SVEUČILIŠTE U ZAGREBU
Fakultet elektrotehnike i računarstva

Master Programme

Computing

Advanced Architectures of Telecommunication Networks

Lecture 5: Cloud virtualization technologies. Network Function Virtualization.

prof. dr. sc. Lea Skorin-Kapov

Ac. year 2022/2023

The telecom world is changing to become virtualized and cloudified

Outline

- Cloud computing
- Virtualization technologies
- Network Function Virtualization
- Network virtualization

Cloud Computing

Cloud computing

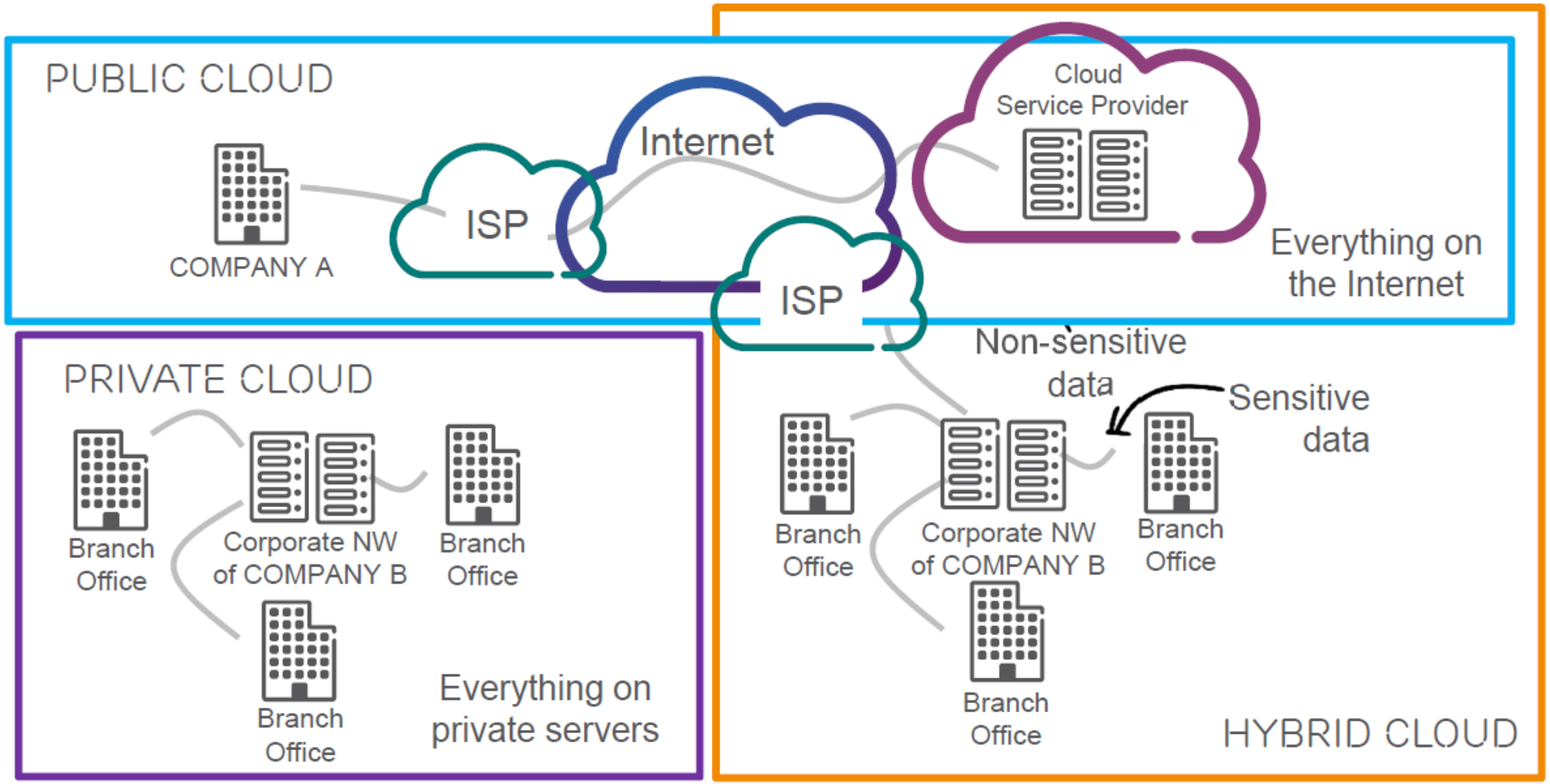
- A model for enabling ubiquitous, on-demand network access to a shared pool of configurable resources (e.g., networks, servers, storage, applications, services) that can be rapidly provisioned and released with minimal management effort or service provider interaction
- From an end user/company perspective:
  - **Elasticity**: gives illusion of “infinite” computing resources available rapidly and on-demand
  - **Reduced capital expenditures** (CapEx): costs related to aquiring and maintaining physical resources
  - **Multitenancy**: several different *cloud* customers are accessing the same computing resources

Key concepts

- **Transparency**
  - The user does not know the specification of the system on which the applications and services will run
  - Data is stored in locations unknown to users
  - The administrative system is not under the control of the user
  - Access to applications and services is provided via the Internet

- **Virtualization**
  - The ability to run multiple operating systems on one physical or multiple physical computers
  - The same goes for data storage
  - Resources can be shared or pooled
  - Computers have virtualization support (hypervisor technology)
  - Examples: Xen, VMware, Wine, ..

Deployment models

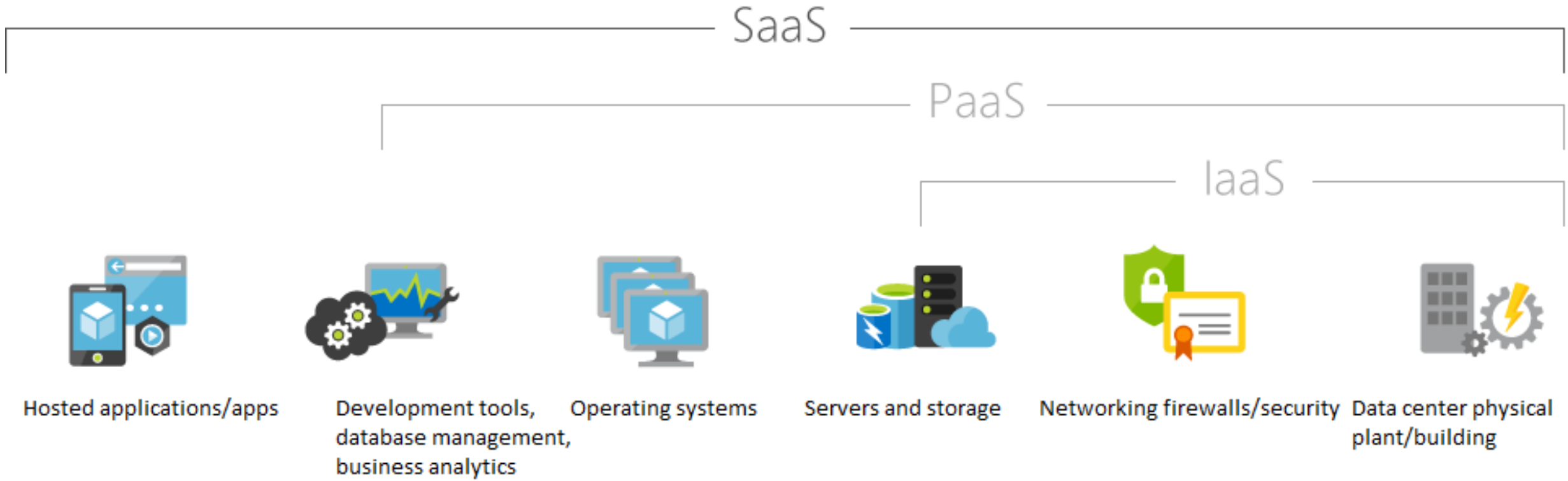


source: Ericsson

Deployment models: comparison

- Private cloud: implemented within the IT environment of an organization
  + high security, high reliability, greater control of the cloud infrastructure
  - limited scalability, high costs
- Public cloud: infrastructure made available to the general public from a third-party service provider via the Internet
  + high scalability, low cost
  - moderately secure, medium performance and reliability
- Hybrid cloud: a combination of a public and private cloud that interoperates
  - users typically outsource non business-critical information and processing to the public cloud, while keeping business-critical services and data in their control

