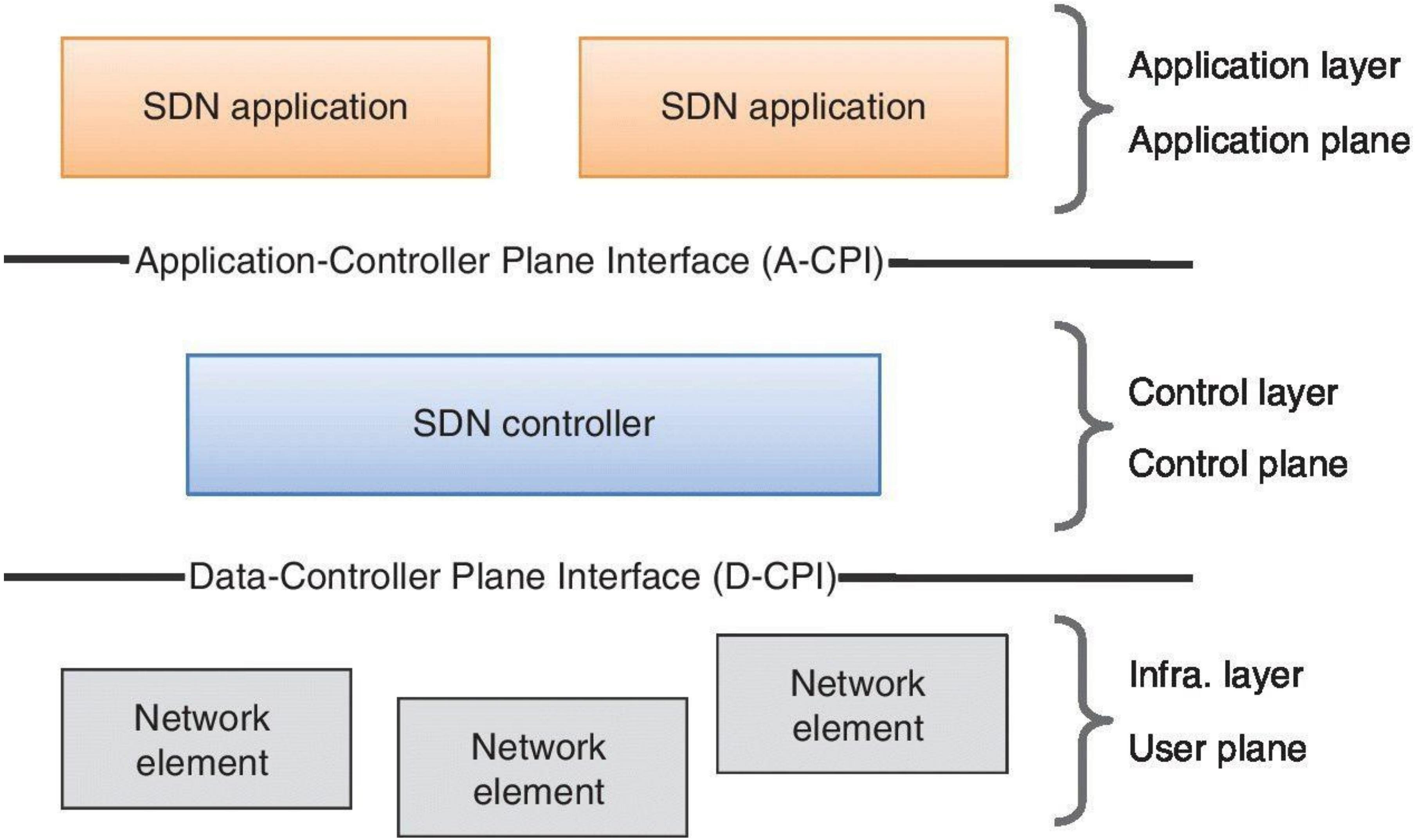


OpenFlow Specification Timeline

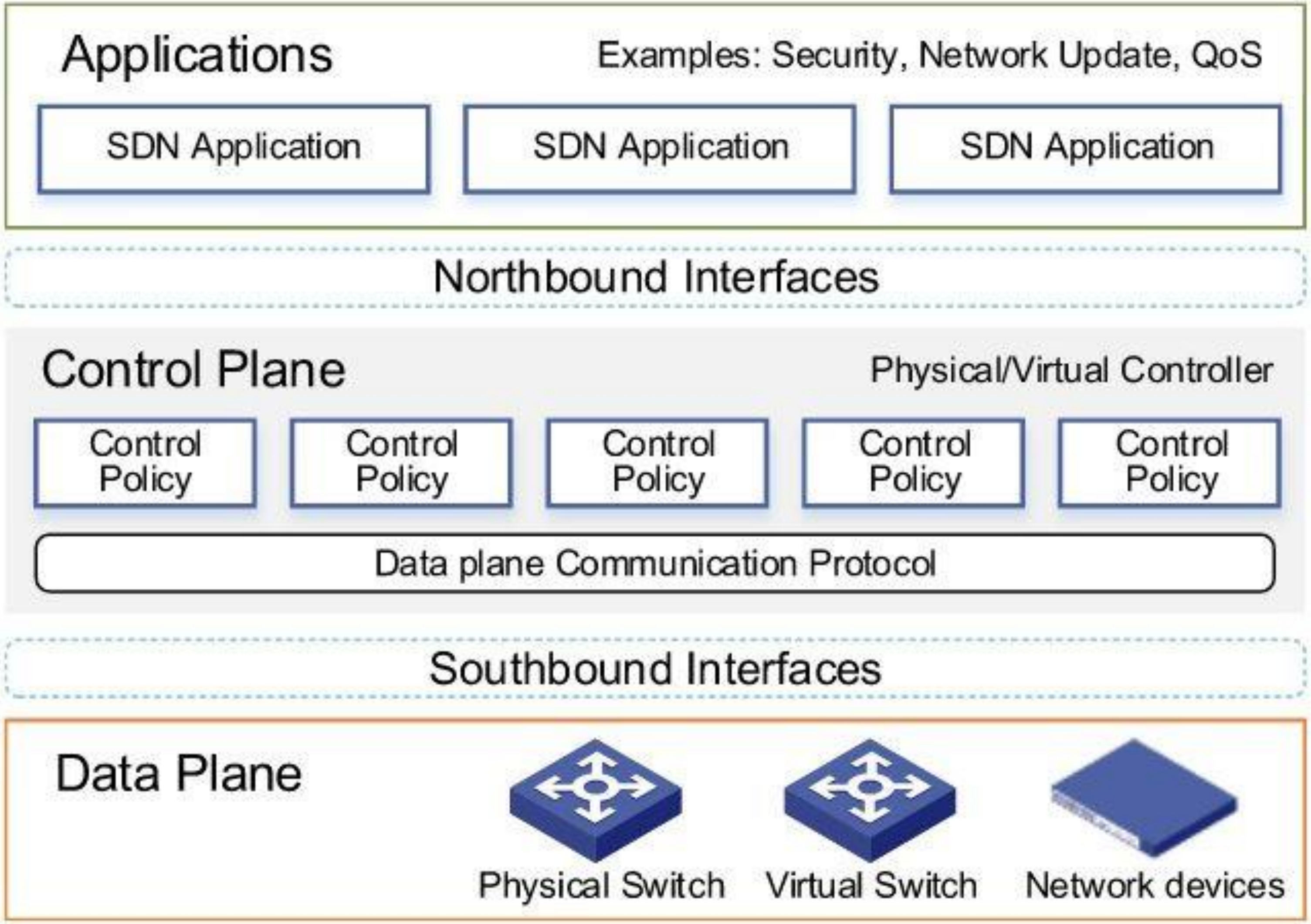
OPEN NETWORKING FOUNDATION



Abstract view of basic SDN components



SDN architecture overview



Examples of SDN Controllers [4, 5, 7, 16]

- Open source, LF Networking
- First release: **2014**
- Southbound Protocols: OpenFlow, OVSDB, NETCONF, BGP, LISP, SNMP...
- Written in **Java**

- Open source, ONF, LF
- First release: **2014**
- Southbound Protocols: P4, OpenFlow, NETCONF, TL1, SNMP, CLI, BGP, RESTCONF...
- Written in **Java**

- Ryu SDN Framework
- Open source (NTT)
- First release: **2014**
- Southbound Protocols: OpenFlow, Netconf, OF-config...
- Written in **Python**

- Calico
