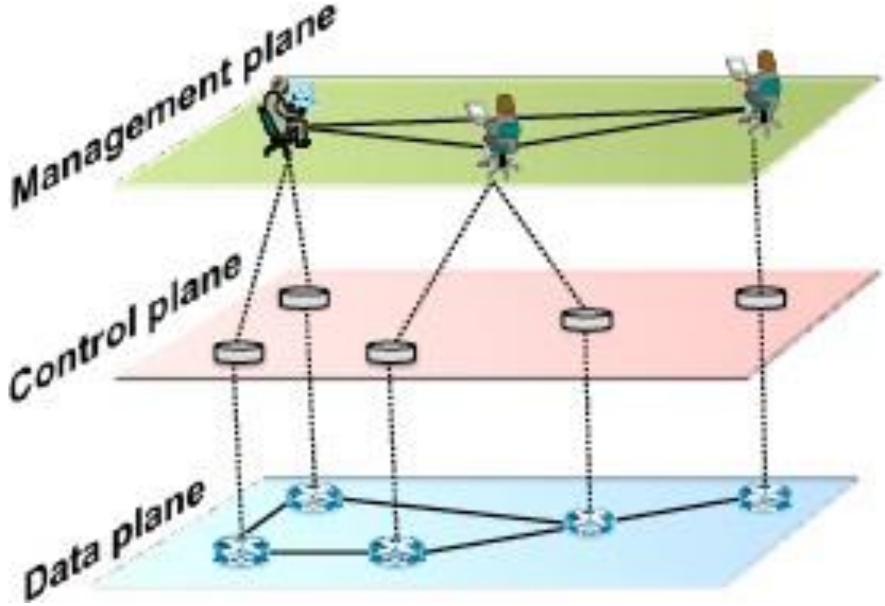
Software-Defined Networking

Towards Network Programmability

What's Software-Defined Networking?

Network's Functional Planes



Source: "Software-Defined Networking: A Comprehensive Survey", Kreutz et al., https://arxiv.org/pdf/1406.0440.

What's Software-Defined Networking?

› Main principle: data plane decoupled from control plane.

Why Software-Defined Networking?

- The Internet has been the victim of its own success!
- Extremely hard to configure, manage, and evolve.
- "Vertically integrated": tight coupling of control- and data planes embedded/distributed in network devices.
- Proliferation of "middleboxes".

Source: N. McKeown, Stanford, ONF

Why Software-Defined Networking?

- The Internet has been the victim of its own success!
- Extremely hard to configure, manage, and evolve.
- "Vertically integrated": tight coupling of control- and data planes embedded/distributed in network devices.
- Proliferation of "middleboxes".

Software-Defined Networking to the rescue!

- Separation of control plane from data plane.
  - Network control "logically centralized" in the ***controller***.
  - Forwarding hardware simplified.
- Programmable networks to facilitate management and control and combat "network ossification".
- Data plane "commoditization".

Software-Defined Networking Architectures

IETF ForCES

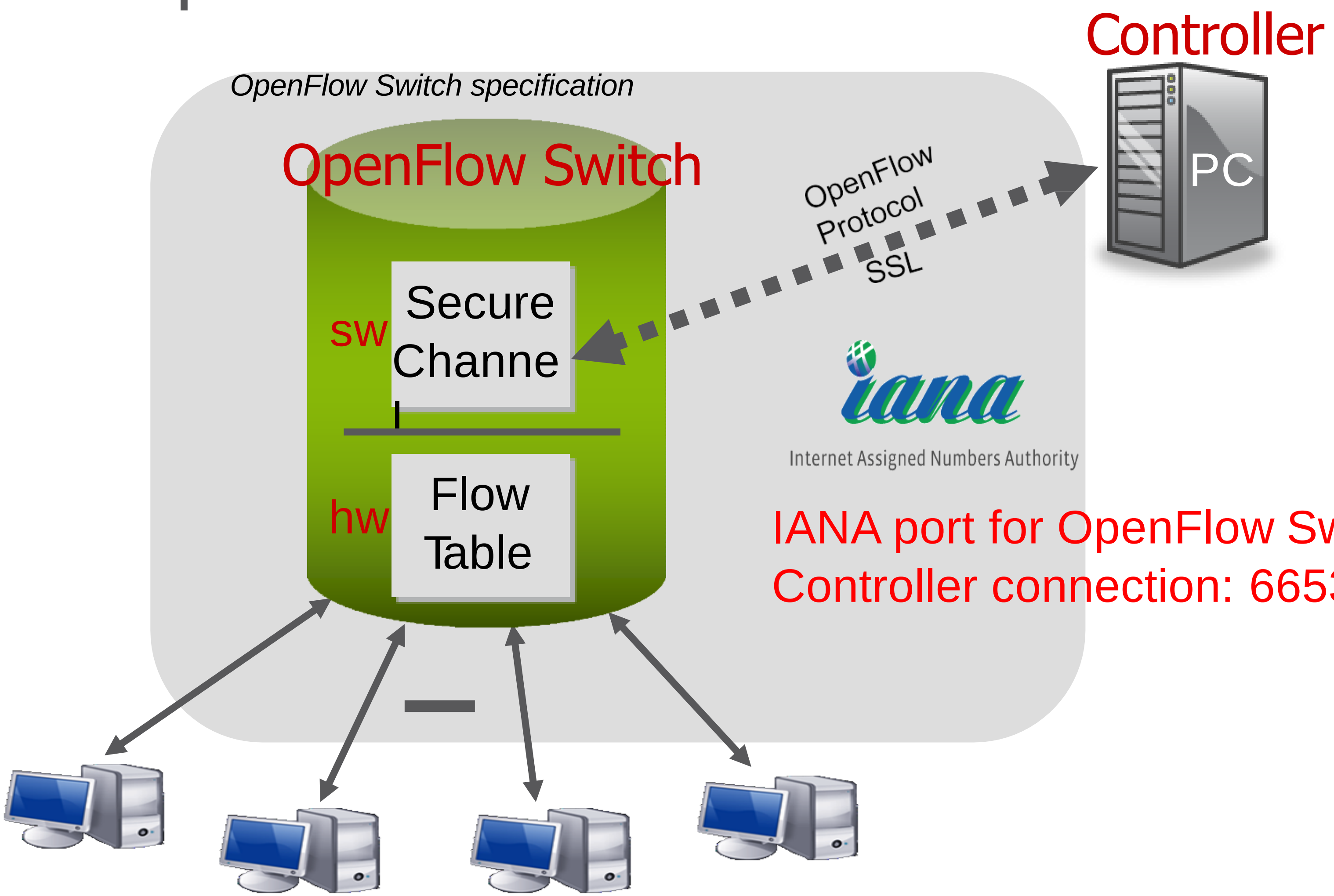
- Forwarding Element (FE) and Control Element (CE)
- Both reside in the network device
- FE and CE communicate using the ForCES protocol

ONF OpenFlow

- Decoupling between control- and data planes
- Controller and switch communicate using the OpenFlow protocol

ONF OpenFlow Architecture

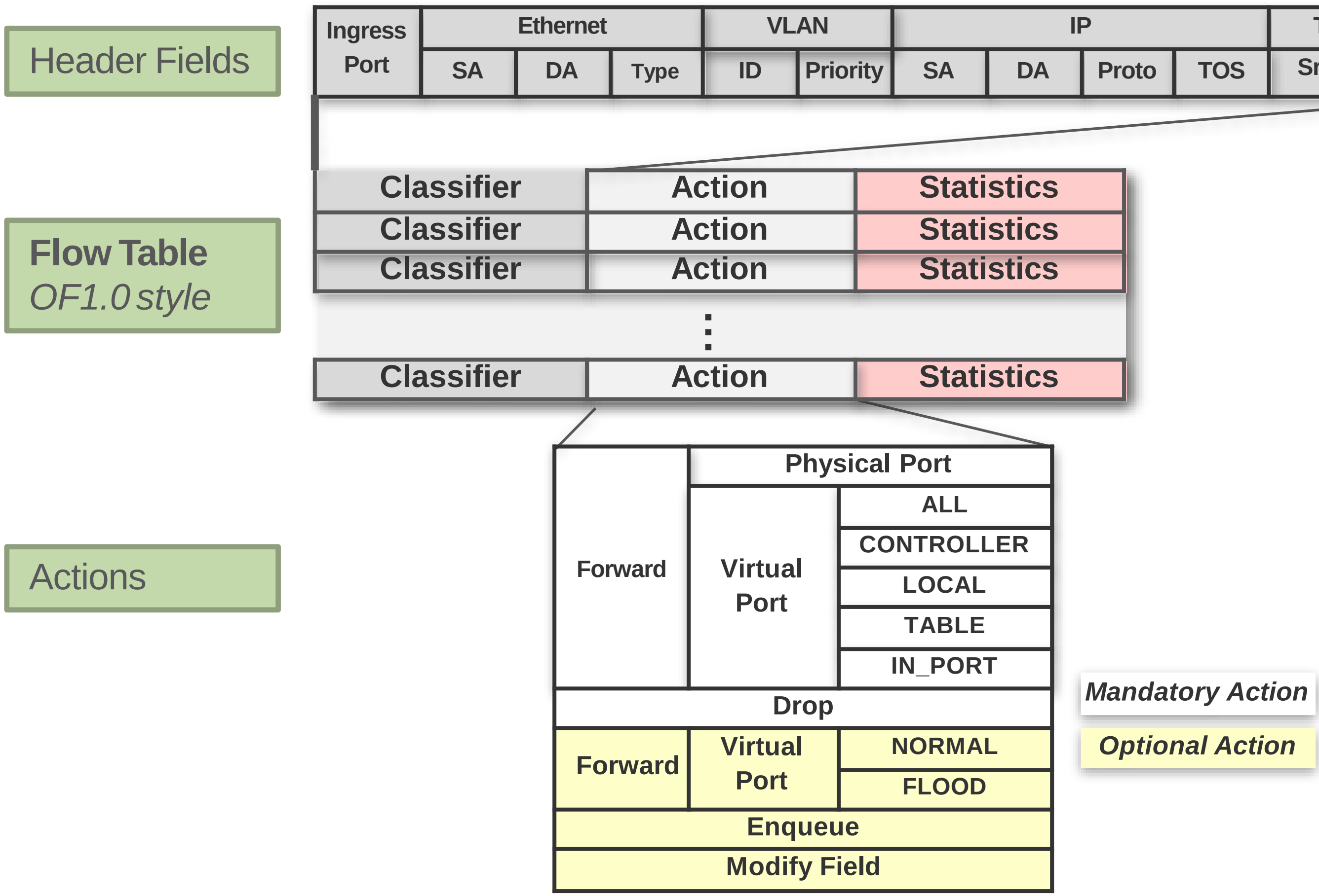
ONF OpenFlow Architecture



IANA port for OpenFlow Switch-Controller connection: 6653

Source: The Stanford Clean Slate Program, http://cleanslate.stanford.edu

OpenFlow 1.0 Flow Table & Fields



OpenFlow Table Entries

	Switch port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Prot	TCP sport	TCP dport	Action
Switching	*	*	00:1f :..	*	*	*	*	*	*	Port6
Flow switching	Port3	00:20 ..	00:1f ..	0800	Vlan1	1.2.3.4	5.6.7.8	4	17264	Port6
Firewall	*	*	*	*	*	*	*	*	22	Drop
Routing	*	*	*	*	*	*	5.6.7.8	*	*	Port6
VLAN switching	*	*	00:1f ..	*	Vlan1	*	*	*	*	Port6,port7, port8

Each Flow Table entry has two timers:

hard_timeout
seconds after which the flow is removed
zero mean never times-out

idle_timeout
seconds of no matching packets after which the flow is removed
zero means never times-out

If both **idle_timeout** and **hard_timeout** are set, then the flow is removed when the first of the two expires.

OpenFlow Standards

Evolution path:

- OF 1.0 (03/2010): Most widely used version, MAC, IPv4, single table (from Stanford)
- OF 1.1 (02/2011): MPLS tags/tunnels, multiple tables, counters (from Stanford)
- OF 1.2 (12/2011): IPv6, extensible expression
- OF-Config 1.0 (01/2012): Basic configuration: queues, ports, controller assign
- OF 1.3.0 (04/2012): Tunnels, meters, PBB support, more IPv6
- OF-Config 1.1 (04/2012): Topology discovery, error handling
- OF-Test 1.0 (2H2012): Interoperability conformance test processes, suites, labs
- OF 1.3.2 (May 2013), 19 errata, final review
- OF 1.4 (Aug. 2013), 9 changes + 13 extensions, More extensible wire protocol, Flow monitoring, Eviction, Vacancy events, Bundles
- OF 1.5.1 (Dec. 2014), Egress Tables, Packet type aware pipeline, Extensible flow entry statistics

Source: ONF

SDN: Some Definitions

SDN: Some Definitions

- *“The SDN architecture decouples the network control and forwarding functions enabling the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services.”* Open Networking Foundation (opennetworking.org)

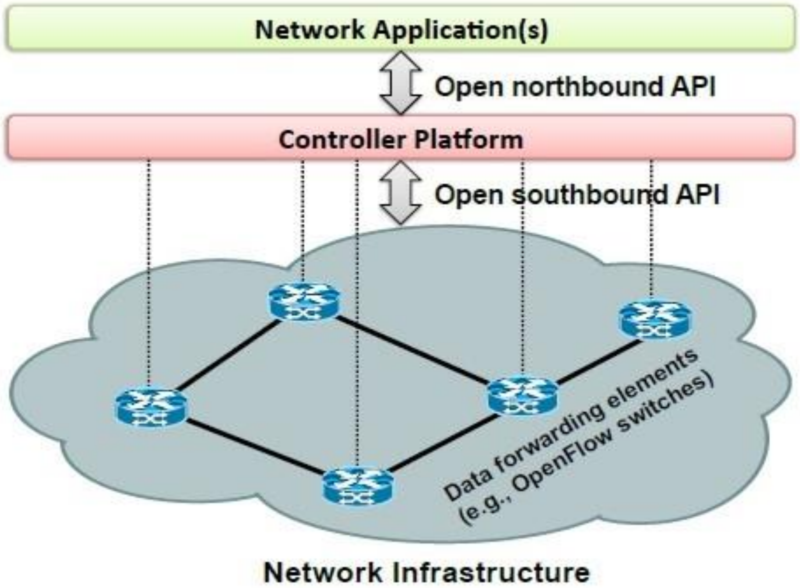
- *“Software Defined Networking (SDN) refactors the relationship between network devices and the software that controls them. Opening up the interfaces to programming the network enables more flexible and predictable network control, and makes it easier to extend the network with new functionality.”* ACM Sigcomm Simposium on Software-DefinedNetworking Research 2016

SDN: Definitions, Concepts, and Terminology

SDN refers to software-defined networking architectures where:

- Data- and control planes decoupled from one another.
- Data plane at forwarding devices managed and controlled remotely by a “controller”.
- Well-defined programming interface between control- and data planes.
- Applications running on controller manage and control underlying data plane

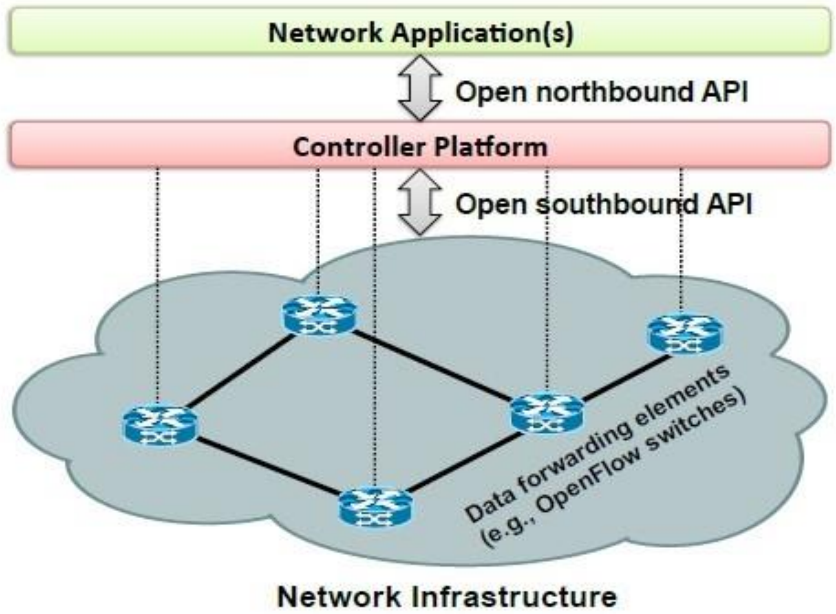
SDN architecture



Source:
“Software-Defined Networking: A Comprehensive Survey”, Kreutz et al., https://arxiv.org/pdf/1406.0440.

SDN: Definitions, Concepts, and Terminology

- **Data plane**: network infrastructure consisting of interconnected forwarding devices (a.k.a., forwarding plane).
- **Forwarding devices:** data plane hardware- or software devices responsible for data forwarding.
- **Flow:** sequence of packets between source-destination pair; flow packets receive identical service at forwarding devices.
- **Flow rules:** instruction set that act on incoming packets (e.g., drop, forward to controller, etc)
- **Flow table:** resides on switches and contains rules to handle flow packets.

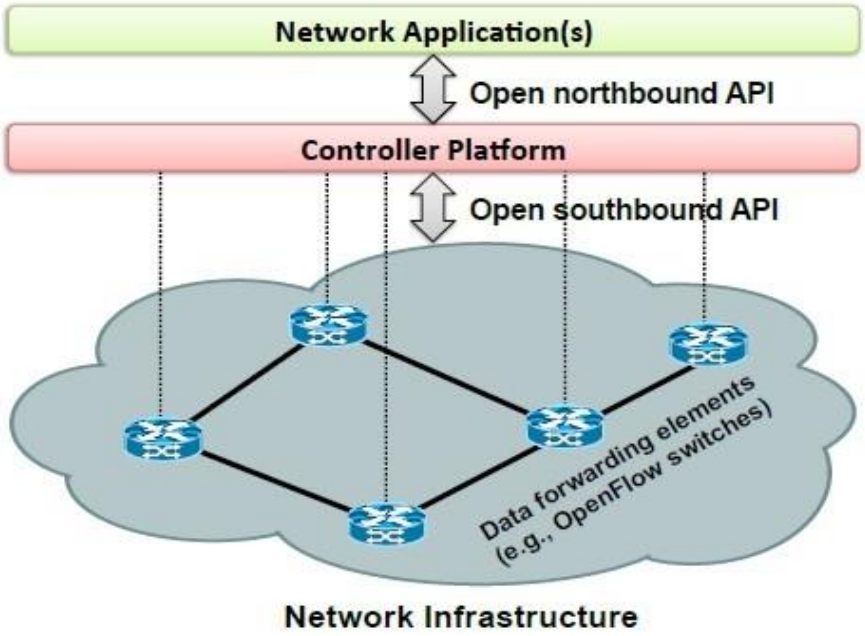


Source:
"Software-Defined Networking: A Comprehensive Survey", Kreutz et al., https://arxiv.org/pdf/1406.0440.

SDN: Definitions, Concepts, and Terminology

› **Control plane:** controls the data plane; logically centralized in the “controller” (a.k.a., network operating system).

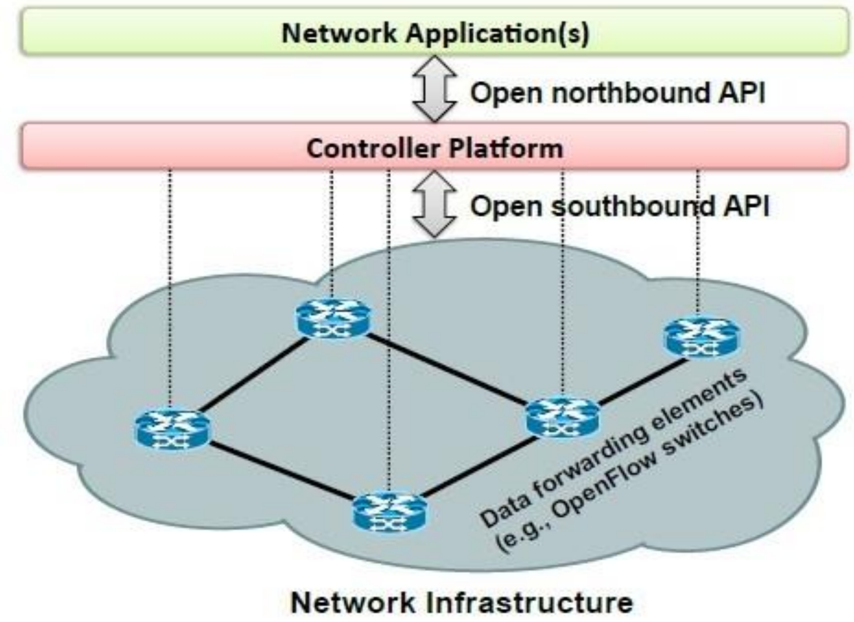
› **Southbound interface:** (instruction set to program the data plane) + (protocol between control- and data planes).



Source:
“Software-Defined Networking: A Comprehensive Survey”, Kreutz et al., https://arxiv.org/pdf/1406.0440.

SDN: Definitions, Concepts, and Terminology

- **Northbound interface:** API offered by control plane to develop network control- and management applications.
- **Management plane:** functions, e.g., routing, traffic engineering, that use control plane functions and API to manage and control network infrastructure.



Source:
"Software-Defined Networking: A Comprehensive Survey",
Kreutz et al., https://arxiv.org/pdf/1406.0440.

SDN Applications

- Traffic engineering:
  - Provide adequate QoS, improve network utilization, reduce power consumption, balance load.
- Wireless network management/control and mobility support:
  - Seamless handover, load balancing, interoperability between heterogeneous networks, dynamic spectrum usage.
- Measurement and monitoring:
  - Packet sampling, traffic matrix estimation
- Security:
  - Firewalling, access control, DoS attack detection/ mitigation, traffic anomaly detection.
- Data-center networking:
  - Data center QOS and traffic engineering, fault detection and resilence, dynamic provisioning, security.

Source: “Software-Defined Networking: A Comprehensive Survey”, Kreutz et al., https://arxiv.org/pdf/1406.0440.

Network Function Virtualization (NFV)

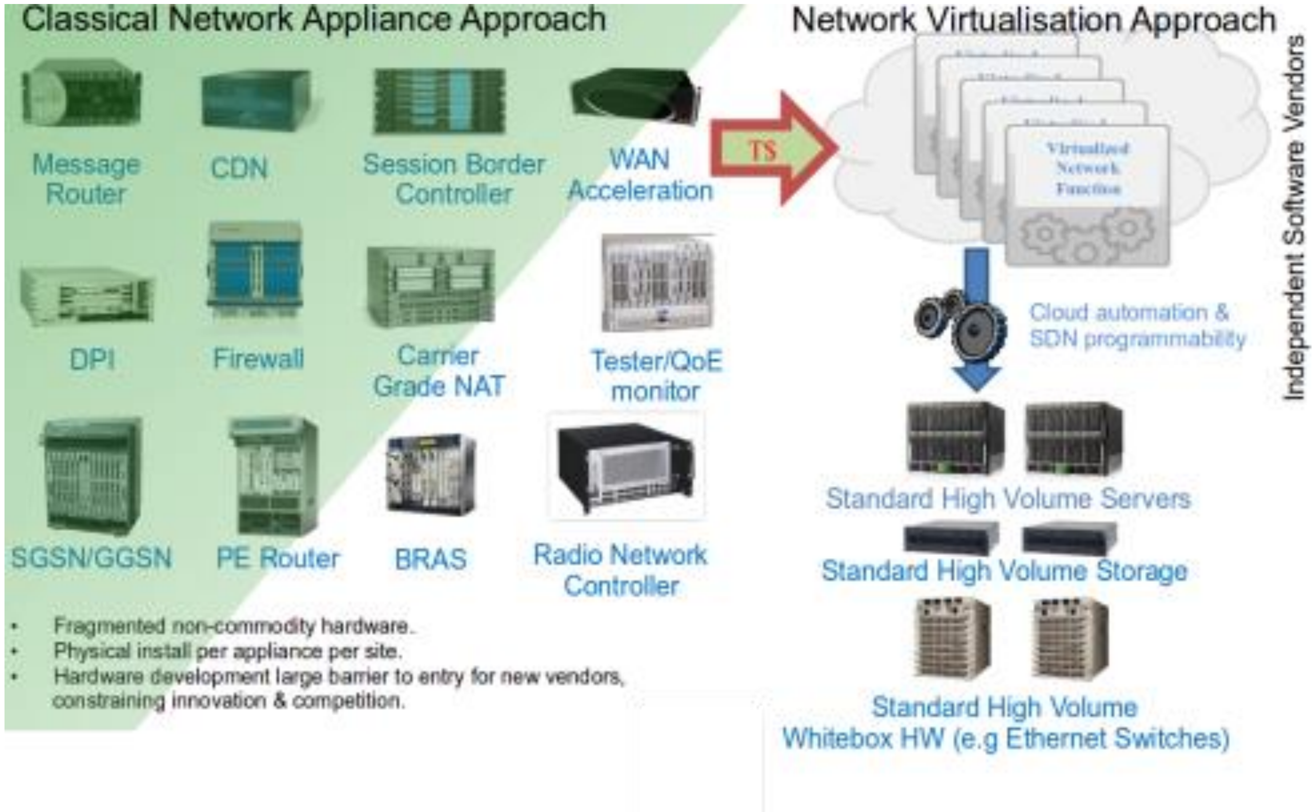
Motivation

Source: Why Virtualization is Essential for 5G – Francis Chow (5G Summit 2015)

Problem Statement

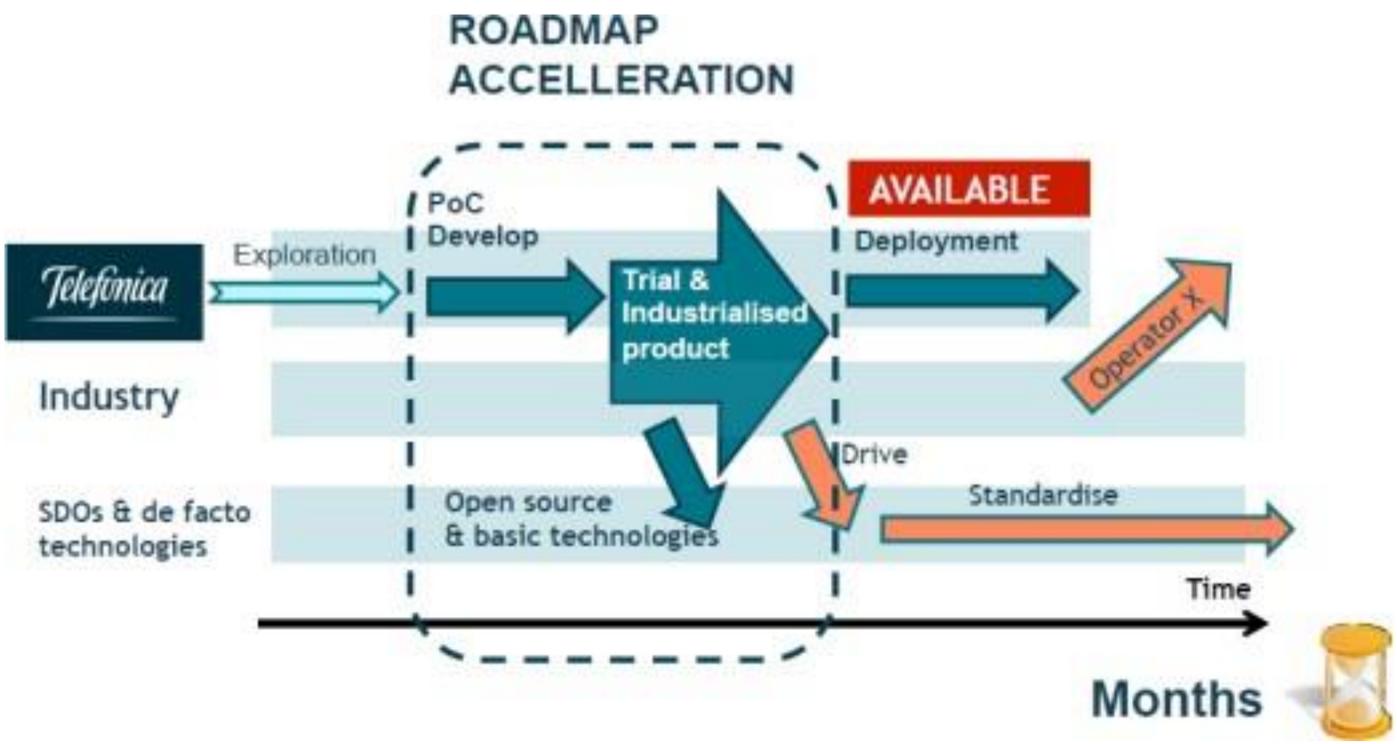
- Complex carrier networks
  - with a large variety of proprietary nodes and hardware appliances
- Launching new services is difficult and takes too long
- Space and power to accommodate
  - requires just another variety of box, which needs to be integrated
- Operation is expensive
  - Rapidly reach end of life due to
    - existing procure-design;
    - integrate-deploy cycle

Some Changes



Source: ETSI NFV ISG – DIRECTION & PRIORITIES – Steven Wright (NFV World Congress 2015)

Transformation



Source: Adapted from D. Lopez Telefonica I+D, NFV

Why NFV?

