Orchestrating a brighter world

NEC

Network Programmability
Research Topics in SDN & NFV

Dirk Kutscher
Fabian Schneider

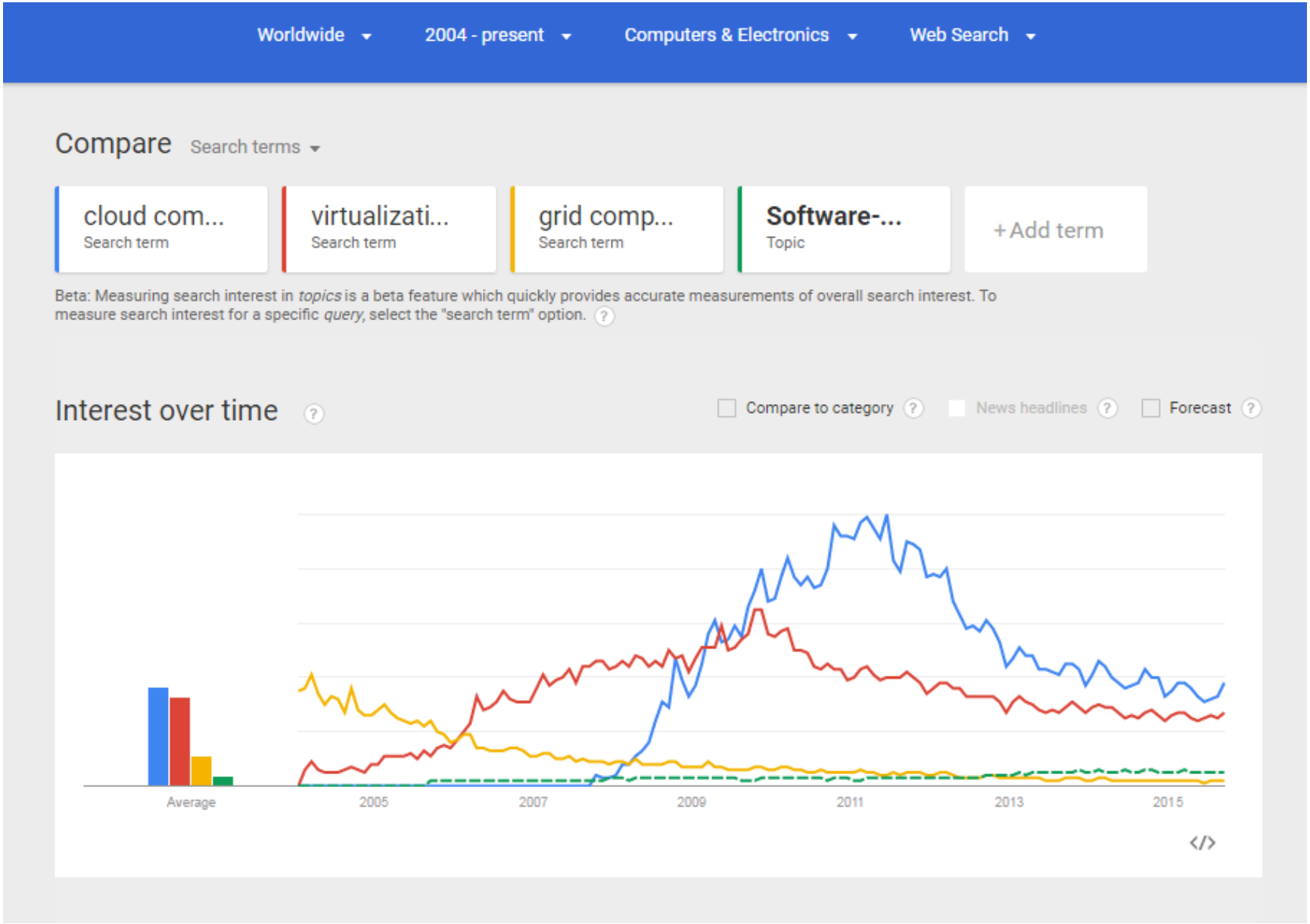
NEC Laboratories Europe

NEC Laboratories Europe (NLE) - Overview

- **~100 leading researchers from all over Europe and world-wide in Heidelberg**, and London (NEC E HQ)
- **Close links with leading European research institutes & universities**
- **Collaboration with major industry in Europe**,
  eg. network operators, ICT vendors, automotive, utilities….
- **Research areas in NLE**
  - 4G/5G, **Future Internet, SDN**
  - **Information-Centric Networking**
  - **Cloud platform**, management & services
  - Security, Privacy & Performance
  - Internet of Things (M2M)
  - ITS and Green Telematics
  - Smart Energy
  - **Standardization**
- **NEC Laboratories Europe**
  - Kurfürstenanlage 36
  - D 69115 Heidelberg
  - *www.neclab.eu*

Programmability Intro

Google Zeitgeist on Cloud Computing Keywords



Technology Trends

Softwarization and programmability

- Move more and more functions from HW to SW to increase agility…

Commoditization

- initially: servers
  currently: L2/L3 switches
  next: base stations?

  → Whiteboxes (“bare metal switches”) based purely on merchant silicon

Cloudification and automation

- Cloud is not about virtualization. It is about hyper-scale, elasticity, resource pooling, … but most of all AUTOMATION

Classical Network Appliance Approach

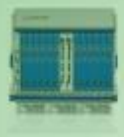
Message Router

CDN

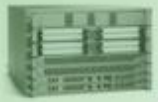
Session Border Controller

WAN Acceleration

DPI



Firewall



Carrier Grade NAT

Tester/QoE monitor

SGSN/GGSN

PE Router

BRAS

Radio Access Network Nodes

- Fragmented non-commodity hardware.
- Physical install per appliance per site.
- Hardware development large barrier to entry for new vendors, constraining innovation & competition.

Orchestrated, automatic & remote install.

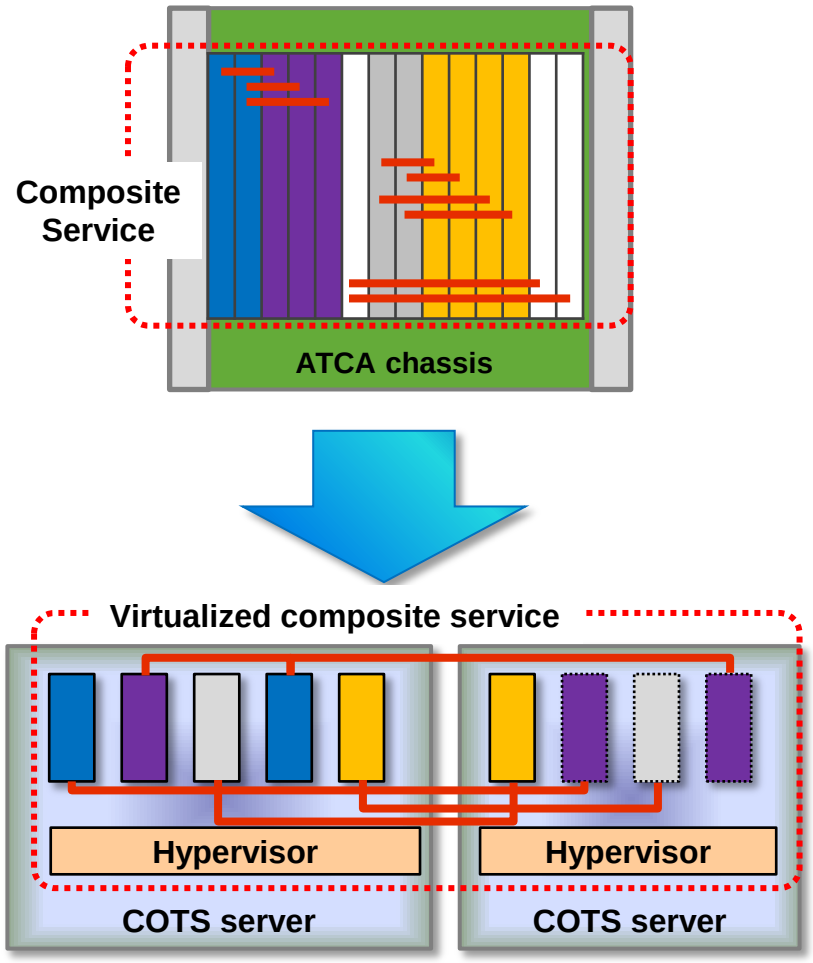
Standard High Volume Servers

Standard High Volume Storage

Standard High Volume Ethernet Switches

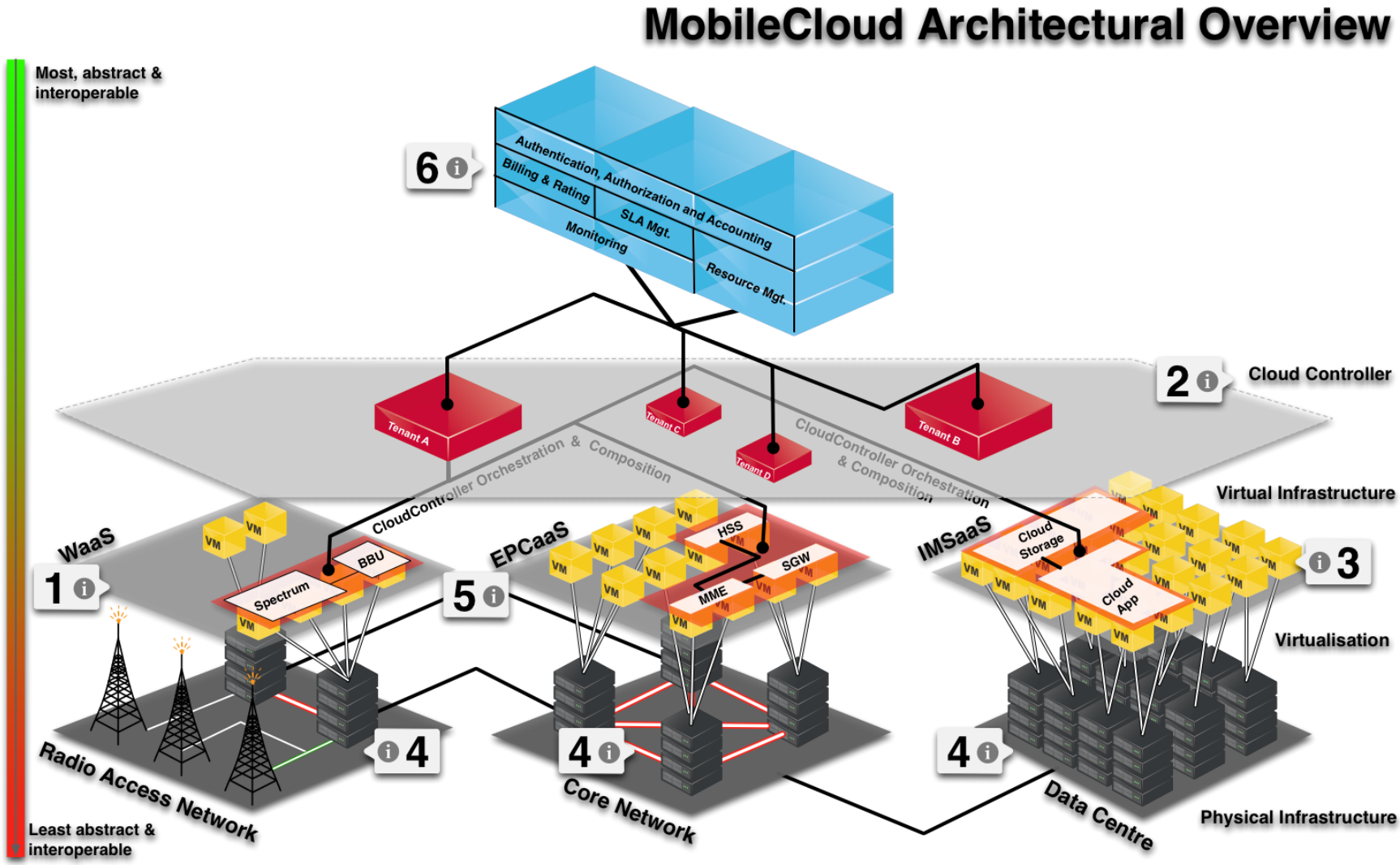
Network Virtualisation Approach

Telco Infrastructure Softwarization

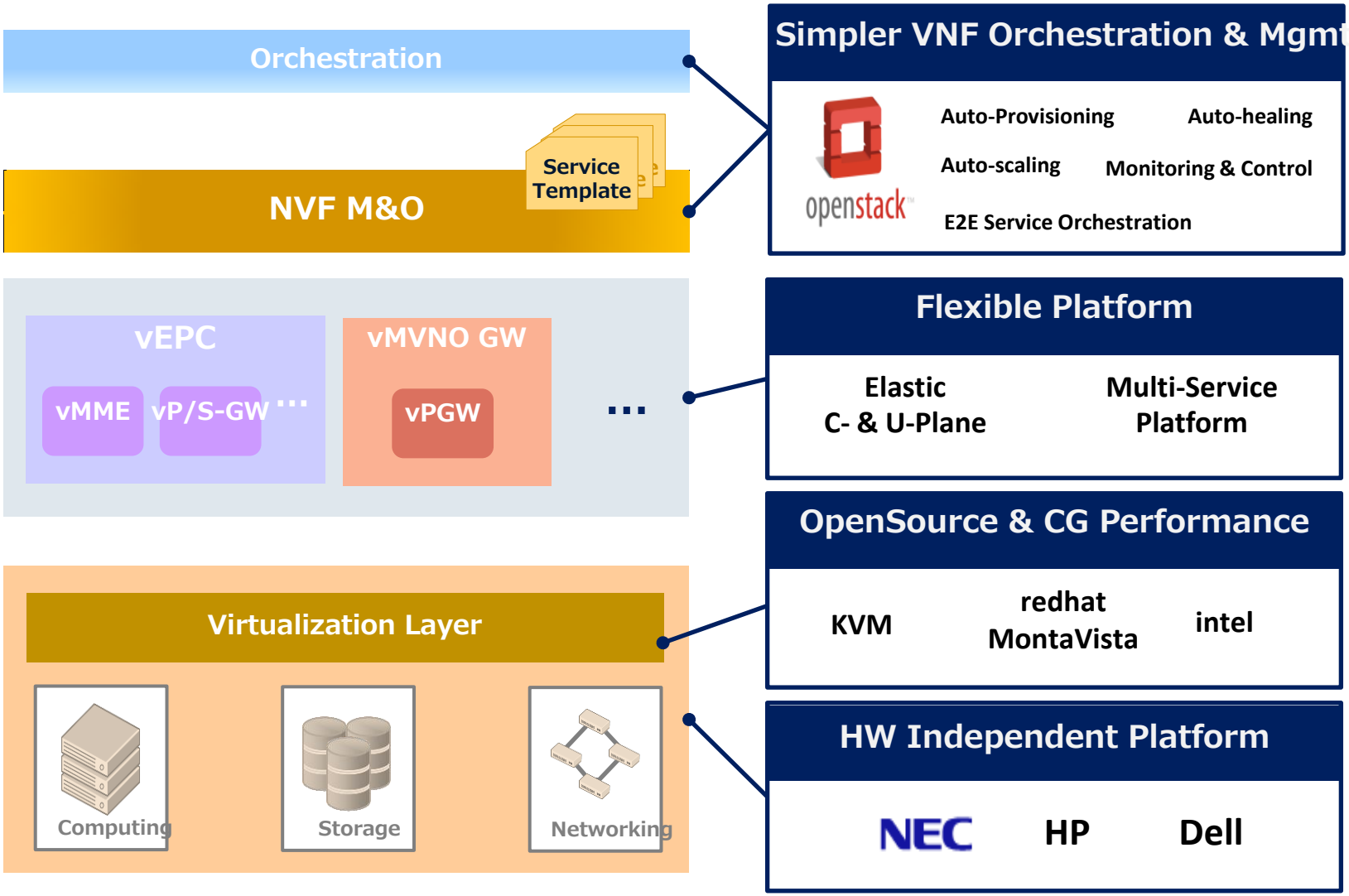


Transformation from ATCA to VNF

Example



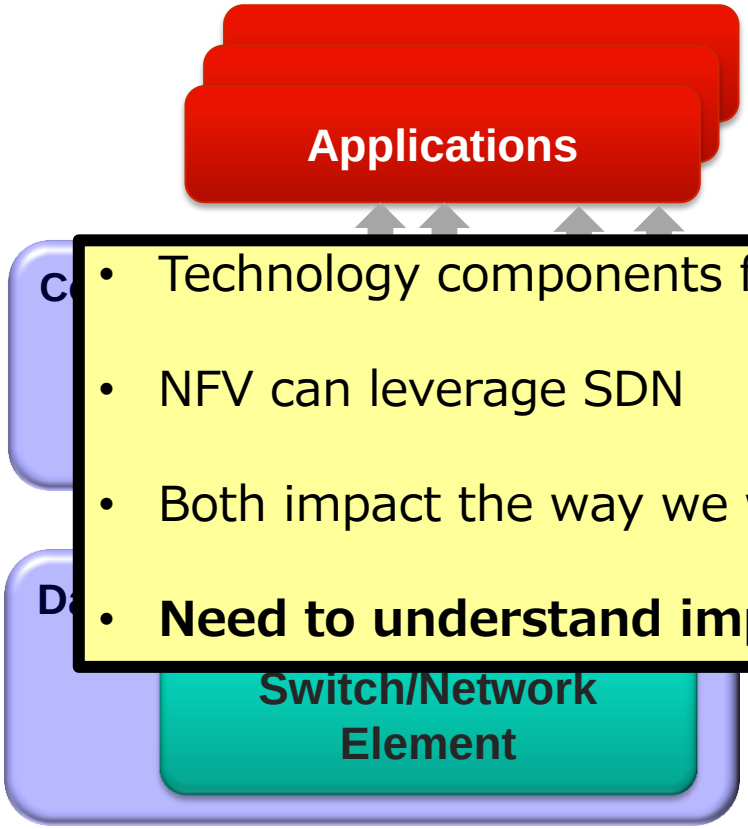
Implementation: NEC SDN/NFV

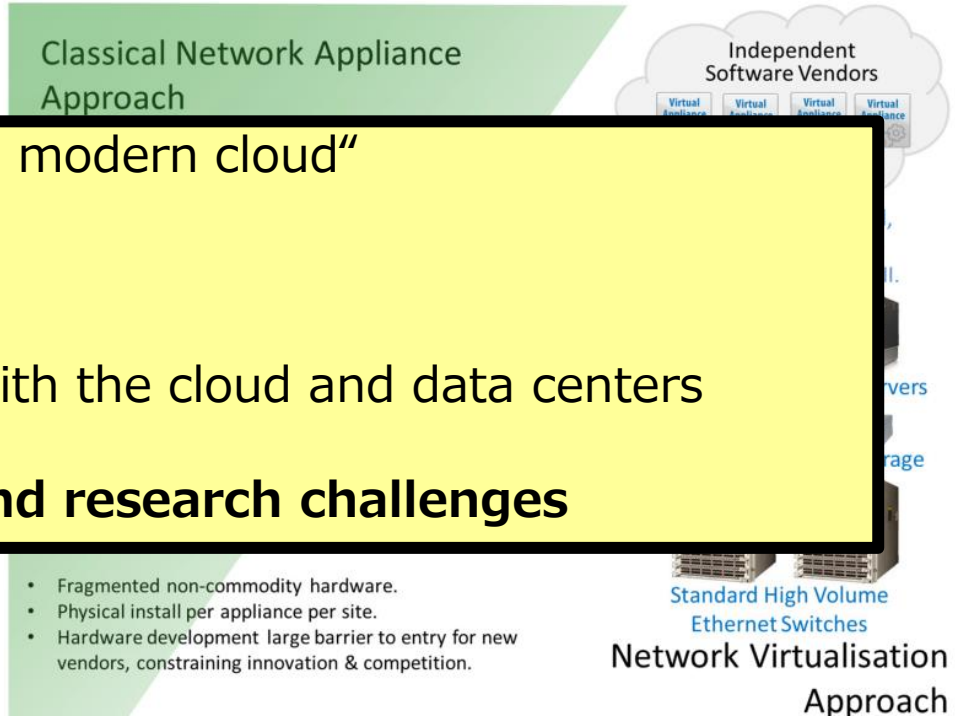


Two Main Technology Components

Software-Defined Networking

Network Functions Virtualization



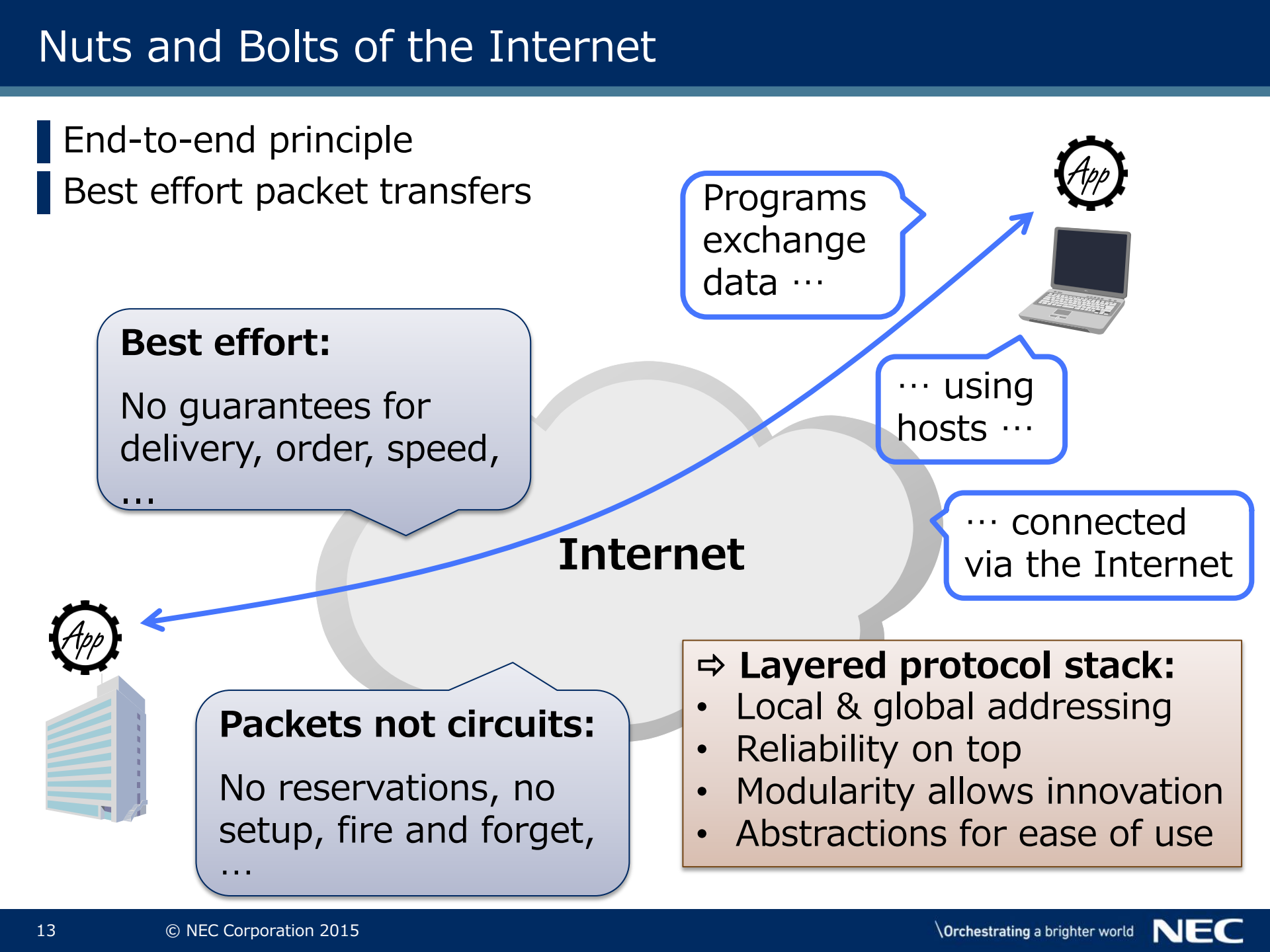


- Technology components for „the modern cloud“
- NFV can leverage SDN
- Both impact the way we work with the cloud and data centers
- **Need to understand impact and research challenges**

Software-Defined Networking

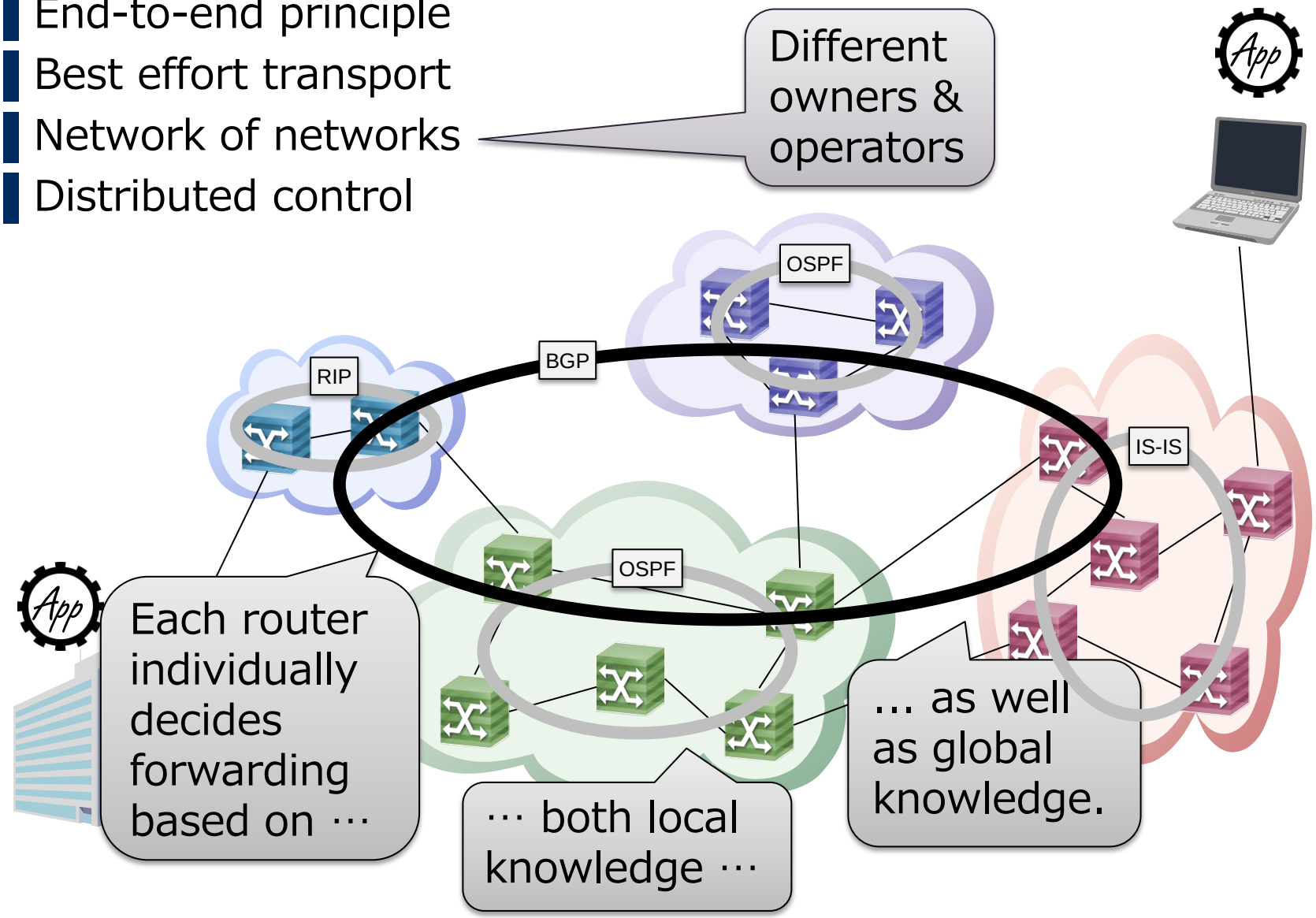
Nuts and Bolts of the Internet

- End-to-end principle
- Best effort packet transfers



Nuts and Bolts of the Internet

- End-to-end principle
- Best effort transport
- Network of networks
- Distributed control



Pre-SDN State of Networking

Data plane and hosts

- Built on under-specification
  - **Best-effort packet** delivery
  - OSes provide open API to network stack
- **Fast innovation** & adoption
  - Web, P2P, VoIP, OSN, ...
  - Ethernet, optical, WiFi, 3-4-5G
- Huge **commercial success**
  - Internet started as research experiment, now:
  - Yahoo, Google, Facebook, Apple, Microsoft, Amazon, ...

Inside the network

- **Distributed** self-adapting network control plane
- Closed switches/routers
  - Bundled with control SW
  - **Proprietary interfaces**
- Huge amounts of protocols
  - Slow to standardize
- **Slow innovation**
  - Vendors write the code
  - New features need to yield short-term business

Time Warp to 2006 @ Stanford University

Nick McKeown
CS Professor

"We need a security architecture for enterprise networks"

"Ok, … so ... What is the programming interface to the network?"

"Well, you can change parameters in network control protocols!"

"Let's build a programming interface to forwarding."

OpenFlow

Martin Casado
CS Student &
SW Engineer

SANE
Usenix Sec '06

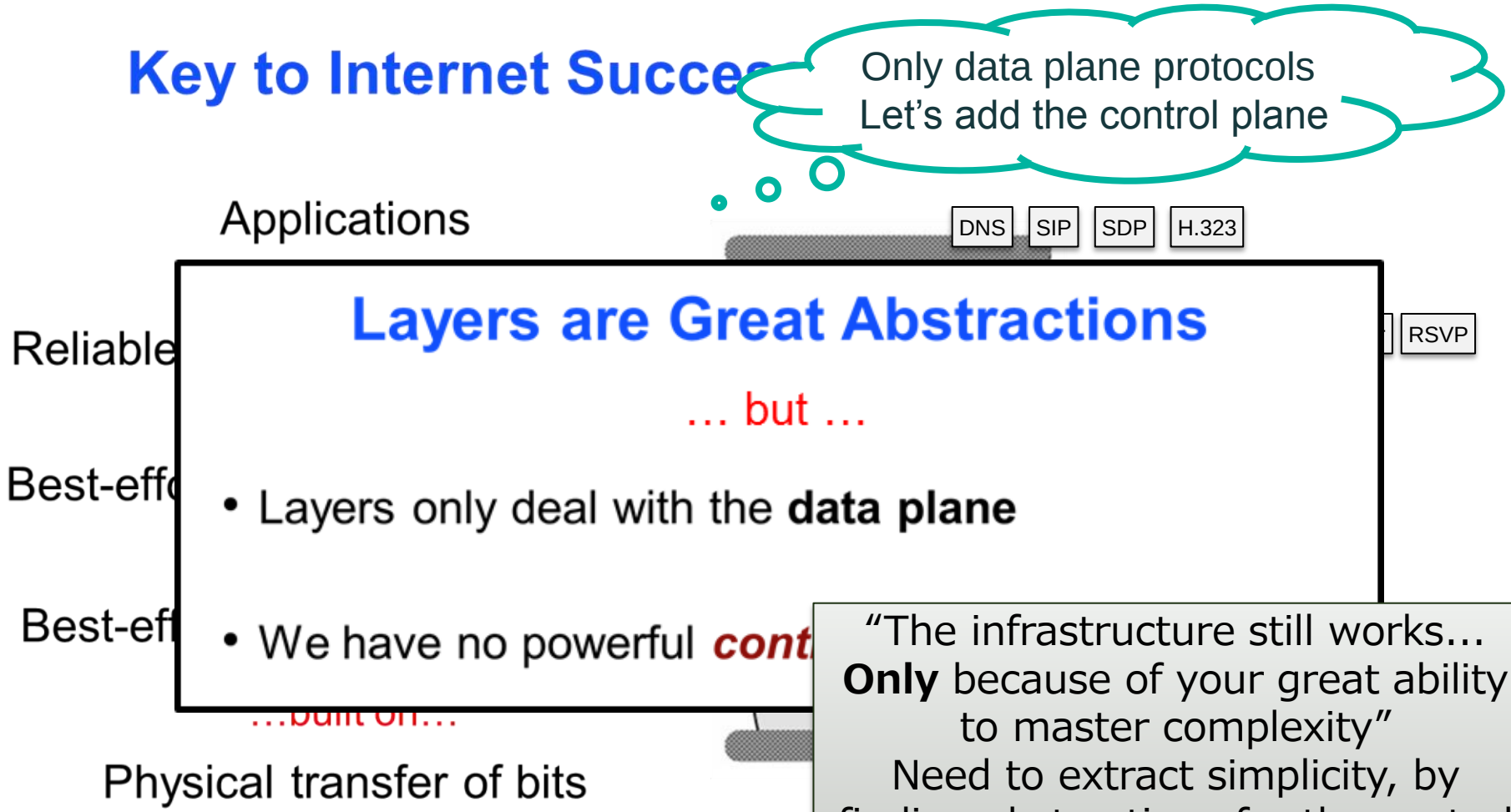
Ethane
SIGCOMM '07

OpenFlow
CCR '08

NOX
CCR '08

OVS
HotNets '09

But wait! Protocols enabled Internet's success …



"The infrastructure still works… **Only** because of your great ability to master complexity"
Need to extract simplicity, by finding abstractions for the control plane!

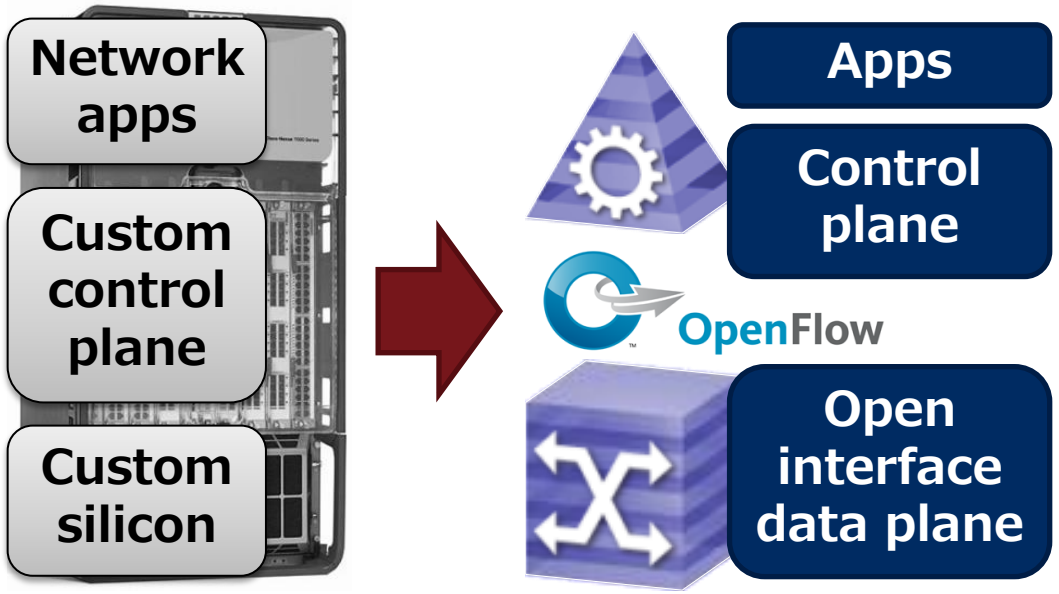
Scott Shenker

Borrowed from Scott Shenkers talk "The Future of Networking, and the Past of Protocols" @ ONS'11
http://opennetsummit.org/archives/oct11/shenker-tue.pdf

OpenFlow in a Nutshell

- Provide open protocol/interface to separate control from data plane
- Rules (Match/Action tuples + priority + counters)
  - **Match**: Select traffic based on packet header fields (e.g. L2 MACs, IP, TCP/UDP, VLAN, MPLS, …)
  - **Actions**: Define what to do with selected traffic (e.g. drop, forward to port, rewrite header, to controller)

- Support for quick & dynamic network configuration
- Latest specs:
  - OF 1.0.2, OF 1.3.5, OF 1.5.1
  - OF-Config 1.2.0
  - OF Test 1.0.1



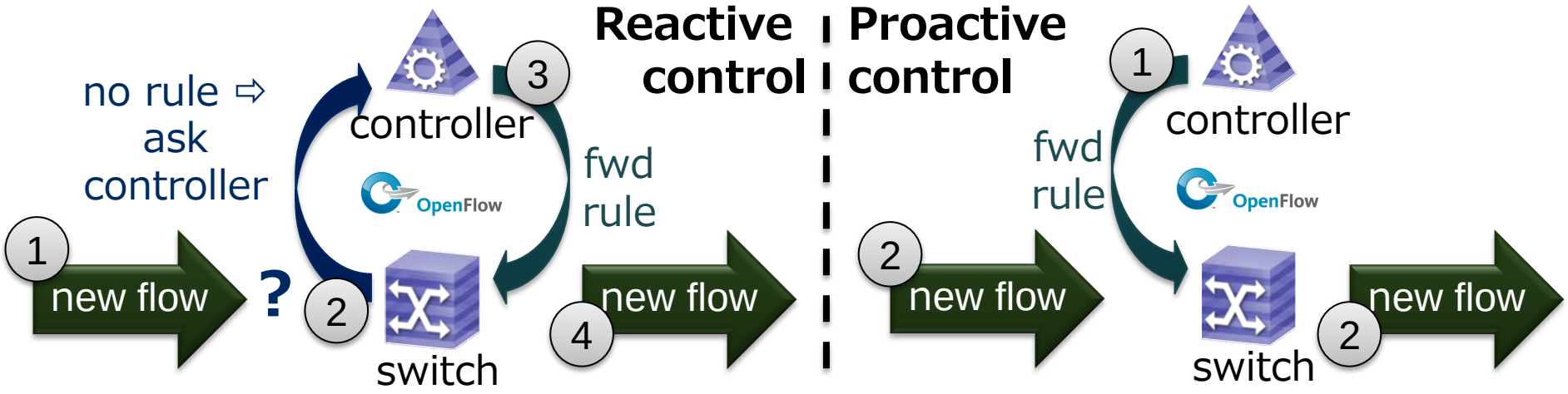
Clarifications on OpenFlow

#1: Flow granularity is not predefined!

- **No restriction** to often mentioned
  - 3-tuple (IP addresses and transport protocol) or
  - 5-tuple (adding in port numbers)
- Allows a **choice of granularity**
  - Matching on any selection of supported match fields possible

#2: **No limitation** to “reactive” control;

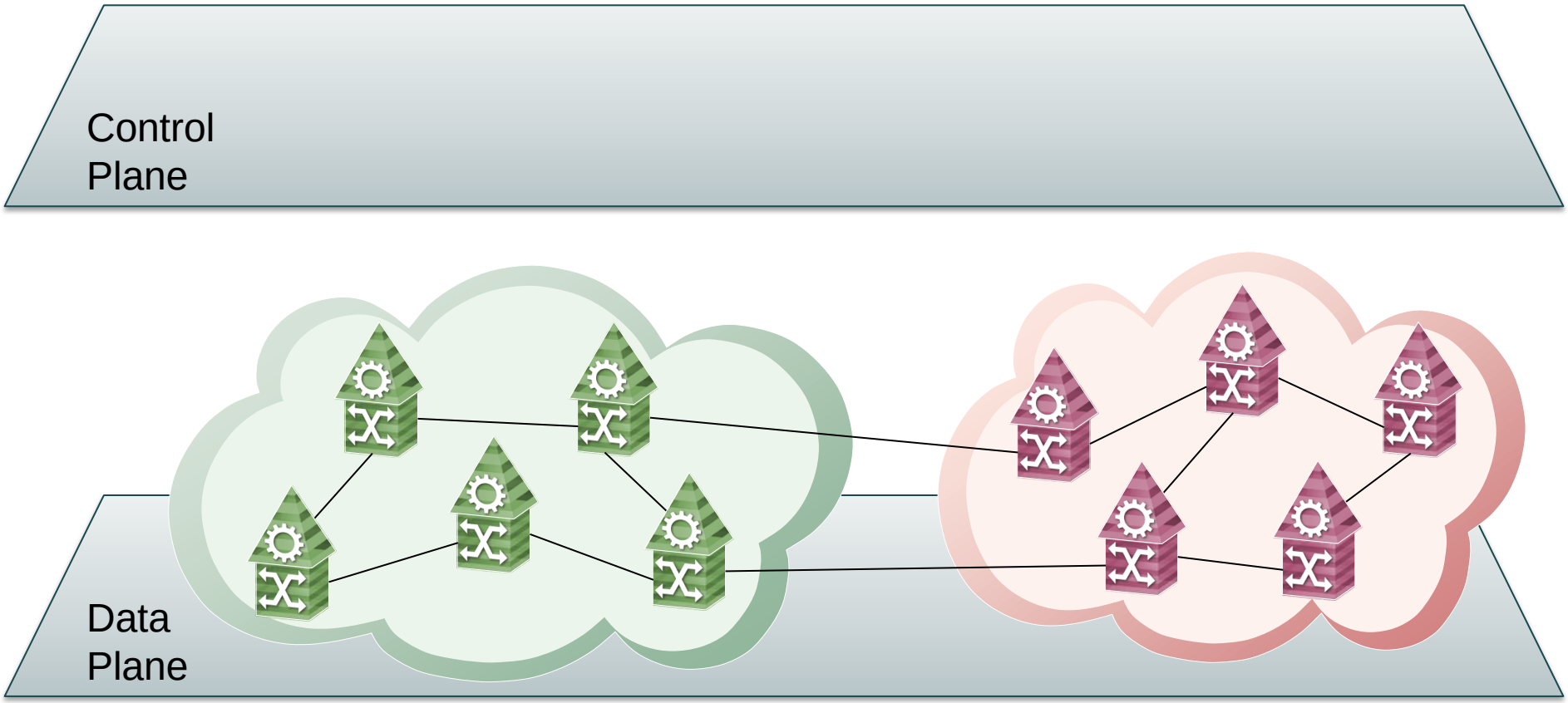
- “Proactive” equally possible



SDN Principle #1: Decouple Control and Data Plane

Control logic, routing, algorithms

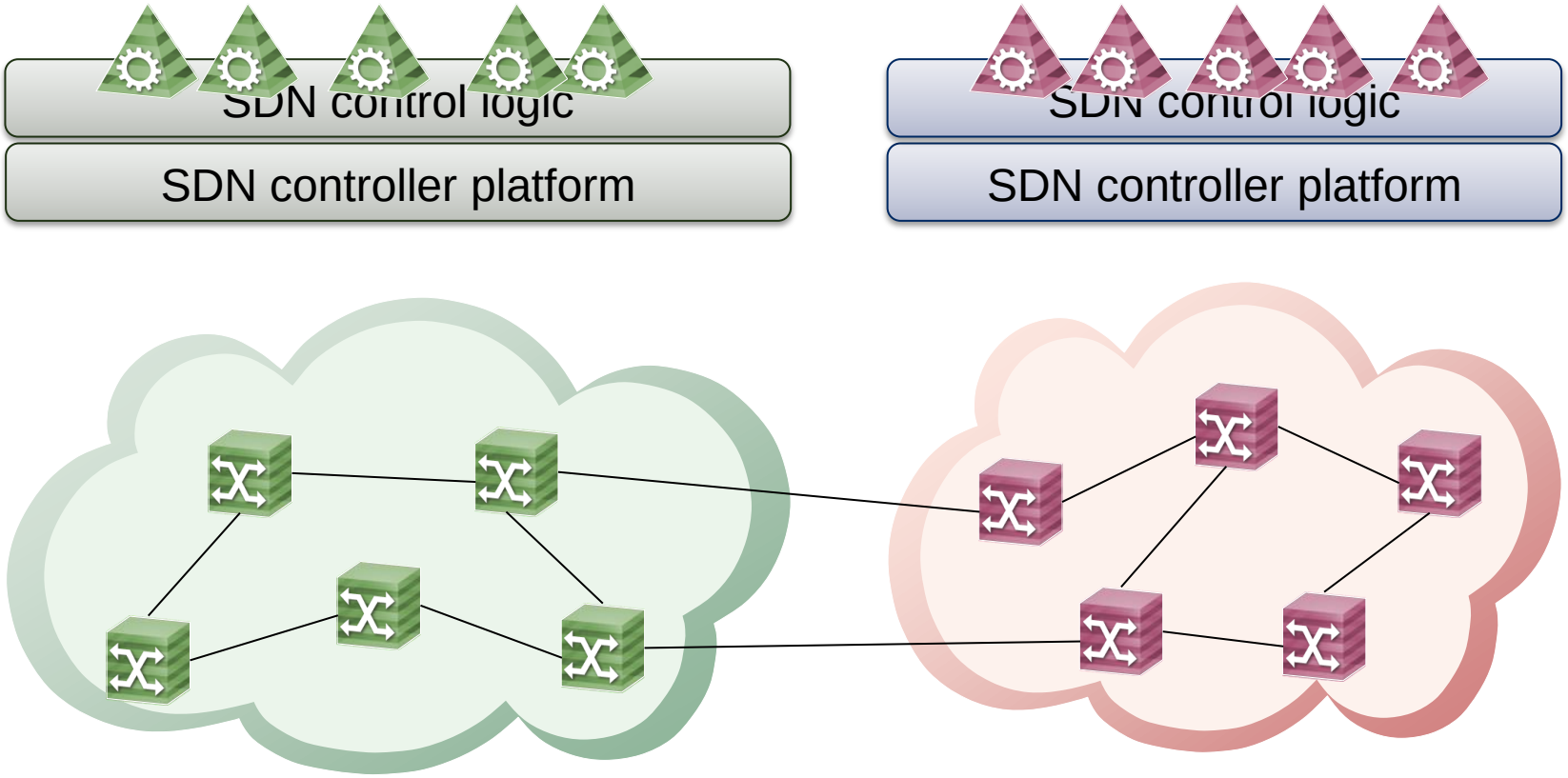
Packet forwarding & processing



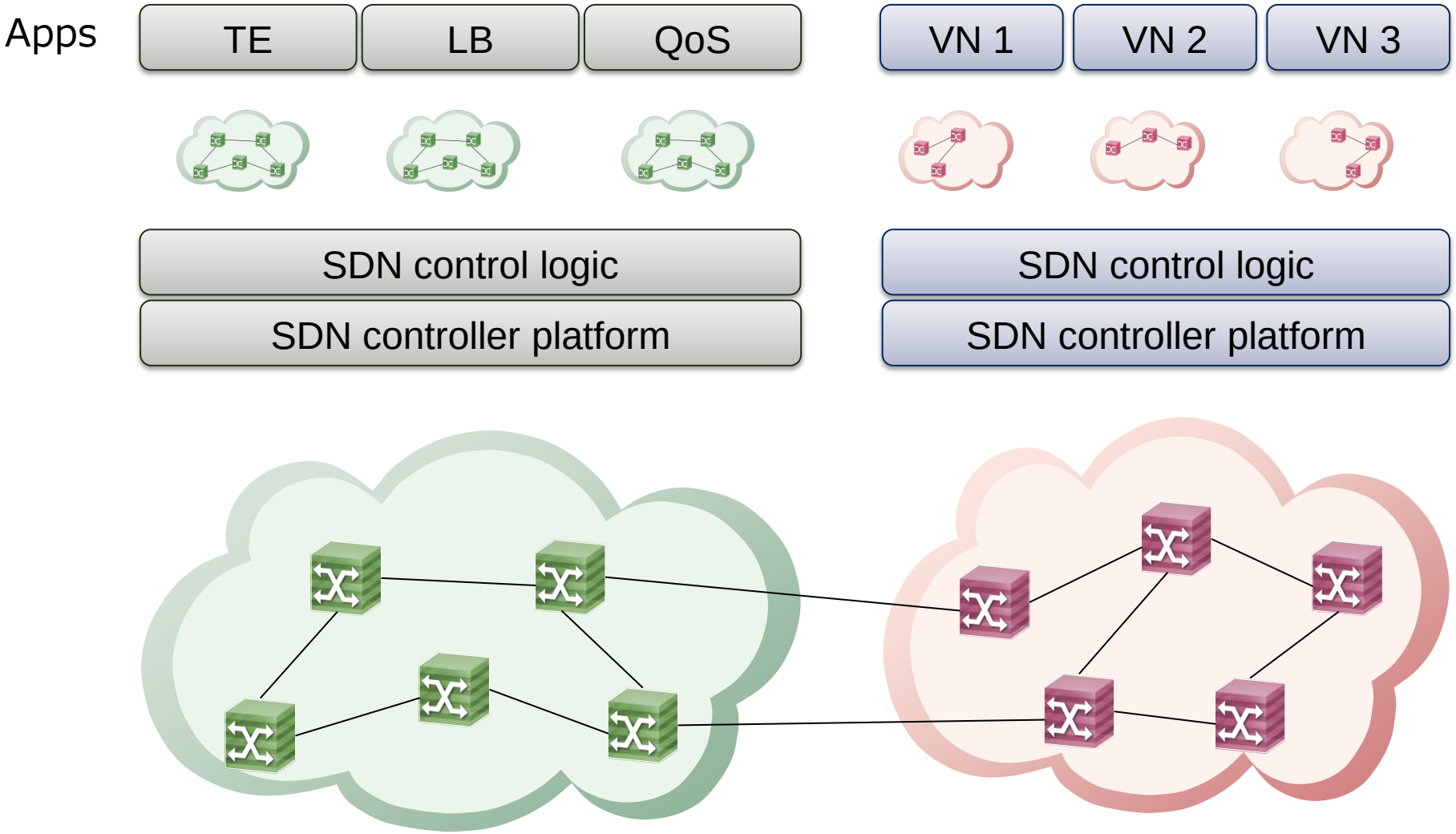
SDN Principle #2: Logically Centralized Controller

Control logic, routing, algorithms

Packet forwarding & processing



SDN Principle #3: Expose Network to Applications

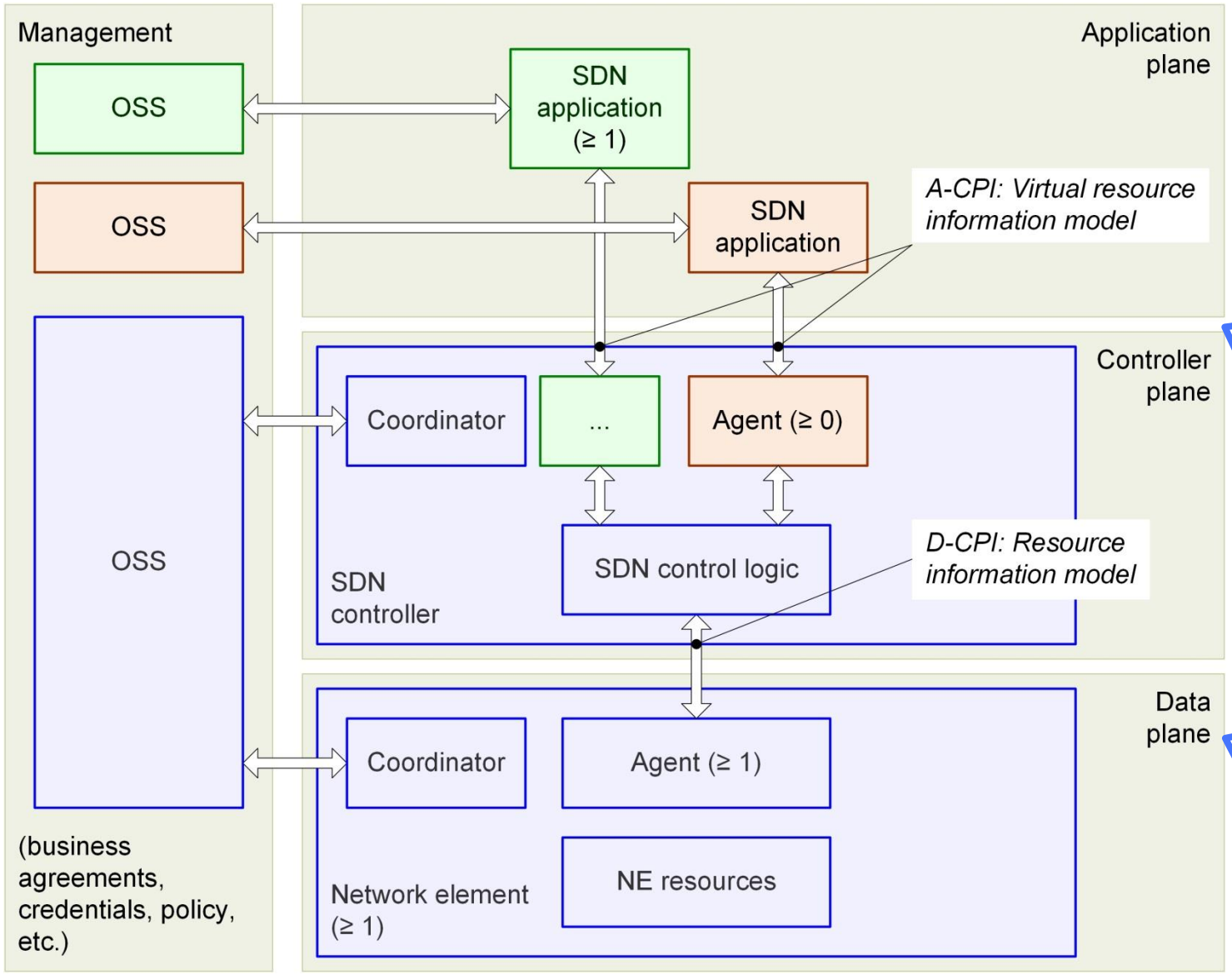


Principles of SDN

ONF TR-504

1. Decoupling of controller and data plane
2. Logically centralized control
3. Expose abstract network resources to external applications

SDN Architecture



ONF TR-504

ONF TR-502

A-CPI: Application-Controller Plane IF

Often called: Northbound IF

D-CPI: Data-Controller Plane IF

Often called: Southbound IF

Important Properties of SDN Architecture

Apps are network aware: **SDN-enabled Apps**

- Communicate their requirements/polices to the network
- Can monitor network state and adapt accordingly

Logically centralized network control: **SDN Network Controller**

- Controller translates from app requirements to low-level rules
- Controller summarizes the network state for apps

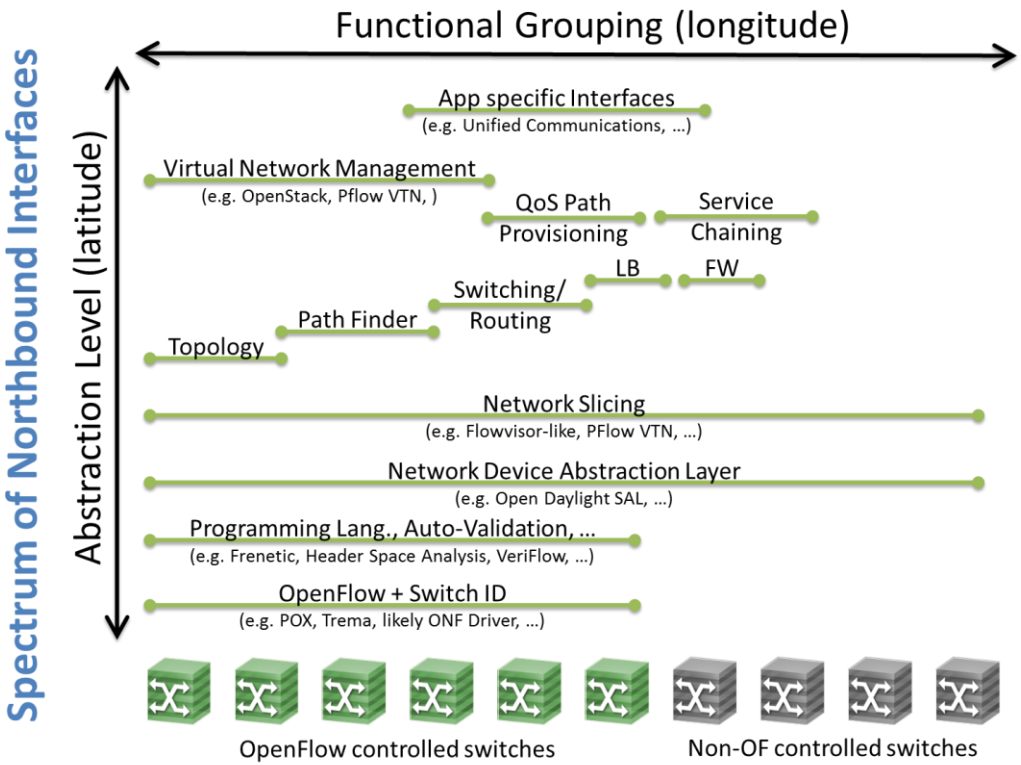
Well-understood driver-like model for devices: **SDN Datapath**

- Programmatic low-level control of all forwarding and configuration
- API for capabilities advertisement and publishing stats
- **No resource contention** with other entities
  - Controller “owns” this device, subject to capabilities advertisement

Defining Northbound Interfaces (NBIs)

Not an easy task

- Level of **abstraction unclear**
  - Bottom: OF+SwitchIDs (e.g. Trema, NOX/POX)
  - Middle: network prog. lang (e.g. Frenetic)
  - Top: Neutron/Quantum level
- **Scope unclear**
  - Single NBI to rule them all
  - Or one per operation call



Current trend: **Intent**

- „Finance staff needs priority access to SAP servers“
- Finance = Set of ports; SAP server = Set of IPs; Priority = reserved BW
- Include conflict detection and conflict resolution in design

OpenFlow ++ ?

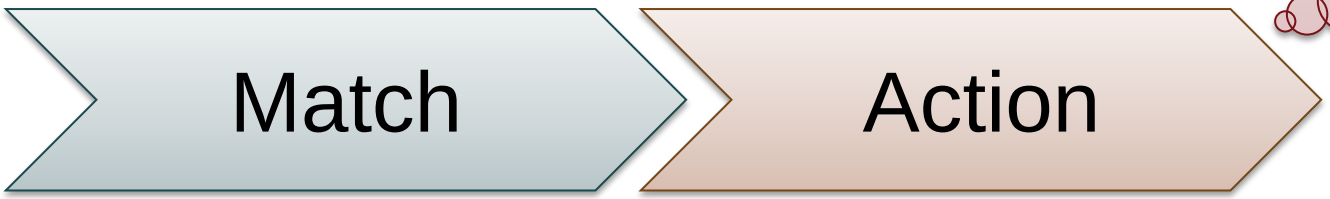
Today:

Limited to existing data plane protocols

Today:

Limited to typical packet forwarding

Is Match-Action enough ?



Next: PIF
protocol independent forwarding

User-defined protocol headers

Next: OpenState

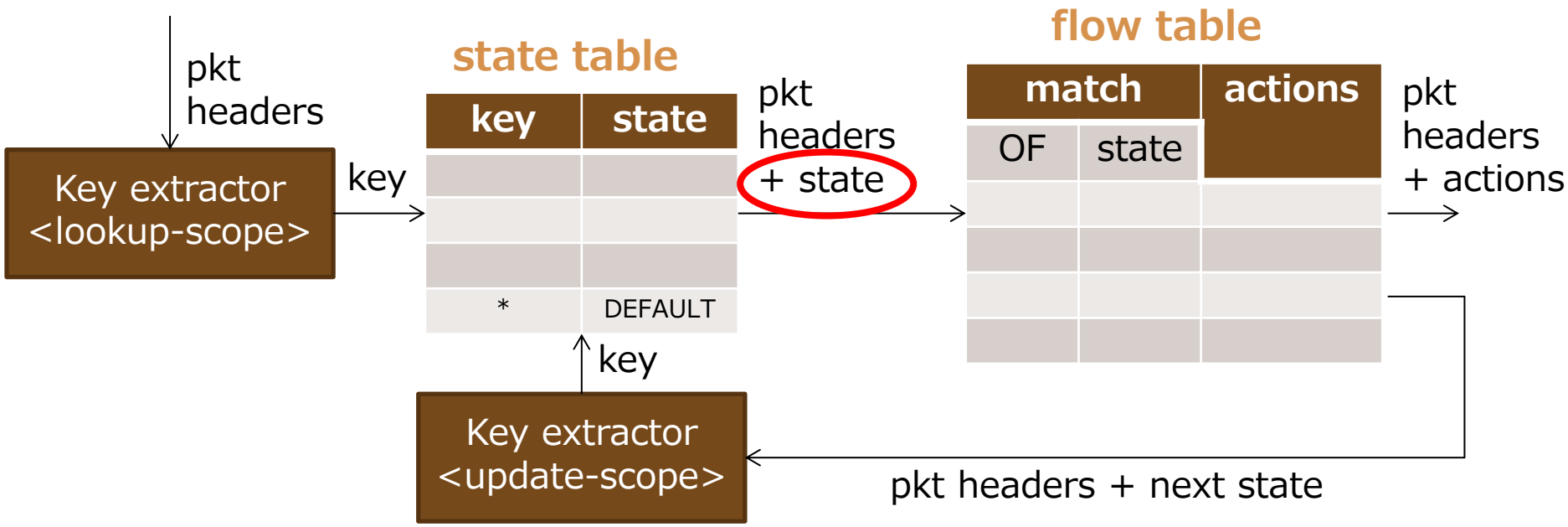
Stateful matching

Next: In-switch processing

Autonomous Functions, Reply packets

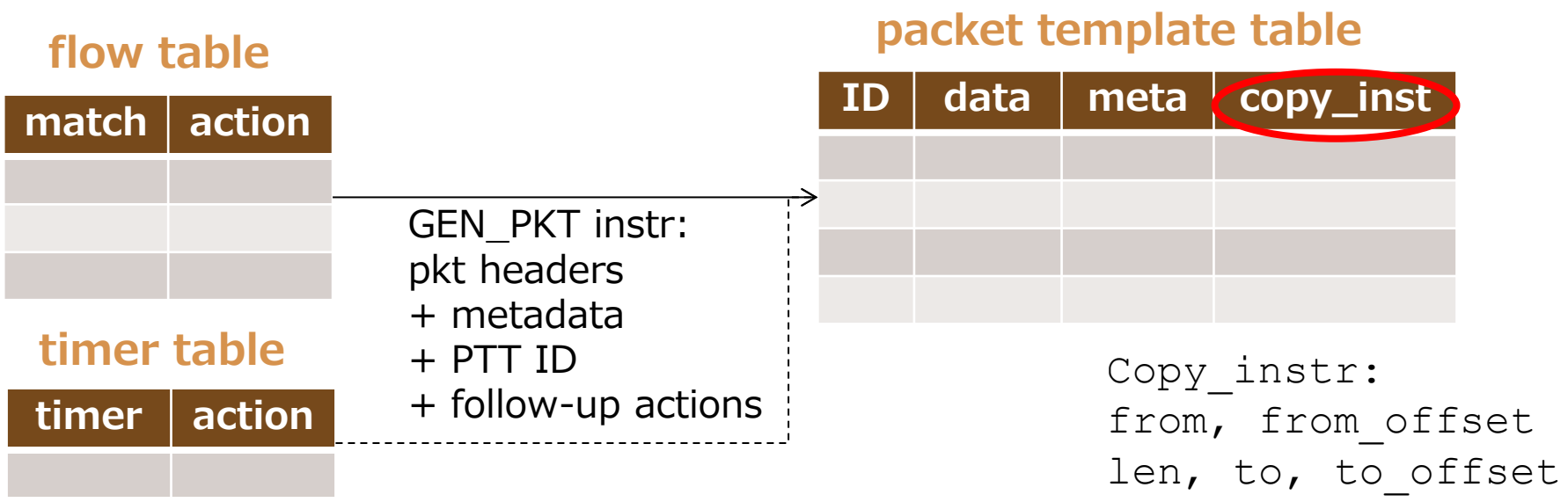
OpenState: Stateful flow processing

- New “state table” type: exact match, DEFAULT if miss
- Move state in packet metadata to flow table
- Key extractor: full headers ⇨ selected bit sequence
- Actions can include state updates
- Update scope can be different from lookup scope



PacketTemplates: In-switch packet generation

- New “packet template table” type:
  - Template has byte array of packet contents and metadata
- Copy instructions: Program how to change template
  - Data source can be a received packet or a table (e.g. state table)
- New instruction to refer to packet from flow table
  - Additional action specify what to do with the generated packet
- Option to use timers as triggers



Summary

Initial goals

- Make network forwarding „**programmable**“
- Find good **abstractions** for network control

SDN principles

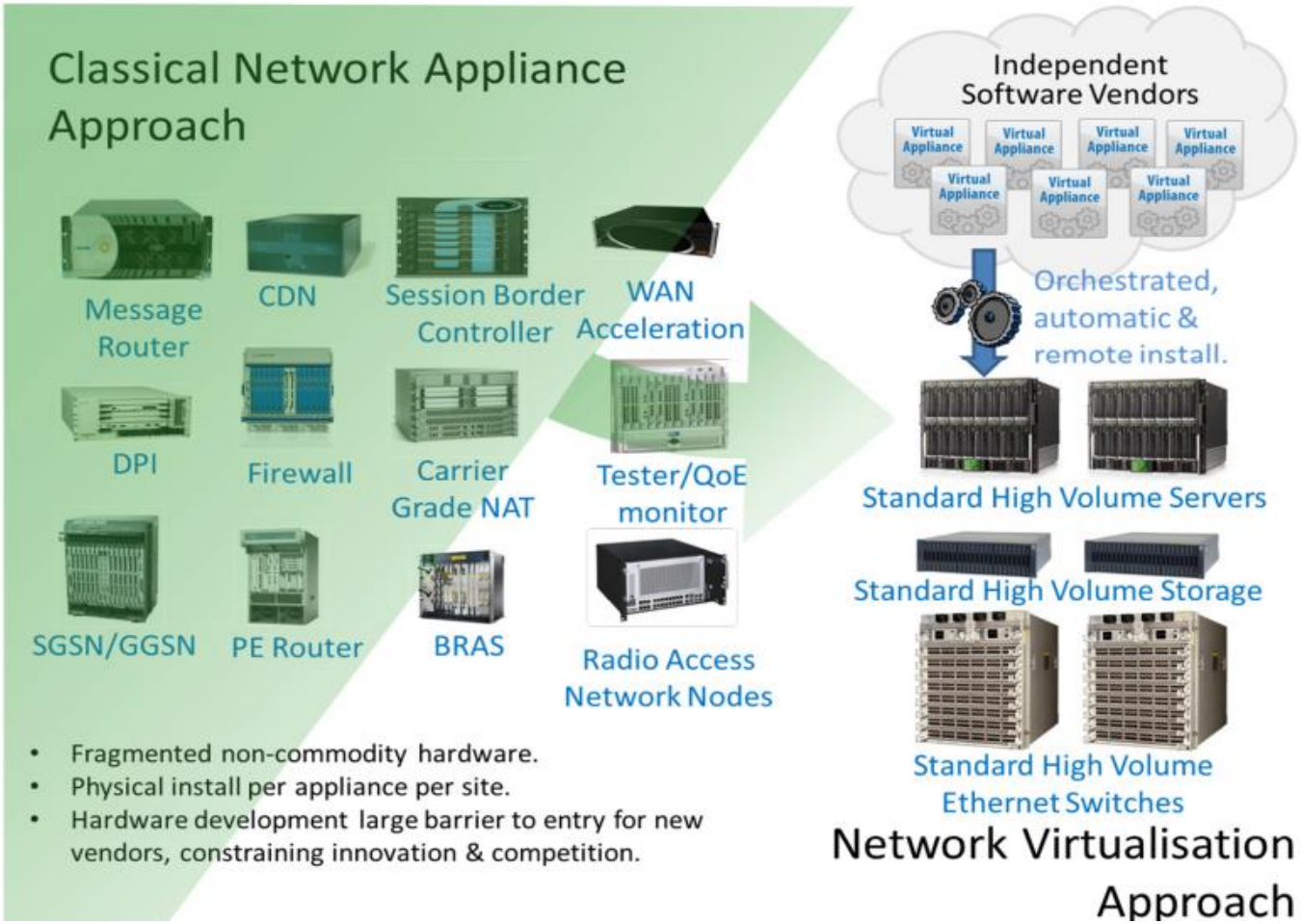
- **Decouple** data and control plane
- Logically centralize control
- Expose network **to applications**

Current Hot Topics

- Develop **higher-level abstractions** than OpenFlow
- **Extend programmability** beyond legacy forwarding
- Better approaches to **integrate network and VM control**

Network Functions Virtualization

NFV = Network Functions Virtualization



NFV = Network Functions Virtualization

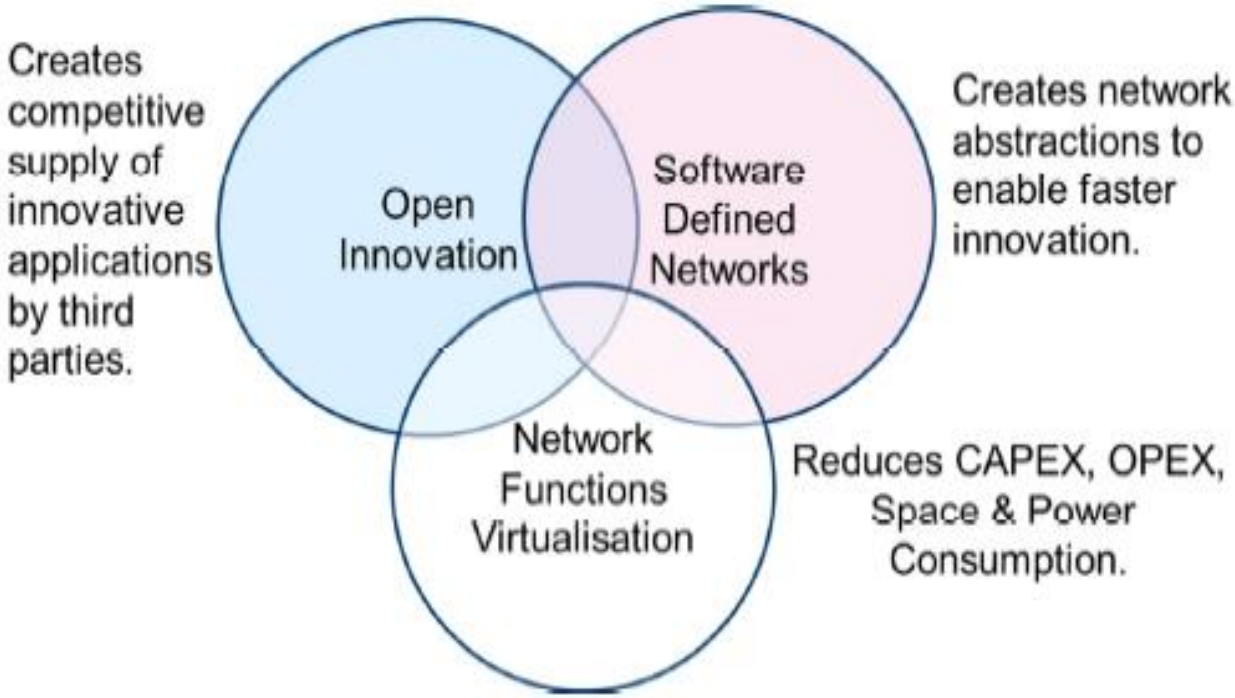
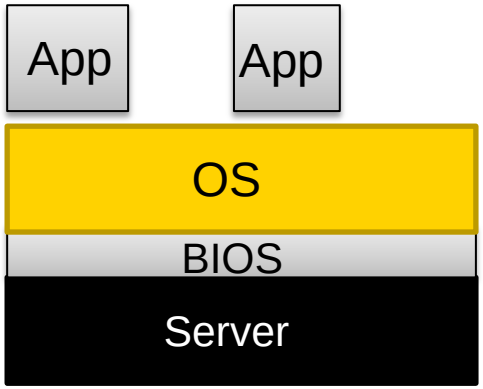


Figure 2: Network Functions Virtualisation Relationship with SDN

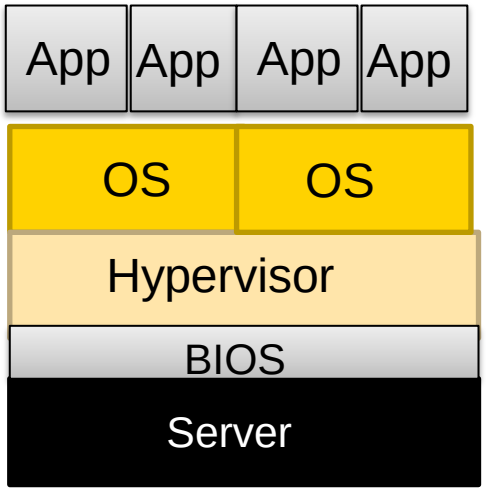
Source: ETSI ISG NFV White Paper; http://portal.etsi.org/NFV/NFV_White_Paper.pdf

Software on top of COTS Hardware: two approaches

COTS=commercial off-the shelf

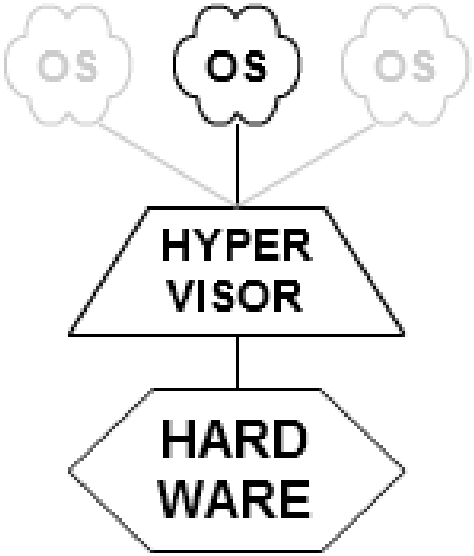


- Traditional Approach



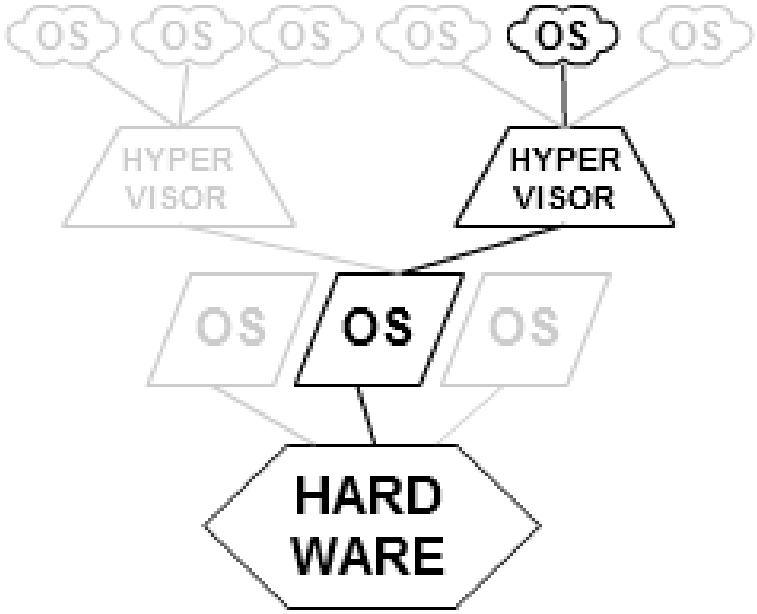
- Abstraction:
  - Multiple OS / Apps on one server
- Mobility:
  - VM migration
- Cloning:
  - Scale in/out
- Resource management
- Real Isolation

Virtualization Approaches



TYPE 1

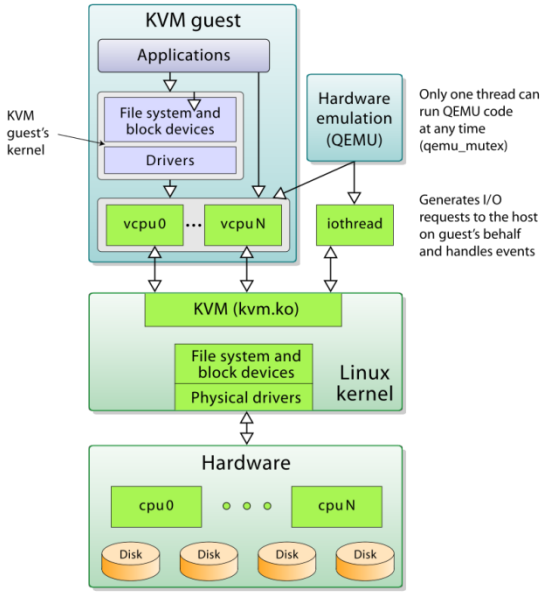
native
(bare metal)

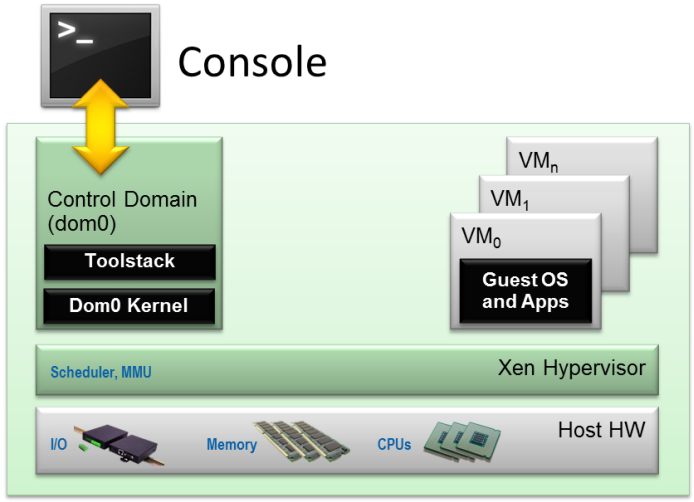


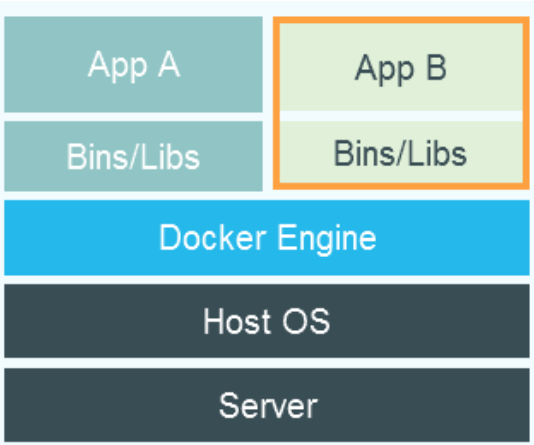
TYPE 2

hosted

Virtualization Approaches







Type 2: Kernel VMs
- Linux Kernel as hypervisor
- „Fat" hypervisor
- VM as user space process
- (still priv. Mode)
- Shared resources (drivers)

Type 1: HV on bare metal
- Shared domain concept
- Pure hypervisor w/o linux kernel/drivers
- Drivers in one (or more) VMs (driver domain)
- Split driver model

Process virtualization
- Linux Containers
- Leaner, but less isolation capabilities
- Own namespace/libs per app but
- all apps use same Kernel

Pictures taken from projects' websites

Networking in the virtualization era: Some Facts

- **Every server** has a **SOFTWARE VSWITCH** inside
  - to switch packets across multiple VMs
- **Network functions** (inside VMs) are **I/O intensive**
  - **IT functions** are **computing intensive**
- **Consolidation of VMs** on a single server brings **CAPEX reduction**
  - trend towards **many VMs on a PC**

Performance Requirements

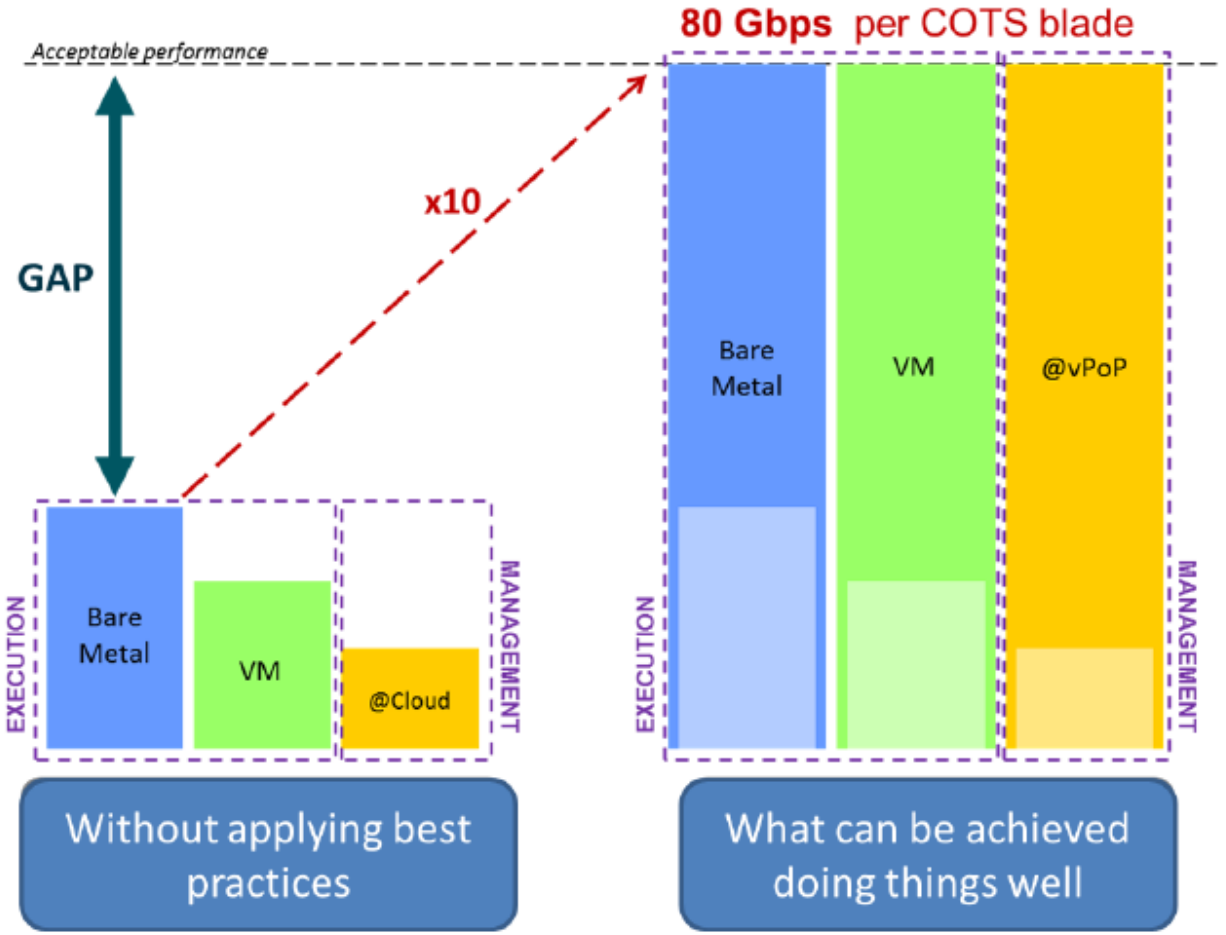
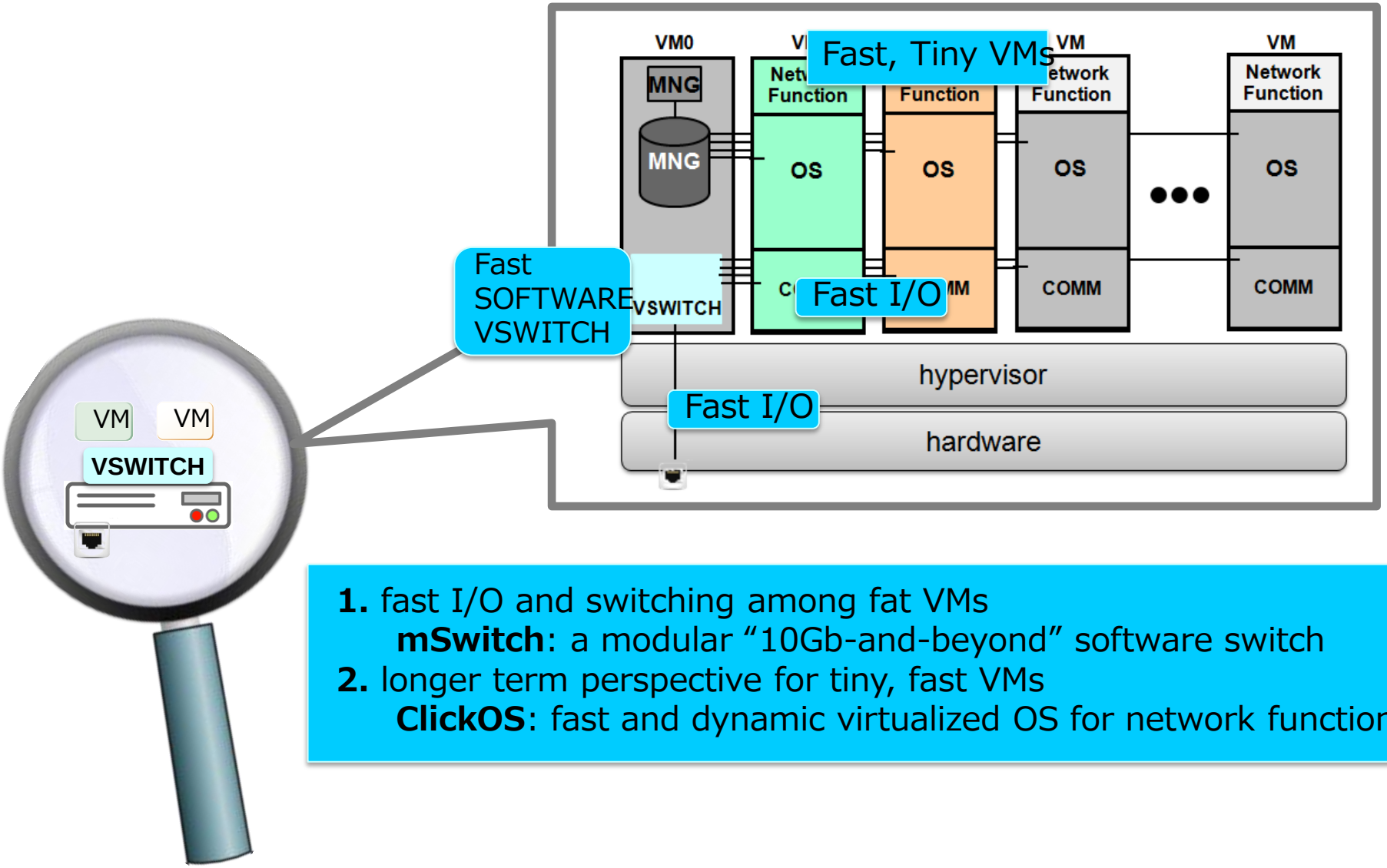


Figure 3: Performance Illustration

High level technical overview:

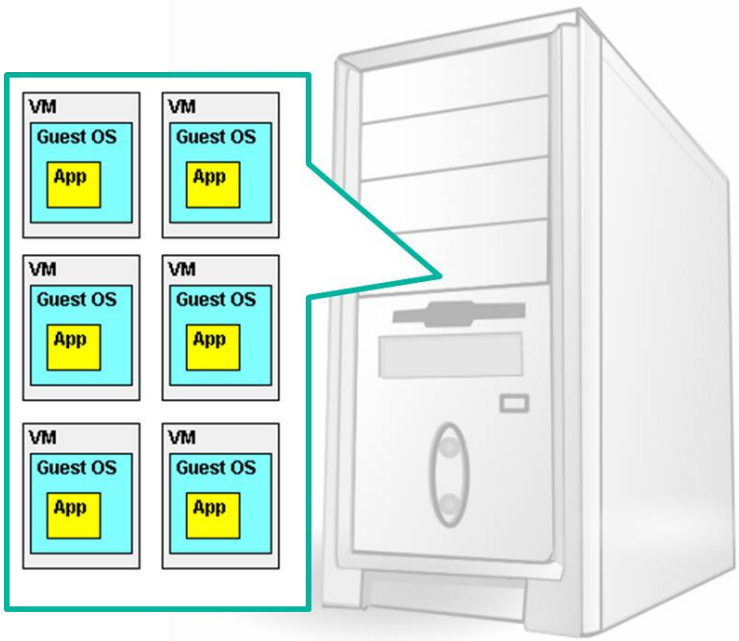
what is needed?



1. fast I/O and switching among fat VMs
mSwitch: a modular "10Gb-and-beyond" software switch
2. longer term perspective for tiny, fast VMs
ClickOS: fast and dynamic virtualized OS for network functions

Network Functions Virtualization: Smaller VMs

- Classical virtualization of nodes:
  - Fat, big and slow



- New virtualization technology
  - Tiny & tailored
  - Fast (10Gbps on a PC & small boot times)
  - Natively support NW functions

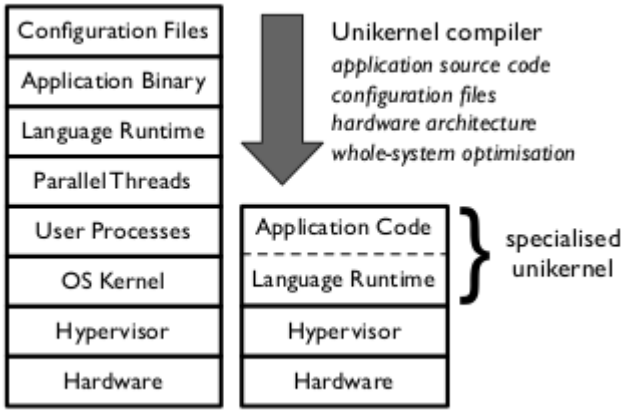
For details, see EWSDN 2012

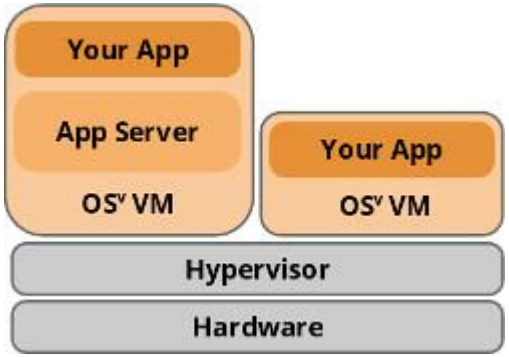
S. Niccolini: Free your middlebox functions down to the data plane with tiny, fast network VMs
http://www.ewsdn.eu/presentations/NetworkFunctionsVirtualization-EWSDN-v0.3-public.pdf

Uni-Kernels, Cloud Operating Systems

Specialized OS kernels

- Linked with application code
- More efficient than traditional layering
- For example: many system calls can be transformed to library calls
- Requires pre-compilation of code, configuration into specialized uni-kernel





Mirage OS

- https://mirage.io
- Ocaml-based runtime and app development environment

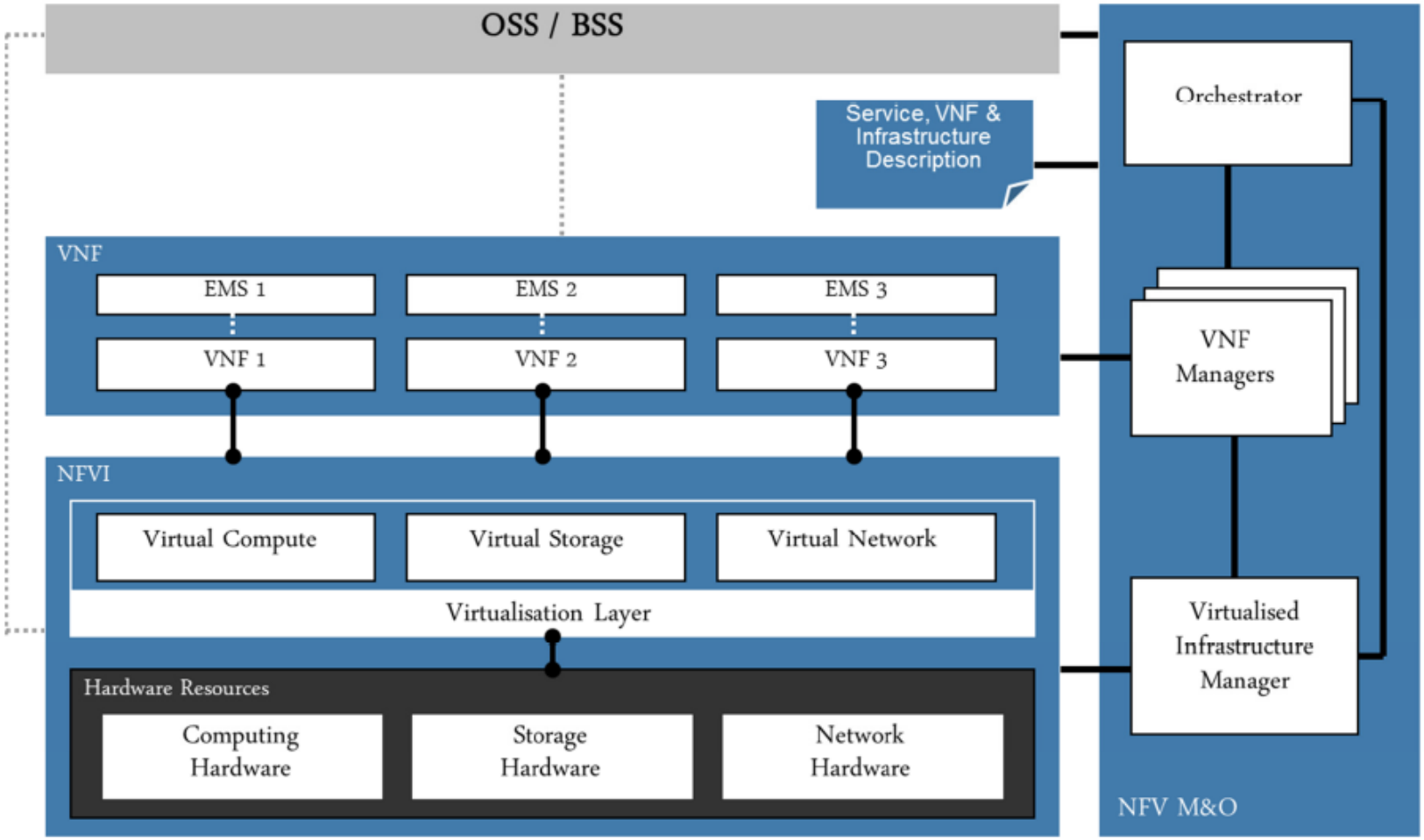
OsV

- http://osv.io/
- Optimized OS
- Applications as single processes, no protection between user space and kernel
- Optimizations: zero-copy, lockless network APIs, enhanced JVM etc.
- Linux-compatible

Some Research Challenges

- **Virtualization Performance**
- **More fine-granular virtual machines, appliances**
- **Higher dynamicity in instantiating, destroying VMs**
- **DC workload assessment, prediction**
- **New functions based on library OSs**

NFV Architecture (ETSI NFV ISG)



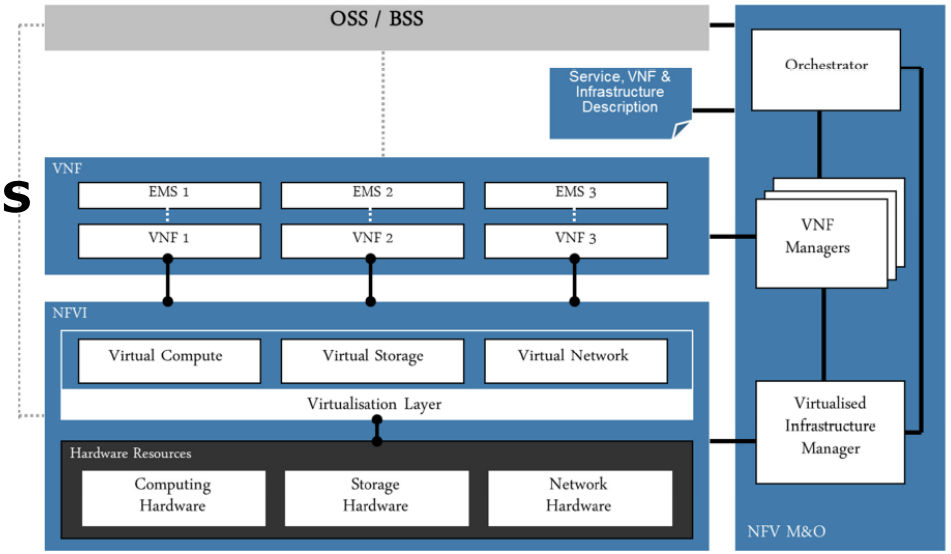
NFV Architecture (ETSI NFV ISG)

Infrastructure (NFVI)

- Provides virtual resources to support execution of VNFs
- COTS hardware, accelerator components
- Software-layer to virtualize and abstract from underlying hardware

Virtualized Network Functions (VNFs)

- Software implementations of network functions
- Running on the NFVI
- Can come together with an Element Management System (EMS)



Management and Orchestration (MANO)

- Orchestration and life-cycle management of physical/software resources
- Life-cycle management of VNFs
- Examples: on-boarding, failure recovery, scale-out
- Interacts with Operations/Business Support Systems (OSS/BSS)

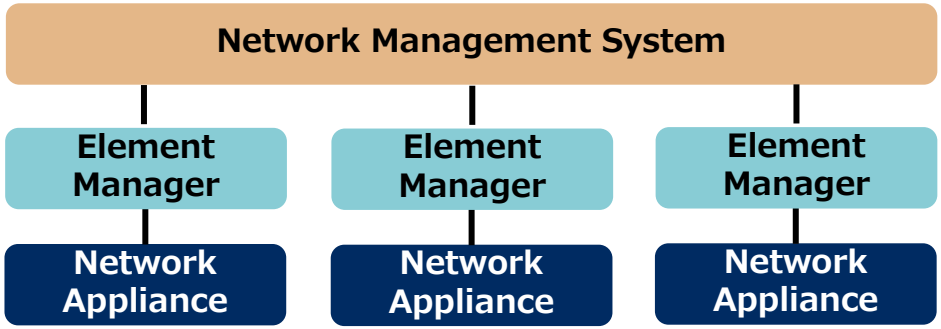
Automation in NFV

Two main motivations for moving telco networks into the cloud

1. **Resource efficiency** (consolidation of functions onto fewer servers)
2. **Automation** (fully automated management of resources and functions)

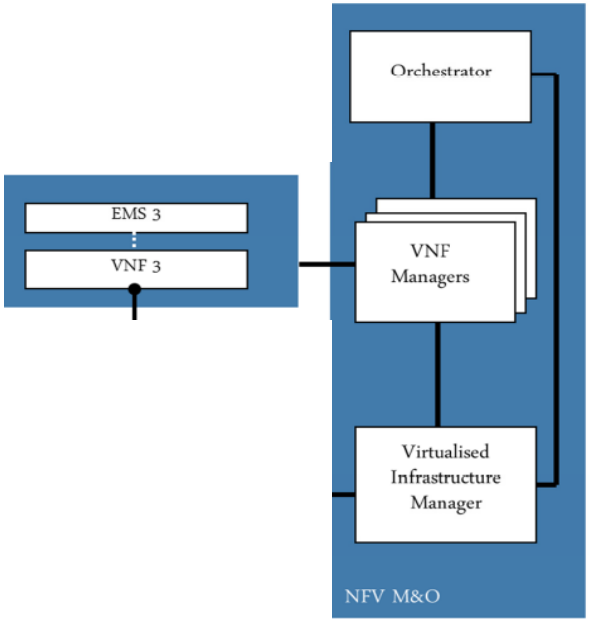
Traditional telco management approach

- Specific (vendor-provided) element manager and network management systems
- Specific management approaches, fault recovery mechanisms etc.

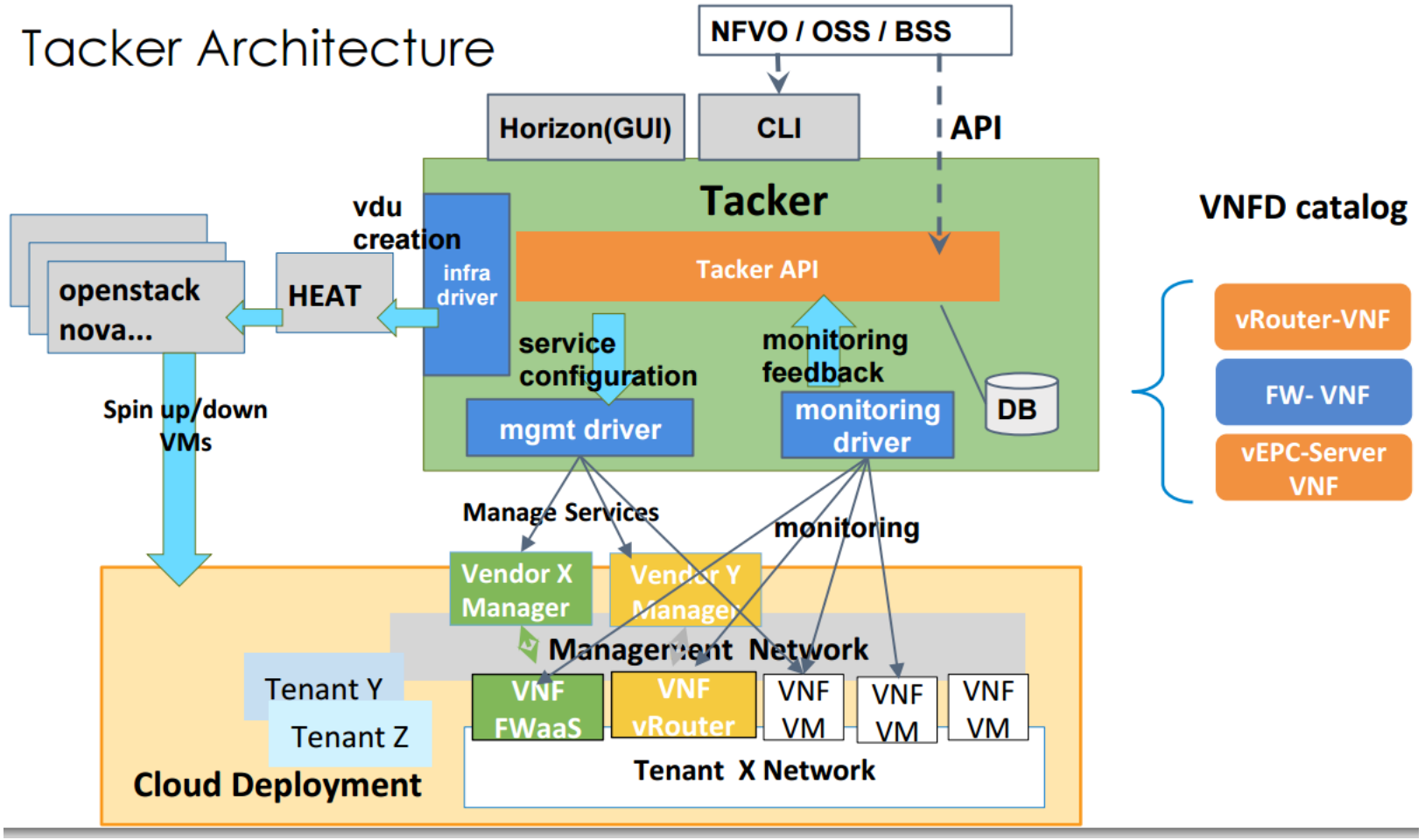


Cloud management approach

- Generalized VNF management
- Part of NFV management and orchestration

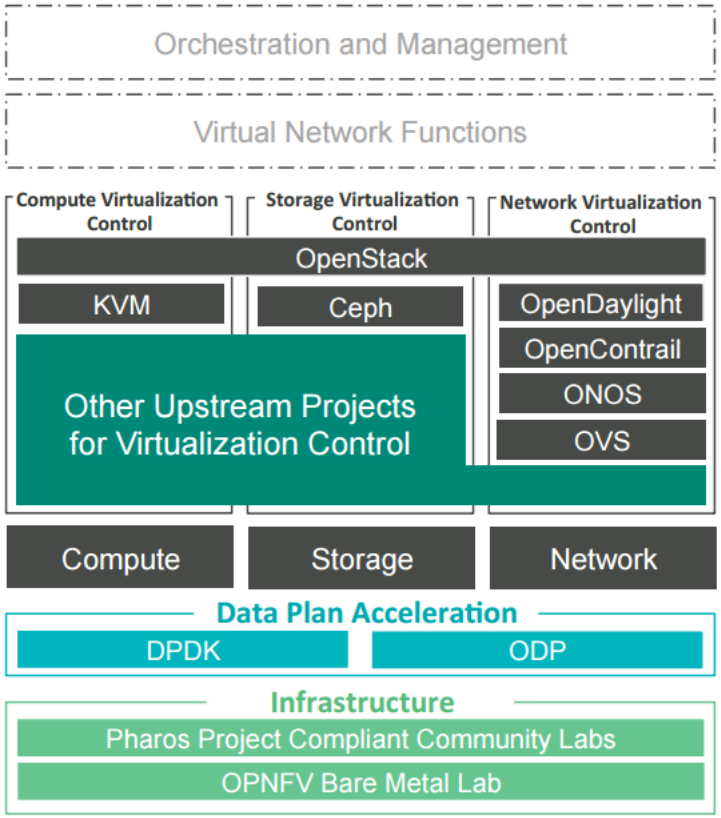


OpenStack Tacker



https://www.ietf.org/proceedings/93/slides/slides-93-nfvrg-25.pdf

Open Source NFV



OPNFV

www.opnfv.org

Some Research Challenges

Next steps for auto management

- **Managing larger sets of smaller VMs**
- **Light-weight management**
- **Predicting load, failures**
- **Optimizing instantiation, VM positioning etc.**

Service Chaining

Telco networks

- Operator services
- Optimizers
- Service differentiation
- Policy enforcement for QoS, resource management etc.

Traffic passes through different network functions

- Depending on user context, flow specifics, network weather condition etc.

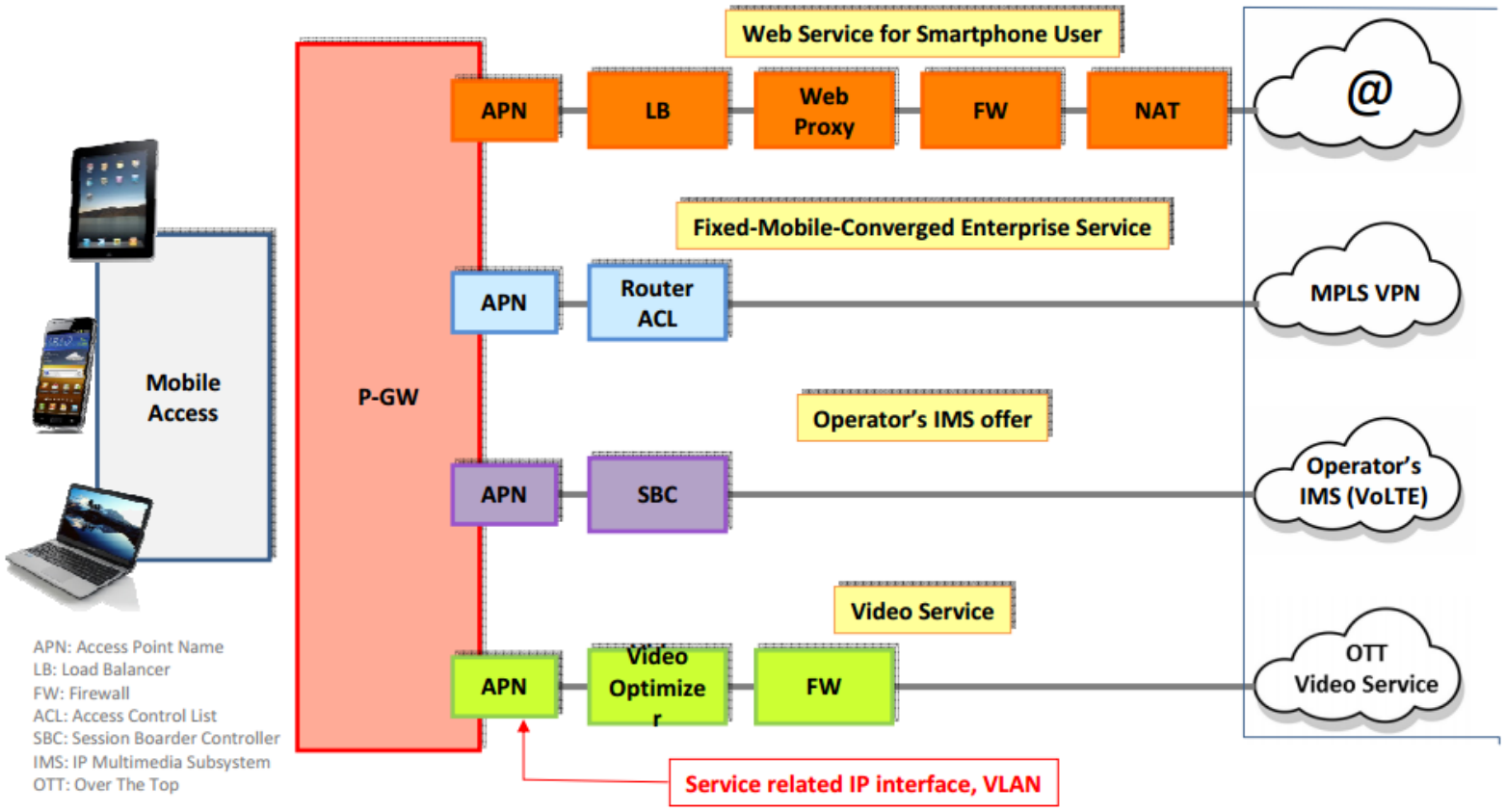
Different types of functions

- Network layer translation: CG-NAT/PAT
- Security related: intrusion-detection, firewalls, access control, en/decryption etc.
- Qos related: video optimizers, TCP optimizers, policers
- User identification/tracking: HTTP header enrichment
- Other „value-added“ services: parental control, malware protection

Vodafone Example

Service Function Chaining

Current Common Approach – Logical View on Typical Use Cases



http://www.ikr.uni-stuttgart.de/Content/itg/fg524/Meetings/2013-11-15-Muenchen/01_ITG524_Muenchen_Haeffner.pdf

Service Chaining Requirements

Legacy service chaining

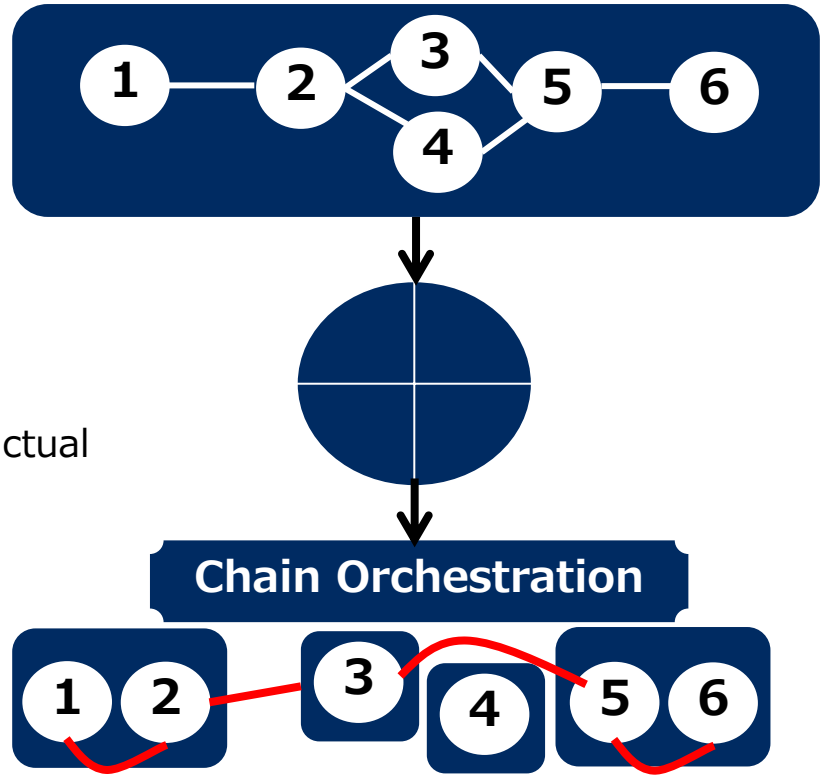
- Static chains, hard-wired on network layer – through manual configuration
- Not very flexible – difficult to change chain configurations
- Connectivity graphs dependent on actual network graph
- Too complicated to allow for per-user chains
- Not compatible to cloud networking (automatic scale out etc.)

Ideally:

Abstract specification
- Conditions
- Scaling requirements

- Service chain compiler
  - Creating initial embedding on actual network graph

- Management and orchestration
  - Actual instantiation
  - Monitoring
  - Scaling, graph adaptation etc.



Implementing Service Chaining

Flow Steering – different approaches

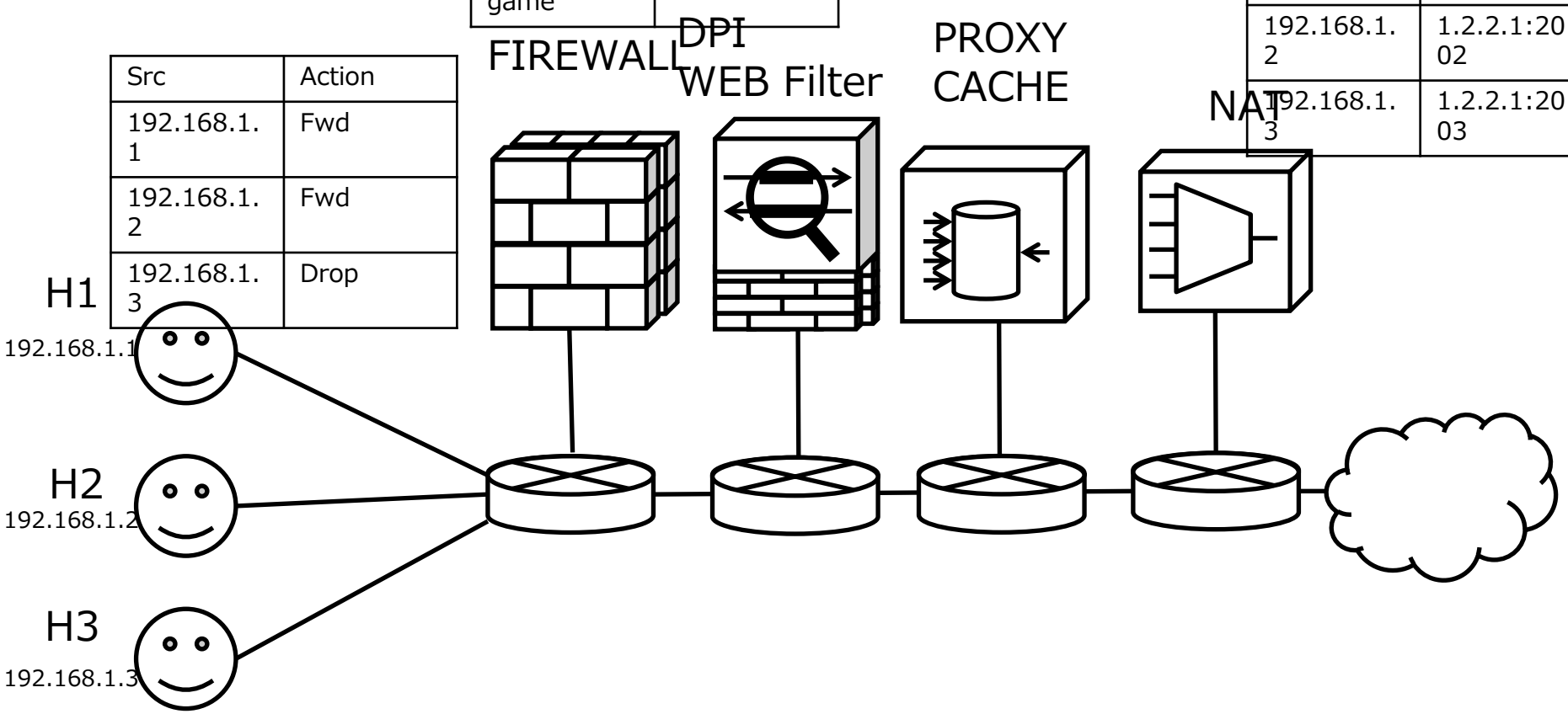
- VLAN, MPLS, VXLAN, GRE
- Policy Routing
- OpenFlow

Dst	Action
Adult content	Drop
Online game	Drop

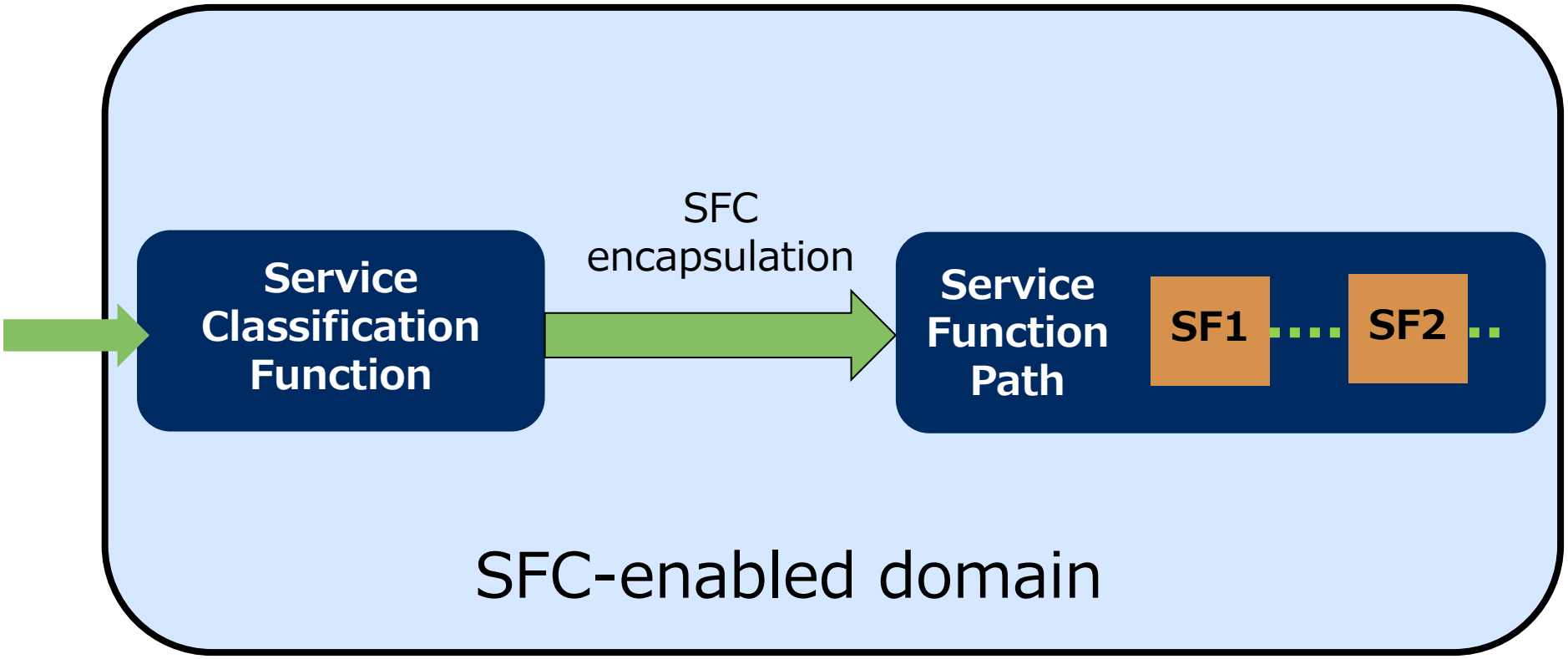
Dst	Action
NEC	From Cache
IEEE library	From Cache

Origin	Mapped
192.168.1.1	1.2.2.1:2001
192.168.1.2	1.2.2.1:2002
192.168.1.3	1.2.2.1:2003

Src	Action
192.168.1.1	Fwd
192.168.1.2	Fwd
192.168.1.3	Drop

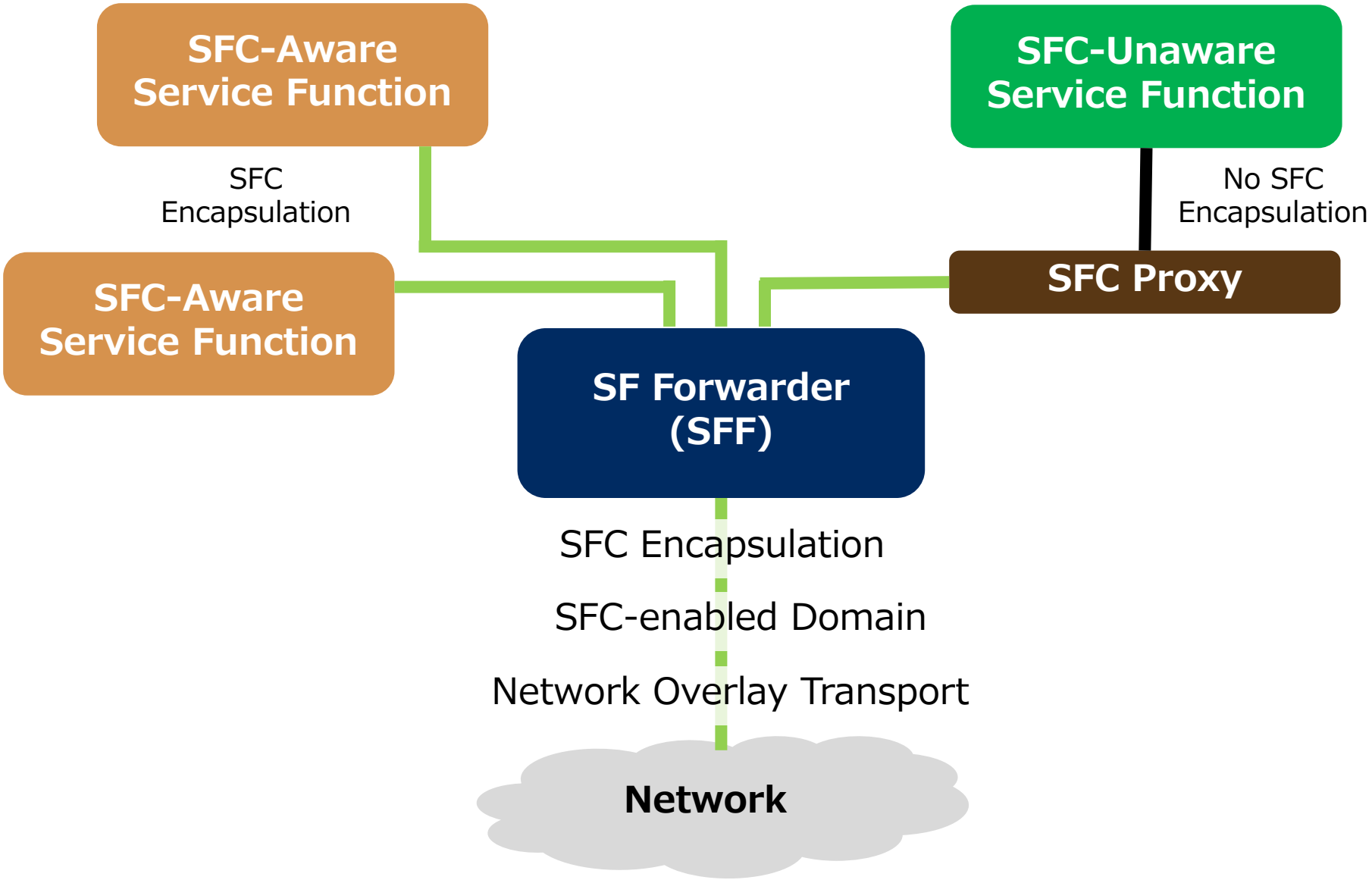


IETF SFC Architecture



- Encapsulation-based approach
- Intended to support SFC-unaware service functions

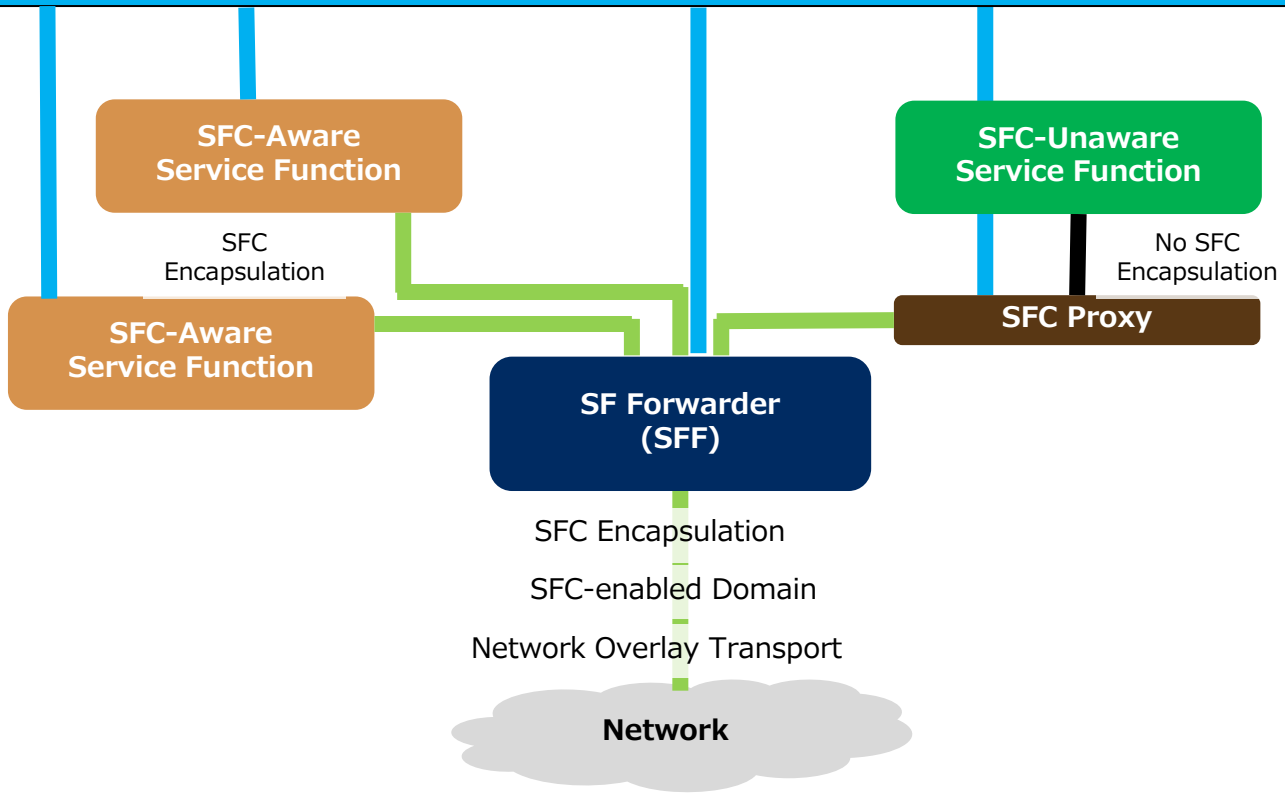
IETF SFC Architecture



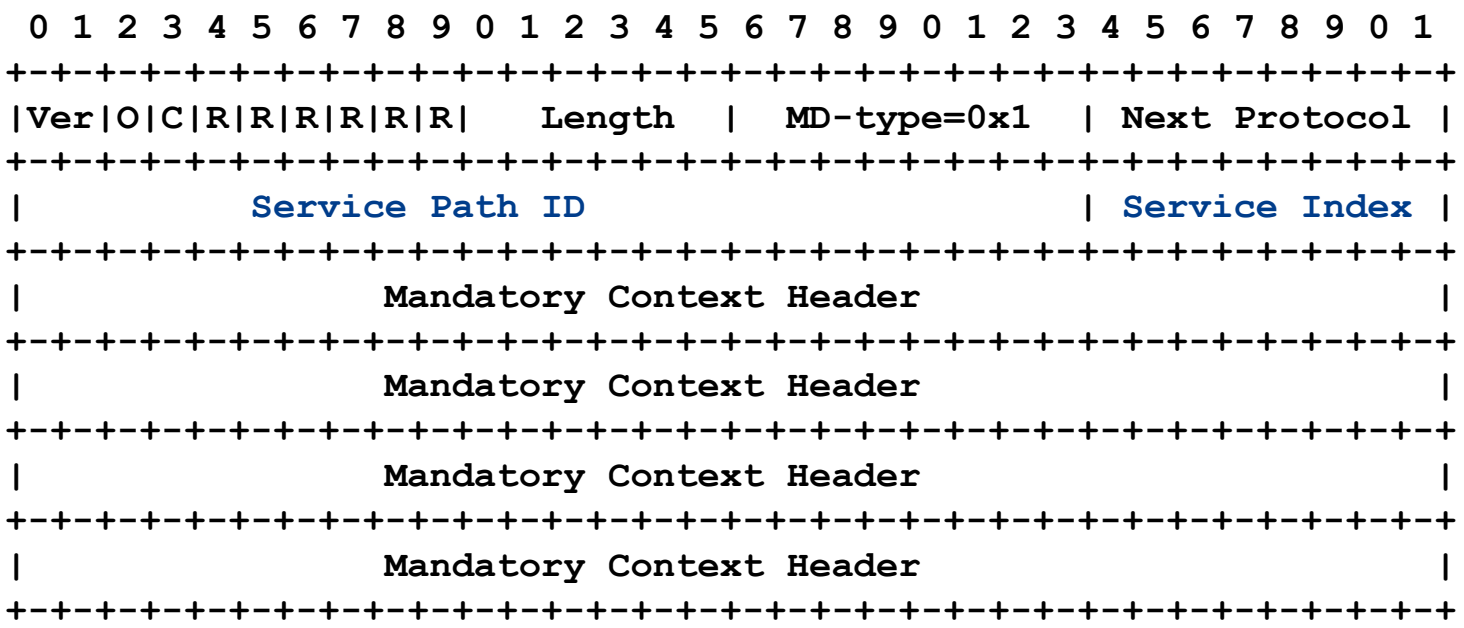
IETF SFC Architecture

SFC Control Plane

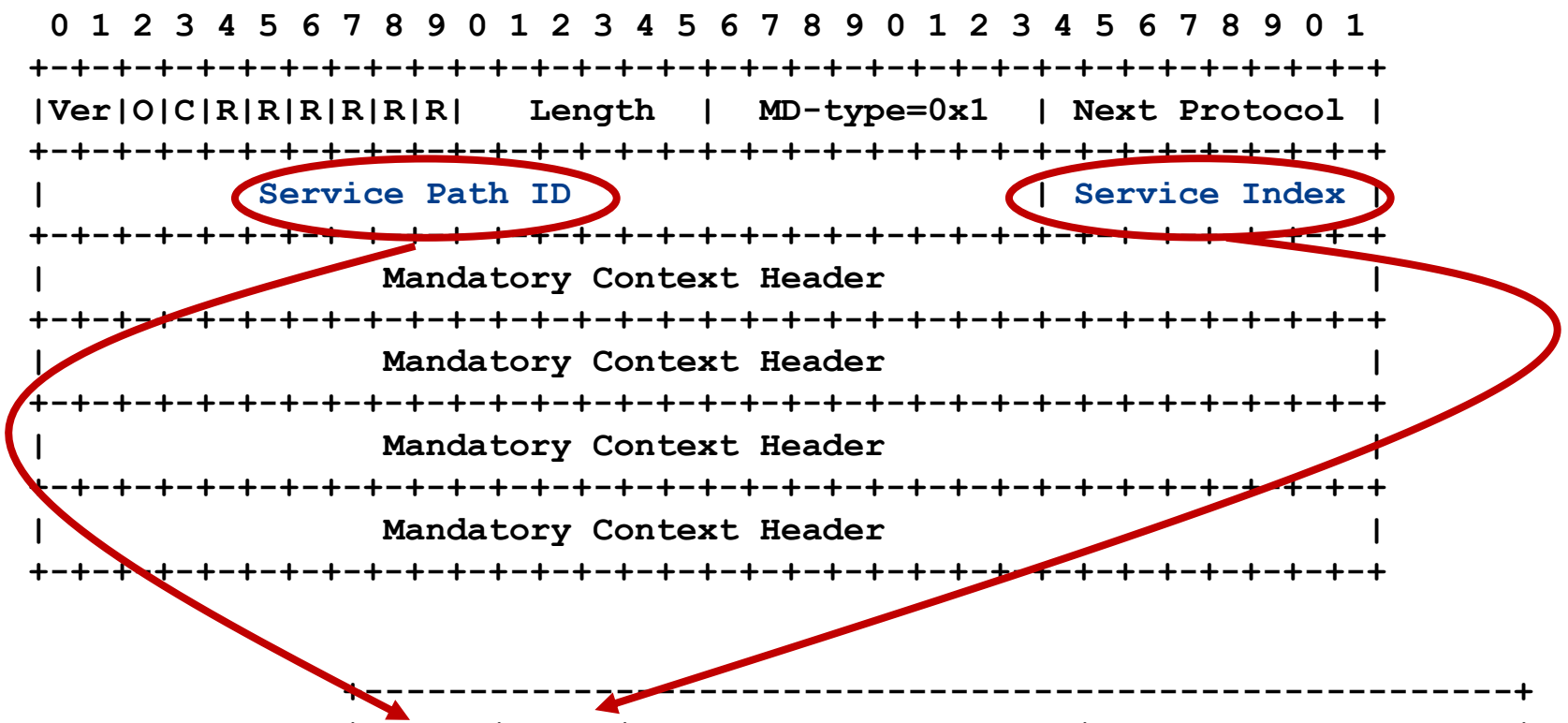
- Domain-wide view of SF resources and their locators
- Uses SFC Policy to construct service function chains and actual paths
- Selection of SFs for SFCs (statically or dynamically)
- Provide SFC data plane to components
- Provide metadata and usage info to classifiers
- Provide policy information to SFC elements to interpret metadata



NSH Header Type 1



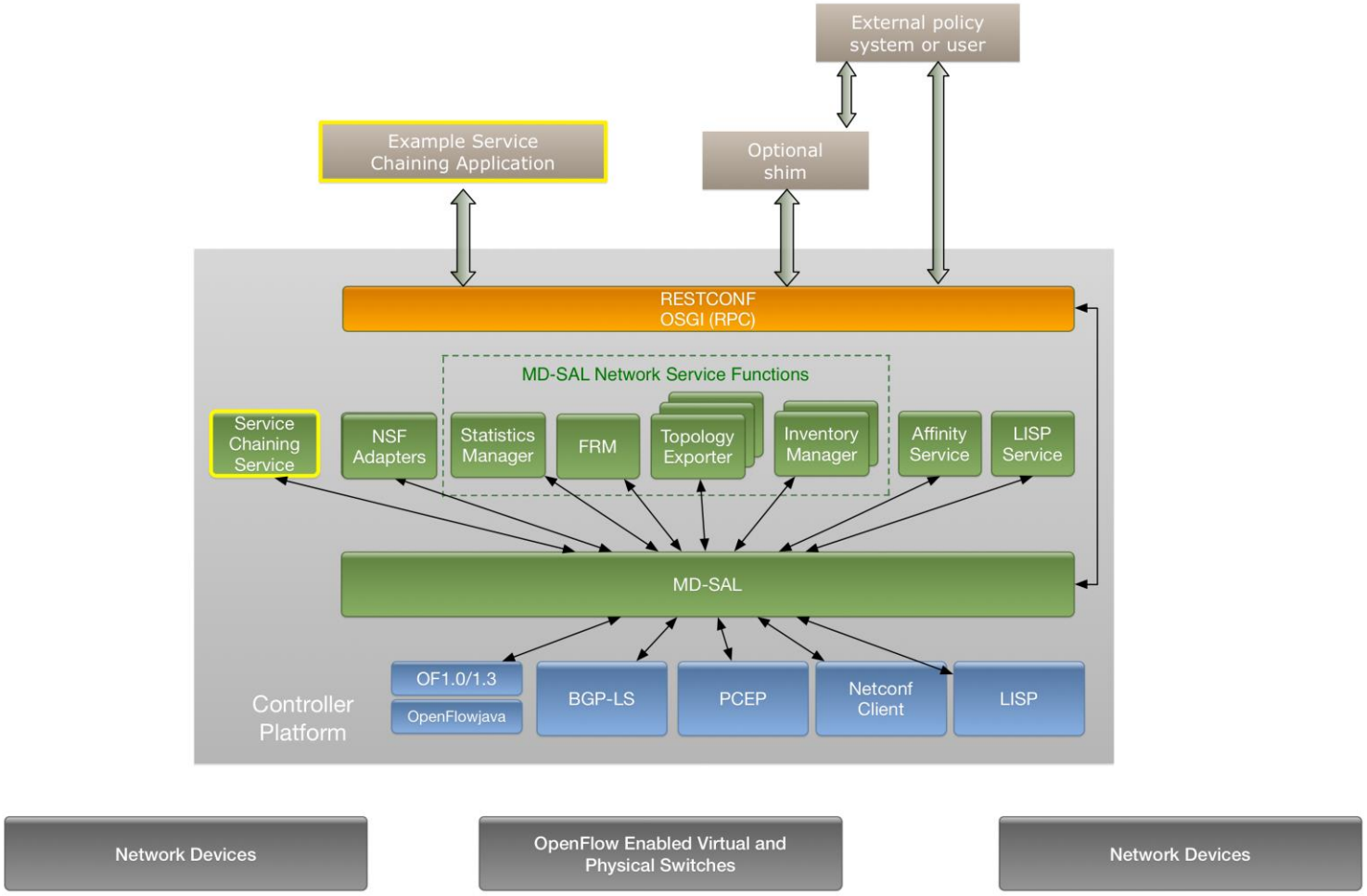
NSH Header Type 1



Maping example

SPI	SI	NH	Transport
10	255	1.1.1.1	VXLAN-gpe
10	254	2.2.2.2	nvGRE
10	251	10.1.2.3	GRE
40	251	10.1.2.3	GRE
50	200	01:23:45:67:89:ab	Ethernet
15	212	Null (end of path)	None

OpenDaylight SFC Project



https://wiki.opendaylight.org/view/Service_Function_Chaining:Main

Service Chaining Research Topics

Transport Performance

- Impact on TCP performance
- Placement optimizations

Security

- Service chaining in the presence of ubiquitous encryption
- „Cooperative Traffic Management“

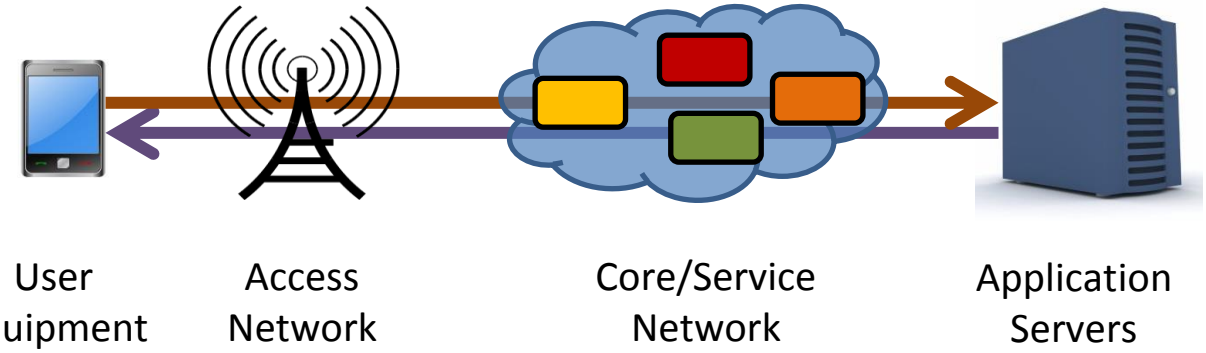
Micro service chains

- Fine-granular service chaining structure
- Compose functions from building blocks
- Reason about optimal ordering, parallelism, placement

New approaches to chaining in-network functions

- Arumaithurai et al.; Exploiting ICN for Flexible Management of Software-Defined Networks
- Integrating distributed computing and storage: Named Function Networking

Applying SDN/NFV



- Softwarization previously in data centers only (cloud)
- Now moving towards core and access networks

- New options for enhanced functionality, new features, news architectures

Information-Centric Networking

Accessing Named Data Objects (NDOs) in the network

- ADUs, chunks, fragments

Data-centric security approach

- Disentangled means for name-content binding validation, publisher authentication, confidentiality

Name-Content binding validation:
hash-based schemes

Publisher authentication

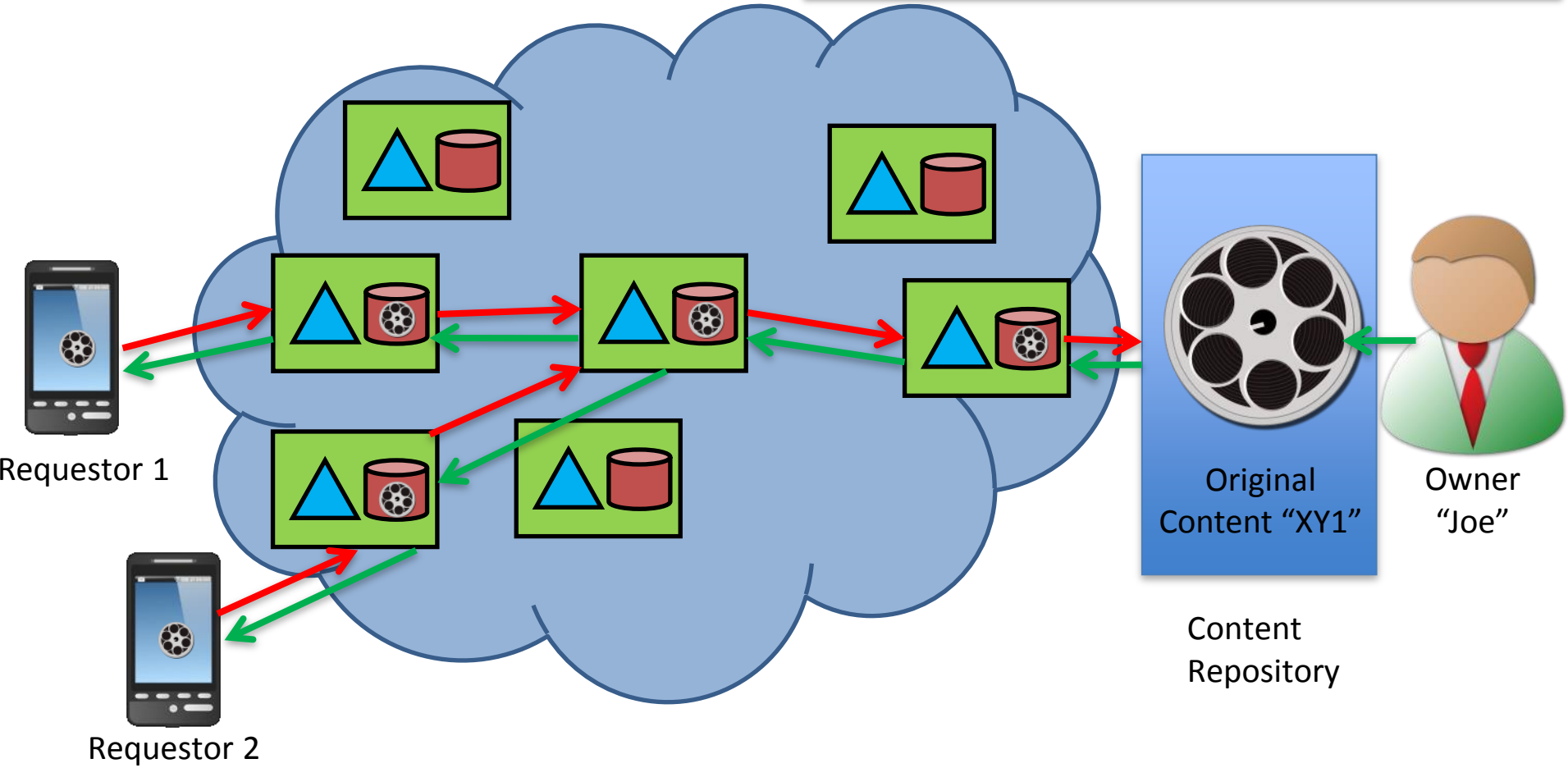
- One approach: publishers to sign NDOs, signature part of NDO meta data; trust model a la PKI

Confidentiality

- Encrypting payload with publisher key

ICN Overview

- Request Response
- Pending Interesting Tables
- Receiver-driven
- Route-by-name (prefix)
- Ubiquitous caching



ICN Traffic and Resource Management

Key ICN properties

- Requesting individual Named Data Objects
- Ubiquituous Caching

Implicit caching

- Every router can store NDO – depending on configuration, policy etc.
- Even with encrypted traffic, caching can help with local retransmissions, media re-play etc.

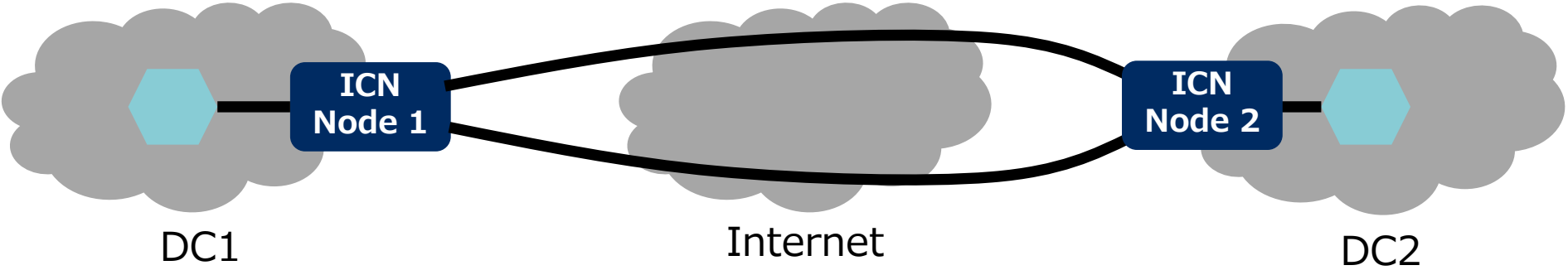
Flexible multipath communication

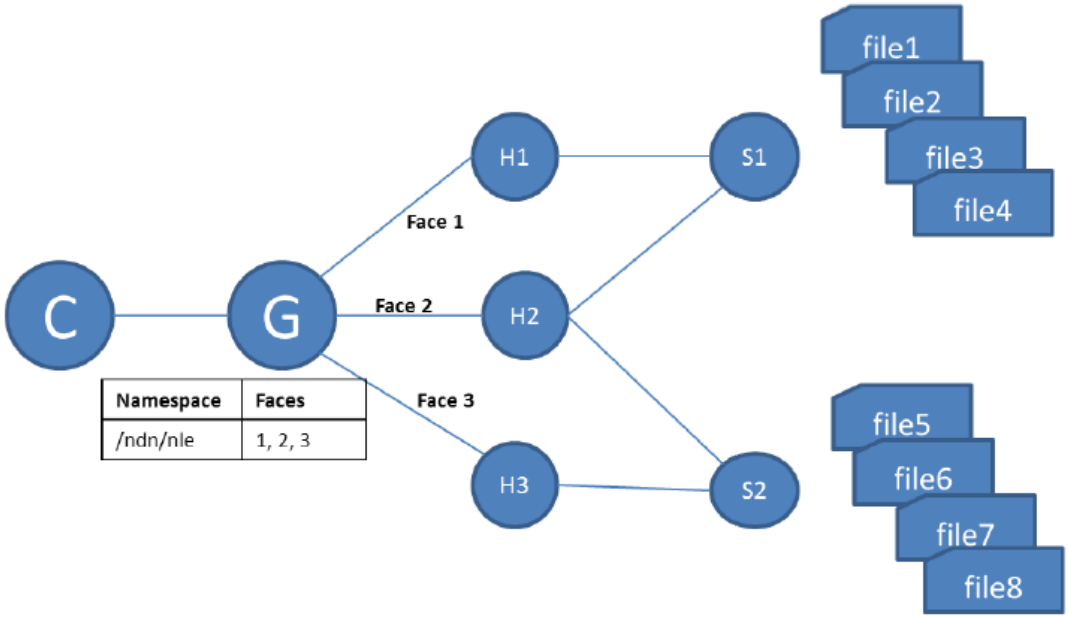
- Every router can make forwarding decisions depending on strategy, network characteristics, name prefix, policy

Easy policing and filtering

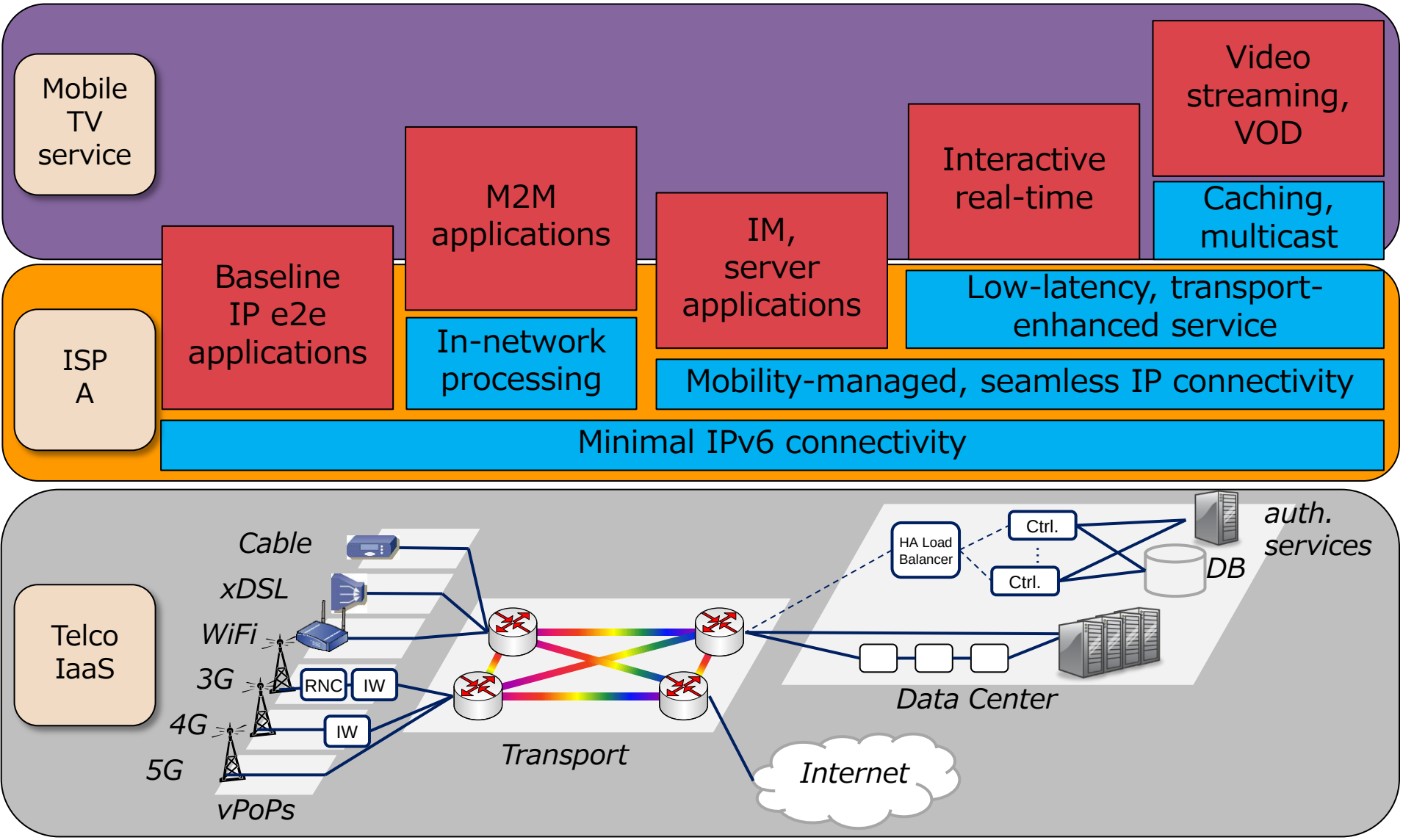
- Requestors, publishers and requestors see ICN requests and responses
- Policing without DPI
- Enabling other optimizations: in-network pre-fetching etc.

Better Multipath-Support through ICN

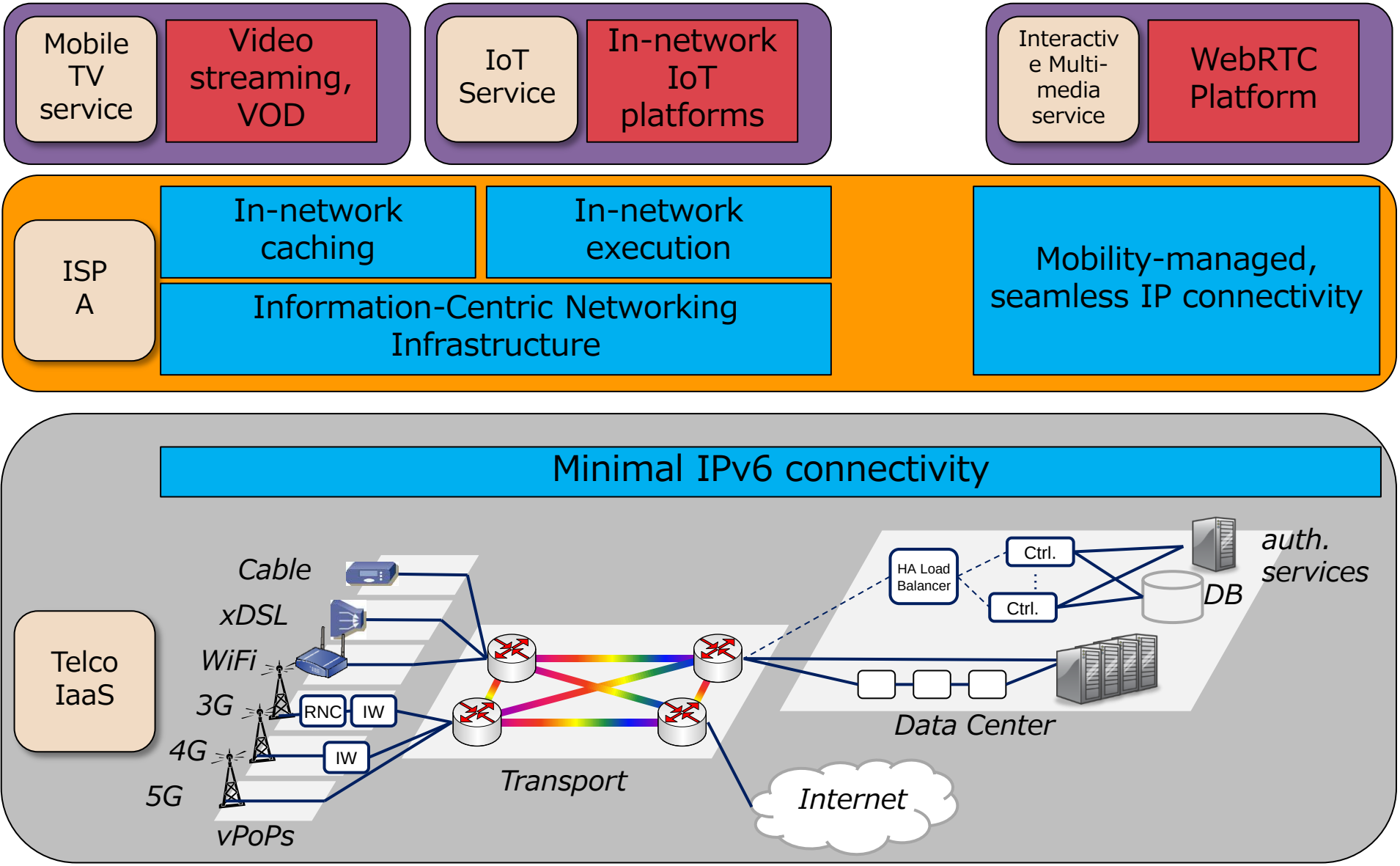




ICN in 5G Blueprint – Organizational Layers



5G Blueprint Reshuffled and Modernized



SDN References

- SANE: http://dl.acm.org/citation.cfm?id=1267346
- Ethane: http://dl.acm.org/citation.cfm?id=1282382
- OpenFlow: http://dl.acm.org/citation.cfm?id=1355746
- NOX: http://dl.acm.org/citation.cfm?id=1384625
- OVS: http://openvswitch.github.io/papers/hotnets2009.pdf

- ONF SDN Arch: TR-502, Overview TR-504
- Further ONF specifications: www.opennetworking.org

- More scientific papers @ https://sites.google.com/site/sdnreadinglist/

NFV References

Overview

- https://portal.etsi.org/nfv/nfv_white_paper.pdf
- https://portal.etsi.org/nfv/nfv_white_paper2.pdf
- https://portal.etsi.org/Portals/0/TBpages/NFV/Docs/NFV_White_Paper3.pdf

Open Source

- www.opnfv.org
- https://wiki.opendaylight.org/view/Service_Function_Chaining:Main

IRTF NFVRG

- https://irtf.org/nfvrg

IETF Service Chaining

- https://tools.ietf.org/wg/sfc/

Interested in a Master Thesis project in our lab?

Dirk.Kutscher@neclab.eu

www.neclab.eu

Orchestrating a brighter world

未来に向かい、人が生きる、豊かに生きるために欠かせないもの。
それは「安全」「安心」「効率」「公平」という価値が実現された社会です。

NECは、ネットワーク技術とコンピューティング技術をあわせ持つ
類のないインテグレーターとしてリーダーシップを発揮し、
卓越した技術とさまざまな知見やアイデアを融合することで、
世界の国々や地域の人々と協奏しながら、
明るく希望に満ちた暮らしと社会を実現し、未来につなげていきます。


Orchestrating a brighter world
NEC