Cloud services

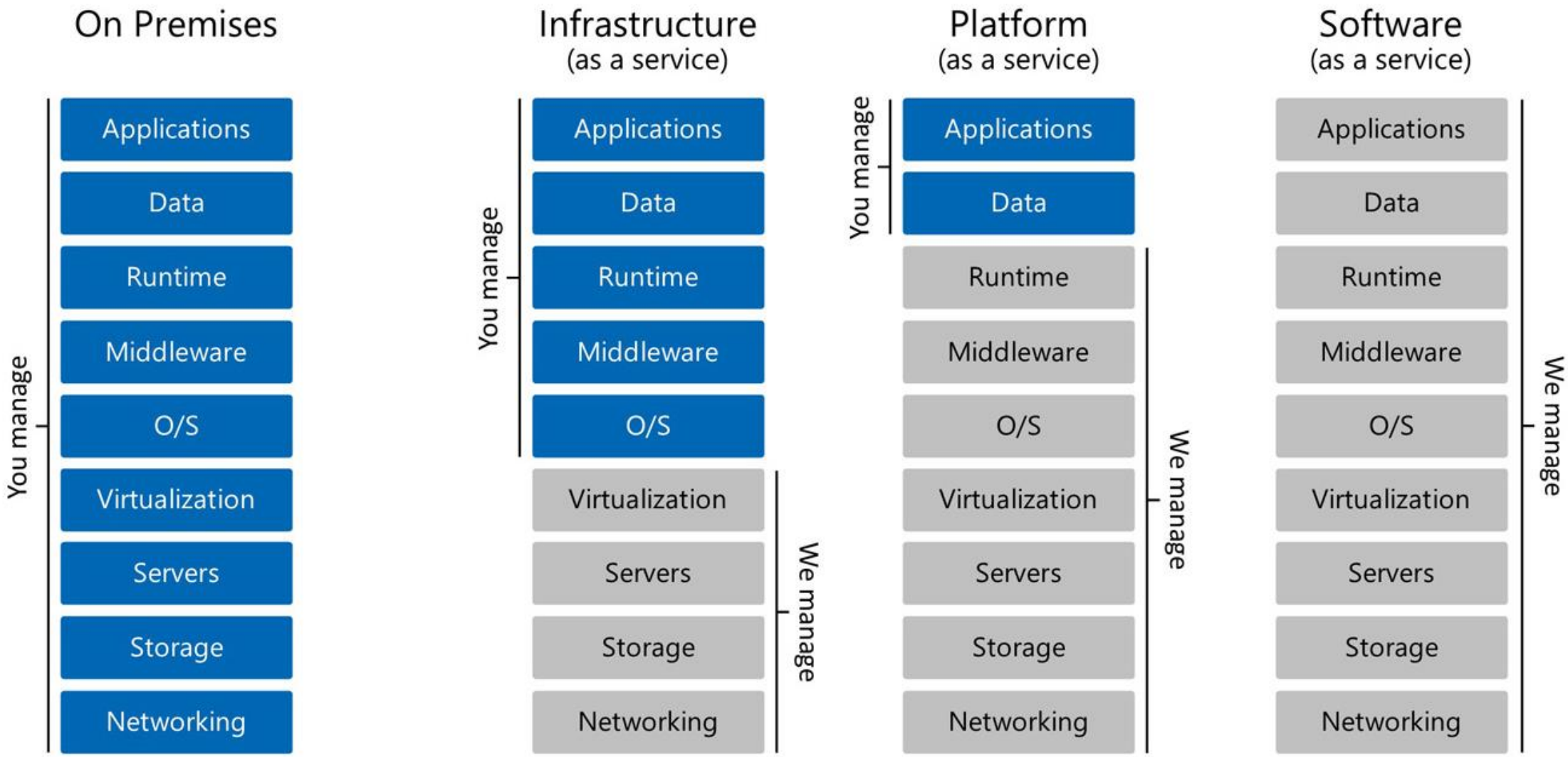


source: https://cic.gsa.gov/solutions/saas

Cloud services: explained

- **Software as a Service (SaaS)**: provide services to customers in the form of application software running on and accessible in the cloud
  - Examples: G Suite (Gmail, Google docs, Google Drive, etc.), Dropbox, Office 365, YouTube, slack, Zoom, Cisco WebEx, GitHub….
- **Platform as a Service (PaaS)**: provide customers with a development environment accessible in the cloud (e.g., programming languages, execution environments); customers create, deploy, manage and run their own apps
  - Examples: Google App Engine, Windows Azure Platform, Amazon Elastic Kubernetes Service (EKS)
- **Infrastructure as a Service (IaaS)**: provide customers with access to the resources of the underlying cloud infrastructure (compute, storage, network)
  - Examples: Amazon Elastic Compute Cloud, Google Compute Engine, MS Azure, vmware (e.g., VMware Telco Cloud Platform)

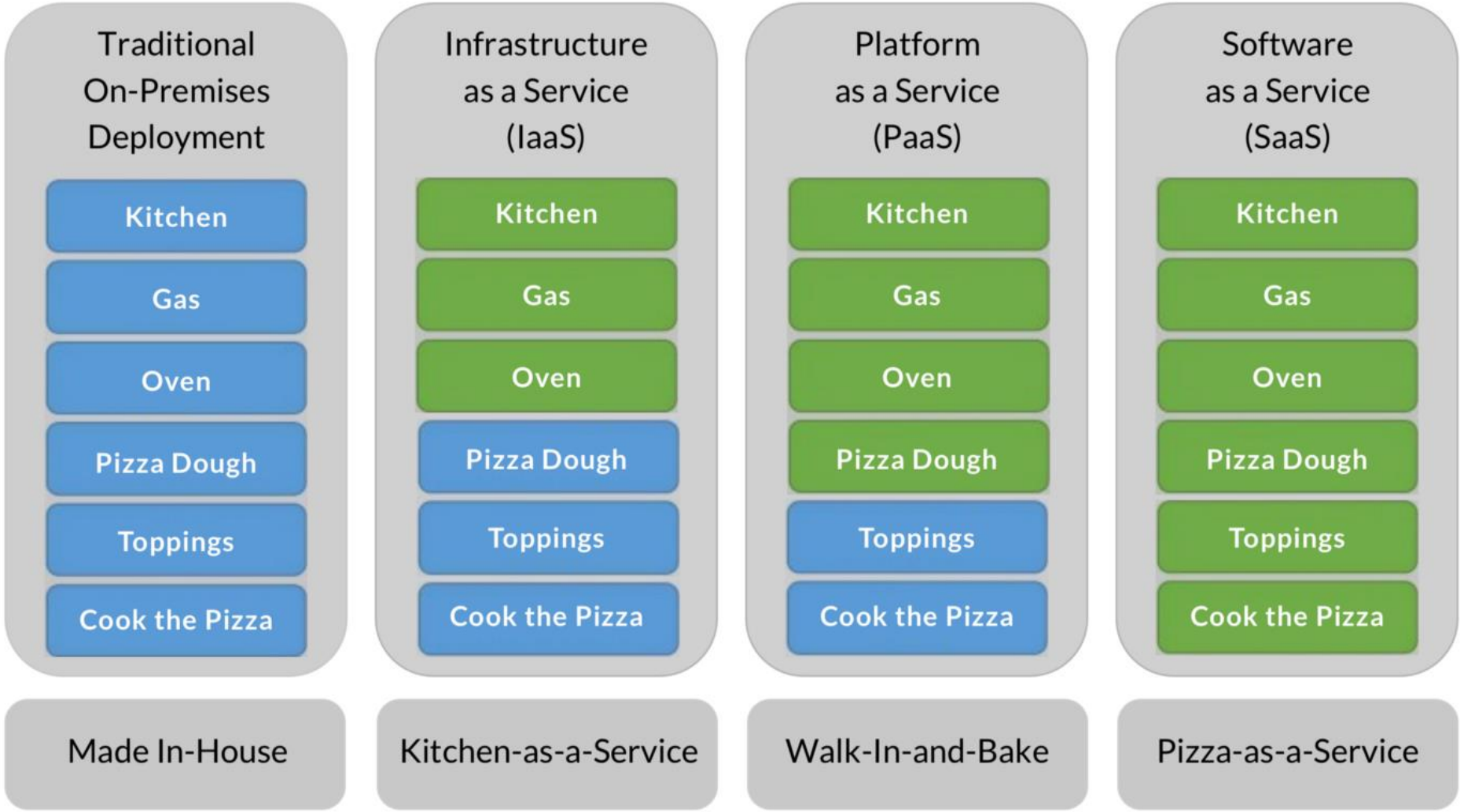
Cloud services: who manages what?



source: Microsoft

Example: Pizza as a Service ☺

source: https://m.oursky.com/saas-paas-and-iaas-explained-in-one-graphic-d56c3e6f4606



Virtualization technologies

Virtualization – the foundation of cloud computing

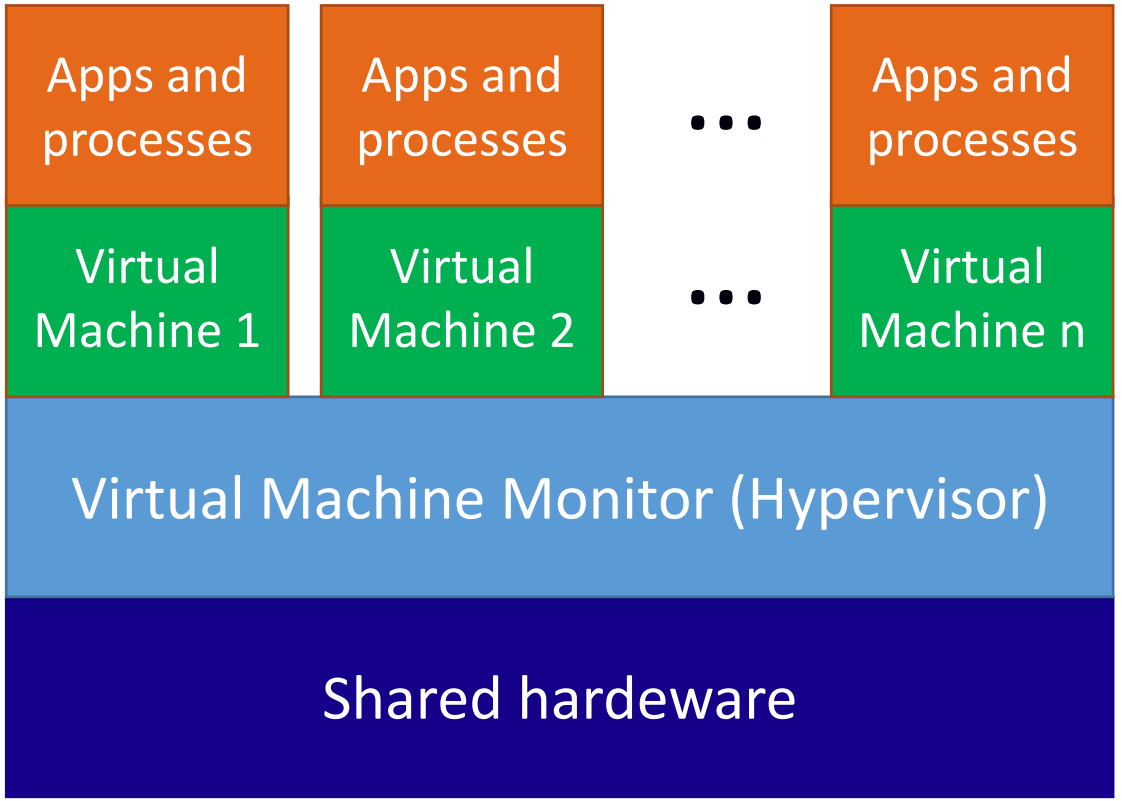
- the term “virtualization” was coined back in the 1960s (IBM)
- the process of turning physical resources into logical, or virtual, resources
- an abstraction layer is provided between software and the actual physical hardware
- multiple virtual instances share the same physical hardware resources

What can be virtualized?