1.Virtualization: Use network resource without worrying about where it is physically located, how much it is, how it is organized, etc.

2. Orchestration: Manage thousands of devices

3. Programmable: Should be able to change behavior on the fly.

4. Dynamic Scaling: Should be able to change size, quantity, as a F(load)

5. Automation: Let machines / software do humans' work

6. Visibility: Monitor resources, connectivity

7. Performance: Optimize network device utilization

8. Multi-tenancy: Slice the network for different customers (as-a-Service)

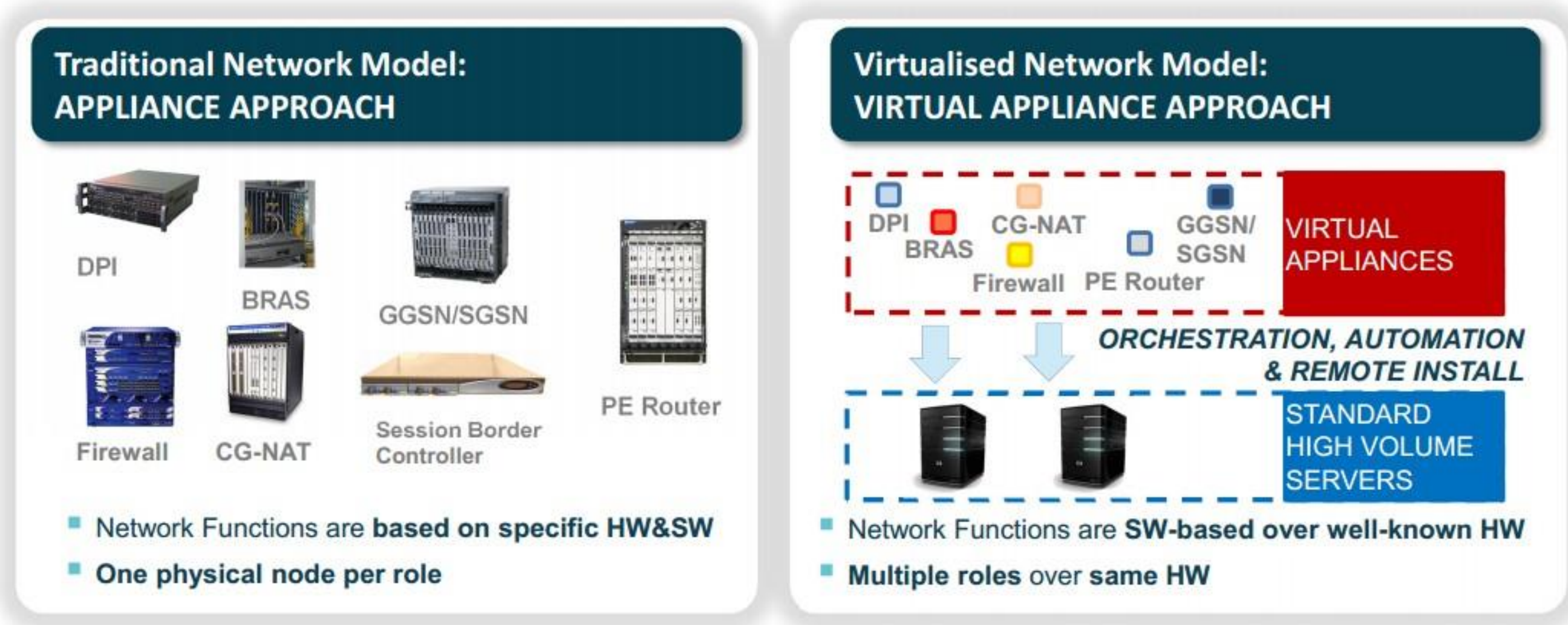
9. Service Integration: Let network management play nice with OSS/BSS

10. Openness: Full choice of modular plug-ins

Note: These are exactly the same reasons why we need/want SDN.

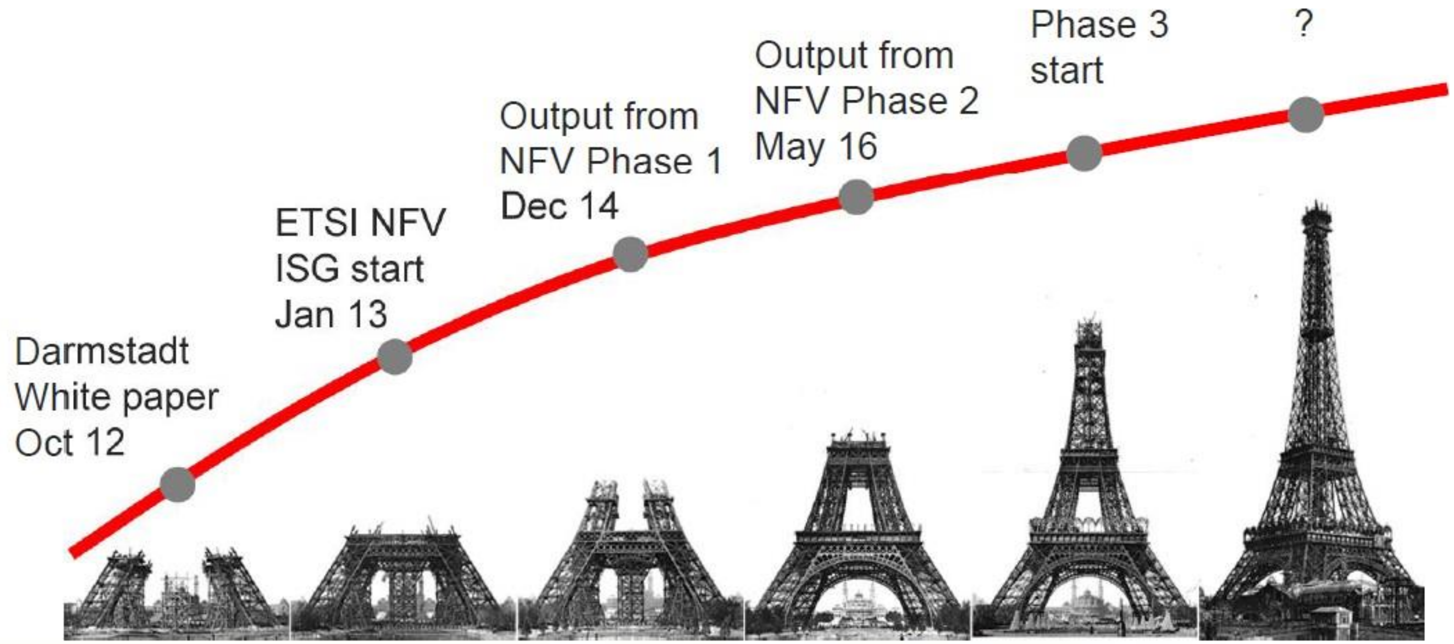
The NFV Concept

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



Source: Adapted from D. Lopez Telefonica I+D, NFV

The Making of NFV



Souce: NFV Orchestration | Fueling innovation in operator networks – Federico Descalzo (TM FORUM 2016)

Benefits & Promises of NFV

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

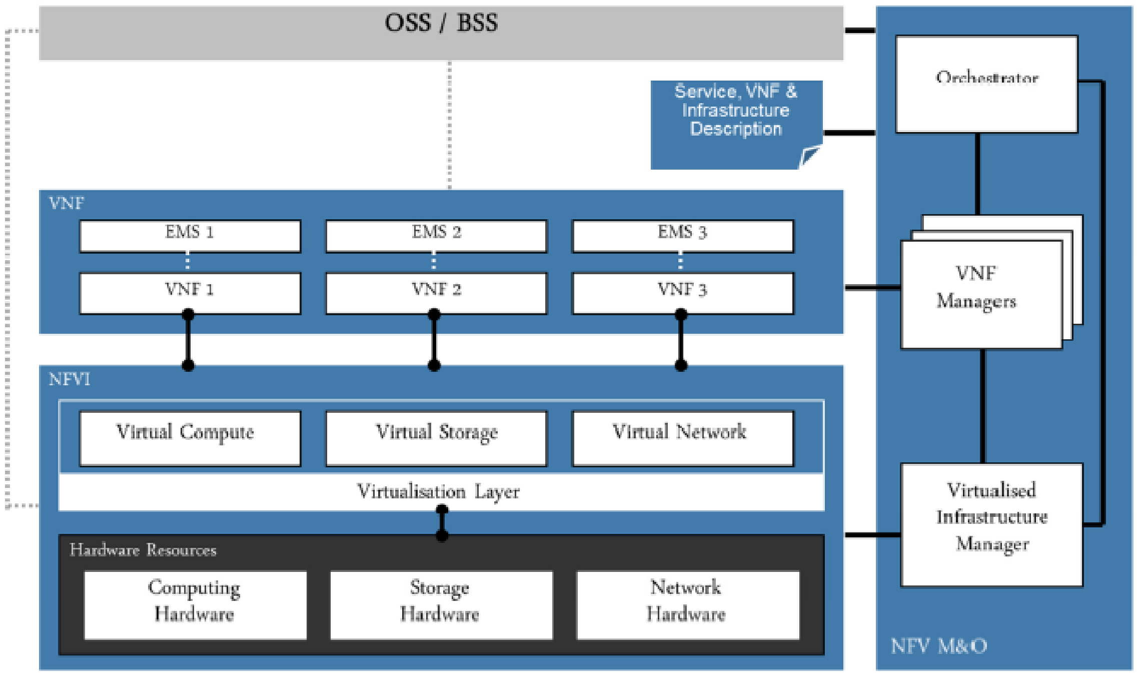
Source: NFV

Benefits & Promises of NFV

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

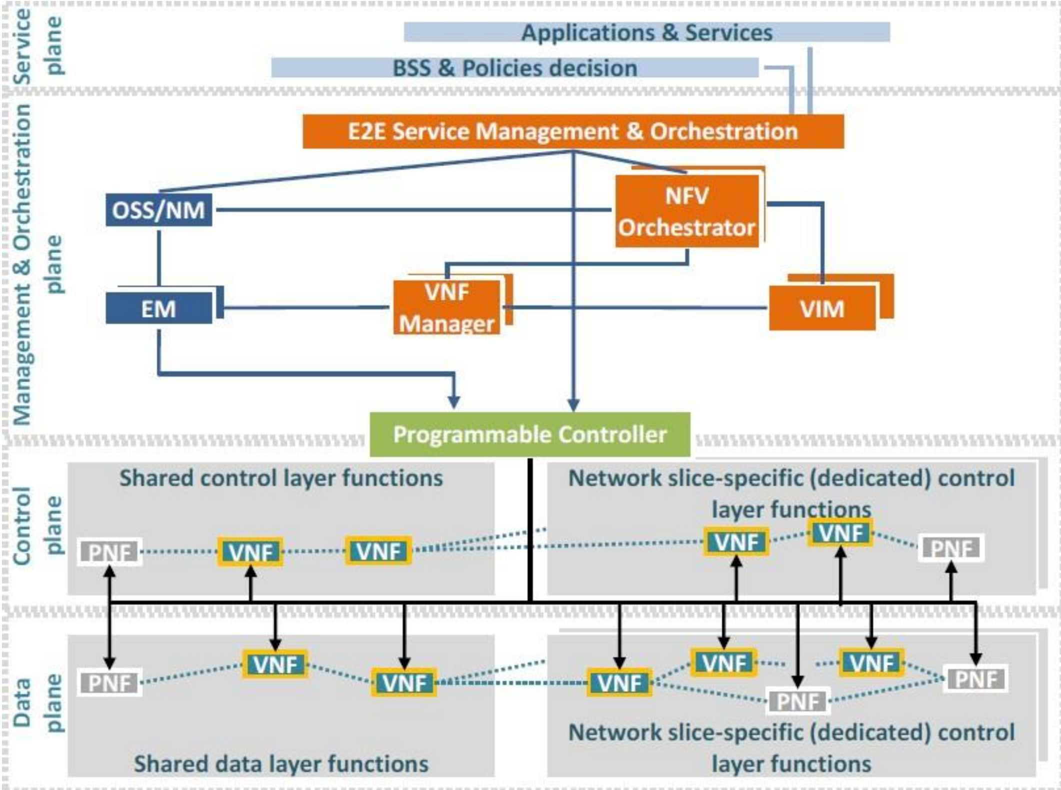
Source: Adapted from D. Lopez Telefonica I+D, NFV

ETSI NFV Architectural Framework



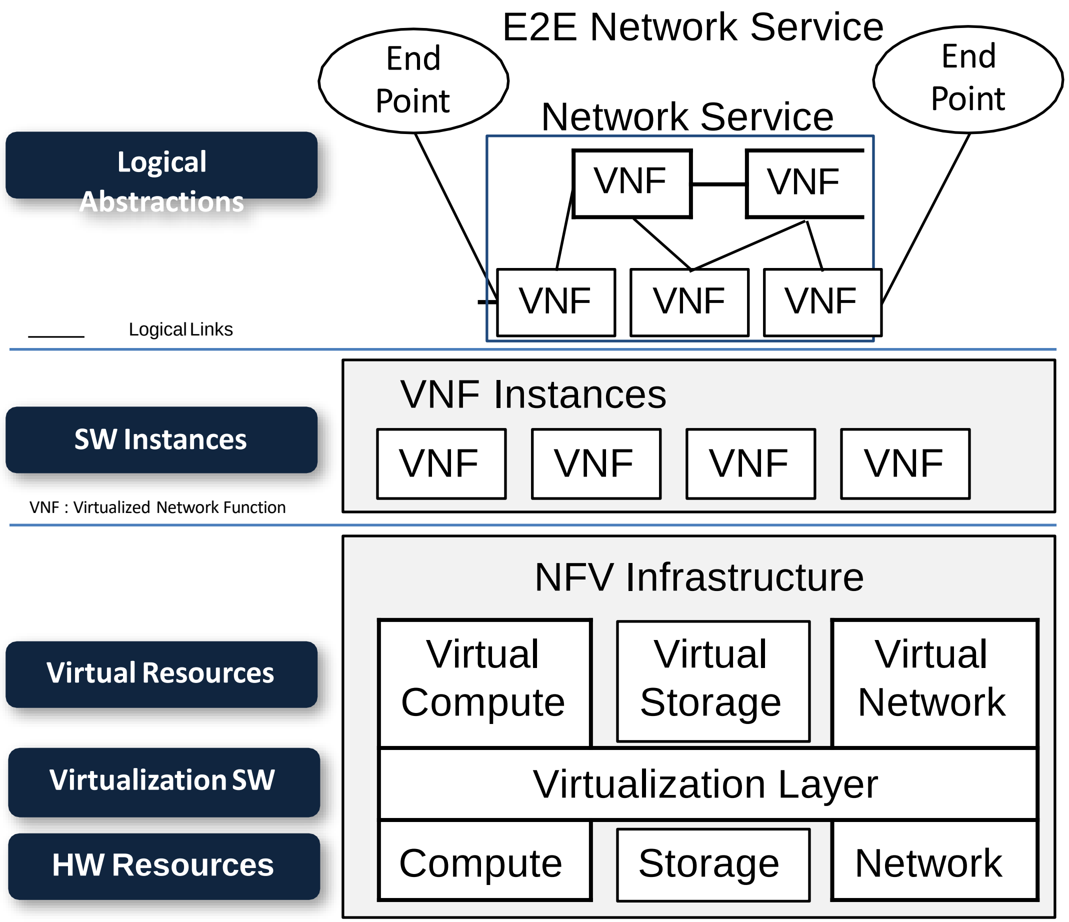
Souce: ETSI NFV White Paper 2

NFV



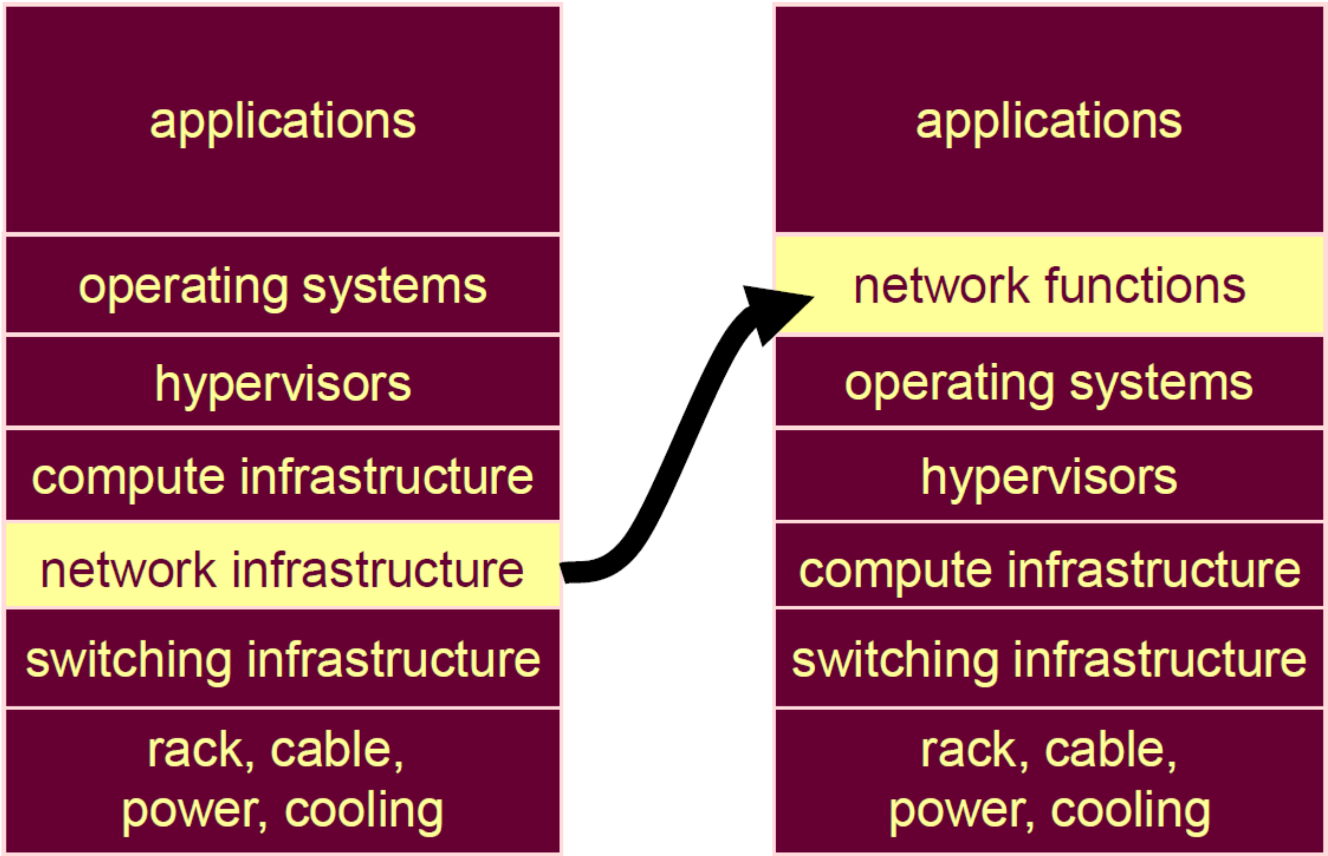
Source: View on 5G Architecture - 5G PPP Architecture Working Group (2016)

NFV Layers



Source: Adapted from D. Lopez Telefonica I+D, NFV

Rethinking relayering



NFV Concepts

- **Network Function (NF):** Functional building block with well defined interfaces and well defined functional behavior
- **Virtualized Network Function (VNF):** Software implementation of NF that can be deployed in a virtualized infrastructure
- **VNF Set:** Connectivity between VNFs is not specified, e.g., residential gateways
- **VNF Forwarding Graph:** Service chain when network connectivity order is important, e.g., firewall, NAT, load balancer
- **NFV Infrastructure (NFVI):** Hardware and software required to deploy, manage and execute VNFs including computation, networking, and storage.
- **NFV Orchestrator:** Automates the deployment, operation, management, coordination of VNFs and NFVI.

Source: Adapted from Raj Jain

NFV Concepts

- **NFVI Point of Presence (PoP):** Location of NFVI
- **NFVI-PoP Network:** Internal network
- **Transport Network:** Network connecting a PoP to other PoPs or external networks
- **VNF Manager:** VNF lifecycle management e.g., instantiation, update, scaling, query, monitoring, fault diagnosis, healing, termination
- **Virtualized Infrastructure Manager:** Management of computing, storage, network, software resources
- **Network Service:** A composition of network functions and defined by its functional and behavioral specification
- **NFV Service:** A network services using NFs with at least one VNF.

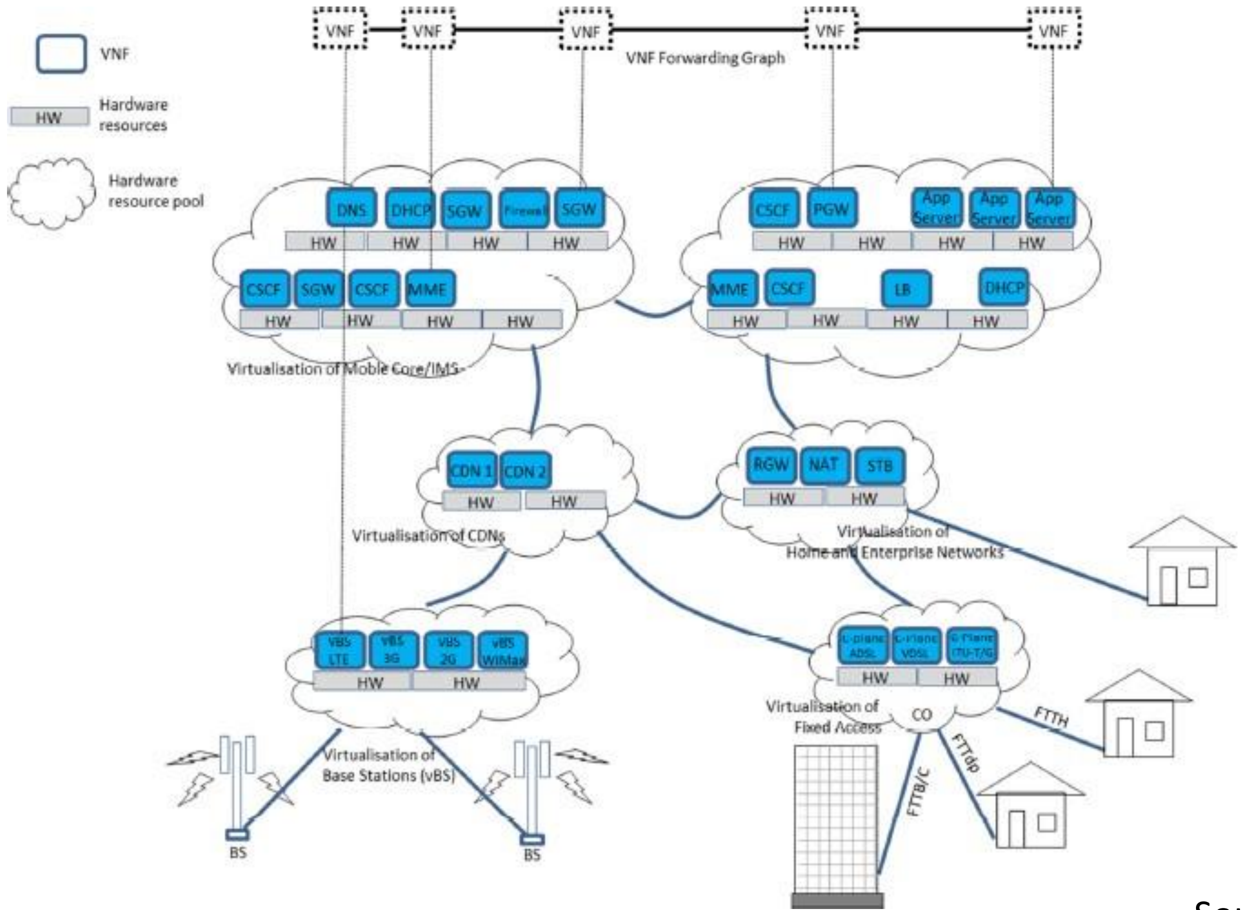
Source: Adapted from Raj Jain

NFV Concepts

- **User Service:** Services offered to end users/customers/subscribers.
- **Deployment Behavior:** NFVI resources that a VNF requires, e.g., Number of VMs, memory, disk, images, bandwidth, latency
- **Operational Behavior:** VNF instance topology and lifecycle operations, e.g., start, stop, pause, migration, …
- **VNF Descriptor:** Deployment behavior + Operational behavior

Source: Adapted from Raj Jain

Overview of ETSI NFV Use Cases



Souce: ETSI NFV White Paper 2

Architectural Use Cases

- Network Functions &Virtualisation Infrastructure as a Service
  - Network functions go cloudlike
- Virtual Network Function as a Service
  - Ubiquitous, delocalized network functions
- Virtual Network Platform as a Service
  - Applying multi-tenancy at the VNF level
- VNF Forwarding Graphs
  - Building E2E services by composition

Service-Oriented Use Cases

- Mobile core network and IMS
  - Elastic, scalable, more resilient EPC
  - Specially suitable for a phased approach
- Mobile base stations
  - Evolved Cloud-RAN
  - Enabler for SON
- Home environment
  - L2 visibility to the home network
  - Smooth introduction of residential services
- CDNs
  - Better adaptability to traffic surges
  - New collaborative service models
- Fixed access network
  - Offload computational intensive optimization
  - Enable on-demand access services

NFV Framework Requirements

1. General: Partial or full Virtualization, Predictable performance

2. Portability: Decoupled from underlying infrastructure

3.Performance: Conforming and proportional to NFs specifications and facilities to monitor

4. Elasticity: Scalable to meet SLAs. Movable to other servers.

5. Resiliency: Be able to recreate after failure.

Specified packet loss rate, calls drops, time to recover, etc.

