Flexible Service Chaining. Requirements and Architectures.

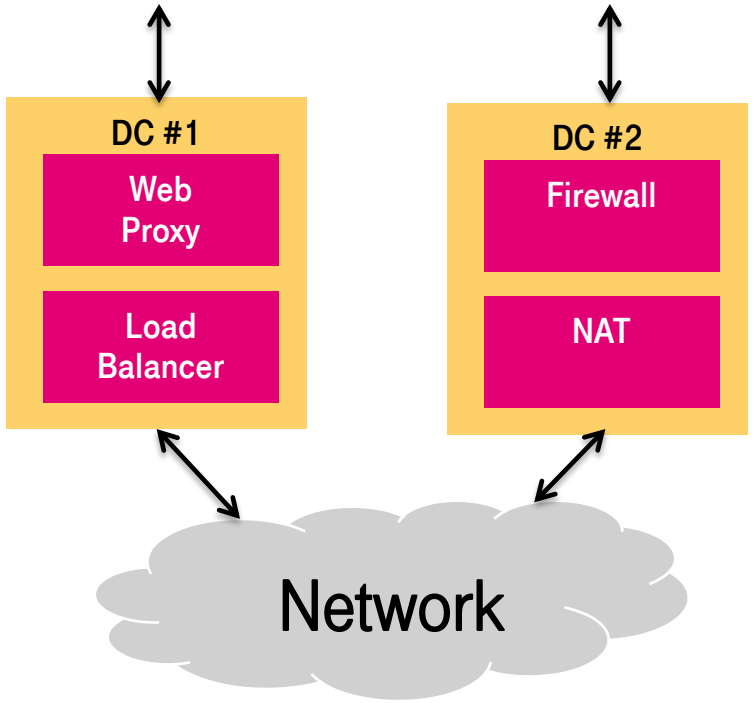
Nicolai Leymann, Deutsche Telekom AG

Agenda

1. Overview Service Chaining
2. Network Function Virtualization and Service Chaining
3. Use Cases and Existing Implementations
4. Architecture
5. Summary

Flexible Service Chaining.
Overview.

Service Chaining

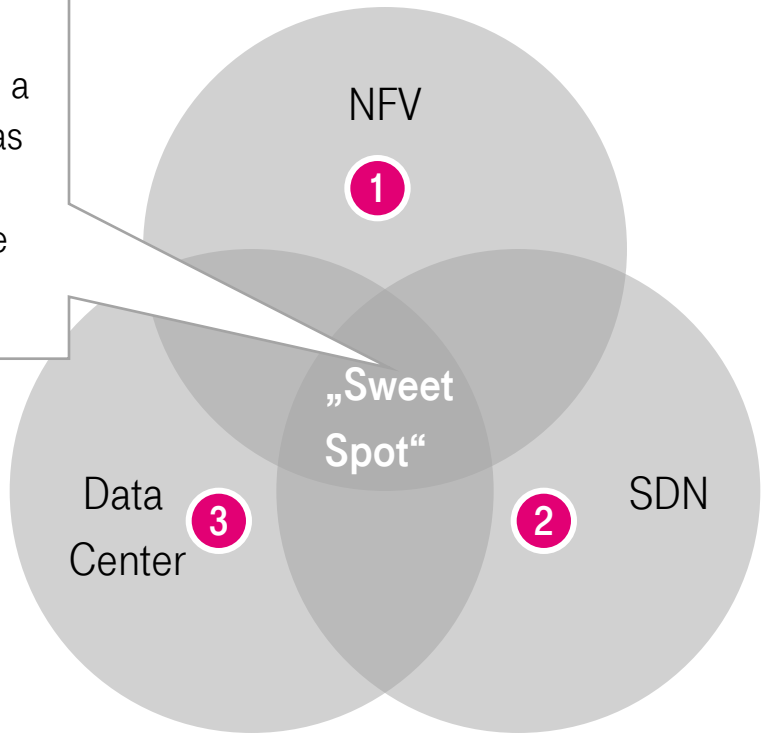


Remarks

- Generic Term, used by many standardization bodies.
- Consists of a “transport part” to connect “service bringing functions” over one or more existing networks
- Not necessarily related to virtualization (assumption: not all components of a chain are virtualized, depending on requirements).
- Chain can be distributed (e.g. over several data centers, network components, ...)

Flexible Service Chaining.
Bringing Bits & Pieces 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)

Data Center 3

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

SDN 2

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

Flexible 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.)

Flexible 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.

Flexible Service Chaining.
Requirements.

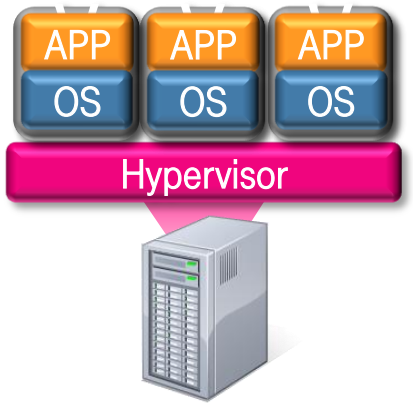
Requirements for Flexible and Elastic Service Chains

- 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

Flexible 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

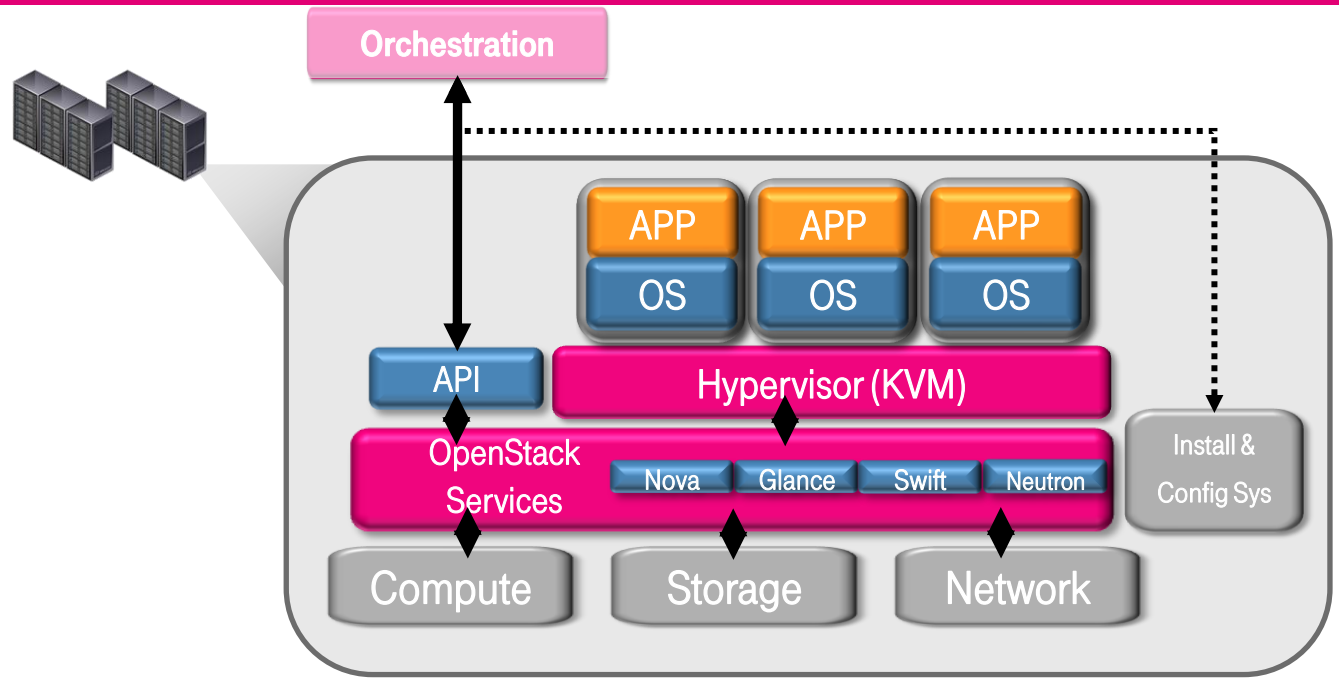
- Abstraction: decouples applications from underlying infrastructure
- Flexibility: flexible resource allocation and management
- Sustainability: enables reduction of energy costs and environmental footprint
- Automation: offers new ways for operational processes

Challenges

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

Flexible Service Chaining.
Data Center Architecture (2 of 2).

Generic Data Center Architecture

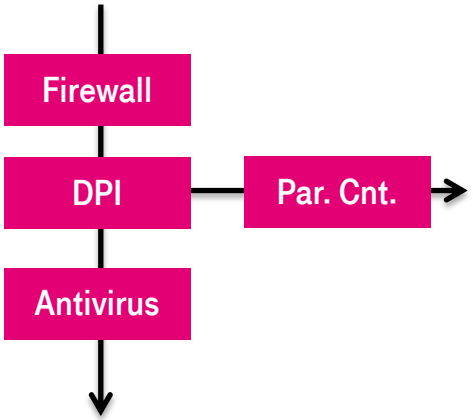


- Orchestration Layer for flexible and dynamic resource provisioning
- Based on standard north and southbound interfaces
- Compute, Storage and Network resources provided “on Demand”

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

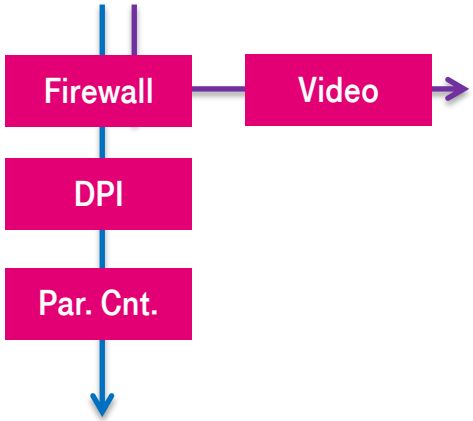
Flexible Service Chaining.
Chaining Scenarios.

Service Path Forking



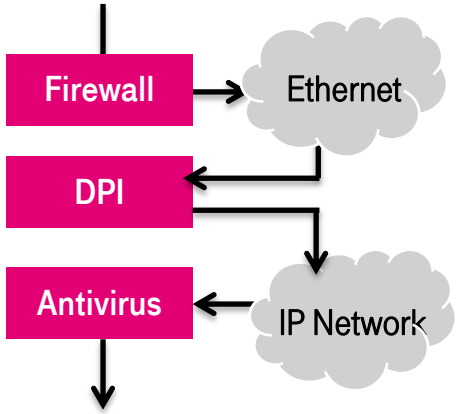
- A single flow being forked within one component of the Service Chain

Service Function Sharing



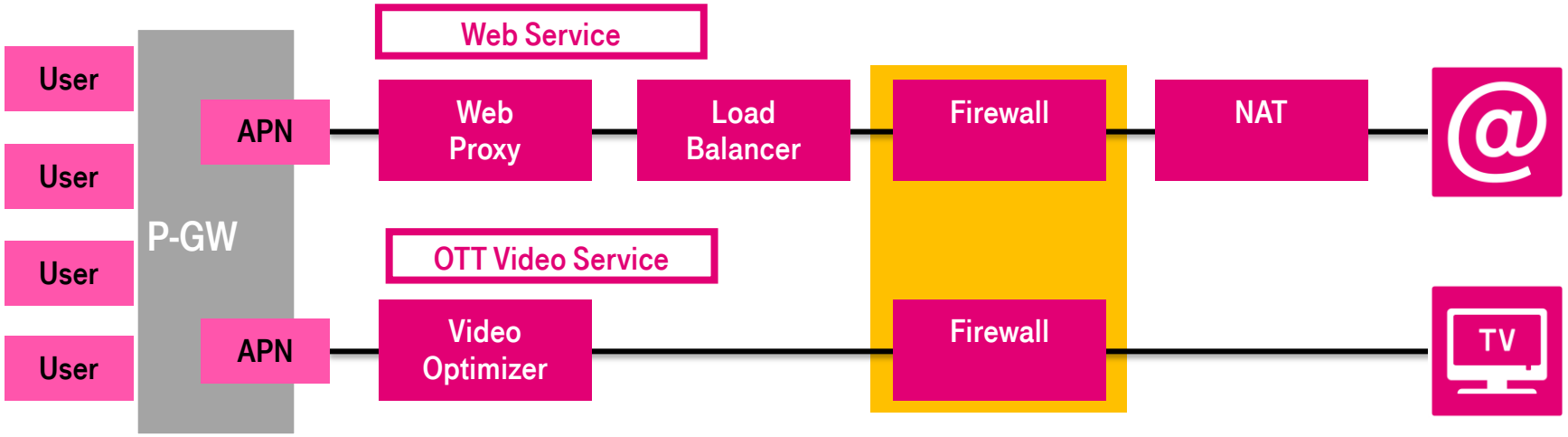
- Parts of the Service Chain are being reused for other Service Chains

Multiple Underlay Networks



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

Flexible Service Chaining.
Use Cases: GI-LAN (today).

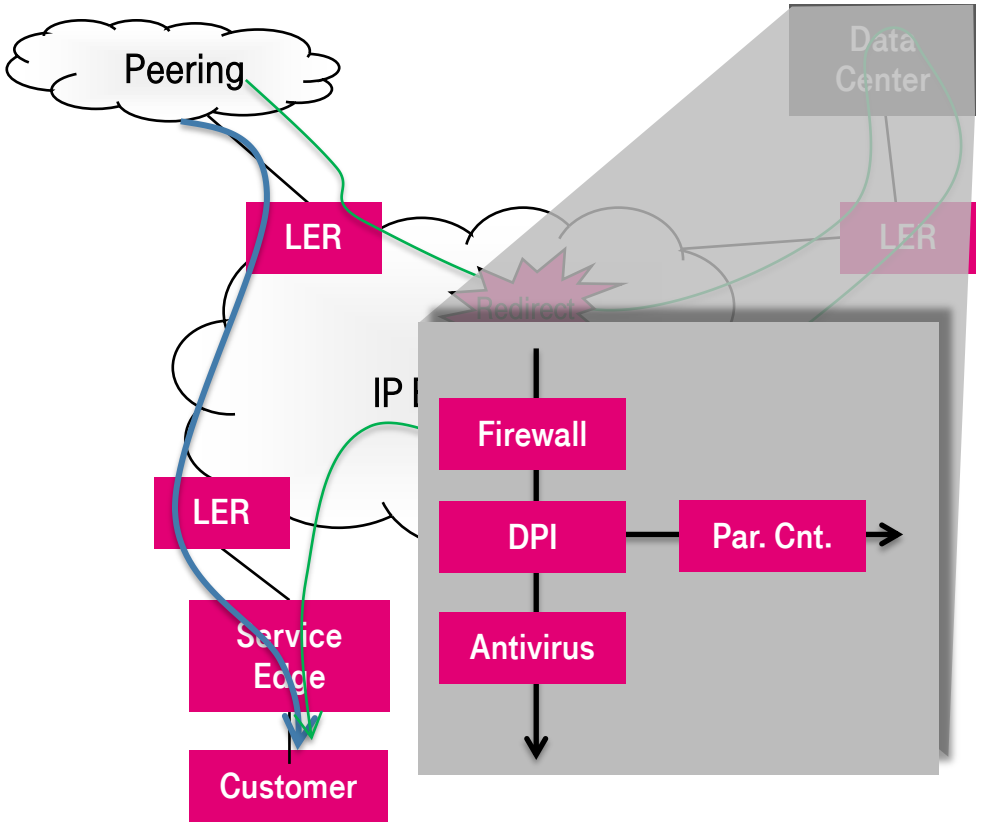


- With deployment of additional value-added services increasing number of functions required in Gi-LAN. Some functions in dedicated devices, sometimes multiple functions in one box.
- Due to fast service introduction cycles service chains emerge, growth & change evolutionary.
- Very often static IP links, 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.

Flexible Service Chaining.

Use Cases: Traffic Steering and Service Chaining.

Dynamic Traffic Steering and Service Chaining

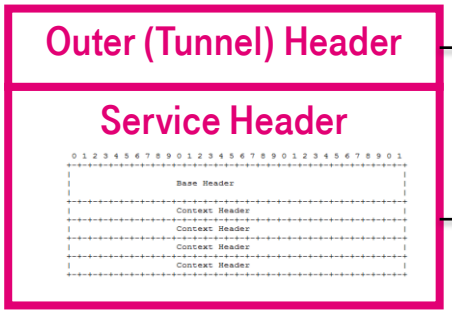


Remarks

- 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

Flexible Service Chaining.
Architecture Options: Service Header

Service Header



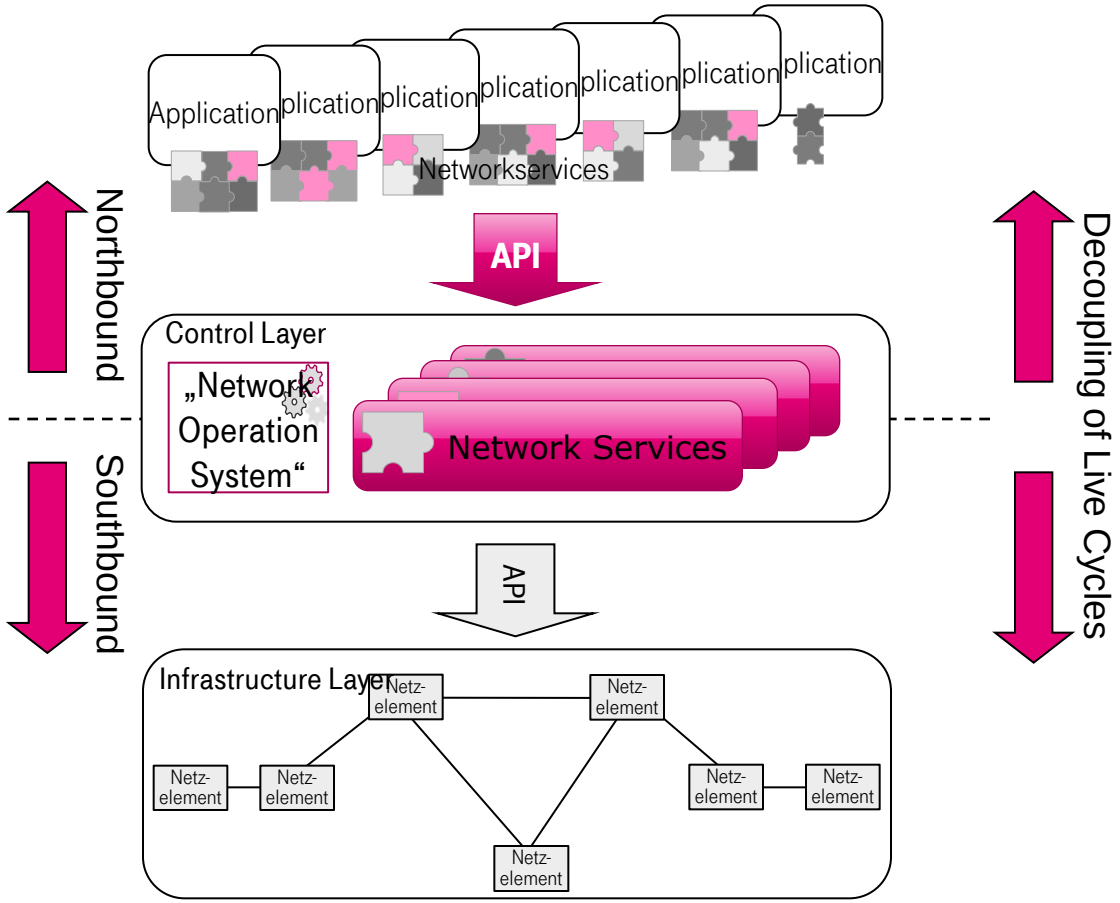
- Used to build an overlay network in order to address components of the service
- Independent of underlying transport network

- 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

Flexible 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

Flexible 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, ...)

Thank You.

ERLEBEN, WAS VERBINDET.