SDN Modern Networking Fundamentals

Network Function Virtualization

Future Internet Laboratory
School of Electronics and Telecommunications, Hanoi University of Science and Technology
Contact: thanh.nguyenhuu@hust.edu.vn

Contents

- Concepts
- Why NFV?
- Service provisioning, service orchestration
- NFV platforms
- Issues
- Benefits of NFV

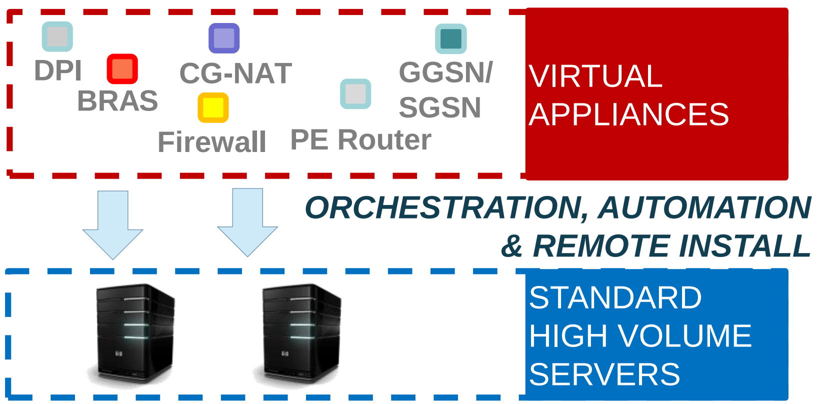
Concept

- A means to make the network more flexible and simple by minimising dependence on HW constraints

Traditional Network Model:
APPLIANCE APPROACH

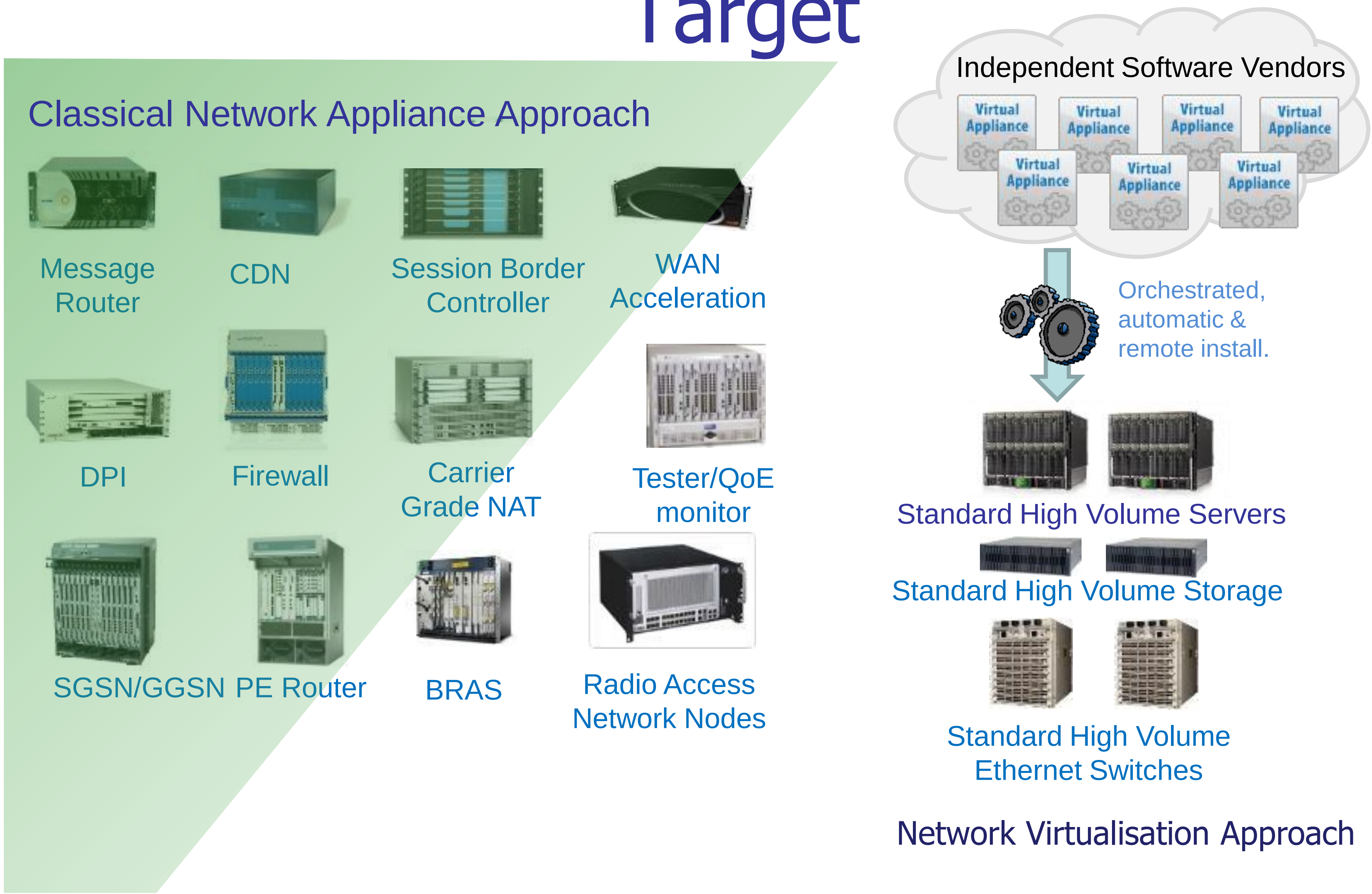
- Network Functions are **based on specific HW&SW**
- **One physical node per role**

Virtualised Network Model:
VIRTUAL APPLIANCE APPROACH



- Network Functions are **SW-based over well-known HW**
- **Multiple roles** over **same HW**

Target



VIỆN ĐIỆN TỬ - VIỄN THÔNG
S E T School of Electronics and Telecommunications

Re-thinking the Network Device Architecture

Conventional device

- Application
- Operating system
- Hypervisor
- Compute infrastructure
- Network infrastructure
- Switching infrastructure
- Rack, cable, power, cooling

NFV approach

- Application
- Virtual network functions
- Operating system
- Hypervisor
- Compute infrastructure
- Switching infrastructure
- Rack, cable, power, cooling

Why NFV?

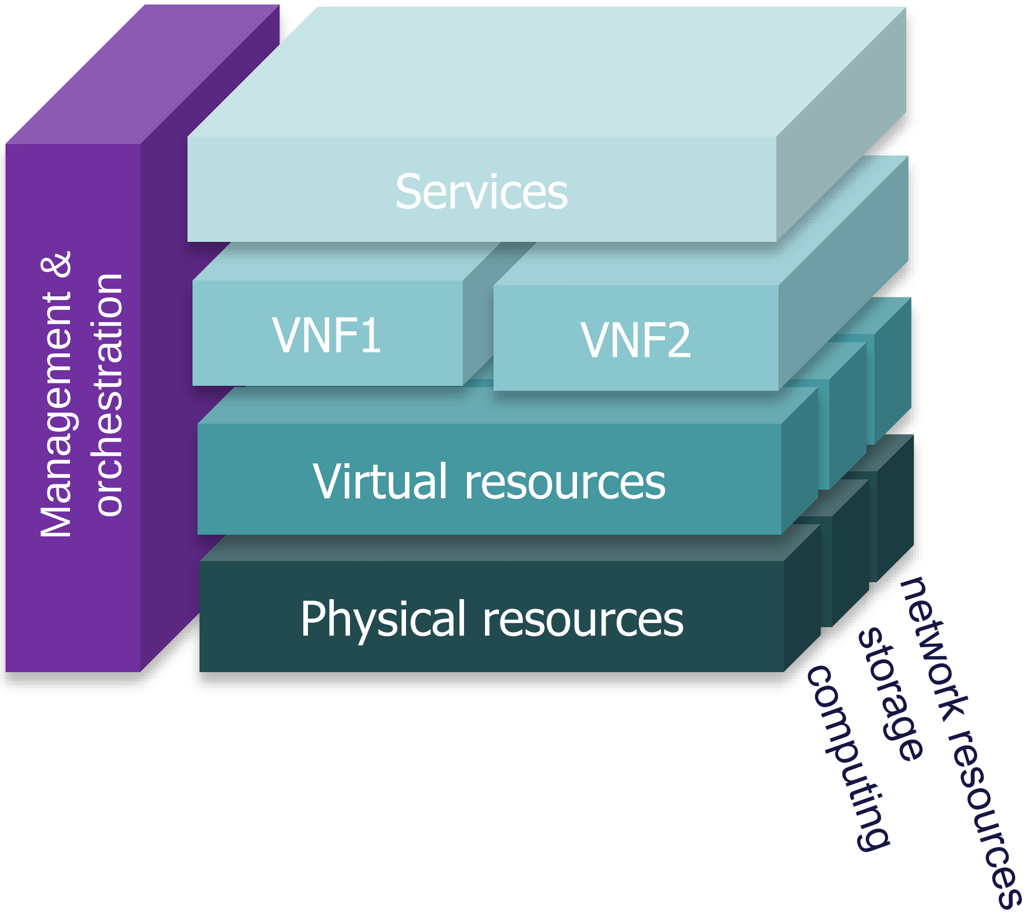
- Implementing network functions in software in stead of proprietary hardware
  → Leveraging (high volume) standard servers and IT virtualization
- Supporting multi-versioning and multi-tenancy of network functions
  → Allowing the deployment of a single physical platform for different applications, users and tenants
- Enabling new ways to implement resilience, service assurance, test and diagnostics and security surveillance
- Providing opportunities for pure software players
- Facilitating innovation towards new network functions and services in a pure software
  → Softwarization and cloudification of network functions
- Applicable to any data plane packet processing and control plane functions, in fixed or mobile networks

Remarks

- NFV will only scale if management and configuration of functions can be automated
- NFV ultimately transforms the way network operators architect and operate their networks, but change can be incremental

High-Level NFV Framework

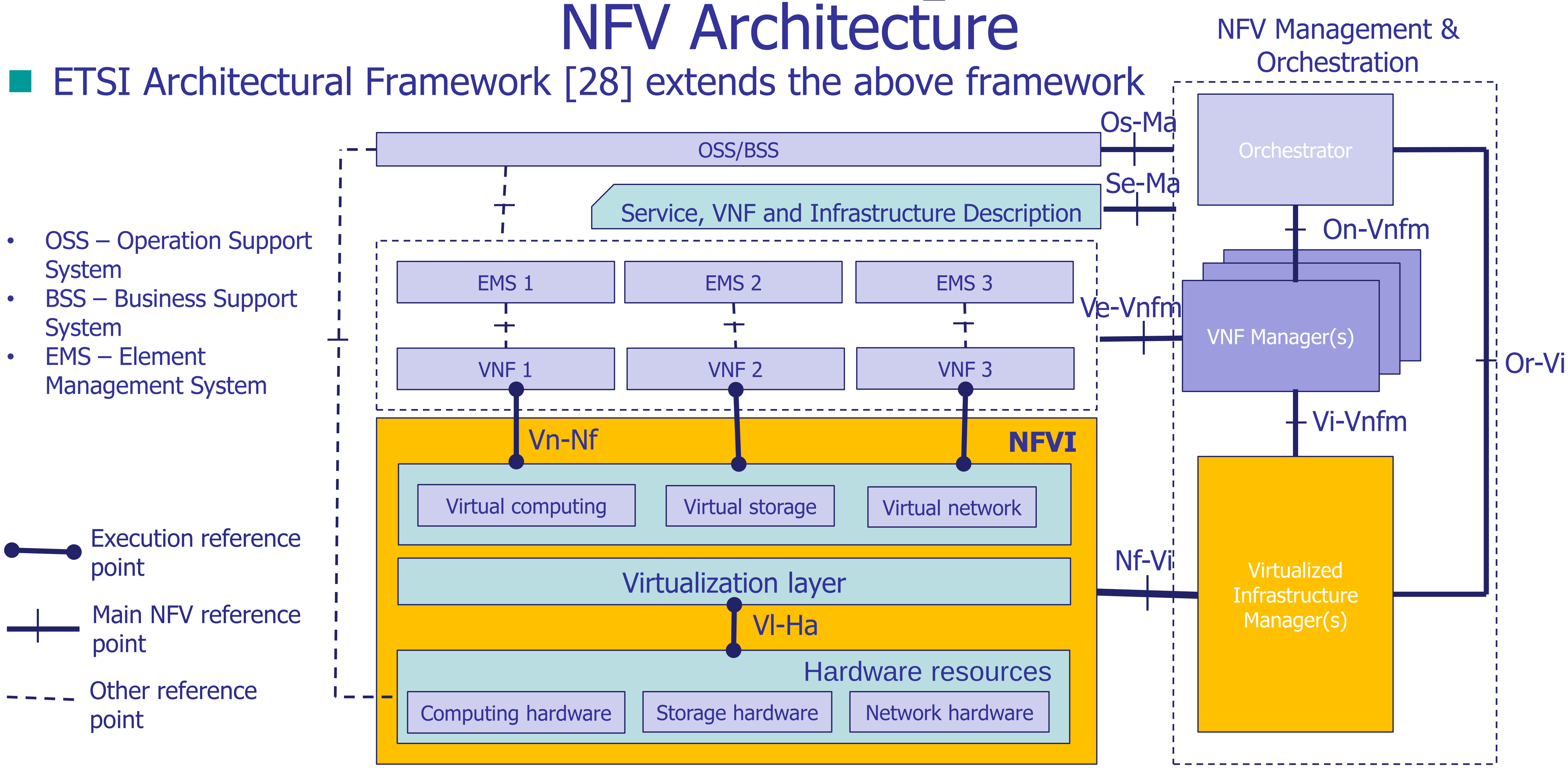
- Physical resources
  - Commercial off-the-shelf computing hardware, storage, network providing processing, storage and connectivity to VNFs
- Virtual resources
  - Abstractions of physical resources
  - Virtual resources are decoupled from physical resources
- Virtual Network Function (VNF)
  - Function blocks with well defined external interfaces and functional behaviors
  - Deployed on virtual resources
- Service
  - Offering provided by Telecom Service Provider composed of one or more VNFs
  - Made up by number, types, order of VNFs of the service → decided by service functional and behavioral specification
- Management and orchestration (NFV MANO)
  - Focusing on virtualization-specific management tasks necessary in NFV framework
  - Defining interfaces used for communications between NFV MANO components and with conventional network components (e.g., OSS and BSS)
  - Provision and configuration of VNFs
  - Orchestration and lifecycle management of resources and VNFs



NFV Architecture

- ETSI Architectural Framework [28] extends the above framework

- OSS – Operation Support System
- BSS – Business Support System
- EMS – Element Management System



NFV Architecture

- Network Function Virtualization Infrastructure (NFVI)
  - Hardware/software components which build up environment deploying, managing and executing VNFs
  - Virtualization layer
    - Abstracts hardware resources and decouples VNF software from underlying hardware
- Virtualized Infrastructure Manager
  - Controls and manages the interaction of a VNF with computing, storage and network resources as well as their virtualization
    - Resource management
    - Operations and monitoring
- Element Management System (EMS)
  - Performs typical management functionalities for VNFs

NFV Architecture

- Orchestrator
  - Orchestration and management of NFV infrastructure and software resources, and realizing network services on NFVI
- VNF Manager
  - VNF lifecycle management (instantiation, update, query, scaling, termination)
- Service, VNF and Infrastructure Description
  - Provides information regarding VNF deployment template, VNF forwarding graph, etc.
- OSS/BSS
  - OSS/BSS of an NFV operator

Design Considerations [22]

- Network architecture and performance
  - Should be similar to services/function running on dedicated hardware
    - →performance evaluation of all layers in the architecture required
    - →function placement mechanisms with resource allocation
- Security and resilience
  - Functions or services from different subscribers should be protected/isolated
  - Network Function Virtualization Infrastructure (NFVI) should be protected from the delivered subscriber services
- Reliability and availability
  - Outage (of TSP) should be below recognizable levels
  - Service recovery should be performed automatically
  - Functions with high level of resiliency should be protected

Design Considerations

- Heterogeneity
  - NFV platforms should be an open, shared environment for vendor-neutral applications/hardwares
  - Enabling end-to-end services to be created on top of multiple infrastructural domains
- Legacy support
  - Providing support for both physical and VNFs
  - Supporting migration from legacy to NFV environments
- Network scalability and automation
  - Should be scalable to million of subscribers
  - Scale only all VNFs are automated
  - VNFs should dynamically be deployed and removed

NFV and SDN

- NFV and SDN are different and independent
  - SDN
    - redefines network architecture
  - NFV
    - redefines network equipment architecture
- Commonalities
  - Move functionality to software
  - Use commodity servers and switches over proprietary appliances
  - Leverage application program interfaces (APIs)
  - Support more efficient orchestration, virtualization, and automation of network services

→ However, SDN makes NFV and NV more compelling and visa-versa

NFV and SDN

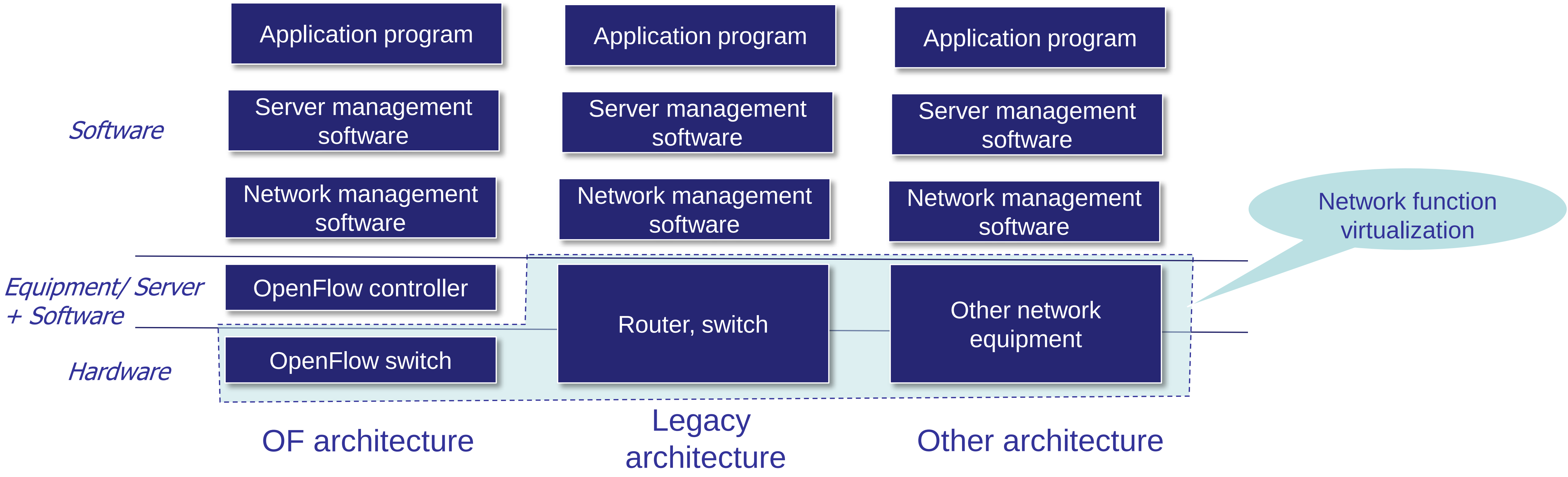
- Why NFV and SDN?
  - Virtualization: use network resource without worrying about where it is physically located, how much it is, how it is organized.
  - Orchestration: manage thousands of devices
  - Programmable: should be able to change behavior on the fly; services are provisioned end-to-end (network, cloud and end-device)
  - Dynamic scaling: should be able to change size, quantity
  - Automation: services are provisioned by automated mechanisms
  - Visibility: monitor resources, connectivity
  - Performance: optimize network device utilization
  - Multi-tenancy: sharing infrastructure by multi-tenants
  - Service integration
  - Openness: Full choice of modular plug-ins

NFV and SDN

	NFV	SDN
Approach	Service/function abstraction	Networking abstraction
Formalization	ETSI NFV Industry Standard Group	ONF
Advantage	Promises to bring flexibility and cost reduction	Promises to bring unified programmable control and open interfaces
Protocol	Multiple control protocols (e.g., SNMP, NETCONF, OpenFlow)	OpenFlow is the de-factor standard
Applications run	Commodity servers and switches	Commodity for control planes and possibility for specialized hardware for data plane

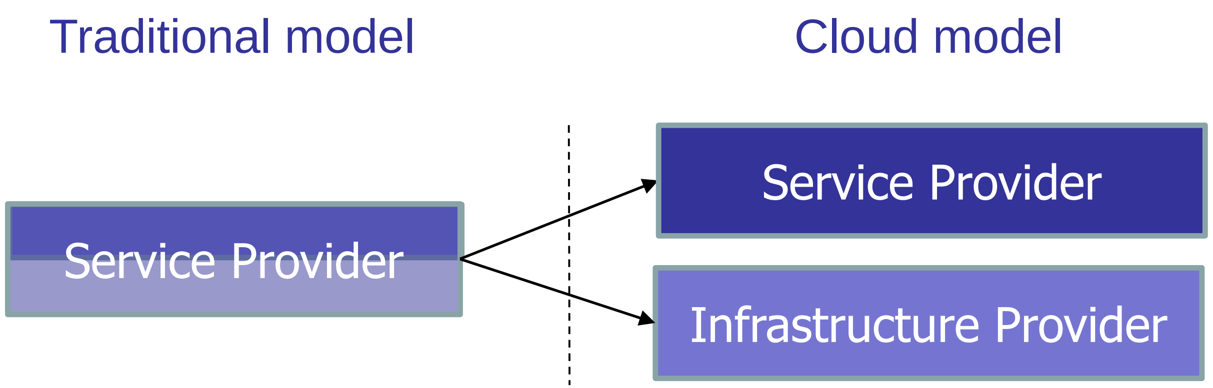
NFV and SDN

■ Scope of NFV and OpenFlow/SDN



NFV and Cloud

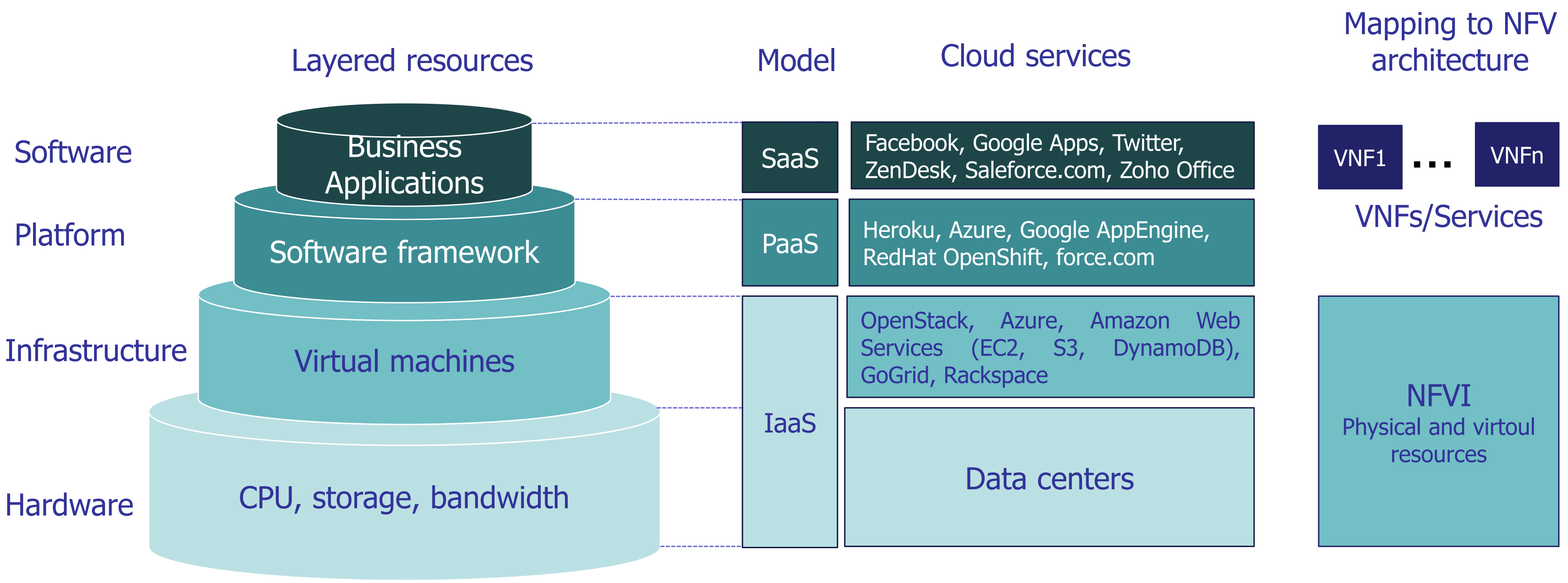
- Cloud model
  - Infrastructure provider
    - manage cloud platforms
    - lease resources according to a usage-based pricing model
  - Service provide
    - rent resources from infrastructure providers to serve the end users
- Service models
  - Software-as-a-Service (SaaS)
  - Platform-as-a-Service (PaaS)
  - Infrastructure-as-a-Service (IaaS)



NFV and Cloud

- It this beneficial to provision VNFs in the cloud
  - →deploying VNFs in dedicated VMs in the cloud
  - Rapid deployment of new services
  - Ease of scalability
  - Reduce duplication: reuse VNFs in different services
  - Efficiency and expense reduction

NFV and Cloud

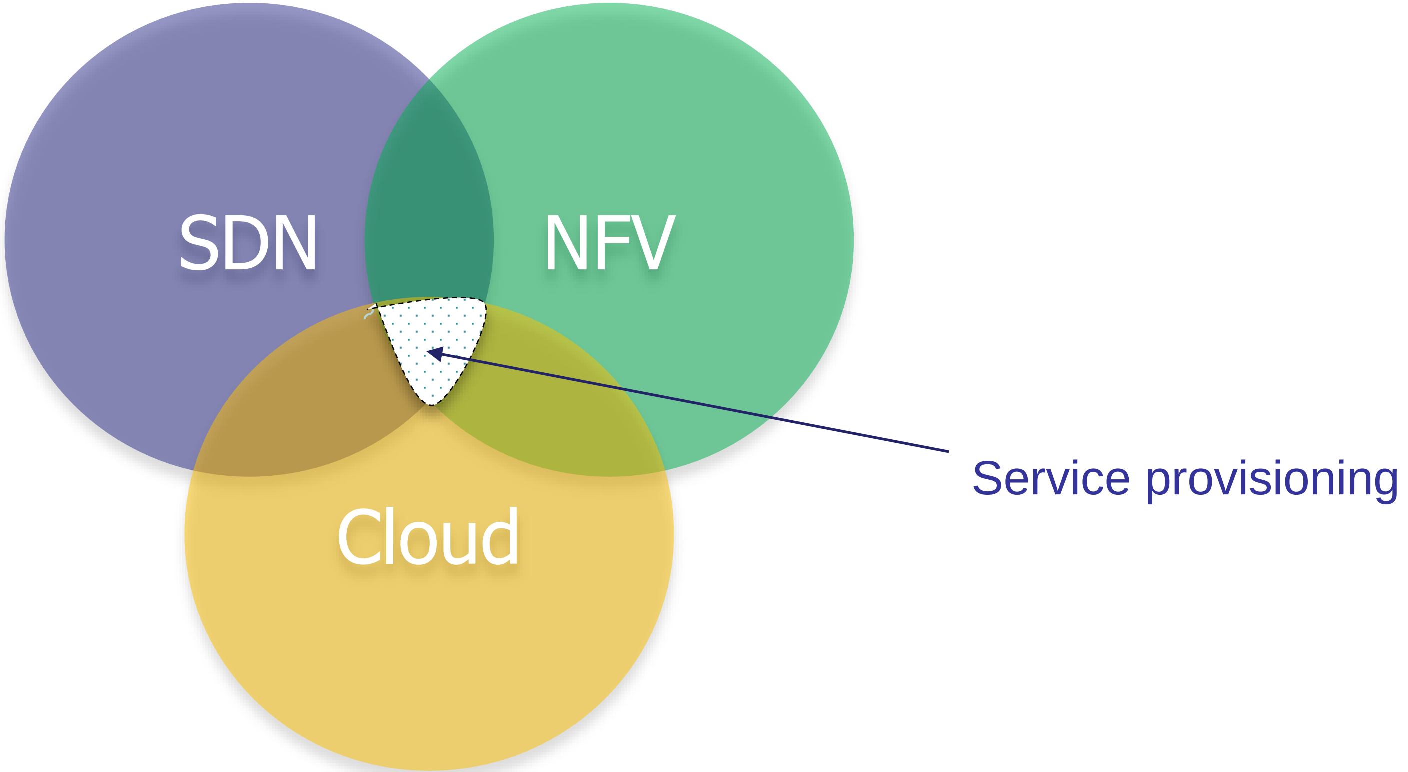


NFV and Cloud

	NFV	Cloud
Approach	Service/function abstraction	Computing abstraction
Formalization	ETSI NFV Industry Standard Group	DMTF (Distributed mnt. task force) Cloud Management Working Group
Latency	Expectation for low latency	Some latency is acceptable
Protocol	Multiple control protocols (e.g., SNMP, NETCONF, OpenFlow)	OpenFlow
Reliability	Strict 5 NINES availability requirements	Less strict reliability requirements

NFV + SDN + Cloud

- Service Provisioning = SDN + NFV + Cloud
  - Although service provisioning does not necessarily require SDN + NFV + Cloud, these technologies enable fast service prototyping, optimization and deployment



State of the Art

- Standardization activities
  - IETF Service Function Chaining Working Group (IETF SFC WG): to work on function chaining [23]
    - Architecture for service function chaining that includes the protocols/protocol extensions to convey the service function chain (SFC) and service function path information
  - IRTF NFV Research Group (NFVRG): The Internet Research Task Force (IRTF) research group NFVRG
    - Focusing on research problems associated with NFV-related topics
  - ATIS NFV Forum
    - Industry group created by the Alliance for Telecommunications Industry Solutions (ATIS)
    - Aiming at developing specifications for NFV, focusing on aspects of NFV including inter-carrier inter-operability and new service descriptions and automated processes

State of the Art

- NFV Industry Specification Groups
  - An Industry Specifications Group (ISG) has been formed under ETSI to study NFV
    - ETSI is the European Telecommunications Standards Institute with >700 members
  - NFV Members
    - Acme Packet, Allot, Amdocs, AT&T, ALU, Benu Networks, Broadcom, BT, Cablelabs, Ceragon, Cisco, Citrix, DT, DOCOMO, ETRI, FT, Fraunhofer FOKUS, Freescale, Fujitsu Labs, HP, Hitachi, Huawei, IBM, Intel, Iskratel, Italtel, JDSU, Juniper, KT, MetraTech, NEC, NSN, NTT, Oracle, PT, RadiSys, Samsung, Seven Principles, Spirent, Sprint, Swisscom, Tektronix, TI, Telefon Ericsson, Telefonica, TA, Telenor, Tellabs, UPRC, Verizon UK, Virtela, Virtual Open Systems, Vodafone Group, ZTE
  - Working and expert groups
    - Architecture of the Virtualisation Infrastructure (INF)
    - Management & Orchestration (MANO)
    - Performance & Portability Expert Group (PER)
    - Reliability & Availability (REL)
    - Security Expert Group (SEC)
    - Software Architecture (SWA)

State of the Art

- Projects and platforms
  - Open Platform for NFV (OPNFV) [24]
    - open source project founded and hosted by the Linux Foundation
  - OpenMANO [25]
    - an open source project led by Telefonica, which is aimed at implementing ETSI's NFV MANO framework
  - CPLANE NETWORKS Dynamic Virtual Networks
    - A service orchestration platform
  - OpenStack
    - Open source cloud and NFV platform
  - OpenDaylight
    - Open source project hosted by The Linux Foundation.
    - Support a solid foundation for Network Functions Virtualization
  - OpenSDNCore
    - Testbed for SDN/NFV environments developed by Fraunhofer FOKUS

Use Cases

ETSI GS NGV 001 [27] specifies 9 use cases of NFV

- Network Function Virtualization Infrastructure as a Service
  - NFVI as the infrastructure to provide environment for VNFs to be executed
  - An approach to support cloud computing applications and VNF instances from different administrative domains
- Virtual Network Function as a Service (VNFaaS)
  - Virtualization of the CPE functions in the enterprise network
    - Routing, VPN termination, QoS support etc.
  - Virtualization of the Provider Edge router functions at the edge of the core network
    - Provisioned Virtual Private Network Services (PVPNS), IP VPN, VPLS etc.
- Virtual Network Platform as a Service (VNPaaS)
  - Similar to VNFaaS but with larger scale of service, programmability and scope of control
    - Provide virtual networks rather than single network functions
    - Allow the user of service to introduce their own VNF instances
- VNF Forwarding Graphs (Service chaining)
  - Define the VNFs involved in a service, relationship and the interconnection graph between these VNFs (will further be discussed in Network Service Orchestration section)

Use Cases

- Virtualization of Mobile Core Network and IMS
  - Aiming at applying virtualization to Evolved Packet Core (EPC) (e.g., MME, S/P-GW, PCRF etc.) and IP Multimedia Subsystem (IMS) (e.g., P/S/I-CSCF, MGCF, AS etc.)
- Virtualization of Mobile Base Station
  - Aiming at virtualizing and deploying RAN nodes onto standard IT servers, storages and switches to leverage more efficient resource utilization
- Virtualization of the Home Environment
  - To virtualize home CPE, including residential gateways for Internet and VoIP services, set-top-box
- Virtualization of CDNs
  - To virtualize all components of CDN, including cache nodes
- Fixed Access Network Function Virtualization
  - To virtualize access technologies such as DSL/FTTx towards reducing hardware complexity and energy consumption

Use cases of SDN/NFV in the IoT

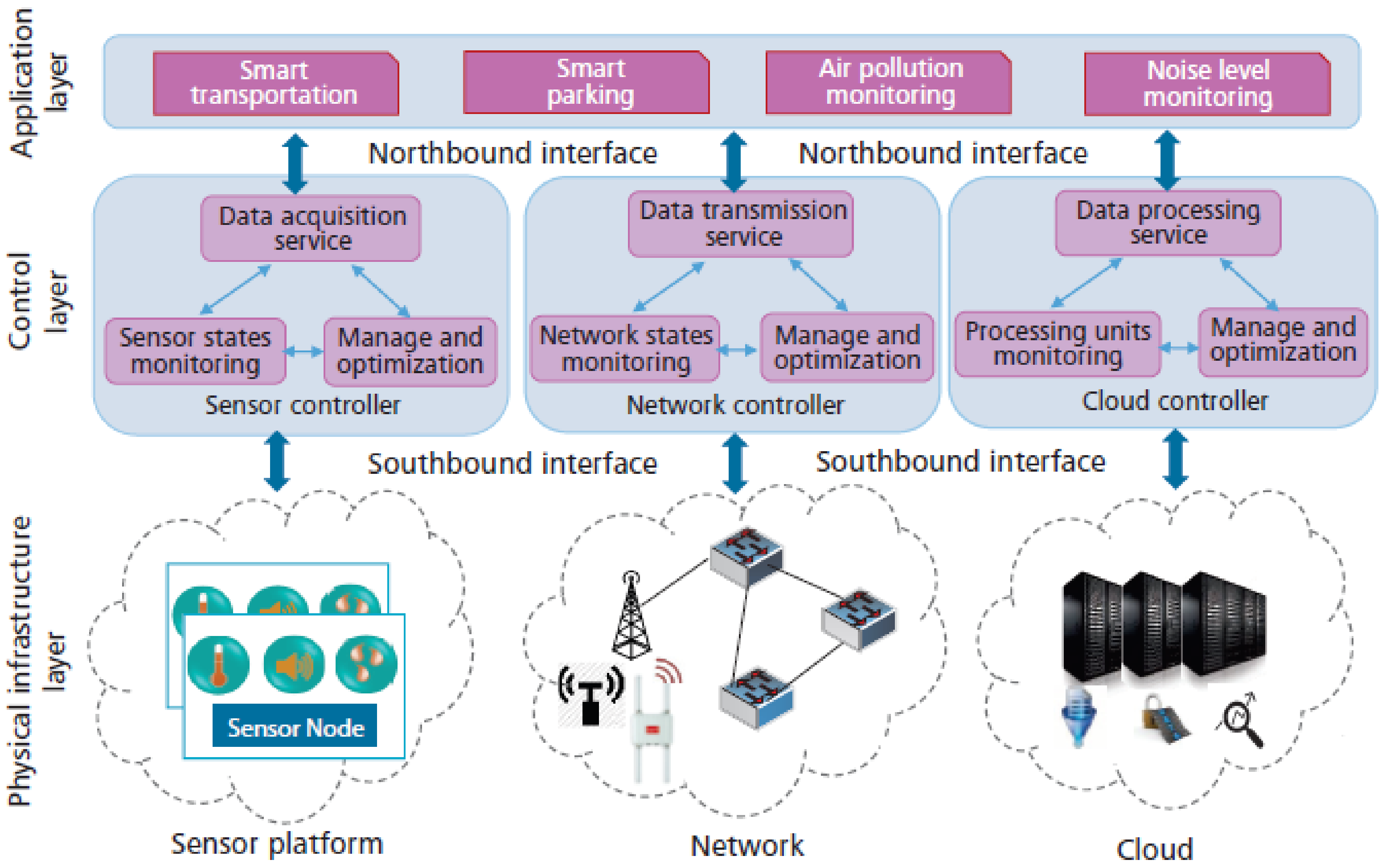
- Software-Defined Internet of Things [29]
- Convergence of NFV and SDN in Edge Cloud [30]

Software-Defined Internet of Things (SD-IoT) [29]

❑ IoT: State of the art and trends and solutions

	State of the art	Problems	The trends
Data acquisition	Application-oriented wireless sensor platforms. The control functions are preset in the firmware	Hard to customize in run time. Hard to implement dynamic optimization. High capital and maintenance cost.	Over the air programming to update sensor firmware.
Data transmission	Distributed protocols, such as WiFi, ZigBee, TCP/IP. The control protocols embed in each forwarding device.	Hard to control and evolve. No QoS guarantee.	Software-defined network. Network as a service with QoS guarantee.
Data processing	Each application developing data processing pipelines from the scratch.	The time cycle to develop a new application is long. Hard to share data processing resources.	Cloud based data processing to provide various data processing software, platform, and tool.

SD-IoT Architecture



Architecture of software-defined IoT. [29]

SD-IoT Architecture (*cont*…)

- SD-IoT architecture overview
  - Physical infrastructure layer
    - composed of various kinds of physical devices including sensor platforms, gateways, base stations, switches/routers, and servers.
    - Leave the decision-making to the control layer by interacting with it through southbound interfaces
  - Control layer
    - Manage physical devices the physical devices with various characteristics and functions through different southbound interfaces.
    - Provide services to the application layer through northbound interfaces.
  - Application layer
    - Developers can customize data acquisition, transmission and processing without worrying about the required change of configurations in physical devices.

SD-IoT System Design

- Sensor platform and data acquisition service
  - Provide APIs for applications to specify data requirements
  - The sensor controller base on the global view to dynamically active/deactive sensors and customize configurations to satisfy application requirements and reduce energy consumption.
- Network and data transmission service
  - Applications requirements: destination and QoS parameters
  - Specify destination: IP address
  - QoS: basic transmission, latency sensitive transmission, bandwidth guarantee transmission
- Cloud data center and data processing service
  - The could controller maps the application’s resource request to underlying server pools

SD-IoT System Design (*cont...*)

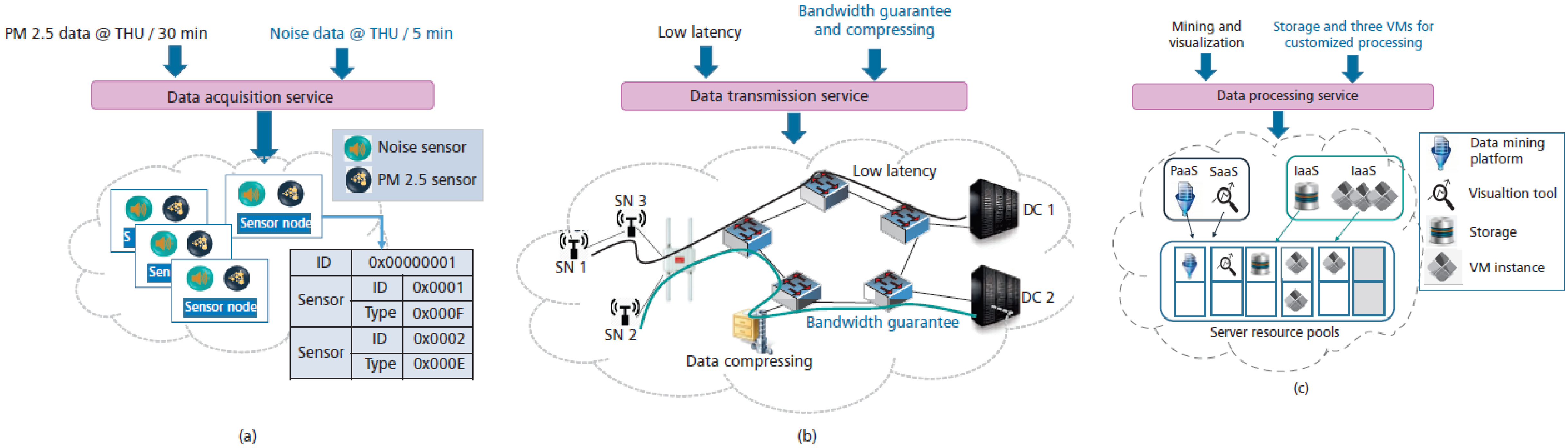


Illustration of data acquisition, transmission and processing services:a) data acquisition service; b) data transmission service; and c) data processing service [29]

HomeCloud: Convergence of NFV and SDN in Edge Cloud [30]

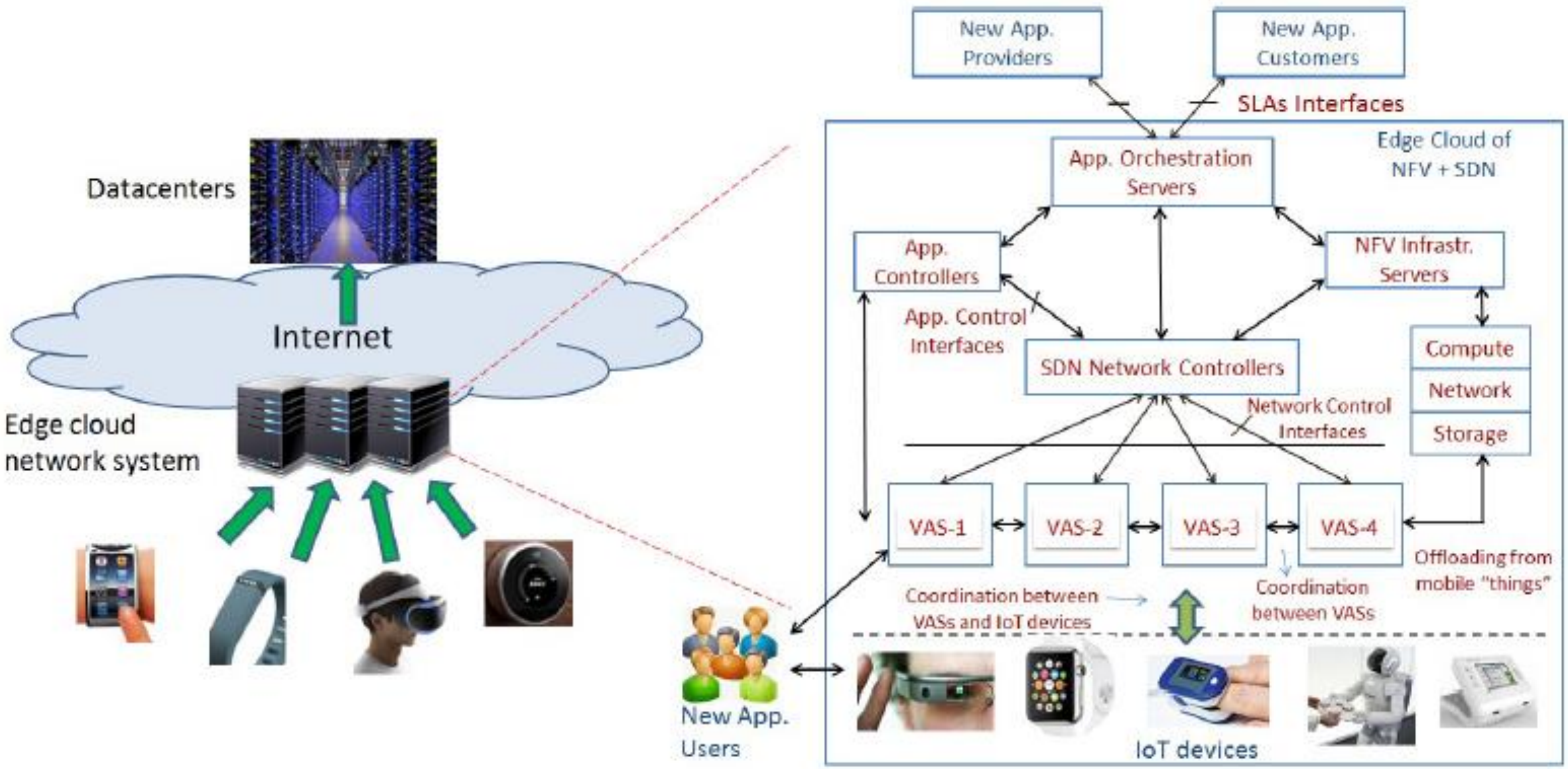
- Current cloud computing model: good and bad
  - Good
    - Provide an on-demand pay-as-you-go service to the users which lowers the owning cost for general customers
    - Provide elasticity of computing, storage, and networking resources which is flexible and scalable
    - Facilitate big-data analytics
  - Challenges in the IoT world / bad
    - Volume and velocity of data accumulation of IoT devices
    - Latency due to the distance between edge IoT devices and datacenters
    - Monopoly vs. open IoT competition

Why Edge Cloud?

- Address challenges of conventional cloud computing in the IoT world

Characteristics	Conventional cloud computing	Edge cloud and edge computing
Major applications	Most of the current mainstream cloud-involved applications	Applications on IoT, VR, AR, smart homes, smart cities, smart energy, smart vehicles.
Availability	A small number of large-sized datacenters	A large number of small-sized datacenters
Proximity of services and resources; Data processing location	Usually in remote datacenters and far from users	At the edge close to the users
End-to-end latency	High, due to the distance between the edge and remote datacenters	Low, due to proximity of the users
Backbone network bandwidth consumption	High, since huge data need to be transferred to the datacenters first	Low, since data are locally processed and stored in edge cloud
Scalability	Scalable at the center	Scalable both center and edge
Security (e.g., attacks on data enroute)	Data subject to attack due to long-distance transmission; Physical security depends on large facilities	Lower risk for enroute attacks; Physical security varies and different mechanisms needed

Convergence of NFV and SDN in Edge Cloud



Homecloud architecture for IoT application delivery [30]

VIỆN ĐIỆN TỬ - VIỄN THÔNG
School of Electronics and Telecommunications

Edge Cloud System Architecture

- NFV
  - create VNFs replacing traditional and specialized equipment from proprietary vendors
- SDN
  - configure, control, and manage VNFs created by NFV
- Automated orchestration
  - optimize the resource allocation and provision for the IoT applications under deployment
- Dynamic offloading
  - Process/VMs migration: enable moving the entire OS and its running applications in a mobile environment.
  - Program partitioning: decide which parts can be run on mobile devices and which parts can be run on edge servers.

IoT Applications Benefiting from Edge Cloud

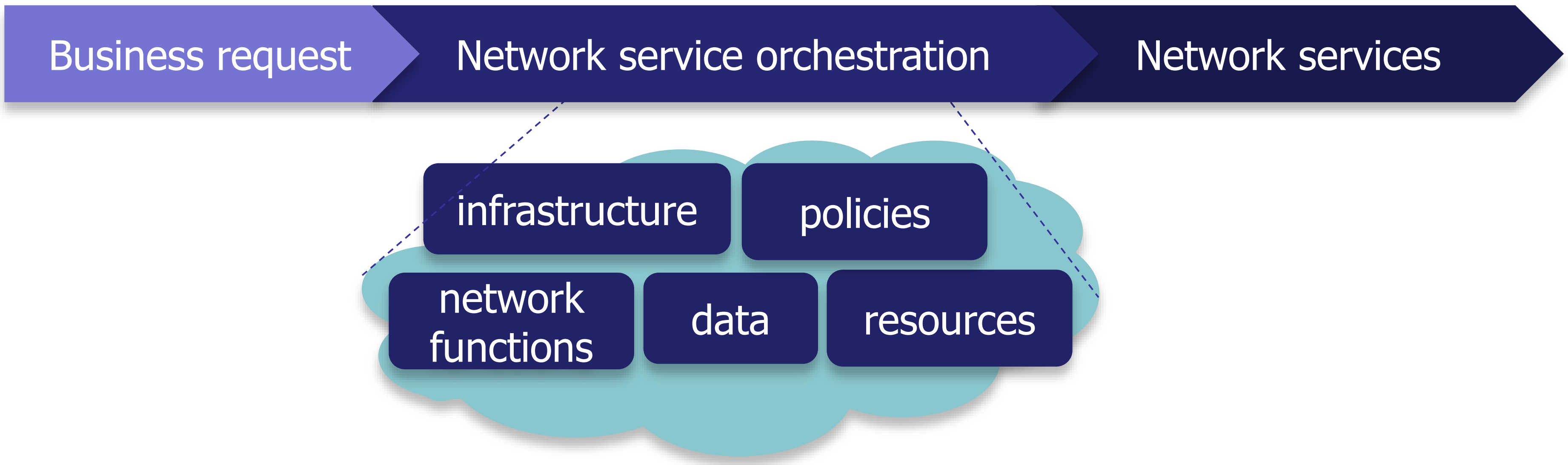
- Applications require low latency
  - Wearable came applications, industrial monitoring, and controlling applications require response time to be as low as 10 to 50 miliseconds
- Applications require high data bandwidth
  - Most of data are processed by the edge cloud first to reduce the volume of data sent to the remote datacenters, e.g., augmented reality (AR) and virtual reality (VR) applications.
- Applications involving large amount of IoT devices with limited capacities
  - Most of computation and data processing tasks can be offloaded to the edge cloud to save energy on IoT devices and get the tasks done faster in the edge cloud

Network Service Orchestration

- How does SDN/NFV change service provisioning?
  - Old way
    - Services are provisioned in a “static way” based on fixed network functions
    - Inflexible, difficult to customized
  - New way
    - Services are provisioned dynamically, flexibly based on network service orchestration

Network Service Orchestration

- What is Network Service Orchestration?
  - Network service orchestration is the process of automated arrangement, coordination and management of complex network functions and services.
  - The arrangement of various network functions to create a new service is called service chaining

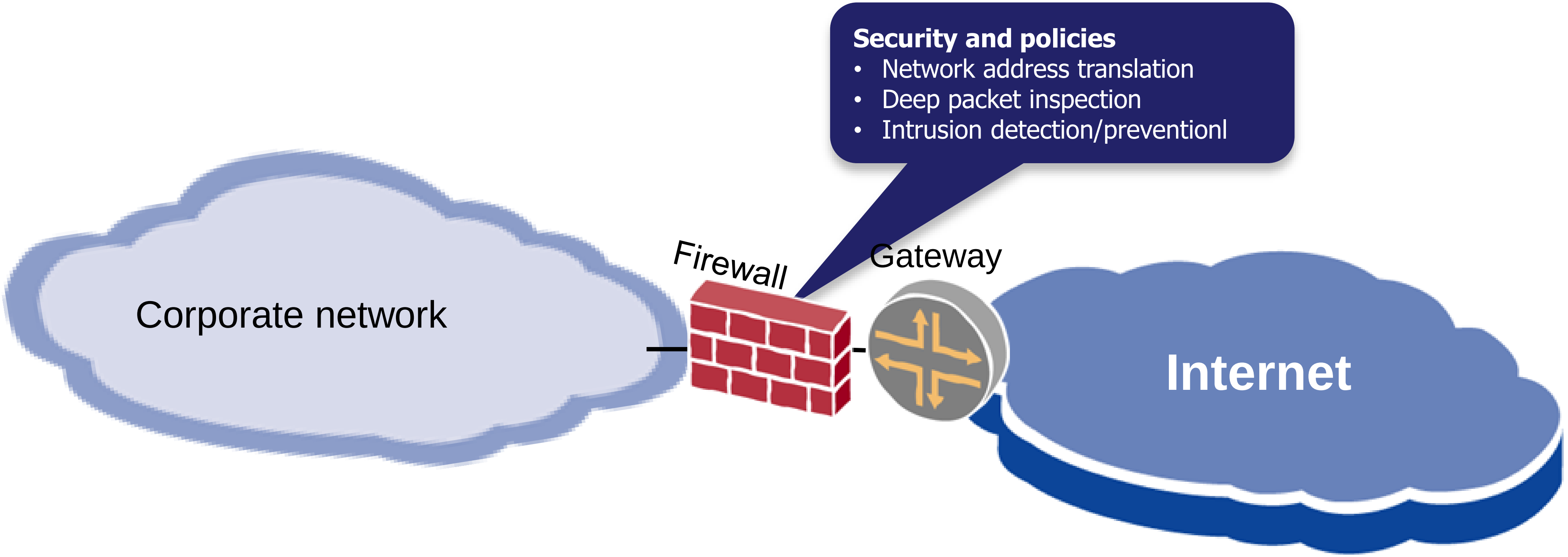


Network Service Orchestration

- Advantages of Network Service Orchestration
  - Reduce human errors by automated processes
  - Scalable to large, complex applications and services
  - Centralized management of resources, policies
    - Reduce the time and efforts to deploy multiple instances of a single application
  - Flexible, dynamic, easy to create and customize new network services

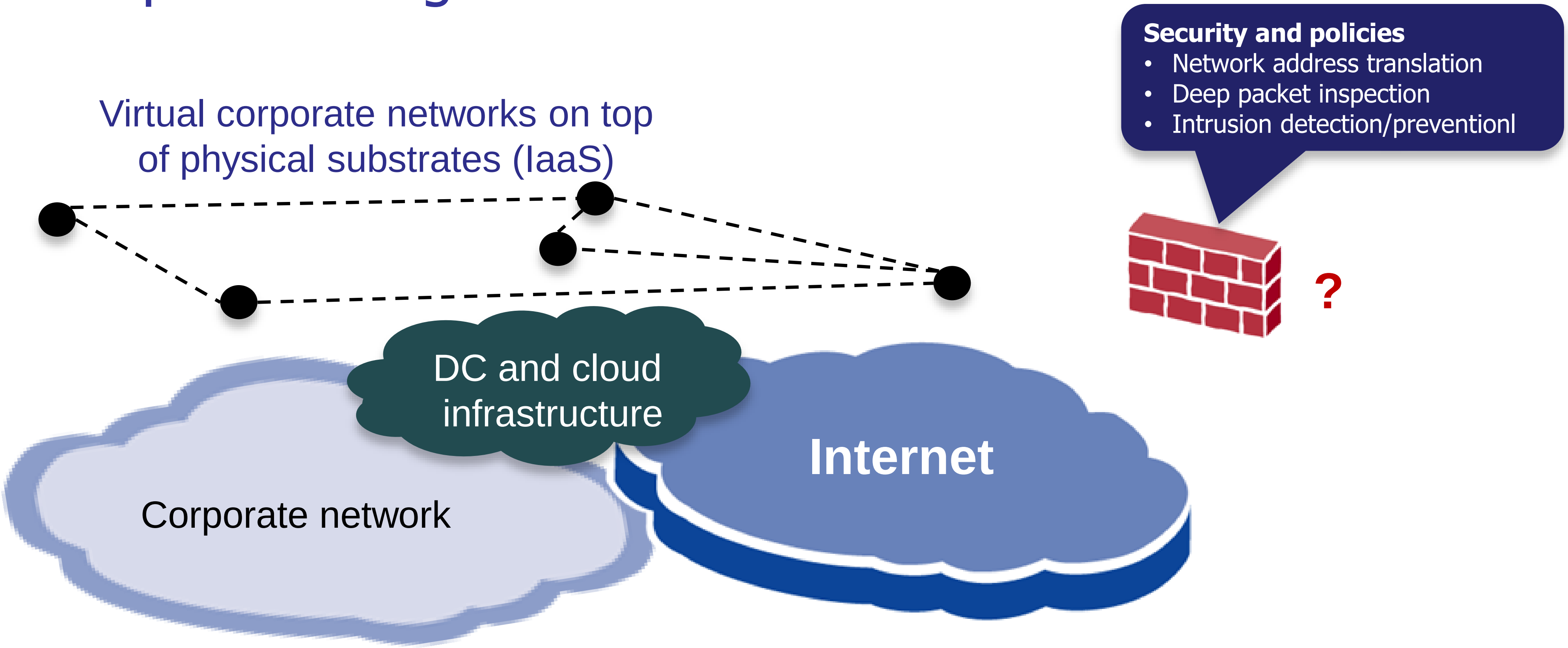
Network Service Orchestration

- How does Network Service Orchestration work?
  - Traditional way of service provisioning



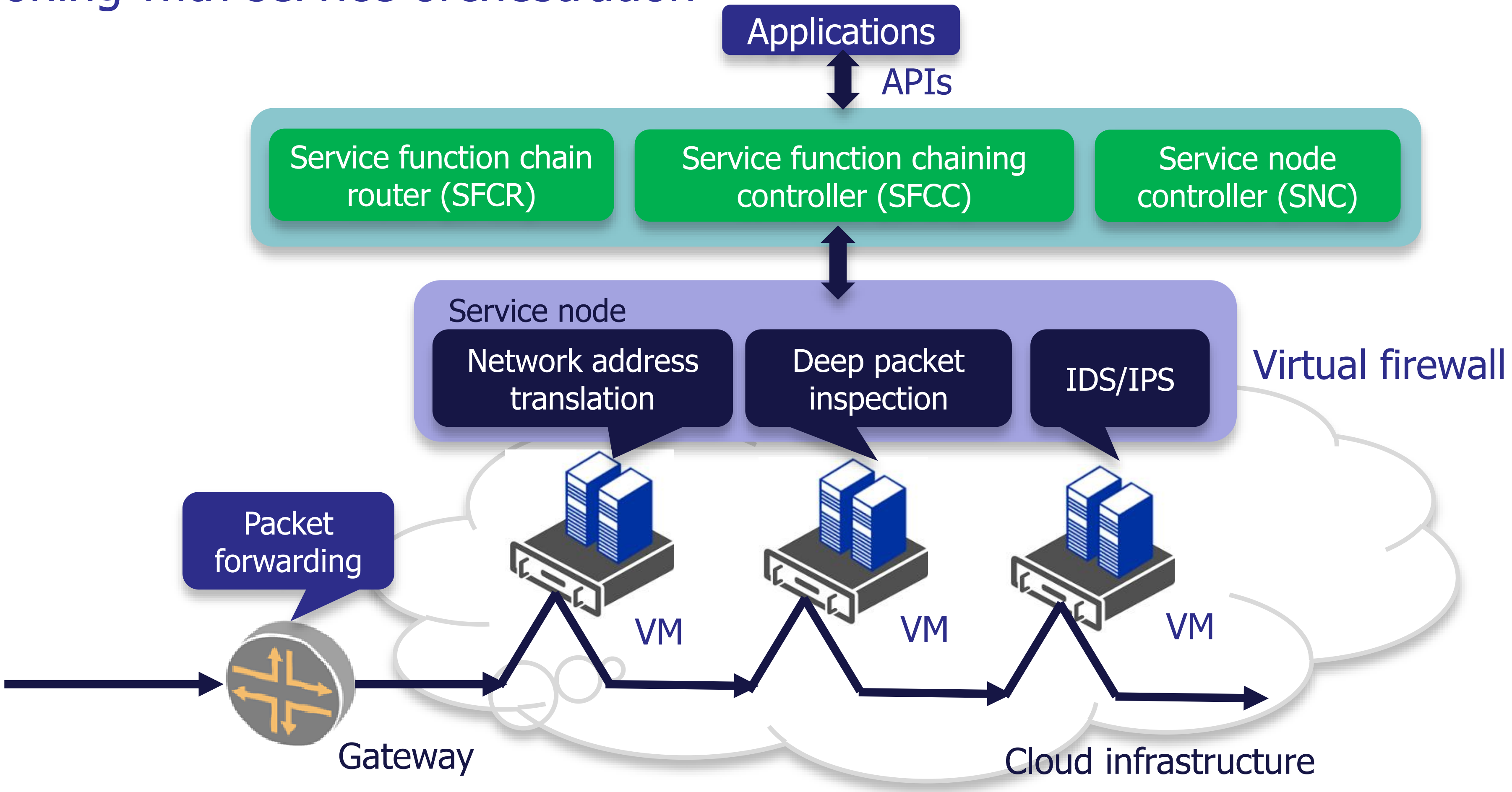
Network Service Orchestration

- How does Network Service Orchestration work?
  - Service provisioning with service orchestration



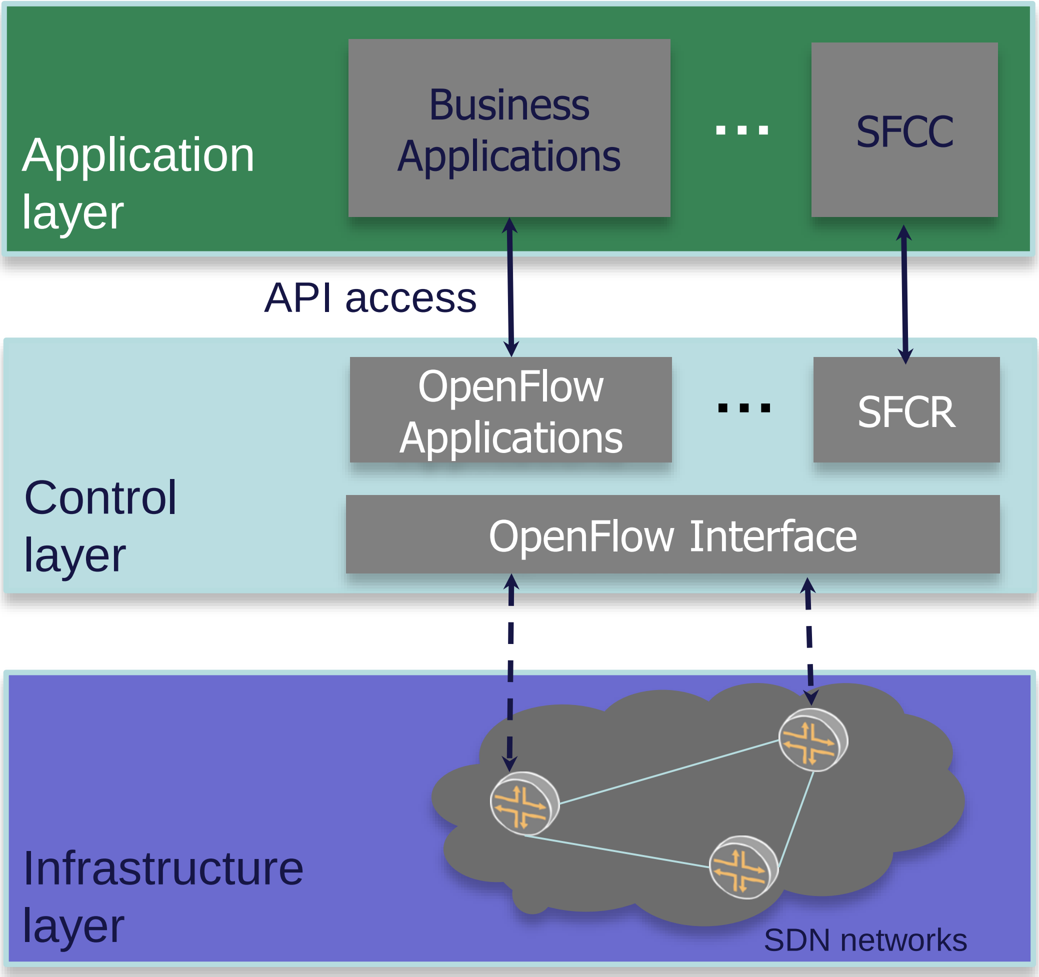
Network Service Orchestration

- How does Network Service Orchestration work?
  - Service provisioning with service orchestration



Network Service Orchestration

- NFV Service Chaining Architecture [26]
  - Service Function Chaining Controller (SFCC):
    - Controlling all system components
    - Creating/removing service chains
    - Handling user (de-)registration, service instance allocation
    - High-level API to the service chaining system
  - Service Function Chaining Router (SFCR):
    - Routing packets to components in the service chain
    - Installing OpenFlow rules
  - Service Node Controller (SNC):
    - Managing service nodes and service instances
    - Creation, usage, configuration, and destruction of service instances



Some Issues

- Performance
  - Does NFV perform as well as conventional model? How to provision/optimize resources and functions to meet service requirements?
    - Latency, delay
    - QoS
    - Complexity
- Security and resilience
  - How to protect and isolate resources and functions of different subscribers?
  - How to make Virtual Network Functions more reliable and resilient?
- Heterogeneity
  - How to enable third-party network functions, hardware, software as well as network resources from different vendors?
- Scalability
  - NFV solutions should scale well and be able to support large number of users

Benefits of NFV

- Reduced equipment costs (CAPEX)
  - through consolidating equipment and economies of scale of IT industry
- Increased speed of time to market
  - by minimising the typical network operator cycle of innovation
- Availability of network appliance multi-version and multi-tenancy
  - allows a single platform for different applications, users and tenants
- Enables a variety of eco-systems and encourages openness
- Encouraging innovation to bring new services and generate new revenue streams

Benefits of NFV

- Flexibility to easily, rapidly, dynamically provision and instantiate new services in various locations
- Improved operational efficiency
  - by taking advantage of the higher uniformity of the physical network platform and its homogeneity to other support platforms
- Software-oriented innovation to rapidly prototype and test new services and generate new revenue streams
- More service differentiation & customization
- Reduced (OPEX) operational costs: reduced power, reduced space, improved network monitoring
- IT-oriented skillset and talent

Conclusions

- So, why do we need/want NFV(/SDN)?
  - Virtualization: Use network resource without worrying about where it is physically located, how much it is, how it is organized, etc.
  - Orchestration: Manage thousands of devices
  - Programmable: Should be able to change behavior on the fly.
  - Dynamic Scaling: Should be able to change size, quantity
  - Automation
  - Visibility: Monitor resources, connectivity
  - Performance: Optimize network device utilization
  - Multi-tenancy
  - Service Integration
  - Openness: Full choice of modular plug-ins

References

[1] Stefan Saroiu, Krishna P. Gummadi, and Steven D. Gribble. 2003. Measuring and analyzing the characteristics of Napster and Gnutella hosts. *Multimedia Syst.* 9, 2 (August 2003), 170-184

[2] A. Fischer, J. F. Botero, M. T. Beck, H. de Meer and X. Hesselbach, "Virtual Network Embedding: A Survey," in *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 1888-1906, Fourth Quarter 2013

[3] J.E. van der Merwe, I.M. Leslie, Switchlets and dynamic virtual ATM networks, in: *Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management* (IM'97), 1997, pp. 355–368.

[4] R. Boutaba, W. Golab, Y. Iraqi, B. St-Arnaud, Grid-controlled lightpaths for high performance grid applications, *Journal of Grid Computing* 1 (4) (2003) 387–394

[5] J. Touch, S. Hotz, The X-Bone, in: *Proceedings of the Third Global Internet Mini-Conference at GLOBECOM'98*, 1998, pp. 44–52.

[6] J.D. Touch, Dynamic internet overlay deployment and management using X-Bone, *Computer Networks* 36 (2-3) (2001) 117–135

[7] A. Sundararaj, P. Dinda, Towards virtual networks for virtual machine grid computing, in: *Proceedings of the Third USENIX Virtual Machine Research and Technology Symposium* (VM'04), 2004, pp. 177–190

[8] M. Boucadair, P. Levis, D. Griffin, N. Wang, M. Howarth, G. Pavlou, E. Mykoniati, P. Georgatsos, B. Quoitin, J.R. Sanchez, M. Garcia-Osma, A framework for end-to-end service differentiation: network planes and parallel Internets, *IEEE Communications* 45 (9) (2007) 134–143.

References

[9] N. Wang, D. Griffin, J. Spencer, J. Griem, J.R. Sanchez, M. Boucadair, E. Mykoniati, B. Quoitin, M. Howarth, G. Pavlou, A.J. Elizondo, M.L.G. Osma, P. Georgatsos, A framework for lightweight QoS provisioning: network planes and parallel Internets, in: *Proceedings of the 10th IFIP/IEEE International Symposium on Integrated Network Management* (IM'07), 2007, pp. 797–800

[10] L. Peterson, T. Anderson, D. Culler, T. Roscoe, A blueprint for introducing disruptive technology into the Internet, *SIGCOMM Computer Communication Review* 33 (1) (2003) 59–64

[11] G.P. Group, GENIdesign principles, Computer 39 (9) (2006) 102– 105

[12] A. Blenk, A. Basta, M. Reisslein and W. Kellerer, "Survey on Network Virtualization Hypervisors for Software Defined Networking," in *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 655-685, Firstquarter 2016

[13] R. Sherwood, G. Gibb, K.-K. Yap, G. Appenzeller, M. Casado, N. Mckeown, and G. Parulkar, "FlowVisor: A network virtualization layer," *OpenFlow Consortium*, Tech. Rep., 2009

[14] E. Salvadori, R. D. Corin, A. Broglio, and M. Gerola, "Generalizing virtual network topologies in OpenFlow-based networks," in *Proc. IEEE Globecom*, Dec. 2011, pp. 1–6

[15] R. D. Corin, M. Gerola, R. Riggio, F. De Pellegrini, and E. Salvadori, "VeRTIGO: Network virtualization and beyond," in Proc. Eu. Workshop on Software Defined Networks (EWSDN), 2012, pp. 24–29

[16] A. Al-Shabibi, M. De Leenheer, M. Gerola, A. Koshibe, W. Snow, and G. Parulkar, "OpenVirteX: a network hypervisor," in Proc. Open Networking Summit (ONS), Santa Clara, CA, Mar. 2014

References

[17] Nguyen Huu Thanh, Vu Anh Vu, Nguyen Duc Lam, Nguyen Van Huynh, Tran Manh Nam, Ngo Quynh Thu, Thu-Huong Truong, Nguyen Tai Hung, Thomas Magedanz, "A Generalized Resource Allocation Framework in Support of Multi-Layer Virtual Network Embedding based on SDN", *Journal of Computer Networks*, Volume 92, Part 2, 9th December 2015, Pages 251 – 269

[18] D. Drutskoy, E. Keller, and J. Rexford, "Scalable network virtualization in software-defined networks," *IEEE Internet Computing*, vol. 17, no. 2, pp. 20–27, 2013

[19] A. Al-Shabibi, M. D. Leenheer, and B. Snow, "OpenVirteX: Make your virtual SDNs programmable," in *Proc. HotSDN*, 2014, pp. 25–30

[20] L. Liao, V. C. M. Leung, and P. Nasiopoulos, "DFVisor: Scalable network virtualization for QoS management in cloud computing," in *Proc. CNSM and Workshop* IFIP, 2014, pp. 328–331

[21] L. Liu, R. Mu˜noz, R. Casellas, T. Tsuritani, R. Mart´ınez, and I. Morita, "OpenSlice: an OpenFlow-based control plane for spectrum sliced elastic optical path networks," *OSA Optics Express*, vol. 21, no. 4, pp. 4194–4204, 2013

[22] Rashid Mijumbi, Joan Serrat, Juan Luis Gorricho, Niels Bouten, Filip De Turck, Raouf Boutaba, "Network Function Virtualization: State-of-the-Art and Research Challenges", *IEEE Communications Surveys and Tutorials*. September 2015

[23] "The Internet Engineering Task Force (IETF) Service Function Chaining (SFC) Working Group (WG). Documents," https://datatracker.ietf. org/wg/sfc/documents/

References

[24] "Open Platform for NFV (OPNFV)," https://www.opnfv.org/about, 2015

[25] D. R. Lopez, "OpenMANO: The Dataplane Ready Open Source NFV MANO Stack," in IETF Meeting Proceedings, Dallas, Texas, USA, March 2015

[26] Jeremias Blendin et al., "*Software-Defined Network Service Chaining*", 2014 Third European Workshop on Software-Defined Networks

[27] ETSI GS NFV 001, Network Function Virtualization – Uses Cases, October 2013

[28] ETSI GS NFV 002, "Architectural Framework," Oct 2013

[29] J. Liu, Y. Li, M. Chen, W. Dong and D. Jin, "Software-defined internet of things for smart urban sensing," in *IEEE Communications Magazine*, vol. 53, no. 9, pp. 55-63, September 2015.

[30] J. Pan and J. McElhannon, "Future Edge Cloud and Edge Computing for Internet of Things Applications," in *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 439-449, Feb. 2018.