6. Security: Role-based authorization, authentication

7. Service Continuity: Seamless or non-seamless continuity after failures or migration

NFV Framework Requirements

8.Service Assurance: Time stamp and forward copies of packets for Fault detection

9.Energy Efficiency Requirements: Should be possible to put a subset of VNF in a power conserving sleep state

10.Operational and Management Requirements: Incorporate mechanisms for automation of operational and management functions

11.Transition: Coexistence with Legacy and Interoperability among multi-vendor implementations

12.Service Models: Operators may use NFV infrastructure operated by other operators

...

Challenging Path upfront:

Not as simple as cloud applied to telco

The network differs from the computing environment in 2 key factors…

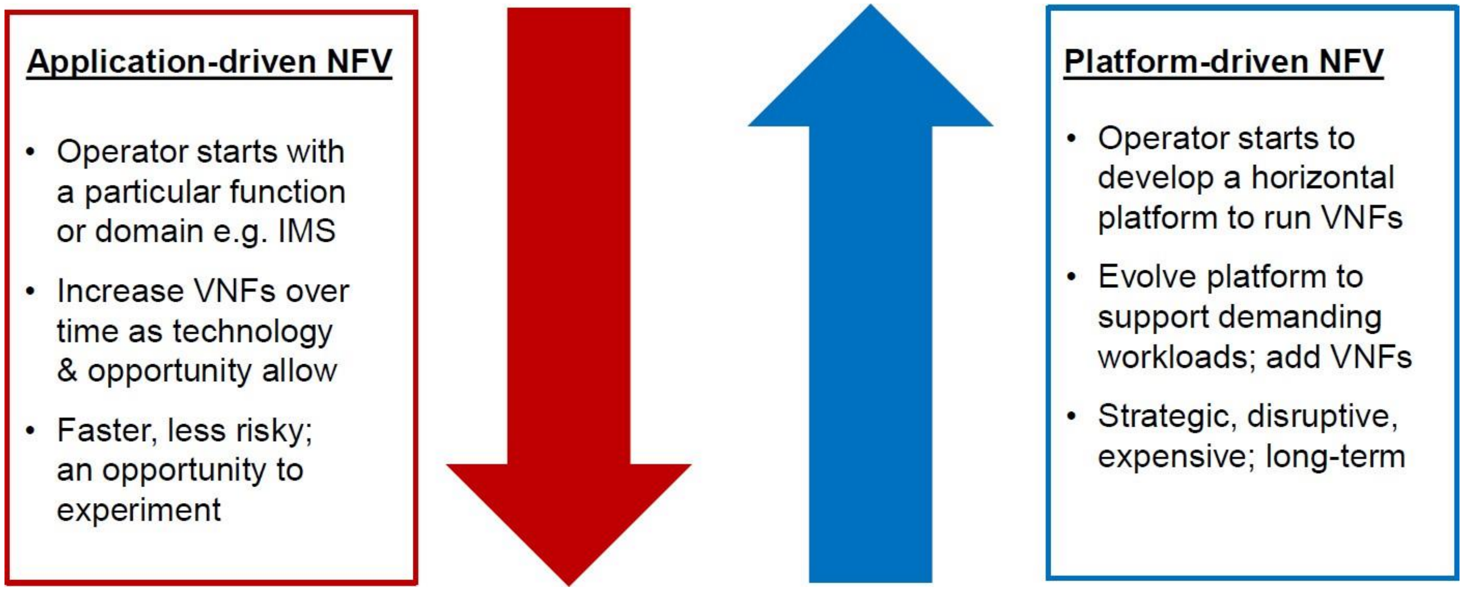
1. **Data plane workloads** (which are huge!) → HIGH PRESSURE ON **PERFORMANCE**
2. **Network requires shape** (+ E2E interconnection) → GLOBAL NETWORK VIEW IS REQUIRED FOR **MANAGEMENT**

…which are big challenges for vanilla cloud computing.

AN ADAPTED VIRTUALISATION ENVIRONMENT IS NEEDED
TO OBTAIN **CARRIER-CLASS** BEHAVIOUR

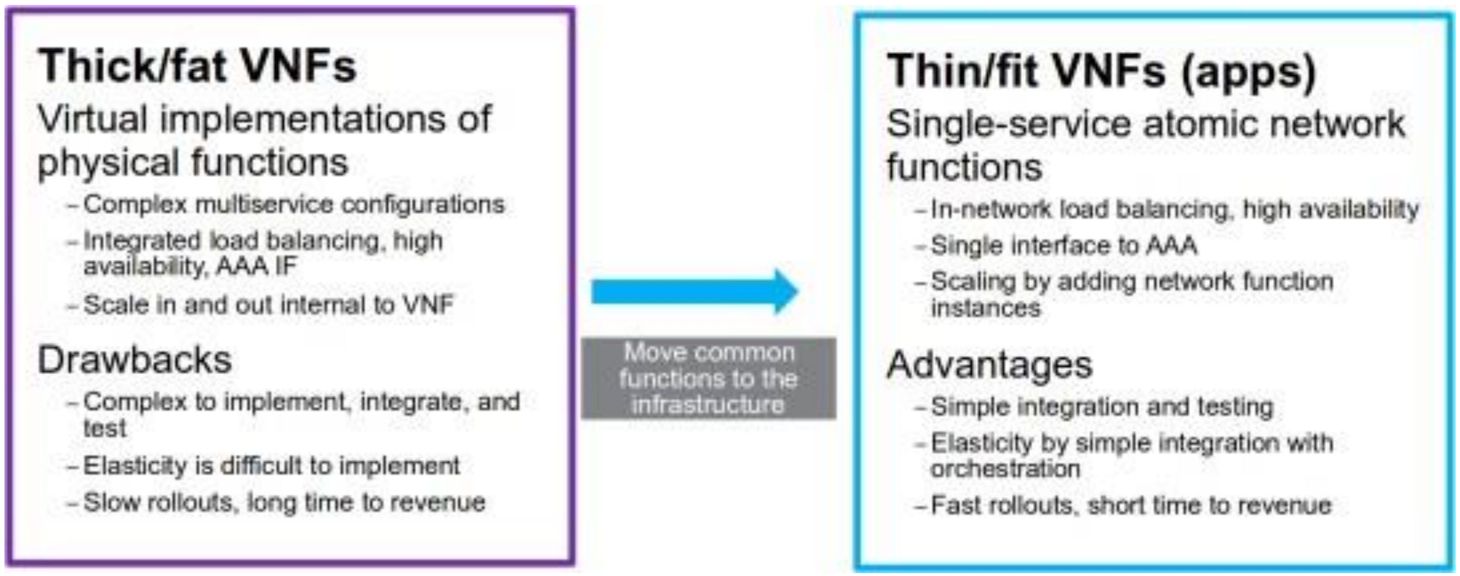
Source: Adapted from D. Lopez Telefonica I+D, NFV

The Road to NFV



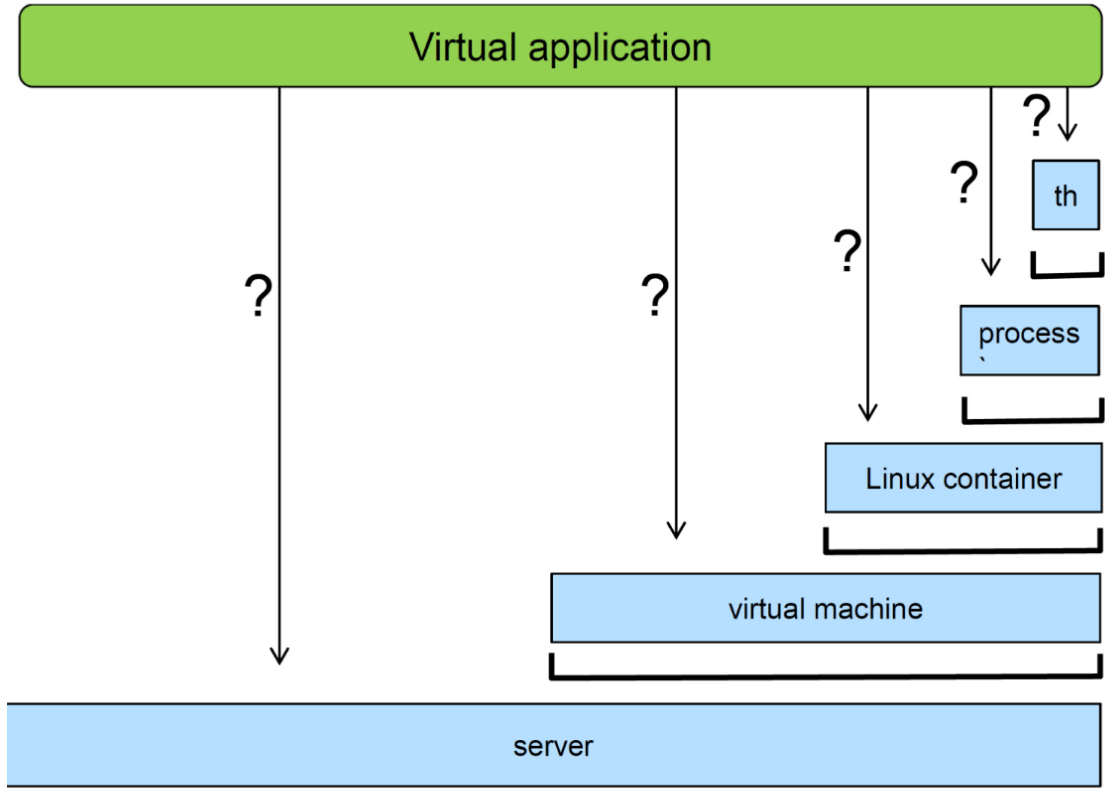
Source: Gabriel Brown, Heavy Reading

Fat vs. fit VNFs

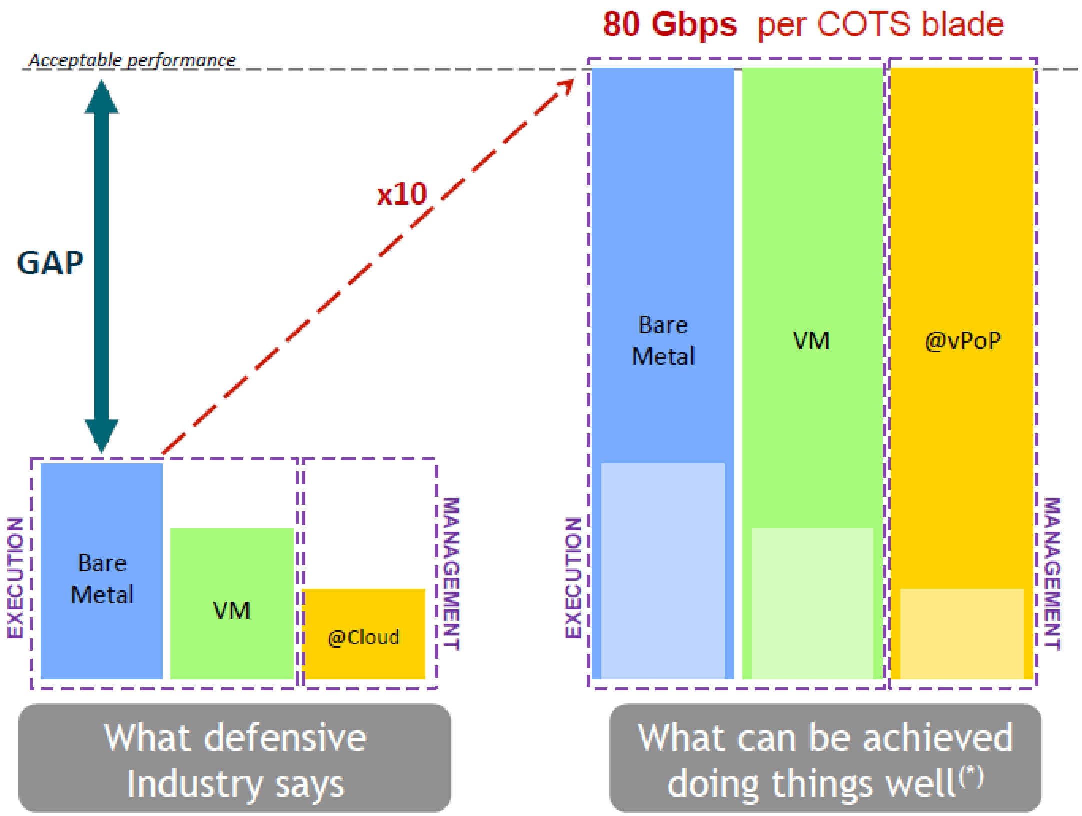


Source: SDN and NFV Stepping Stones to the Telco Cloud – Prodip Sen (ONS 2016)

Alternative options to virtualize NFV apps

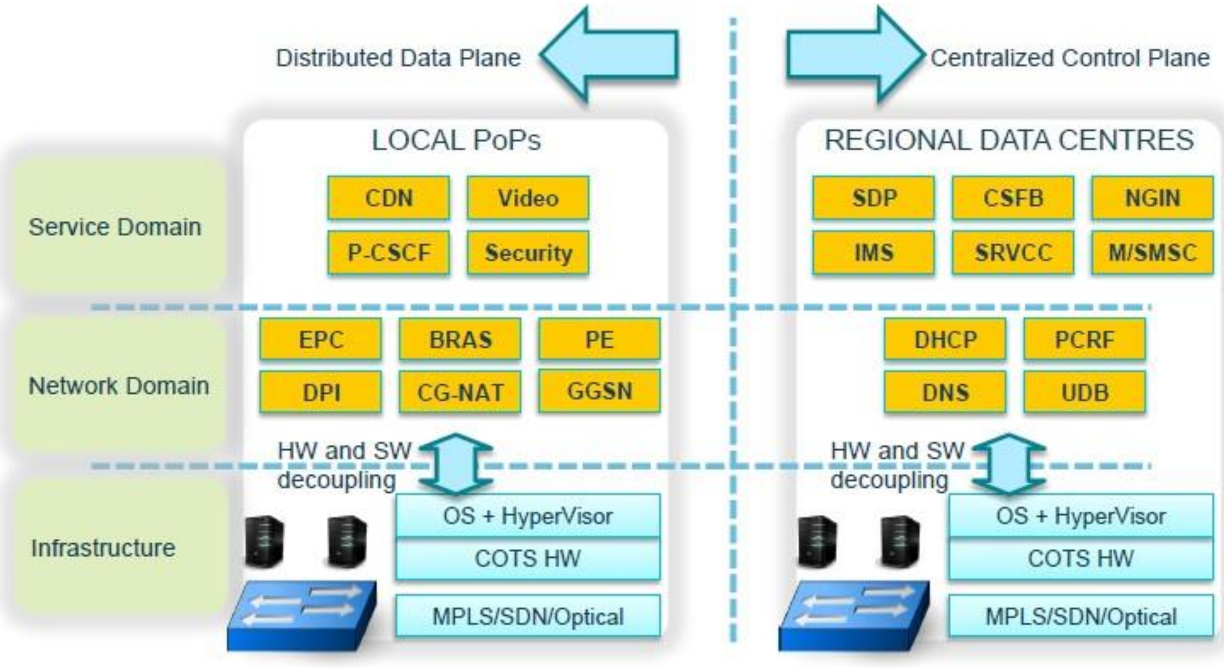


Performance Challenges



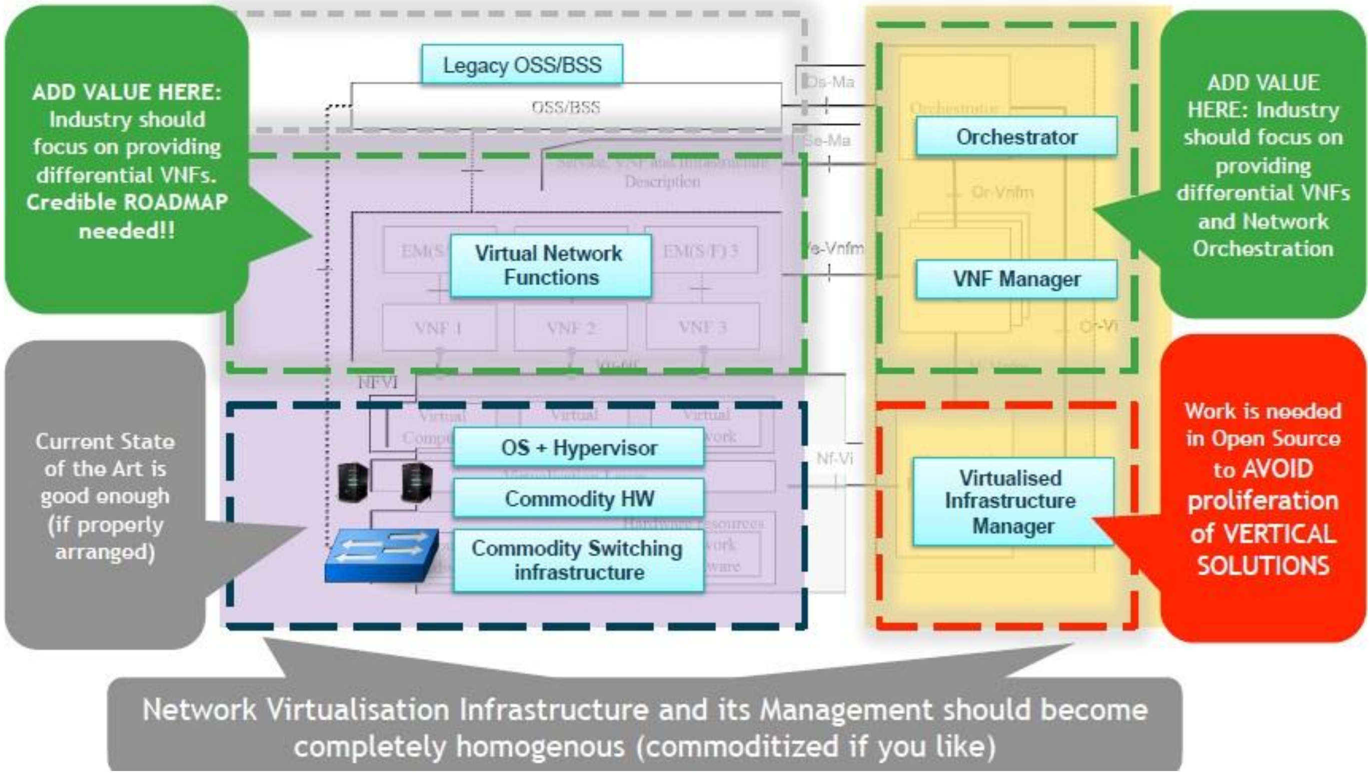
Souce: ETSI NFV White Paper 2

Portability Challenges



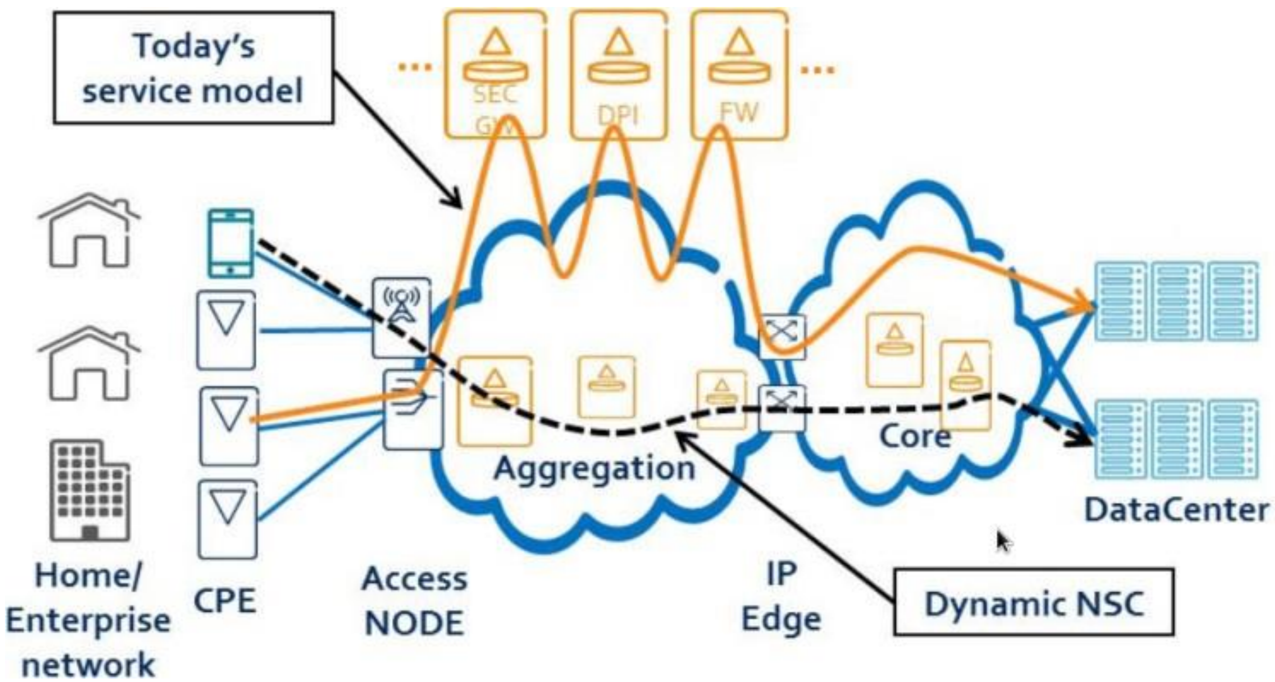
Source: Adapted from D. Lopez Telefonica I+D, NFV

Integration Challenges



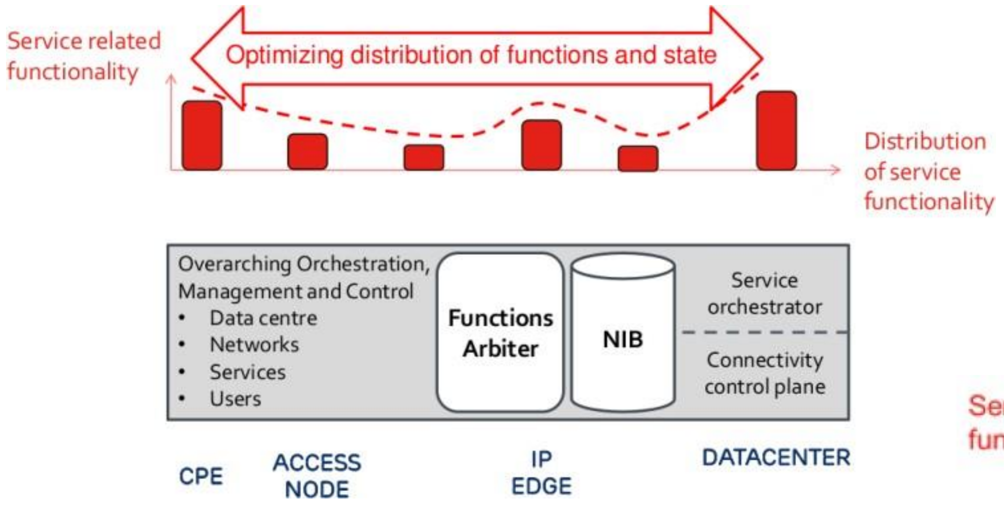
Source: Adapted from D. Lopez Telefonica I+D, NFV