- Servers – a physical server can be abstracted into virtual servers
- Storage – a physical storage device can be abstracted into virtual storage device/disk
- Network – physical routers and switches can be abstracted into logical networks (e.g., VLAN)

Virtual Machines (VMs)

- traditionally: apps run directly on an OS on a PC or server, with each PC/server running one OS at a time
- virtualization: enables a single PC/server to simultaneously run multiple OSs or multiple sessions of a single OS
- a host machine can support multiple VMs

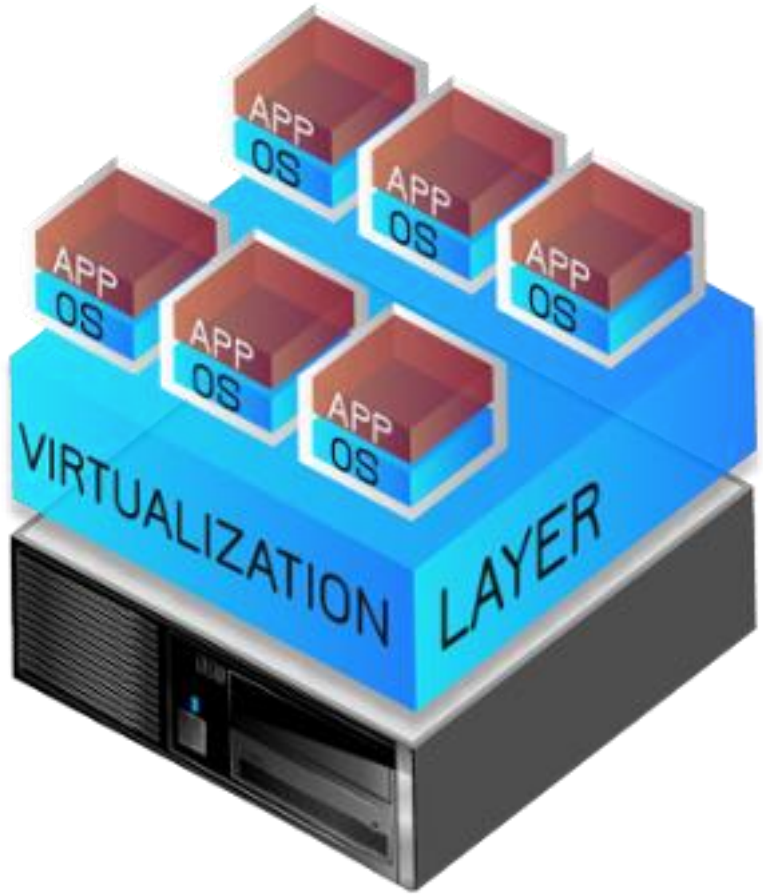


→ *host machine*: actual machine where virtualization takes place

→ *guest machine*: the VM

consolidation ratio: the number of guest machines that can exist on a single host

→ 6:1 consolidation ratio



Traditional server architecture

Virtualized server architecture

Today, more virtual servers deployed in the world than physical servers!

Hypervisor (aka *Virtual Machine Monitor*, VMM)

- software that creates and runs VMs
- enables one host computer to support multiple guest VMs by virtually sharing its resources (e.g., memory, processing)

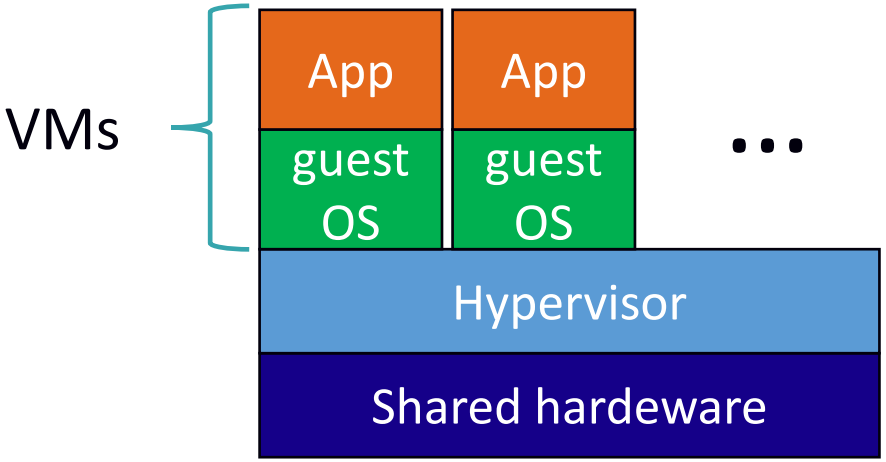
Benefits:

- Speed: allow VMs to be created “instantly”
- Efficiency: more efficient utilization of one physical server by running several VMs on one physical machine’s resources
- Flexibility: OSs and associated apps can run on a variety of HW types because the hypervisor separates the OS from underlying HW
- Portability: VMs that the hypervisor runs are independent from the physical (host) machine, and are thus portable

Types of hypervisors

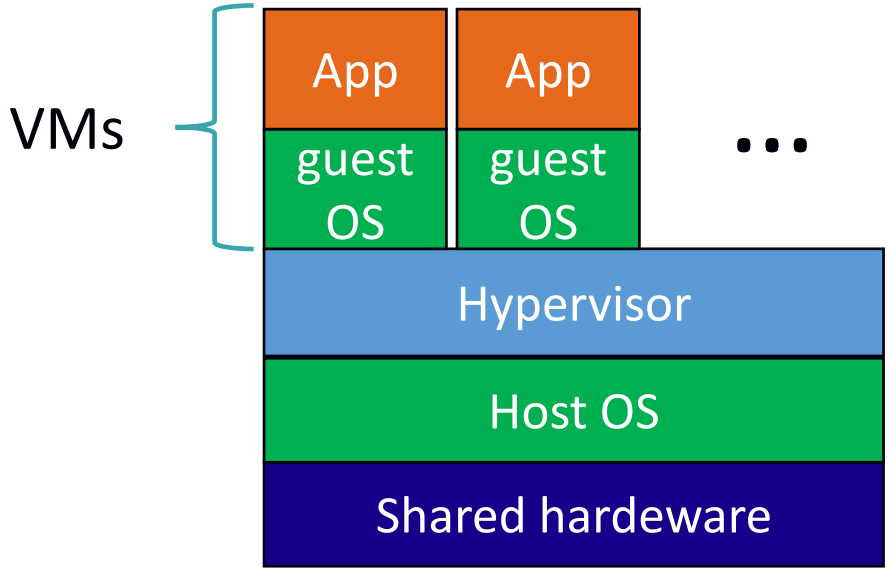
Type 1 (native, or"bare metal"): acts like a lightweight OS and runs directly on the host's hardware (can directly control physical resources)

Examples: Citrix/Xen Server, VMware ESXi, Microsoft Hyper-V



Type 2 ("hosted"): runs on top of a conventional host OS, and relies on the OS to handle hardware interactions on its behalf

Examples: Oracle VM VirtualBox, VMware Workstation

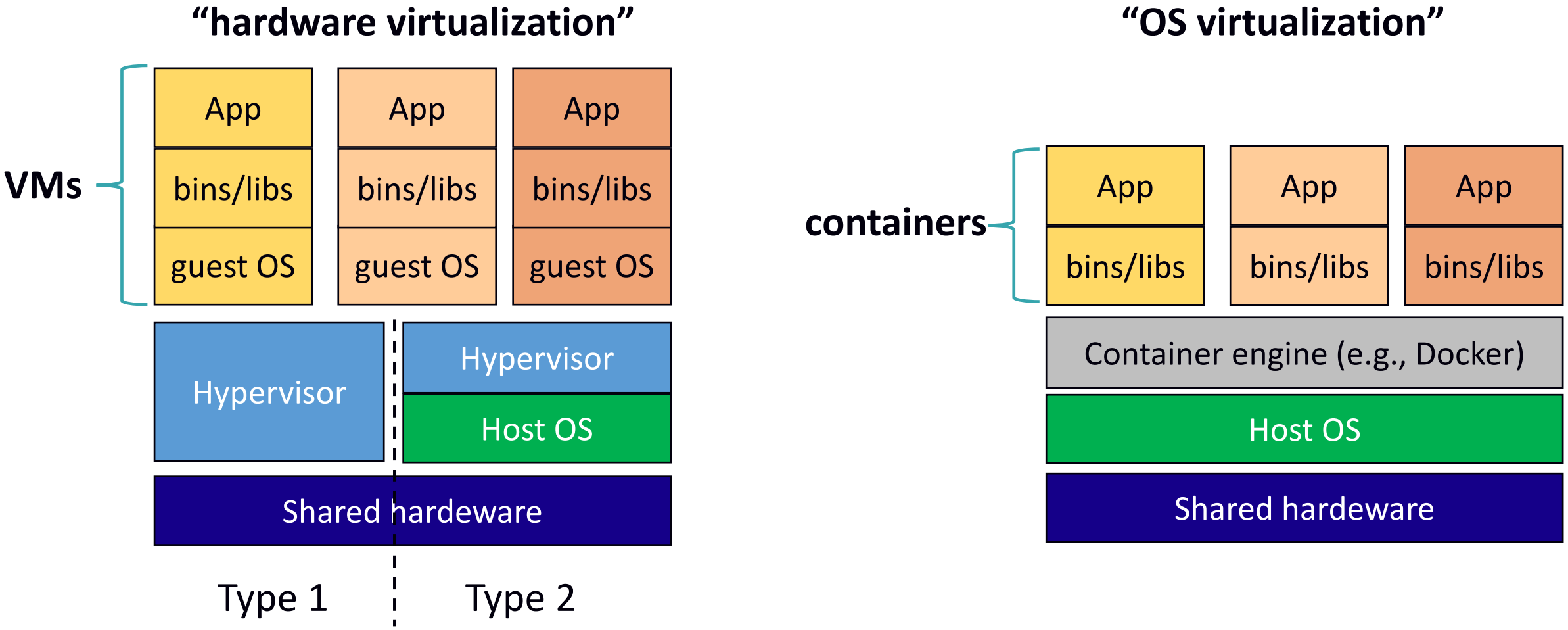


→ Typically, Type 1 hypervisors perform better than Type 2 (no "extra layer", more secure)

Virtualization vs. containerization

- **Hypervisors**:
  - Isolate (virtual) machines
  - Allow an OS to run independently from the underlying hardware through the use of VMs.
  - Share virtual computing, storage and memory resources.
  - Can run multiple OSs on top of one server (Type 1) or installed on top of one standard OS and isolated from it (Type 2).
- **Containers**:
  - Isolate processes
  - Allow applications to run independently of an OS.
  - Containerized apps on a host share a common OS kernel
  - Can run on any OS — all they need is a container engine to run.
  - Extremely portable since in a container, an application has everything it needs to run.

Difference between VMs and containers



→ instead of virtualizing the underlying hardware, containers virtualize the OS (typically Linux or Windows) so each individual container contains *only* the app and its libraries and dependencies.
→ containers are smaller and lighter weight compared to use of hypervisor/VMs

Network Function Virtualization
NFV

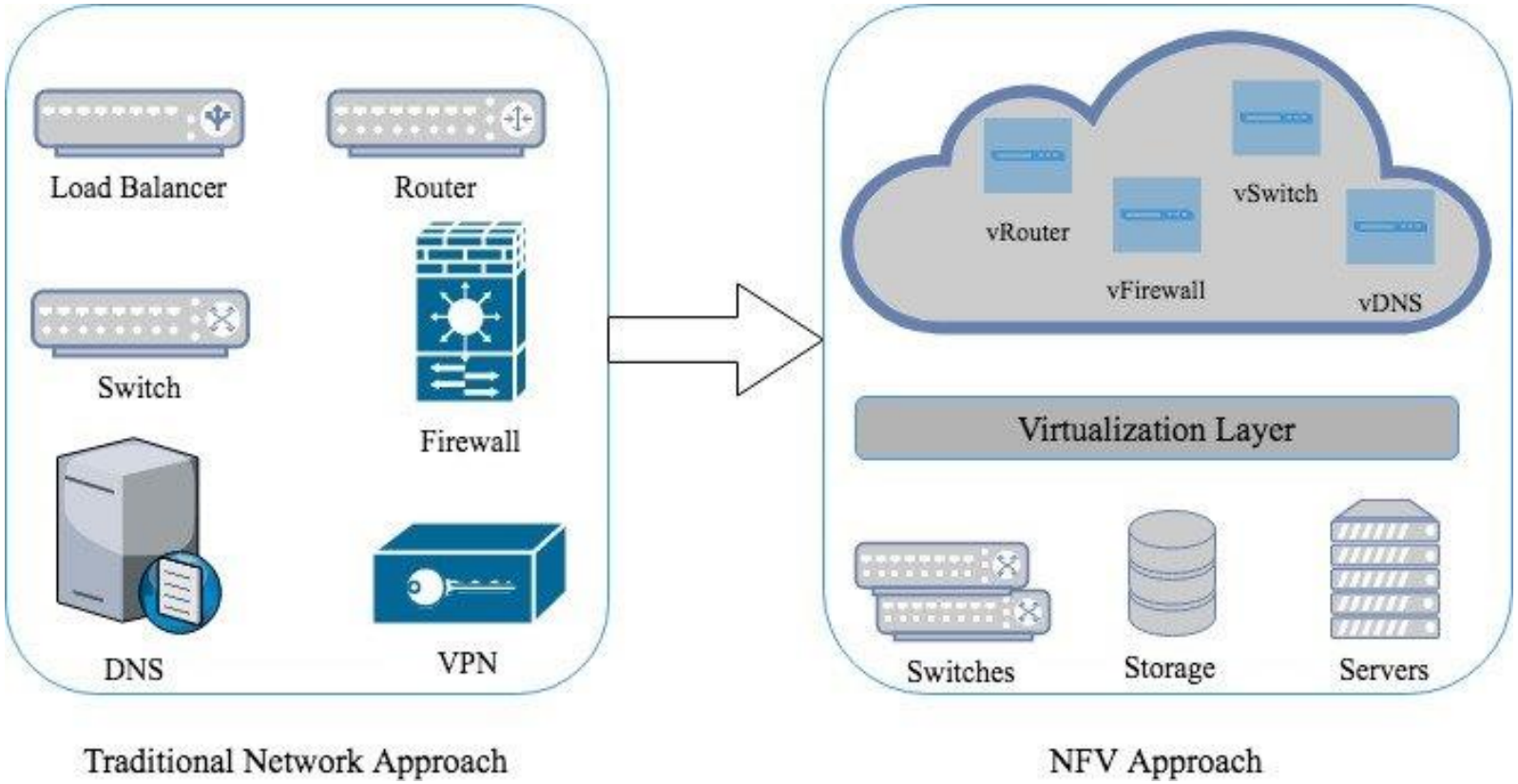
video source: ETSI

Motivation: shortcomings of traditional network design

- **Complex network design**
  - large variety of proprietary nodes and hardware appliances
- **Launching new services is difficult, takes long, and is expensive**
  - new services may require additional different types of hardware
  - new hardware → high CapEx; additional space and power
- **Expensive operation**
  - hardware appliances rapidly reach end of life (need to repeat procure-design-integrate-deploy cycle)

NFV concept

- A network architecture or concept used to virtualize entire network node functions, such as routers, firewalls, and load balancers
- Involves implementing network functions in software that can run on a range of industry standard server hardware



Alwakeel, Ahmed & Alnaim, Abdulrahman & Fernández, Eduardo. (2019). Toward a Reference Architecture for NFV. 1-6

Example VNFs

- Switching: routers, NATs, switches
- Mobile network nodes (Packet Data Gateway, MME…)
- Home routers and set top boxes
- Tunneling gateway elements
- Signalling nodes
- Network-wide functions (AAA servers, policy control…)
- Application level optimization (Content Delivery Networks, Load Balancers)
- Security functions (firewalls, intrusion detection system…)

Why NFV?

Enables service providers to manage and expand their network capabilities **on demand** using **virtual, software-based applications** instead of today's physical nodes

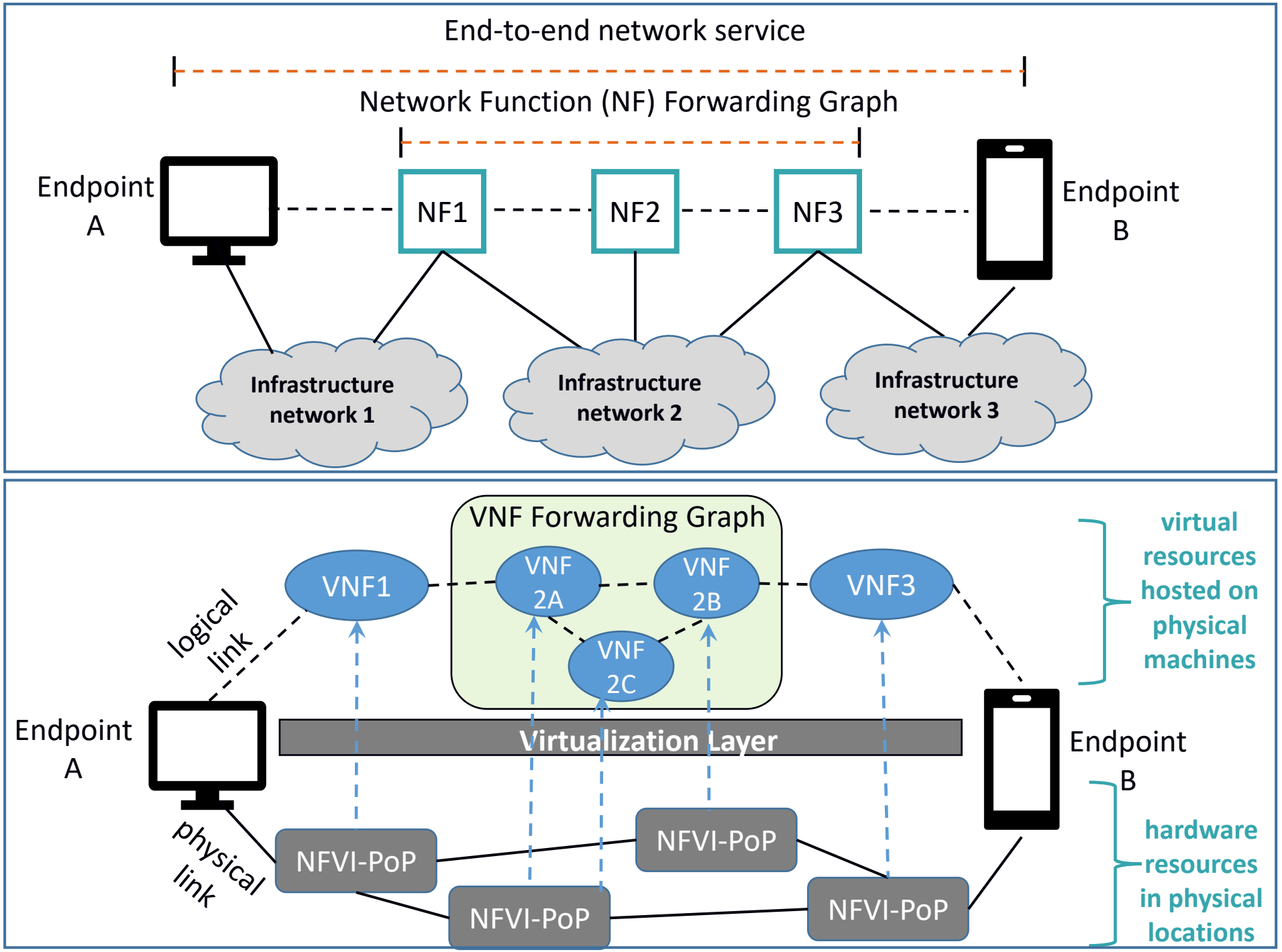
BENEFITS:

- Reduced equipment costs (CAPEX)
- Reduced operational costs (OPEX)
- Improved operational efficiency
- Increased speed of time to market
- Allows a single platform for different applications and users
- Increased flexibility (rapidly provision new services)

NFV supports:

- **virtualization:** use network resources without worrying about where they are physically located, how they are organized, etc.
- **orchestration:** manage thousands of devices
- **programmability:** should be able to change behavior on the fly
- **performance:** optimize network device utilization
- **multi-tenancy:** slice the network for different customers
- **visibility:** monitor resources, connectivity

Example of the use of NFV: an end-to-end network service with VNFs



Key NFV principles involved in creating network services

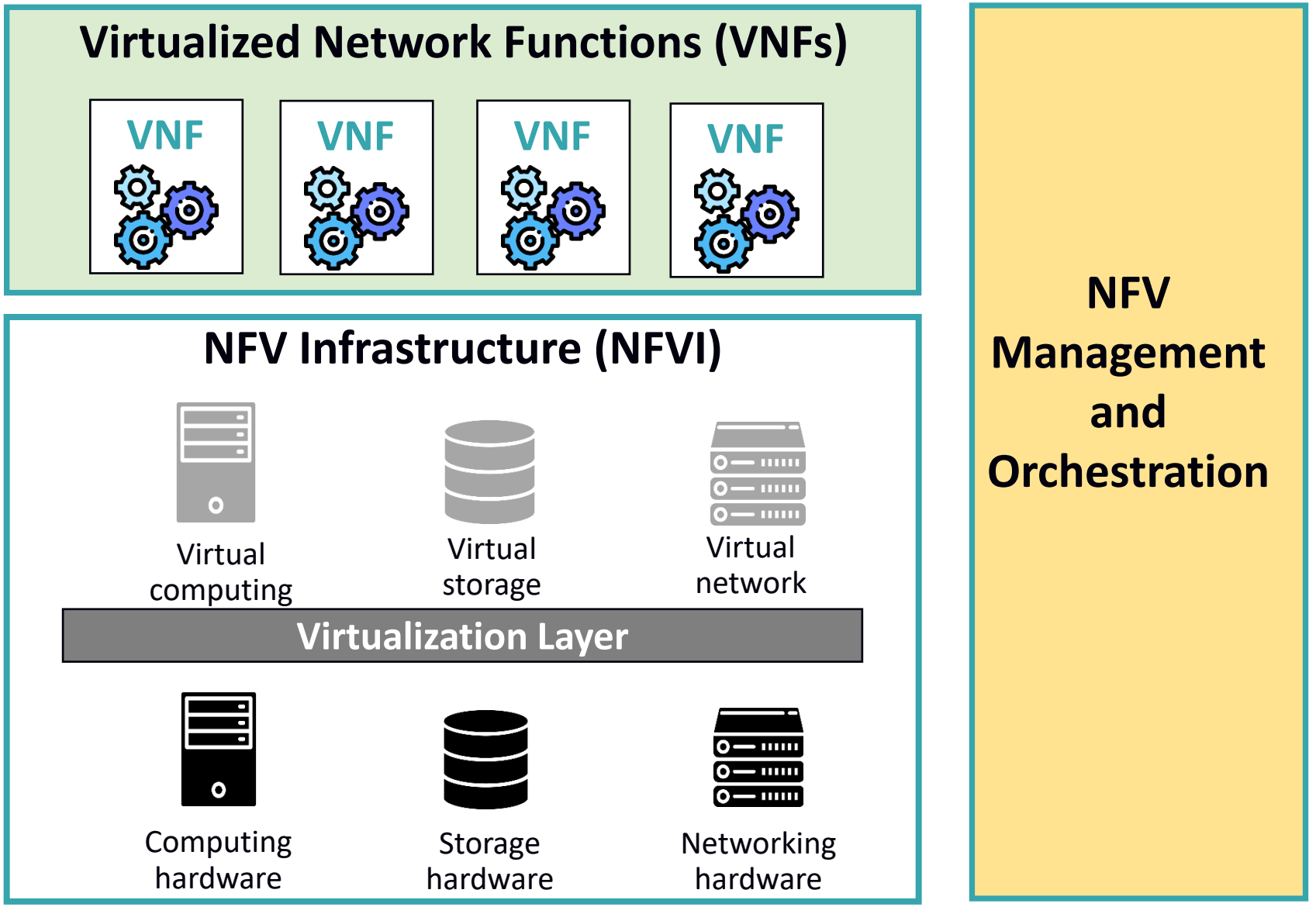
- **Service chaining:**
  - each VNF provides limited functionality on its own
  - VNFs are building blocks used to create end-to-end network services
  - for a given traffic flow within an application, a provider steers the flow through multiple VNFs to achieve a desired functionality → VNF forwarding graph (*e.g., pass through a firewall, NAT, and load balancer*)

- **Distributed architecture:**
  - a VNF can be made of one or more VNF components (VNFC)
  - each VNFC can be deployed in one or more instance
  - instances may be deployed on separate, distributed hosts to provide scalability and redundancy

Key definitions

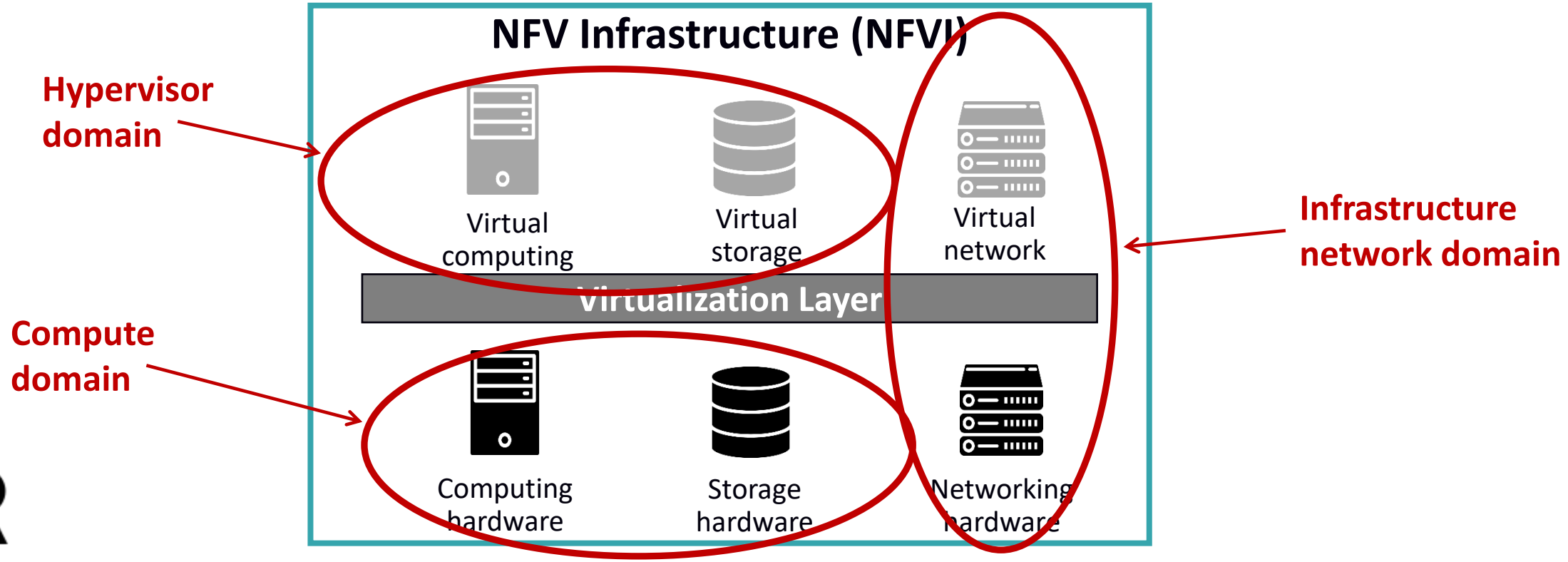
- **Network Function (NF)**: function within a network infrastructure which has well defined external interfaces and well defined functional behavior (i.e., what is defined as a network node today)
- **Network Functions Virtualization (NFV)**: principle of separating network functions from the hardware they run on by using virtual hardware abstraction
- **Network Functions Virtualization Infrastructure (NFVI):** totality of all hardware and software components that build up the environment in which VNFs are deployed
- **Virtualized Network Function (VNF)**: implementation of an NF that can be deployed on an NFVI
- **NFV Management and Orchestration (MANO)**: orchestration and lifecycle management of physical and/or software resources that support the infrastructure virtualization, and the lifecycle management of VNFs

High-level NFV Framework



NFV Infrastructure (NFVI)

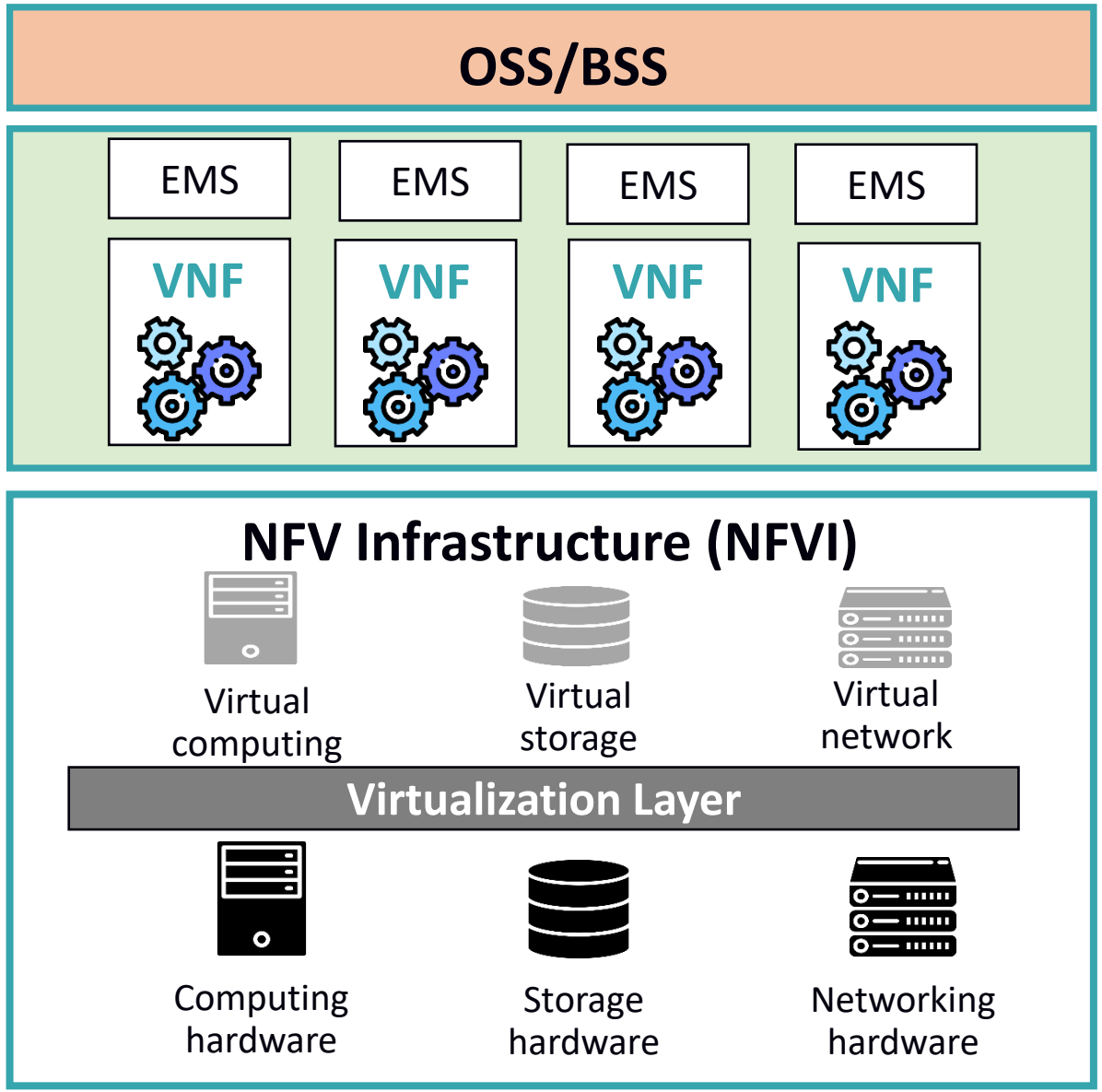
- all HW and SW components that build up the environment in which VNFs are deployed
- the NFVI can span across several locations → multiple points of presence (PoPs)
- NFVI-PoP: a network point of presence where a network function can be deployed as a VNF
- the network providing connectivity between these locations is considered part of the NFVI



ETSI NFV Reference Architectural Framework

- **extends the previously shown High-level NFV Framework**
- focuses further on functional blocks, their interactions, and the changes that may occur within the operator's network as a result of the virtualization of networking functions.

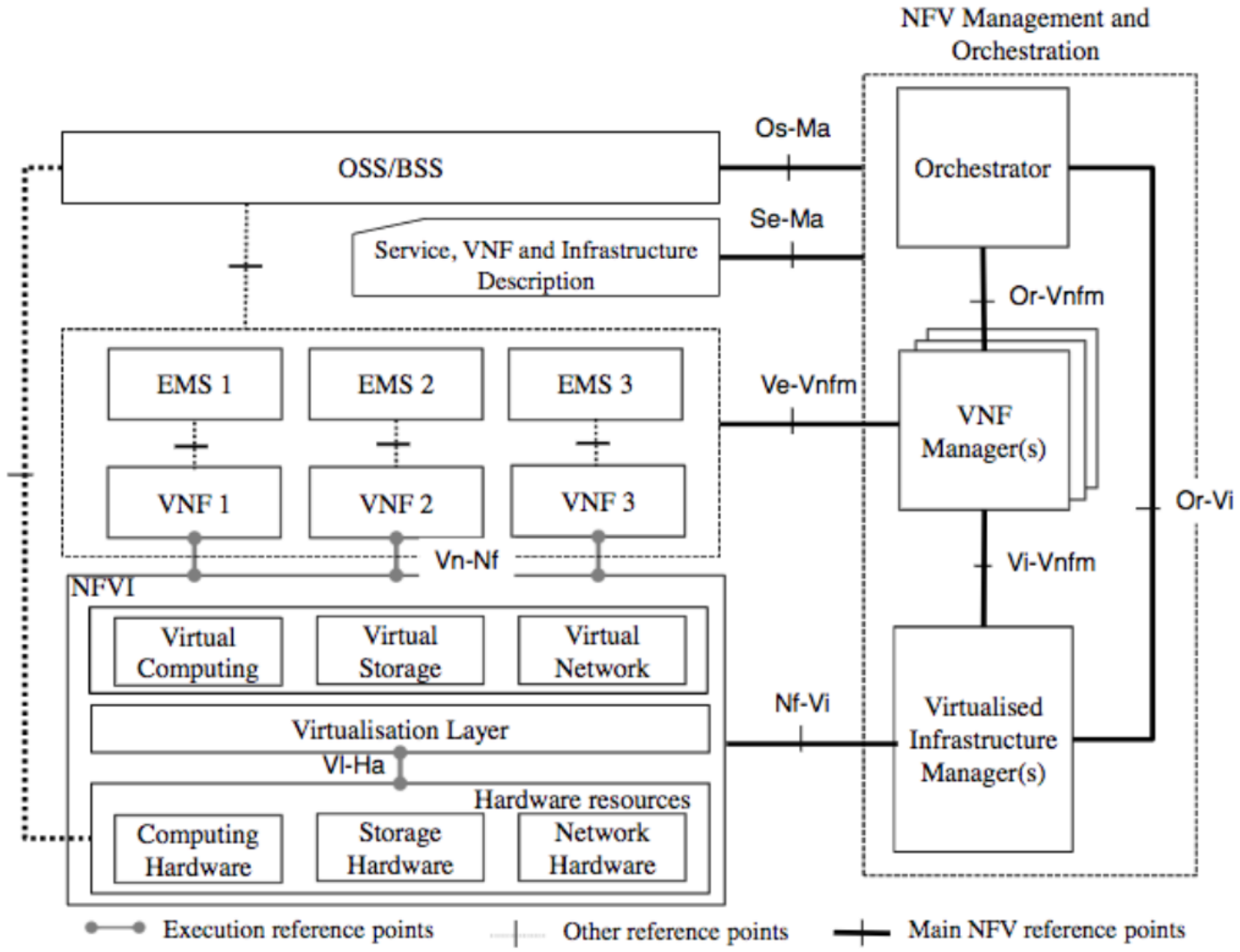
OSS: Operational Support Systems
BSS: Business Support Systems
EMS: Element Management System



NFV Management and Orchestration (MANO)

ETSI NFV Reference Architectural Framework

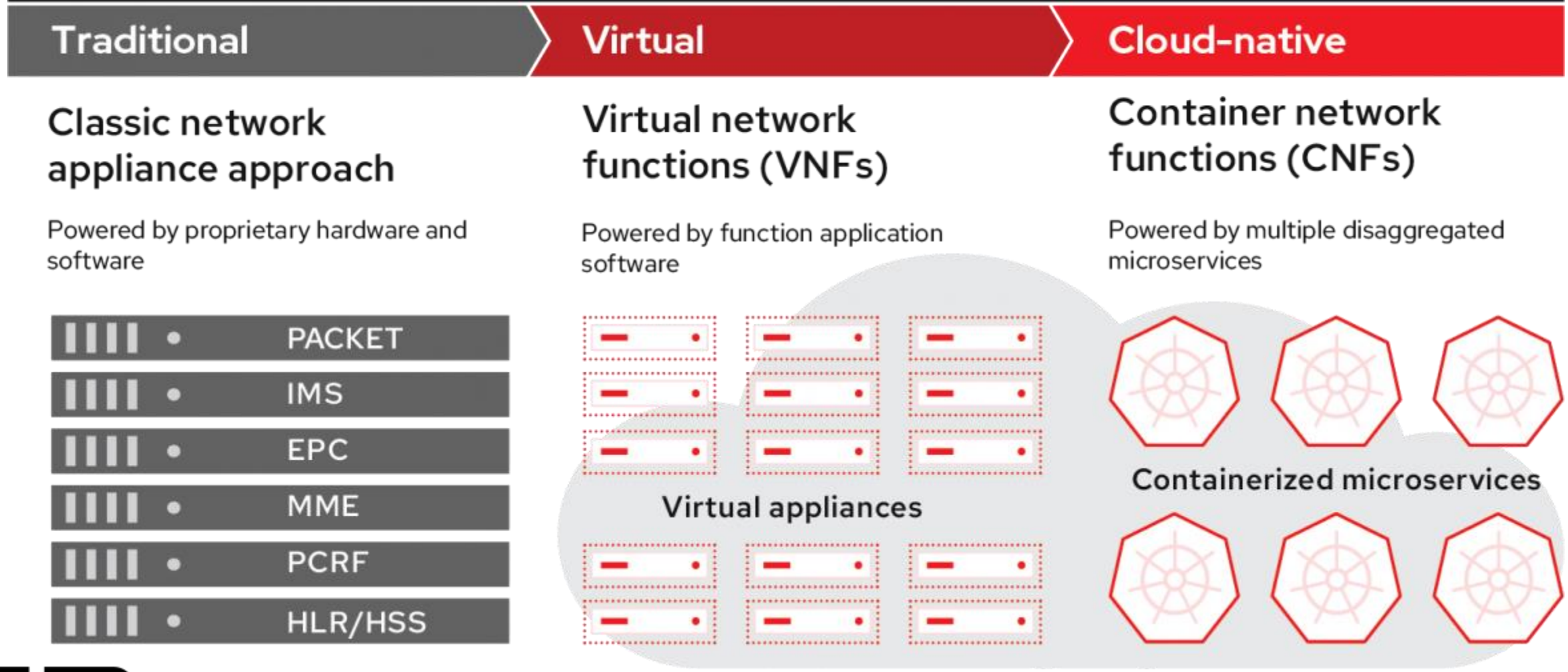
- **extends the previously shown High-level NFV Framework**
- focuses further on functional blocks, their interactions, and the changes that may occur within the operator's network as a result of the virtualization of networking functions.



Evolution of network functions

CNFs: also referred to as Cloud-native Network Functions !

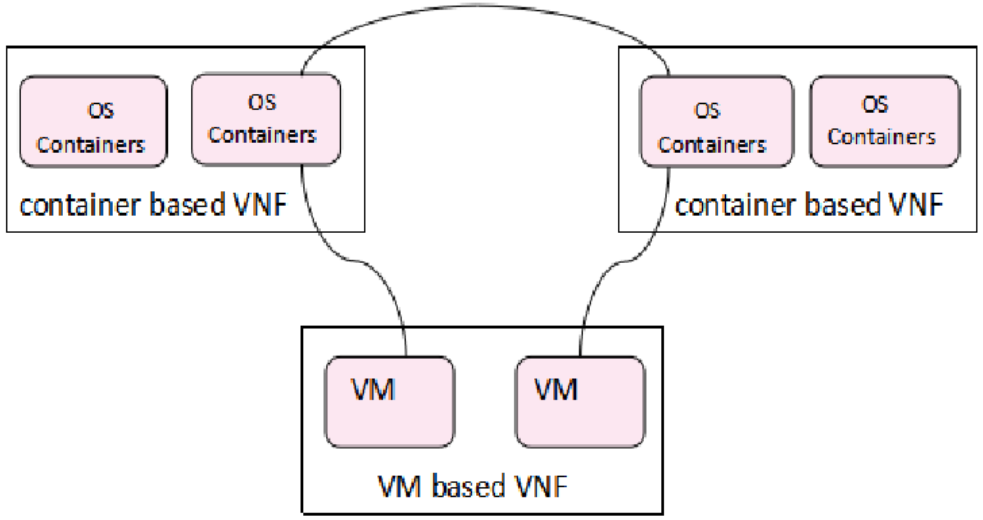
Figure: The journey of network functions in telecommunications



https://www.redhat.com/en/topics/cloud-native-apps/vnf-and-cnf-whats-the-difference

VM-based VNF vs. Container-based VNF

- in an NFV system, a Network Service can consist of multiple VNFs
- A VNF can be container-based, VM-based, or hybrid (some VNF components implemented in containers and others in VMs)
- communication should be enabled, regardless of type
- CNF (Cloud-native Network Function): a particular type of VNF that is designed, deployed and managed using cloud native technologies → i.e., it is deployed using container-based technologies



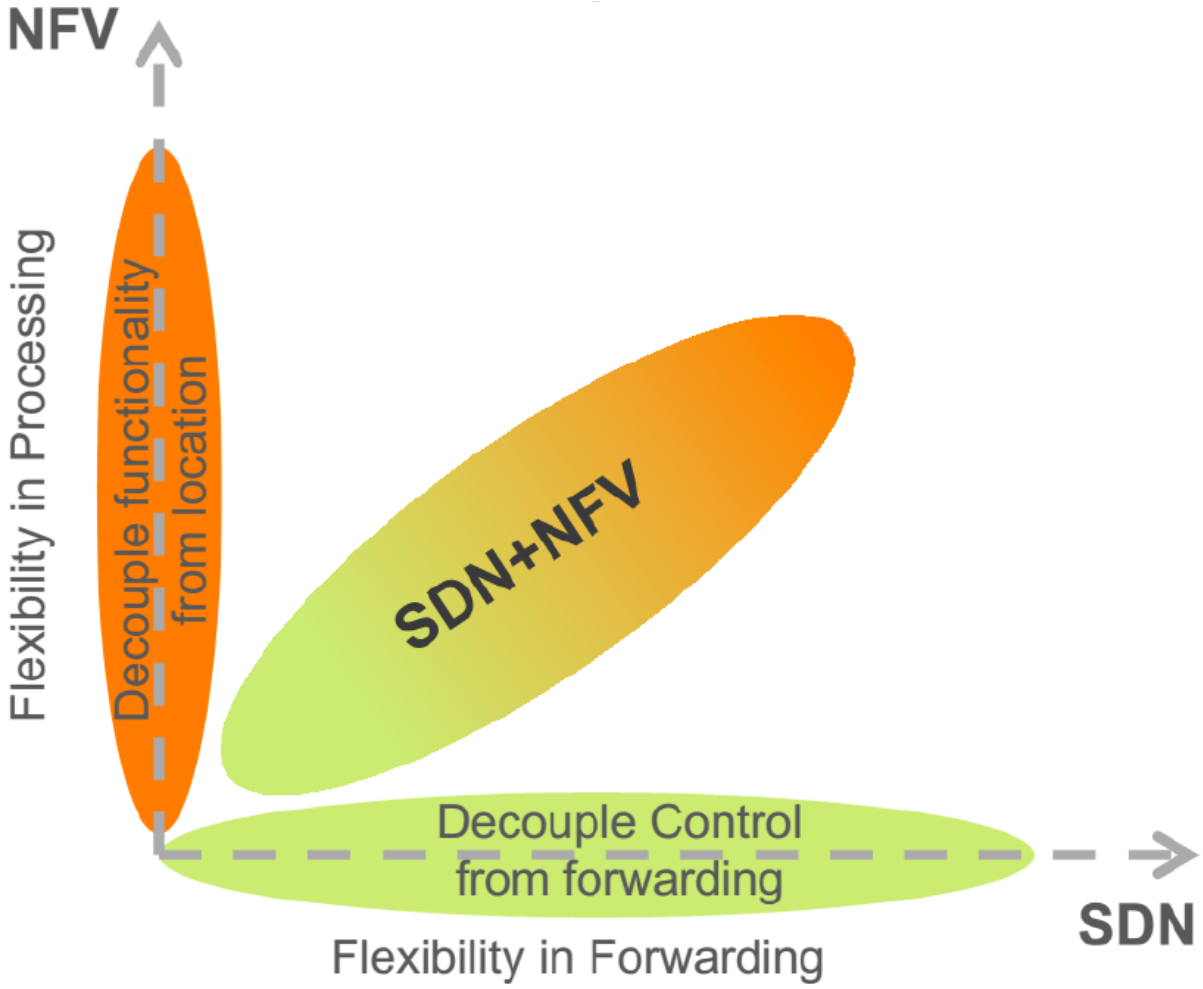
https://www.etsi.org/deliver/etsi_gr/NFV-IFA/001_099/038/04.01.01_60/gr_NFV-IFA038v040101p.pdf

Bringing together SDN and NFV

SDN and NFV (reminder from previous lectures)

SDN: separates control and user planes

NFV: separates software (applications) from hardware



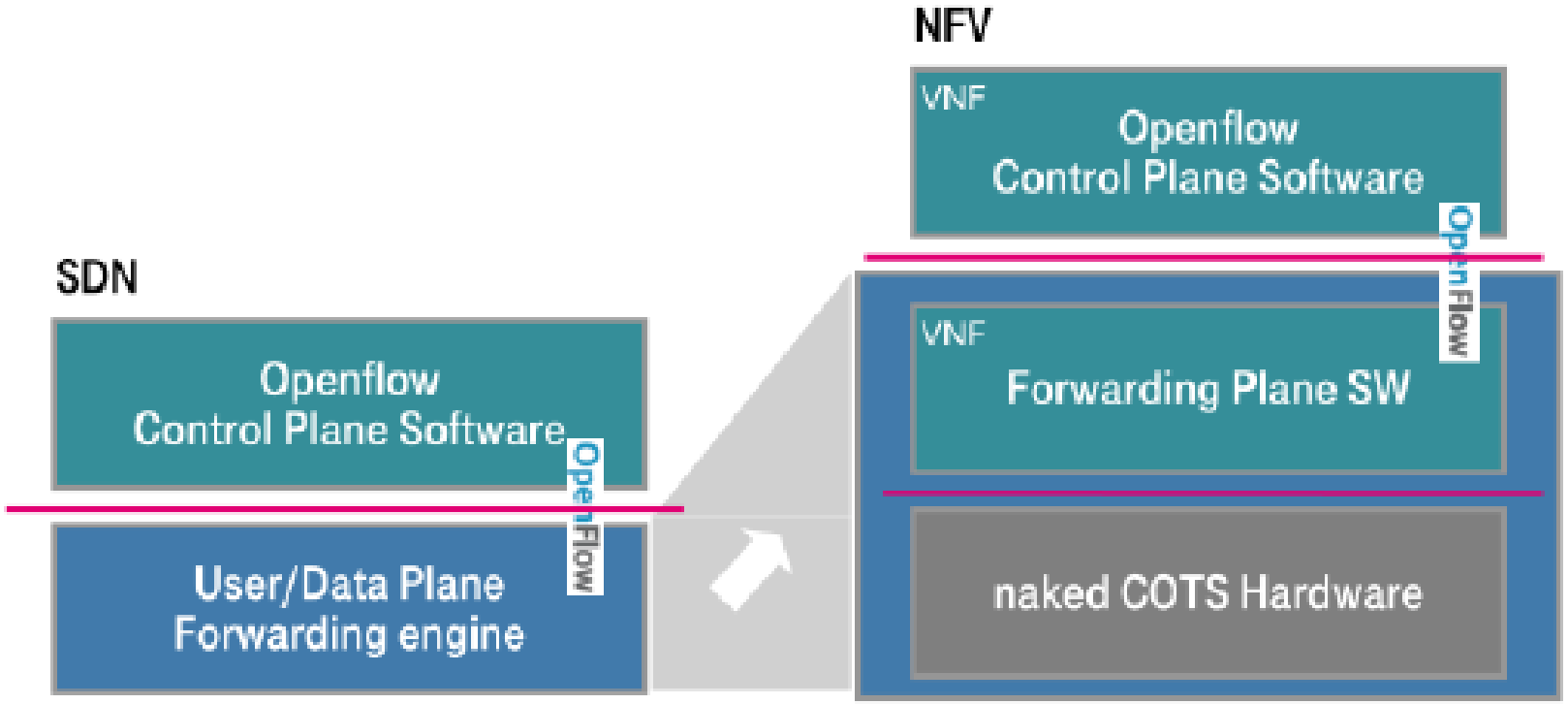
https://www.dca.fee.unicamp.br/~chesteve/ppt/SIGCOMM16-Tutorial-5G-SDN-NFV-part1.pdf

SDN and NFV (reminder from previous lectures)

SDN: flexible forwarding of traffic in a physical or virtual network environment

NFV: flexible placement of virtualized network functions across the network and cloud

→ complementary approaches



SDN and NFV

- SDN can be seen as an ***enabler*** for NFV
- Major challenge for NFV: enable configuration of a network so that VNFs running on servers are connected to the network at the appropriate place, with approapriate connectivity to other VNFs, and with desired QoS
- SDN: enables users and orchestration software to dynamically configure the network and the distribution and connectivity of VNFs.
  - Without SDN, NFV requires much more manual intervention

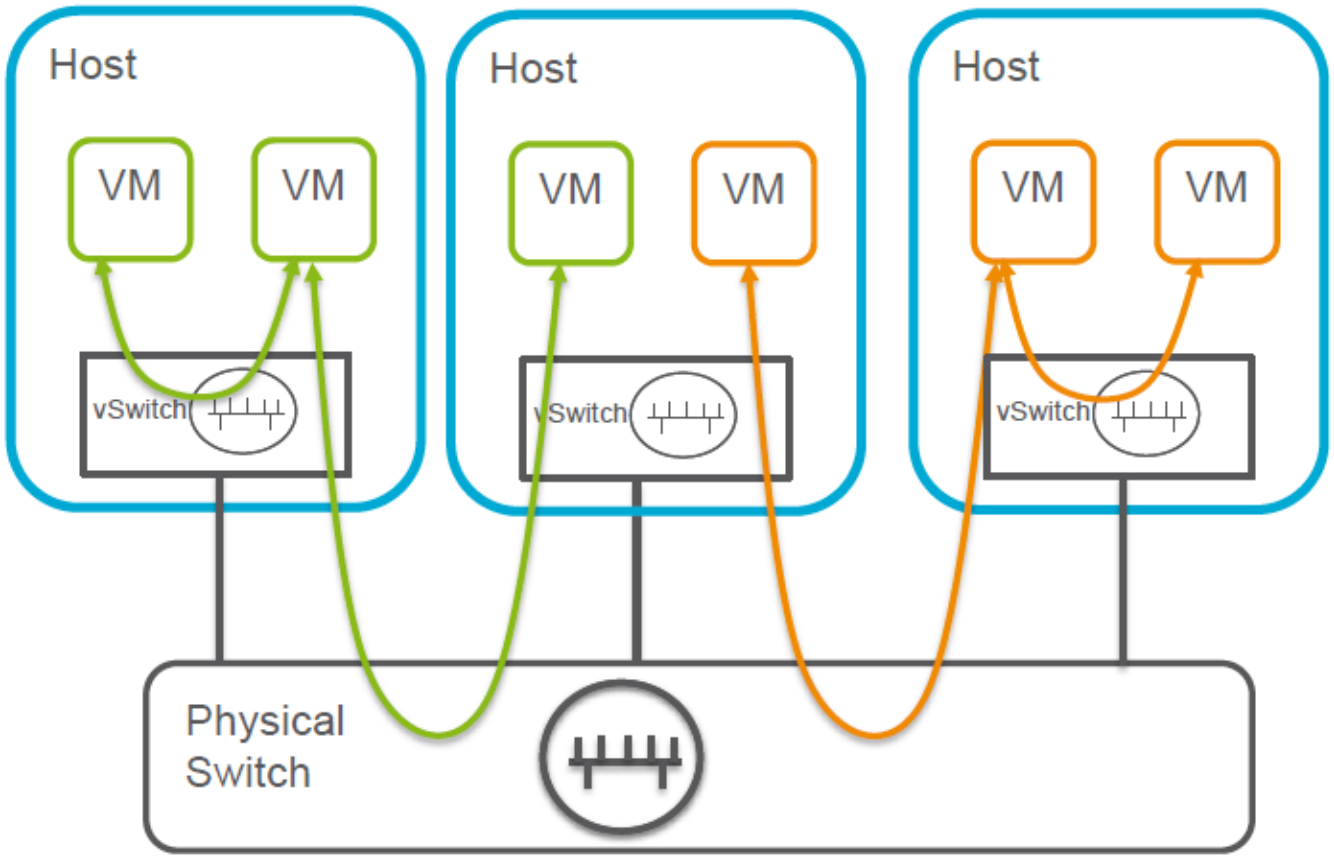
Network Virtualization

Virtual networks

- Virtual network: abstraction of physical network resources as seen by some upper layer software
  - Same physical networking infrastructure shared among isolated virtual apps
  - Users of a single virtual network are not aware of the details of the underlying physical network or of other virtual network traffic sharing the physical network resources
- Two common approaches for creating virtual networks
  1. protocol-based methods that define virtual networks based on fields in protocol headers (VLAN – layer 2, VPN – layer 3)
  2. VM-based methods, in which networks are created among a set of VMs by the hypervisor

Virtual switch

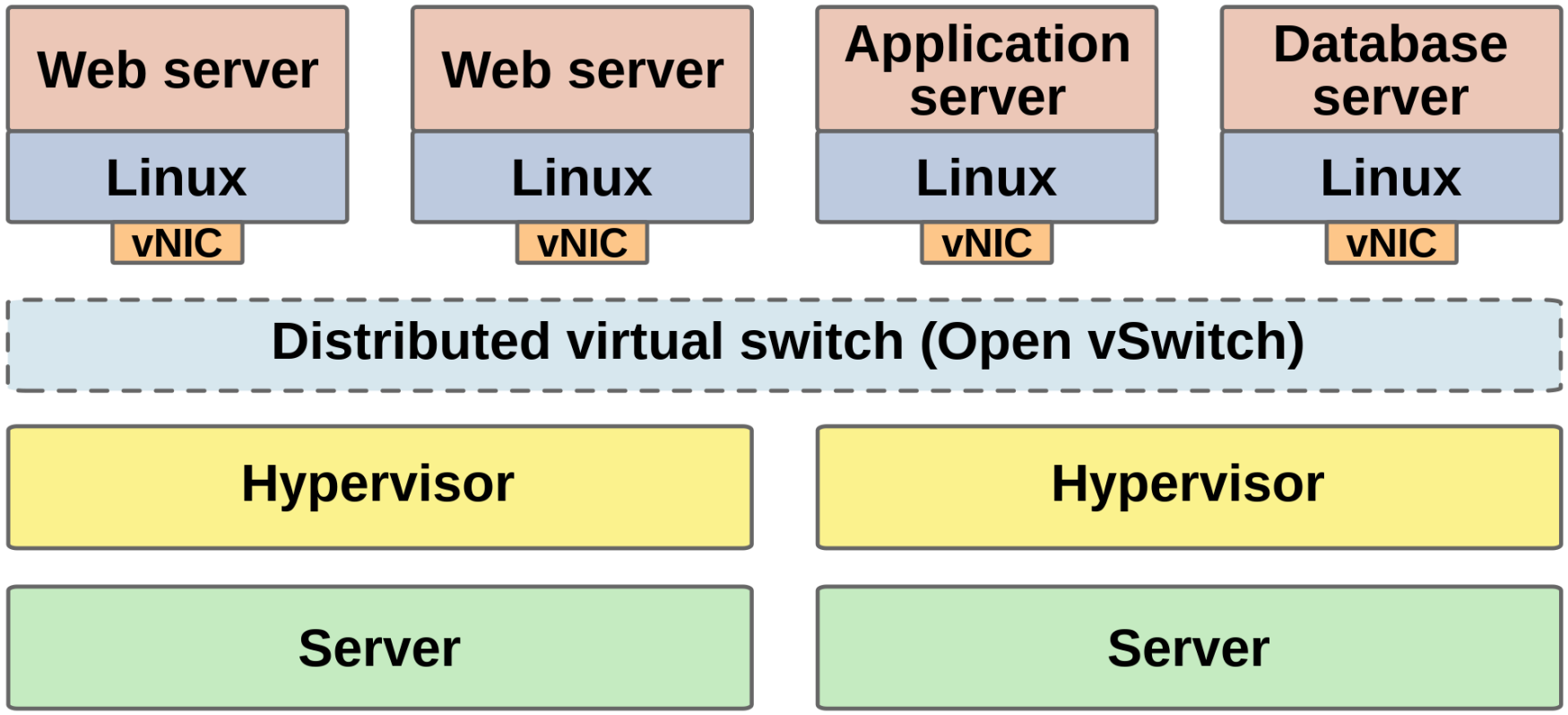
- interconnects virtual NICs of VMs with each other and with the NIC of the compute node
- vSwitch allows a VM to communicate with another VM
- A vSwitch forwards traffic between different VMs
  - on the same physical host,
  - on different hosts through the physical network
- A vSwitch typically runs as a software-based switch in the hypervisor
- Also isolates traffic beteen tenants (**green** and **orange** tenants in the picture)



source: Ericsson

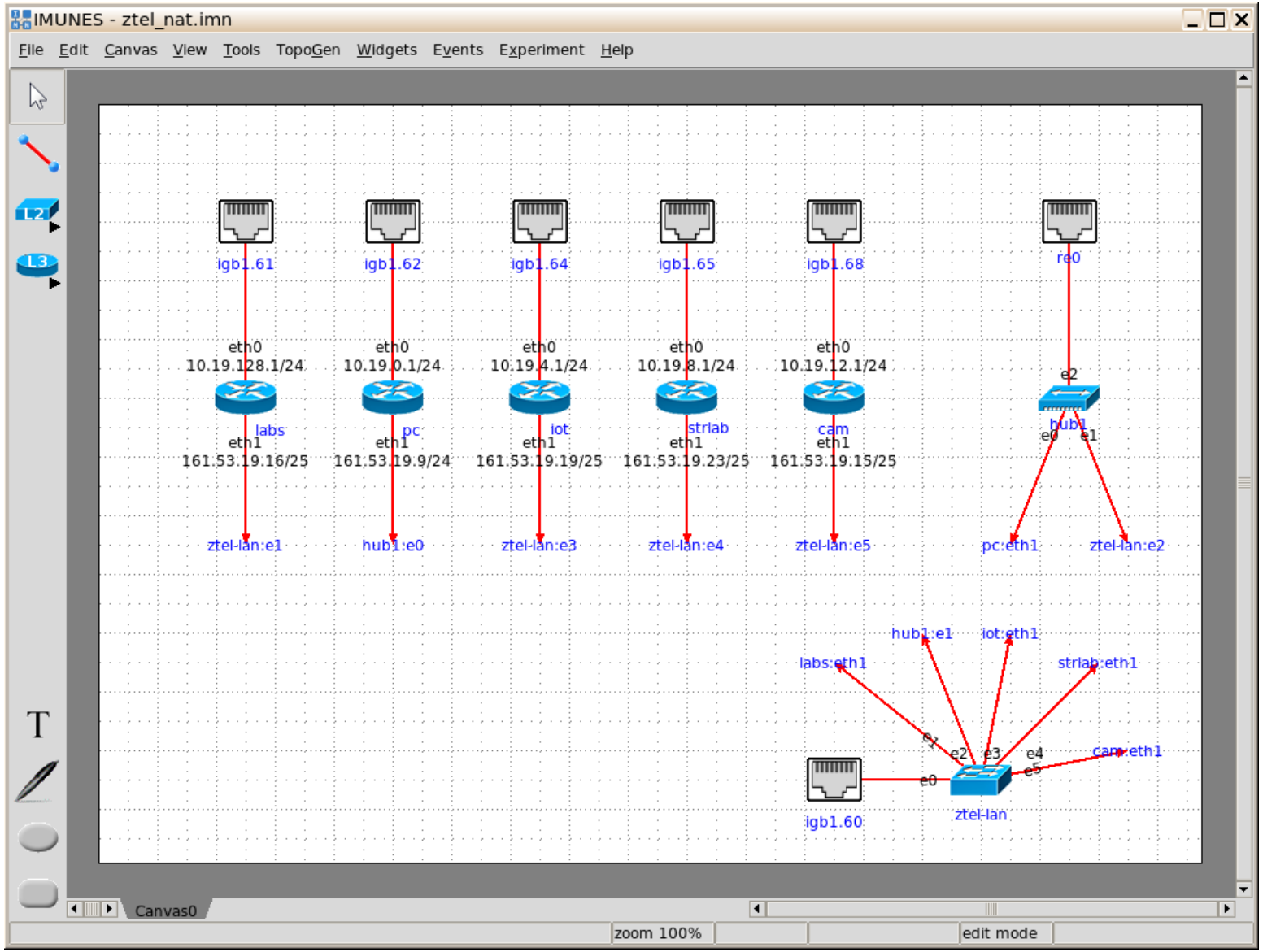
Example network virtualization: Open vSwitch

- open source implementation of a distributed virtual multilayer switch - designed to be used as a vswitch in virtualized server environments



Example: Dept. of Telecommunications

- Department NAT
- one physical node hosts 5 virtual routers
- separates private department networks



Some standards-developing organizations and initiatives…

- **ETSI: European Telecommunications Standards Institute**
  - taken lead role in defining standards for NFV (since 2012)
- **ONF: Open Networking Foundation**
  - industry consortium (networking-equipment vendors, semiconductor companies, computer companies, software companies, telecom service providers, data-center operators…) dedicated to the promotion and adoption of SDN through open standards development
- **Open development initiatives**
  - E.g., Cloud Native Computing Foundation, OpenStack, OpenDaylight…

Some useful videos:

- Virtualization explained
- Containerization explained
- Containers vs VMs: What's the difference?
- What is a container?
- Container orchestration explained