Network Function Virtualization and Software Defined Networking Cooperation

Eugen Borcoci

University POLITEHNICA Bucharest,

Eugen.Borcoci@elcom.pub.ro

InfoSys 2015 Conference May 24- 29, 2015, Rome

Acknowledgement

This overview is compiled, based on several public documents belonging to different authors' and groups working on NFV, SDN and Future Internet: conferences material, studies, research papers, standards, projects, overviews, tutorials, etc. (see Reference list).

Network Function Virtualization and Software Defined Networking Cooperation

Motivation of this talk

- Future Generation Networks/Internet - challenges -> need to solve the current networks limitation and ossification- as to support global integration of various forms of communications and better serve the large range of today's services and applications

  - **Evolutionary approach**
  - **Clean slate approach**
  - **Combined solutions**
  - **Novel significant trends**
    Network Functions Virtualization (NFV)
    Software Defined Networking (SDN)
    Cloud computing (CC)
    Information Centric Networking (ICN)

Network Function Virtualization and Software Defined Networking Cooperation

Motivation of this talk (cont'd)

- **Network Function Virtualization**

- **NFV objectives**:
  - Improved capital efficiencies vs. dedicated HW implementation solutions, by:
    - Using COTS HW to provide **Virtualized Network Functions (VNFs)** through **SW virtualization techniques**
    - Sharing of HW and reducing the number of different HW architectures

- **Improved flexibility in assigning VNFs to HW**
  - better scalability
  - decouples functionality from location
  - enables time of day reuse
  - enhance resilience through **Virtualization**, and facilitates resource sharing

- **Rapid service innovation** through SW -based service deployment

- Common automation and operating procedures $\Rightarrow$ **Improved operational efficiencies**

- **Reduced power usage**
  - (migrating workloads and powering down unused HW)

- **Standardized and open I/Fs**: between VNFs infrastructure and mgmt. entities

Network Function Virtualization and Software Defined Networking Cooperation

Motivation of this talk (cont'd)

- **Software Defined Networking (SDN)**
- SDN objectives:
  - **Separation of the Control Plane (CPl) from the Data Plane (DPl)**
  - A centralized control and view of the network
    - underlying network infrastructure is abstracted from the applications.
  - Open ("south") interfaces between the CPl (controllers) and DPl elements.
  - Network programmability: by external applications including network management and control (open "north" interfaces)
  - Independency of operators w.r.t. Network equipment vendors
  - Technology to be used in Cloud data centers as well in WANs

  - ***OpenFlow*** : typical ("vertical") protocol for communication between DPl and CPl

Network Function Virtualization and Software Defined Networking Cooperation

Motivation of this talk (cont'd)

- **NFV- SDN cooperation**
- **Recognized :** highly synergistic nature of NFV and SDN
- Both build on the rapid evolution of IT and cloud technologies
- **SDN features as:**
  - separation CPI/DPI (to form a domain-wide view of a network),
  - ability to abstract and programmatically control network resources
- **fit nicely into the NFV paradigm** ⇒ SDN can play a significant role in the orchestration of the NFV Infrastructure resources (both physical and virtual) enabling :
  - provisioning and configuration of network connectivity and bandwidth
  - automation of operations
  - security and policy control
- The SDN controller maps to the overall concept of network controller identified in the NFV architectural framework
- **However:**
  - **the majority of SPs and enterprises <u>still analyze</u> when and how to adopt NFV technologies; the current awareness in industry is rather low and lower than for SDN**

CONTENTS

CONTENTS

1. ⇨**Network Function Virtualization**
    1. NFV Actors
    2. NFV Approach
    3. High level view of NFV Use cases
    4. NFV Terminology (ETSI)
    5. NFV Requirements
    6. Network Services in NFV
    7. NFV Reference Architectural Framework
    8. NFV Use Cases

1. Network Function Virtualization

- **1.1 NFV Actors**
- **ETSI NFV Group**
  - **Global (operators-initiated)** ***Industry Specification Group (ISG)*** **under the auspices of ETSI**
  - ~200 members (2014)
  - –28 Tier-1 carriers (and mobile operators) & service providers, cable industry
- **Open membership**
  - ETSI members sign the "Member Agreement"
  - Non-ETSI members sign the "Participant Agreement"
- Operates by consensus (formal voting only when required)
- Deliverables: requirements specifications, architectural framework, PoCs, standards liaisons

- Face-to-face meetings quarterly
- **Currently: four (4) WGs, two (2) expert groups (EGs), 4 root-level work items (WIs)**
  - WG1: Infrastructure Architecture
  - WG2: Management and Orchestration
  - WG3: Software Architecture
  - WG4: Reliability & Availability
  - EG1: Security
  - EG2: Performance &
- Network Operators Council (NOC): technical advisory body

1. Network Function Virtualization

- **1.1 NFV Actors**

- **Open Networking Foundation (ONF)**
  - Active also in NFV area
    - E.g. of document: “OpenFlow-Enabled SDN and Network Functions Virtualization,” 2014, see References
    - Plan to more close cooperation with ETSI

- **Internet Research Task Force (IRTF)**
  - RFC 7426, Jan 2015: “Software-Defined Networking (SDN): Layers and Architecture Terminology” , see Refs.
  - proposes a common terminology for SDN layering and architecture based on significant related work from the SDN research community.

1. Network Function Virtualization

- **1.2 NFV Approach**

- Network services are provisioned differently w.r.t current networks practice

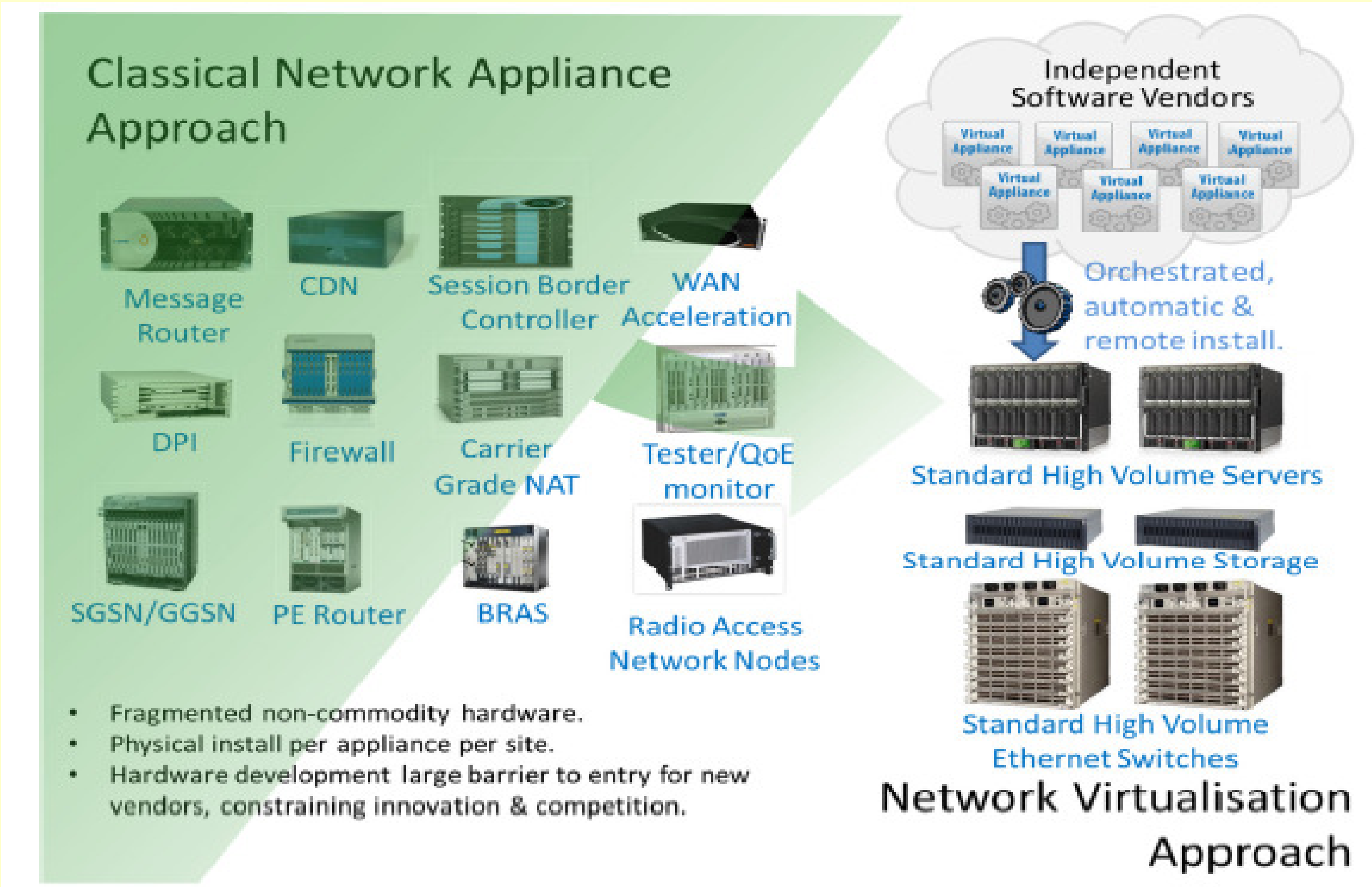
  - **Decoupling SW from HW**
    - network element is no longer a collection of integrated HW@SW entities ⇒ they may ***evolve independently***

  - **Flexible network function deployment:**
    - The SW/HW detachment allows to reassign and share the infrastructure resources
    - HW and SW can perform different functions at various times
    - The pool of HW resources is already in place and installed at some **NFVI-PoPs** ⇒ the actual NF SW instantiation ***can be automated***
      - leverages the different cloud and network technologies currently available
      - helps NOs to ***faster deploy new network services*** over the same physical platform

  - **Dynamic operation**
    - network function are performed by ***instantiable SW components*** ⇒
      - greater flexibility to scale the actual VNF performance in a more dynamic way
      - finer granularity, e.g., according to the actual traffic

1. Network Function Virtualization

- **1.2 NFV Approach (cont'd)**
- **NFV vision (source : ETSI)**



1. Network Function Virtualization

- 1.2 NFV Approach (cont'd)

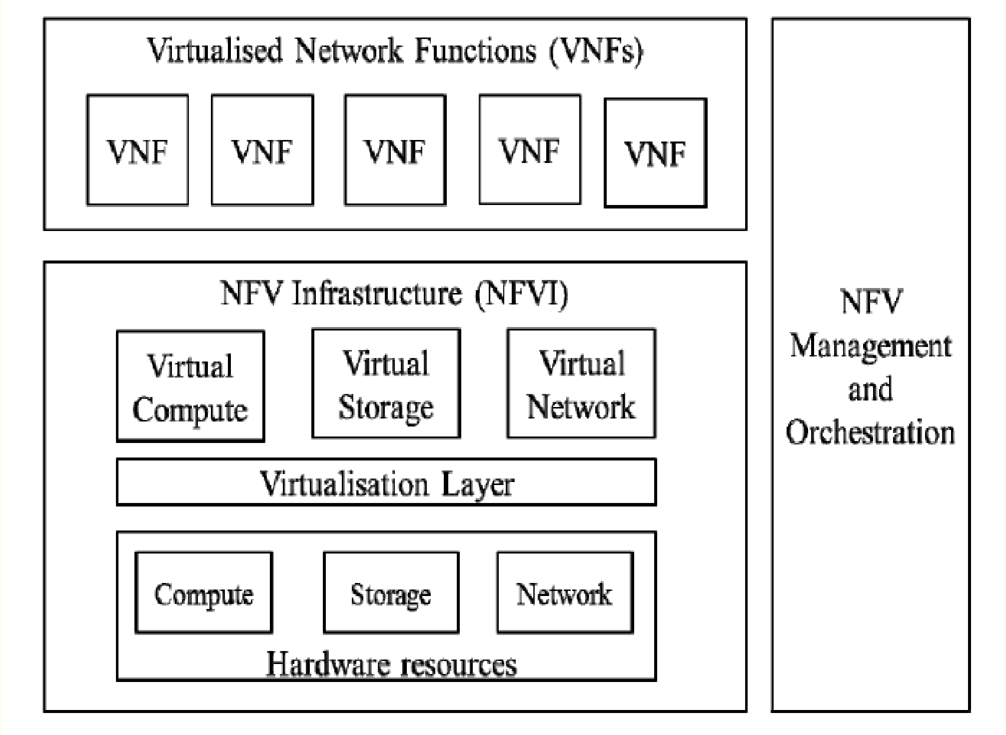
- **High level view of NFV framework**

- **Working domains**
- **VNF,** as the SW implementation of a NF which is capable of running over the NFVI.

- **NFV Infrastructure (NFVI),** including the diversity of physical resources and how these can be virtualized.
  - NFVI supports the execution of the VNFs.

- **NFV Management and Orchestration (NFV-MANO)**
  - orchestration and lifecycle management of physical and/or SW resources that support the infrastructure virtualization, and the VNFs lifecycle management

  - NFV MANO focuses on all virtualization-specific management tasks



Source: ETSI

1. Network Function Virtualization

- **1.2 NFV Approach (cont'd)**
- **High level view of NFV framework**

- The NFV framework enables
  - dynamic construction and management of VNF instances
  - and the relationships between them regarding data, control, management, dependencies and other attributes.

- **Three architectural views** (perspectives) of VNFs
  - *Virtualization deployment/on-boarding perspective*; the context can be a Virtual Machine (VM)

  - *vendor-developed SW package perspective* ; the context - several inter-connected VMs and a deployment template describing their attribute

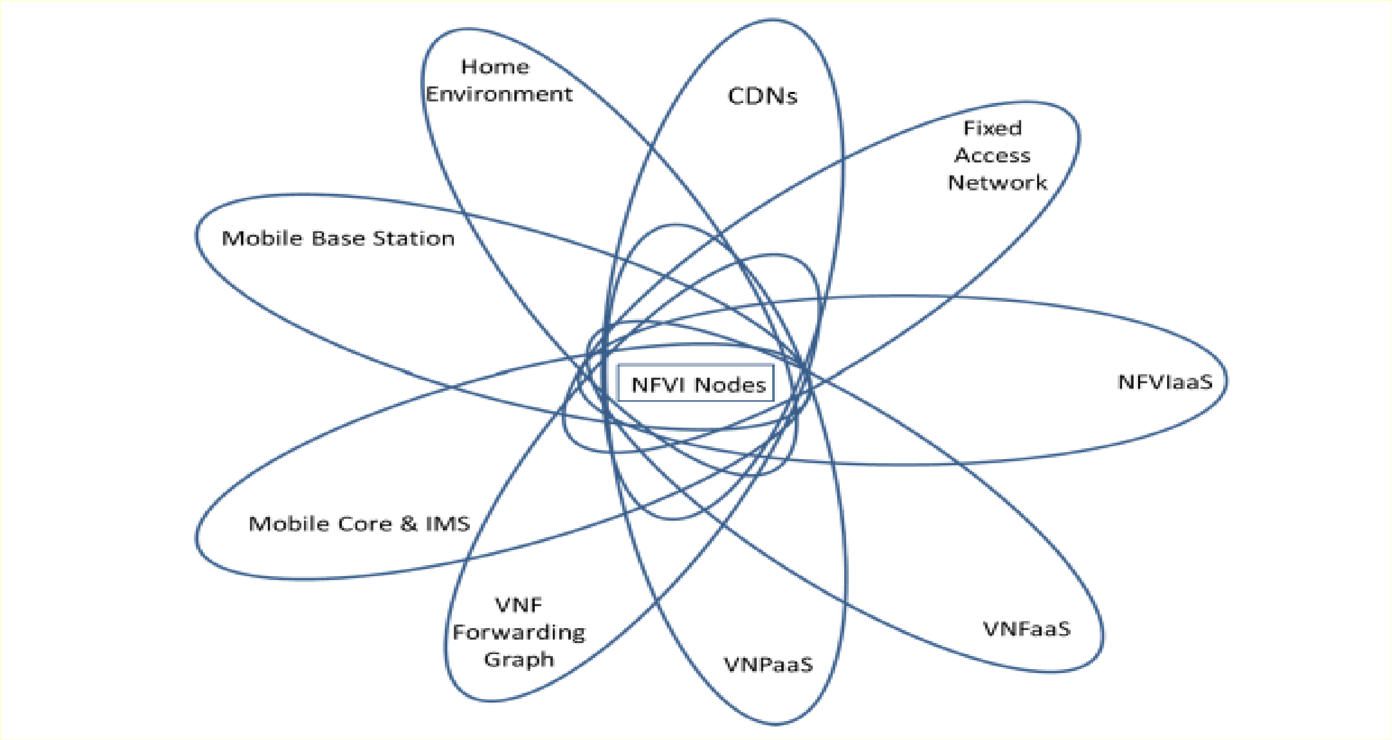
  - operator perspective; the context - operation and management of a VNF received in the form of a vendor software package

1. Network Function Virtualization

- **1.2 NFV Approach (cont'd)**
- **High level view of NFV framework**
- Within each of the above contexts, at least the following relations exist between VNFs:
  - **VNF Forwarding Graph (VNF-FG)** covers the case where network connectivity between VNFs is specified, e.g. a chain of VNFs on the path to a web server tier
  - **VNF Set** covers the case where the connectivity between VNFs is not specified
- Within a VNF-FG, the sequences of VNFs to be traversed by traffic flows are either decided at design time of the VNF-FG or at runtime

1. Network Function Virtualization

- ## 1.3 High level view of NFV Use cases
  - **Source:** Network Functions Virtualization – White Paper #3, https://portal.etsi.org/Portals/0/TBpages/NFV/Docs/NFV_White_Paper3.pdf



1. Network Function Virtualization

- **1.3 High level view of NFV Use cases (cont'd)**
  - **NFV Use Cases – defined by ETSI**

Cloud Use Cases	Mobile Use Cases	Data Center Use Cases	Access/ Residential
NFV Infrastructure as a Service (IaaS) (NFVIaaS)	Virtualization of the Mobile Core Network and IMS	Virtualization of CDNs	Virtualization of the Home Environment
Virtual Network Functions as a Service (VNFaaS)	Virtualization of Mobile Base Station		Fixed Access Network Functions Virtualization
Service Chairs (VNF Forwarding Graphs)			
Virtual Network Platform as a Service (VNPaaS)			

1. Network Function Virtualization

- **1.3 High level view of NFV Use cases (cont'd)**
- **Examples of network functions supported by NFV [See Refs.: A.Reid].**

  - **Switching elements** such as Broadband Remote Access Server (BRAS) or Broadband Network Gateway (BNG), carrier grade NAT, and routers
  - **Mobile network nodes** such as :
    - Home Location Register/Home Subscriber Server (HLR/HSS),
    - Serving GPRS Support Node (SGSN)
    - Mobility Management Entity (SGSN-MME)
    - Gateway GPRS support node/Packet Data Network Gateway (GGSN/PDN-GW)
    - RNC
    - NodeB and Evolved Node B (eNodeB) in 4G
  - **Functions in home routers and set top boxes**
  - **Virtualized home environments**
  - **Tunneling gateway elements** such as IPSec/SSL VPN gateways
  - **Traffic analysis elements** such as Deep Packet Inspection (DPI)
  - **Quality of Experience (QoE) measurement**

1. Network Function Virtualization

- **1.3 High level view of NFV Use cases (cont'd)**
- **Examples of network functions supported by NFV [See Refs.: Reid].**
  - **Service Assurance, Service Level Agreement (SLA)** monitoring, Test and Diagnostics
  - **Next-Generation Networks (NGN) signaling**
    - Session Border Controller (SBCs)
    - IP Multimedia Subsystem (IMS)
  - **Converged and network-wide functions** such as AAA servers, policy control and charging platforms.
  - **Application-level optimization**
    - Content delivery network (CDNs)
    - Cache Servers
    - Load Balancers
    - Application Accelerators
  - **Security functions**
    - Firewalls, virus scanners, intrusion detection systems, spam protection

1. Network Function Virtualization

- **1.4 NFV Terminology (ETSI)**

- **Main NFV terminology**
  - **FB – Functional Block**

- **Main Concepts: (NF, NFV, NFVI, VNF, NS, NFV-MANO)**

  - **Network Function (NF):** FB within a network infrastructure having well-defined external I/Fs and well-defined functional behaviour (today a NF is often a network node or physical appliance).

  - ***NFV: principle of separating NFs from the hardware they run on, by using virtual hardware abstraction***

  - **NFV Infrastructure (NFVI):** all HW&SW components building up the environment in which VNFs are deployed
    - The NFVI can span across several locations, e.g. places where data centres are operated
    - The network providing connectivity between these locations is regarded to be part of the NFVI

  - **Virtualized Network Function (VNF):** an NF implementation that can be deployed on NFVI

1. Network Function Virtualization

- **1.4 NFV Terminology (ETSI) (cont'd)**

- **Main NFV terminology (ETSI) (cont'd)**
- **Main Concepts: (NF, NFV, NFVI, VNF, NS, NFV-MANO)**

  - **NFVI component:** NFVI HW resource that is not field replaceable, but is distinguishable as a COTS component at manufacturing time

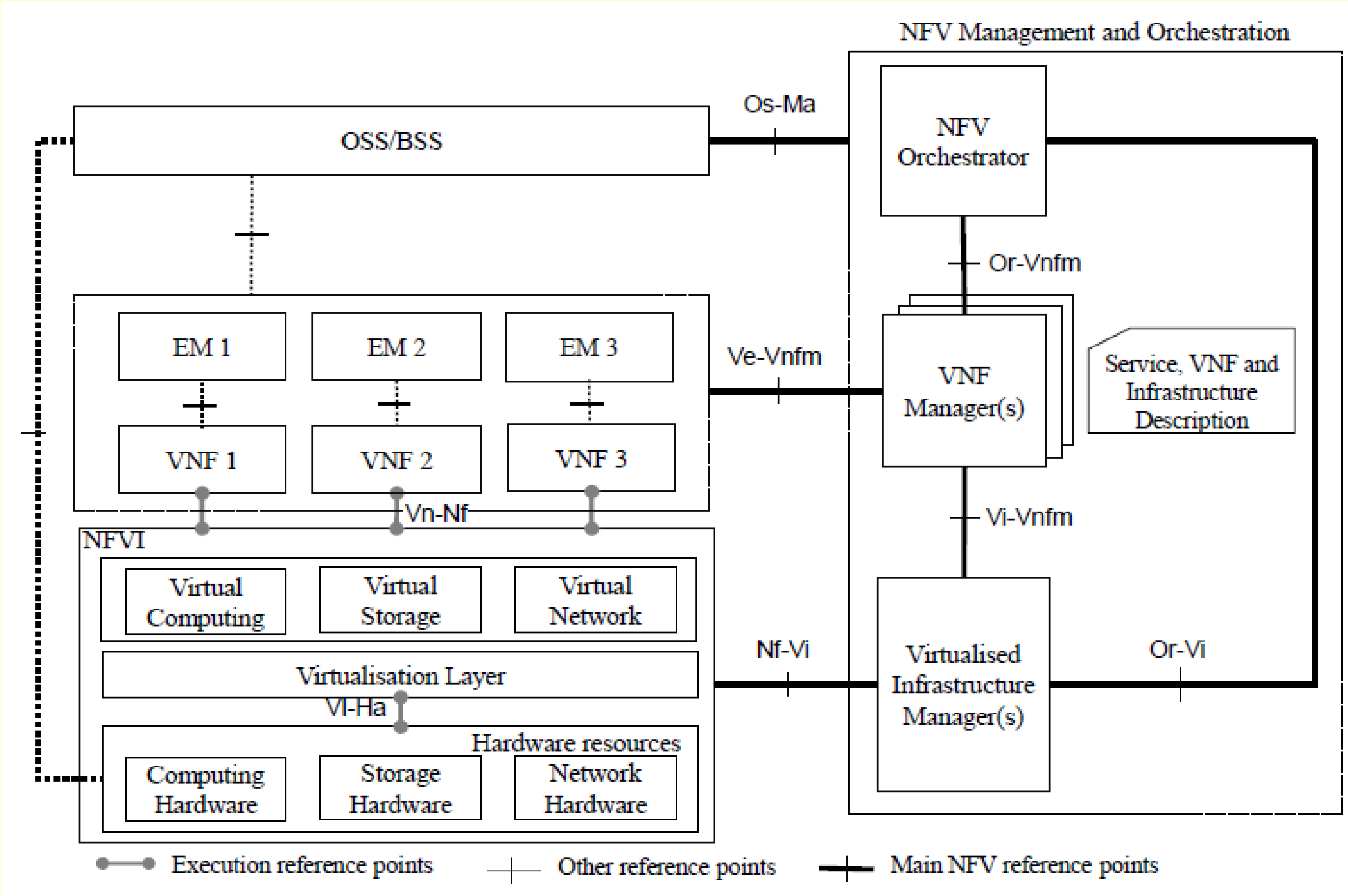
  - **Network Service (NS) :** composition of NFs defined by its functional and behavioural specification
    - The NS contributes to the behaviour of the higher layer service, which is characterized by at least performance, dependability, and security specifications.
    - (Individual NF behaviours + net. infrastructure composition mechanism) ==> E2E NS behaviour

  - **NFV framework:** totality of all entities, reference points, information models and other constructs defined by the specifications published by the ETSI ISG NFV

1. Network Function Virtualization

1.4 NFV Terminology

- **NFV reference architectural framework Source: ETSI**
- Details - to be discussed in Section 1.7



1. Network Function Virtualization

- **1.4 NFV Terminology (ETSI) (cont'd)**
- **Specific NFV terminology (ETSI)**
  - **Compute domain:** domain within the NFVI including servers and storage
  - **Compute node**: abstract definition of a server
  - **Infrastructure network domain**: domain within the NFVI including all networking that interconnects compute/storage infrastructure
  - **Lifecycle management:** set of functions required to manage the instantiation, maintenance and termination of a VNF or NS
  - **Network forwarding path:** ordered list of connection points forming a chain of NFs, along with policies associated to the list
  - **NFVI-Node:** physical device[s] deployed and managed as a single entity, providing the NFVI Functions required to support the execution environment for VNFs
  - **NFVI Point of Presence (NFVI-PoP):** N-PoP where a NF is or could be deployed as Virtual Network Function (VNF)

1. Network Function Virtualization

- **1.4 NFV Terminology (ETSI) (cont'd)**

- **Specific NFV terminology (ETSI) (cont'd)**
  - **Network Functions Virtualization Management and Orchestration (NFV-MANO):** functions collectively provided by NFVO, VNFM, and VIM
    - VIM Virtualised Infrastructure Manager, VNFM Virtualised Network Function Manager
  - **NFV Orchestrator (NFVO):** FB aiming to optimize the resource allocation
    - manages the Network Service (NS) lifecycle
    - coordinates the management of:
      - NS lifecycle,
      - VNF lifecycle (supported by the VNFM)
      - NFVI resources (supported by the VIM)
  - **NFV-MANO Architectural Framework**
    - collection of all FBs (those in NFV-MANO category + others interworking with NFV-MANO)
    - data repositories used by these FBs
    - reference points and I/Fs of the FBs
  - **Network Point of Presence (N-PoP):** location where a NF is implemented as either a Physical Network Function (PNF) or a Virtual Network Function (VNF)

1. Network Function Virtualization

- **1.4 NFV Terminology (ETSI) (cont'd)**
- **Specific NFV terminology (ETSI) (cont'd)**

  - **Network service descriptor:** template describing
    - the NS deployment, including service topology (constituent VNFs, their relationships, Virtual Links, VNF Forwarding Graphs)
    - NS characteristics such as SLAs + other artefacts necessary for the NS on-boarding and lifecycle management of its instances

  - **Network service orchestration**: subset of NFVO functions, responsible for NS lifecycle management

  - **Network stability:** NFV framework ability to maintain stable and fast operation
    - while providing its function
    - to resume its designated behaviour as soon as possible under difficult conditions (overload, or other anomalies not exceeding the design limits)

  - **NF forwarding graph:** graph of logical links connecting NF nodes for the purpose of describing traffic flow between these NFs

  - **NFV-Resource (NFV-Res):** NFV-Resources existing inside the NFVI and usable by the VNF/VNSF to allow for their proper execution

1. Network Function Virtualization

- **1.4 NFV Terminology (ETSI) (cont'd)**

- **Specific NFV terminology (ETSI) (cont'd)**
  - **Network Interface Controller (NIC):** device in a compute node that provides a physical I/F with the infrastructure network
  - **Virtualized NIC (vNIC):** Virtualized NIC created for a VM by a hypervisor
  - **Physical Network Function (PNF):** implementation of a NF via a tightly coupled SW&HW system
  - **Virtual Application (VA):** piece of SW which can be loaded into a VM (A VNF is one type of VA)
  - **Virtual link (VL)** : set of connection points + their inter-connectivity relationships + any associated target perf. metrics (e.g. bandwidth, latency, QoS)
    - The VL can interconnect two or more entities (VNF components, VNFs, or PNFs) and it is supported by a Virtual Network (VN) of the NFVI

1. Network Function Virtualization

- **1.4 NFV Terminology (ETSI) (cont'd)**

- **Specific NFV terminology (ETSI) (cont'd)**
  - **Virtual Machine (VM):** virtualized computation environment behaving very much like a physical computer/server
    - It has its processor, memory/storage, interfaces/ports like of a Phy computer/server
    - Usually - it is generated by a Hypervisor, which partitions the underlying physical resources and allocates them to VMs
    - VMs are capable of hosting a VNF Component (VNFC)
  - **Virtual Network (VN):** it routes information among the network I/Fs of VM instances and physical network I/Fs providing connectivity
    - The virtual network is bounded by its set of permissible network I/Fs
  - **Virtualization container:** partition of a compute node that provides an isolated virtualized computation environment
    - Examples of virtualization container includes VM and OS container

1. Network Function Virtualization

- **1.4 NFV Terminology (ETSI) (cont'd)**
- **Specific NFV terminology (ETSI) (cont'd)**
  - **Virtualized CPU (vCPU): -** created for a VM by a hypervisor. In practice, a vCPU may be
    - a time sharing of a real CPU
    - In multi-core CPUs, it may be an allocation of one or more cores to a VM.
    - The hypervisor may *emulate* a CPU instruction set such that the
      - vCPU instruction set ≠ native CPU instruction set
      - emulation will significantly impact performance
  - **Virtualized Infrastructure Manager (VIM):** FB controlling and managing the NFVI *compute, storage and network resources*
    - usually within one operator's Infrastructure Domain (e.g. NFVI-PoP)
  - **VNF Instance:** run-time instantiation of the VNF SW
    - resulting after :completing its components instantiation and of the connectivity between them
    - using the VNF deployment and operational information captured in the VNFD, as well as additional run-time instance-specific information and constraints

1. Network Function Virtualization

- **1.4 NFV Terminology (ETSI) (cont'd)**

- **Specific NFV terminology (ETSI) (cont'd)**
  - **VNF Descriptor (VNFD):** configuration template that describes a VNF in terms of its deployment and operational behaviour
    - is used in the process of VNF on-boarding and managing the lifecycle of a VNF instance
  - **VNF Manager (VNFM):** FB responsible for the VNF lifecycle management
  - **Virtualized Storage (vStorage):** Virtualized non-volatile storage allocated to a VM
  - **Virtualized Switch (vSwitch):** Ethernet switch implemented by the hypervisor that interconnects vNICs of VMs with each other and with the NIC of the compute node
  - **VNF Forwarding Graph (VNF FG):** NF forwarding graph where at least one node is a VNF
  - **VNF Set:** collection of VNFs with unspecified connectivity between them

1. Network Function Virtualization

- **1.5 NFV Requirements**
- **General**
  - The NFV framework should be able to
    - permit SP/NOs to *partially* or *fully virtualize* the NFs needed to create, deploy and operate the services they provide
      - “Partial virtualization, e.g. : virtualization in CPI only, not in DPI
        Impact on the non-virtualized network should be manageable, predictable
        Impact on the management system of the legacy network should be manageable
    - support NSes composed of Phy NFs and VNFs as a VNF FG implemented across N-PoP multi-vendor environments
      - These can be instantiated by a single operator or in a cooperating environment
- **Portability**
  - NFV framework should be able to
    - load, execute, move VNFs across different std. N-PoP multivendor environments
    - support an I/F to VNF associated SW instances from the underlying infrastructure
    - provide and optimize location, reservation and allocation of the required resources of the VNFs

1. Network Function Virtualization

- **1.5 NFV Requirements (cont'd)**
- **Performance**
  - The NFV framework should be able to
    - appropriately Instantiate and configure any given VNF over the underlying infrastructure
    - describe the underlying infrastructure reqs. of a VNF, so that it can be sized for a given perf. target while the resources are allocated and isolated/shared accordingly
    - collect for any running VNF instance, all its perf. Information
    - collect performance information on resource usage at infrastructure level (e.g. hypervisor, NIC, virtual switch)

1. Network Function Virtualization

- **1.5 NFV Requirements (cont'd)**
- **Elasticity**
  - The VNF vendor shall describe an info model for each component capable of parallel operation, the range (min- max) of instances supported
    - and additional info (needed processing power, packet throughput, storage, memory, cache requirements) for each component
  - NFV framework should be able to
    - Support SLA-base *scaling* of the VNFs
    - The scaling request or automatic decision may be granted/denied, depending on NO view/policy, resource constraints or external inputs
  - The VNF user shall be capable (through std. info model) to request, for each component (capable of scaling), specific min-max limits within the ranges specified by the VNF vendor, as to fulfill individual SLA
  - NFV framework should be able to
    - move some or all VNF components from one compute resource onto a different resource, while meeting the service continuity requirements for VNF components

1. Network Function Virtualization

- **1.5 NFV Requirements**
- **Resiliency**
  - The NFV framework shall be able to
    - Provide mechanisms to recreate (on demand, or automatic) NFs after a failure
    - Provide means to classify sets of VNFs having similar reliability/ availability reqs., in resilience categories
    - Support standard- based replication of state data (sync./async.) and preservation of data integrity as to fulfill SLAs
  - The NFV framework (including orchestration and other necessary functions for service continuity) shall facilitate resiliency schemes both in CPI and DPI
  - The SLA shall specify the metrics to define the values and variability of "stability"
  - Network stability needs $\Rightarrow$ the NFV framework shall support mechanisms to measure various metrics and ensure that they are met per each SLA
    - *packet loss rate, drop, latency, delay variation, time to detect and recover faults, etc.*

1. Network Function Virtualization

- **1.5 NFV Requirements**
- **Security**

  - The NFV framework shall implement sec. counter-measures to address:
    - Vulnerabilities of ther virtualization layer, protection of data on shared storage resources, protection of I/Fs, isolation of distinct VNFs, separation between VNF sets,
    - secure mgmt. of VNF sets by other third party entities
  - The NFV framework shall provide mechanisms
    - For the NO to control/verify the elements config. that virtualize the hW resources
  - Mgmt & Orchestration (M&Or) and NFVI - will use std. mechanisms for AAA , encryption and validation

  - Access to NFV functions via NFV exposed API at all layers shall be protected by std. mechanisms (AAA, encryption, confidentiality, data integrity)
  - M&Or shall provide ≥ 2 levels of privileges to API clients

  - The NFV exposed API should be divided into multiple subsets of APIs, so the clients with different privilege levels will only able to use APIs allowed for their level
  - The M&Or shall be able to authorize client's privilege to use certain APIs

1. Network Function Virtualization

- **1.5 NFV Requirements**
- **Service Continuity**
  - The NFV framework shall provide capabilities such as:
    - The SLA shall describe the level of service continuity required **(seamlesss, non-seamless)** and required attributes
    - In case of HW/SW failures, the continuity shall be restored conforming the SLA specifications
  - **The VNF framework will evaluate the impact of VNF instance migration upon the service continuity in the SLA limits**
    - the communication between the migrated VNF instance and other entities shall be maintained regardless of locations and awareness of migration

- **Service Assurance**
  - The NFV framework shall provide mechanisms for time-stamping of HW (NICs,devices, NIDs, etc.) that sit beneath the NFVI
    - Copy packets/frames, timp-stamp the copies, forward the time stamped copies to a configured destination
  - Interrogation should be possible if a particular NIC provides HW time-stamping facilities
  - The mgmt. system shall be able to detect VNF instances failures and/or network reachability to that set of VNF instances and take remediation actions

1. Network Function Virtualization

- **1.5 NFV Requirements**
- **Operational and Management Requirements**
  - The NFV framework shall incorporate mechanisms for automation of operational and mgmt. functions (e.g. creation, scaling, healing of VNF instances)
  - It shall provide M&O functionality, for VNF instances lifecycle : *instantiation, resource allocation, scaling, termination*
  - The M&O functionality shall
    - be neutral w.r.t the logical function provided by VNFs and will interact with other operation systems
    - manage the VNFs and VNFs instances lifecycle; maintain the integrity of VNFs
    - manage the NFVI in coordination with other applicable mgmt. system
    - monitor and collect NFVI resource usage
    - support std. APIs
    - manage policies and constraints (e.g replacing VMs); enforce policies and constraints when allocating and resolving conflicts regarding NFVI resources
    - manage the assignment of NFVI resources to VNFs- assuring the sharing of resources

1. Network Function Virtualization

- **1.5 NFV Requirements**
- **Energy Efficiency Requirements**

  - Studies indicate that NFV could potentially save 50% of energy consumption w.r.t current network infrastructures

  - The NFV framework shall be able to
    - place the VNFs on particular resources considering the possible needs of moving them aiming to create power saving capable states

    - provide mechanisms enabling authorized entities to control and optimize energy consumption on demand

    - provide an information model including attributes defining the time-frame required for a compute resource, hypervisor, and/or VNF, to return to a normal operation mode after leaving a power-saving mode

1. Network Function Virtualization

- **1.5 NFV Requirements**

- **Coexistence with existing networks – transition**
  - The NFV framework shall coexist with legacy network equipment, i.e.:
    - to work in a hybrid network (classical + NFV)
    - Support transition from Phy NFs to VNfs defined by a VNF graph

  - The NFV framework
    - in conjunction with legacy mgmt. system shall support the same service capabilities and acceptable performance impact within service SLA, during transition period to NFV-type working-style of the network

    - Shall be able to interwork with legacy mgmt. systems with minimal impact on existing network nodes and I/Fs

    - Shall assure, during the transtion, security of VNFs, without impacting the existent Phy. NFs

1. Network Function Virtualization

- **1.5 NFV Requirements**
- **Service Models**
  - **In NFV approach it will be possible for an SP to run VNF instances on a NFVI operated by other SP** (including infrastructure elements common with cloud Computing services)
    - This will enlarge the scope of a single SP
  - Maintenance for NFV framework and hosted VNFs will be different w.r.t. non- virtualized network functions
  - Multi-vendor environment will exist needing special features for operation and maintenance actions

- **Deployment models**
  - The NFV framework shall
    - provide the same level of service availability for fully or partially virtualized scenarios
    - permit identification of different types of flows (Data Plane, Control Plane)
  - User Services shall be agnostic w.r.t. virtual/non-virtual implementation

1. Network Function Virtualization

- **1.5 NFV Requirements**
- **Service Models (cont'd)**

  - **Service models details**
  - The NFV framework shall
    - facilitate NOs to consume NFVI resources operated by a different infrastructure provider, via std. APIs, without degraded capabilities- compared to self-operated infrastructure resources
    - **allow VNFs from different SPs to coexist on the same infrastructure, while facilitating appropriate isolation between the resources allocated to different SPs**
    - **allow each SP to assure and bill its services for its users**

    - allow instantiation of different VNFs across infrastructures that are independently administered
    - provide mechanisms for AAA, measurements, notifications and diagnostics for SLAs
    - Maintain the relationships between
      - VNFs/applications
      - The service making use of them
      - The resources allocated
- **Maintenance models**
  - ETSI defined also serviceability and maintainability requirements describing how to run an NFV framework and the hosted VNFs
  - They mostly deals with exchanging SW or HW and with diagnosis and elimination of problems

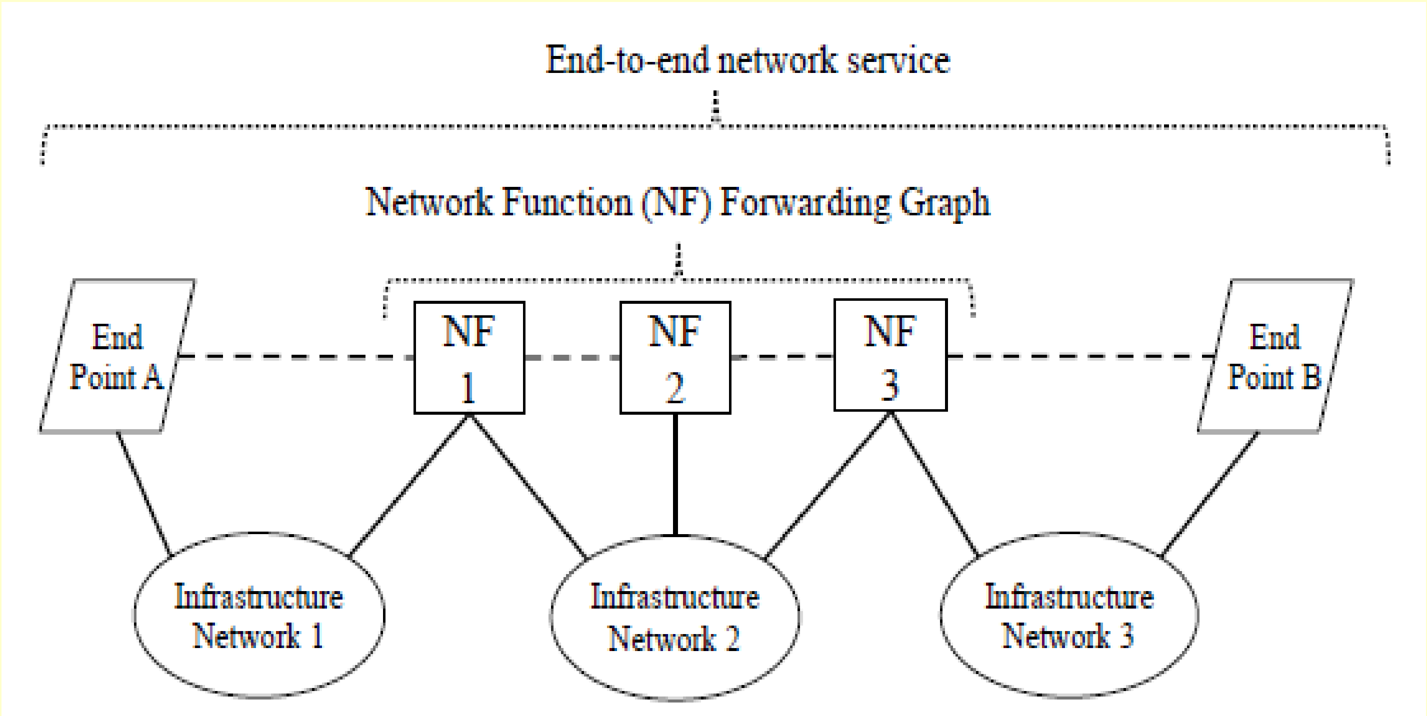
1. Network Function Virtualization

- **1.6 Network Services (NS) in NFV**
  - **A network service**
    - can be viewed as a ***Forwarding Graph of Network Functions (NFs)*** interconnected by supporting network infrastructure.
    - These NFs can be implemented in a single operator network or interwork between different operator networks.
  - The underlying NF behaviour contributes to the behaviour of the higher-level service.
  - The NS behaviour is determined by its constituent FBs, (it can include individual NFs, NF Sets, NF Forwarding Graphs (NF-FG) , and/or the infrastructure network
  - An NF FG can have network function nodes connected by logical links : unidirectional, bidirectional, multicast and/or broadcast
  - **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 (see some ETSI Use cases)

1. Network Function Virtualization

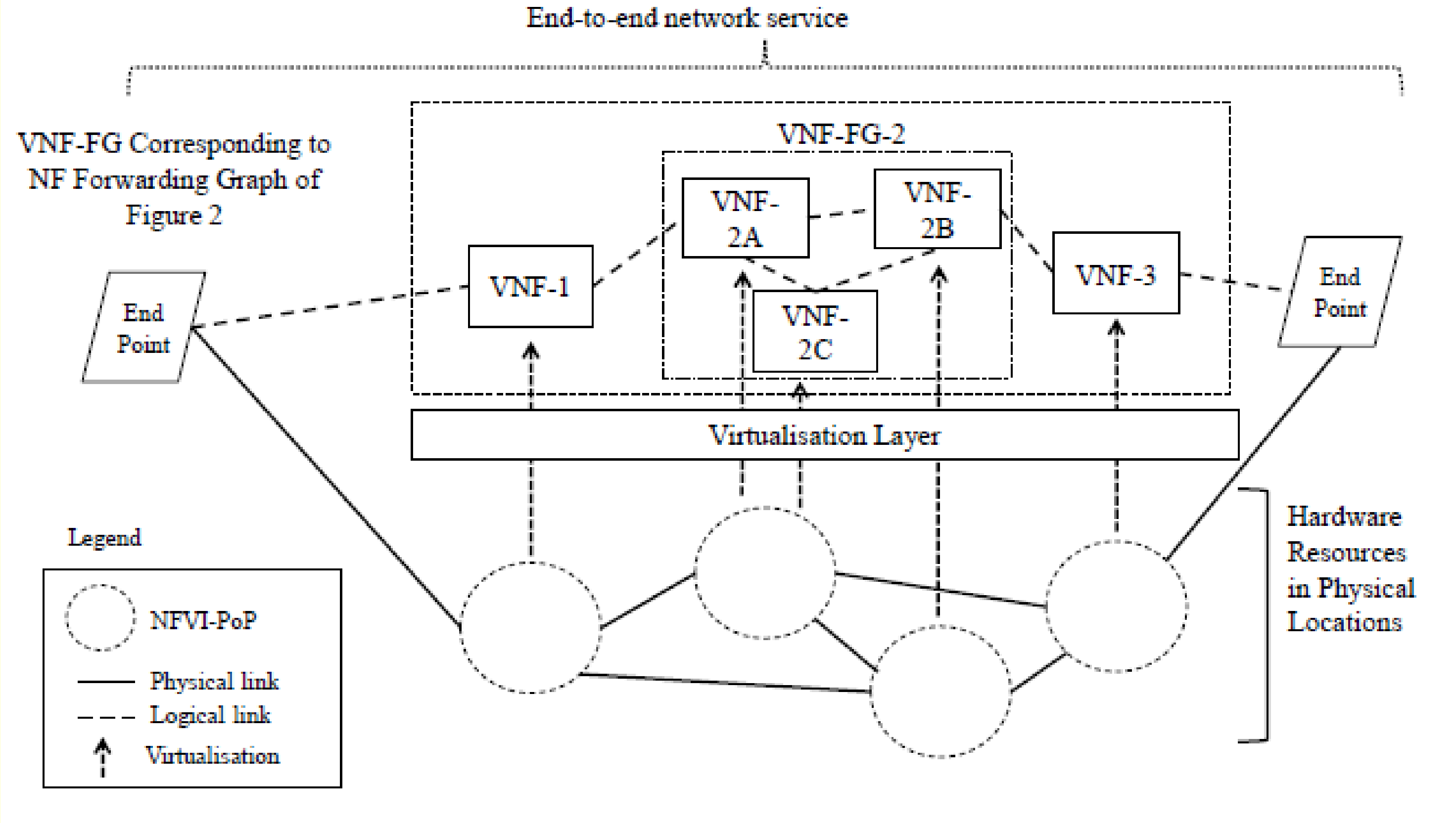
- **1.6 Network Services in NFV**

- Graph representation of an E2E network service (ETSI)
  - Dotted lines = logical I/Fs provided by the Network Infrastructures (wired/wireless)
  - E2E network service = { A, NF-FG, B}



1. Network Function Virtualization

- **1.6 Network Services in NFV**
- **Example**
  - E2E network service and the different layers involved in its virtualization process. *Source: ETSI*
  - Virtualization layer makes decoupling of HW/SW in NFs and abstraction of HW resources



1. Network Function Virtualization

- **1.6 Network Services in NFV (cont'd)**
- **Example (cont'd)**
  - The NFVI-PoPs includes resources for ***computation, storage and networking*** deployed by a NO
  - VNFs run on top of the **virtualization layer, which is part of the NFVI**
  - The VNF-FGs can be nested (see VNF-FG-2)
  - Objective : the I/Fs between NFs and/or VNFs and the infrastructure in a multi-vendor environment should be based upon accepted stds.

  - **The exact physical deployment of a VNF instance on the infrastructure is not visible from the E2E service perspective**
    - exception of guaranteeing specific policy constraints (e.g. location awareness required to implement a virtualized CDN cache node)
    - Therefore: the VNF instance can to be implemented on different physical resources,
      - e.g. compute resources and hypervisors
      - and/or be geographically dispersed
    - However, the VNF instances and their supporting infrastructure need to be visible for configuration, diagnostic and troubleshooting purposes

1. Network Function Virtualization

- **1.7 NFV Reference Architectural Framework**

- Functional Blocks
  - **Operations and Business Support Systems (OSS/BSS)**

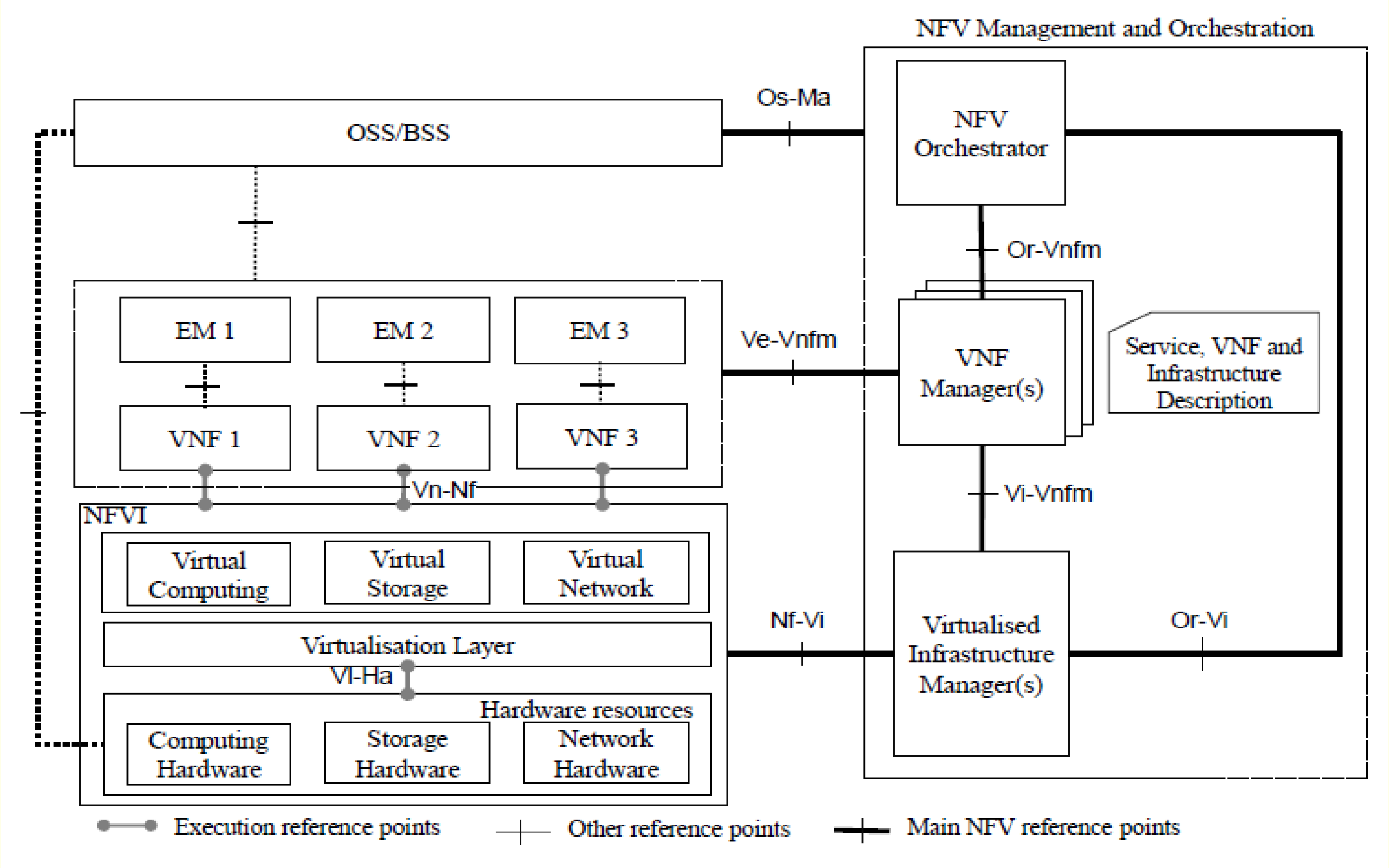
  - **Element Management (EM)**
  - **Virtualised Network Function (VNF)**

  - **NFV Infrastructure (NFVI)**
    - Hardware and virtualized resources
    - Virtualization Layer.

  - **NFV Management and Orchestration**
    - NFV Orchestrator.
    - VNF Manager(s).
    - Virtualised Infrastructure Manager(s).
    - Service, VNF and Infrastructure Description

1. Network Function Virtualization

- 1.7 **NFV reference architectural framework Source: ETSI**
  - The dotted RPs are available in deployment; might need extensions for handling NFV
    - They are not the main focus of NFV at present
  - The Arch. framework does not specify NFs; they are decided of the owner of the network.



1. Network Function Virtualization

- **1.7 NFV Reference Architectural Framework**

- **Overview of the Functional Blocks (cont'd)**

- **Virtualized Network Function (VNF)**
  - A VNF is a virtualization of a NF in a legacy non-virtualized network.
  - **NFs examples:**
    - 3GPP Evolved Packet Core (EPC) network elements,
      - e.g. Mobility Management Entity (MME)
      - Serving Gateway (SGW), Packet Data Network Gateway (PGW)
    - Elements in a home network
      - e.g. Residential Gateway (RGW)
    - Conventional network functions
      - e.g. Dynamic Host Configuration Protocol (DHCP) servers, firewalls, etc.
  - NF functional behaviour and state are largely independent of whether the NF is virtualised or not.
  - The functional behaviour and the external operational interfaces of a Phy NF (PNF) and a VNF are expected to be the same.
  - A VNF may have n ≥ 1internal components ; e.g., one VNF can be deployed over multiple VMs, (each VM hosts a single VNF component)

  - **Element Management (EM)**
  - EM performs the typical mgmt. functionality for one or several VNFs

1. Network Function Virtualization

- **1.7 NFV Reference Architectural Framework**
- **Overview of the Functional Blocks (cont'd)**

- NFV differentiates two types of networks:
  - *NFVI-PoP network*: interconnects the computing and storage resources contained in an NFVI-PoP. It also includes specific switching and routing devices to allow external connectivity.
  - *Transport network:* interconnects NFVI-PoPs, NFVI-PoPs to other networks owned by the same or different NO, and NFVI-PoPs to other network appliances or terminals not contained within the NFVI-PoPs.

1. Network Function Virtualization

- **1.7 NFV Reference Architectural Framework**
- **Overview of the Functional Blocks (cont'd)**

- **NFV Infrastructure (NFVI)**
  - **Definition:** totality of all HW/SW components which build up the environment in which VNFs are deployed, managed and executed.
  - NFVI can span across several locations, i.e. places where NFVI-PoPs are operated.
  - The network providing connectivity between these locations is regarded to be part of the NFVI
  - From the VNF's perspective, the virtualization layer (VL) and the HW resources look like a single entity providing the VNF with desired virtualized resources

- **HW Resources**
  - ***Computing, storage and network*** providing processing, storage and connectivity to VNFs through the virtualization layer (e.g. hypervisor).
  - Computing HW is assumed to be COTS as opposed to purpose-built HW.
  - Storage resources
    - shared network attached storage (NAS)
    - storage that resides on the server itself.
  - Computing and storage resources are commonly pooled.
  - Network resources: switching functions, e.g. routers, and wired or wireless links; they can span different domains

1. Network Function Virtualization

- **1.7 NFV Reference Architectural Framework**
- **Overview of the Functional Blocks (cont'd)**

- ***Virtualization Layer (VL) and Virtualised Resources***

  - The VL abstracts the HW resources and decouples the VNF SW from the underlying HW, thus ensuring a HW independent lifecycle for the VNFs.

  - **The VL is responsible for:**
    - Abstracting and logically partitioning Phy resources, commonly as a HW abstraction layer.
    - Enabling the SW that implements the VNF to use the underlying virtualised infrastructure.
    - Providing virtualised resources to the VNF, so that the latter can be executed.

  - The VL allows that SW of VNFs can be deployed on different physical HW resources.
    - Typically, this type of functionality is provided for computing and storage resources in the form of hypervisors and VMs.

1. Network Function Virtualization

- **1.7 NFV Reference Architectural Framework**
- **Overview of the Functional Blocks (cont'd)**

- ***Virtualization Layer (VL) and Virtualised Resources (cont'd)***
  - A VNF can be deployed in one or several VMs

  - The NFV AF does not impose any specific virtualization layer solution.
    - Rather, NFV expects to use VLs with std. features and open execution RPs towards VNFs and HW (computation, network and storage).

- In some cases VMs might have direct access to HW resources (e.g.NICs) for better performance.

- However in NFV, VMs shall
  - always provide standard ways of abstracting HW resources
  - without restricting its instantiation or dependence on specific HW components.

1. Network Function Virtualization

- **1.7 NFV Reference Architectural Framework**
- **Overview of the Functional Blocks (cont'd)**
- ***Virtualization Layer (VL) and Virtualised Resources (cont'd)***
- **Hypervisors** : common solution for VNFs deployment

- **Other solutions** to realize VNFs
  - SW running on top of a non-virtualised server by means of an operating system (OS)
  - VNFs implemented as an application that can run on virtualised infrastructure or on bare metal

- To ensure **operational transparency:** VNF operation should be independent of its deployment scenario
  - The VL realizes virtualized network paths connecting VMs of a VNF and/or between different VNF instances
  - Possible techniques
    - network abstraction layers that isolate resources via virtual networks and network overlays
    - Virtual Local Area Network (VLAN)
    - Virtual Private LAN Service (VPLS)
    - Virtual Extensible Local Area Network (VxLAN)
    - Network Virtualization using Generic Routing Encapsulation (NVGRE), etc.

- Other possible forms of virtualization of the transport network include centralizing the control plane of the transport network and separating it from the forwarding plane, and isolating the transport medium (i.e. SDN)

1. Network Function Virtualization

- 1.7 NFV Reference Architectural Framework
- **Overview of the Functional Blocks (cont'd)**
- ***Virtualised Infrastructure Manager(s)(VIM)***
  - VIM control and manage the interaction of a VNF with computing, storage and network resources under its authority, as well as their virtualization.
  - One may deploy multiple VIM instances
  - **VIM performs:**
    - **Resource management**:
      - Inventory of SW (e.g. hypervisors), computing, storage and network resources dedicated to NFVI
      - Allocation of virtualization enablers, e.g. VMs onto hypervisors, compute resources, storage, and relevant network connectivity.
      - Management of infrastructure resource and allocation, e.g. increase resources to VMs, improve energy efficiency, and resource reclamation
  - **Operations, for:**
    - Visibility into and management of the NFV infrastructure.
    - Root cause analysis of performance issues from the NFVI perspective.
    - Collection of infrastructure fault information.
    - Collection of information for capacity planning, monitoring, and optimization

1. Network Function Virtualization

- **1.7 NFV Reference Architectural Framework**
- **Overview of the Functional Blocks (cont'd)**
- ***NFV Orchestrator***
  - It orchestrates and manages the NFVI and SW resources and realizes network services on NFVI.

- ***VNF Manager(s)***
  - It is responsible for VNF lifecycle management (e.g. instantiation, update, query, scaling, termination).
  - Multiple VNF Managers may be deployed; a VNF Manager may be deployed for each VNF, or a VNF Manager may serve multiple VNFs.

- ***Service, VNF and Infrastructure Description***
  - Data-set providing info regarding the VNF deployment template, VNF-FG, service-related information, and NFVI information models.
    - These templates/descriptors are used internally within NFV M&O.

  - The NFV M&O FBs handle information contained in the templates/ descriptors and may expose (subsets of) such information to applicable FBs, as needed.

- **Operations Support Systems and Business Support Systems** belongs to the Operator (OSS/BSS)

1. Network Function Virtualization

- **1.7 NFV Reference Architectural Framework**

- **Reference Points**
- ***Virtualization Layer - HW Resources - (Vl-Ha)***
  - I/F: the VL to HW resources, to create an execution environment for VNFs, and collect relevant HW resource state information for managing the VNFs without being dependent on any HW platform.

- ***VNF - NFV Infrastructure (Vn-Nf)***
  - It 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 HW-independent lifecycle, performance and portability requirements of the VNF

- ***NFV Orchestrator - VNF Manager (Or-Vnfm)***

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

1. Network Function Virtualization

- **1.7 NFV Reference Architectural Framework**

- **Overview of the Functional Blocks (cont'd)**
- **Reference Points (cont'd)**

- ***Virtualised Infrastructure Manager - VNF Manager (Vi-Vnfm)***
  - Resource allocation requests by the VNF Manager.
  - Virtualised HW resource configuration and state information (e.g. events) exchange.

- ***NFV Orchestrator - Virtualised Infrastructure Manager (Or-Vi)***
  - Resource reservation and/or allocation requests by the NFV Orchestrator.
  - Virtualised HW resource configuration and state information (e.g. events) exchange.

- ***NFVI - Virtualised Infrastructure Manager (Nf-Vi)***
  - Specific assignment of virtualized resources in response to resource allocation requests.
  - Forwarding of virtualized resources state information.
  - HW resource configuration and state information (e.g. events) exchange.

1. Network Function Virtualization

- **1.7 NFV Reference Architectural Framework**

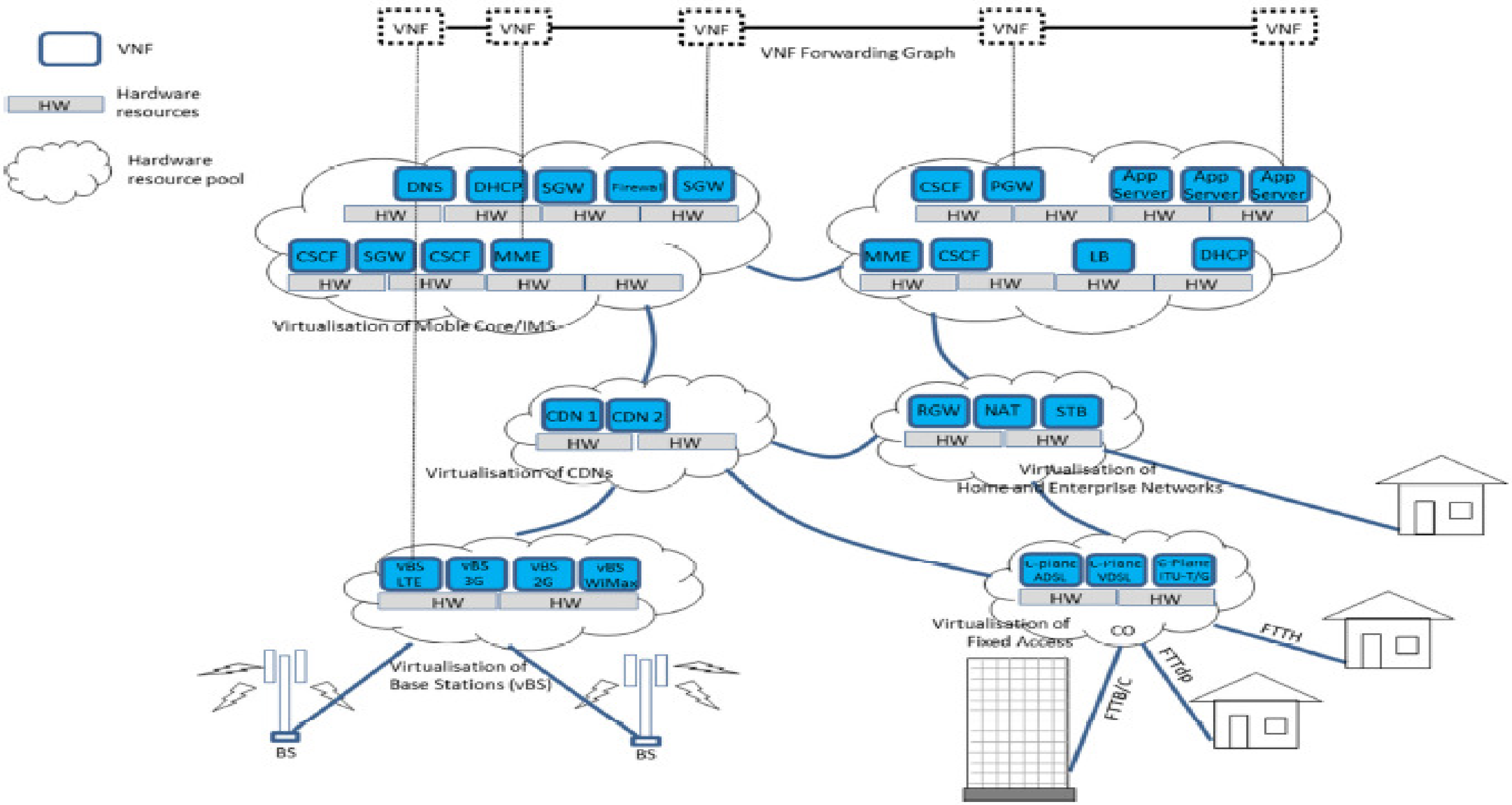
- **Overview of the Functional Blocks (cont'd)**

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

- ***VNF/EM - VNF Manager (Ve-Vnfm)***
  - Requests for VNF lifecycle management
  - Exchanging configuration information
  - Exchanging state information necessary for network service lifecycle management

1. Network Function Virtualization

- **1.8 NFV Use Cases (ETSI)**
- Overview of NFV Use Cases. Source: https://portal.etsi.org/nfv/nfv_white_paper2.pdf



1. Network Function Virtualization

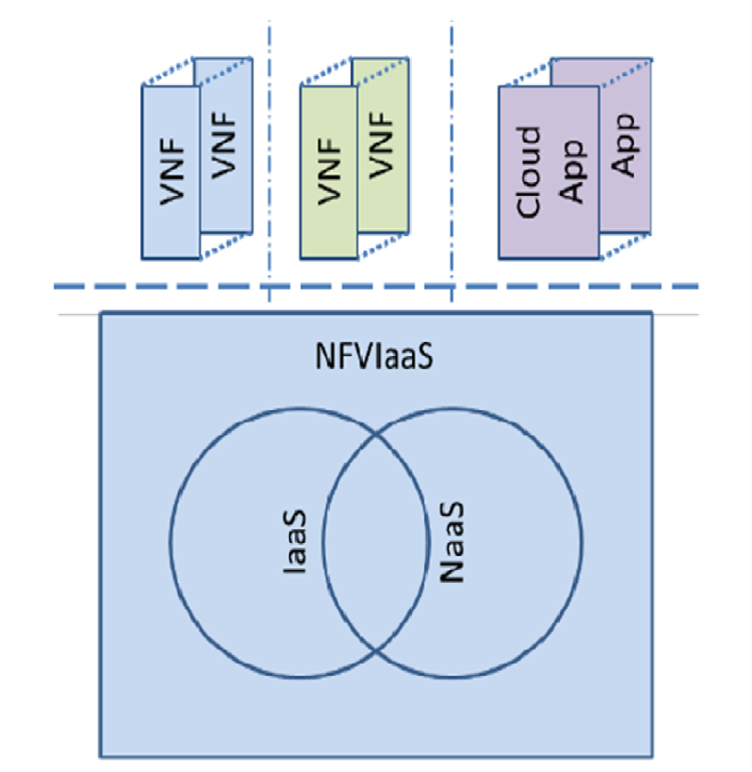
- 1.8 NFV Use Cases (ETSI)
- **SPs can offer cloud services in addition to their network services**

- **Cloud computing is based on:**
  - **Compute, storage, networking**
- **VNF also needs**
  - **Compute, storage, networking**

- **Resource pooling: both in CC and NFV**

- **Such resources enable a SP to offer both cloud services + network services**

- **Examples of Use Cases**
- **NFV Infrastructure as a Service (NFVIaaS)**
  - NFVI provides
    - an environment in which (VNFs) can execute
    - compute capabilities comparable to an **IaaS cloud service**
      - as a run time execution environment
      - as well as support the dynamic network connectivity services ~ comparable to NaaS (e.g. on- demand connection between CSPs and CSCs, or between two computing nodes)



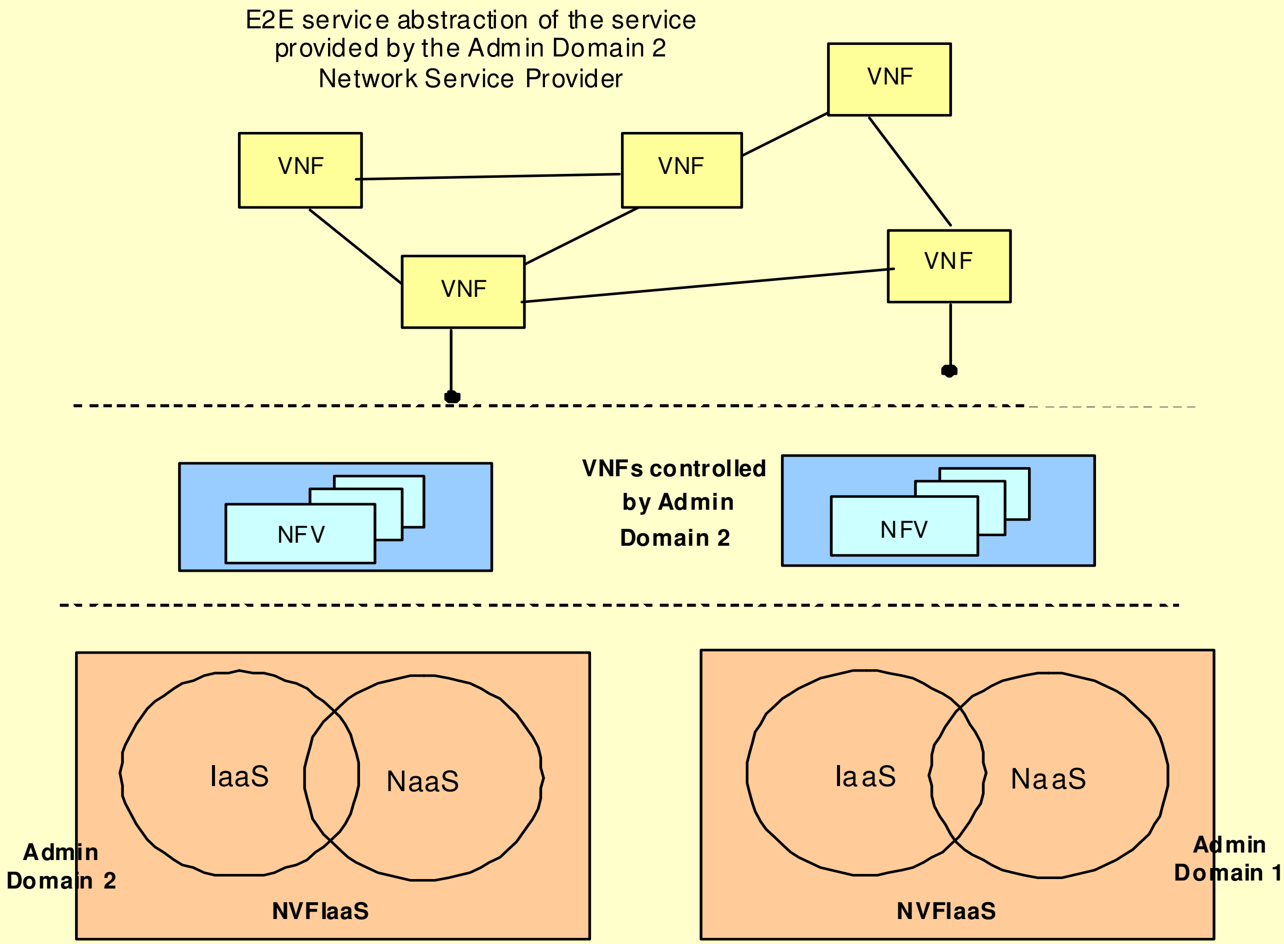
Note : NaaS is not yet standardized

1. Network Function Virtualization

- **1.8 NFV Use Cases (ETSI)**
- **NFV Infrastructure as a Service (NFVIaaS) (cont'd)**
  - Private Cloud (see NIST-reference architectural model) can be a convenient model for NFVIaaS
  - It is desirable for an SP to be able to run VNF instances inside a NFVI owned by other SP
    - The SP offers its NFVI as a service to other SPs
    - In such a way the users may have access to globally developed services
  - Resources to be pooled
    - In NFV model: Compute, Hypervisor and Network domains of the NFVI
    - In CC model: elements supporting IaaS or NaaS services
  - NFVIaaS will offer multi-tenant support; on top of it one may have
    - Different VNFs
    - Cloud applications

1. Network Function Virtualization

- **1.8 NFV Use Cases (ETSI)**
- **NFV Infrastructure as a Service (NFVIaaS) (cont'd)**
- **Example: Admin Domain 2 running VNFs on the NFVI provided by Admin Domain 1**



1. Network Function Virtualization

- **1.8 NFV Use Cases (ETSI)**
- **NFV Infrastructure as a Service (NFVIaaS) (cont'd)**
- **Virtualization Target**
  - The NFVI should be available as an exec. environment for SW entities
  - The NFVIaas should support
    - dynamic creation of connectivity (e.g. NaaS) between Phy and virtual network termination points (e.g. VNF instances, Phy network termination)
    - generic computing loads ("cloud apps") on IaaS basis
    - delivery of NFVIaaS services within one admin. domain or across admin boundaries
  - SP1 has to choose the terms of the commercial offer proposed for other SPs; however these are not subject of stds.
  - A target of stds. is the metadata description of the types of NFVI resources that can be made available through NFVIaaS

1. Network Function Virtualization

- **1.8 NFV Use Cases (ETSI)**

- **NFV Infrastructure as a Service (NFVIaaS) (cont'd)**

- **Virtualization Target (cont'd)**

  - SP2 could be interested to run a VNF instance on NFVI@SP1
    - Resiliency reasons:
      - NFVI@SP1 and NFVI@SP2 are distinct and independent
      - Running VNF instances on other SPs NFVIs can improve resiliency
      - Virtualization typically converts infrastructure failures into capacity reduction
    - Latency reasons
      - E.g. latency can be reduced by placing selected NFs closer to consumer (CDN use case)
    - Regulatory requirements reasons

- **Coexistence of virtualized and non-virtualized NFs**
  - Non-virtualized NFs should coexist in parallel with VNFs
  - VNFs from multiple SPs may coexist within the same NFVI
  - NFVI should provide isolation between them (including failure cases)

CONTENTS

2. Software Defined Networking

- **2.1 Introduction**

- Recent industry/research effort resulted in new approaches:
  - **Software- Defined Networking (SDN)** –new networking architecture
  - **Open Networking Foundation** (ONF- non-profit industry consortium) → several OpenFlow I/F specifications for SDN

- **Promises for enterprises and carriers** :
  - higher programmability opportunities, automation, and network control
  - enabling them to build highly scalable, flexible networks
  - fast adapt to changing business needs

- *Source: Software-Defined Networking: The New Norm for Networks ONF White Paper April 13, 2012*

- ***Note:***
  - ***traditional TCP/IP networking control : fully distributed***
  - ***SDN : more centralized (at least logical)***

2. Software Defined Networking

- **2.1 Introduction (cont'd)**

- **SDN + OpenFlow I/F(first standard) advantages:**
  - *high-performance, granular traffic control* across multiple vendors' network devices
  - *centralized management and control* of networking devices improving automation and management
  - *common APIs abstracting the underlying networking details* from the orchestration and provisioning systems and applications;
  - *flexibility:* new network capabilities and services with no need to configure individual devices or wait for vendor releases
  - *programmability* by operators, enterprises, independent software vendors, and users (not just equipment manufacturers) using common programming environments
  - *Increased network reliability* and *security* as a result of centralized and automated management of network devices, uniform policy enforcement, and fewer configuration errors

2. Software Defined Networking

- **2.1 Introduction (cont'd)**
- **SDN + OpenFlow advantages:**
  - *more granular network control* with the ability to apply comprehensive and wide-ranging policies at the session, user, device, and application levels
  - *better end-user experience* as applications exploit centralized network state information to seamlessly adapt network behavior to user needs
  - *protects existing investments* while future-proofing the network
  - **With SDN, today's static network can evolve into an extensible service delivery platform capable of responding rapidly to changing business, end-user, and market needs.**

2. Software Defined Networking

- **2.2 Early SDN activities**
- Early SDN **products and activities examples**
  - 2008: Software-Defined Networking (SDN) : NOX Network Operating System [Nicira]; OpenFlow switch interface [Stanford/Nicira]

- **Open Networking Foundation (2011)**
  - ***https://www.opennetworking.org/***
- **2013 status and activities**
  - **Membership surpassed 100 companies**
  - Published *OpenFlow* Switch Specification 1.3.2
  - Published *OpenFlow Configuration and Management Protocol* 1.1.1 (OF-Config 1.1.1)
  - Approved OpenFlow Switch Specification 1.4
  - Near completion of OpenFlow Switch Specification 1.5
  - Launched the OpenFlow Software Driver Competition
  - Created the ONF Chipmakers Advisory Board (CAB)
  - Created the Research Associates Program
  - Launched the OpenFlow Conformance Testing Program

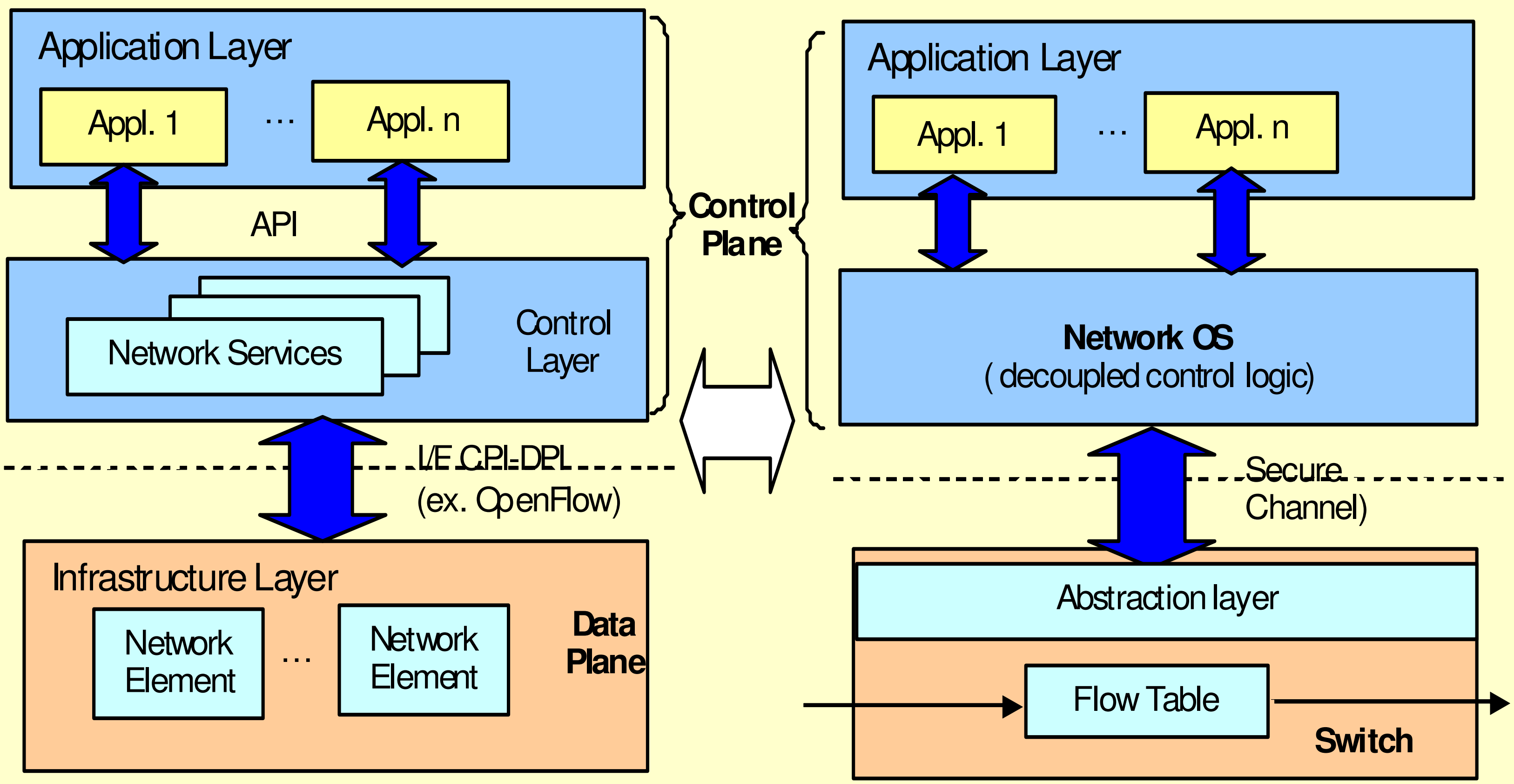
2. Software Defined Networking

2.3 SDN Basic Architecture

- **Evolutionary**
- **CPI and DPI are separated**
- Network intelligence is (logically) centralized in SW-based SDN controllers, which maintain a global view of the network.
- Execute CPI SW on general purpose HW
  - Decoupled from specific networking HW
  - CPI can use use commodity servers
- Data Plane (DPI) is programmable
- Maintain, control and program data plane state from a central entity
- The architecture defines the control for a network (and not for a network device) The network appears to the applications and policy engines as a single, logical switch
- This simplified network abstraction can be efficiently programmed

2. Software Defined Networking

2.3 SDN Basic Architecture (cont'd)



SDN Generic Architecture

2. Software Defined Networking

- **2.3 SDN Basic Architecture (cont'd)**
- **Control Plane**
  - **Control Applications/Program**
    - operates on view of network **:**
    - performs different functions (routing, traffic engineering, QoS, security, etc.)
    - **Input**: global network view (graph/database)
    - **Output**: configuration of each network device
    - Control program –seen as whole could be not a distributed system
      Abstraction hides details of distributed state
  - **Network OS:** distributed system creating a consistent, global and up-to-date network view
    - In SDN it runs can on controllers (servers) in the network
    - It creates the "lower layer" of the Control Plane
    - Examples: NOX, ONIX, Trema, Beacon, Maestro, …

- **Data Plane** : forwarders/switches (**Forwarding elements -FE**)
  - NOS uses some abstraction to:
    - Get state information from FE
    - Give control directives to FE

2. Software Defined Networking

- **2.3 SDN Basic Architecture (cont'd)**
- **Advantages**
- **Centralization allows:**
  - To **alter network behavior in real-time** and faster deploy new applications and network services (hours, days not weeks or months as today).

  - **flexibility** to configure, manage, secure, and optimize network resources via dynamic, automated SDN programs (not waiting for vendors) .

- **APIs facilitate implementation of:**
  - **common network services**: routing, multicast, security, access control, bandwidth management, QoS, traffic engineering, processor and storage optimization, energy usage
  - **policy management**, custom tailored to meet business objectives
    - Easy to define and enforce consistent policies across both wired and wireless connections on a campus

- Manage the entire network : intelligent orchestration and provisioning systems

2. Software Defined Networking

- **2.3 SDN Basic Architecture (cont'd)**
- **Advantages (cont'd)**

  - ONF studies open APIs to promote multi-vendor management:
    - possibility for **on-demand resource allocation, self-service provisioning,** truly virtualized networking, and secure cloud services.

  - SDN control and applications layers, business apps can operate on an **abstraction of the network**, leveraging network services and capabilities without being tied to the details of their implementation.

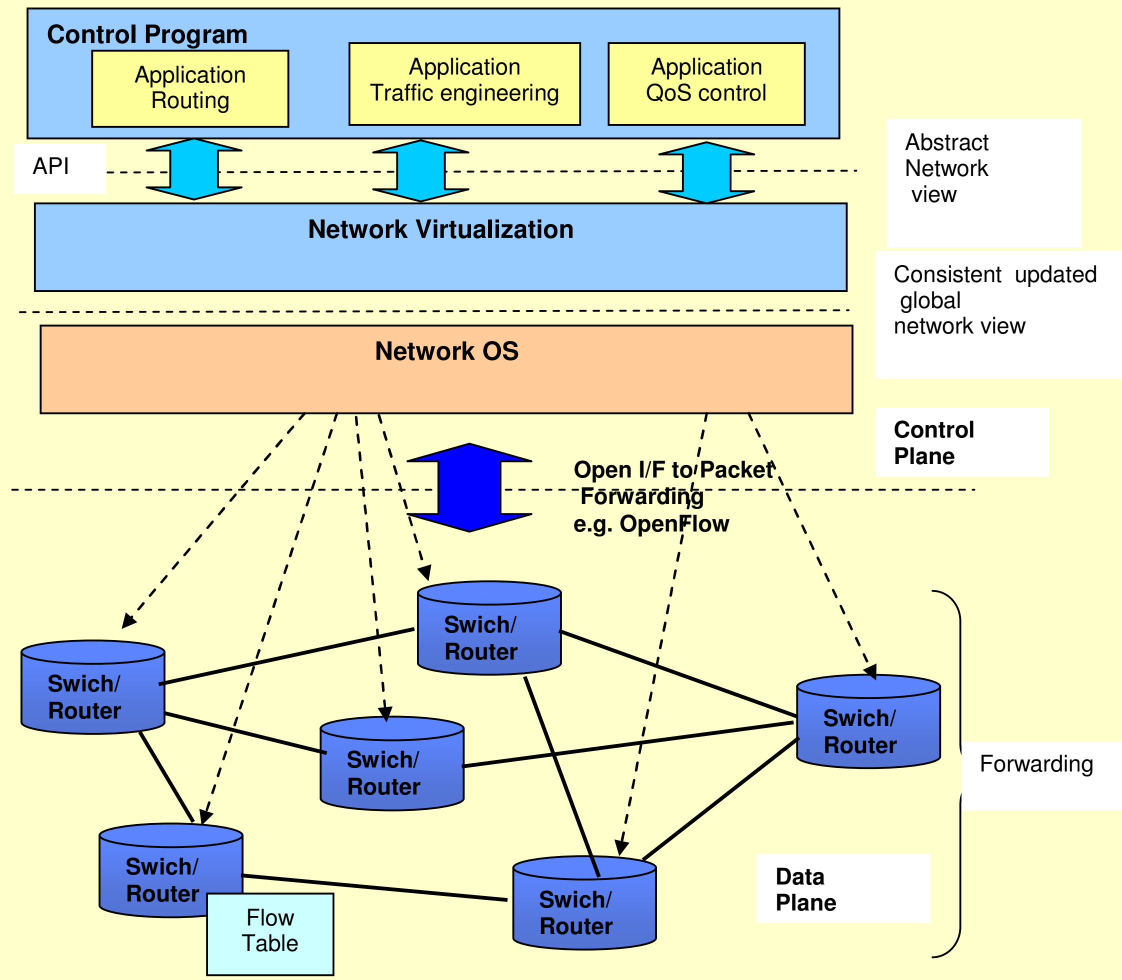
- **Open SDN issues/problems**
  - Balance between distribution – centralization (physical/logical)
  - **Scalability**
    - Controller scalability (w.r.t. processing power)
    - Communication Control Plane- Data plane
    - Multiple controllers
      - How many controllers
      - CPI topology, controller location, inter-controller communication
      - Consistency, Synchronization
  - Reliability (single points of failures?)

2. Software Defined Networking

- **2.3 SDN Basic Architecture (cont'd)**

- **Network OS:**
  - **Distributed system that creates a consistent, updated network view**
  - **Executed on servers (controllers) in the network**
  - **Examples: NOX, ONIX, HyperFlow, Floodlight, Trema, Kandoo, Beacon, Maestro,..**

- **Uses forwarding abstraction in order to:**
  - **Collect state information from forwarding nodes**
  - **Generate commands to forwarding nodes**



2. Software Defined Networking

- **OpenFlow Summary**

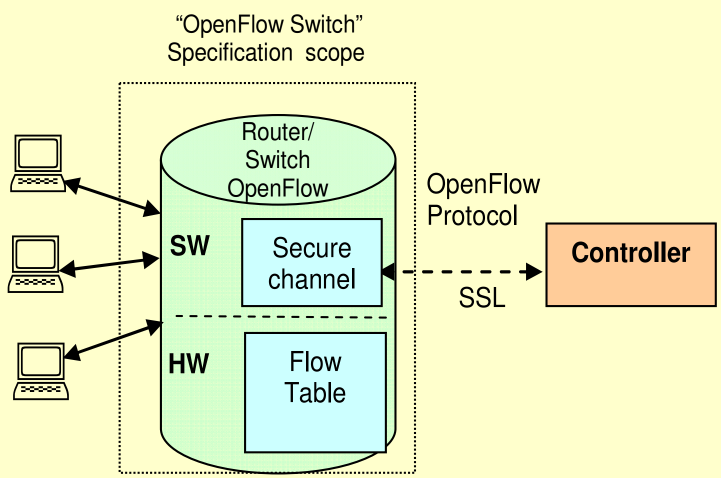
  - the **first SDN standard** communications: CPI-DPI I/F
  - allows direct access to the Fwd. Plane of network devices (switches and routers), both physical and virtual (hypervisor-based)
  - allows to move network control out of the networking switches to logically centralized control software
  - can be compared to the instruction set of a CPU
  - specifies basic primitives to be used by an external SW application to program the FwdPl (~ instruction set of a CPU would program a computer system)

2. Software Defined Networking

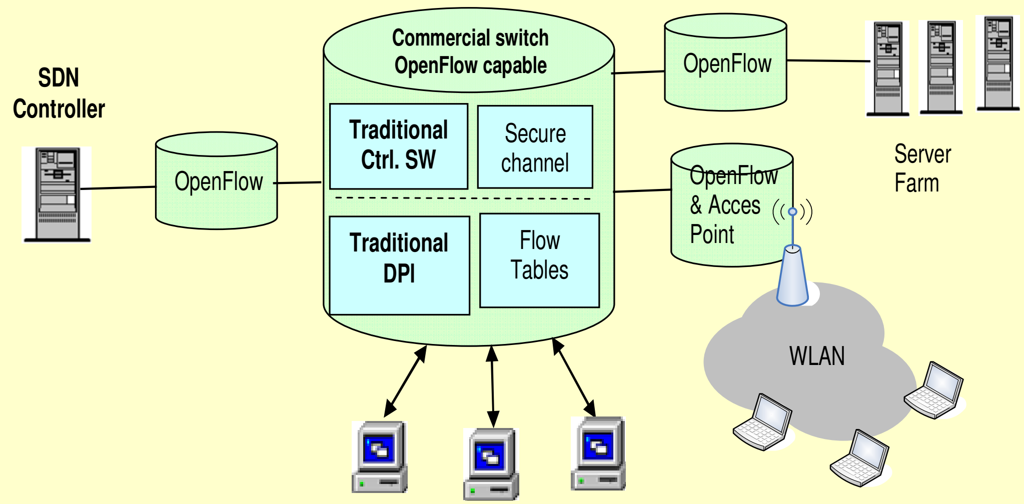
- **OpenFlow Summary**
  - uses the *concept of flows* to identify network traffic based on pre-defined match rules that can be statically or dynamically programmed by the SDN control SW
  - allows IT admin to define how traffic should flow through network devices based on parameters such as usage patterns, applications, and cloud resources
  - allows the network to be programmed on a per-flow basis (provides – if wanted- extremely granular control), enabling the network to respond to real-time changes at the application, user, and session levels

2. Software Defined Networking

- **Open Flow Summary**
- ***Initial proposal: Source : "OpenFlow: Enabling Innovation in Campus Networks"-*** *N.McKeown, T.Anderson, H.Balakrishnan, G.Parulkar, L.Peterson, J.Rexford, S.Shenker, J.Turner*



Ref1: Figure 1: Idealized OpenFlow Switch. The Flow Table is controlled by a remote controller via the Secure Channel.



Ref1: Figure 2: Example of a network of OpenFlow-enabled commercial switches and routers.

In Port	VLAN ID	Ethernet			IP			TCP	
		SA	DA	Type	SA	DA	Proto	Src	Dst

Table 1: The header fields matched in a "Type 0" OpenFlow switch.

2. Software Defined Networking

- **Open Flow Development**

- **OpenFlow Protocol**
  - Version 1.1 2011
  - Version 1.2 2012
  - Current version of OpenFlow is 1.4. (2013)

  - Many OpenFlow capable switches have been developed
  - SDN controllers (open source or commercial) are developed

CONTENTS

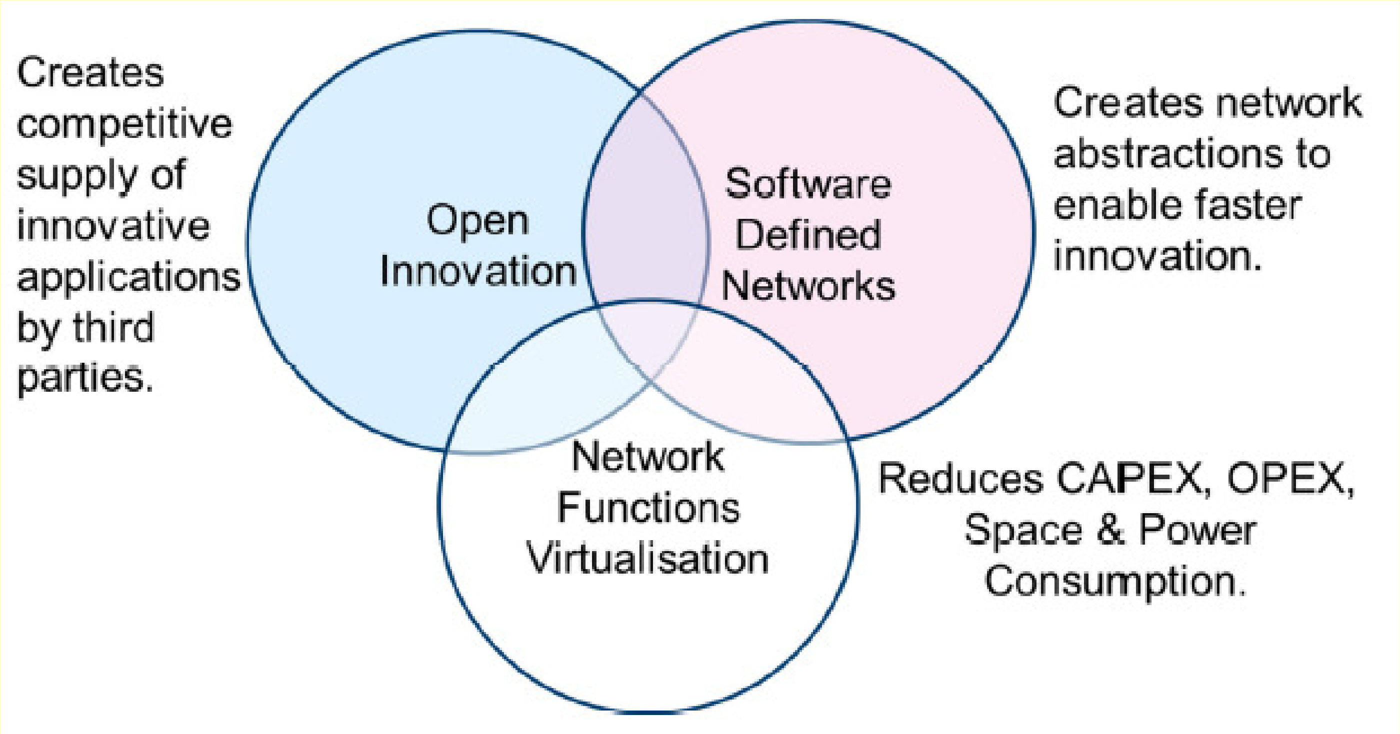
3. NFV-SDN Coperation

- **1.5 NFV&sDN**

  - NFV is complementary to SDN, but not dependent on it (or vice-versa)
  - NFV can be implemented without a SDN, although the two concepts and solutions can be combined with potentially greater value
  - NFV goals can be achieved using non-SDN mechanisms, relying on the techniques currently in use in many data centres
    - But SDN separation CPI/DPI - can enhance perf., simplify compatibility with existing deployments, and facilitate O&M
  - NFV is able to support SDN by providing the infrastructure upon which the SDN software can be run
  - NFV aligns closely with the SDN objectives to use commodity servers and switches.
  - There exists cooperation ETSI-ONF

3. NFV-SDN Coperation

- **1.5 NFV&SDN**
  - Source: ETSI



3. NFV-SDN Coperation

- **SDN versus Network Functions Virtualization (NFV)**
  - **NFV : ETSI Industry Specification Group initiative** - to virtualize network functions previously performed by proprietary dedicated hardware
  - goal : reduce the telecom network infrastructure cost
    - by allowing the appropriate functions to run on a common, commodity platform hosting the necessary virtualized environments.
  - **NFV in the market today includes**:
    - ***Virtual Switching*** – physical ports are connected to virtual ports on virtual servers with virtual routers using virtualized IPsec and SSL VPN gateways.
    - ***Virtualized Network Appliances*** –dedicated functional boxes can be replaced with a virtual appliance. (e.g. firewalls and gateways, Broadband Remote Access Servers (BRAS) , LTE Evolved Packet Core (EPC)).
    - ***Virtualized Network Services*** –e.g. network management applications such as traffic analysis, network monitoring tools, load balancers and accelerators.
    - ***Virtualized Applications*** – almost any application (e.g. . cloud applications: virtualized storage and photo imaging services, to support the explosion inmedia communications)

3. NFV-SDN Coperation

- **SDN versus Network Functions Virtualization (NFV)- cont'd**
- *Source: ONF: OpenFlow-enabled SDN and Network Functions Virtualization, Feb. 2014*

  - By enabling NFV with OpenFlow-enabled SDN, network operators can realize even greater benefits from this promising new use of cloud technology.

  - OpenFlow-based SDN can accelerate NFV deployment by offering a scalable, elastic, and on-demand architecture
    - well suited to the dynamic NFV communications requirements for both virtual and physical networking infrastructures.

3. NFV- SDN Cooperation

- ONF: NFV and SDN – industry view on architecture Source: ONF

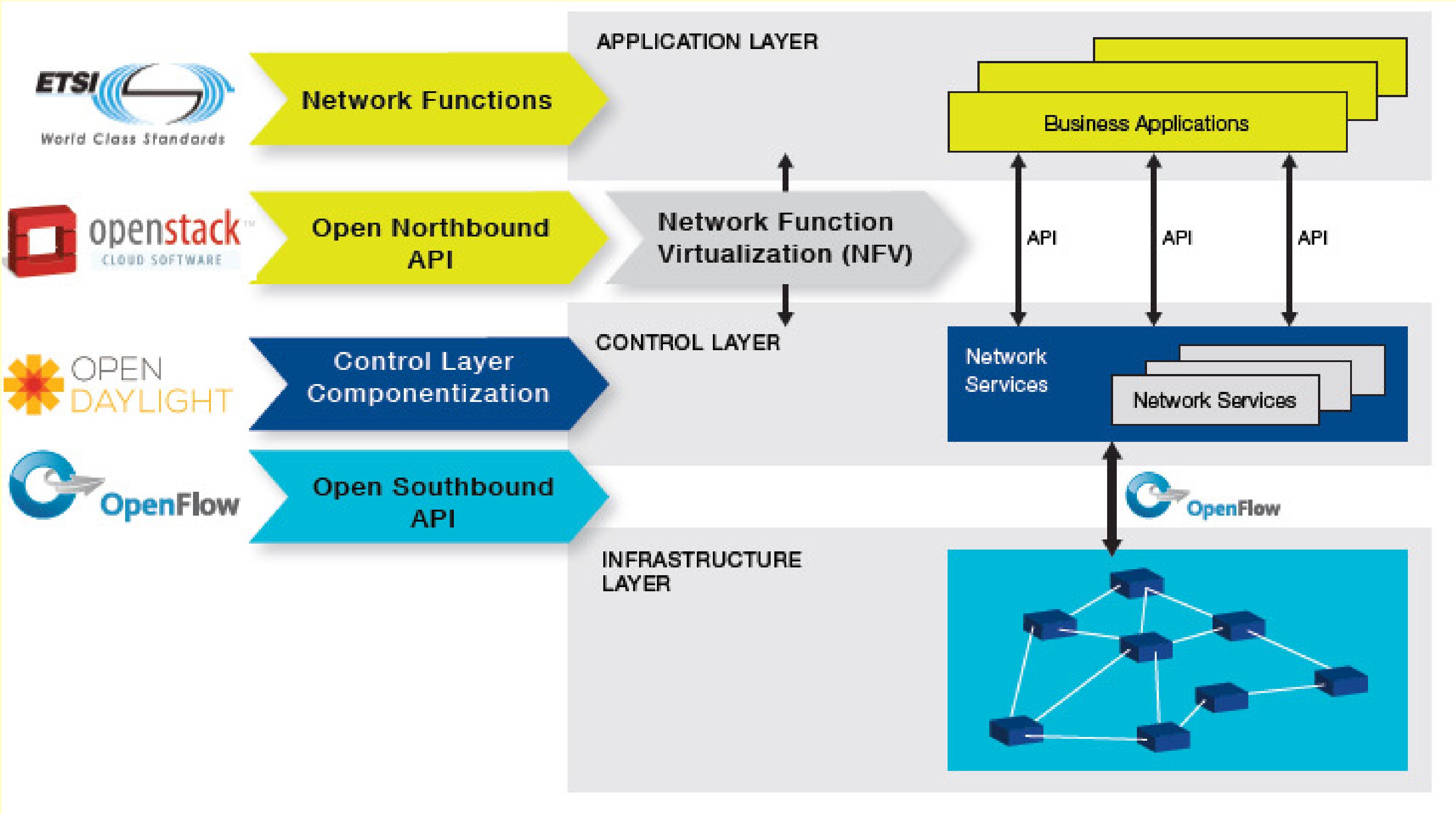


FIGURE 2: NFV and SDN Industry Map

3. NFV SDN-Cooperation

- **OPEN DAYLIGHT**
  - Linux foundation → An open source SW activity
  - 2014: 36 member companies
  - Why open?

  - **General goal:**
  - for SDN and NFV early adoption, create, for the industry benefit an **open, reference framework** for programmability and control through an **open source SDN and NFV solution**
  - develop an SDN controller for a wide range of applications

  - **Aim** :
  - to maintain the flexibility and choice to allow organizations to deploy SDN and NFV at will,
    - but reducing risks of adopting early-stage technologies and integrating in existing infrastructure investments.

3. NFV- SDN Cooperation

- **OPEN DAYLIGHT (cont'd)**
- SW characteristics:
  - **JAVA,** supporting a wide range of I/Fs to applications, principally using REST technologies.
  - Includes a CLI to allow human interaction,
- It supports
  - JAVA RMI for closer coupling to the software.
  - a wide range of protocols for interacting with the network: *NetConf, SNMP, Open Virtual Switch Data Base (OVSDB), OpenFlow, BGP, Path Computation Engine Protocol (PCEP) , Locator/Identifier Separation Protocol (LISP)*
  - The arch. also explicitly allows adding new I/Fs, e.g. proprietary.
  - The system core is based on YANG models to describe the services, I/Fs, data storage
  - This enables automatic code generation (not fully) and a common model-driven dispatch mechanism to support the flexibility needed.

3. NFV- SDN Cooperation

- **OpenStack**
  - OpenStack - **free and open-source cloud computing software platform**
  - Users primarily deploy it as an Infrastructure as a Service (IaaS) solution
  - It consists of a series of interrelated projects that control
    - pools of processing, storage,
    - and networking resources throughout a data center
  - through a web-based dashboard, command-line tools, or a RESTful API.
  - OpenStack.org released it under the terms of the Apache License.
  - OpenStack start:2010 as a joint project of Rackspace Hosting and NASA.
  - Currently -managed by the **OpenStack Foundation**
    - non-profit corporate entity established in September 2012 to promote OpenStack software and its community
  - More than 200 companies have joined the project

3. NFV- SDN Cooperation

- **OpenStack (cont'd)**
  - OpenStack has a modular architecture with various code names for its components.
  - Examples of components
  - Compute (Nova)
    - It is a cloud computing fabric controller, which is the main part of an IaaS system.
    - designed to manage and automate pools of computer resources and can work with widely available virtualization technologies, as well as bare metal and high-performance computing (HPC) configurations.
    - KVM, VMware, and Xen are available choices for hypervisor technology
  - Object Storage (Swift)
    - a scalable redundant storage system
    - objects and files are written to multiple disk drives spread throughout servers in the data center, with the OpenStack SW responsible for ensuring data replication and integrity across the cluster

3. NFV- SDN Cooperation

- **OpenStack (cont'd)**
  - Examples of components
  - **Networking (Neutron)**

  - It is a system for managing networks and IP addresses.
  - It ensures the network is not a bottleneck or limiting factor in a cloud deployment, and gives users self-service ability, even over network configurations.

  - It provides networking models for different applications or user groups
    - flat networks or VLANs that separate servers and traffic
  - It manages IP addresses, allowing for dedicated static IP addresses or DHCP.
  - Users can create their own networks, control traffic, and connect servers and devices to one or more networks.

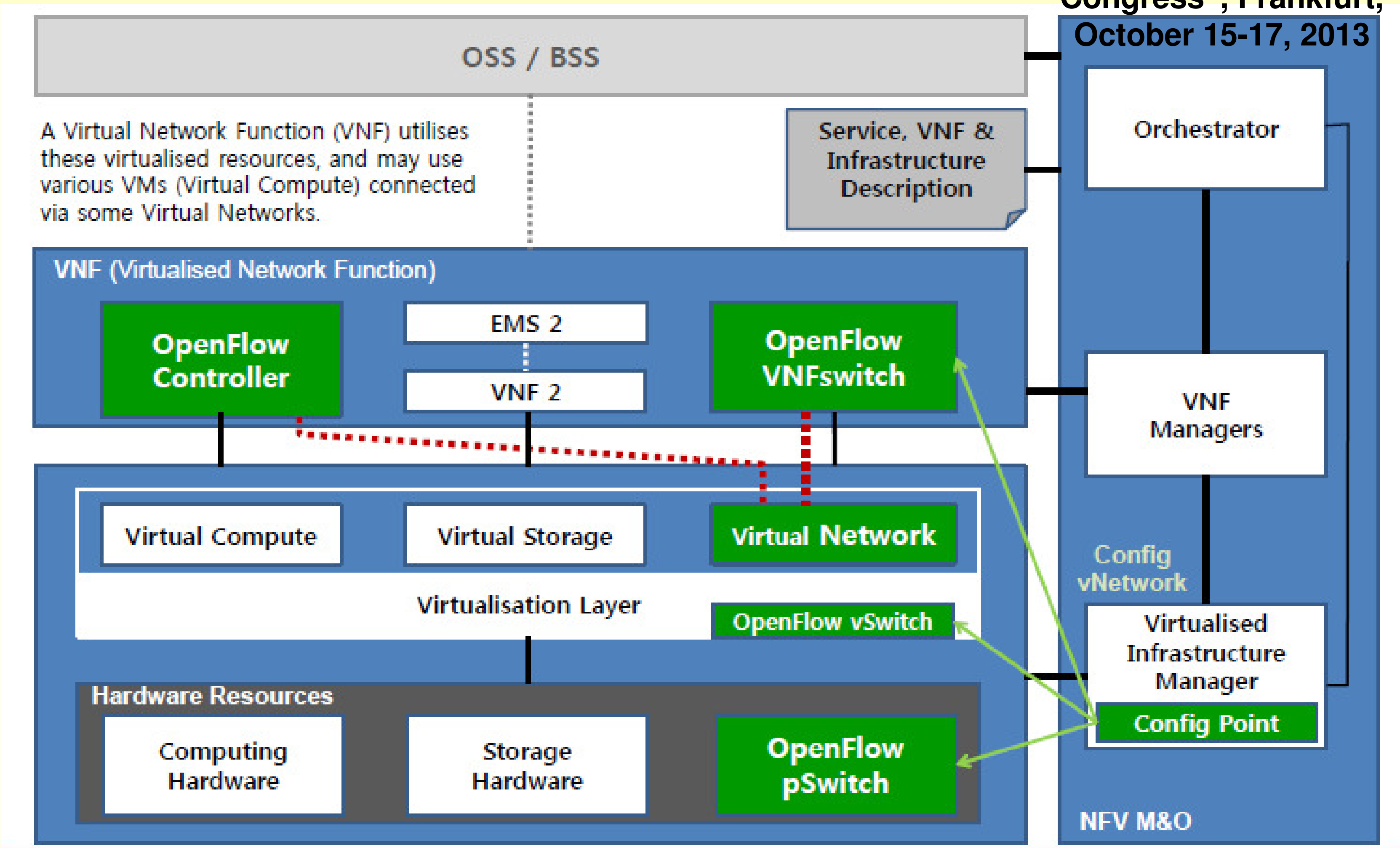
  - Administrators can use SDN technology like OpenFlow to support high levels of multi-tenancy and massive scale

  - It provides an extension framework that can deploy and manage additional network services—such as intrusion detection systems (IDS), load balancing, firewalls, and virtual private networks (VPN)

3.NFV-SDN Cooperation

- NFV - SDN – Architectural stack

Source: "SDN and OpenFlow World Congress", Frankfurt, October 15-17, 2013



3. NFV- SDN Cooperation

- **SDxCentral defined and categorized the most common SDN & NFV Use Cases**
  - **https://www.sdxcentral.com/sdn-nfv-use-cases/**

- **Network Access Control**
  - Description
    - Set appropriate privileges for users or devices accessing the networks: access control limits, incorporation of service chains and/or QoS
    - Generally follows the user/device as they connect from different parts of the network.
  - **Specific Use Cases**: Campus NAC, Remote Office/Branch NAC, M2M NAC, Unified Communications Optimization

- **Network Virtualization**
  - Creates an abstracted VN on top of a physical network;
    - allows a large number of multi-tenant networks to run over a Phy network,
    - spanning multiple racks in the data center or locations if necessary
    - including fine-grained controls and isolation
    - insertion of acceleration or security services
  - **Specific Use Cases:Data** Center Virtual Networks, Campus / Branch Virtual Networks, Data Center Micro Segmentation, Network Functions as a Service

3. NFV SDN Cooperation

- **SDxCentral Use Cases (cont'd)**

- **Virtual Customer Edge**
  - Virtualizing the customer edge either through creation of a virtualized platform on customer premises or by pulling in the functions closer to the core on a virtualized multi-tenant platform hosted either in a carrier point-of-presence,regional datacenter, central datacenter (enterprise, telco or over-the-top cloud SP)

  - **Specific Use Cases:** On-premise vCPE, On-premise vCPE (OTT),vCE (Telco),vCE (OTT)(aka: Cloud CPE)

- **Dynamic Interconnects**
  - Creation of dynamic links between locations, including between DCs, enterprise and DCs, and other enterprise locations, as well as dynamically applying appropriate QoS and BW allocation to those links.

  - **Specific Use Cases**: BWoD, Virtual Private Interconnects / Cloud Bursting, Dynamic Enterprise VPN, Cross Domain Interconnect, Multi-Layer Optimization

3. NFV - SDN Cooperation

- SDxCentral Use Cases (cont'd)

- **Virtual Core and Aggregation**
  - Virtualized core systems for service providers including support infrastructure such as vIMS, vEPC, as well as dynamic mobile backhaul, virtual PE and NFV GiLAN infrastructure

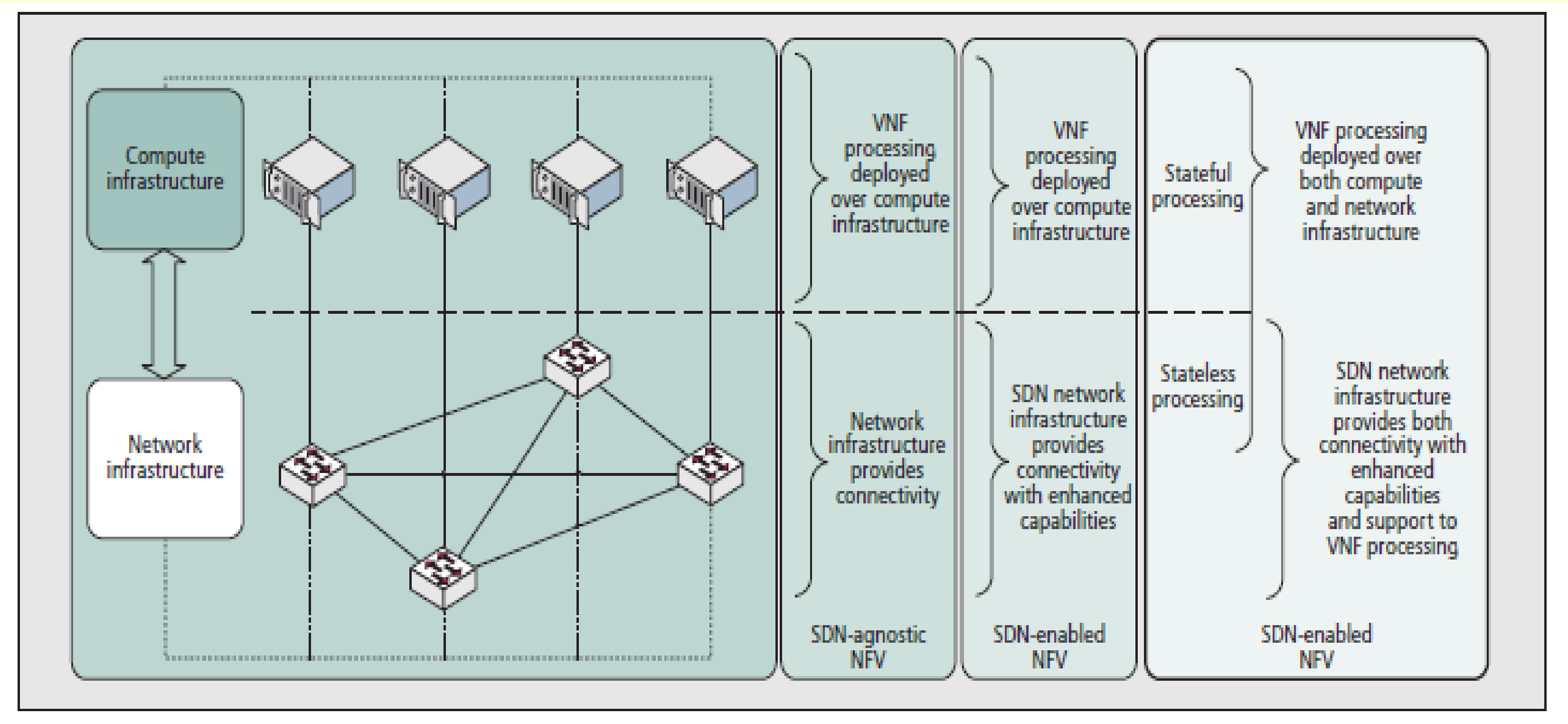
  - **Specific Use Cases:** vEPC & vIMS, vPE, GiLAN, Mobile Network Virtualization

- **Datacenter Optimization**
  - Using SDN and NFV, optimizing networks to improve application performance by detecting and taking into account affinities, orchestrating workloads withnetworking configuration (mice/elephant flows)

  - **Specific Use Cases:** Big Data Optimization, Mice/Elephant Flow Optimization

3. NFV SDN Cooperation

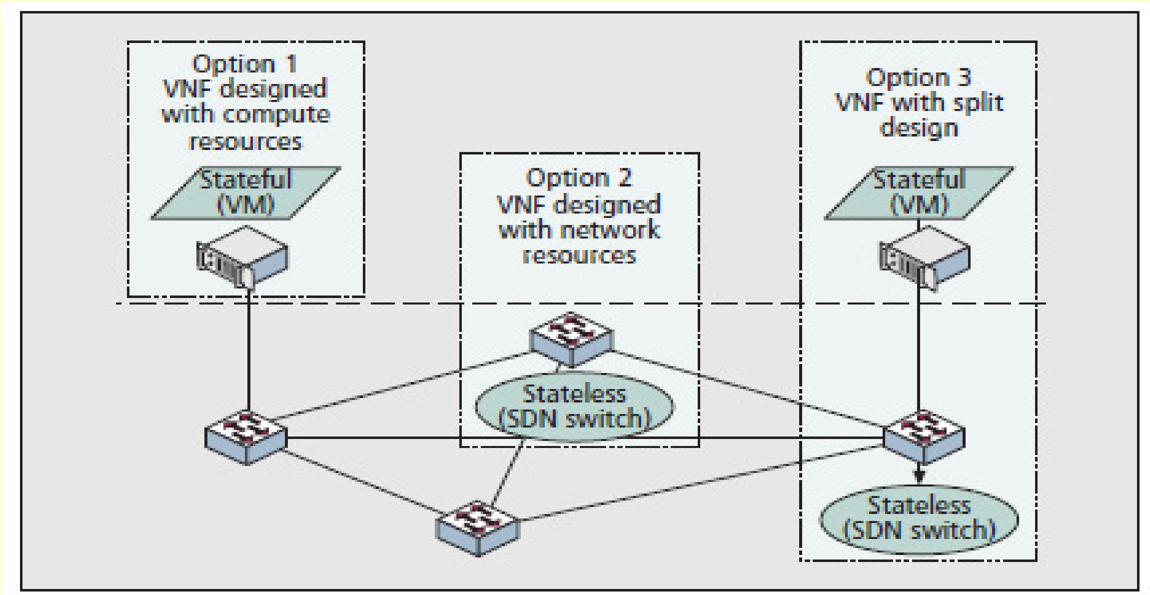
- **SDN enabled NFV Architecture**
- **Architectural evolution variants**
  - SDN- agnostic
  - SDN- aware
  - SDN - enabled



Source: J.Matias, et.al., an SDN-Enabled NFV Architecture, IEEE Comm. Magazine April 2015

3. NFV- SDN Cooperation

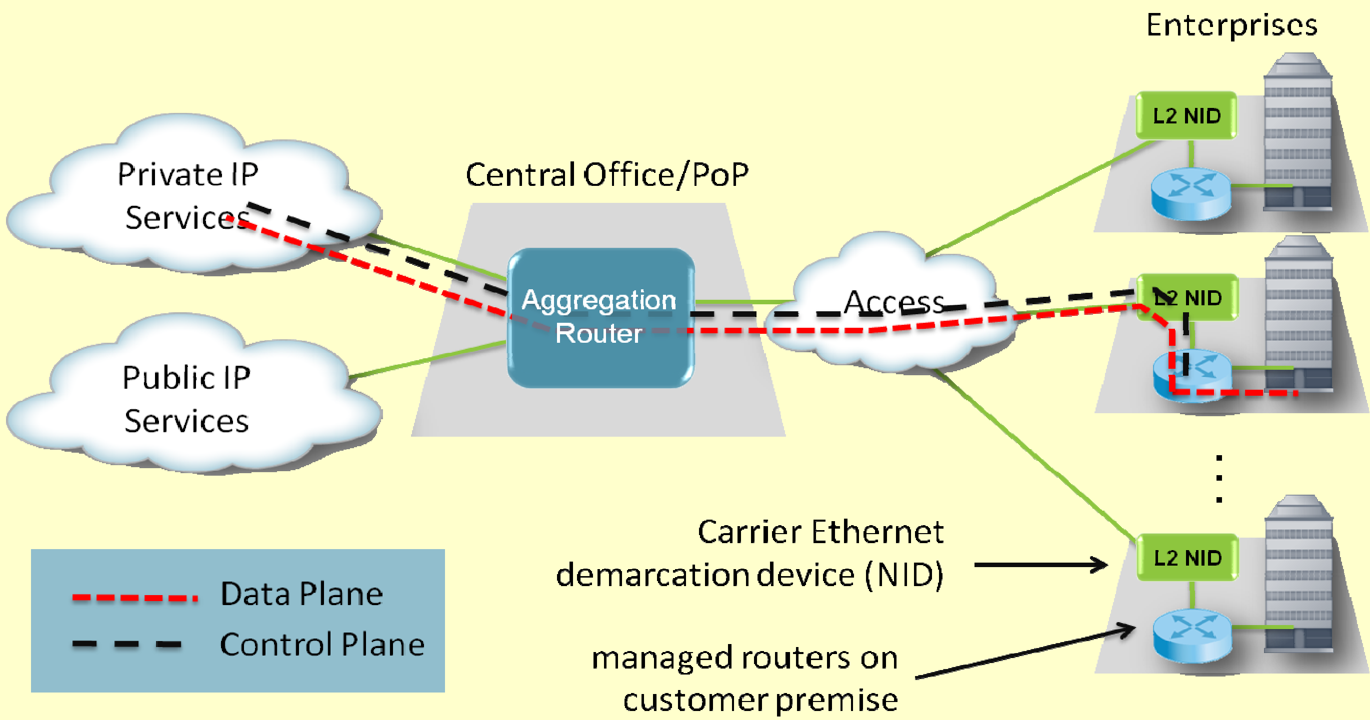
- **SDN enabled NFV Architecture (cont'd)**
  - SDN - enabled NFV could be designed in several variants
    - Source: J.Matias, et.al., an SDN-Enabled NFV Architecture, IEEE Comm. Mag. April 2015
  - VNF design approaches;
    - solved with compute resources
    - solved with network resources
    - solved by splitting the design : separation of stateful and stateless components
  - Option selection: trade-off – scenario/ complexity/ performance, etc.



Source: J.Matias, et.al., an SDN-Enabled NFV Architecture, IEEE Comm. Magazine April 2015

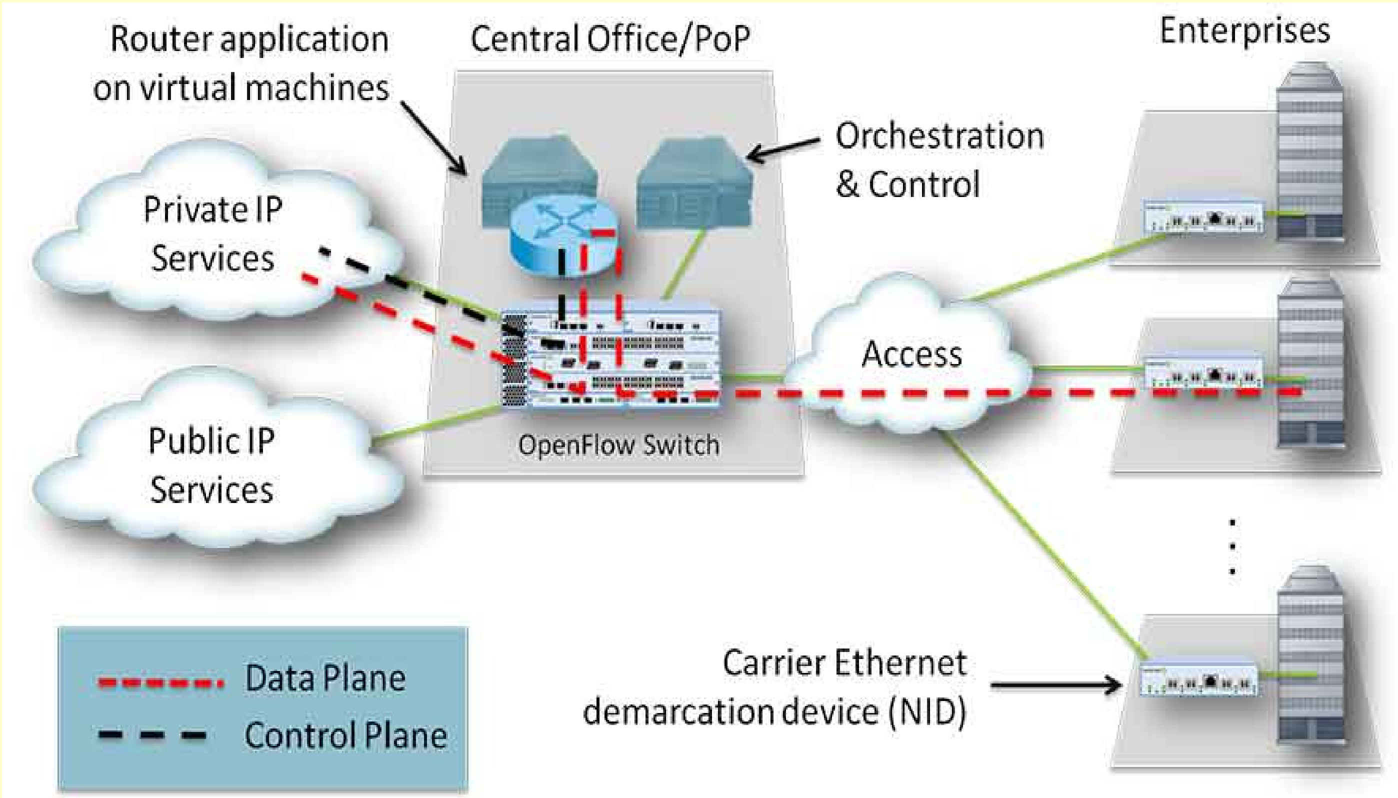
3. NFV SDN Cooperation

- **Example 1: NFV –SDN use case**
  - **Managed router service using a router at the customer site**
  - **Today solution**
  - Source: https://www.sdxcentral.com/articles/contributed/nfv-and-sdn-whats-the-difference/2013/03/



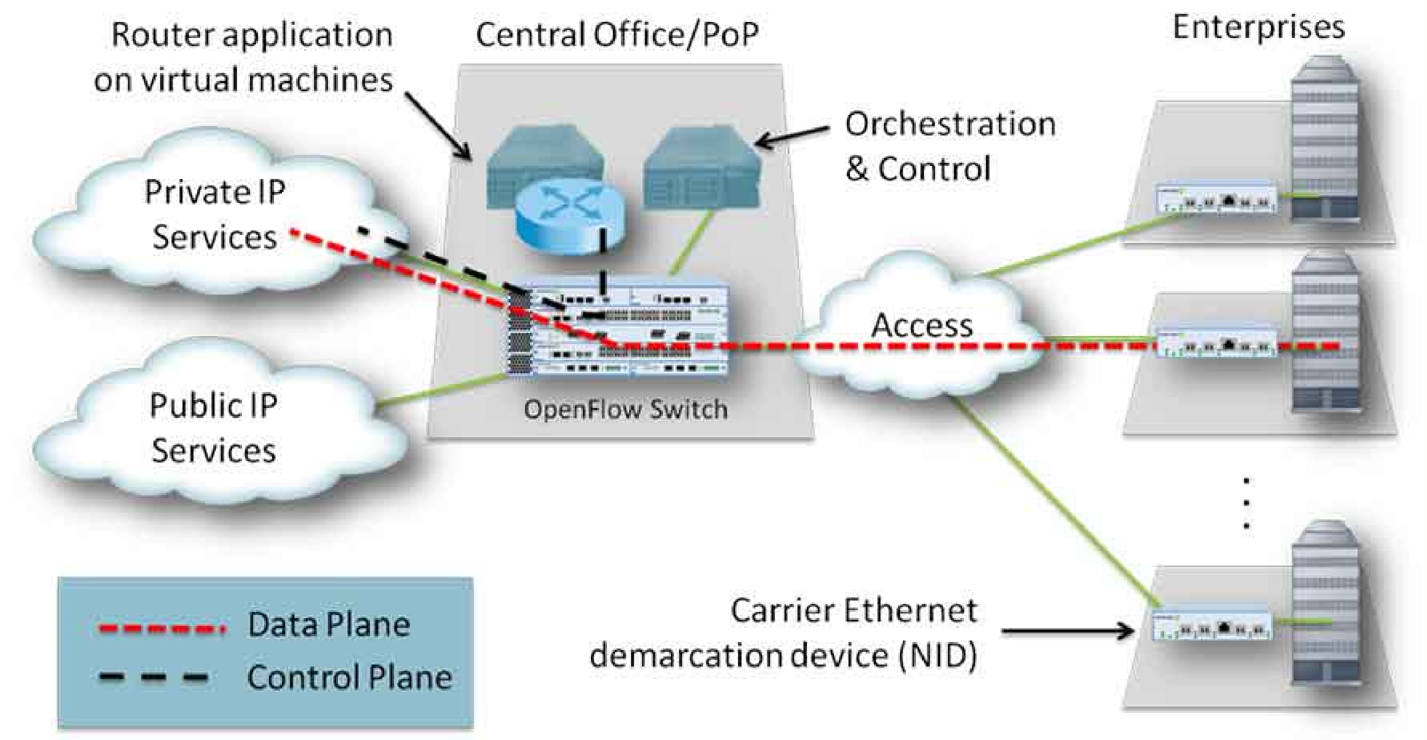
3. NFV- SDN Cooperation

- **Example 1: NFV –SDN use case**
  - **Managed router service using a router at the customer site**
  - **NFV approach**



3. NFV - SDN Cooperation

- **Example 2: NFV –SDN use case**
  - **Managed router service using a router at the customer site**
  - **NFV & SDN approach**
    - **Better performance**



3. NFV- SDN Cooperation

- Telco Cloud Environment- perspectives
- **Interdependence clouds - telecommunications**
- **Telcos *supporting* the cloud**
  - In a cloud environment, communication endpoints are user devices and VMs hosted in different PHY locations
  - New requirements: network capacity should be dynamically adjustable depending on demands variations
  - New challenges to the network, including both the data center (DC) and the WAN segments
  - To provide assured levels of performance to cloud services,
    - cloud and telco services need to be provisioned, managed, controlled, and monitored in an integrated way.

3. NFV - SDN Cooperation

- **Telco Cloud Environment- perspectives**
- **Telcos *using* the cloud**
- Current solution for services: the establishment, management, and composition of service functions (SFs) (e.g., router, firewall) follow a rigid, static, and time consuming process.
  - (Costly over-provisioning is usually applied)
- Cloud computing + SDN + NFV can make SF management processes more agile.
- SF is a FB for a specific treatment of received packets and has well defined external interfaces
  - It can be embedded in a virtual instance or directly in a PHY element
- Virtual SFs allow to compose and organize virtual SFs dynamically
  - opening a new set of business opportunities — and technical challenges

3. NFV - SDN Cooperation

- **Telco Cloud Environment- perspectives**

- **Telcos *using* the cloud (cont'd)**

- **SF chaining (SFC)** is of high interest

- SFC definition: it is "an ordered set of SFs applied to packets and/or frames selected as a result of classification"
  - It is a particular case of service composition
  - It requires the placement of SFs and the adaptation of traffic forwarding policies of the underlying network to steer packets through an ordered chain of service components

- Current status: the lack of automatic configuration and customization capabilities increases the operational complexity

3. NFV- SDN Cooperation

- Telco Cloud Environment- perspectives
- **Carrier Cloud Opportunities**

- Traditional cloud infrastructures are not suitable for all types of businesses, neither to network SFs. Most network SFs have carrier grade requirements (guaranteed QoS, availability, fault recovery, security, etc.
- Telcos, can create an E2E cloud, integrating their network management capabilities, but adapted to a more agile and cloud service-oriented operation model (on-demand, self-service, elastic).

- A near-future telco cloud infrastructure could comprises traditional centralized DC domains + WAN domain
- The telco has the advantage of its already established distributed facilities (PoPs) to host small cloud environments.
- It is also possible for this distributed cloud infrastructure to extend itself into the customer site.
- **Clouf4NFV is such an example**

3. NFV- SDN Cooperation

- **Telco Cloud Environment- perspectives**
- **Carrier Cloud Opportunities**

- Traditional cloud infrastructures are not suitable for all types of businesses, neither to network SFs.
  - Most network SFs have carrier grade requirements (guaranteed QoS, availability, fault recovery, security, etc.

- Telcos, can create an E2E cloud, integrating their network mgmt. capabilities, but adapted to a more agile and cloud service-oriented operation model (on-demand, self-service, elastic).

- A near-future telco cloud infrastructure could comprises traditional centralized DC domains + WAN domain
- Telco advantage: already has established distributed facilities (PoPs) capable to host small cloud environments.
- Such distributed cloud infrastructure can be extended into the customer site.
- **Clouf4NFV Platform is such an architectural example**

3. NFV- SDN Cooperation

- **Telco Cloud Environment- perspectives**
- **Carrier Cloud Opportunities**

- **Cloud4NFV Platform**

  - It builds on cloud, SDN, and WAN technologies to allow SFs to be managed on an *as-a-service* basis

  - It aims to to improve the management of SFs within telco environment,
    - but can also be used to build new services based on the concept SF-as-a-Service (SFaaS),
    - SFs or bundles containing a combination of SFs can be offered as a service to customers

3. NFV SDN Cooperation

- Telco Cloud Environment- perspectives
- **Carrier Cloud Opportunities**

- **Cloud4NFV Platform**

  - **Functionalities**
  - Automated deployment, configuration, and life cycle management (instantiation, configuration, update, scale up/down, termination) of SFs

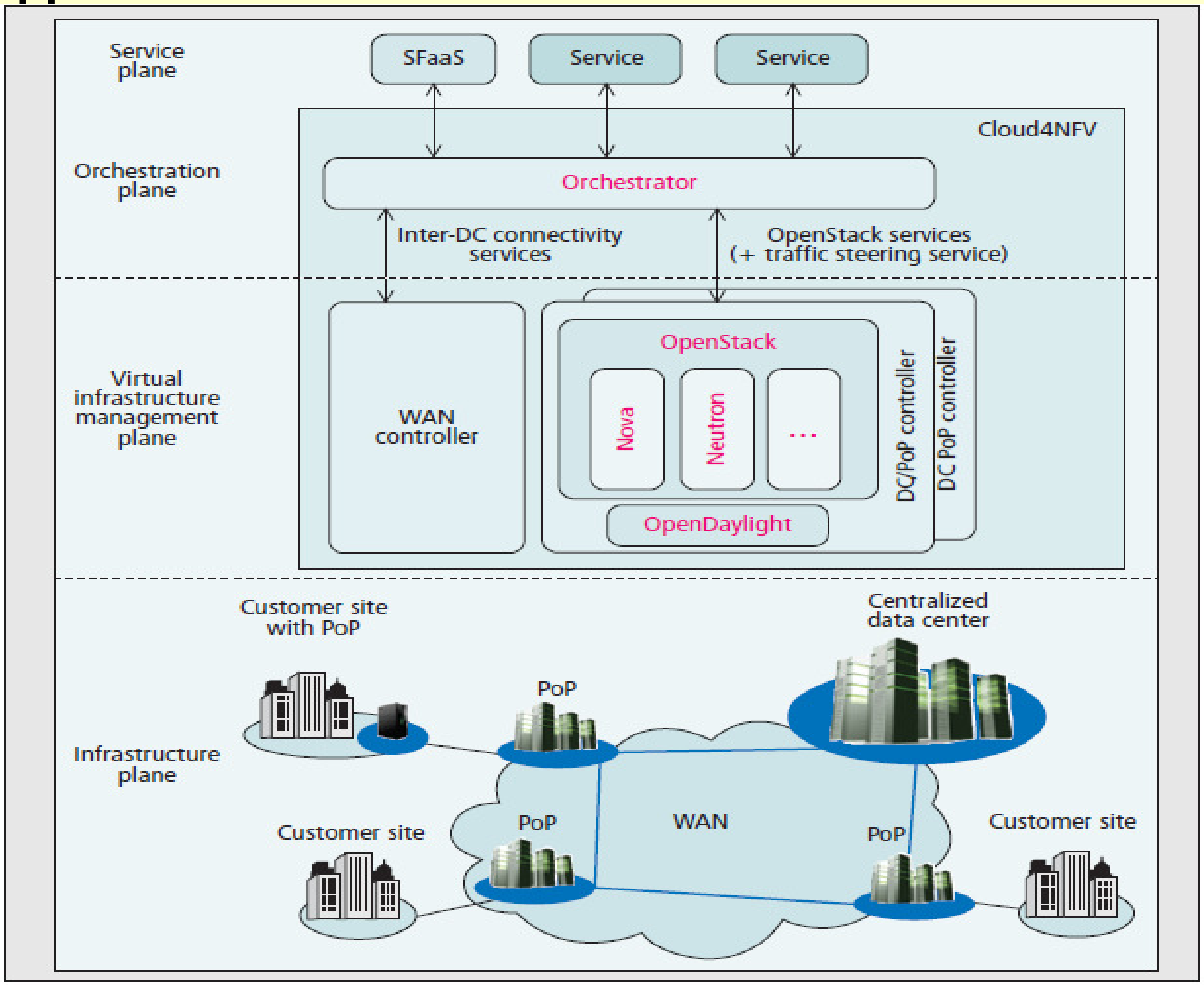
  - Exposure of functionalities such as service :deployment and provisioning, monitoring, reconfiguration, and teardown

  - Federated management and optimization of WAN and cloud resources for accommodating SFs

  - Support of SF composition through SFC

3. NFV SDN Cooperation

- Telco Cloud Environment- perspectives
- **Carrier Cloud Opportunities**



Cloud4NFV Platform architecture

Source:

João Soares,et.al, Toward a Telco Cloud Environment for Service Functions, IEEE Communications Mag., Feb. 2015

4. Conclusions

- NFV major recent IT evolution – driven by telecom world
  - NFV becomes a real technological trend

- SDN is a major development driven mainly by the industry
  - SDN is actually maturing

- **Cooperation (Convergence?) of them is expected**

- Standardization organizations and forums cooperation is progressing towards this goal
  - ETSI, ONF,IETF,IRTF, ITU-T, etc.

- Open Projects – bring significant contributions: OpenStack, OpenDaylight, etc.

- Thank you !
- Questions?

List of Acronyms

- AAA Authentication Authorization and Accounting
- BBU Baseband Processing Unit
- BGP Border Gateway Protocol
- BNG Broadband Network Gateway
- BRAS Broadband Remote Access Server
- BS Base Station
- BSS Business Support System
- CC Cloud Computing
- CCN Content Centric Networking
- CDN Content Distribution Network
- CON Content Oriented Networking
- COTS Commercial-off-the-Shelf
- CPI Control Plane
- CSCF Call Session Control Functions
- DHCP Dynamic Host Configuration Protocol

List of Acronyms

- HLR Home Location Register
- HSS Home Subscriber Server
- IaaS Infrastructure as a Service
- ICN Content Centric Networking
- IETF Internet Engineering Task Force
- IRTF Internet Research Task Force
- IMS IP Multimedia System
- ISG Industry Specification Group
- IT Information Technology
- ITU-T International Telecommunication Union
- LISP ` Locator/Identifier Separation Protocol
- LTE Long Term Evolution
- MANO Management and Orchestration
- M&O Management and Orchestration
- MME Mobility Management Entity
- NaaS Network as a service

List of Acronyms

- NFV Network Functions Virtualization
- NFVI Network Functions Virtualization Infrastructure
- NFVO Network Functions Virtualization Orchetsration
- NFVIaaS Network Functions Virtualization Infrastructure as a Service
- NGN Next Generation Network
- NS Network Service
- NOS Network Operating System
- ONF Open Networking Foundation
- OSS Operations Support System
- PaaS Platform as a Service
- PCEP Path Computation Engine Protocol
- PDN Packet Data Network
- PE Provider Edge
- PNF Physical Network Function

List of Acronyms

▪	SaaS	Software as a Service
▪	SDN	Software Defined Network
▪	SDO	Standards Development Organisation
▪	SGSN	Serving GPTRS Support Node
▪	SLA	Service Level Agreement
▪	S/P-GW	Serving and Packet Data Networks Gateway
▪	SP	Service Provider
▪	TCO	Total Cost of Ownership
▪	VIM	Virtual Infrastructure Manager
▪	VL	Virtualization Layer
▪	VLAN	Virtual Local Area Network
▪	VPLS	Virtual Private LAN Service
▪	VxLAN	Virtual Extensible Local Area Network
▪	VM	Virtual Machine
▪	VN	Virtual Network
▪	VNF	Virtual Network Function
▪	VNF-FG	VNF Forwarding Graph
▪	VNFM	Virtual Network Function Management
▪	VNPaaS	Virtual Network Platform as a Service
▪	VPN	Virtual Private Network
▪	WG	Working Group

References-1

1. ETSI- Network Functions Virtualization – Introductory White Paper, https://portal.etsi.org/nfv/nfv_white_paper.pdf
2. Network Functions Virtualization – Update White Paper, https://portal.etsi.org/nfv/nfv_white_paper2.pdf
3. Network Functions Virtualization – White Paper #3, https://portal.etsi.org/Portals/0/TBpages/NFV/Docs/NFV_White_Paper3.pdf
4. ETSI GS NFV 003 V1.2.1 (2014-12), Network Functions Virtualization (NFV);Terminology for Main Concepts in NFV, http://www.etsi.org/deliver/etsi_gs/NFV/001_099/003/01.02.01_60/gs_NFV003 v010201p.pdf
5. ETSI NFV 001 V1.1.1, 2013-10, NFV Use Cases
6. ETSI GS NFV 004 v1.1.1 2013-10, NFV Virtualization Requirements
7. ETSI GS NFV 002 v1.2.1 2014-12, NFV Architectural Framework
8. ONF, "OpenFlow-Enabled SDN and Network FunctionsVirtualization," https://www.opennetworking.org/images/stories/downloads/sdn-resources/solutionbriefs/sb-sdn-nvf-solution.pdf;
9. https://www.sdxcentral.com/sdn-nfv-use-cases/
10. E. Haleplidis *et al.*, "Software-Defined Networking (SDN): Layers and Architecture Terminology," RFC 7426, Jan. 2015; http://tools.ietf.org/search/rfc7426; accessed 01 Feb. 2015.

References-1

11. C.Kolias, Bundling NFV and SDN for Open Networking, NetSeminar @ Stanford, May 22, 2014, http://netseminar.stanford.edu/seminars/05_22_14.pdf
12. Andy Reid, Network Functions Virtualization and ETSI NFV ISG, http://www.commnet.ac.uk/documents/commnet workshop networks/CommNets EPSRC workshop Reid.pdf
13. J.Metzler, The 2015 Guide to SDN and NFV, http://www.webtorials.com/content/2014/11/the-2015-guide-to-sdn-nfv.html
14. C.Marcondes-NFV-WPEIF2014, https://www.google.ro/?gws_rd=ssl#q=ETSI+NFV+USe+cases++tutorial
15. N.McKeown, T.Anderson, et. Al., OpenFlow: Enabling Innovation în Campus Networks, - http://www.openflow.org/documents/openflow-wp-latest.pdf.
16. M.Mendonca, et. al., A Survey of Software-Defined Networking: Past, Present,and Future of Programmable Networks, http://hal.inria.fr/hal-00825087/
17. A. Greenberg, et. al., A clean slate 4d approach to network control and management. ACM SIGCOMM Computer Communication Review, 35(5):41-54, 2005.
18. M. Casado, et.al., Fabric: a retrospective on evolving sdn, Proceedings of the first workshop on Hot topics in software defined networks, HotSDN '12, pages 85{90, New York, NY, USA, 2012. ACM.
19. Open networking foundation, https://www.opennetworking.org/about.
20. Open Networking Research Center (ONRC). , http://onrc.net.
21. Open vswitch and ovs-controller, http://openvswitch.org/.

References-1

22. C.Kolias, Bundling NFV and SDN for Open Networking, NetSeminar @ Stanford, May 22, 2014,
23. J.Matias, J.Garay, N.Toledo, J.Unzilla, and E.Jacob,Toward an SDN-Enabled NFV Architecture, IEEE Communications Magazine April 2015

Annexes

- **Annex 1.**
- **NFV terminology - details**
- **Virtualization Deployment Unit (VDU):** construct used in an information model, supporting the deployment description and operational behaviour
  - of a subset of a VNF
  - or the entire VNF if it was not componentized in subsets
  - In the presence of a hypervisor: a single VNF or VNF subset instance created based on the construct can be mapped to a single VM
  - A VNF may be modelled using one or multiple such constructs, as applicable
- **Virtualized Network Function Component (VNFC):** VNF internal component
  - providing a VNF Provider a defined sub-set of that VNF's functionality,
  - a single instance of this component maps 1:1 against a single Virtualization Container

Annexes

- **Annex 1.**
- **NFV terminology - details**

- **Virtualized Network Function Component (VNFC) Instance:** instance of a VNFC deployed in a specific Virtualization Container instance.
  - It has a lifecycle dependency with its parent VNF instance

- **Virtualized Network Function Package (VNF Package):** archive that includes a VNFD, the software image(s) associated with the VNF, as well as additional artefacts, e.g. to check the integrity and to prove the validity of the archive

1. Network Function Virtualization

- **1.5 NFV Requirements**
- **Resiliency- details**
  - The NFV framework shall be able to
    - Provide mechanisms to recreate (on demand, or automatic) NFs after a failure
    - Provide means to classify sets of VNFs having similar reliability/availability reqs., in resilience categories
    - Support standard- based replication of state data (sync./async.) and preservation of data integrity as to fulfill SLAs
  - The NFV framework (including orchestration and other necessary functions for service continuity) shall facilitate resiliency schemes both in CPI and DPI
  - The SLA shall specify the metrics to define the values and variability of “stability”

  - Network stability needs ⇒ the NFV framework shall support mechanisms to measure the following metrics and ensure that they are met per each SLA
    - *Max non-intentional packet loss rate* (packets dropped due to over-subscription of the servcie intrterconnects and not due to police or filters)
    - *Max rate of non-intentional drop of stable calls* or sessions
    - *Max latency and delay variation* on a per-flow basis
    - *Max time to detect and recover from faults*
    - *Max failure rate of transactions*

1. Network Function Virtualization

- **1.5 NFV Requirements- details**
- **Operational and Management Requirements**
  - The NFV framework shall incorporate mechanisms for automation of operational and mgmt. functions (e.g. creation, scaling, healing of VNF instances) based on
    - pre-defined criteria specified in the VNF info model
    - network capacity adaptation to load
    - SW upgrades
    - new features/nodes introduction
    - function configurations and relocation
  - The NFV framework shall provide M&O functionality, for VNF instances lifecycle : *instantiation, resource allocation, scaling, termination*
  - The M&O functionality shall be able to
    - be limited to the differences introduced by the NFV process
    - be neutral w.r.t the logical function provided by VNFs
    - interact with other operation systems (if they exist), managing the VNFs and/or NFVI
    - use std. information models that describe how to manage the VNF lifecycle

1. Network Function Virtualization

- 1.5 NFV Requirements
- **Operational and Management Requirements (cont'd)- details**
  - The M&O functionality shall be able to
    - manage the VNFs and VNFs instances lifecycle, using the information model in combination with run-time info/constraints, accompanying scheduled or on-demand reqs. regarding VNFs
    - manage the NFVI in coordination with other applicable mgmt. system
    - maintain the integrity of VNFs
    - monitor and collect NFVI resource usage and map such usage against the corresponding VNF instance; monitor resources used per VNF basis; it shall be made aware of NFVI faults and correlate such infoo with other VNF-related info
    - support std. APIs supporting all its action
    - manage policies and constraints (e.g replacing VMs); enforce policies and constraints when allocating and resolving conflicts regarding NFVI resources
    - manage the assignment of NFVI resources to VNFs- assuring the sharing of resources

1. Network Function Virtualization

- **1.5 NFV Requirements**
- **Service Models details**
  - The NFV framework shall
    - facilitate NOs to consume NFVI resources operated by a different infrastructure provider, via std. APIs, without degraded capabilities- compared to self-operated infrastructure resources
    - allow VNFs from different SPs to coexist on the same infrastructure, while facilitating appropriate isolation between the resources allocated to different SPs
    - allow each SP to assure and bill its services for its users
    - allow instantiation of different VNFacross infrastructures thata are independently administered
    - provide mechanisms for AAA, measurements, notifications and diagnostics for SLAs
    - Maintain the relationships between
      - VNFs/applications
      - The service making use of them
      - The resources allocated
- **Maintenance models**
  - ETSI defined also serviceability and maintainability requirements describing how to run an NFV framework and the hosted VNFs
  - They mostly deals with exchanging SW or HW and with diagnosis and elimination of problems