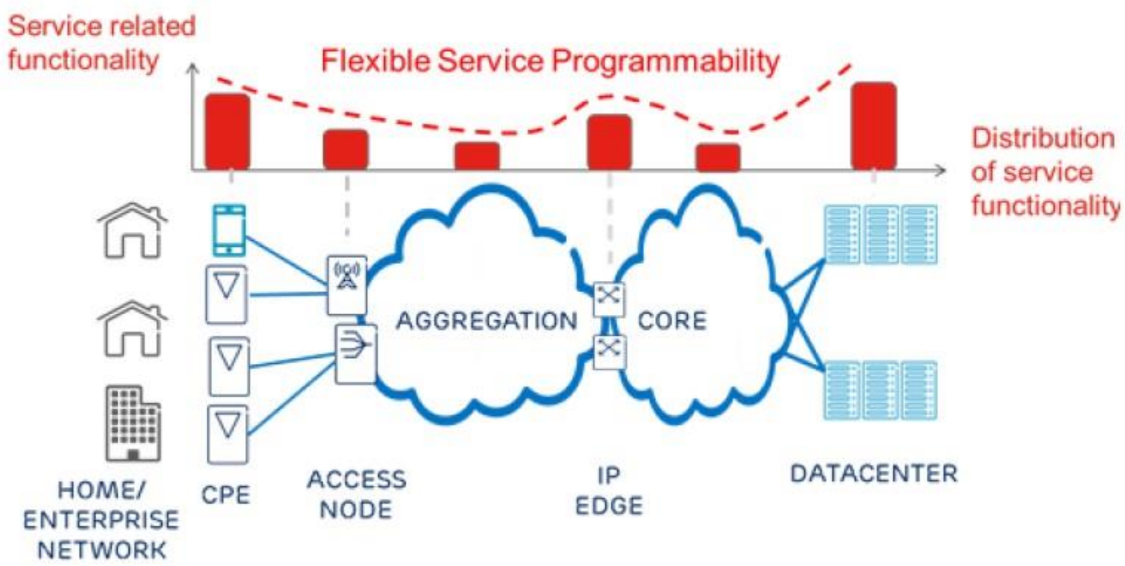
Elasticity Challenges



Source: UNIFY Project Results

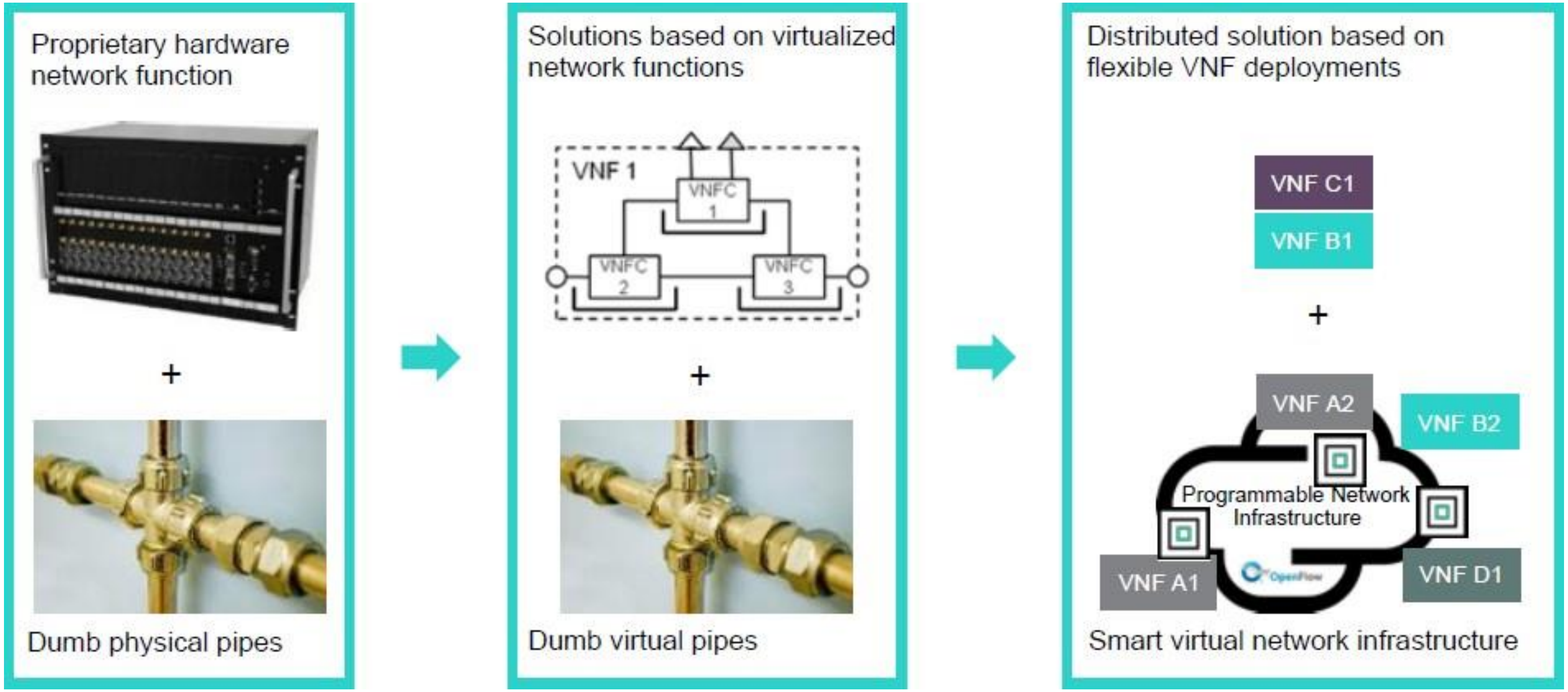
Management & Orchestration Challenges





Source: IEEE PIMRC, 2013

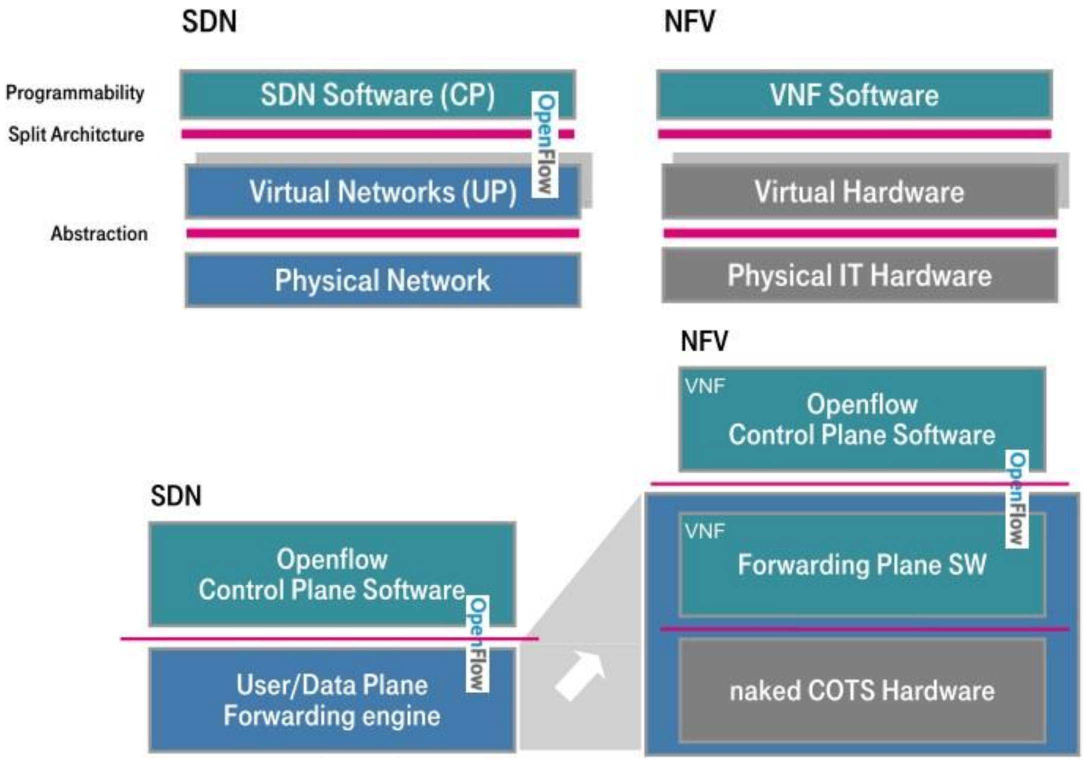
NFV & SDN



Source: SDN and NFV Stepping Stones to the Telco Cloud – Prodip Sen (ONS 2016)

SDN & NFV

- SDN and NFV do NOT depend on each other



Source: Uwe Michel, T-Systems

NFV vs SDN

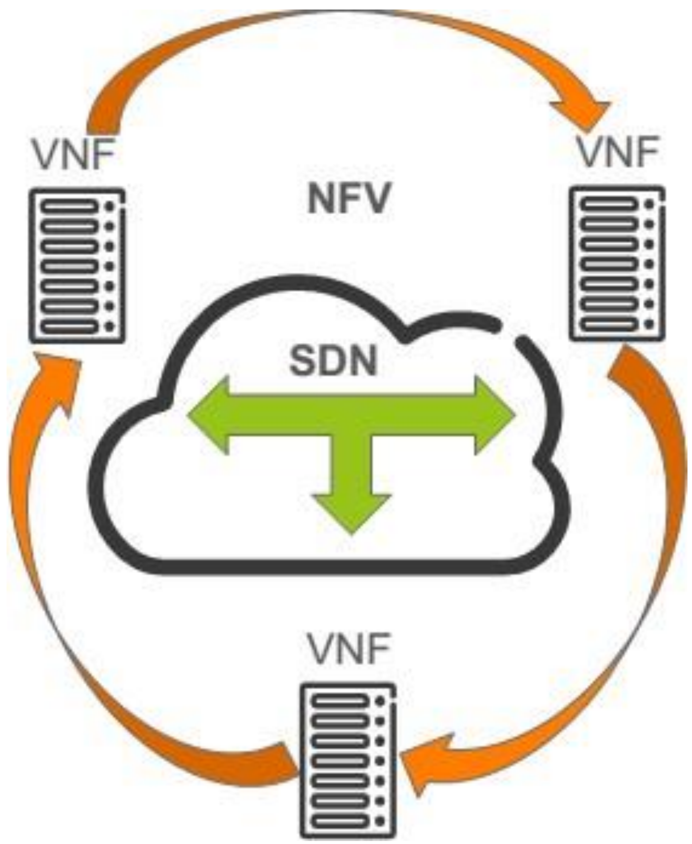
SDN ››› flexible forwarding & steering of traffic in a physical or virtual network environment

[Network Re-Architecture]

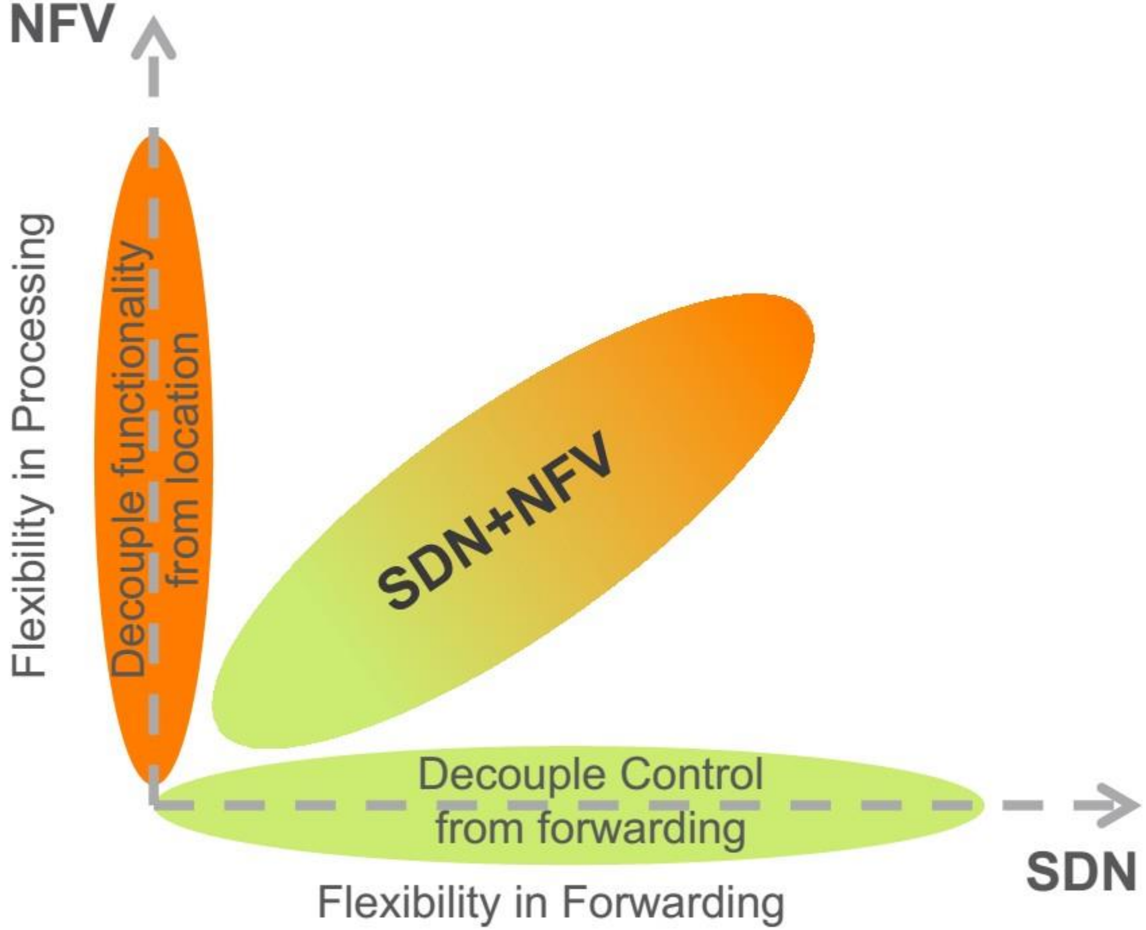
NFV ››› flexible placement of virtualized network functions across the network & cloud

[Appliance Re-Architecture] (initially)

››› SDN & NFV are complementary tools for achieving full network programmability

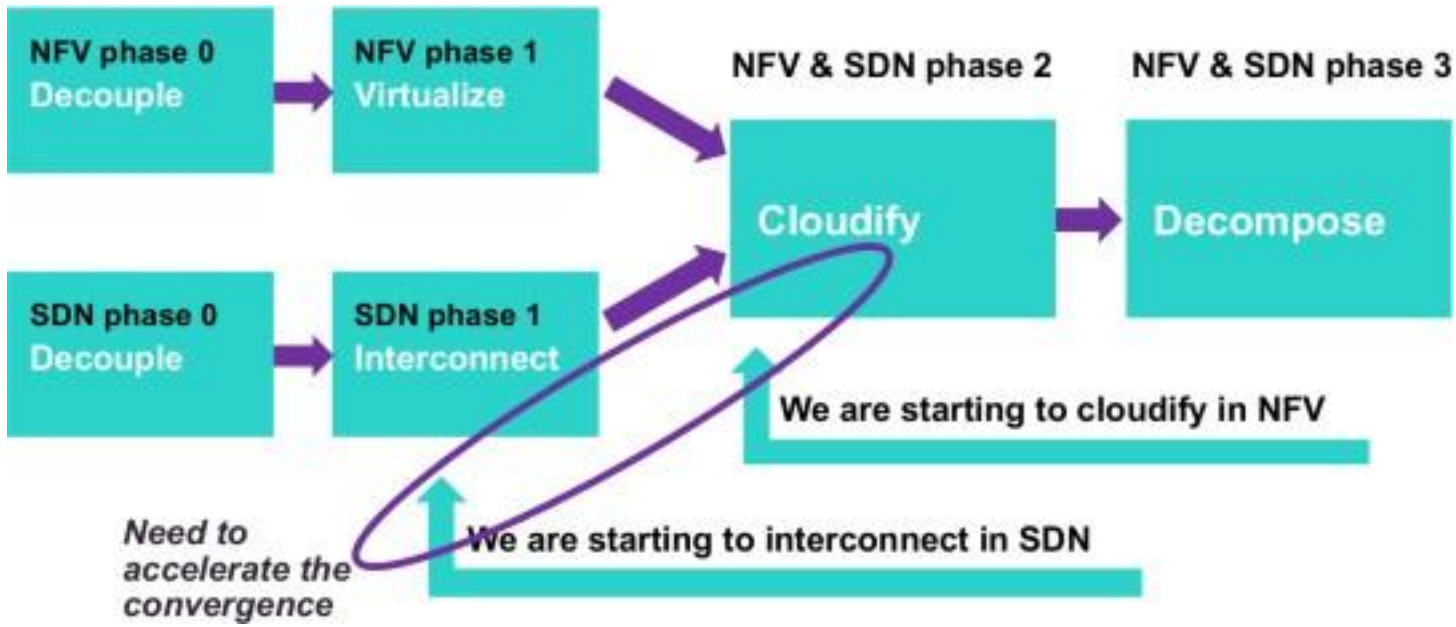


Flexibility with SDN & NFV



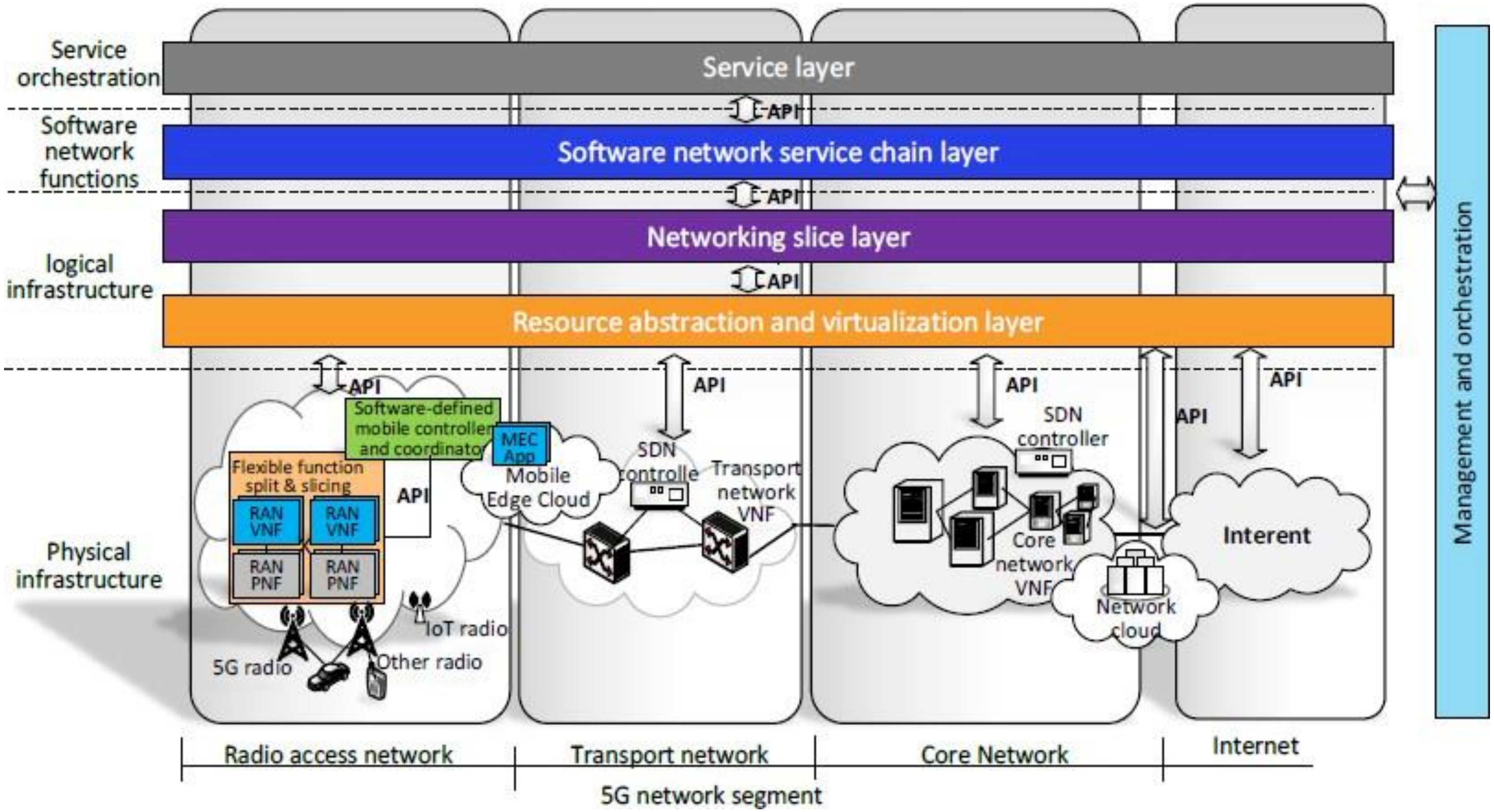
Source: Ahmad Rostami, Ericsson Research (Kista): http://www.itc26.org/fileadmin/ITC26_files/ITC26-Tutorial-Rostami.pdf

SDN & NFV Convergence



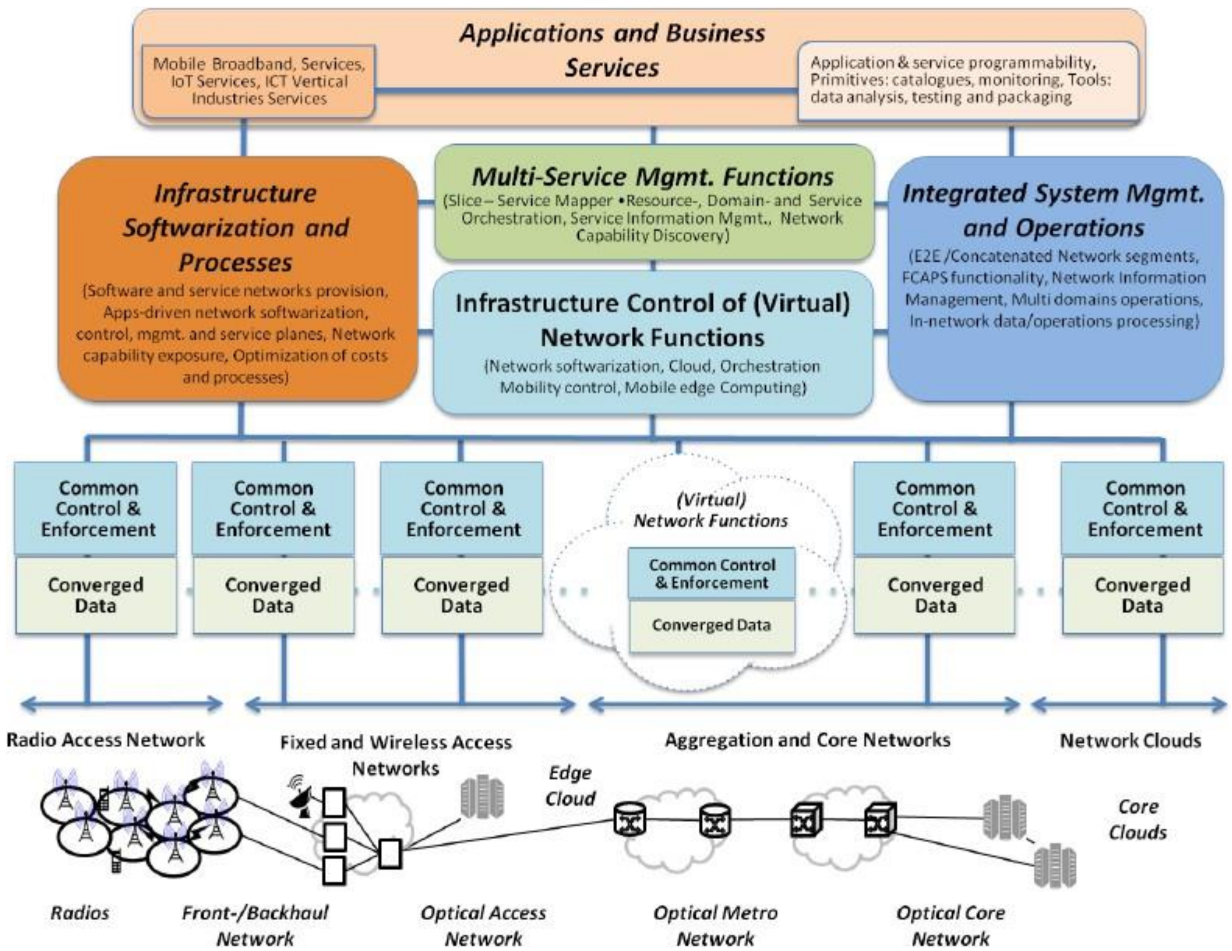
Source: SDN and NFV Stepping Stones to the Telco Cloud – Prodip Sen (ONS 2016)

5G + (NFV & SDN)



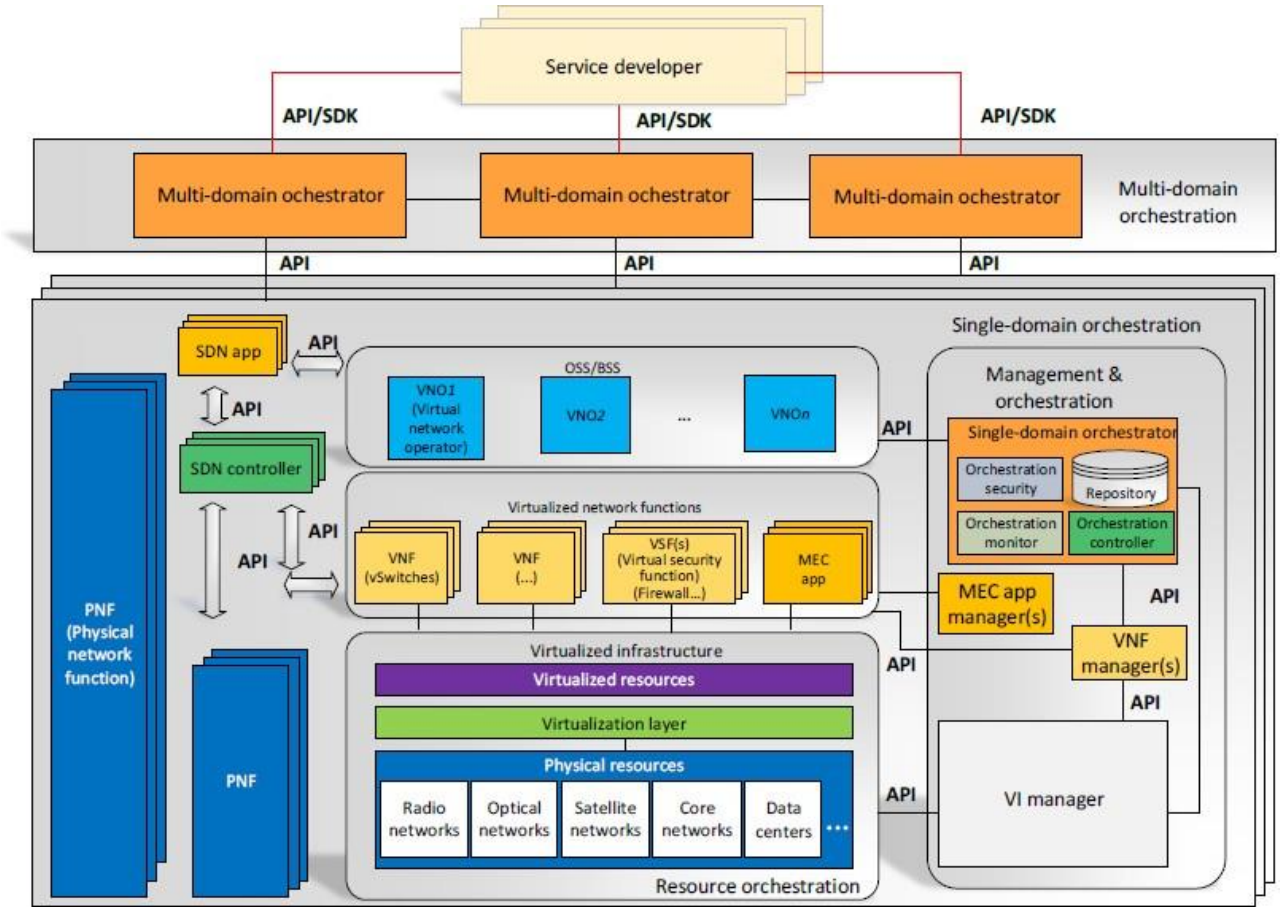
Source: View on 5G Architecture - 5G PPP Architecture Working Group (2016)

5G Softwarization and Programmability Framework



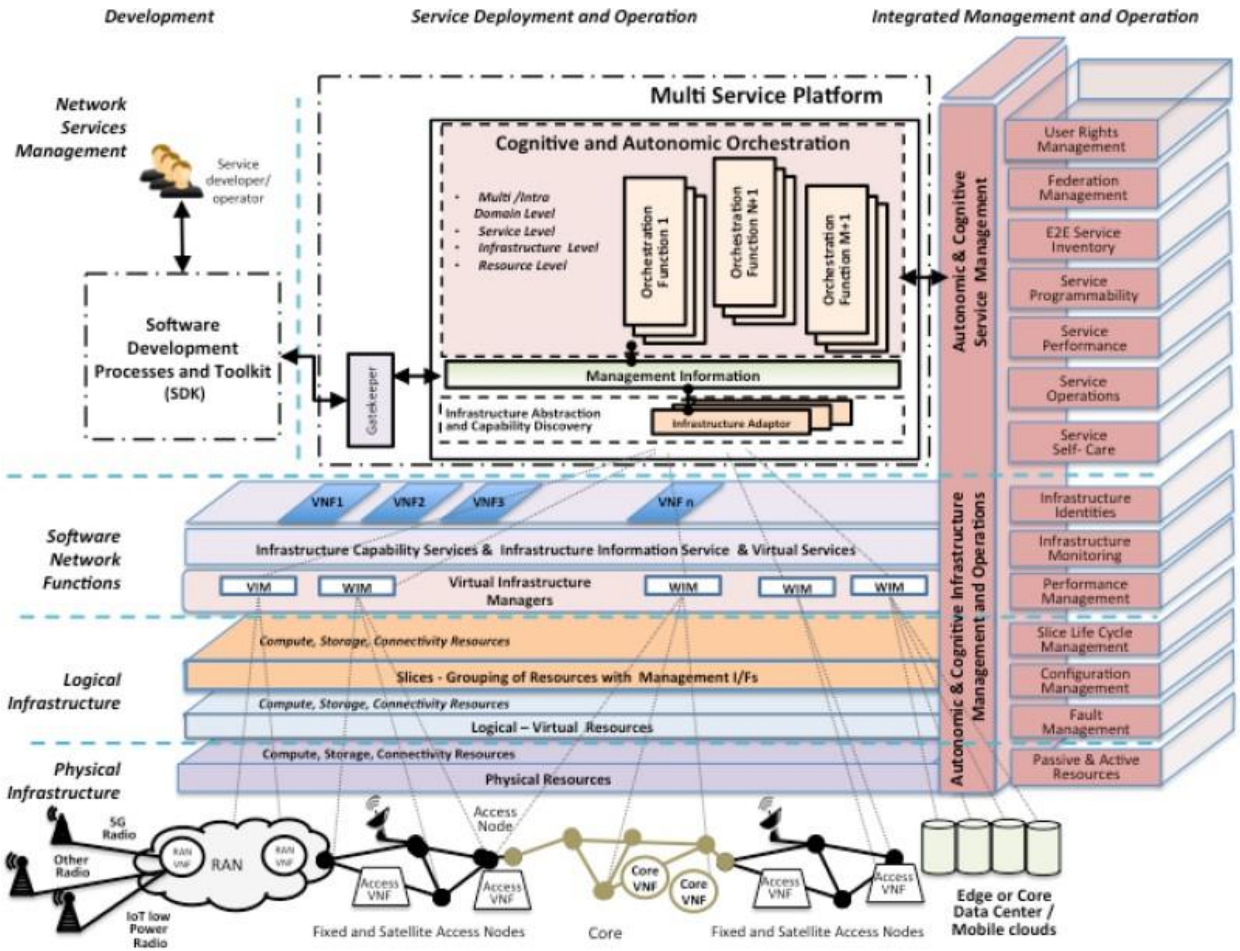
Source: View on 5G Architecture - 5G PPP Architecture Working Group (2016)

Multi Administrative Domains



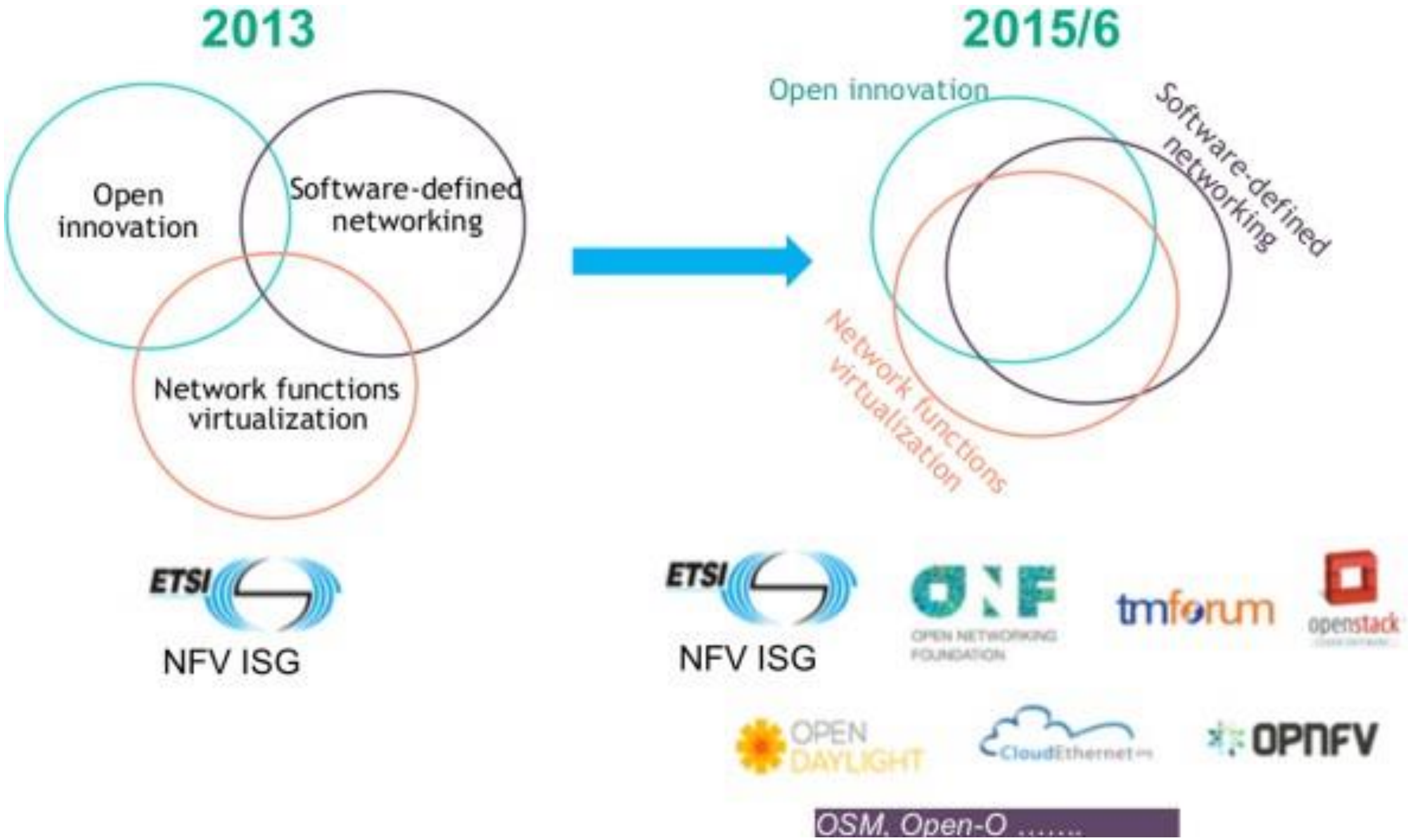
Source: View on 5G Architecture - 5G PPP Architecture Working Group (2016)

The Big Picture



Source: View on 5G Architecture - 5G PPP Architecture Working Group (2016)

SDN/NFV Open Innovation



How important is SDN to NFV?

Very important: **69%**

Somewhat important: **30%**

Not at all important: 1%

(Customer Survey 2015)

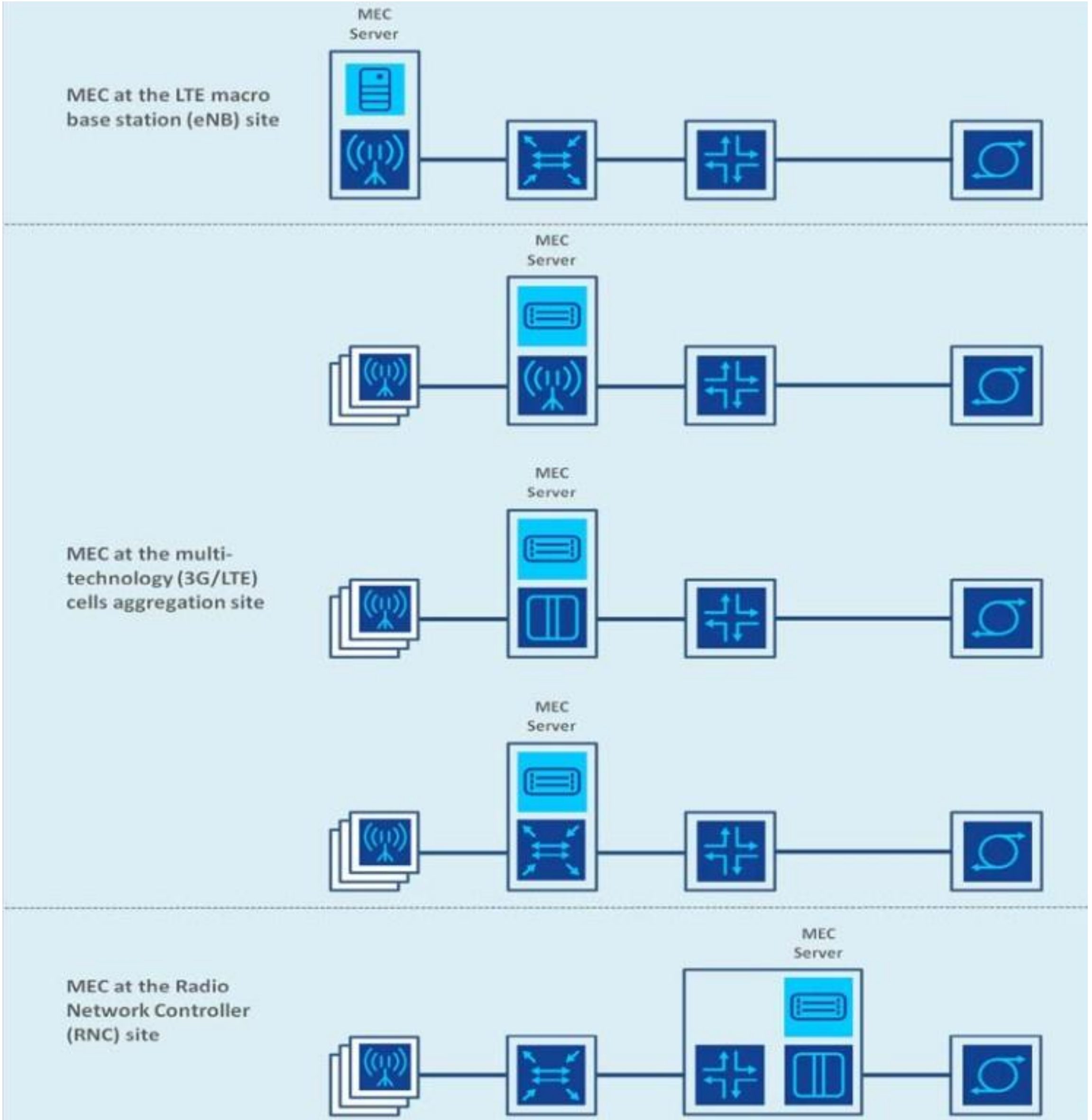
Source: SDN and NFV Stepping Stones to the Telco Cloud – Prodip Sen (ONS 2016)

NFV enables MEC: Mobile Edge Computing

- MEC provides IT and cloud-computing capabilities within the RAN in close proximity to mobile subscribers to accelerate content, services and applications so increasing responsiveness from the edge .
- Standardization bodies: ETSI, 3GPP, ITU-T
- RAN edge offers a service environment with ultra-low latency and high bandwidth as well as direct access to real-time radio network information (subscriber location, cell load, etc.) useful for applications and services to offer context-related services
- Operators can open the radio network edge to third-party partners
- Proximity, context, agility and speed can create value and opportunities for mobile operators, service and content providers, Over the Top (OTT) players and Independent Software Vendors (ISVs)
- Source: https://portal.etsi.org/Portals/0/TBpages/MEC/Docs/Mobile-edge_Computing_-_Introductory_Technical_White_Paper_V1%2018-09-14.pdf

MEC: Mobile Edge Computing

Deployment scenarios of the Mobile-edge Computing server



- Source: https://portal.etsi.org/Portals/0/TBpages/MEC/Docs/Mobile-edge_Computing_-_Introductory_Technical_White_Paper_V1%2018-09-14.pdf

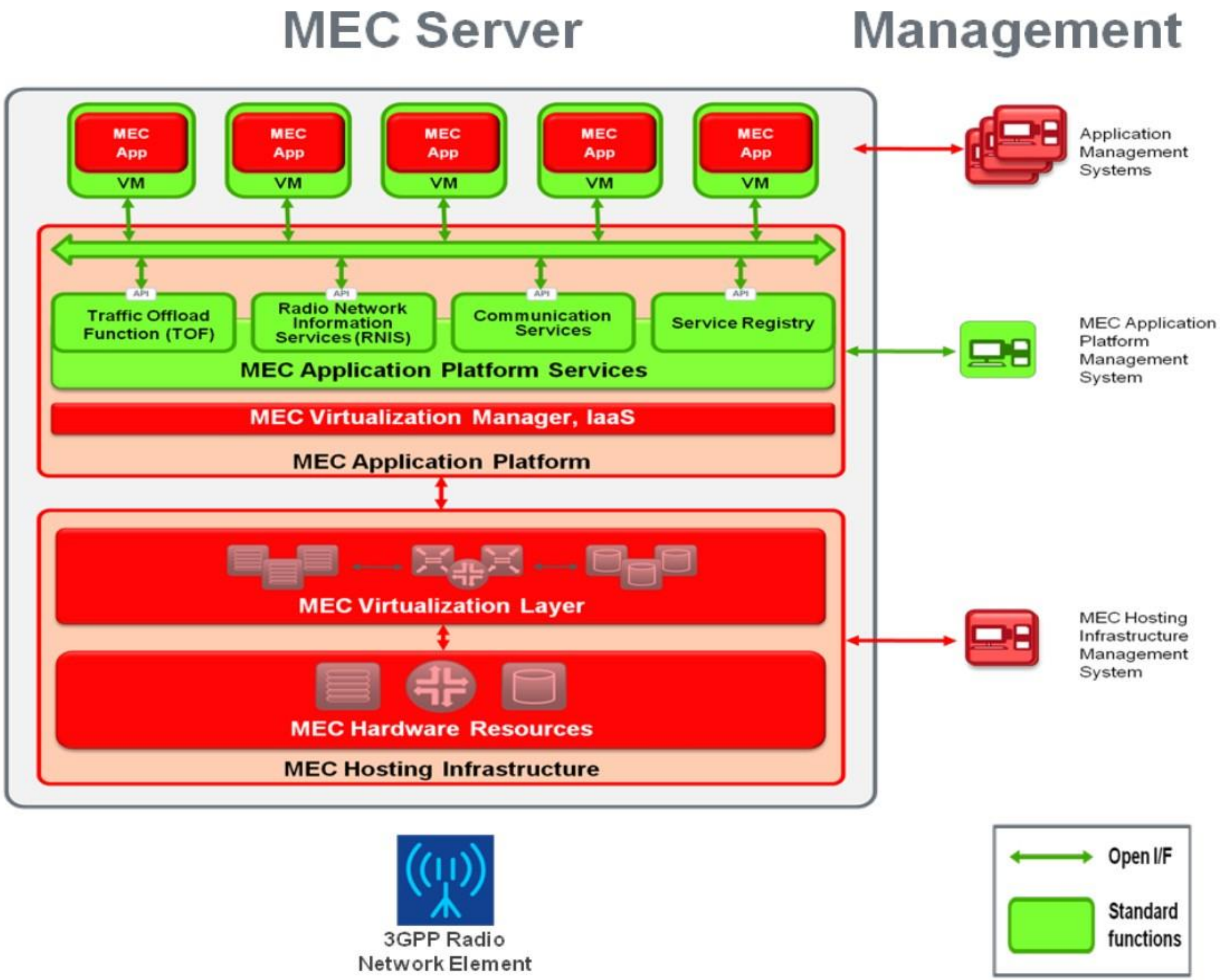
MEC:

Mobile

Edge

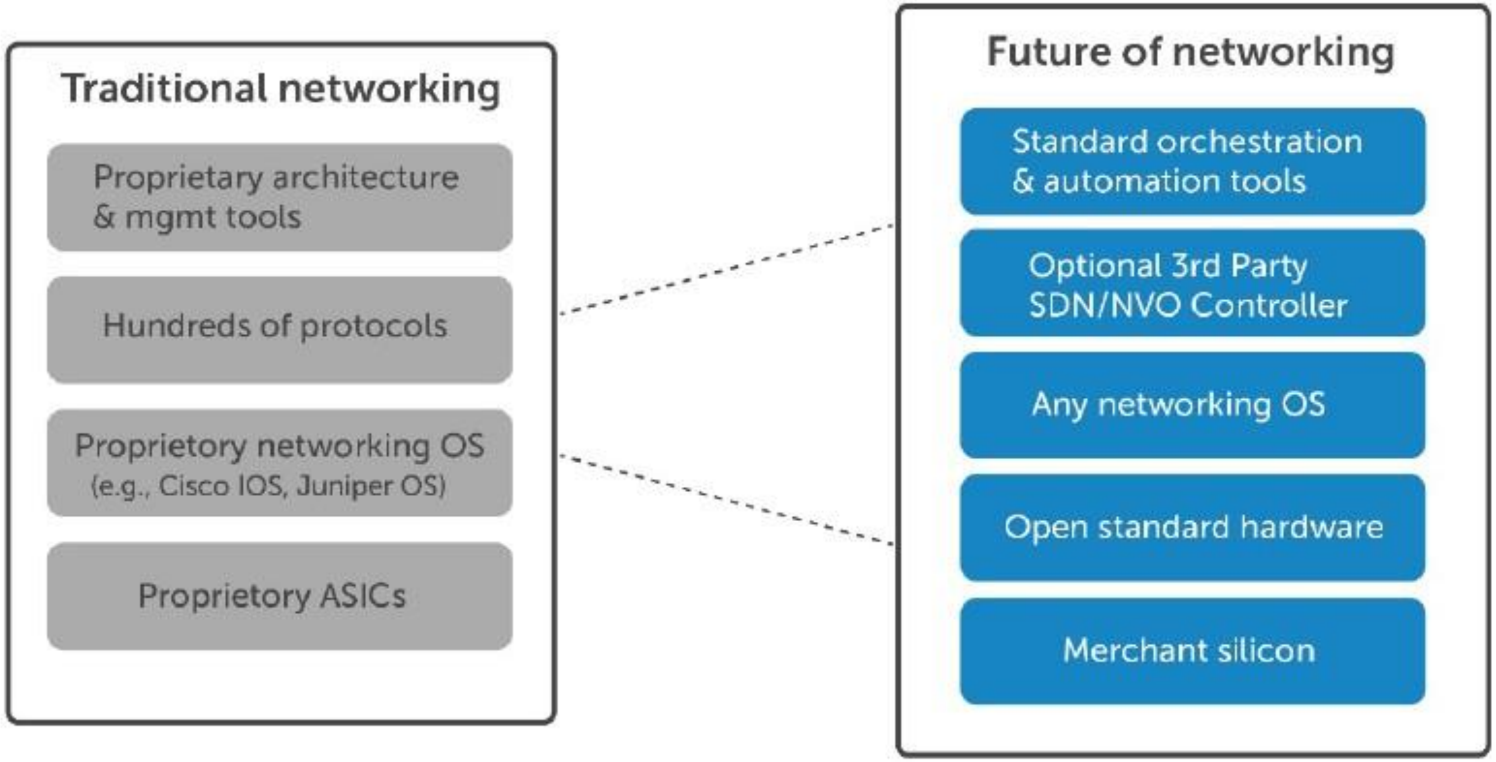
Computing

MEC server platform overvie



- Source: https://portal.etsi.org/Portals/0/TBpages/MEC/Docs/Mobile-edge_Computing_-_Introductory_Technical_White_Paper_V1%2018-09-14.pdf

Summing Up



Source: Software-defined networking (SDN): a Dell point of view - A Dell White Paper (2015)

Enabling Technologies & Open Source Efforts

Enabling Technologies

- Virtualization & Minimalistic OS
  - Docker, ClickOS, Unikernel
- Improving Linux I/O & x86 for packet processing
  - DPDK, Netmap, VALE, Linux NAPI
- Programmable virtual switches / bridges
  - Open vSwitch, P4, Open Networking Linux
- Example start-ups
  - LineRate Systems, 6WIND, Midonet, Vyatta (bought by BCD)

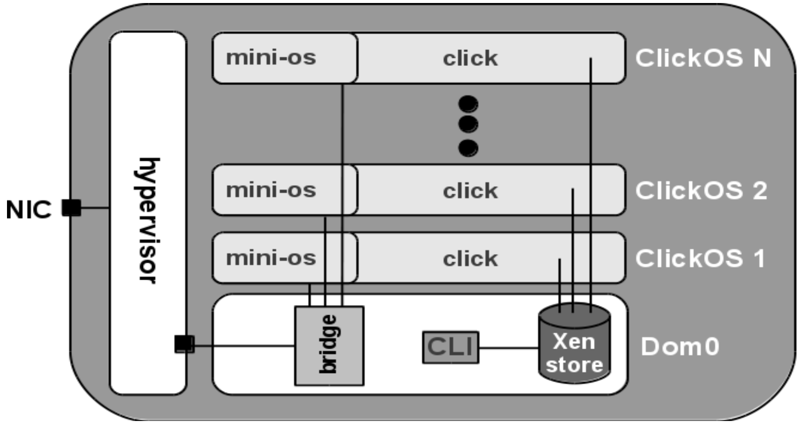


Image source: ClickOS

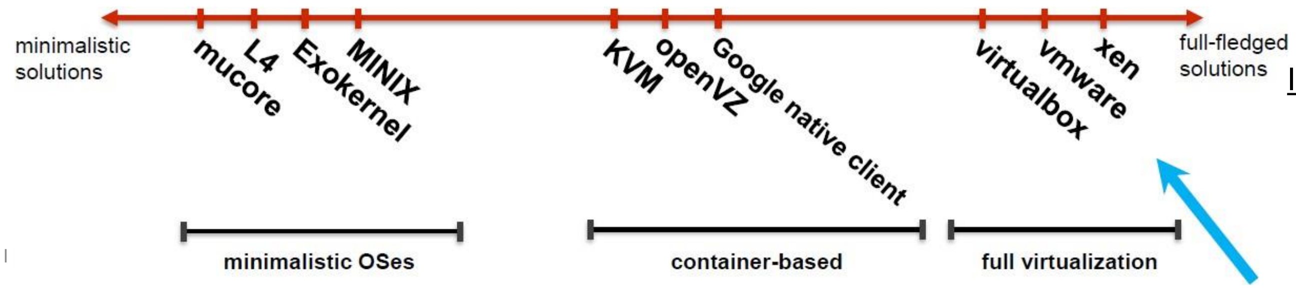


Image source: NEC

CHANGE

Enabling Technologies: Open Source

Why Open Source in Networking?

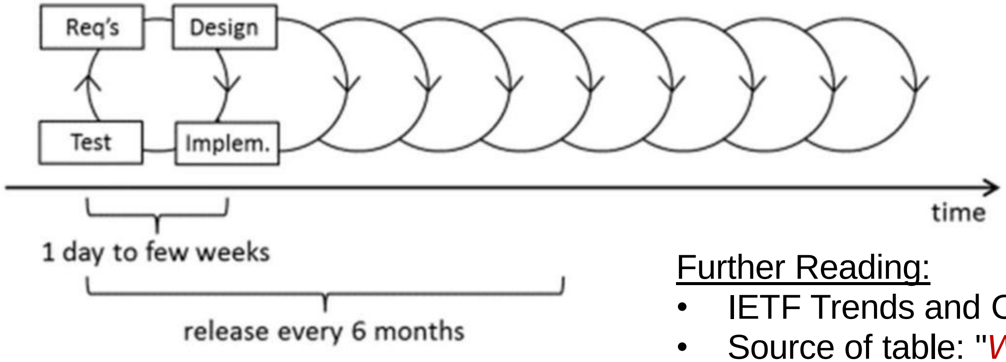
- Higher reliability, more flexibility
- Faster, lower cost, and higher quality development
- Collaborative decisions about new features and roadmaps
- A common environment for users and app developers
- Ability for users to focus resources on differentiating development
- Opportunity to drive open standards

Bottom Line: The open source model significantly accelerates consensus, delivering high performing, peer-reviewed code that forms a basis for an ecosystem of solutions.

Source: Open Source in a Closed Network – Prodip Sen (OPNFV Summit 2015)

SDN/NFV & Open Source: Evolving and accelerating the path of standardization

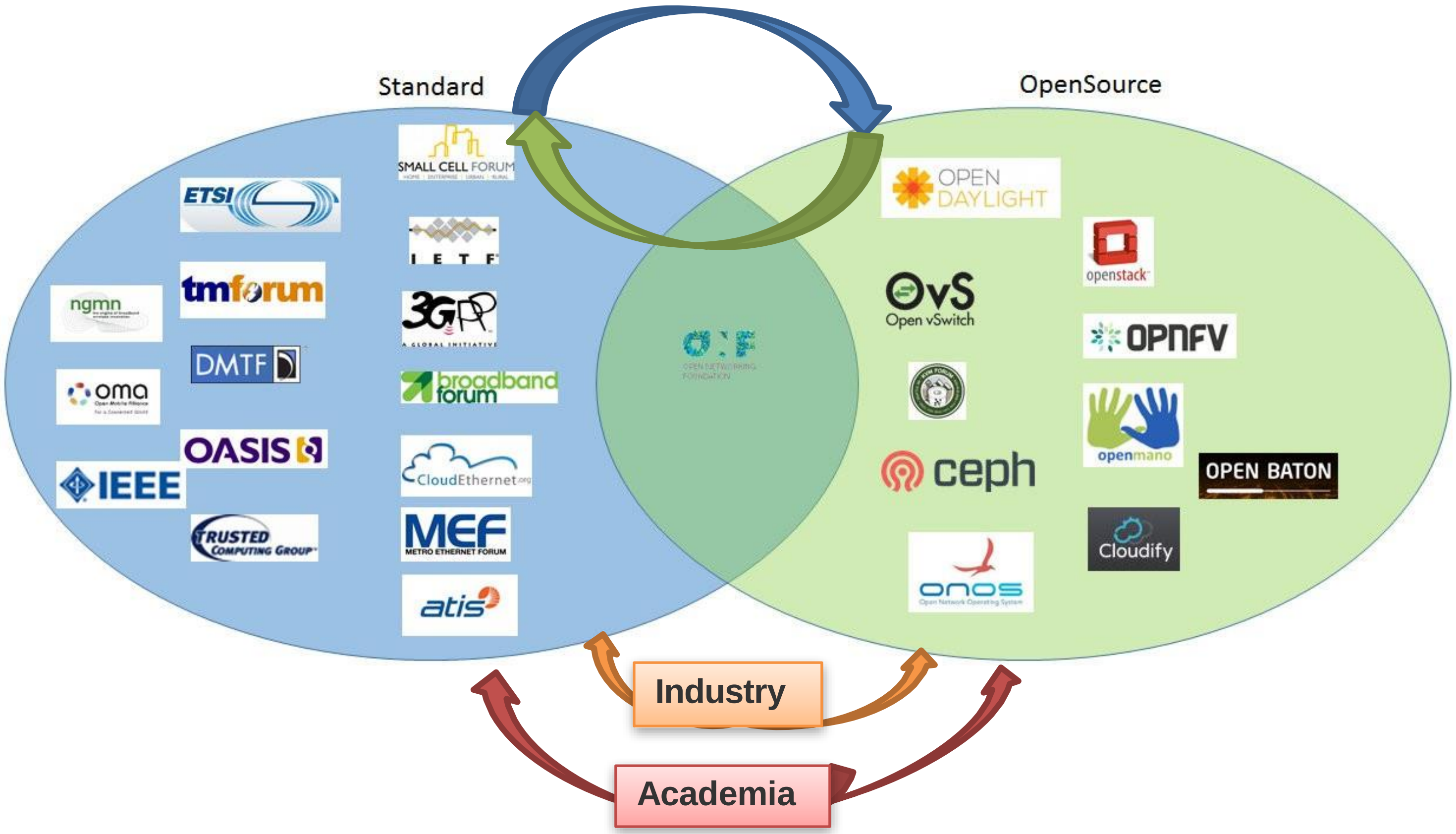
	Present with SDN	Past / Traditional
Drivers	Customer	Vendors
Goals	Address user / operator needs (customization)	Enable multiple solutions (interoperability)
Deliverables	Implementations & PoCs	Documents
Quantity of Standards	Less	More
Timetable	Few years	Many years
Validation	PoCs integral to the process	Products and deployments after release
Point of Control	Contribution to FLOSS codebase Ability to understand codebase	Seat at standards committee table
Parties Involved	Anyone with domain expertise and coding ability	Vendors who can afford membership fees. Experts and academics with high standing in their fields



Further Reading:

- IETF Trends and Observations *draft-arkko-ietf-trends-and-observations-00*
- Source of table: "*When Open Source Meets Network Control Planes.*" In IEEE Computer (Special Issue on Software-Defined Networking), vol.47, no.11, pp.46,54, Nov. 2014.
- Source of figure: A. Manzalini et al., *"Towards 5G Software-Defined Ecosystems"*

Standard / Open Source Organizations



Source: SDN IEEE Outreach, http://sdn.ieee.org/outreach

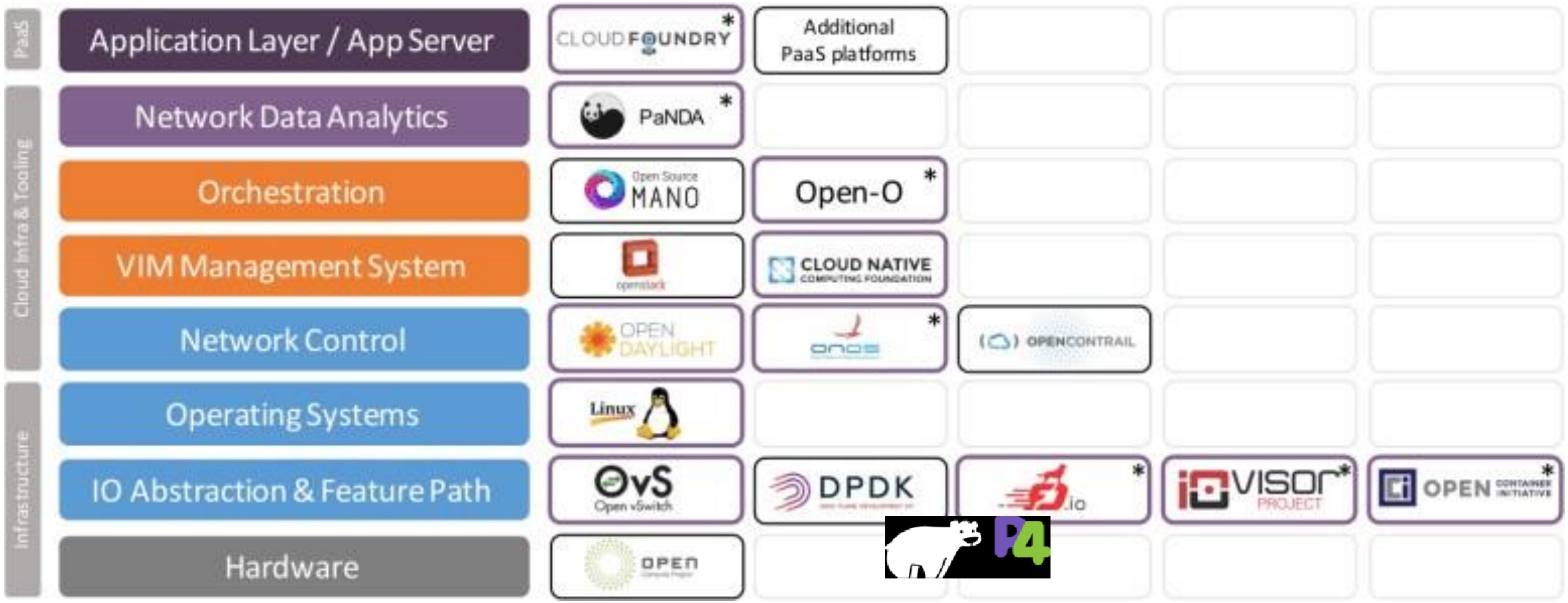
Foundations

Target Collaboration

- Neutral and non-competing
- Legal framework for licensing, copyright, intellectual property management

"Companies feel they can collaborate on an open source project through an independent, not-for-profit entity that they trust - this is incredibly important to them," --Allison Randa (Board President of Open Source Initiative)

Open Source Building Blocks



Source: The Open Source NFV Eco-system and OPNFV's Role Therein – Frank Brockners (OPNFV Summit 2016)

A growing ecosystem...

P4

OvS Open vSwitch

midonet

OVN (Open Virtual Network)

openstack CLOUD SOFTWARE

PROJECT CALICO

onos Open Network Operating System

OPENCONTRAIL

OPEN DAYLIGHT

Open Switch

.io

docker

Open Source Routing A Project of The Network Device Education Foundation

RouteFlow

5G Related Open Source

PROJECT Clearwater

open5GMTC

OpenFog

open5Gcore

OPEN AIR INTERFACE

5G-EmPOWER

Experimentation

Mininet-WiFi
Emulator for Software-Defined Wireless Networks

OMNeT++
Discrete Event Simulator

Mininet

Open
SDN
Core

ns-3

Testbeds

JOIN THE OPEN FEDERATED TESTBED COMMUNITY

Boosting 5G Innovation

5G Berlin

Open SDN Core

5G playground Fraunhofer

open5Gcore

open5Gphotonics

OPEN 5G ACCESS

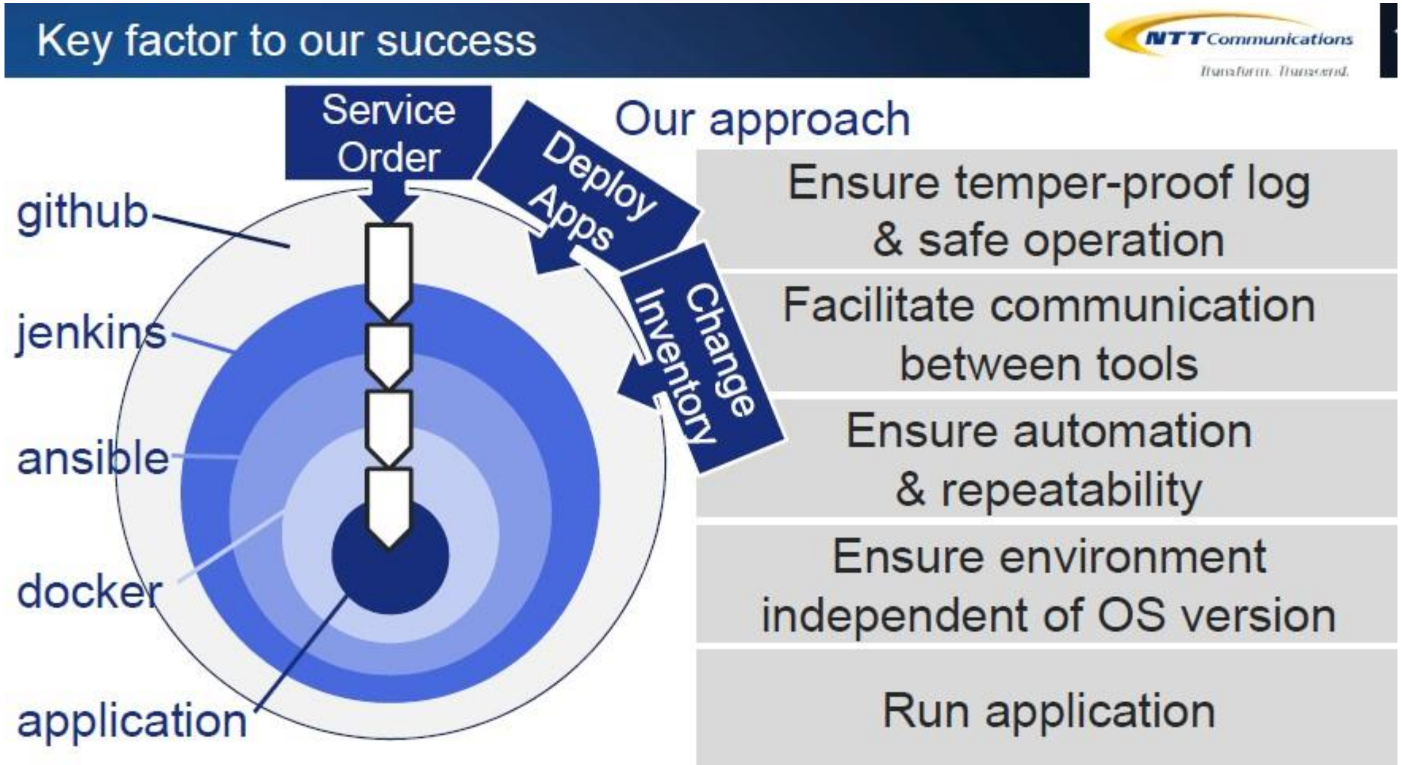
Source: http://www.5g-berlin.org/ http://openfederatedtestbed.org/

Continuous Integration Environment

- Continuous Integration tool chain speeds development and facilitates collaboration
  – Gerrit – code review tool
  – GitHub - code repository
  – Jenkins – automated build tool
  – Maven – code build
  – Ansible/Chef/Puppet – code deployment tool
  – Docker/Vagrant – deployment to Containers/VMs

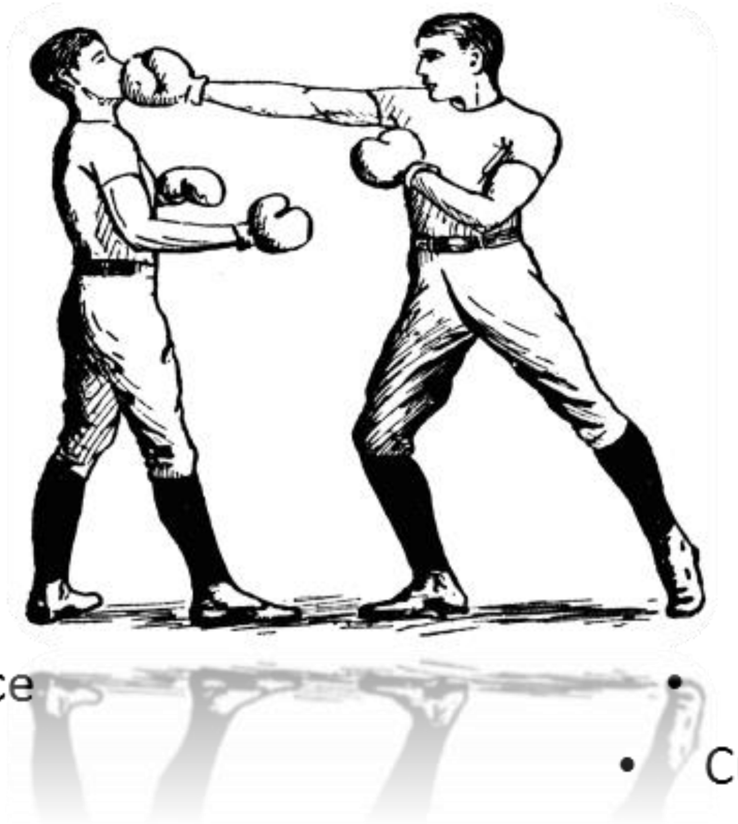
Source: Open Source Carrier Networking – Chris Donley (OPNFV Summit 2016)

Example: NTT (TM FORUM 2016)



Source: A Transformation From Legacy Operation to Agile Operation – Makoto Eguchi (TM FORUM 2016)

The Frontier of Networking



Existing

- CLIs
- Closed Source
- Vendor Lead
- Classic Network Appliances

New

- APIs
- Open Source
- Customer Lead
- Network Function Virtualization (NFV)

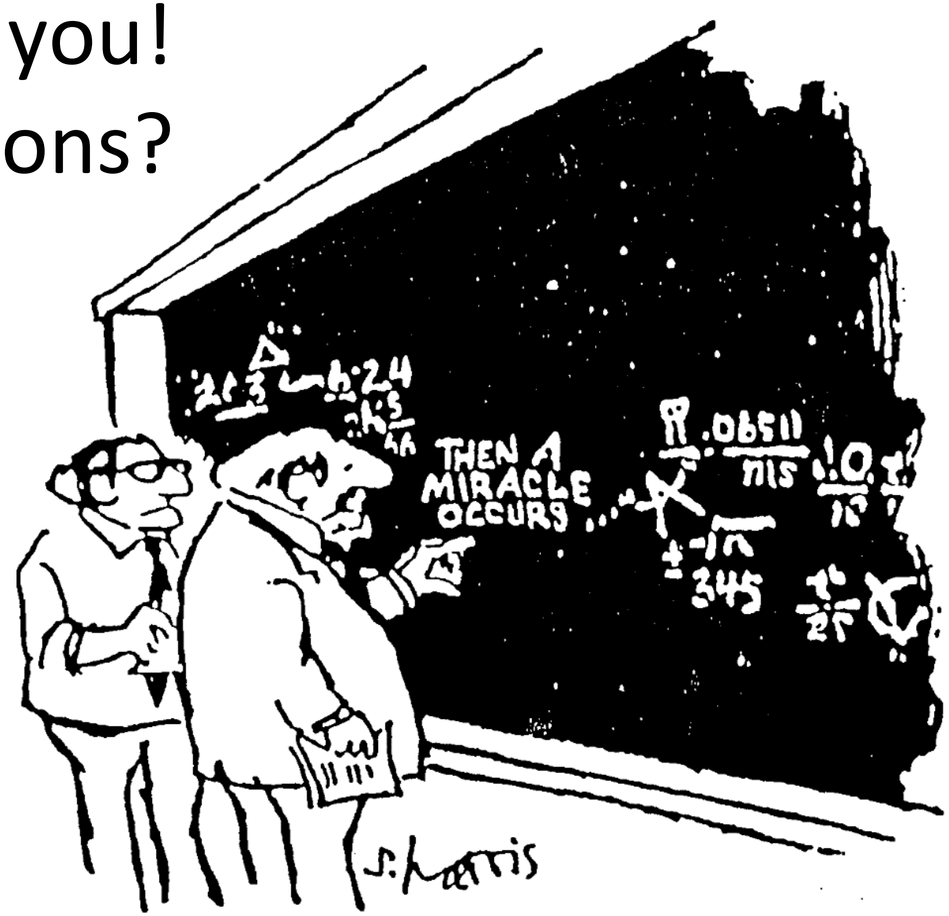
Adapted from: Kyle Mestery, Next Generation Network Developer Skills

Some Takeaways

- Open source speeds up development and promotes interoperability
- Rapid prototypes are a great way to show SDN/NFV value, and now 5G too!
- Developer ramp-up time is challenging,
  - but once integrated, we can make rapid progress
- A common CI environment facilitates sharing between projects
- Cadence, short iteration cycles, fast feedback
- Some people are saying:
  - "open source is the new standardization"
  - "code is the coin of the realm"
- Stay tuned:
  - http://www.sdn-os-toolkits.org/
  - http://sdn.ieee.org/outreach

Open Source
Toolkits & Testbeds

Thank you!
Questions?



5G and the need for programmability

Driving forces behind 5G

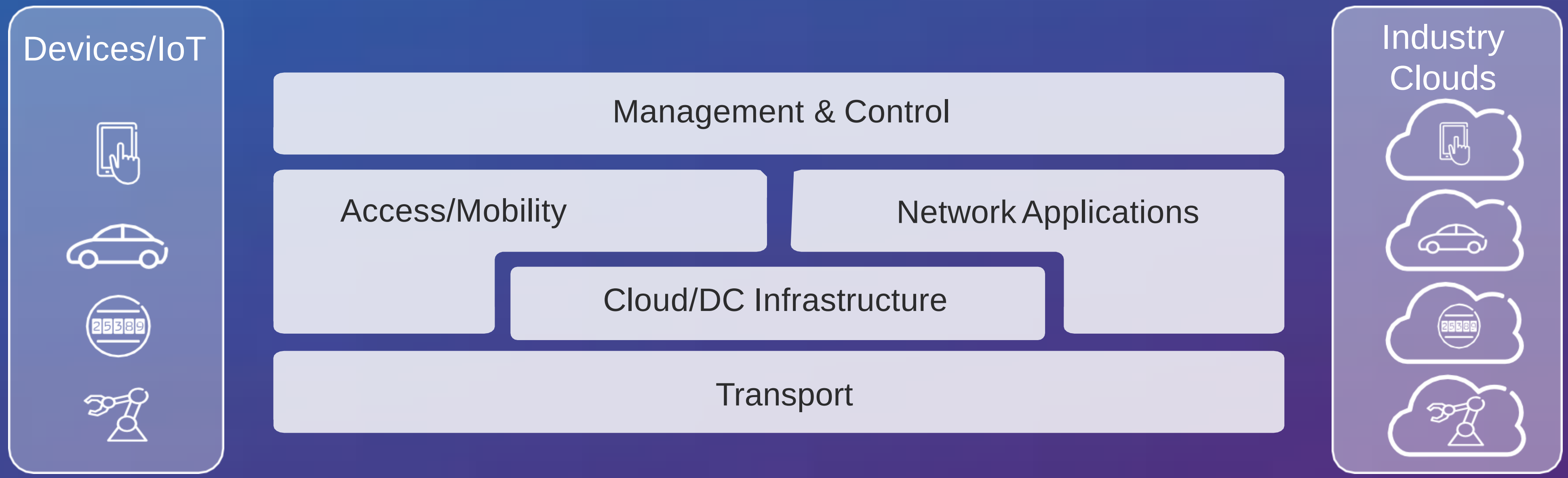
Networked Society

- Improved User Experience
- Massive Traffic Volumes
- Massive No. of Connected Devices
- Massive No. of Services (e.g., IoT)
- Transformed Industries

Technical Drivers

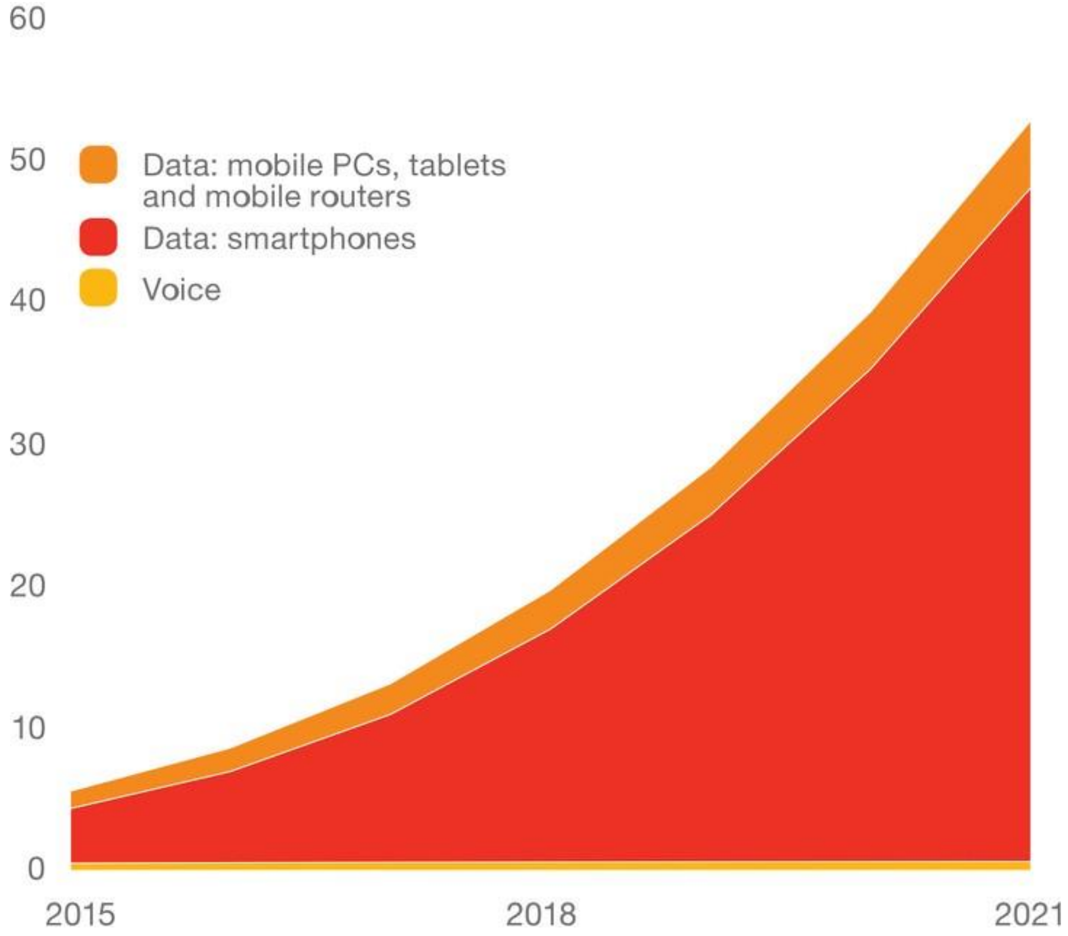
- Network and Service automation
- Resource & Energy Efficiency
- Virtualization & Clouds
- New HW and SW Technologies (SDN & NFV)

architecture



source: Ericsson

Global mobile traffic
(monthly ExaBytes)



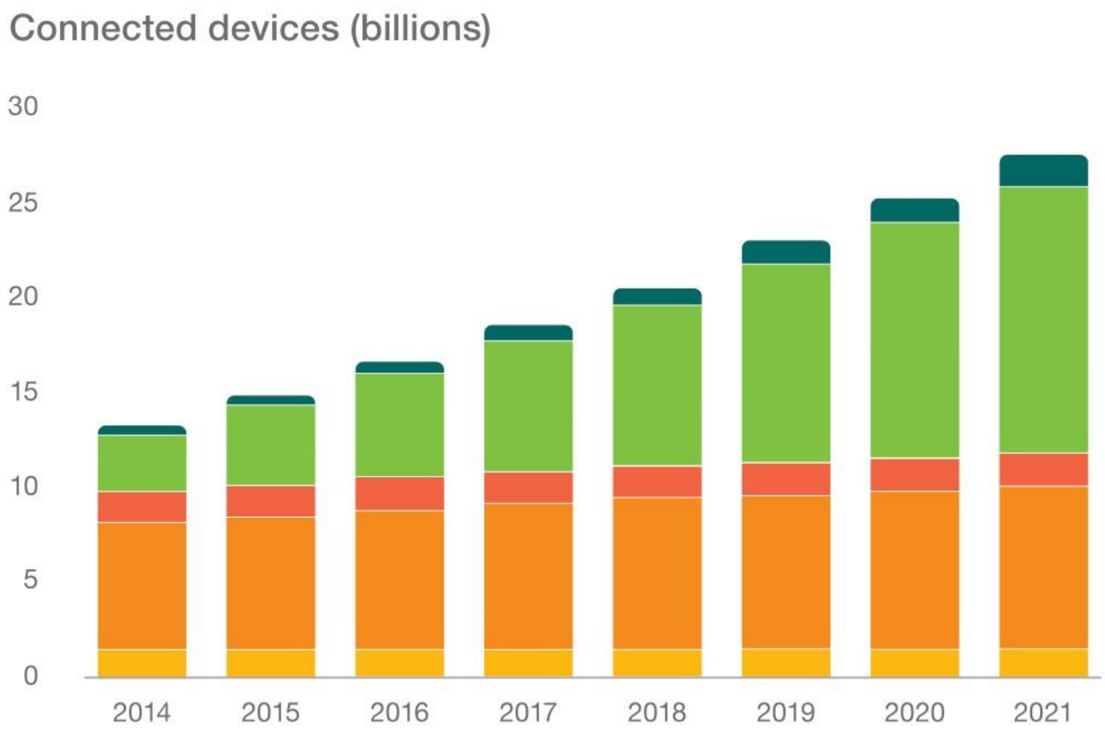
~45% CAGR

12X Growth in smartphone traffic

Around 90% of mobile traffic will be from smartphones by the end of 2021

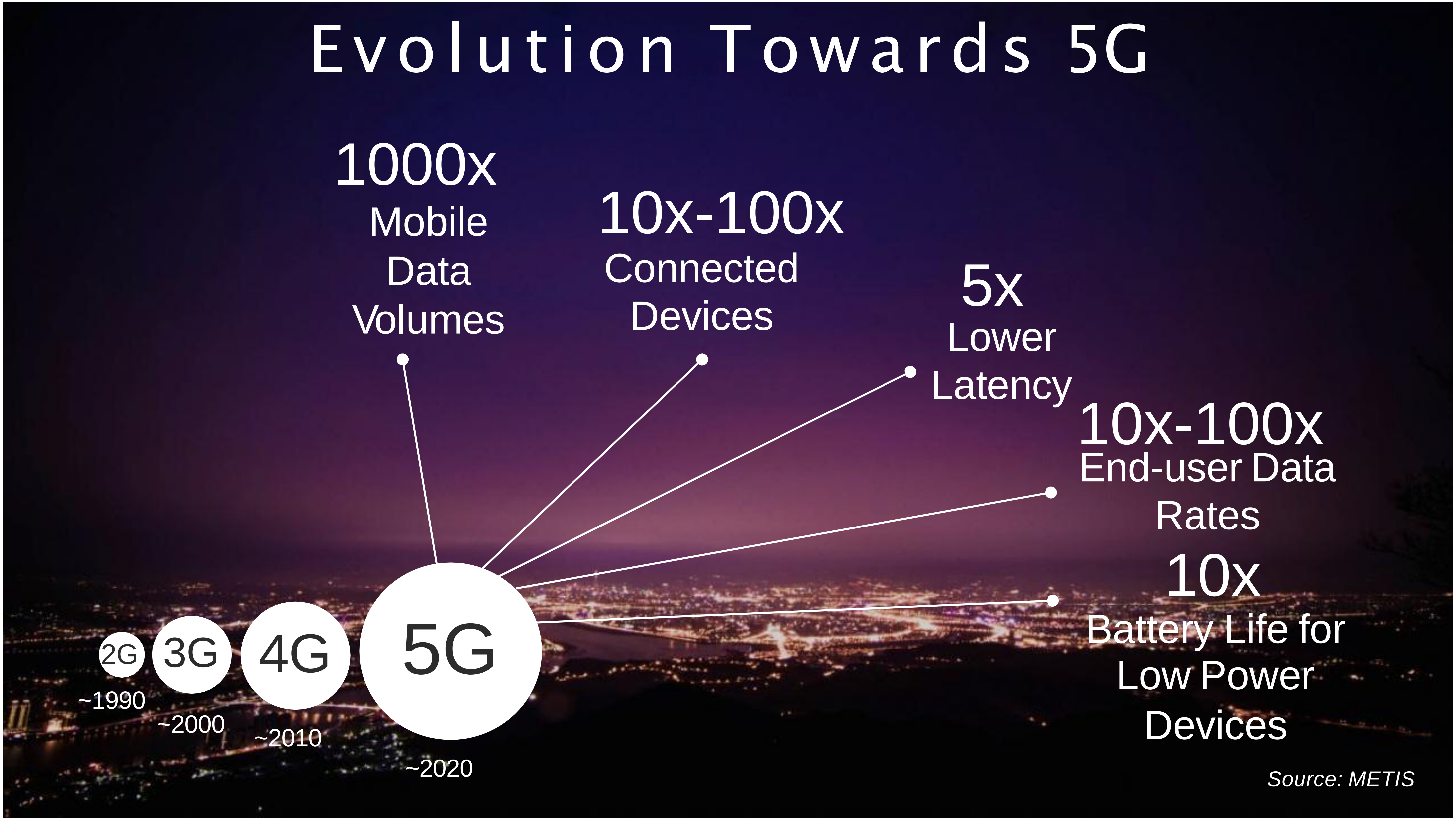
source: Ericsson Mobility Report

IOT TO surpass mobile phones in 2018



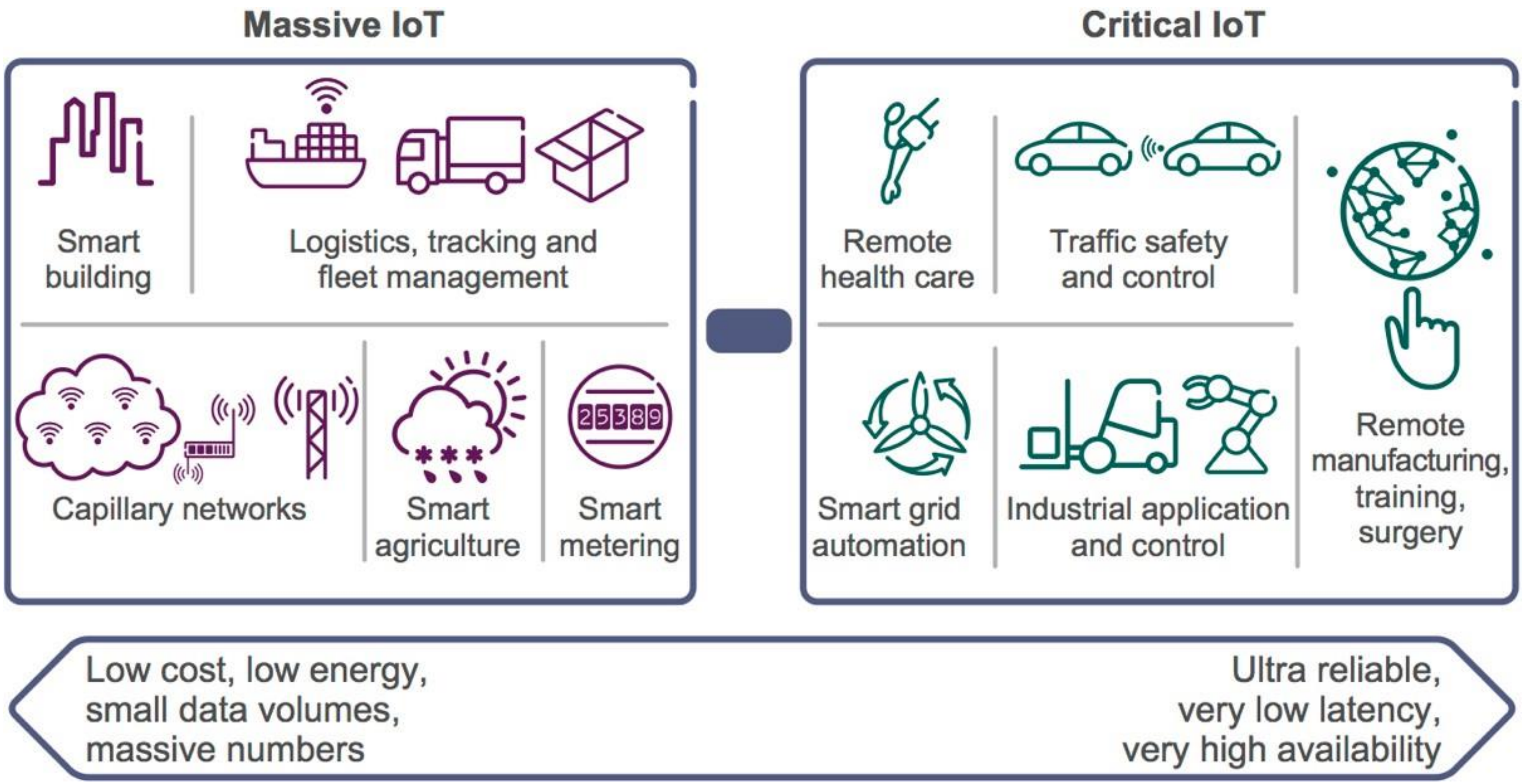
	15 billion (2015)	28 billion (2021)	CAGR 2015–2021
Cellular IoT	0.4	1.5	27%
Non-cellular IoT	4.2	14.2	22%
PC/laptop/tablet	1.7	1.8	1%
Mobile phones	7.1	8.6	3%
Fixed phones	1.3	1.4	0%

source: Ericsson Mobility Report

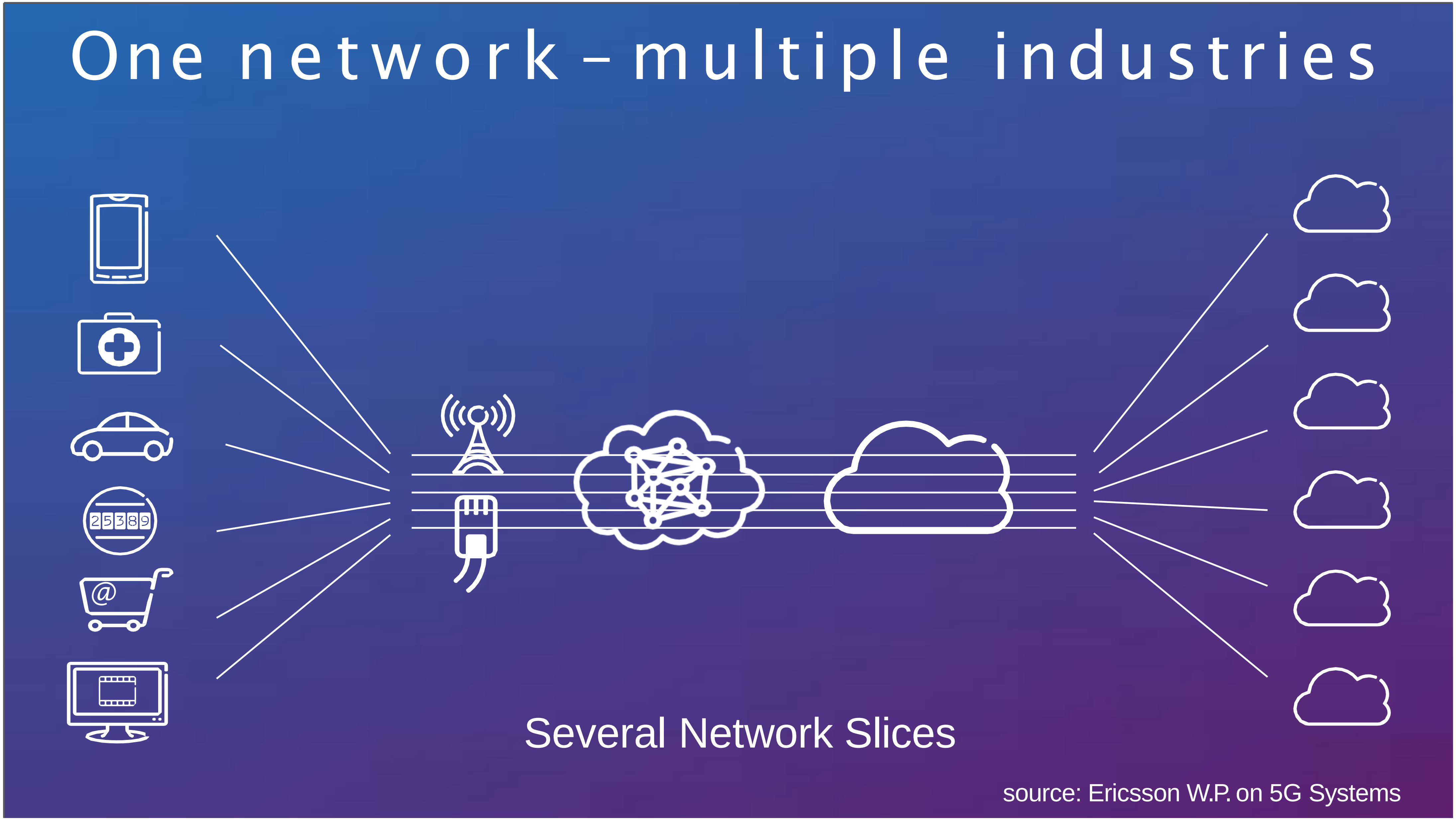

Evolution Towards 5G
1000x
Mobile Data Volumes
10x-100x
Connected Devices
5x
Lower Latency
10x-100x
End-user Data Rates
10x
Battery Life for Low Power Devices
2G
~1990
3G
~2000
4G
~2010
5G
~2020
Source: METIS


5G
USE CASES
BROADBAND EXPERIENCE
EVERYWHERE, ANYTIME
SMART VEHICLES,
TRANSPORT & INFRASTRUCTURE
MEDIA EVERYWHERE
CRITICAL CONTROL
OF REMOTE DEVICES
INTERACTION
HUMAN-IOT
source: Ericsson 5G use cases

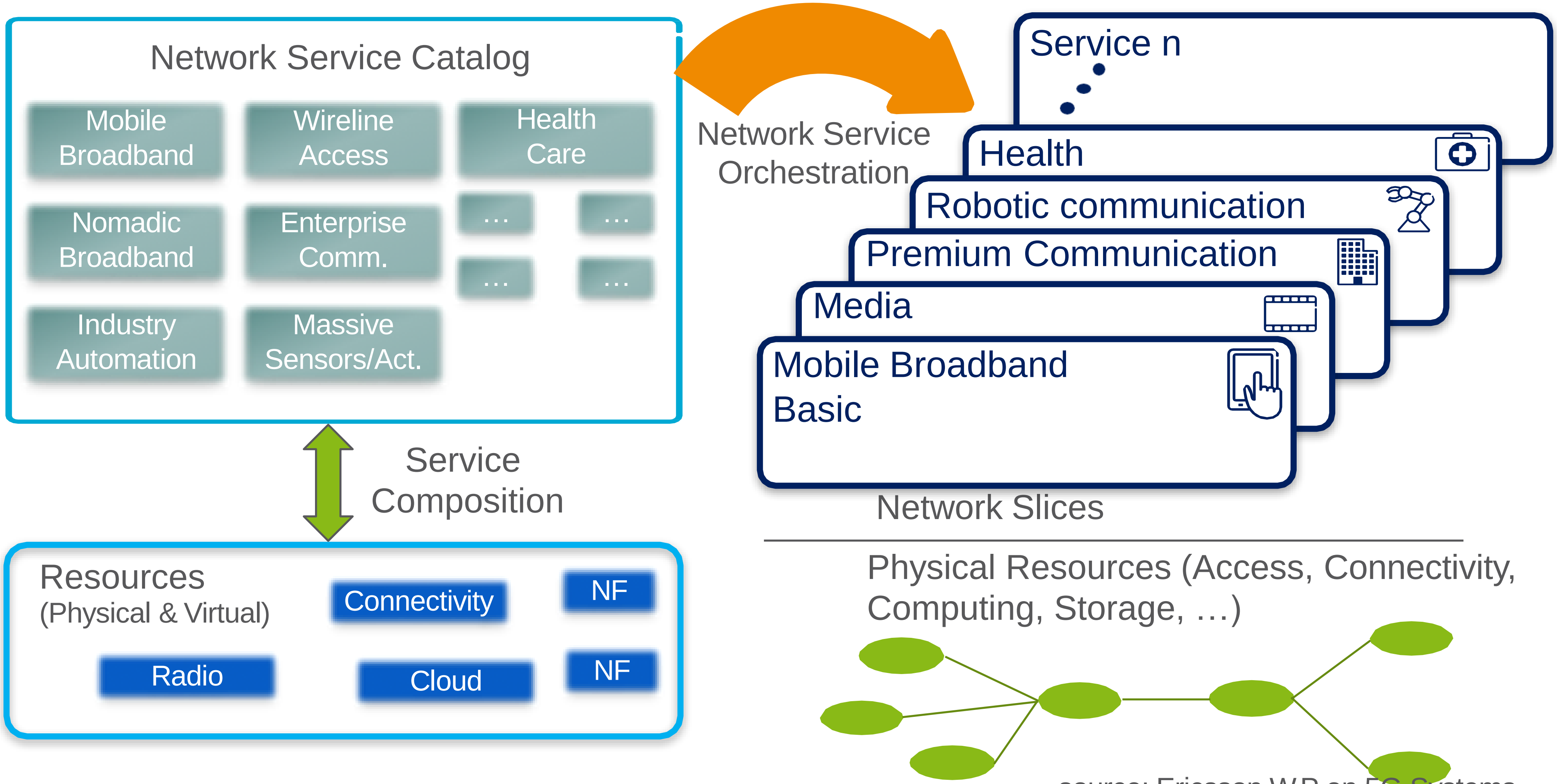
Diverse Requirements of IOT



source: Ericsson W.P. on cellular networks for massive IoT

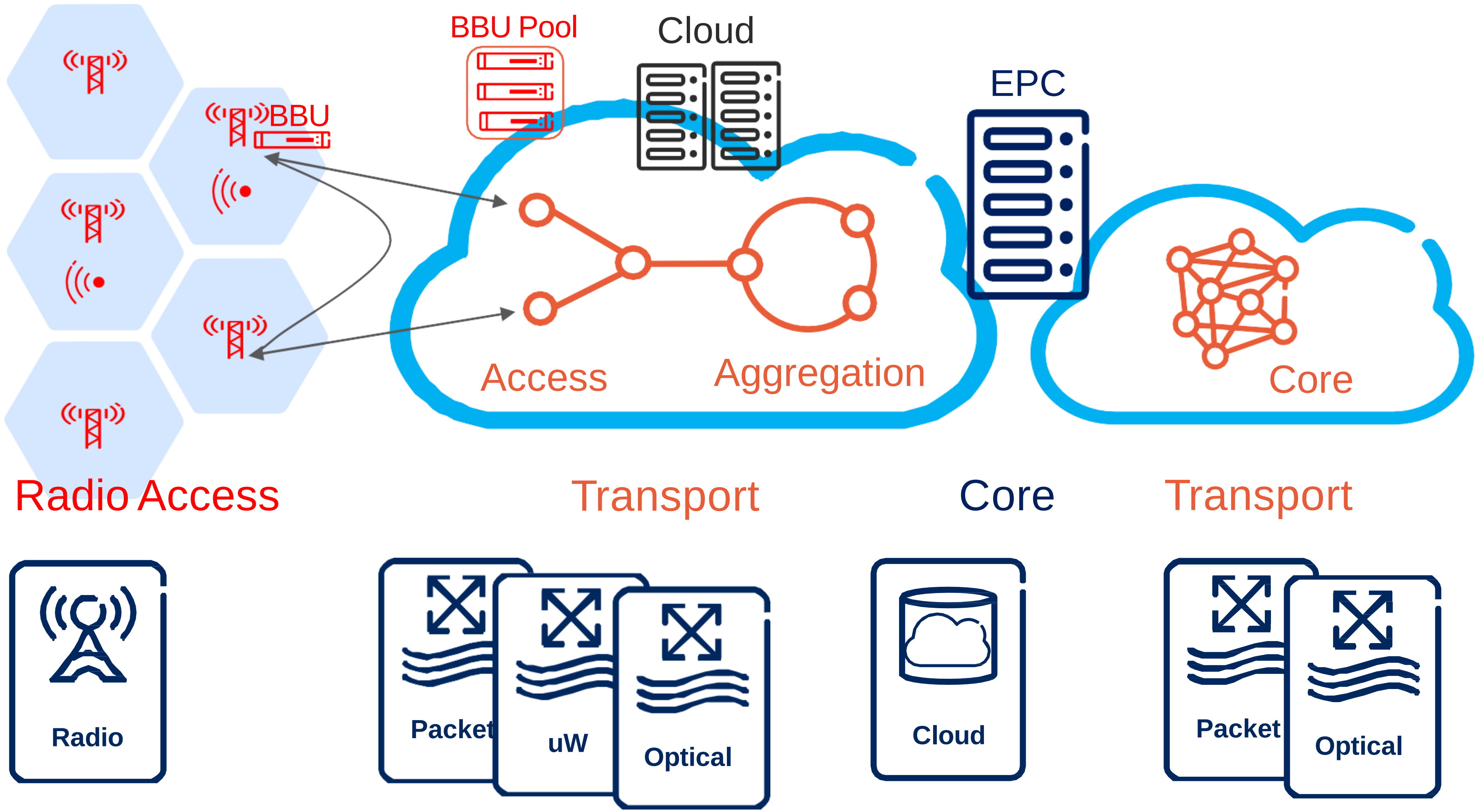

One network – multiple industries
25389
@
Several Network Slices
source: Ericsson W.P. on 5G Systems

Network as a service



source: Ericsson W.P. on 5G Systems

Network Architecture



Programmability in 5G Networks

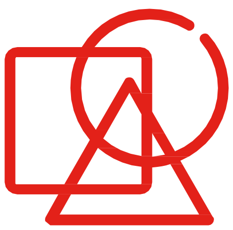
High level of flexibility and programmability in individual domains (mobile core, radio access network and transport network).
Cross-domain programmability and orchestration.

Service Agility

Shorten the time for service creation and service adaptation (e.g., scaling).

Service Diversity

Share a single infrastructure among multiple services with wide range of requirements.

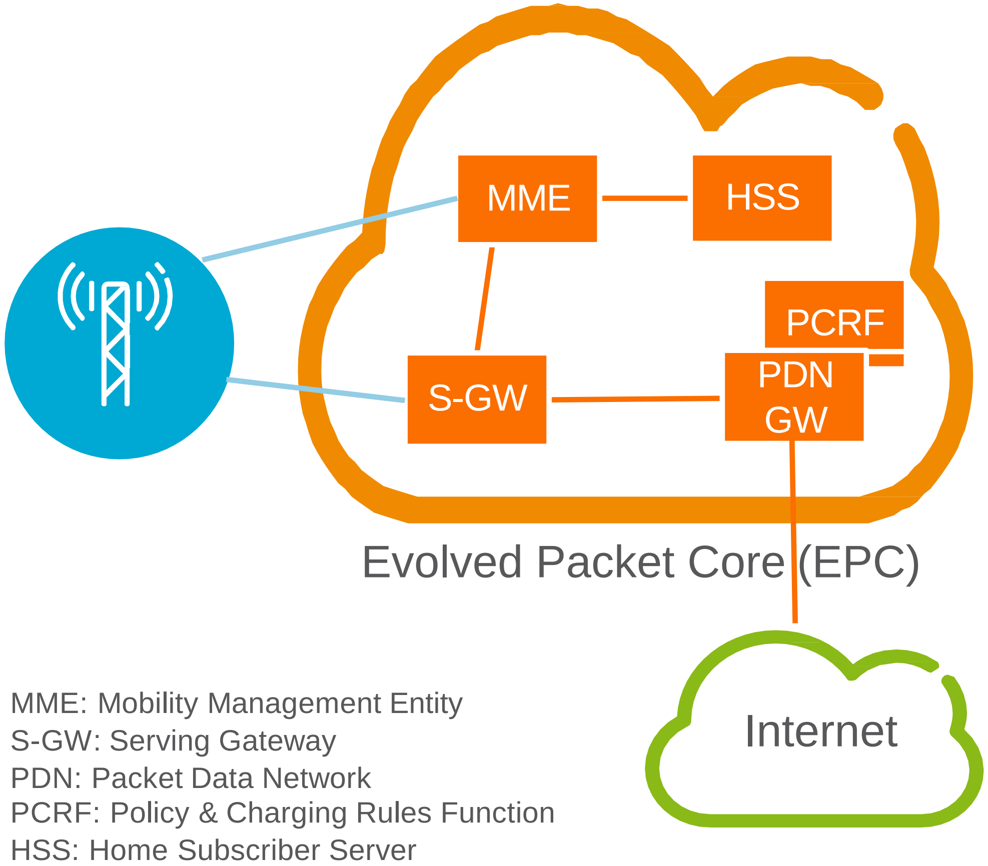


Resource Efficiency

Dynamically allocate the right amount of resources when and where needed.

Programmability in Mobile core

Current Mobile Core Architecture

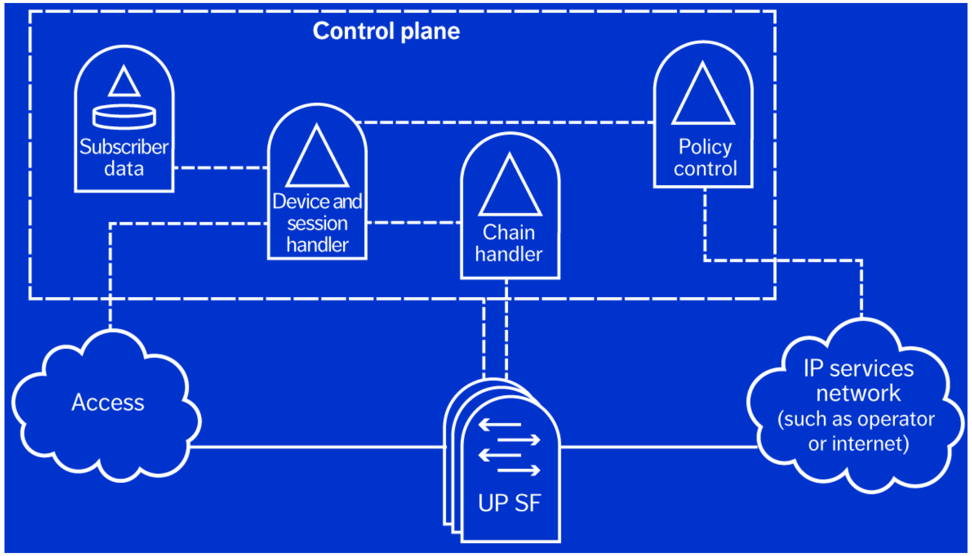


- A single network architecture for multiple services
- Mix of control and user plane functions
- Appliance-based realization

- Difficult to customize
- Scalability

Flexible core architecture

- Separation of control and user-plane functions
- Decompose core functionality into granular functions
- Virtualize functions



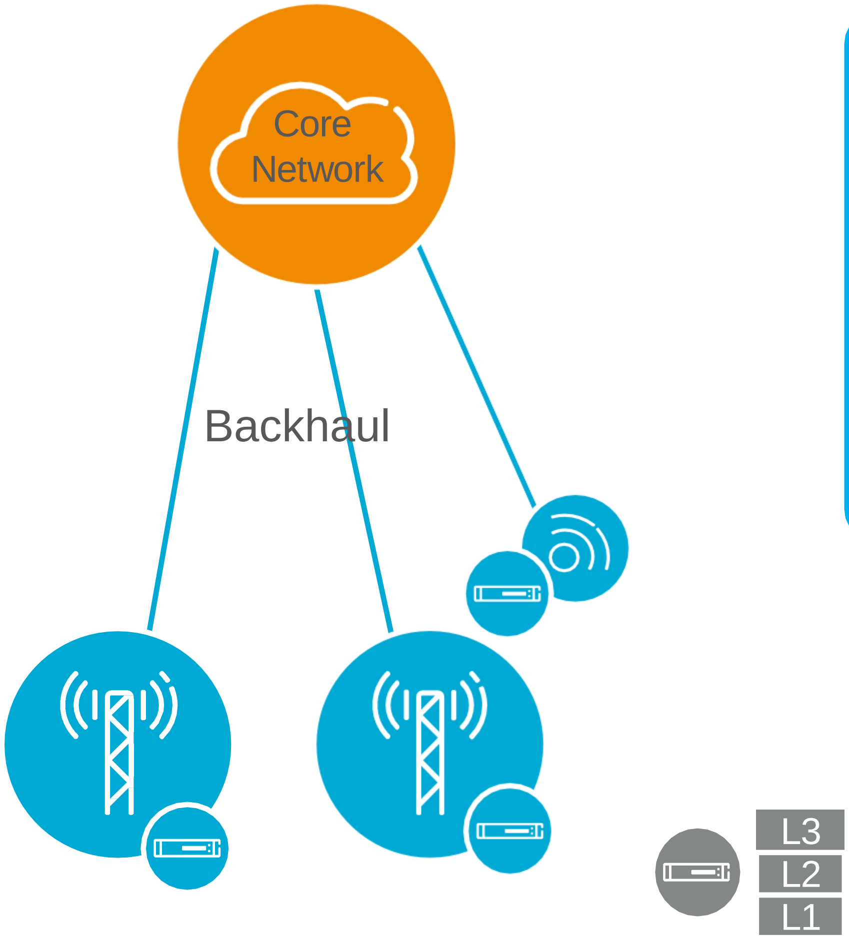
- Customize realization per service/slice
- Centralized control functions
- Selective scaling
- Utilize Cloud Environment
- Flexible placement of functions

Network Function Virtualization (NFV) is an enabler for programmability in mobile core.

Source of fig.: Ericsson Review on 5G Core Flexibility

Programmability in RAN

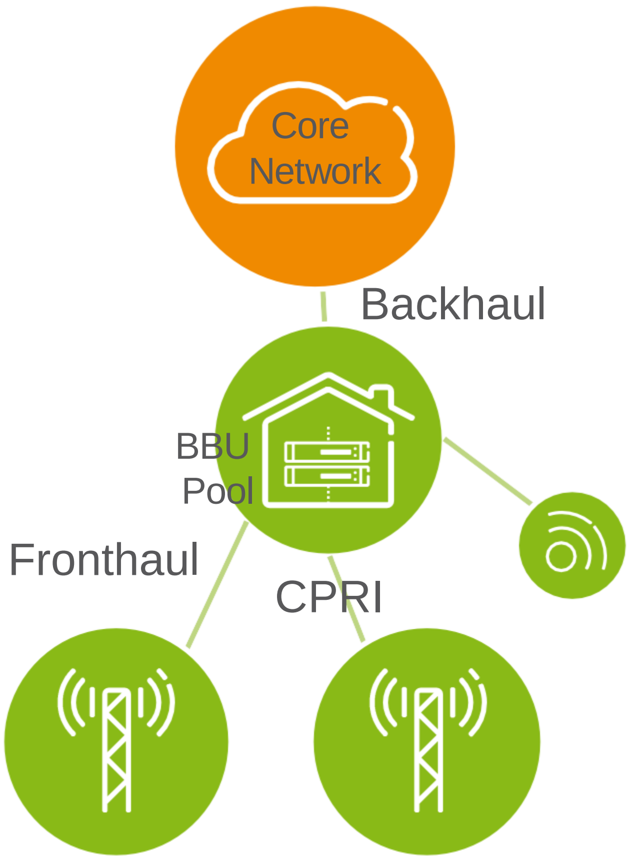
RAN Deployments –I



Distributed Baseband

- Flat Architecture
  - Scaling
- IP connectivity between RAN and Core, and among sites

RAN Deployments –II



Centralized Baseband (C-RAN)

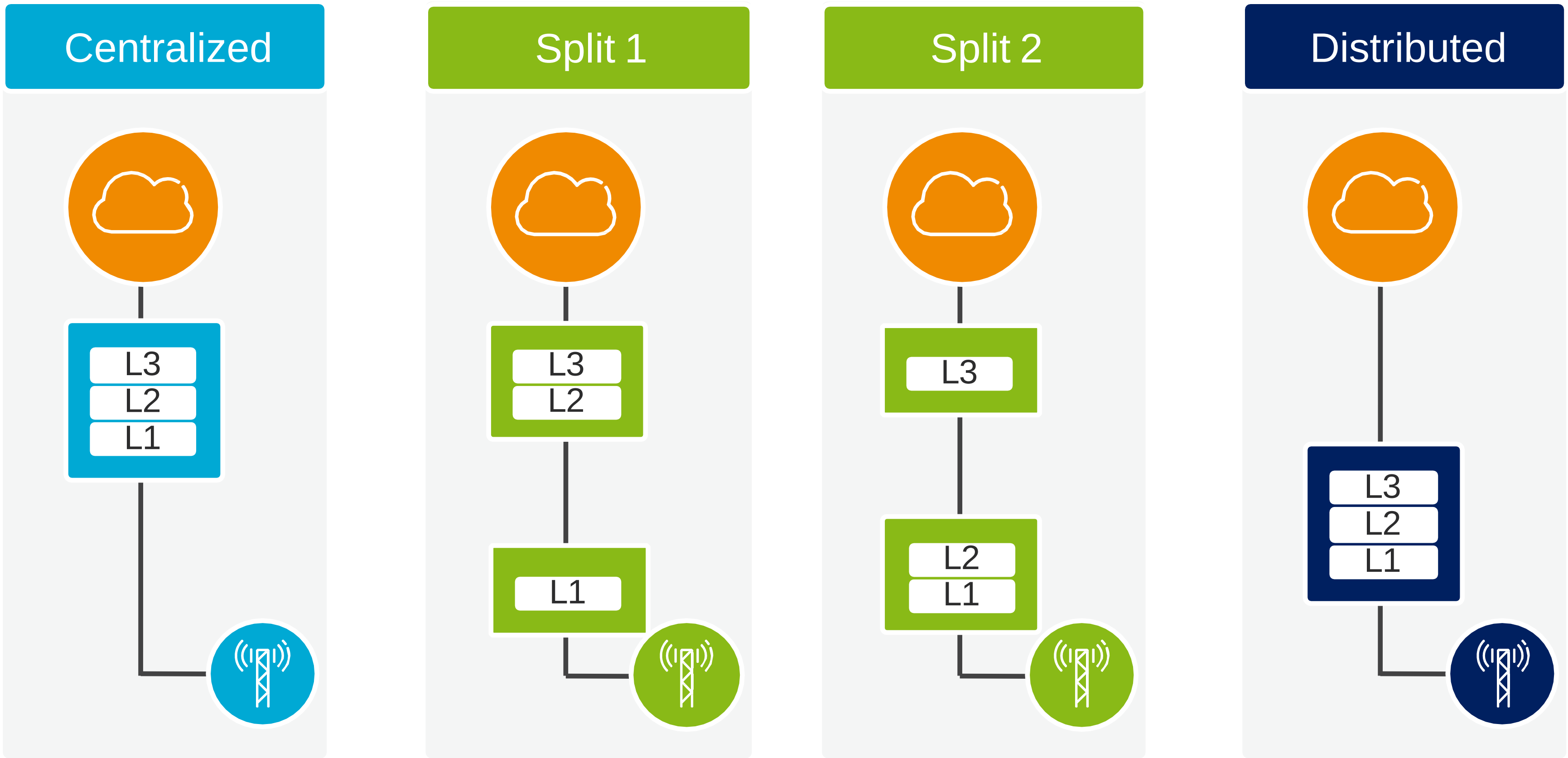
- Pooling gains
- Efficient network management
- Efficient coordination & interference management
- Less network signaling

- Stringent performance requirement on fronthaul (BW, delay and jitter)
- might not be scalable in all 5G scenarios

Need for more flexible split of RAN

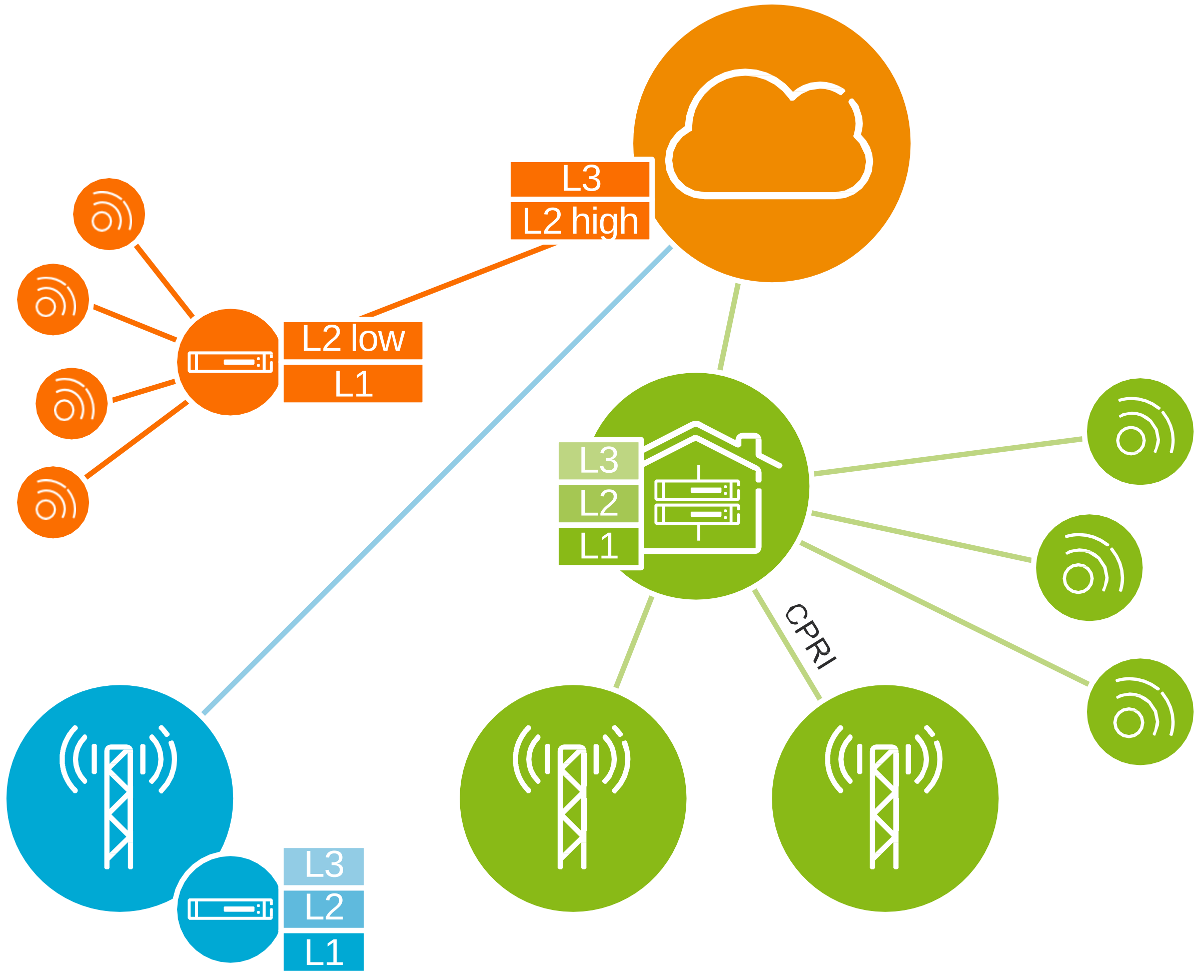
CPRI: Common Public Radio Interface BBU: Baseband processing Unit

CLOUD RAN



Source of fig.: Ericsson W.P. on Cloud RAN

Flexibility with cloud ran



Centralization gains as in C-RAN

- Pooling
- Network Management
- Coordination

Less Transport Requirements

Virtualization gains

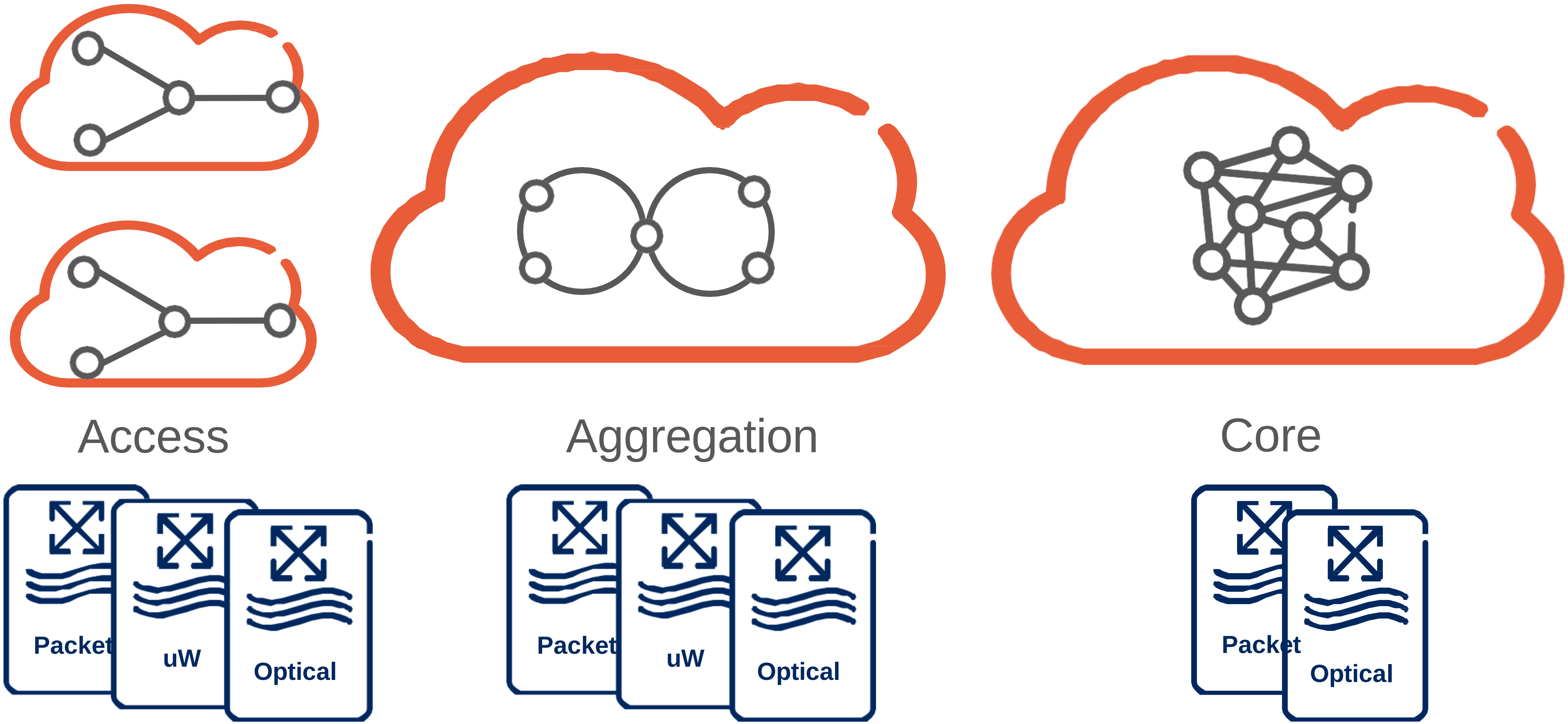
- Selective scaling (E.g. User plane vs Control Plane)
- Cloud-based Realization

Collocation of RAN & Core

Source of fig.: Ericsson W.P. on Cloud RAN

Programmability in Transport

Current transport networks



Current transport network ISsues

- Monolithic realization of control and forwarding functions
- Proprietary Management Interfaces
- Complex control and management
- Several technology domains with independent control

- Lengthy and manual service creation/scaling
- Inefficient Resource Utilization
- Inefficient static sharing
- Difficult cross-layer optimization
- Application Unaware

Programmable Transport

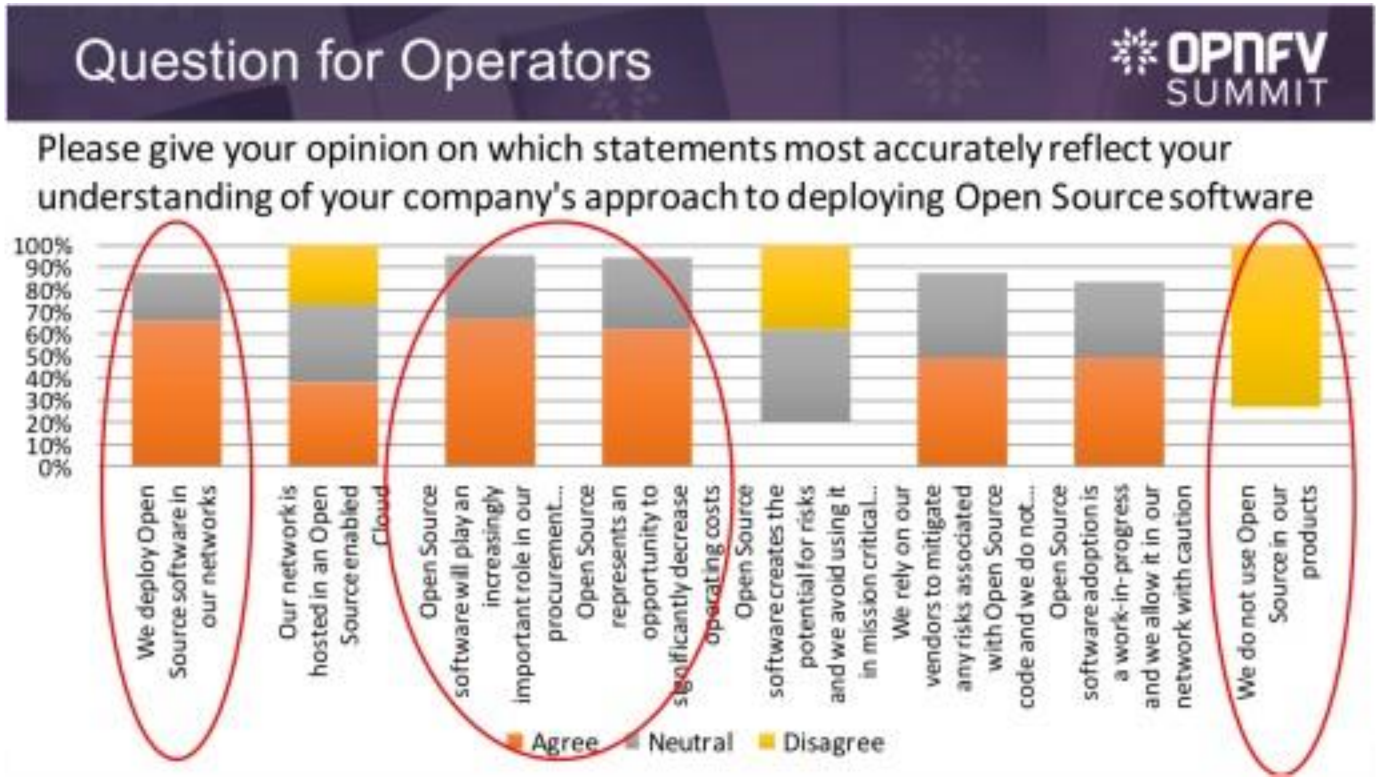
- Separation of control and forwarding functions
- Define interfaces between control and forwarding
- Open up the control plane for programming
- Develop Efficient sharing mechanisms

- Automation of network and services
- Dynamic creation/update of (virtual) connections/tunnels
- Resource-optimized operation
- Cross-layer optimization (e.g. packet-optical convergence)
- Radio-aware adaptations

Software-Defined Networking (SDN) is an enabler for programmability in transport networks.

BACKUP

Perspective



Source: Survey Results: Bridging the Gap Between Open Standards and Open Source - Elizabeth Rose (OPNFV Summit 2016)

Challenges: Closed vs. Open

Carrier Network Software	Open Source Softwa
Proprietary with custom features for specific users	***Open***, generic enough to be used in wide range of applications
Features : Roadmaps agreed between vendors & users before releases	***Features*** emerge based on community engagement. Release feature list is after the fact.
Upgrades: Releases are well planned but slow to implement	***Upgrades***: Frequent upgrades are expected requiring a Continuous Integration environment
Solutions: Typically vertically integrated	***Solutions***: Flexible with many possibilities for horizontal and vertical integration
Guarantees: System / solution vendor responsible.	***Guarantees***: User or separate system integrator responsible.
Carrier Mindset: No room for failure	***Open Source***: Fail fast, and fix faster
Carrier Mindset: One throat to choke	***Open Source***: No throat to choke

Source: Open Source in a Closed Network – Prodip Sen (OPNFV Summit 2015)

Open Source Road

Source: The NFV Revolution Must Be Open – Dave Neary (OPNFV Summit 2016)

Approach for Contributions

- Open Source Projects need more than code!
  - Documentation
  - QA
  - Infrastructure
  - Blogs
  - IRC
- Evaluate how your project idea can fit with the existing project
  - Does it overlap?
  - Does it provide extra value?
  - Can something be abstracted?
  - Is the community interested?
  - Does it fit the communities goals?
  - Do you have a plan for testing, documentation and support?

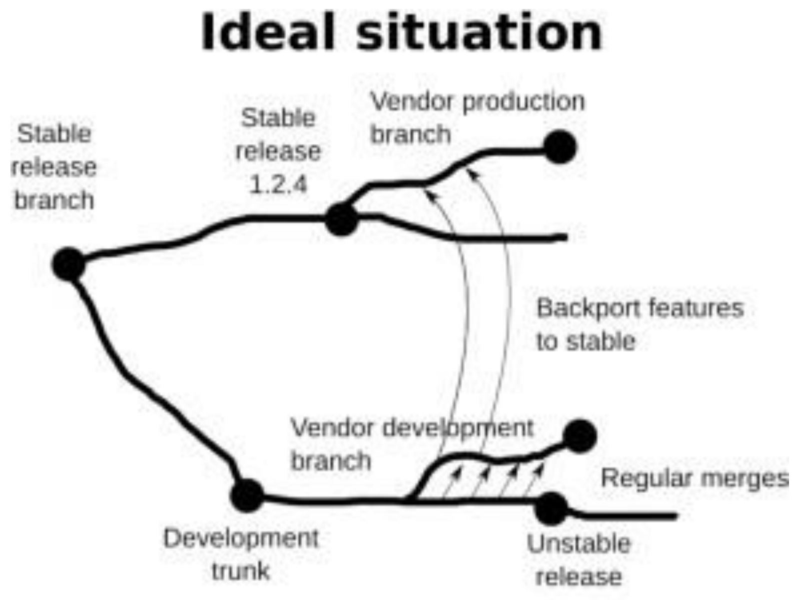
Source: Upstream Open Source Networking Development: The Good, The Bad, and the Ugly – Kyle Mestery, Justin Pettit, Russell Bryant (ONS 2016)

Project Evolution: Examples

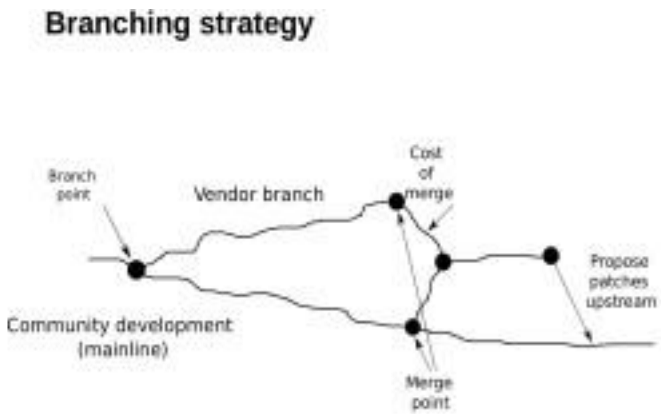
Project	Started	Evolved To	Foundation
OvS Open vSwitch	Small core group of contributors	Taking on a new project (OVN), growing slowly but surely	No foundation*
openstack CLOUD SOFTWARE	Rough consensus of many disparate groups into one codebase	Focused group of repositories	OpenStack Foundation
OPEN DAYLIGHT	Many separate groups under the same large umbrella	More and more projects and repositories	Linux Foundation

Source: Upstream Open Source Networking Development: The Good, The Bad, and the Ugly – Kyle Mestery, Justin Pettit, Russell Bryant (ONS 2016)

Building on Open Source Projects



This is what **upstream** does



Cost: Vendor work + cost of merge + "community overhead"

"Do NOT fall into the trap of adding more and more stuff to an out-of-tree project. It just makes it harder and harder to get it merged. There are many examples of this."
-Andrew Morton

Source: Swimming Upstream – Dave Neary (OPNFV Summit 2016)

Open Ended Questions

- What are the inhibitors to adoption of Open Source Implementations?
  - Issues surrounding licensing (esp. GPL)
  - Industry understanding of Open Source licensing
  - