Software Defined Networking and Network Function Virtualization

Nicolai Leymann

Deutsche Telekom Technik

25.01.2016

ERLEBEN, WAS VERBINDET.

Table of Contents (1 of 2)

1. Part 1: Towards an SDN based and virtualized Network Architecture
   1. Introduction
   2. Definitions and Key Aspects
      1. Overview Data Center Virtualization
      2. Overview Software Defined Networking
      3. Overview Network Function Virtualization
   3. Architecture Blueprint
   4. Summary

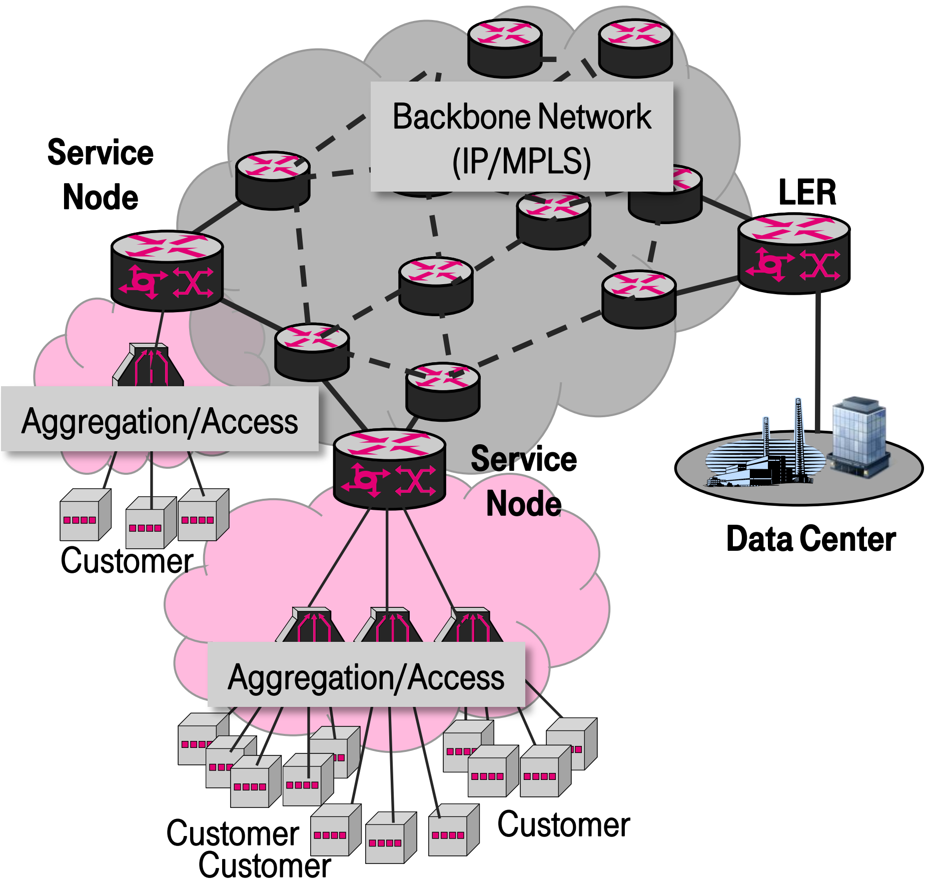
Table of Contents (2 of 2)

1. Part 2: Applying SDN and NFV
   1. Use Cases
   2. Service Chaining
      1. GI-LAN
      2. Traffic Steering
      3. Implementations: Network Service Header
   3. Standardization
   4. Examples for SDN Solutions
      1. OpenDaylight
      2. OpenContrail

Part 1:

Towards an SDN based and virtualized Network Architecture

Software Centric Networking.
Typical Service Provider Network.



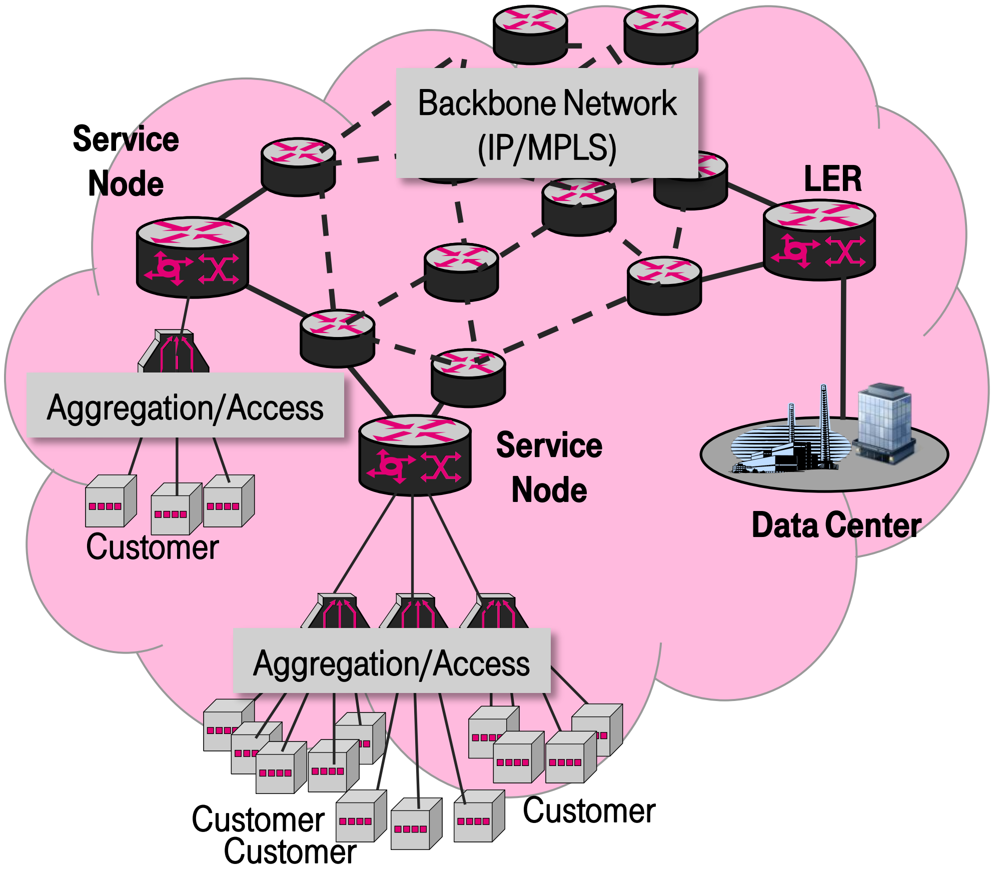
Typical Network Design

- Aggregation/Access Network: Aggregates business and residential Customers (VDSL, GPON, ...)
- Service Node: Terminates customer session and provides basic services (e.g. QoS, Multicast Replication, Bandwidth Control, VPN, P2P, P2MP, ...)
- Backbone Network: Provides connectivity among customers, towards service areas and the internet
- Data Centers: Providing additional services (e.g. IPTV services, web, customer self service, ...)

Software Centric Networking.
Todays (Networking) Challenges.

- Typical network designs are using an access/aggregation network and a backbone network which also interconnects to service areas/data centers.
    - typically, operation and provisioning of dc and network are separated.
- In many cases introducing a new services leads to software updates on networking nodes (service and network are closely coupled).
    - Monolithic Software/Hardware
- Innovation to slow (no ability to “program” or closely interact with the network).
- Modification of forwarding mainly through routing protocols (no external interfaces in order to interact directly with forwarding hardware).
- Low level of automation and lot of manual configuration

Software Centric Networking.
Vision.

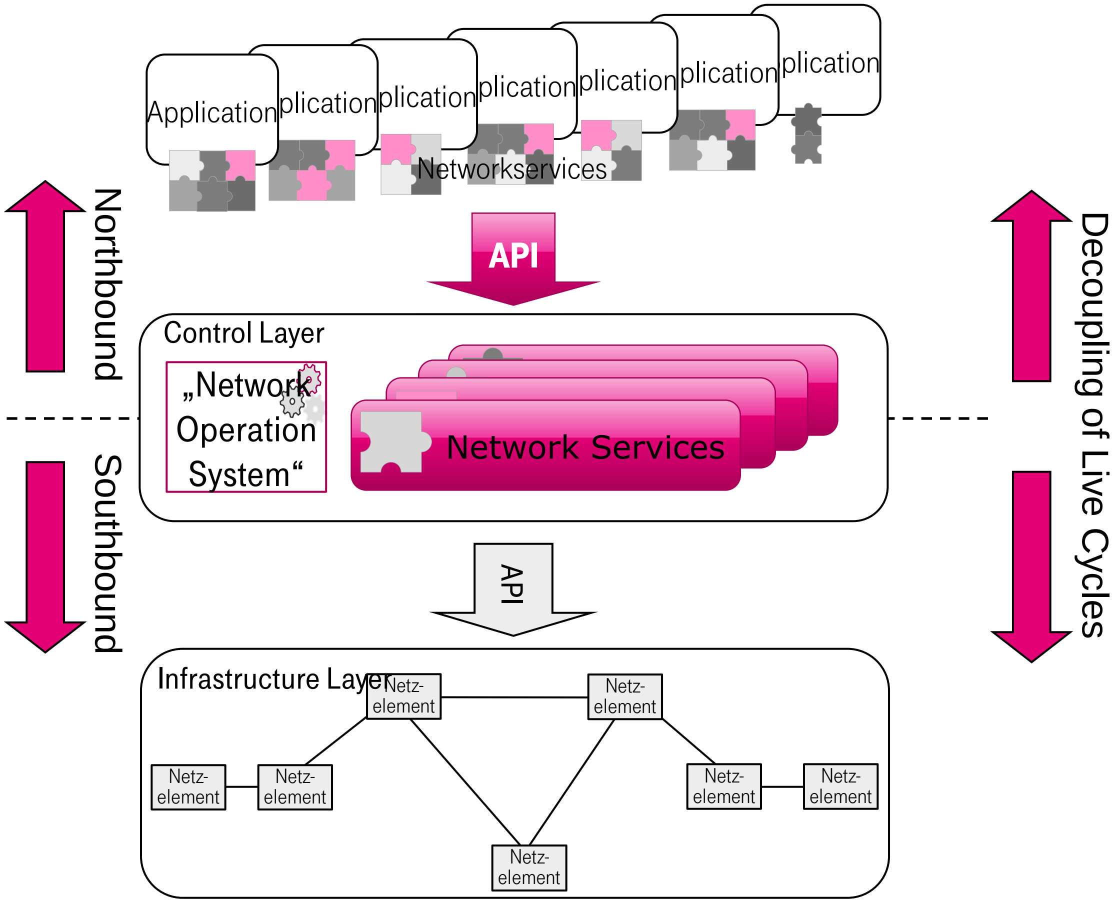


Typical Network Design

- Integrated architecture which takes network and data center into account.
- Service can span from network into data center
- Flexible placement of service functions (driven by requirements).
- Network is programmable (a new service does not lead to a new software release on networking nodes)
- Complexity moved into the data centers (applications).
- High level of automation.

Software Centric Networking.

Moving towards a Programmable Network.



Key Components

- Application used to build service chains sitting on top of an Controller
- Controller offers a common API (or several APIs) northbound
- Controller uses southbound interface towards abstraction layer (e.g. openflow)
- Abstraction layer integrates different NEs and protocols
- Flexible chaining mechanism used to build network path between service nodes

Software Centric Networking.

Key Building Blocks of Upcoming Network Architectures.

"(Network Function) Virtualization in allows to run all applications (like web services, IPTV services, of network functions) on a standard x86 based server infrastructure using an additional abstraction layer. Virtualization allows elastic services, mobility and highly scalable services."

"Software Defined Networking allows network administrators to have programmable central control of network traffic without requiring physical access to the hardware devices. It decouples the control plane from the underlying forwarding (data) plane. It allows fine (external) control over the network."

Based on Wikipedia (01.08.2013)

"High Level of Automation allows a fast and flexible service provisioning. Decoupled life cycles allow the introduction of new services without installing/deploying new software images on networking components."

Software Centric Networking.
Eight Principles for Service Providers.

1. **One integrated strategy** for Software Defined Networking, Network Function Virtualization and Data Center deployments.
2. **Virtualization** of network functions on data center hardware should be the **standard** model when introducing new technologies.
3. **Implement open interfaces** to ensure programmability of the network fabric but don’t limit to a specific protocol or standard.
4. SDN enables different level of control of the network .
   Most attractive use cases for IP flow-level control (e.g. OpenFlow) are at the network edge, e.g. **service chaining.**
5. Combine the mechanisms of SDN and NFV in order to **simplify IT processes**.
6. Accelerate the efforts to **implement network aware orchestration** of services and resources for optimal placement.
7. Use **standardized and common data center technologies and processes** to implement network functions.
8. Take into account that SDN/NFV is still a moving target. Many players on the market, **almost anything runs (for marketing purposes) under the “SDN/NFV**” umbrella.

Strategy

Software Defined Networking

Network Function Virtualization

Data Center

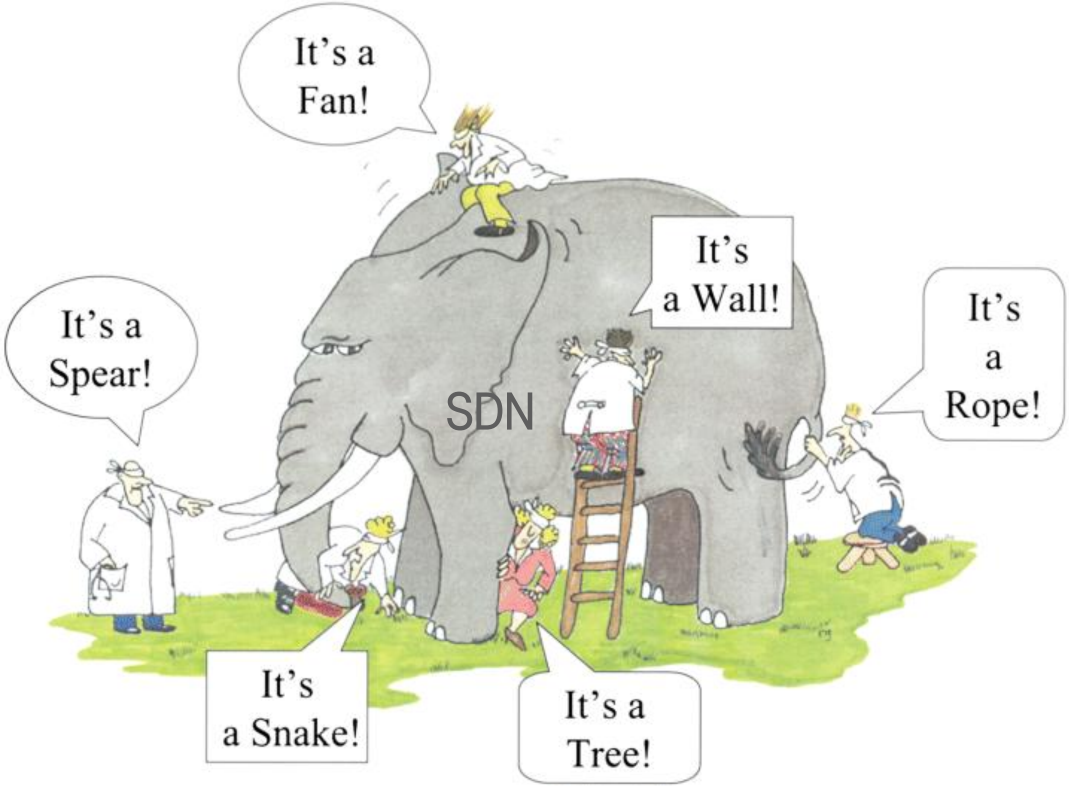
Software Centric Networking.

Are we still in the phase of the six blind men and the elephant?

Software Defined Networking

- Scalability Data Center Networks
- Service Provider Data Networks
- Streamlined and Dynamic, but Not a technology Nor a market
- Standardized platform for Development of Novel Services
- Still Don't Know
- Something to Do NOW!

Source: Tutorial on SDN Marketing Opportunities, Open Networking Summit 2013 (M. Cohn)

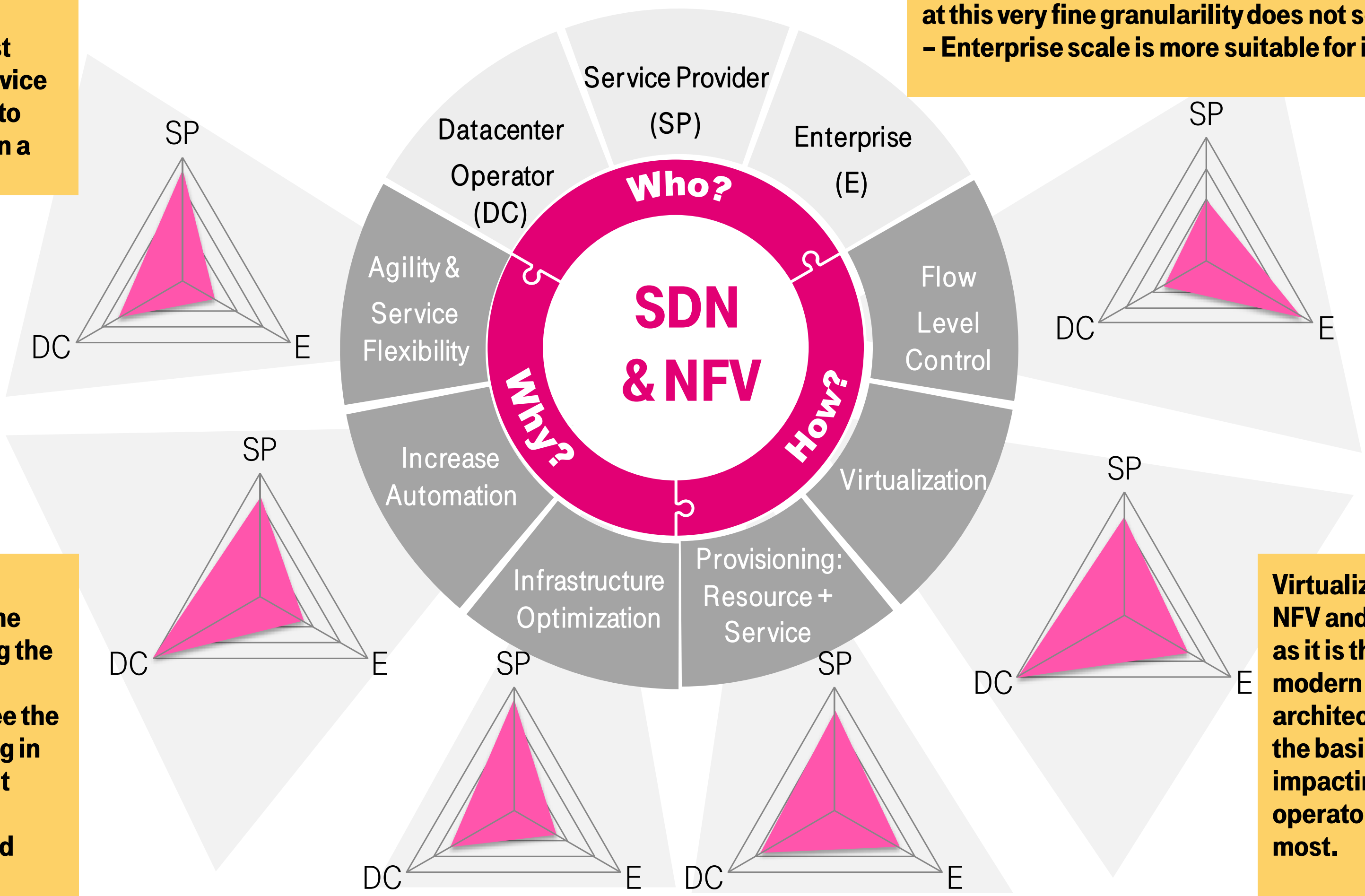


Software Centric Networking.

Impact Analyses.

Agility & Service Flexibility is most important to Service Providers, least to Enterprise, again a matter of scale

The Flow Level Control aspect of SDN is less important to Service Providers as controlling traffic at this very fine granularility does not scale very well – Enterprise scale is more suitable for it.



Data Center Operators are the ones demanding the automation the most, as they see the network as being in their way when it comes to fully automated cloud solutions.

Service Providers stand to gain the most from infrastructure optimization, as the relative percentage of infrastructure is highest

Provisioning is almost equally important to all players

Virtualization is key for NFV and SDN Strategy, as it is the basis for modern Datacenter architectures and also the basis for NFV, thus impacting Data Center operators and SP the most.

Software Centric Networking.
Key Aspects.

Software Defined Networking

- Decoupling of Control Plane and Data Plane
- Application are steering the traffic within the network and interacting with the forwarding nodes.
- Flexible Control of all network functionality (not focussed on forwarding).
- Moving forward towards a „NOS“ – Network Operating System.

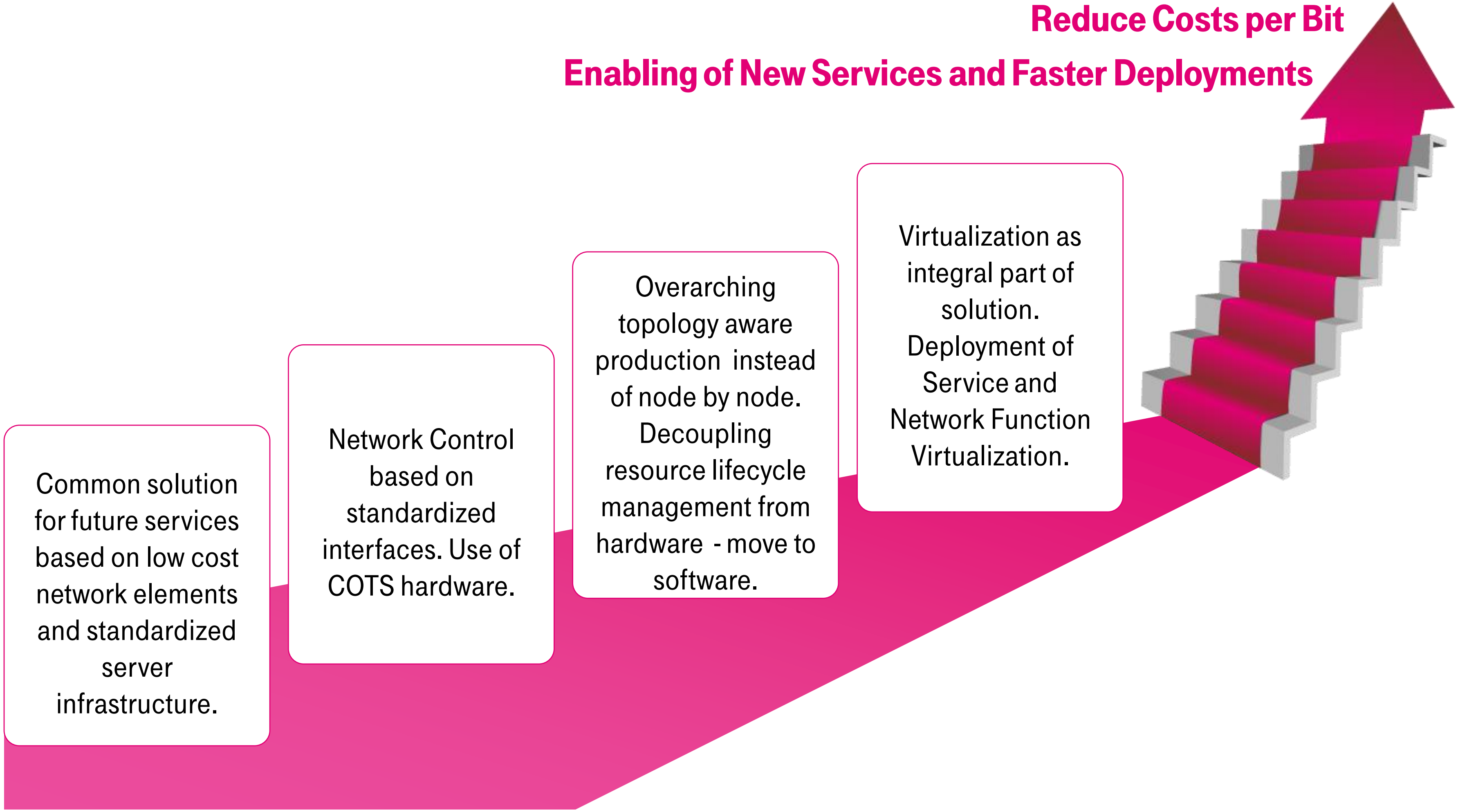
Virtualization (Data Center)

- Use of standard x86 based server and abstraction of physical infrastructure
- Use of a common resource pool
- Decoupling of hardware and application life cycles
- Simplification and automation of provisioning processes for Cloud services
- Opex & Capex savings through lower energy & maintenance costs

Network Function Virtualization

- Network Functions (like firewalling) are running on a virtualized infrastructure in a DC.
- Use of standard x86 based hardware (same data center for all services).
- Complex functionality for subset of services/customers not on every network node.
- Respond on changing requests via horizontal scaling

Software Centric Networking.
Challenges, Requirements and expectations.



Reduce Costs per Bit

Enabling of New Services and Faster Deployments

Common solution for future services based on low cost network elements and standardized server infrastructure.

Network Control based on standardized interfaces. Use of COTS hardware.

Overarching topology aware production instead of node by node. Decoupling resource lifecycle management from hardware - move to software.

Virtualization as integral part of solution. Deployment of Service and Network Function Virtualization.

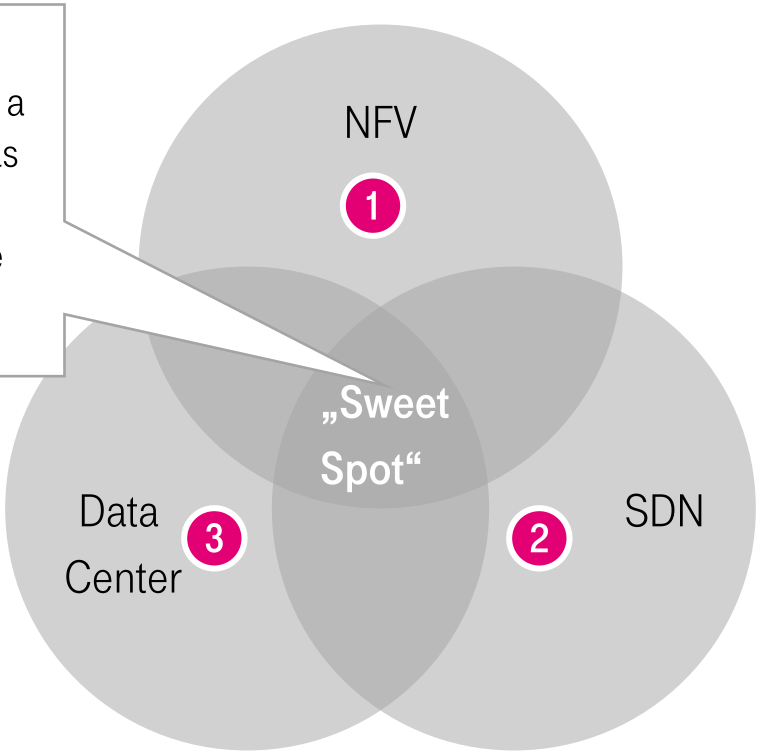
Architecture Blueprint

Towards an SDN based and virtualized Network Architecture

SDN and NFV

Architecture Blueprint: Bringing Everything Together.

End-to-End Use Case (orchestration of NFV services in a Data Center). NFV can be seen as a use case of combining virtualization in DC and Software Defined Networking.



NFV
1

- Virtualization of network functionaliy (e.g. PPP termination, Firewalling, Parental Control)

SDN
2

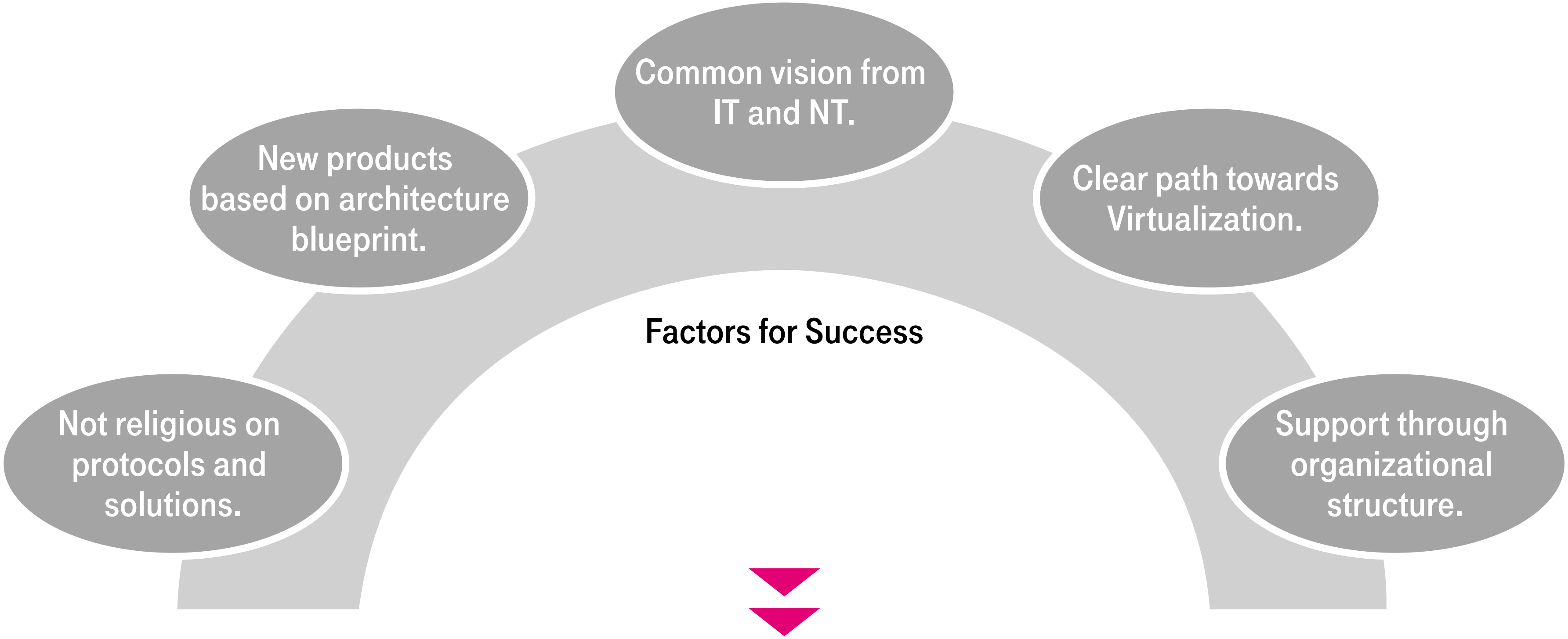
- Separated control /data planes with standardized south and northbound interfaces to control the nodes. Decoupling of life cycles.

Data Center
3

- Standard x86 based hardware with flexible and elastic resource provisioning (computing performance, throughput, ...)

SDN and NFV

Architecture Blueprint: Towards a Common Architecture.



Not religious on protocols and solutions.

New products based on architecture blueprint.

Common vision from IT and NT.

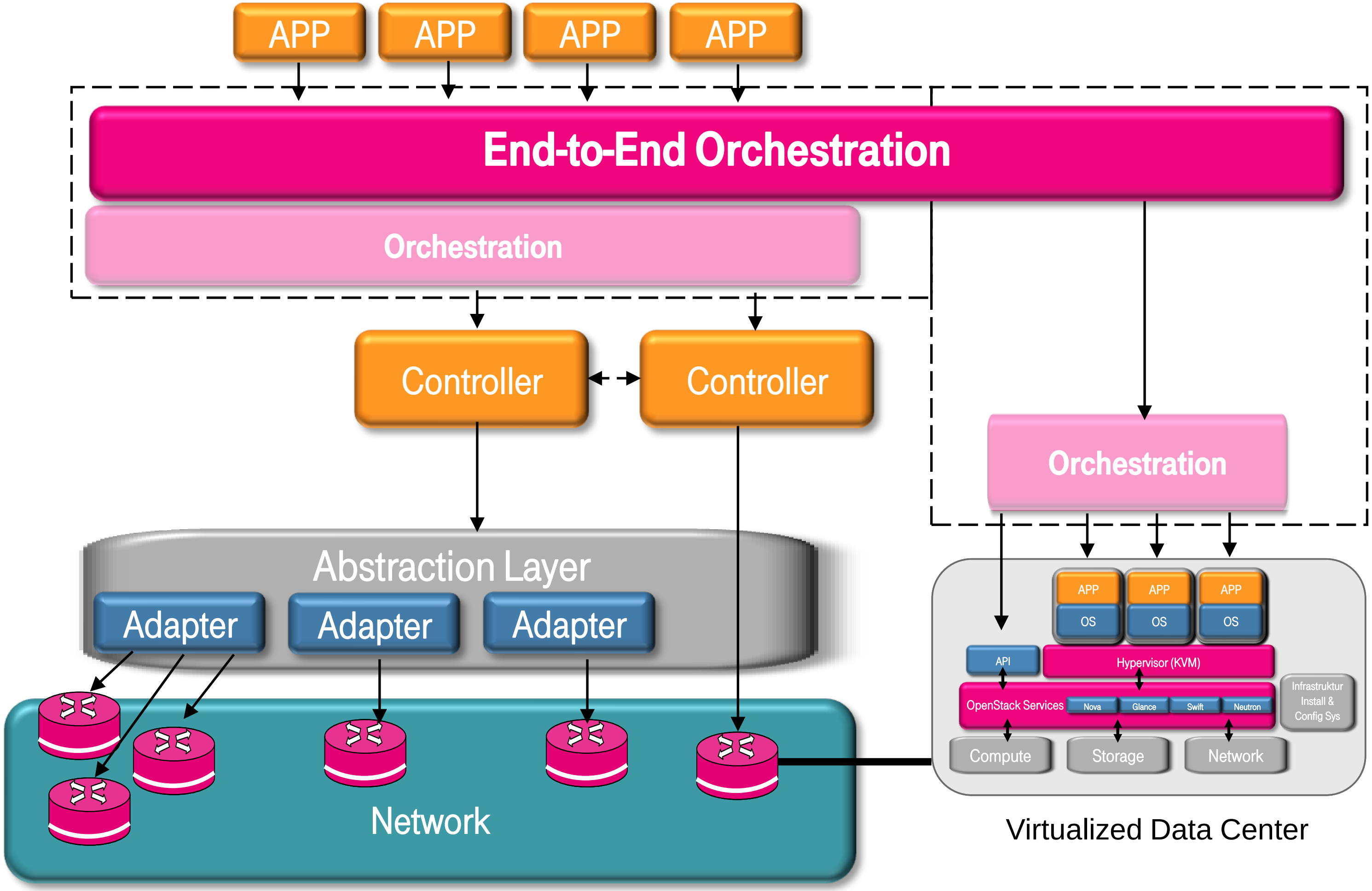
Clear path towards Virtualization.

Support through organizational structure.

Factors for Success

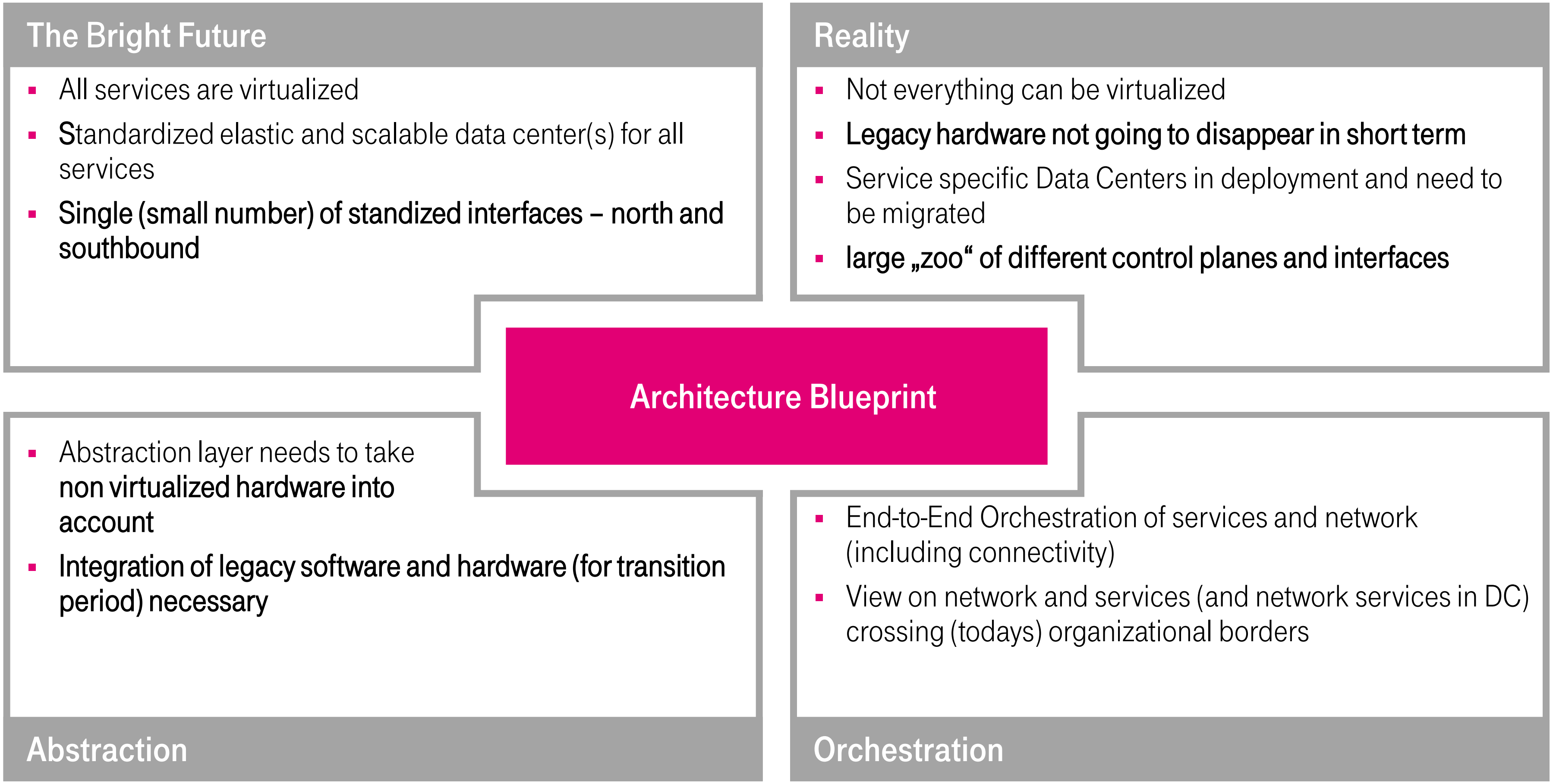
Open minded discussion and flexibility are key factors for success. Architecture and strategy should be seen as enablers for future networks and services.

SDN and NFV
Architecture Blueprint (1 of 2)



Virtualized Data Center

SDN and NFV
Architecture Blueprint (2 of 2)



The Bright Future

- All services are virtualized
- Standardized elastic and scalable data center(s) for all services
- **Single (small number) of standized interfaces – north and southbound**

Reality

- Not everything can be virtualized
- **Legacy hardware not going to disappear in short term**
- Service specific Data Centers in deployment and need to be migrated
- **large „zoo“ of different control planes and interfaces**

Architecture Blueprint

- Abstraction layer needs to take **non virtualized hardware into account**
- **Integration of legacy software and hardware (for transition period) necessary**

Abstraction

- End-to-End Orchestration of services and network (including connectivity)
- View on network and services (and network services in DC) crossing (todays) organizational borders

Orchestration

SDN and NFV
Timeline – Three Phase Approach.

Phase 1:

- Strategy Development
- End-to-End architecture and strategy development
- Definition of use cases based on existing products

Phase 2:

- Demonstrators based on use cases (SDN based traffic steering and virtualization).
- Evaluation of SDN evolution and integration into existing architecture.
- Start of migration of existing services (if applicable).

Phase 3:

- All new services deployed on virtualized platforms.
- Integration of SDN and Virtualization in DC starts.
- SDN/Virtualization integrated into existing architecture. New service only on new architecture (no legacy deployments).

SDN and NFV
Summary.

Evolution

- Technologies like SDN and NFV are enablers for new services and more flexibility
- Network evolution towards a software based/centric approach
- Paradigm shift within IT industry (vendors, telcos, ...)

Opportunities

- Higher flexibility for network and service deployments (faster „time to market“)
- Possibility to provide services end-to-end and ..
- Interact with the underlying network through a single orchestration

Challenges

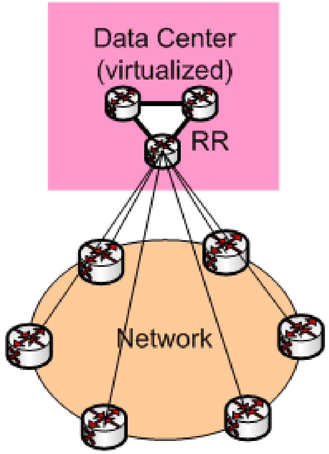
- Industry still in „gold rush mode“
- Many players, many interpretations of SDN
- Impact not only on technology level but also on organizational structures

Part 2:
Applying SDN and NFV

SDN and NFV

Applying SDN and Virtualization: BGP Route Reflectors.

Graphics



Use Case Description

- Today’s route reflectors are running on specialized hardware (typically routers with a lot of memory)
- Route reflectors are being virtualized and moved into a standard data center running on x86 hardware
- Scalability is achieved by adding new virtual machines running the RRs

Benefits / Advantages

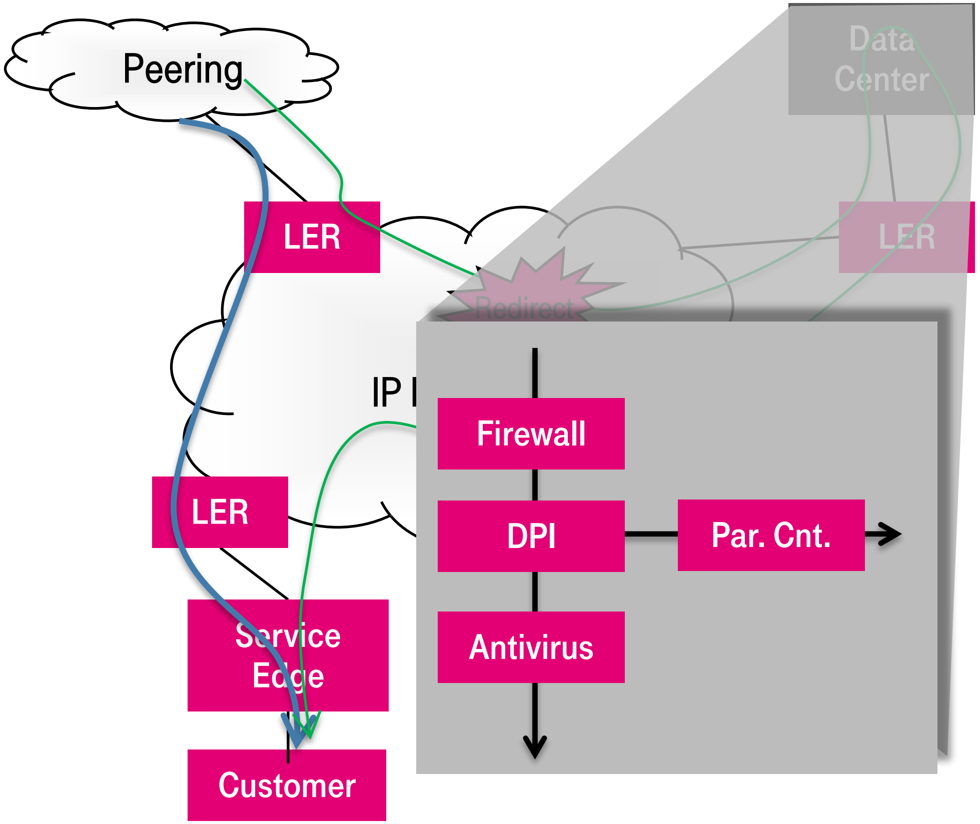
- Only control plane involved, traffic throughput (forwarding performance) not relevant
- Good scalability on standard hardware (e.g. can grow with growing requirement for more performance or memory)
  - Almost unlimited BGP RIB scaling
  - Almost unlimited CPU resource for fast BGP routes calculation
- Simple provisioning of additional RR (e.g. for new services)

Challenges

- “Missing Critical” functionality, in case of malfunction impact on whole network
- Migration strategy from existing solution
- Operational aspects (RR running in DC which might be not under control of network operations)

SDN and NFV
Traffic Steering and Service Chaining.

Dynamic Traffic Steering and Service Chaining



Remarks

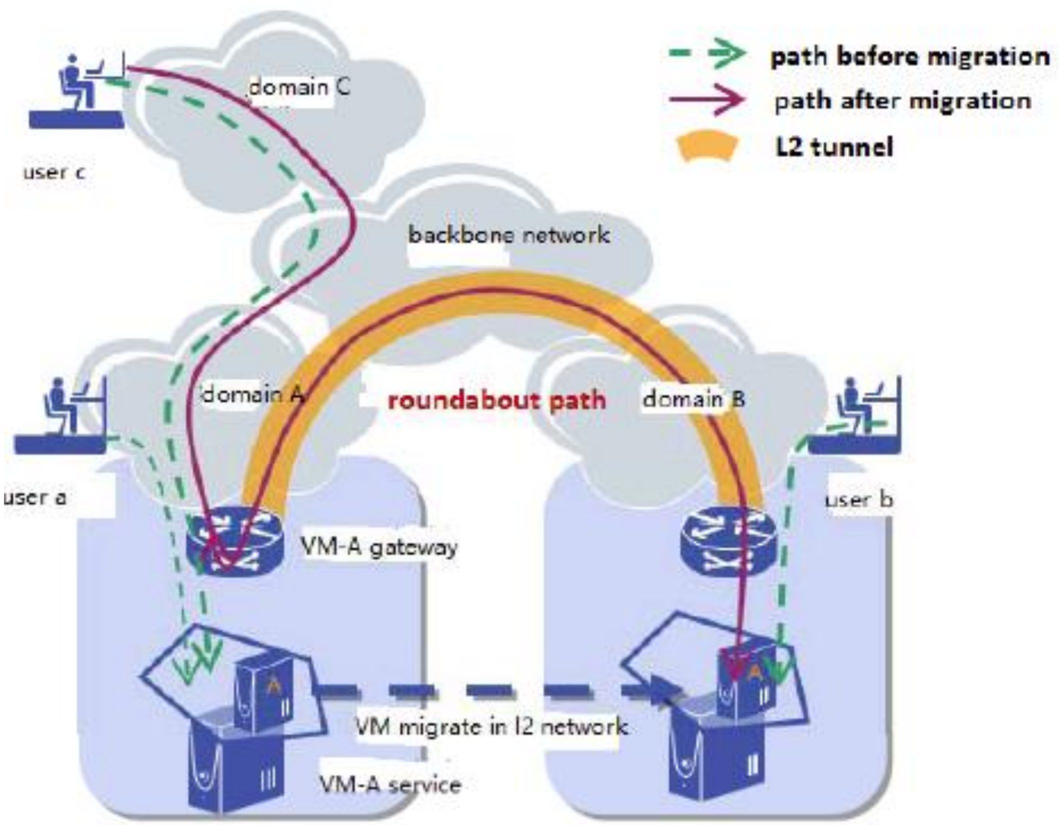
- Need to dynamically classify and steer traffic based on customer demands towards Data Center
- Data Center implements service chaining (providing customer services)
- Traffic classification and steering part of Service Chaining architecture
- End-to-End Orchestration needs to take distributed classification and redirection into account

SDN and NFV

Cross DataCenter VM-Mobility with SDN

Cross DataCenter VM-Mobility with SDN

- VM move from Domain A to B
- VM de-attaches in domain A
- VM attaches in Domain B
- Domain B advertises presence of VM in Domain B
- Establish tunnel to Domain B



SDN and NFV
Use Cases (1 of 3)

Use Case	Brief Description
Inter operator network sharing	• Multiple operators add resources to a common pool forming a global resource view • The demand imposed by the services (e.g. of a particular provider) can be distributed on global resources (as opposed to being distributed on the resources belonging only to the particular provider) • Service cost is formed in real time based on the resources utilized. This requires interaction between multiple SDN domains in peer or overlay model.
Multi-Layer Network resilience	• Operators perform optical applications such as multilayer resilience, multi-domain visualization, service/bandwidth deployment, Traffic Engineering by using SDN controller : • Support automatic per Flow (Lambda) network management, optimization and service provisioning • Intelligence entity to manage the global network information database • Finer granularity will be achieved by combining SDN with IP and optical synergy, i.e., IPoDWDM, OTN
Centralized traffic engineering	• A global data center operator can implement a SDN based WAN that interconnect its data centers. SDN controller performs interior routing and is integrated with Traffic Engineering (TE) function; • Traffic flows and their demands become predictable and well known.
Bandwidth on demand	• The controller is aware about the utilization of the network and shares this information with applications; • Applications calculate the price per unit of bandwidth for a certain period of time; • User can dynamically change the subscribed bandwidth based on need and offered pricing.

SDN and NFV
Use Cases (2 of 3)

Use Case	Brief Description
Flow optimization	• The network controller collects network state and predicts available capacity (network provider domain). • Applications learn about favorable times and locations for flows and registers requests in a scheduler with flexible boundaries (service provider domain) . Initiated by the traffic scheduler the controller programs flows in the network (with load sharing).
Gi-LAN service chaining	• Gi-Interface in Mobile Networks connects the RAN to the mobile packet core. Current Service Chains in Gi-LAN Environments are statically set up. Therefore not flexible enough for new services and elastic service chain handling. With SDN, flexible service chains may be configured, leveraging the APN for selection of the service chain; • Service chains may mix VMs and HW appliances.
Wide area VM mobility	• Implement VM mobility across data centers in same AS. VM is endpoint in a VXLAN based virtual overlay network (at the MAC-Layer) • NFV orchestrates VM-Mobility: Create new VM in datacenter, prepare the move including temporary tunnel for traffic triangulation, move the VM, deviate traffic through tunnel, advertise new reach ability of L2 endpoint • Controller is MP-BGP speaker and advertises both L3 and L2 reach ability information, resulting in path optimization, tunnel can subsequently be removed
Dynamic network service chaining	• Traffic from or towards customers flowing through a chain of service nodes, which perform L4-L7 services, such as DPI, Load Balancers, Firewalls and IDS/IPS, Proxies, Ad insertion, etc. • Service chain can be modified and flexibly configured on the fly as well as personalized per: subscriber class or group, individual subscriber or flows.

SDN and NFV
Use Cases (3 of 3)

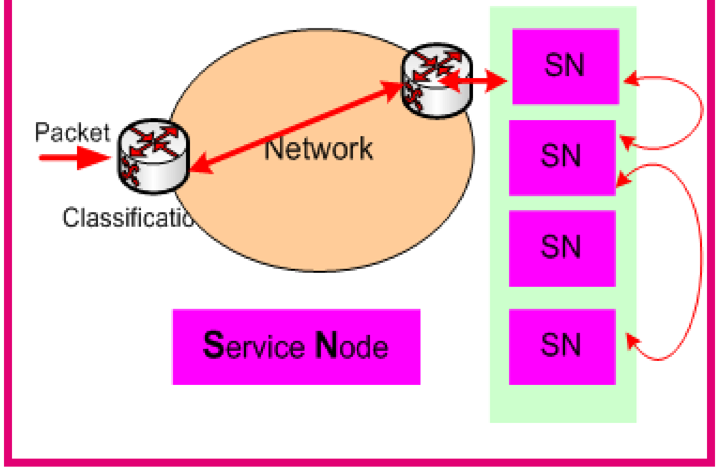
Use Case	Brief Description
DDOS scrubbing	• Probes within the network are constantly monitoring the traffic, checking for DDoS attacks; • If an attack is identified, traffic is first redirected into a data center (where the traffic is “cleaned”) and than sent to the end customer. As a result, the customer is protected against the DDoS attack; • Is a more specific version of the “service chaining” use case.
Security service steering	• Controller collects statistics and forwards via a Northbound API to a Security Control Application (SCA). • SCA detects anomaly and instructs controller to tap(or redirect) the suspicious traffic to an Intrusion Prevention System • Intrusion Prevention System notifies Security Control Application of a detected intrusion attempt • Security Control application triggers creation of a honey pot Virtual Machine and instructs controller to deviate the traffic at the V-Switch to the honey pot Virtual Machine • Is a more specific version of the “service chaining” use case.
Content and CDN optimization	• Network statistics are being monitored in a real time manner to generate statistics about the type of applications and content being used; • The network acts as an application and content aware network by using these statistics to dynamically create CDN nodes and to create optimal paths; • Furthermore, the network can steer traffic towards a destination based on the content and applications being carried within the packet.

Service Chaining

SDN and NFV: Service Chaining

Dynamic Network Service Chaining (NSC)

Graphics



Use Case Description

- Traffic from or towards customers flowing through a chain of service provisioning nodes
- Basic Components:
  - Classification of traffic packets
  - Routing entity which routes the packets towards service chain and/or within service chain
  - Service nodes which applying service related functions
  - Management / Orchestration component
- Components within the chain can be easily modified (and added/removed if necessary)

Benefits / Advantages

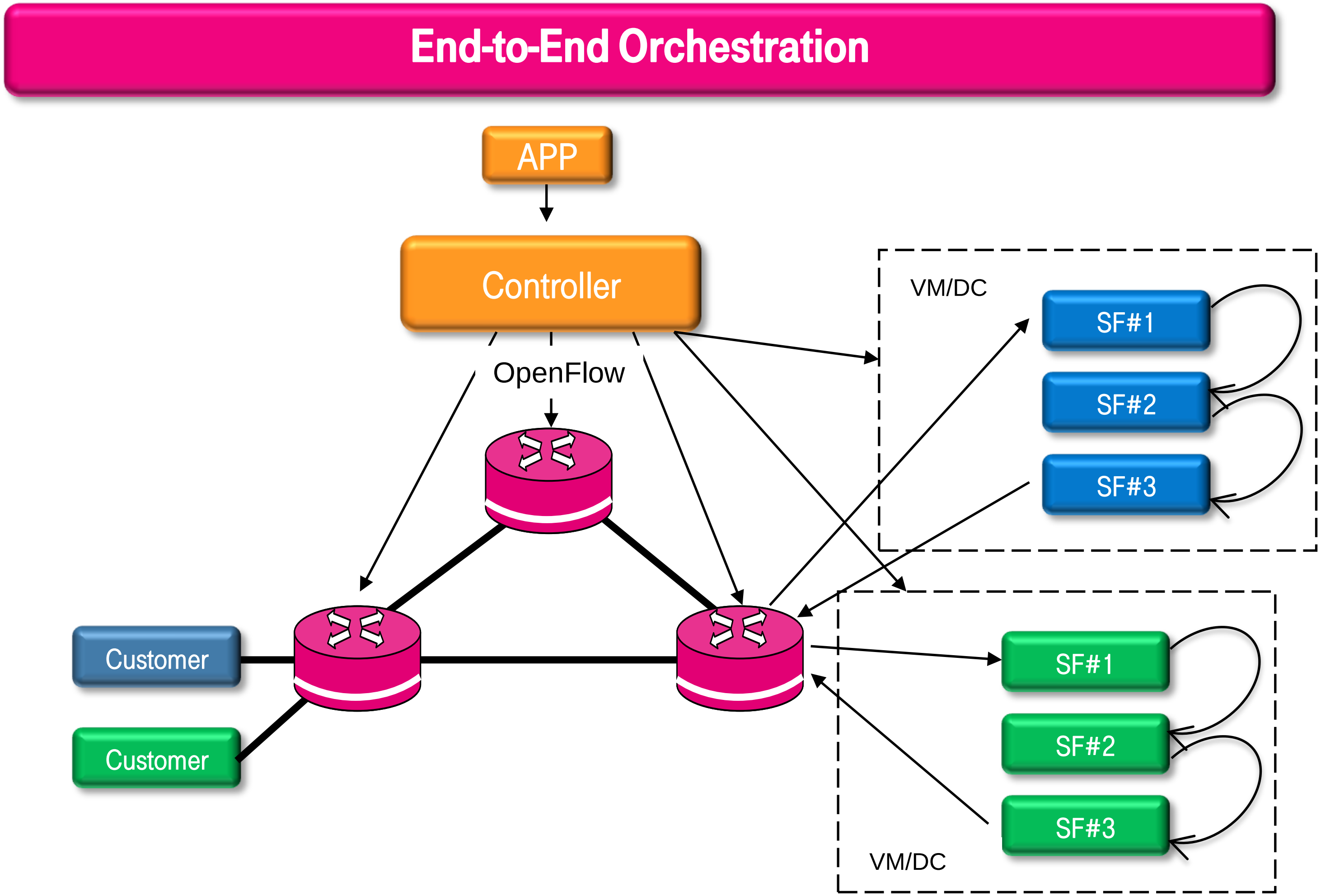
- Higher flexibility compared to existing, static chaining
- Services can be provisioned and changed based on customers demands and Supports "elastic deployments" in virtualized DC
- Migration from existing service chains need more flexible mechanisms
- Reuse of existing service components from the service chain
- If implemented as an overlay, independent of underlying network technology

Challenges

- More dynamic traffic flows, potentially less control
- Integration of service nodes from different sources (e.g. vendors) because of different provisioning models

SDN and NFV: Service Chaining

Network and Service Chain Orchestration.



SDN and NFV: Service Chaining

Motivation

Today

- Service Chaining deployed, typically in static or semi static environments ("hard wired")
- Typically one Service Chain per Service, no reuse of existing components
- Mainly focused (but not only) on mobile networks (e.g. "Gi-Lan")

Tomorrow

- Higher flexibility for fast and simple service delivery
- More elasticity to fulfill temporary needs
- General architecture which can be deployed in fixed and mobile networks

Need for a highly flexible service chaining architecture addressing fixed and mobile networks.

SDN and NFV: Service Chaining

Terminology (draft-liu-service-chaining-use-cases)

Service Processing Function

- a logical entity which can provide one or more service processing functions for packets/frames such as firewall, DPI (Deep Packet Inspection), LI (Lawful Intercept) and etc. Usually these processing functions are computation intensive.

Service Chain

- one or more service processing functions in a specific order which are chained to provide a composite service, and packets/frames from one or more service flow should follow.

Service Chaining

- a mechanism of building service chains and forwarding packets/frames of service flows through them.

Service Path

- a path that traffic flows are forwarded through in a service chain. There might be multiple paths in a service chain.

Service Flow

- packets/frames with specific service characteristics (e.g., packets matching a specific tuple of fields in Ethernet, IP, TCP, HTTP headers and etc.) or determined by some service policies (such as access port and etc.)

SDN and NFV: Service Chaining
Requirements

Requirements for Flexible and Elastic Service Chains

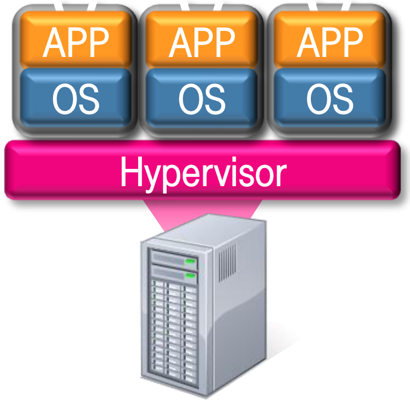
- Flexible and dynamic creation, modification and deletion of Service Chains and/or components of the Service Chain
- Flexible (re-)use of Service Functions
- Agnostic to underlying network technology
- Support of virtualized and non-virtualized service functions
- Support for elasticity and scalability

Service Chaining Architecture, supporting requirements and use cases

SDN and NFV: Service Chaining

Data Center Architecture (1 of 2).

Definition



Key aspects

- Use of standard x86 based server
- Isolation and abstraction of an operating system and its applications in a virtual machine
- Shift from Silos to a common virtualized environment
  (Service are operated as software components based on virtual machines)
- Partitioning of a network into multiple distinct broadcast domains using a common network infrastructure
- Use of a common resource pool for all applications
- Decoupling of hardware and application life.cycles
- Simplification of provisioning processes
- Opex & Capex savings through lower energy & maintenance costs

Benefits / Advantages

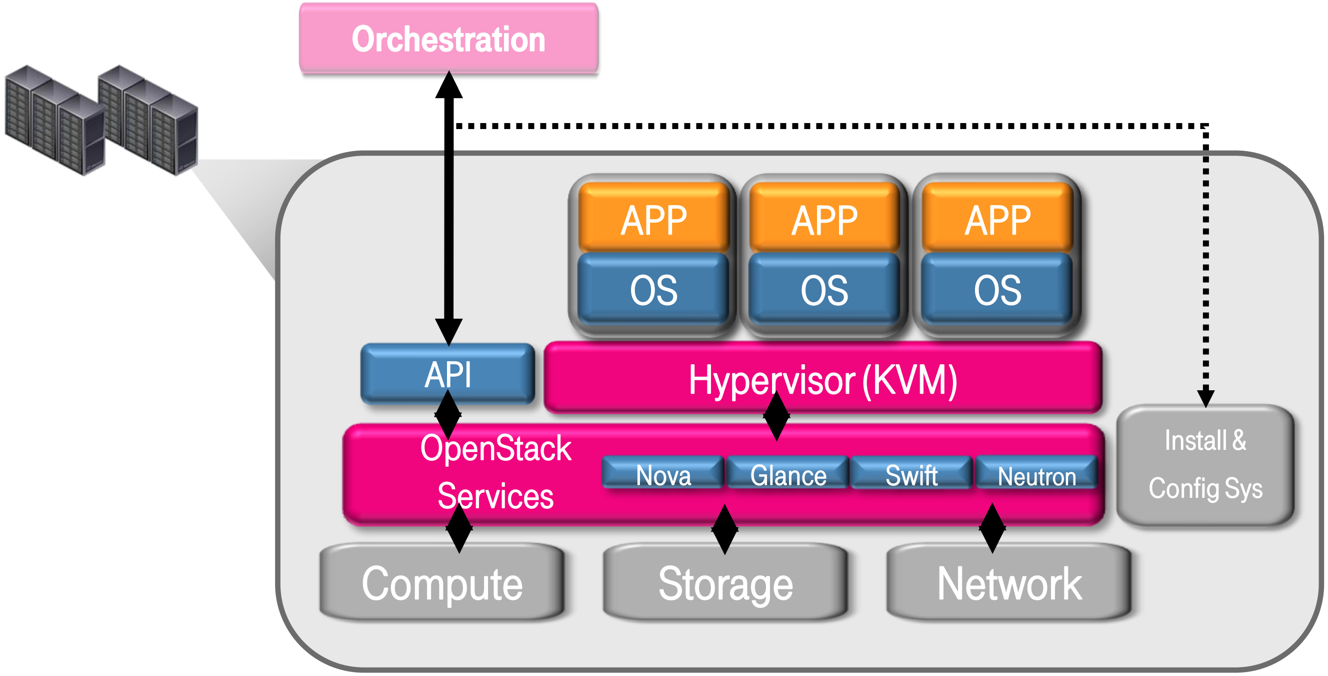
- Abstraction: decouples applications from underlying infrastructure
- Flexibility: flexible resource allocation and management
- Sustainability: enables reduction of energy costs and environmental footprint
- Automation: offers new ways for operational processes

Challenges

- A majority of network functions are not designed to use virtualization
- SLA management has to change

SDN and NFV: Service Chaining
Data Center Architecture (2 of 2).

Generic Data Center Architecture

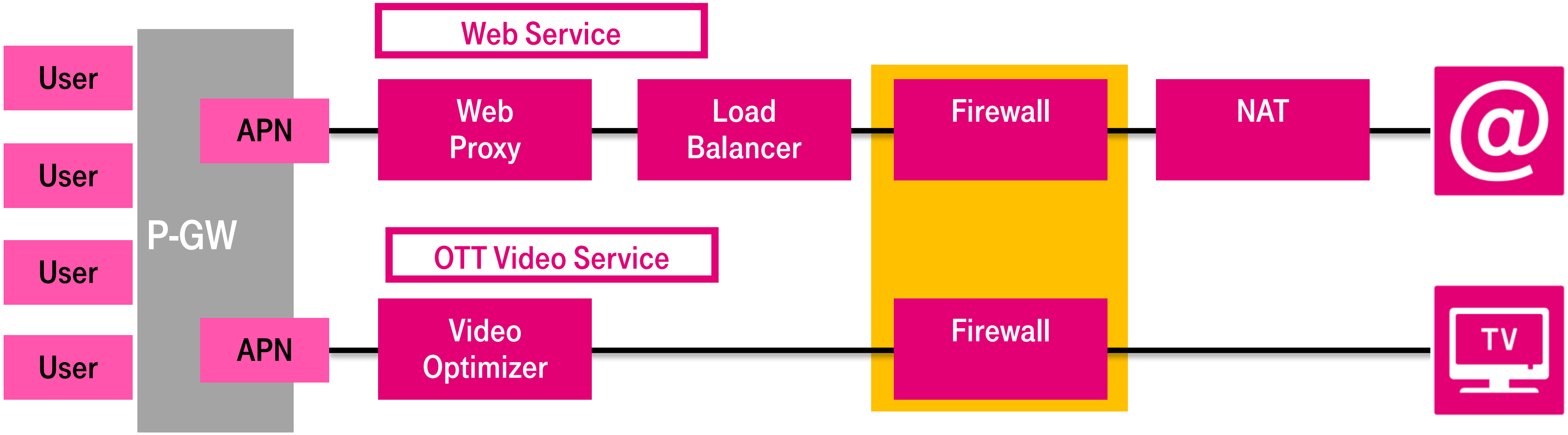


- Orchestration Layer for flexible and dynamic resource provisioning
- Based on standard north and southbound interfaces
- Compute, Storage and Network resources provided "on Demand"

Data Center Architecture as a key component for network function virtualization and flexible service chaining.

SDN and NFV: Service Chaining

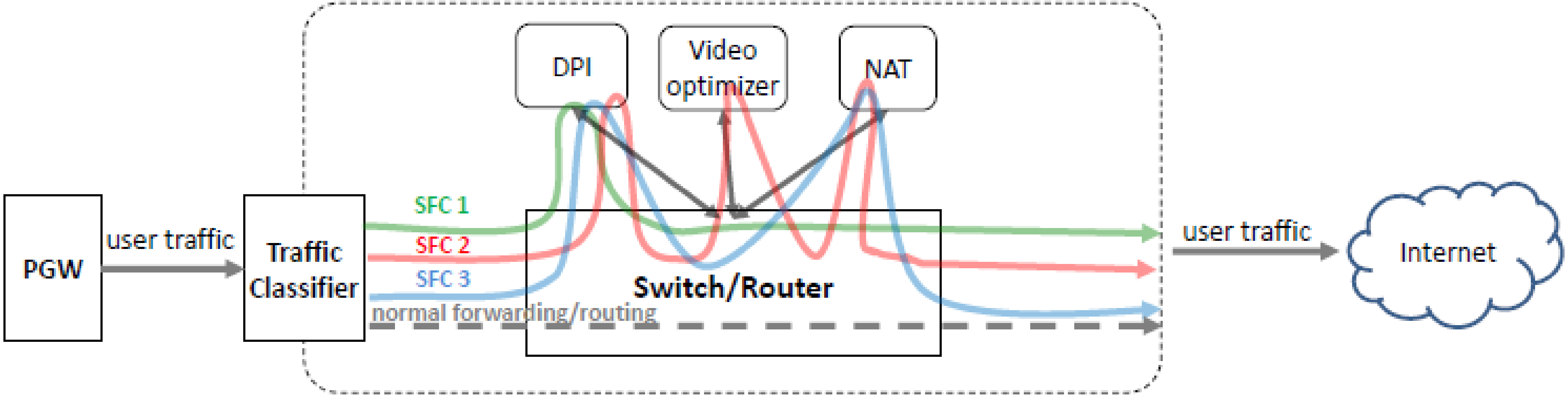
Use Cases: GI-LAN (1 of 2)



- With deployment of additional value-added services increasing number of functions required in Gi-LAN. Some functions in dedicated devices, sometimes multiple functions in one box.
- Due to fast service introduction cycles service chains emerge, growth & change evolutionary.
- Very often static IP links, policy routing, VRFs etc. used to enforce required service sequence.
- Results in steadily increasing, handcrafted complexity and decreased visibility of functional dependencies between service chains and underlying LAN topology. Means expensive OAM.
- Practically impossible to implement automated service provisioning and delivery platform.

SDN and NFV: Service Chaining

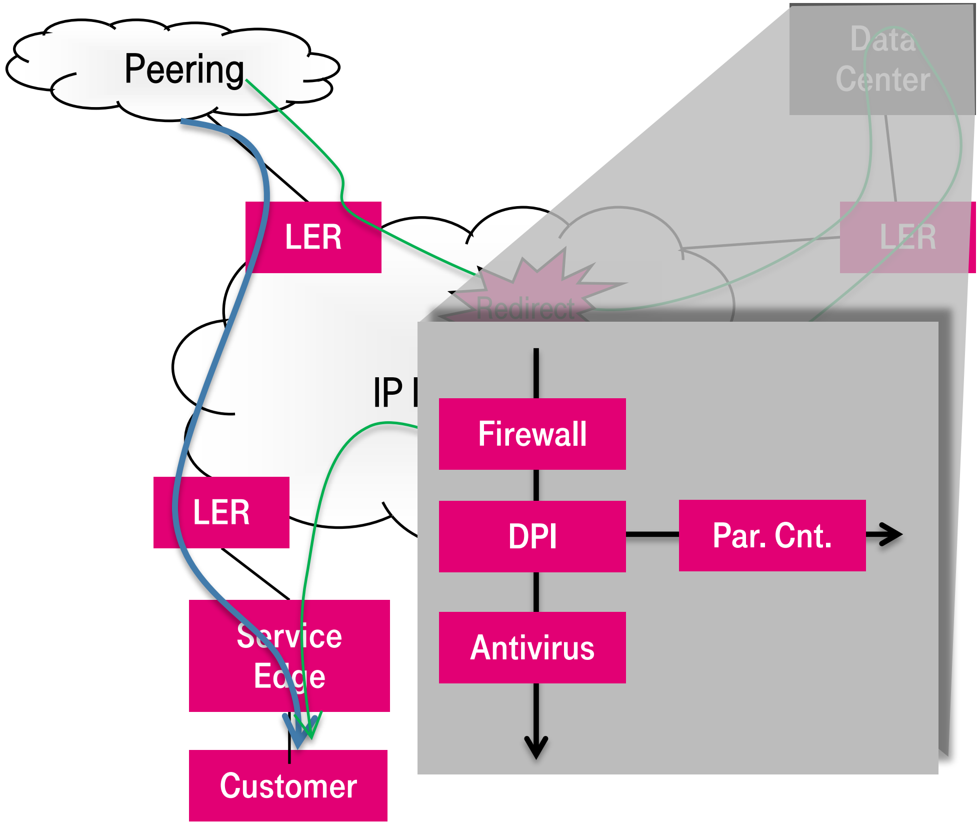
Use Cases: GI-LAN (2 of 2)



- A service function chain is a different path from normal forwarding or routing, where traffic is steered according to service characters
- Traffic must be steered through service functions in a specific sequence
  - Service function chain1: default path
  - Service function chain2: http video traffic
  - Service function chain3: https

SDN and NFV: Service Chaining
Use Cases: Traffic Steering and Service Chaining.

Dynamic Traffic Steering and Service Chaining



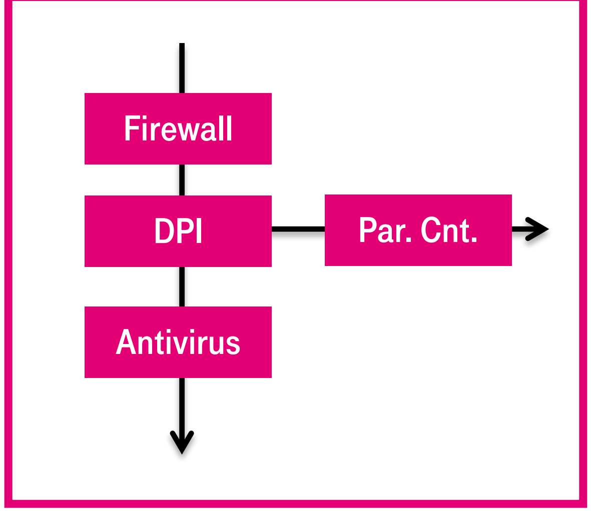
Remarks

- Need to dynamically classify and steer traffic based on customer demands towards Data Center
- Data Center implements service chaining (providing customer services)
- Traffic classification and steering part of Service Chaining architecture
- End-to-End Orchestration needs to take distributed classification and redirection into account

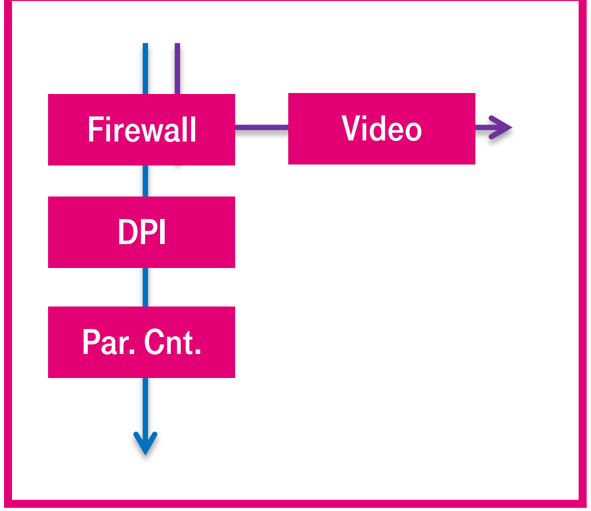
SDN and NFV: Service Chaining

Use Cases: Chaining Variants

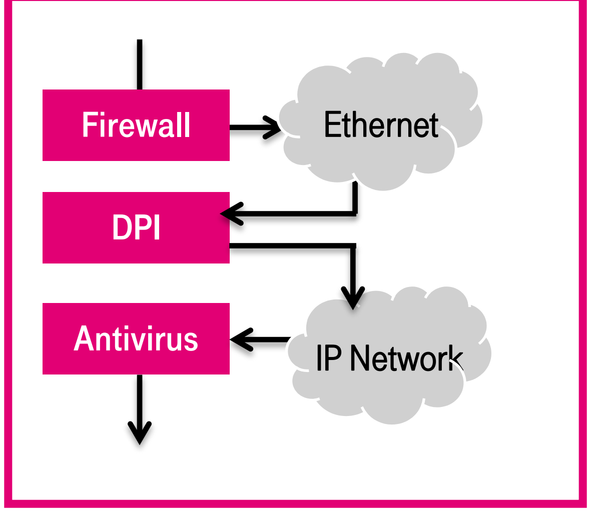
Service Path Forking



Service Function Sharing



Multiple Underlay Networks



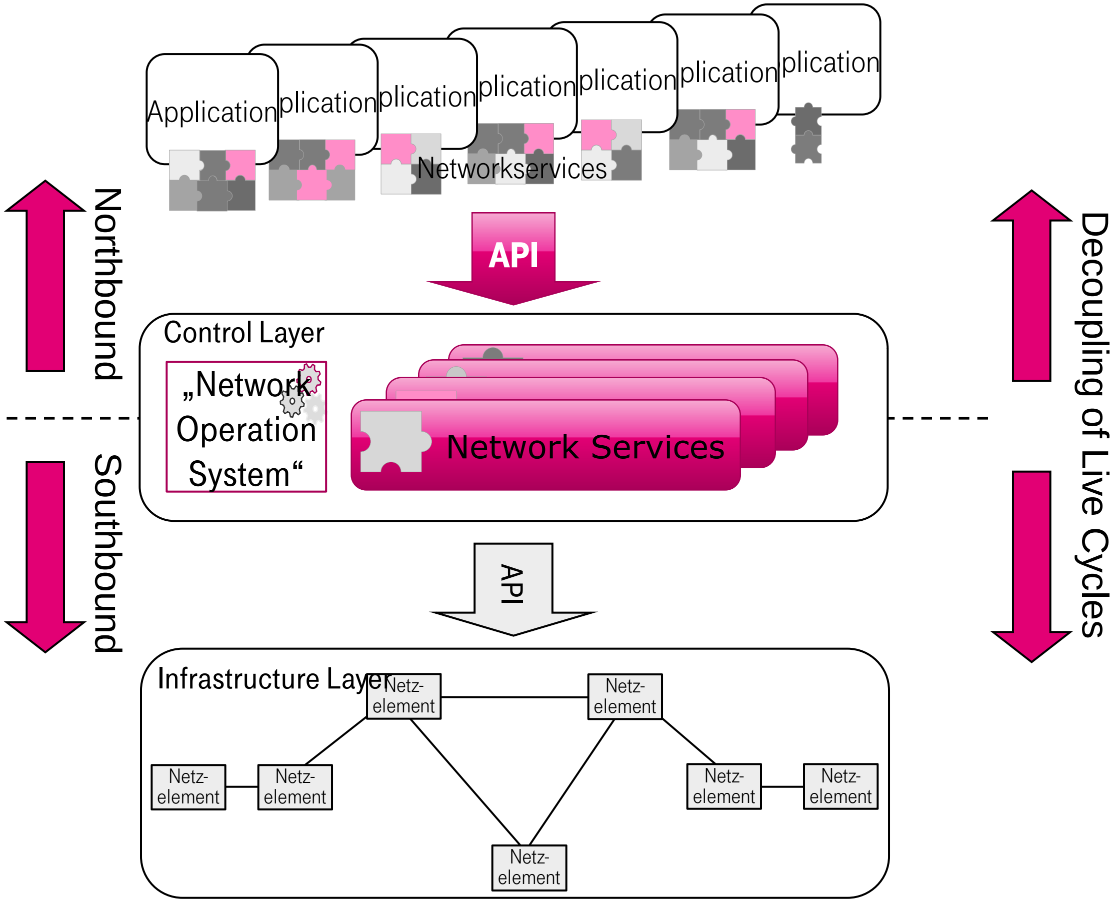
- A single flow being forked within one component of the Service Chain

- Parts of the Service Chain are being reused for other Service Chains

- Service Chaining not to be bound to a single network/transport protocol

SDN and NFV: Service Chaining

Architecture Options: Openflow based Service Chaining



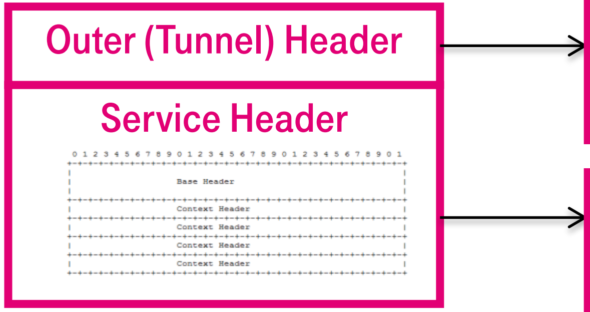
Key Components

- Application used to build service chains sitting on top of an OF Controller
- Controller offers a common API (or several APIs) northbound
- Controller uses southbound interface towards abstraction layer (e.g. openflow)
- Abstraction layer integrates different NEs and protocols
- OpenFlow used to build network path between service nodes

SDN and NFV: Service Chaining

Architecture Options: Service Header (1 of 3)

Service Header



Outer (Tunnel) Header

Service Header

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                          Base Header                          |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Context Header                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Context Header                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Context Header                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Context Header                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

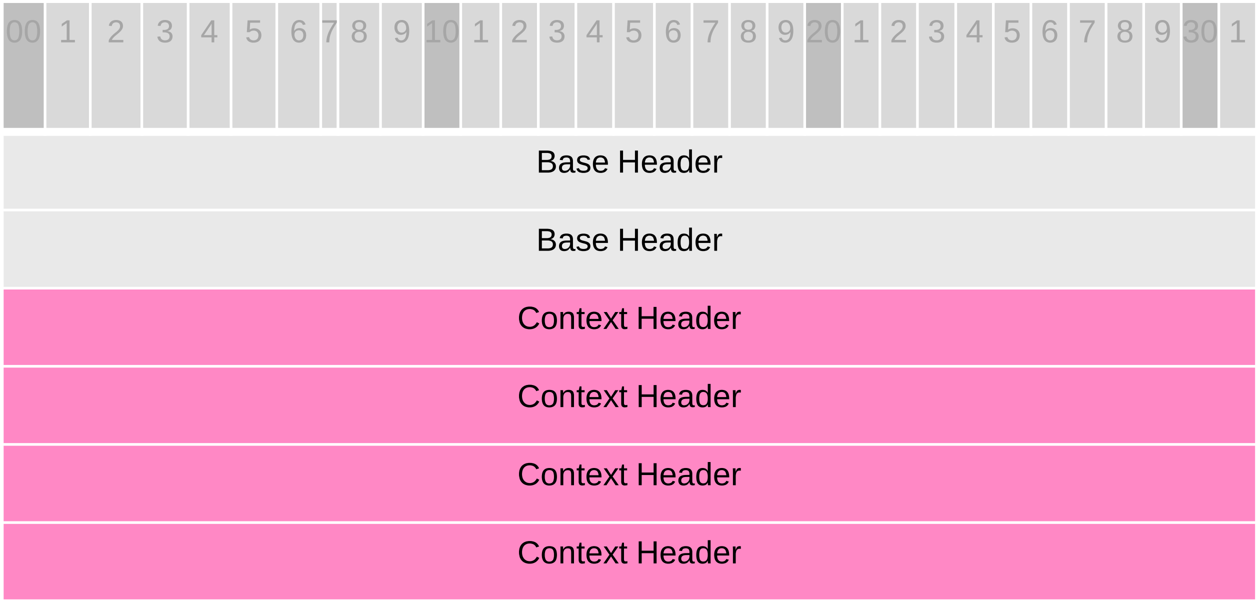
- Used to build an overlay network in order to address components of the service
- Independent of underlying transport network

- Service header is meta data added to a packet or frame that is used to create a service plane
- Added by a service classification function (device or application)

- Service header indicated using protocol type or GAL (in case of MLPS as encapsulation)
- Service header is inserted/removed at start or end of service chain/path or by service function within the chain
- Service header can be changed by node within the service chain (e.g. in case of reclassification)
- Service chain is independent of topology

SDN and NFV: Service Chaining

Architecture Options: Service Header (2 of 3)



- Simple, fixed size header
  - six 32 bits words (2 x word base header, 4 x 32 bit mandatory context (metadata) headers)
- Transport agnostic
  - VXLAN, LISP, NVGRE, MPLS, etc.
- Context headers carry metadata along the service path
  - Significance determined by the control plane;
  - Innovate, create network value!

– Source: linuxplumbers–

SDN and NFV: Service Chaining

Architecture Options: Service Header (3 of 3)

<table>
<tr><td>00</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>20</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>30</td><td>1</td><td></td></tr>
<tr><td>O</td><td>C</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td colspan="16">Protocol Type (16)</td><td colspan="8">Service Index (8)</td><td>Base Header 1</td></tr>
<tr><td colspan="24">Service Path Identifier (24)</td><td colspan="8">-</td><td>Base Header 2</td></tr>
<tr><td colspan="32">Network Platform Context</td><td>Context Header 1</td></tr>
<tr><td colspan="32">Network Shared Context</td><td>Context Header 2</td></tr>
<tr><td colspan="32">Service Platform Context</td><td>Context Header 3</td></tr>
<tr><td colspan="32">Service Shared Context</td><td>Context Header 4</td></tr>
<tr><td colspan="32">Original Packet Payload</td><td></td></tr>
</table>

- **O:** OAM bit indicates packet is an OAM packet and must be punted
- **C:** indicates that the context headers are in use and their allocation (if C = 0, all context values = 0, the headers are still present, just unused)
- **Protocol Type** of the original packet payload
- **Service Index** provides loop detection and location within the service chain. Can be used with a Service Path Identifier to derive unique value for forwarding
- **Service Path Identifier**: identifier of a service path; used for service forwarding
- **Context Headers**: packet metadata

– Source: linuxplumbers–

SDN and NFV: Service Chaining

Summary

Today's Service Chaining does not fulfill existing and new requirements.

Service Chaining needs to be flexible, elastic and scalable.

Traffic classification and redirection part of service chaining.

Service Chaining independent of underlying network topology/network protocols.

Still in an early phase, use cases will evolve during time.

Different approaches possible (e.g. SDN, Overlay, ...)

Standardization

SDN and NFV
Standardisation (1 of 4)

ONF

- non-profit organization founded in March 2011 by Deutsche Telekom, Facebook, Google, Microsoft, Verizon, and Yahoo!
- Currently >70 members.
- ONF is intended to accelerate the delivery and use of Software Defined Networking (SDN) standards, products, services, applications, customers, and users.
- Continued development of the OpenFlow Specification (OpenFlow specification v1.3 has been released).
- OpenFlow implemented by >15 routing and switch vendors. Additionally >10 software implementations exist.
- ONF is more focusing on the southbound interface, i.e., interface between controller and data forwarding NE
  - Extending the technology
  - Encourage the implementation and development
  - Evangelize SDN/OF

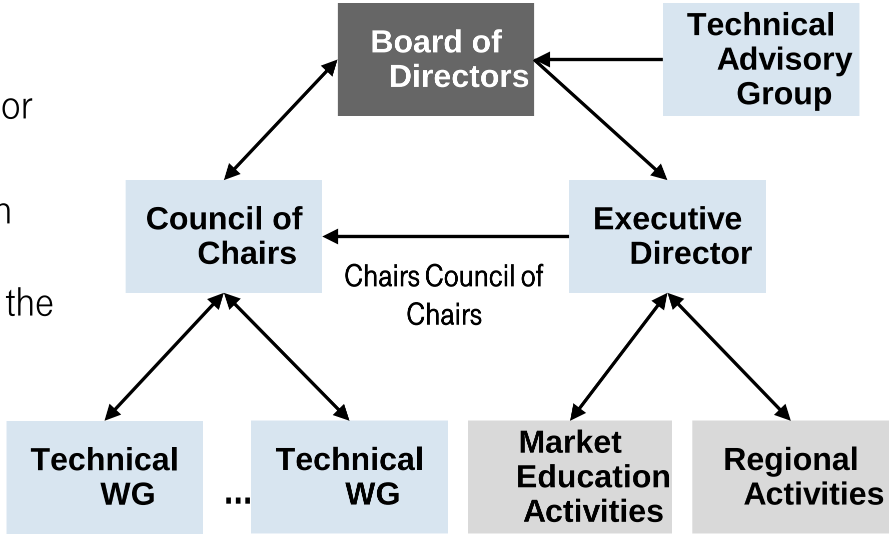
IETF

- IETF started SDN topic since the side meetings at IETF81:
  - SDN BoF at IETF82
  - SDNRG at IETF84
  - SDNRG Draft Charter
- Area ADs discussing where the work will ultimately be progressed.
  - Possible candidates include Routing, Application and Operations.
- IETF SDN research group is trying to describe the whole SDN model/framework, which includes:
  - Hybrid models, Classification of SDN models and their relationship to existing IETF work, SDN model scalability and applicability, Multi-layer programmability and feedback control systems, System Complexity, Network description languages, abstractions and APIs

SDN and NFV
Standardisation (2 of 4)

ONF

- The Board of directors is comprised of users, not vendors
- The Executive director is ONF employee (vendor neutral) and reports to the board
- The Technical advisory group provides opinion on fundamental technical issues (only recommendations, not decisions). It reports to the board.
- The council of chairs assures cross WG consistency and forwards draft standards to the board

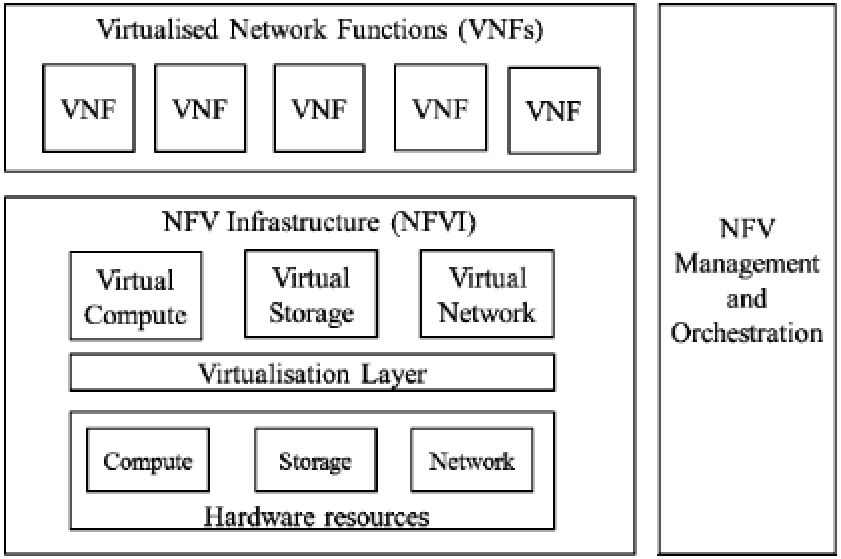


- The WGs define scope, deliverables and timeline
- The following WGs exist currently – Extensibility, Configuration & Management, Testing & Interoperability, Hybrid, Market Education, Architecture & Framework, Forwarding Abstractions. In addition, there are four discussion groups - Skills Certification, New Transport, Security and Japanese

SDN and NFV
Standardisation (3 of 4)

ETSI NFV

- Industry Specification Group (ISG) founded by 7 ISPs in 2012 (today: 230 companies, include 37 service providers)
- The goal is to provide standards for Network Functions Virtualization as well as sharing experiences of NFV development and early implementation.
- Definition of requirements, architecture and gap analysis for the virtualization of network functions
  - Management and Orchestration
  - Taking high performance and portability into account
- Provides an architectural framework to implement NFV
- Prepares PoCs (based on requirements and use cases)



SDN and NFV
Standardisation (4 of 4)

IETF: I2RS

- According to the IETF I2RS group, a routing system is defined as “all or part of a routing network such as an interface, a collection of interfaces, a router, or a collection of routers”.
- Interfaces to the Routing System (I2RS) “facilitate real-time or event driven interaction with the routing system”
- The IETF I2RS work group aims to develop a framework and architecture enabling particular use cases. The initial use cases include:
  - Interactions with the RIB
  - Association of routing protocols with the routing state
  - The ability to extract information from the network
- One of the IETF drafts produced (draft-hares-use-case-vn-vc-00) describes how SDN networks can implement automated network services for Virtual Connection on Demand (VCoD) and Virtual Network on Demand (VNoD) using the IR2S interface

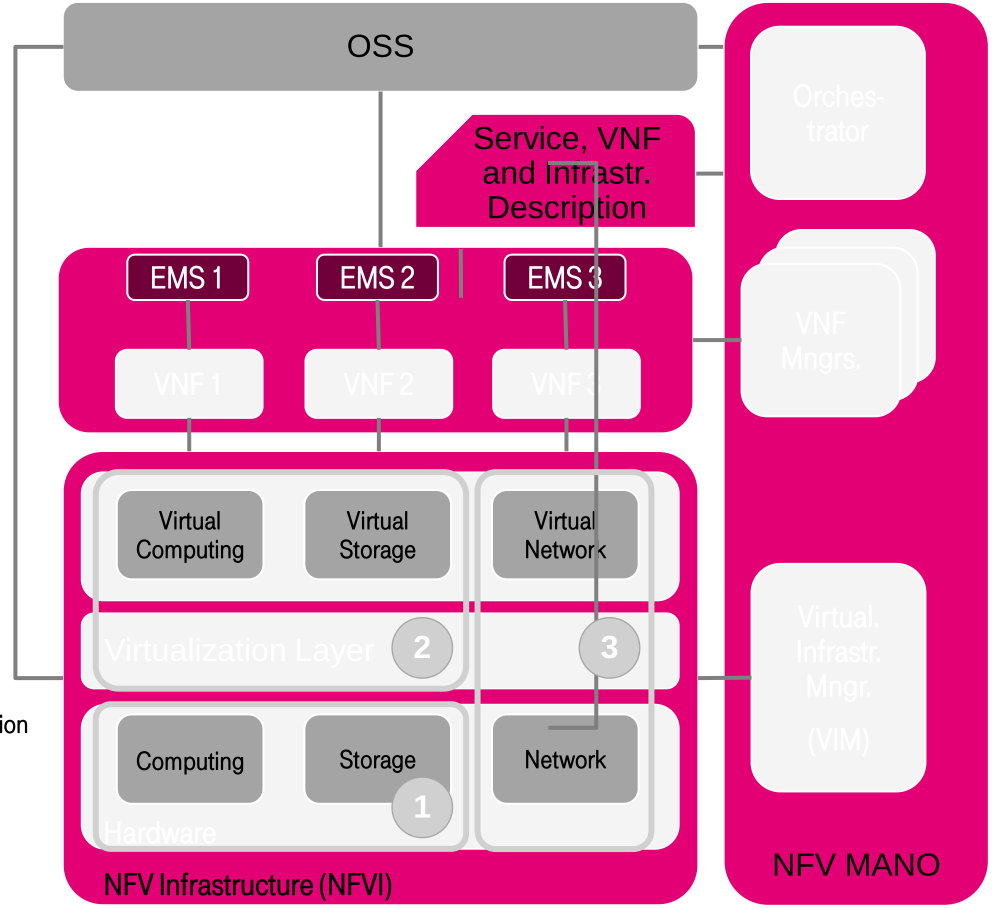
Other IETF Activities

- Security Analysis
  - security of SDN architectures is analyzed in draft-hartman-sdnsec-requirements-00
  - Security of ONF OpenFlow specification is examined in draft-mrw-sdnsec-openflow-analysis-00
- SDN-based Controllers
  - Federated SDN-based Controllers for NVO3 are described in draft-sb-nvo3-sdn-federation-01. This solution has been presented on the SDN summit.
- SFC (Service Function Chaining)
  - Develops a SFC solution based on the “Service Header”
- I2NSF (Interface to Network Security Functions)

SDN and NFV
ETSI NFV Architecture

- recognized consensus for NFV architecture building blocks, functions and reference points is being developed by an ETSI Industry Specification Group.
  - Multiple instances of virtual compute / storage / network share the available hardware
  - The Hypervisor manages virtual compute / storage resources and exposes them to applications
  - Virtual Network Functions are applications consuming the virtual resources
  - Element Managers, OSS and NFV Management and Orchestration (NFV MANO) fulfill different management roles

1 Compute Domain	VNF – Virtual Network Function
2 Hypervisor Domain	EMS – Element Management System
3 Infrastructure Networking Domain	MANO – Management and Orchestration

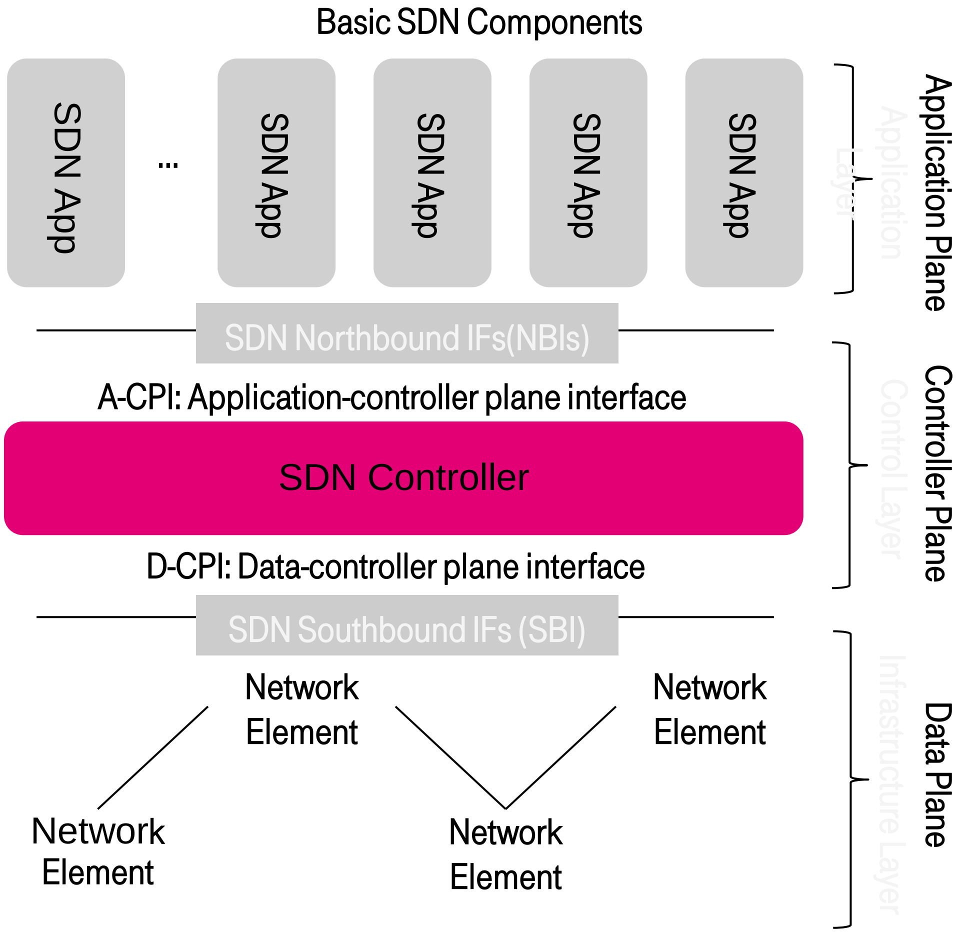


Source: *ETSI GS NFV 002 v1 1 1 NFV Architectural Framework*

SDN and NFV
ONF OpenFlow Architecture

High network flexibility in response to application demand is managed by an intelligent controller interacting with the application plane as well as the networking plane.

- The Application-controller plane interface (A-CPI) is also known as the northbound interface. Via the A-CPI interface:
  - network resources and state are abstracted by the controller for the usage of applications;
  - thus enabling the programmability of the network
- Data-controller plane interface (D-CPI) enables the SDN controller to control data plane resources. It performs functions such as programmatic control of all functions, capabilities advertisement and event notification
- Openflow specifies two southbound (D-CPI) protocols:
  - Openflow protocol: used to manage the flow table on network elements
  - Openflow-Config (OF-Config): used to manage system configuration (e.g. port configuration) of a network element.

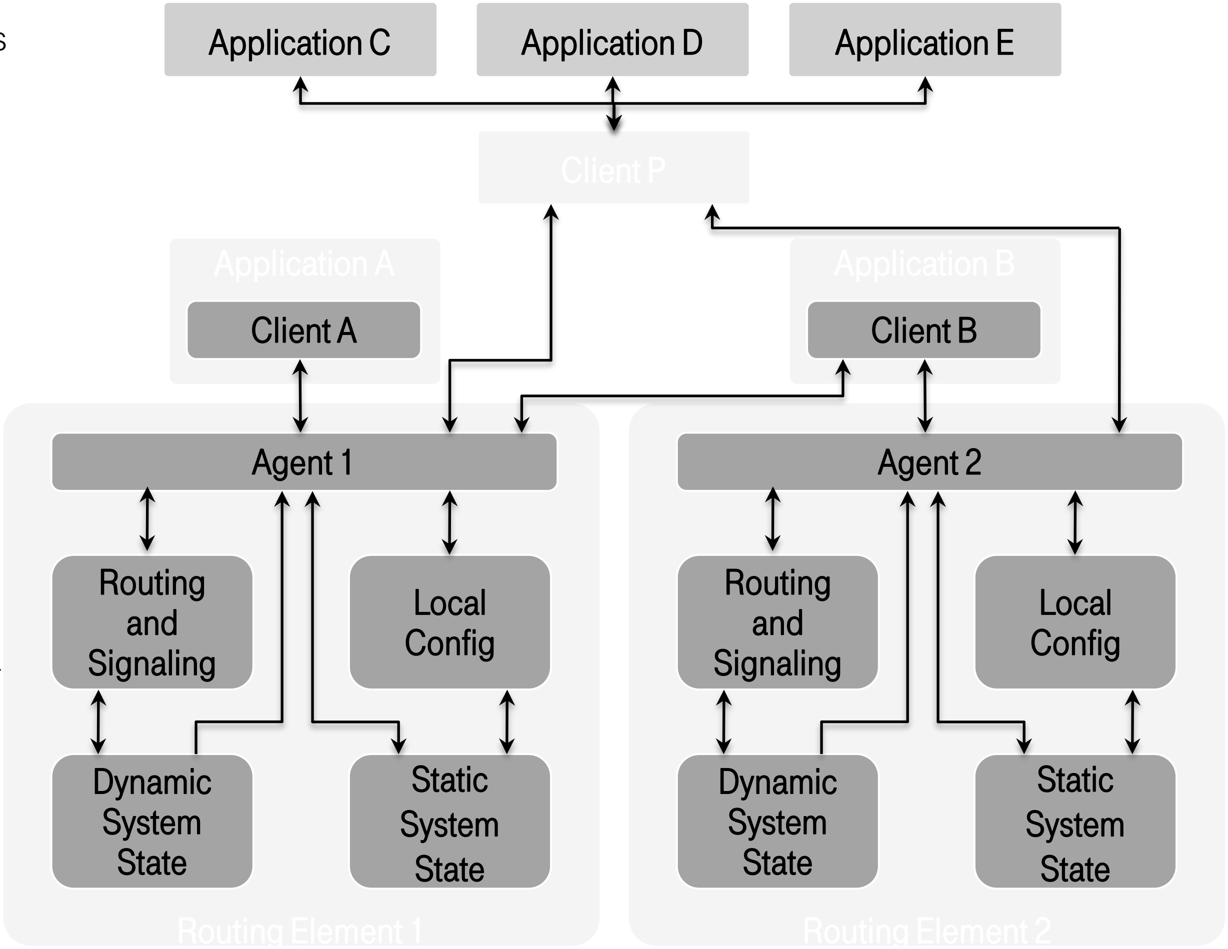


Source: ONF, https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR_SDN_ARCH_1.0_06062014.pdf

SDN and NFV
IETF I2RS Architecture

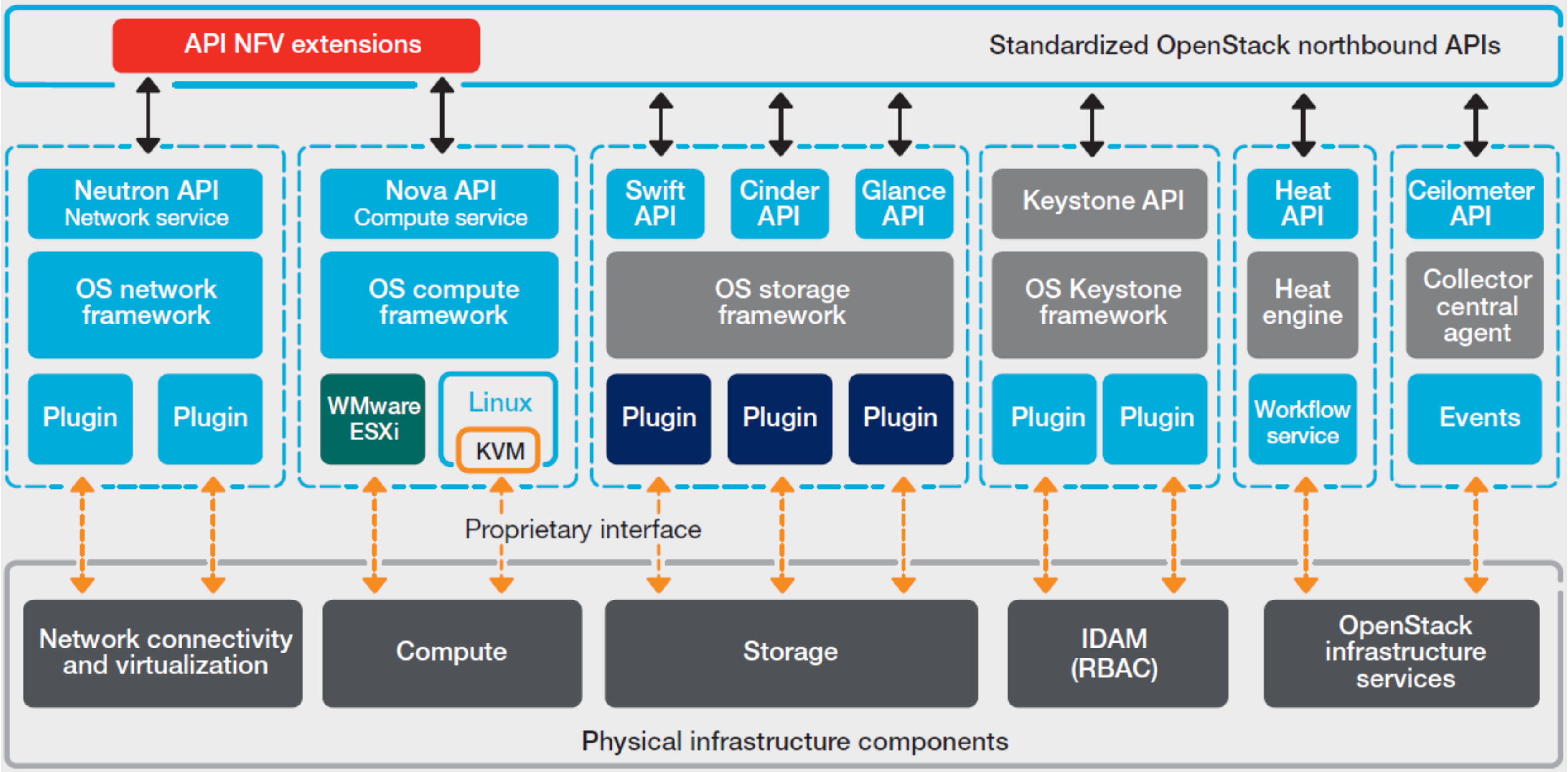
The interface to the routing system (I2RS) architecture is being specified by the IETF. It is in working group draft status (version 09 as of March 6, 2015)

- It introduces centralized control functions in a similar manner as Openflow.
- However, it differs from Openflow in that control functions (routing and signaling) remain integral parts of routing elements (i.e. no full centralization of control);
- The scope of I2RS is to define the interactions between the I2RS agent and I2RS client and the associated behavior;.
- The details of how applications communicate with a remote client are out of scope for I2RS;
- I2RS should be used in conjunction with a protocol such as NETCONF (using YANG datamodels) to implement changes to the local configuration of a routing element (e.g. configuration of routing and signaling).



Source: draft-ietf-i2rs-architecture-09 (March 6, 2015)

SDN and NFV
Openstack Architecture



Solutions

SDN and NFV
SDN/NFV Solutions

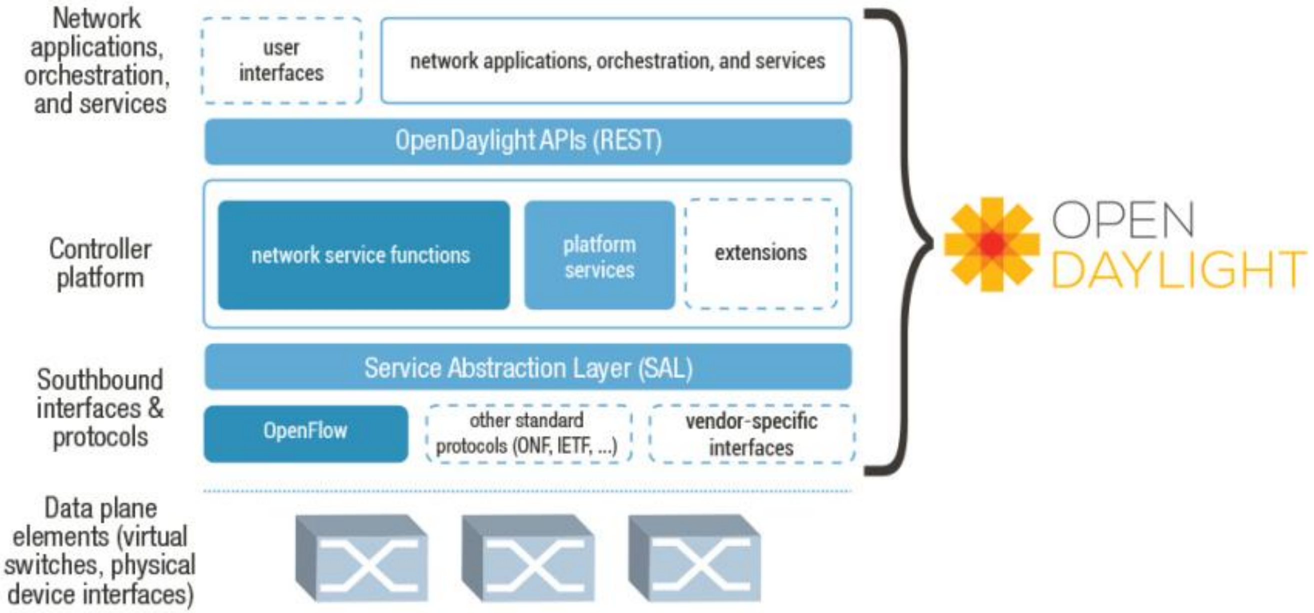
- Wide variety of open source and commercial products available
- covering end-to-end including data center or only parts of the end-to-end scenarios (e.g. focussing on data center only or network control only)
- Examples:
  - OpenDaylight (OpenSource)
  - OpenContrail (Juniper, OpenSource)
  - OnePK (Cisco)

SDN and NFV

Examples: OpenDaylight (OpenSource)

- OpenDaylight is an open source project under the Linux Foundation
- The project aims to create code and blueprints that comprise a community-led, open, industry-supported SDN framework
- It was founded jointly by major IT and network vendors as well as SDN&NFV startups

- The projects within OpenDaylight include SDN controller, Yang tools, OVSDB integration, etc.
- Helium, the second release of the OpenDaylight controller was announced in Sept 2014
- Currently over a dozen of vendors are building controllers based on OpenDaylight code

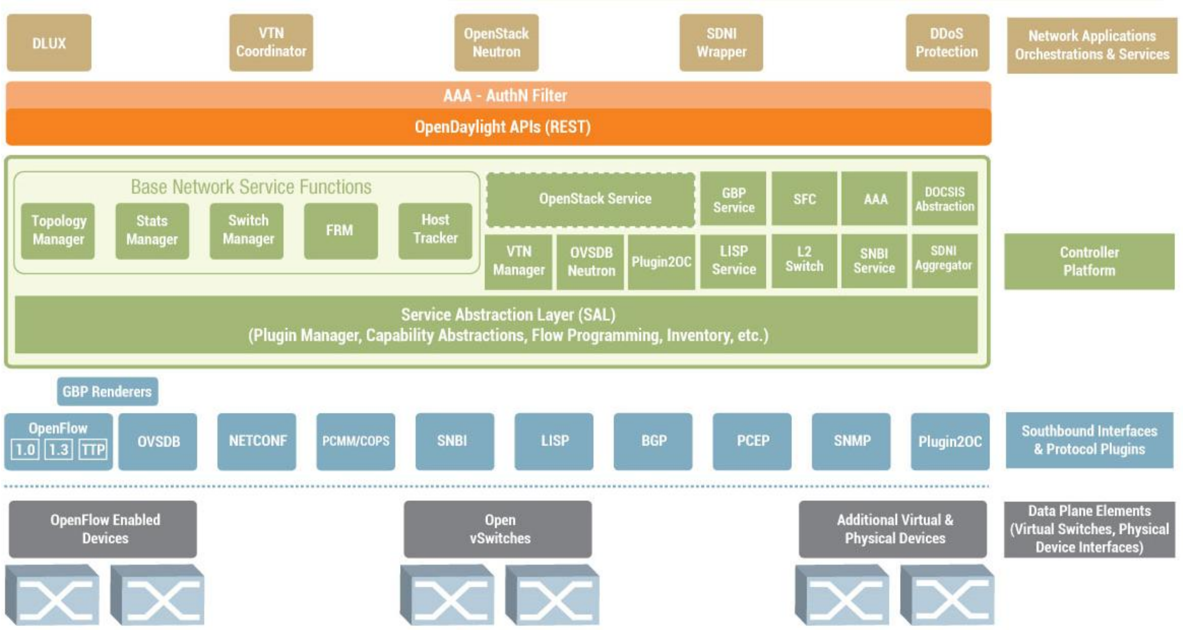


Network applications, orchestration, and services
user interfaces
network applications, orchestration, and services
OpenDaylight APIs (REST)
Controller platform
network service functions
platform services
extensions
Southbound interfaces & protocols
Service Abstraction Layer (SAL)
OpenFlow
other standard protocols (ONF, IETF, ...)
vendor-specific interfaces
Data plane elements (virtual switches, physical device interfaces)
OPEN DAYLIGHT

SDN and NFV

Examples: OpenDaylight (OpenSource)

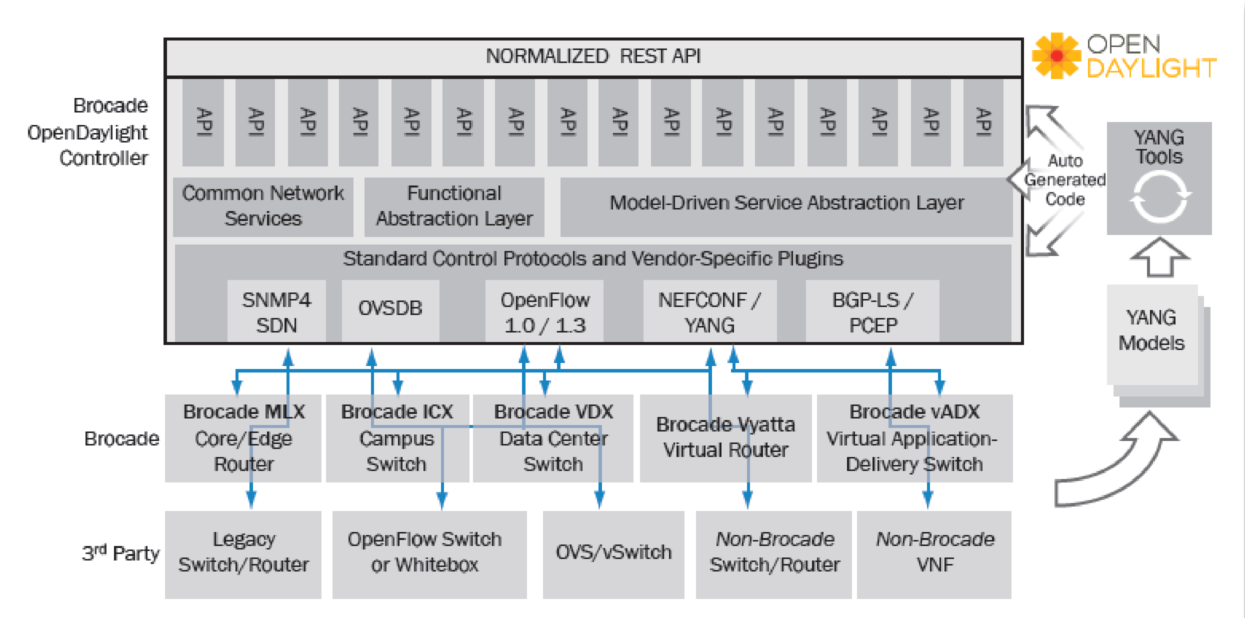
- The OpenDaylight controller:
  - Is modular, pluggable and based on Java
  - The modules cover basic network functions as well as platform oriented services and other extensions
  - Northbound it supports OSGi framework and bidirectional REST
  - The API exposed northbound matches Neutron API precisely
  - Multiple southbound protocols can be utilized including OpenFlow 1.0, OpenFlow 1.3, BGP-LS, etc



SDN and NFV

Examples: Vyatta Controller (OpenDaylight based)

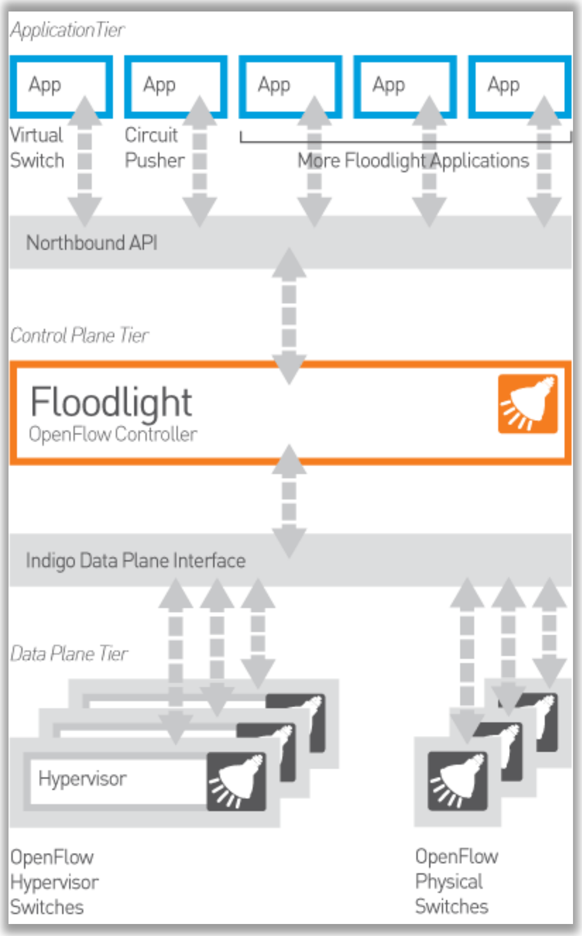
- The Vyatta Controller is built directly from OpenDaylight Helium code, therefore:
  - It can control any network element compatible with the OpenDaylight southbound interfaces
  - network applications developed on Vyatta can be run on any OpenDaylight-based controller
- In addition to OpenDaylight code, Brocade's commercial offer provides enhancements, such as:
  - single-source technical support
  - pre-tested solution and customized distribution packages
  - GUI and tools
  - network applications (licensed separately)



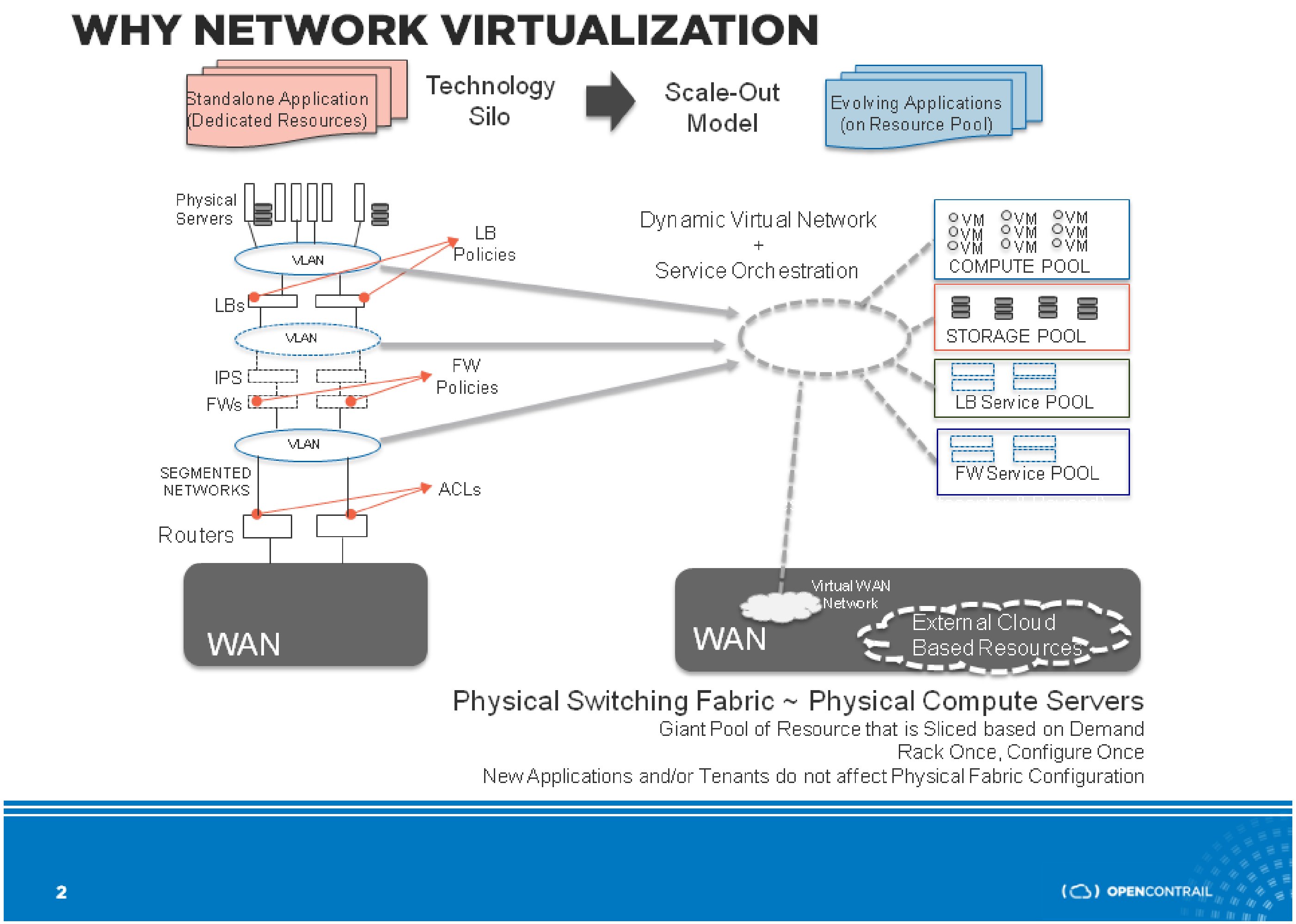
NORMALIZED REST API
OPEN DAYLIGHT
Brocade OpenDaylight Controller
API API API API API API API API API API API API API API API API
Auto Generated Code
YANG Tools
Common Network Services
Functional Abstraction Layer
Model-Driven Service Abstraction Layer
Standard Control Protocols and Vendor-Specific Plugins
SNMP4 SDN
OVSDB
OpenFlow 1.0 / 1.3
NEFCONF / YANG
BGP-LS / PCEP
YANG Models
Brocade
Brocade MLX Core/Edge Router
Brocade ICX Campus Switch
Brocade VDX Data Center Switch
Brocade Vyatta Virtual Router
Brocade vADX Virtual Application-Delivery Switch
3rd Party
Legacy Switch/Router
OpenFlow Switch or Whitebox
OVS/vSwitch
Non-Brocade Switch/Router
Non-Brocade VNF

SDN and NFV
Examples: Floodlight

- Floodlight is:
  - An OpenFlow Java-based controller developed by an open community of developers and a collection of applications built on top the Floodlight Controller
  - The applications are either built as Java modules or are leveraging the services via REST API
  - It is the core of the commercial controller product from Big Switch Networks
  - It supports:
    - OpenFlow ver. 1.0
    - OpenStack cloud orchestration platform



SDN and NFV
Examples: OpenContrail (Juniper, OpenSource)



WHY NETWORK VIRTUALIZATION

Physical Switching Fabric ~ Physical Compute Servers

Giant Pool of Resource that is Sliced based on Demand

Rack Once, Configure Once

New Applications and/or Tenants do not affect Physical Fabric Configuration

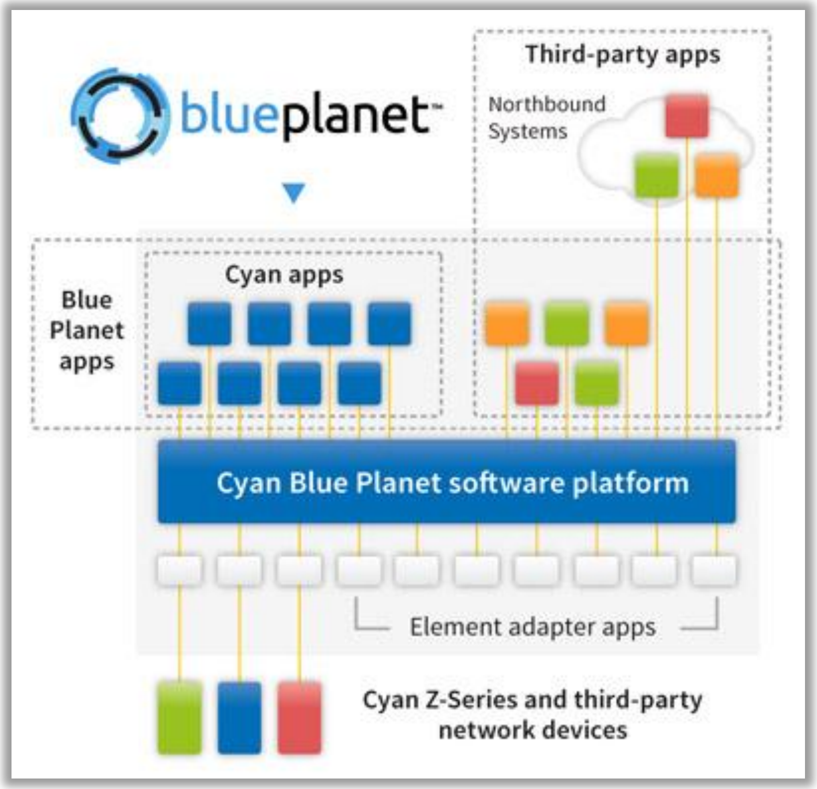
2

OPENCONTRAIL

SDN and NFV

Examples: Blue Planet (Cyan) SDN and NFV Orchestration

- Cyan's Blue Planet System is a carrier-grade, multi-vendor SDN and NFV orchestration platform
- It is built as a family of applications, out of which Planet Operate is the one incorporating centralized control plane
- In addition, Planet Operate performs traffic management, resource management and acts as a multi-layer Path Computation Element
- Blue Planet supports:
    - Southbound - CLI, TL1, NETCONF/YANG, etc.
    - Northbound – supports Blue Planet apps, including Carrier Ethernet Services and Optical Services APIs; as well as "open API" to integrate with OSS/BSS systems

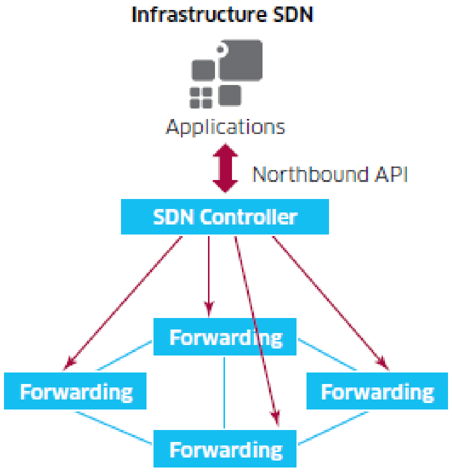


SDN and NFV

Underlay and Overlay or Infrastructure Model

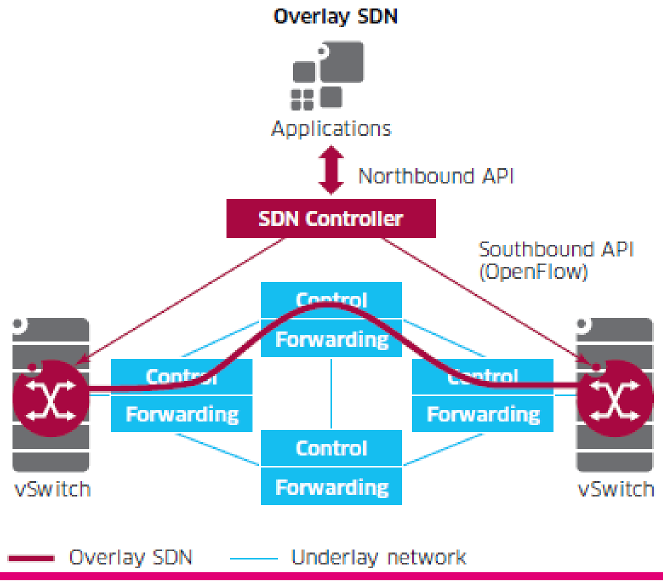
Infrastructure Model

- "Pure SDN" – realizing the original idea of SDN in terms of simplicity
- The network is controlled entirely by the SDN controller, including all forwarding elements (virtual and physical)
- The forwarding elements are simply forwarding; no support for tunneling mechanisms e.g. VXLAN, NVGRE or MPLS over GRE/IP is required



Overlay/Underlay Model

- An overlay implementing the SDN principles is created on top of the (typically existing) underlay network
- The overlay is created by a mesh of virtual tunnels, based e.g. on VXLAN, NVGRE or MPLS over GRE/IP
- Not all forwarding elements are controlled by the controller (e.g. virtual elements only or a combination between virtual and physical)



Thank You

ERLEBEN, WAS VERBINDET.