- Open source (Tigera, a startup)
- Just SDN !
- First release: **2014**
- Written in **Go**

Open source <u>software</u> switches & Network Operating Systems [17, 18, 19, 20, 21]

Main components of an OpenFlow switch [22]

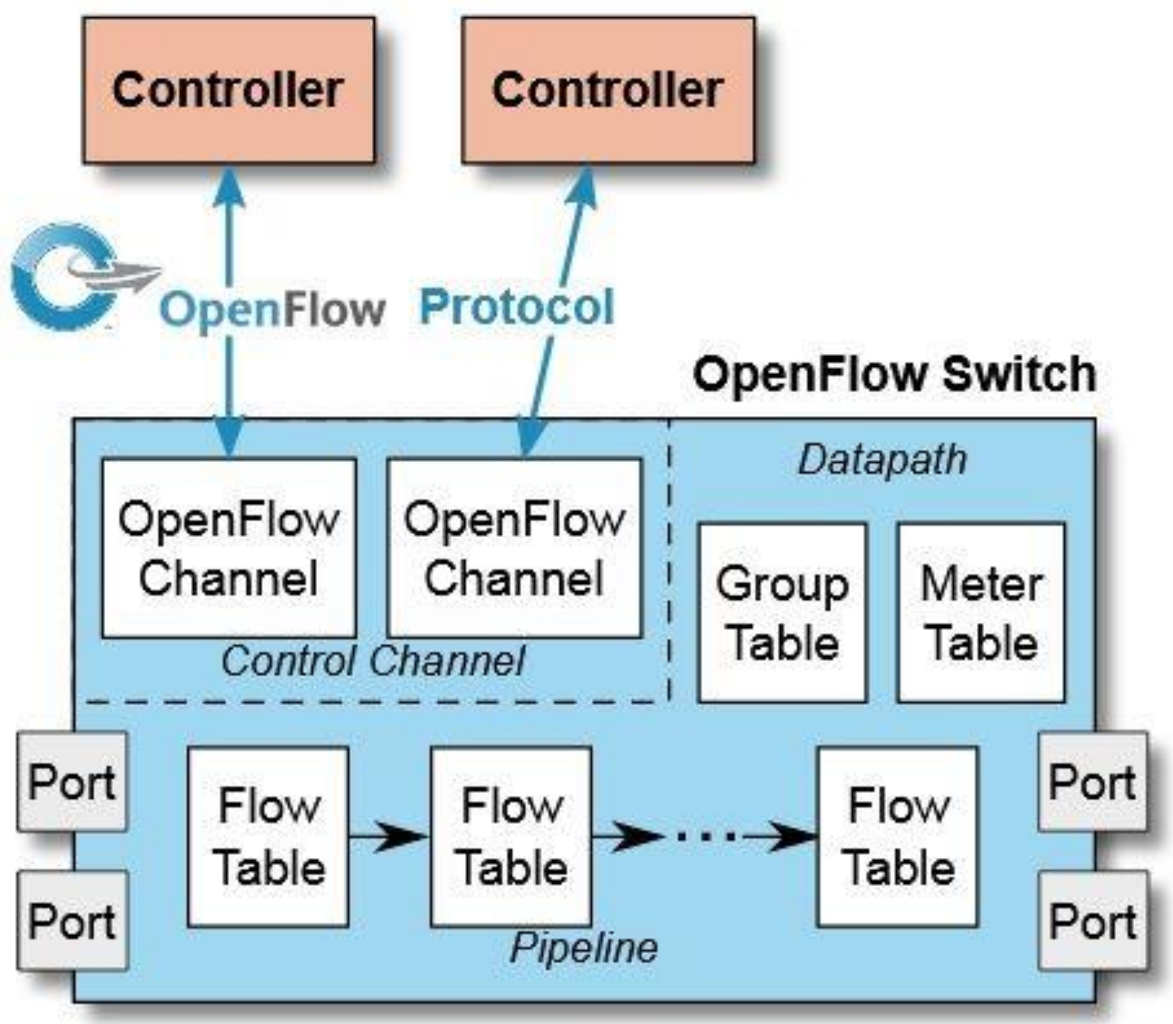
To perform packet **lookups** and **forwarding**, OpenFlow Logical Switch consists of:

- One or more **flow tables**: each flow table has multiple flow entries
- A **group table**: to group multiple flow entries
- A **meter table**: used for traffic shaping
- One or more **OpenFlow control channels** to an external controller

The switch communicates with the controller and the controller manages the switch via the OpenFlow switch protocol.

The controller can **add**, **update**, and **delete flow entries** in flow tables, both reactively (in response to packets) and proactively

Each flow table in the switch contains a set of flow entries; each flow entry consists of **match fields**, **counters**, and a **set of instructions to apply to matching packets**



Main components of an OpenFlow switch

OPENFLOW, a sum up:

- Forwarding table management protocol
- Defines the communication between an OpenFlow controller and an OpenFlow switch
- The protocol consists of a set of messages that are sent from the controller to the switch and a corresponding set of messages that are sent in the opposite direction
- Multiple OpenFlow channels are possible if an OpenFlow switch is managed by multiple controllers
- The OpenFlow channel can be either encrypted using TLS or directly over TCP

Fundamental packet paths:

- **Forward** the packet out a local port, possibly modifying certain header fields first
- **Drop** the packet
- **Pass** the packet to the **controller**

Matching fields: switch input port**,** VLAN ID, VLAN priority, Ethernet source address, Ethernet destination address, Ethernet frame type, src IP@, dst IP@, IP protocol, IP Type of Service (ToS) bits, TCP/UDP src port, TCP/UDP dst port

Other Standard Protocols

NETCONF [23]

- Network elements configuration management protocol
- Transport protocol independent
- Allows the separation of the configuration data from the operational data
- Supports an automated ordering of operations
- Facilitating straightforward rollback operations
- Manages and orchestrates multi-vendor infrastructures

RESTCONF [24]

- Network elements configuration management protocol
- API for create, read, update and delete (CRUD) operations accessing data defined in YANG based on HTTP transactions
- Web-based applications to access the configuration data, state data, data-model-specific remote procedure call (RPC) operations, and event notifications within a networking device, in a modular and extensible manner

Standard Modeling Languages

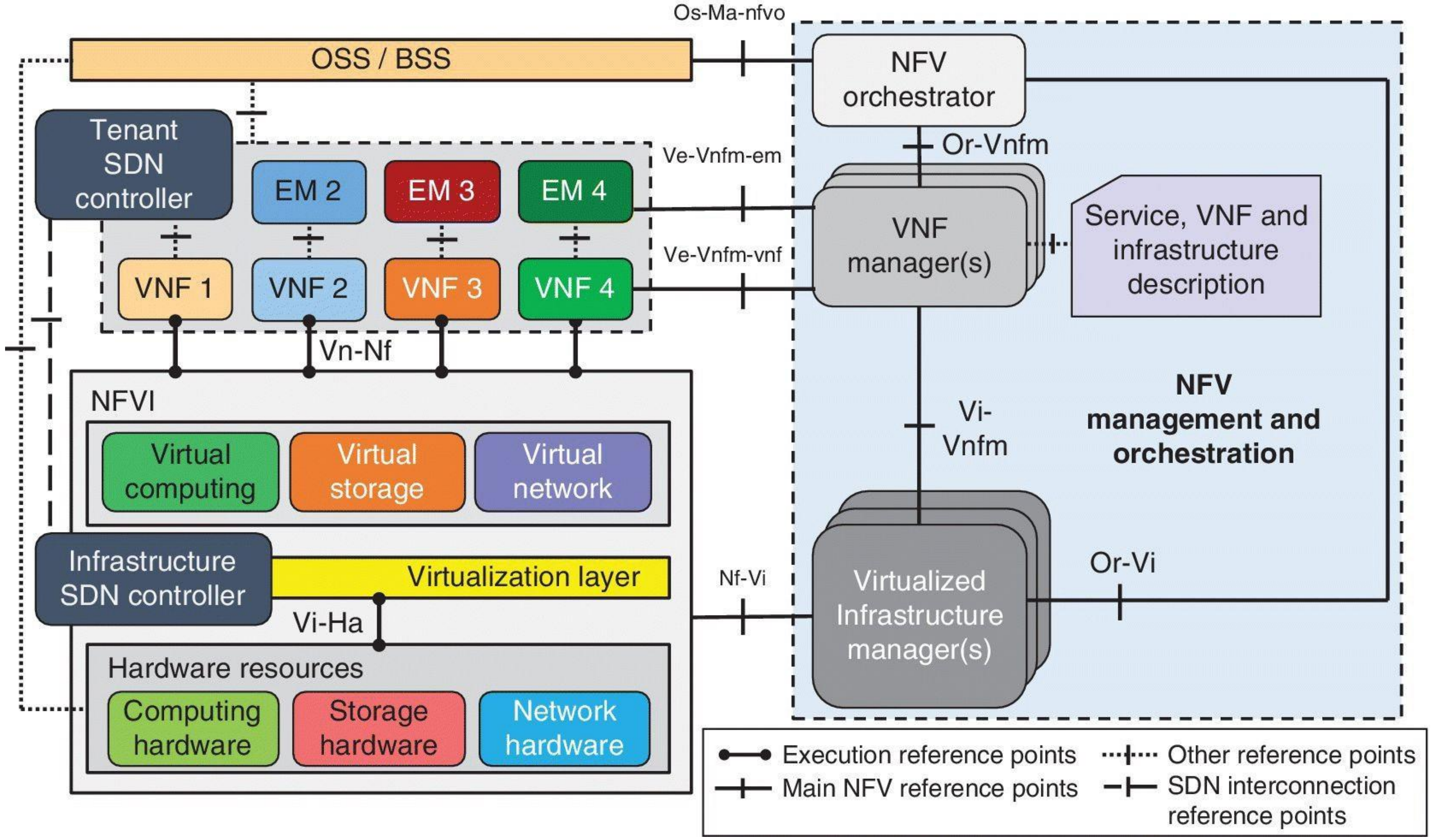
TOSCA (Topology and Orchestration Specification for Cloud Applications) [25]

- Modeling/description language for the orchestration of services and the management of VNF lifecycles
- Specifically designed to support describing both NS descriptors (NSDs) and VNF descriptors (VNFDs)
- Describes topologies of cloud-based web services, their components, relationships, and the processes that manage them, all by the usage of templates

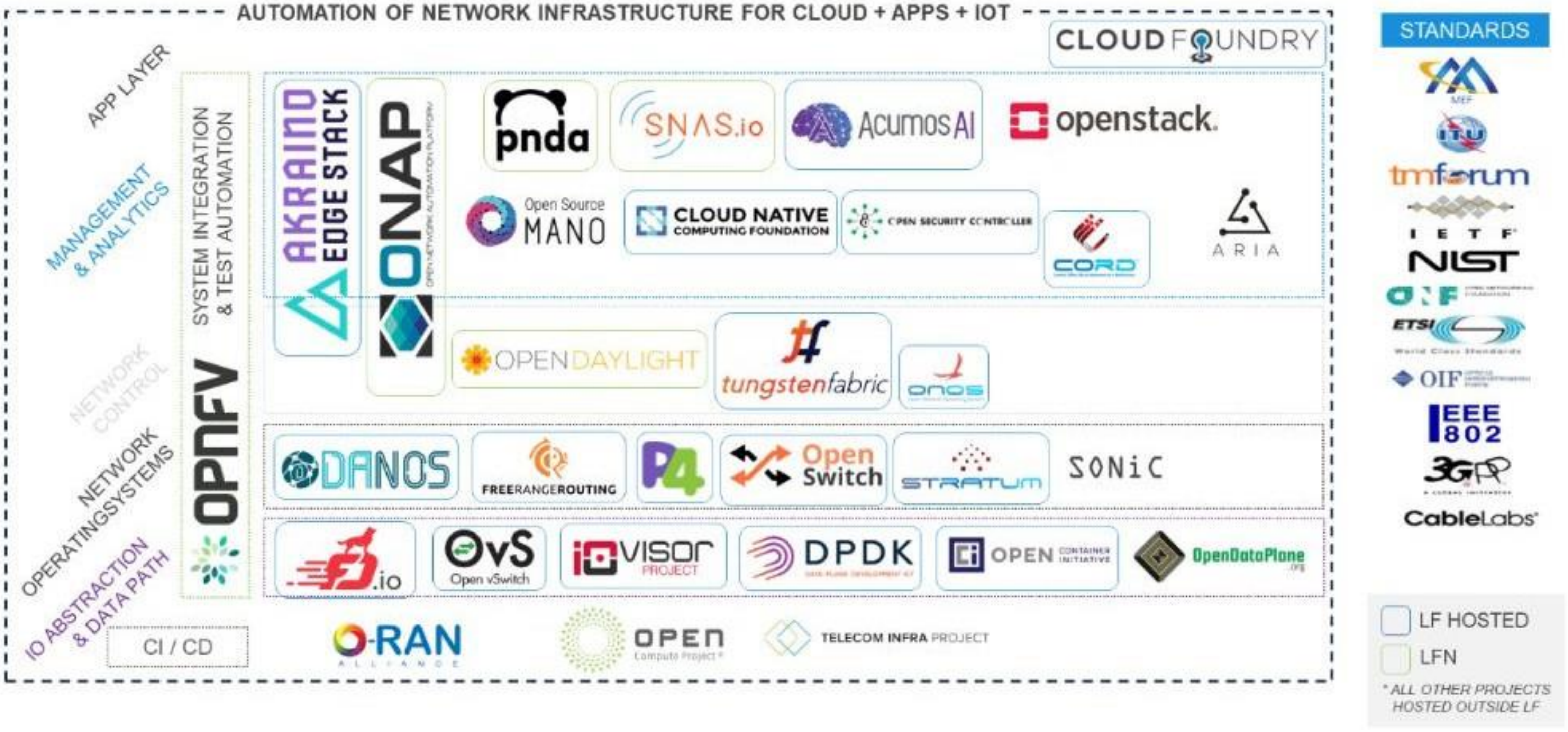
YANG (Yet Another Next Generation) [26]

- Data modeling language
- Complements NETCONF by defining the way in which the information applicable to a node can be read and written
- Provides abstractions of the network resources including both devices and services
- Supports capabilities like the validation of the input data
- Data model elements are grouped and can be used in a transaction

Infrastructure and tenant SDN-C in the ETSI NFV



4. Software Defined Networking (SDN)



5.

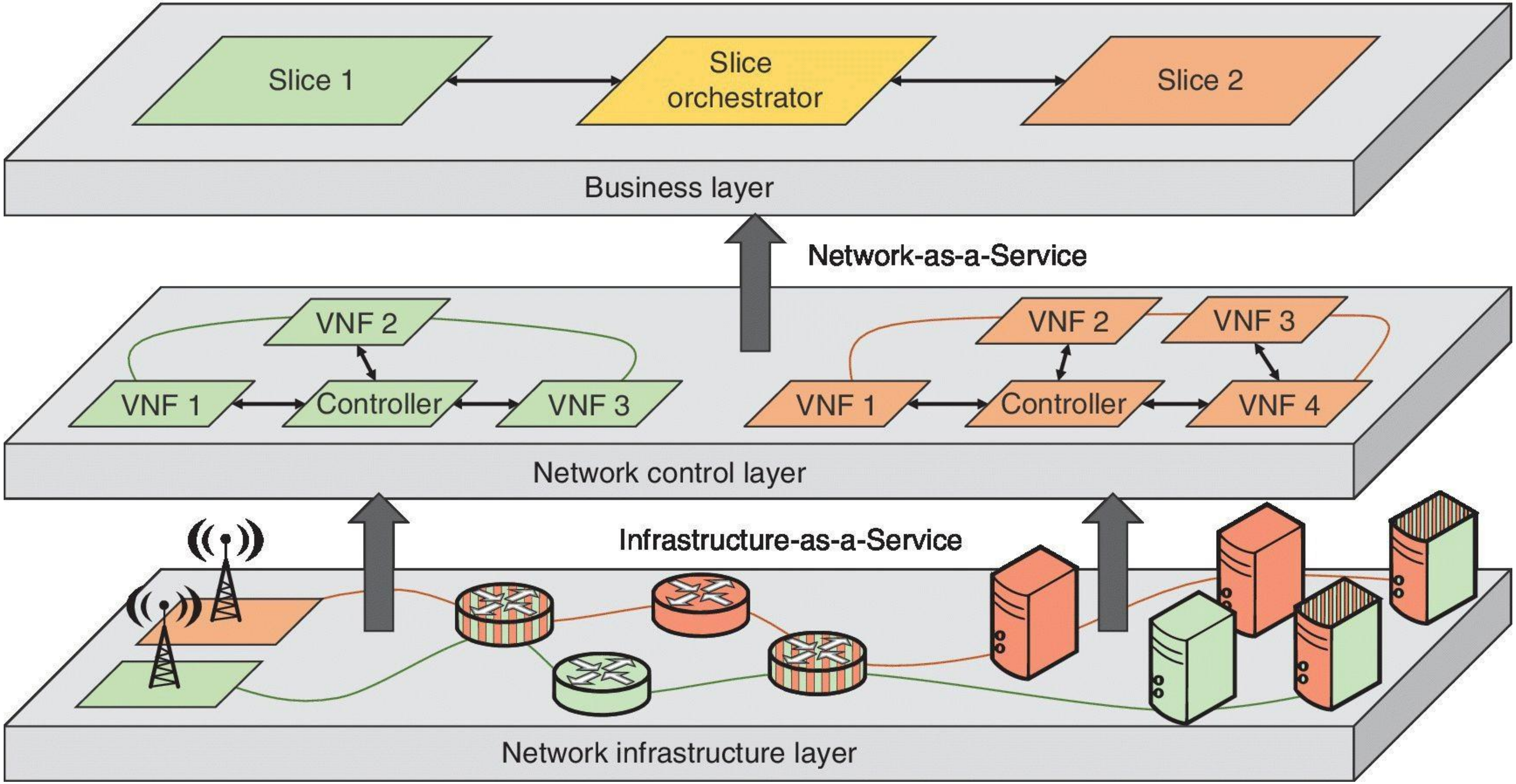
Network Slicing

- What is a Slice ?
- What are the benefits of Network Slicing ?
- How Slicing is implemented ?

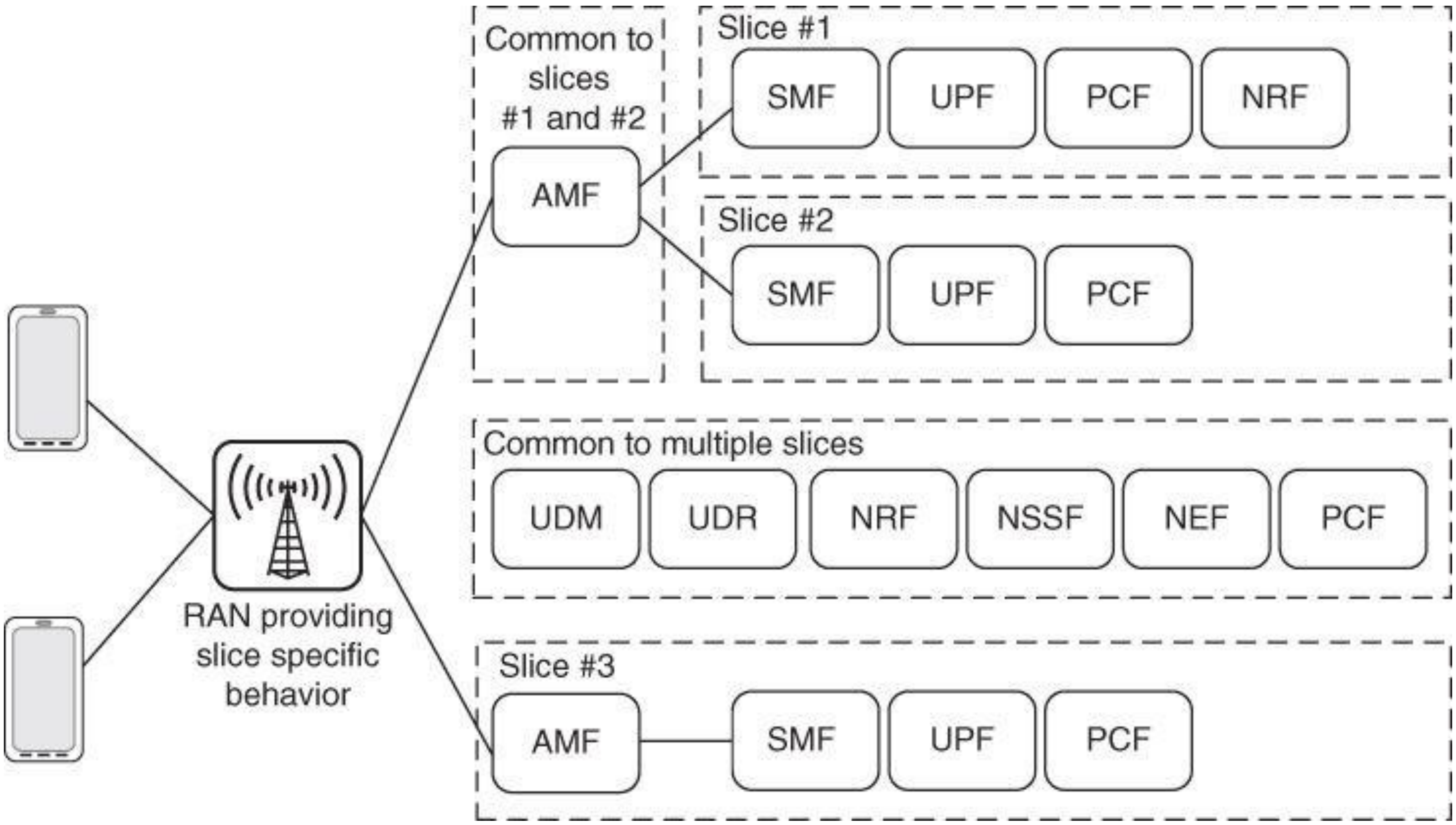
Slicing definition

- A slice is a **logical network** serving a particular application, user or business
- You can think about it as a “VPN” that has a certain **Quality of Service (QoS)** that needs to be maintained to **fulfill a given Service Level Agreement (SLA)**
- A network can be divided to multiple slices
- From the user and the infrastructure point of view, each slice looks like a separate network with its **own reserved resources**

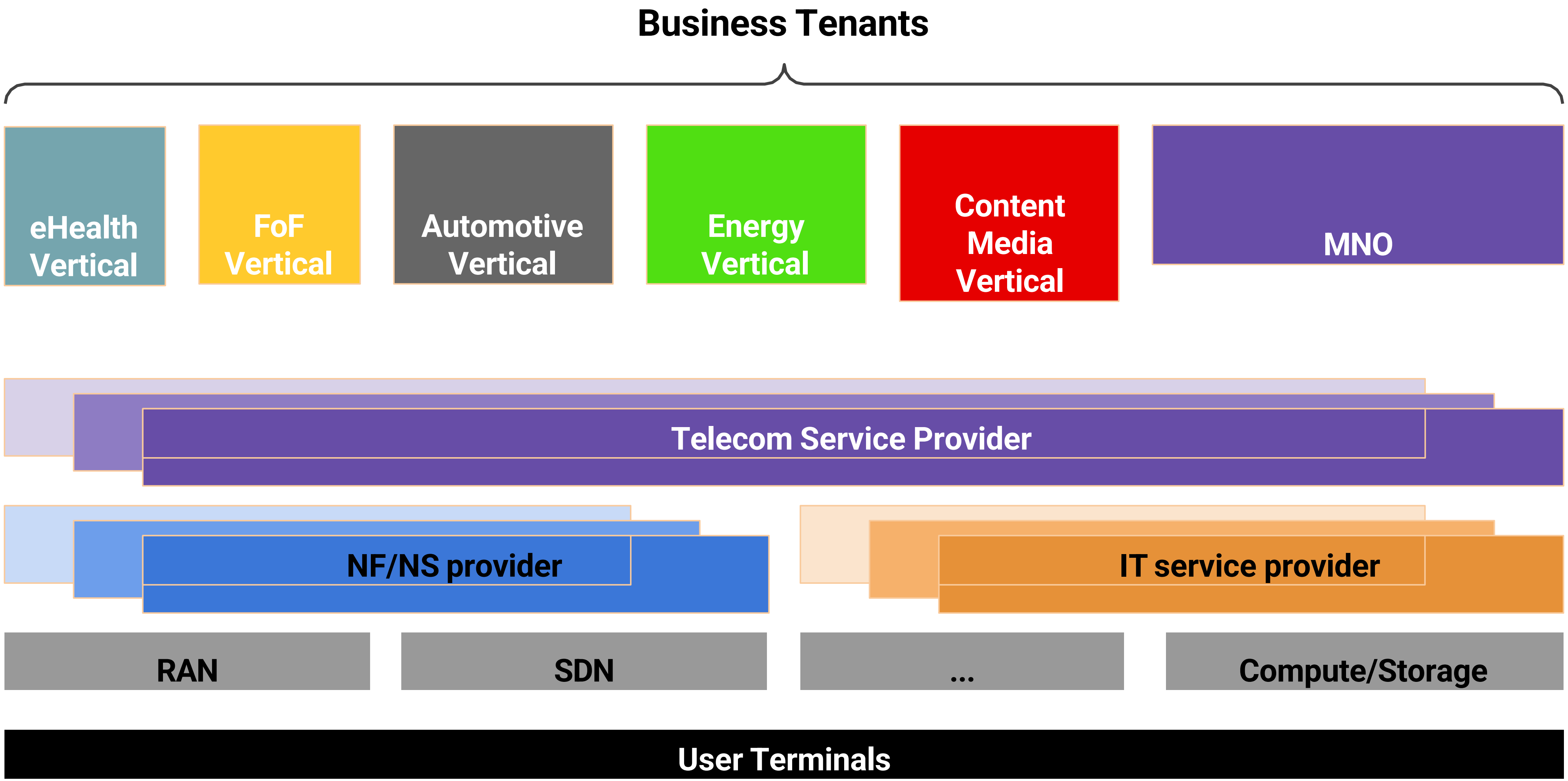
Network-sliced architecture



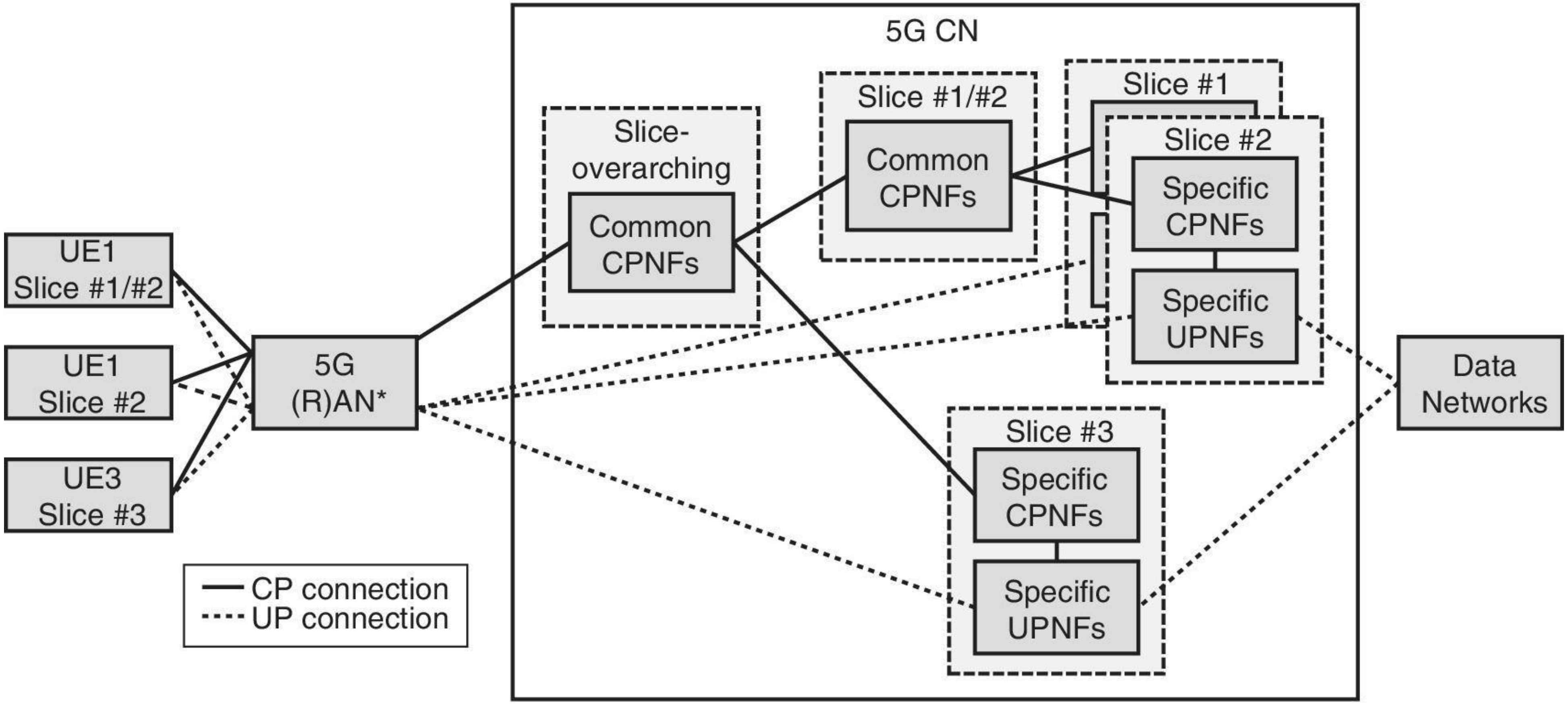
Network-sliced 5G system [10]



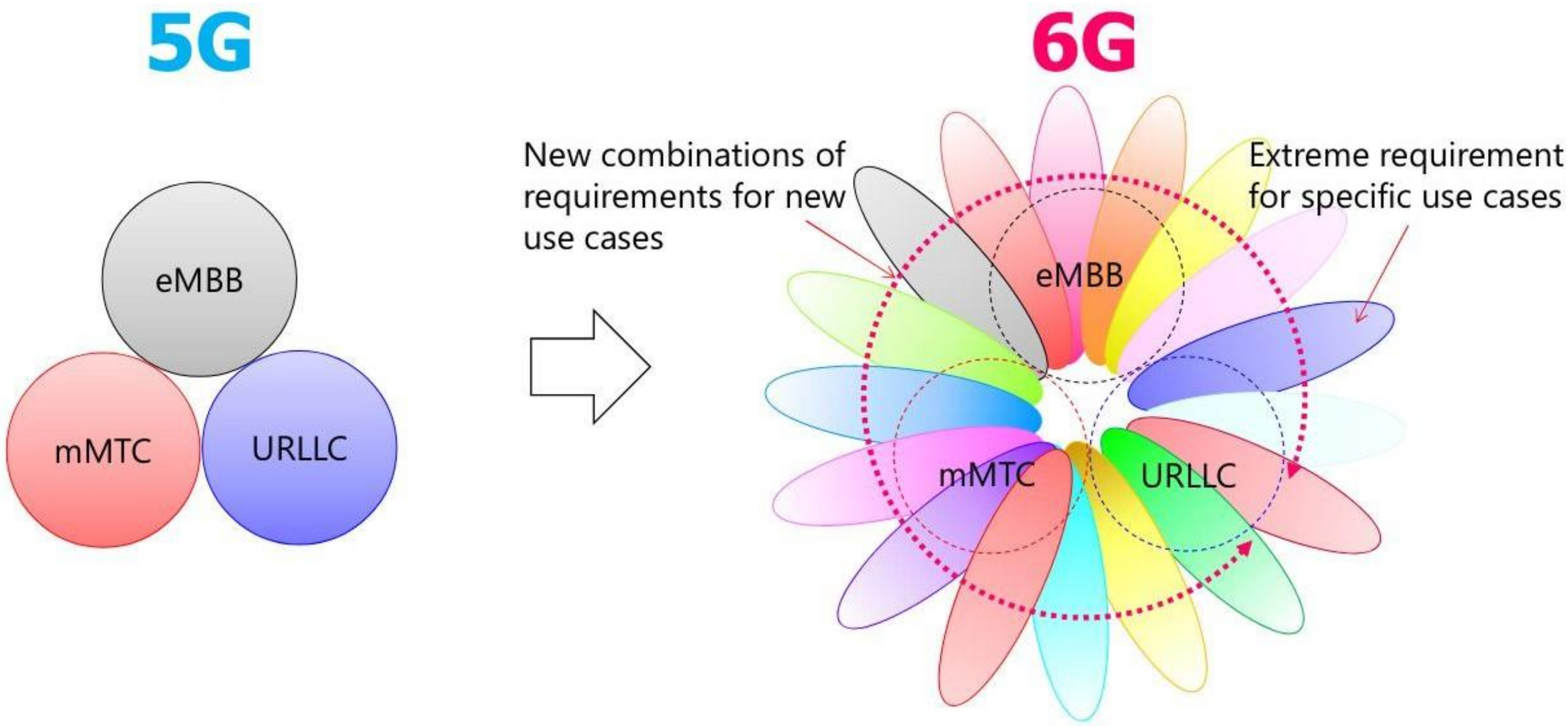
Multi-tenancy and Resource sharing

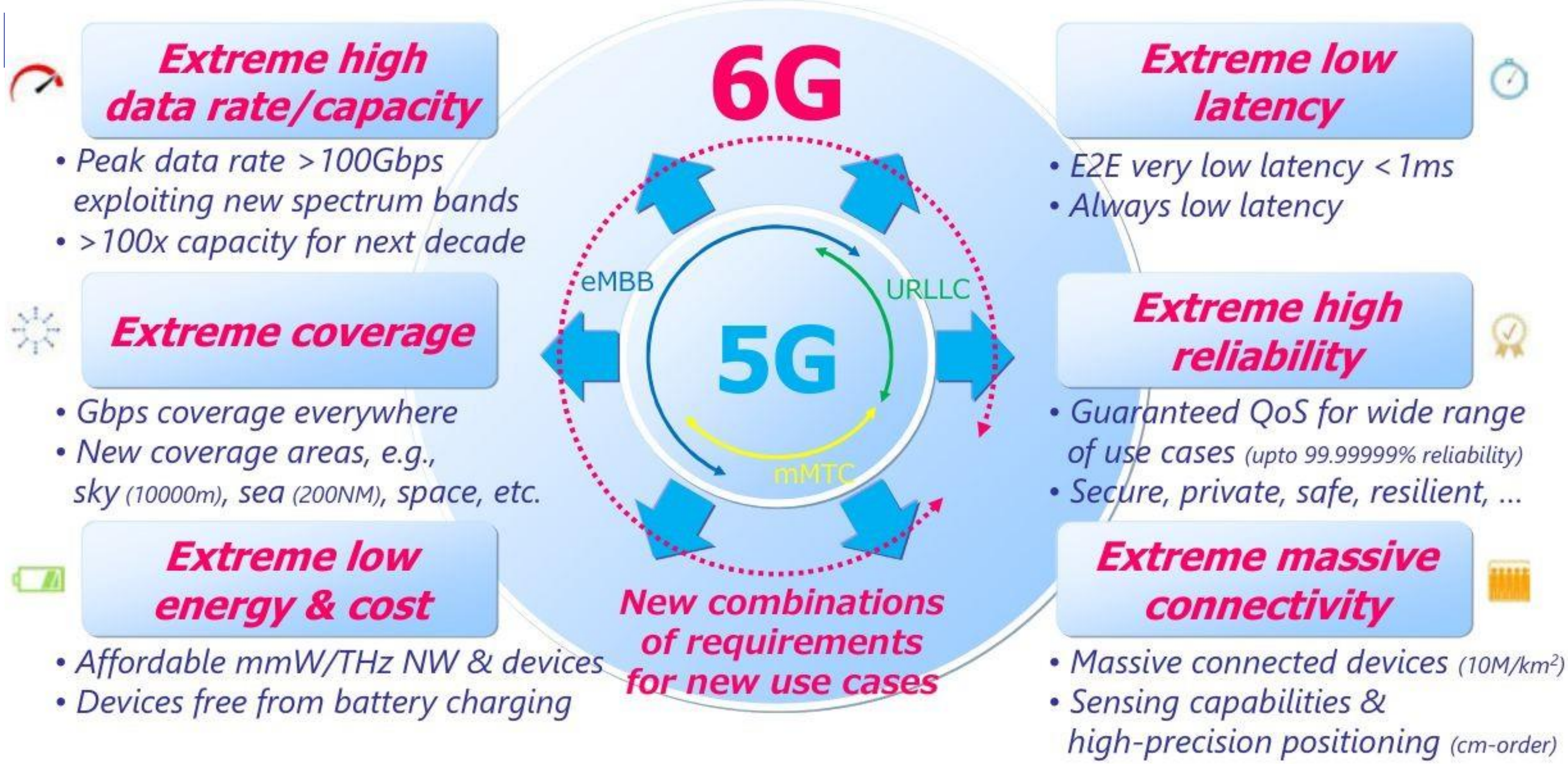


Network slices in the 5G CN with common and slice-specific NFs



* Slicing inside the (R)AN not explicitly shown


5G
eMBB
mMTC
URLLC
6G
New combinations of requirements for new use cases
Extreme requirement for specific use cases
eMBB
mMTC
URLLC


Extreme high data rate/capacity
• Peak data rate >100Gbps exploiting new spectrum bands
• >100x capacity for next decade
Extreme coverage
• Gbps coverage everywhere
• New coverage areas, e.g., sky (10000m), sea (200NM), space, etc.
Extreme low energy & cost
• Affordable mmW/THz NW & devices
• Devices free from battery charging
6G
5G
eMBB
URLLC
mMTC
New combinations of requirements for new use cases
Extreme low latency
• E2E very low latency <1ms
• Always low latency
Extreme high reliability
• Guaranteed QoS for wide range of use cases (upto 99.99999% reliability)
• Secure, private, safe, resilient, ...
Extreme massive connectivity
• Massive connected devices (10M/km2)
• Sensing capabilities & high-precision positioning (cm-order)

6.

Key Takeaways

6. Key Takeaways

- There are 3 big families of use cases in 5G: URLLC, eMBB and mMTC
- 5G CN architecture is based on SBA and each NF implements a REST API
- The main drivers of Telcos softwarization for the 5G are: NFV & SDN
- There are multiple techniques of virtualization and it can be applied to Servers (CPU, Memory, NIC), Network, Software, ...
- NFV is based on Server virtualization and aims at providing more flexibility and resiliency to mobile systems
- NFV leverages cloud computing and high volume servers & COTS to virtualize and provide as-a-service software network functions
- ETSI NFV architecture provides the building blocks of the NFV system components and interfaces
- NFV is backed up by SDN to enable advanced VNFs networking and automation
- SDN controllers control basic software or physical switches that perform matching & forwarding
- NFV and SDN together allows more flexibility and control over the network by slicing the resources and sharing them between multiple tenants

7.
References & Abbreviations

References

1 “5G Evolution and 6G - White Paper”. NTT DOCOMO, INC. January 2020
2 "Network Functions Virtualization— Introductory White Paper". ETSI. October 2012
3 "OpenFlow: enabling innovation in campus networks." McKeown, Nick, et al. ACM SIGCOMM Computer Communication Review 38.2 (2008): 69-74
4 https://www.opendaylight.org/
5 https://www.opennetworking.org/onos/
6 Kubo, Rui, et al. "Ryu SDN framework: Open-source SDN Platform software." NTT Tech. Rev 12.8 (2014): 1-5.
7 https://ryu-sdn.org/
8 Alliance, N. G. M. N. "5G white paper." Next generation mobile networks, white paper 1 (2015)
9 Yala, Louiza, et al. "Testbed federation for 5g experimentation: Review and guidelines." 2019 IEEE Conference on Standards for Communications and Networking (CSCN). IEEE, 2019
[10] Chandramouli, Devaki, Rainer Liebhart, and Juho Pirskanen, eds. 5G for the Connected World. John Wiley & Sons, 2019
[11] Portnoy, Matthew. Virtualization essentials. Vol. 19. John Wiley & Sons, 2012
[12] Goransson, Paul, Chuck Black, and Timothy Culver. Software defined networks: a comprehensive approach. Morgan Kaufmann, 2016
[13] Rajendra, C., F. Syed, and S. Paresh. "Network Functions Virtualization (NFV) with a Touch of SDN." ISBN-10: 0-10-446305-6, November (2016)
[14] ETSI, NFV ISG. "Network Functions Virtualisation (NFV); Management and Orchestration." NFV-MAN 1 (2014): v0
[15] https://osm.etsi.org/
[16] https://www.projectcalico.org/
[17] https://www.openvswitch.org/

References

[18] https://www.ovn.org/en/
[19] https://p4.org/
[20] https://www.openswitch.net/
[21] https://azure.github.io/SONiC/
[22] Open Networking Foundation. “OpenFlow Switch Specification Version 1.5.1”, ONF TS-025. March 26, 2015
[23] Rob, E. "NETCONF Configuration Protocol." *Internet Engineering Task Force IETF*, RFC 4741 (2006)
[24] Bierman, Andy, et al. "RESTCONF protocol." *Internet Engineering Task Force IETF*, RFC 8040. (2017)
[25] Standard, O. A. S. I. S. "Topology and orchestration specification for cloud applications version 1.0." (2013): 1-114.
[26] Bjorklund, Martin. "YANG-a data modeling language for the network configuration protocol (NETCONF)." (2010).
[27] https://inform.tmforum.org/insights/2020/10/dtws-open-digital-architecture-pulls-in-more-big-vendors/
[28] Series, M. "IMT Vision–Framework and overall objectives of the future development of IMT for 2020 and beyond." Recommendation ITU 2083 (2015)

Abbreviations

3GPP	3rd Generation Partnership Project
4G	4th Generation of cellular networks
5G	5th Generation of cellular networks
BSS	Business Support System
CAPEX	Capital Expenditures
CDN	Content Delivery Network
CN	Core Networks
COTS	Commercial Off-The-Shelf
DC	Data Center
DPDK	Data Plane Development Kit
ETSI	European Telecommunications Standards Institute
FoF	Factory of the Future
Gbps	Gigabits per second
gNB	gigabit NodeB
HTTP	HyperText Transfer Protocol
LAN	Local Area Network
LTE	Long Term Evolution
Mbps	Megabits per second
MNO	Mobile Network Operator
NFV	Network Function Virtualization
OPEX	Operational Expenditures
OSM	Open Source Mano
OSS	Operations Support System
OTT	Over The Top
RAN	Radio Access Network
SDN	Software Defined Networking
SR-IOV	Single Root Input/Output Virtualization
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UE	User Equipment
VLAN	Virtual LAN
VM	Virtual Machine
VMM	Virtual Machine Monitor
VNF	Virtual Network Function
VXLAN	Virtual Extensible LAN
WAN	Wide Area Network

Extras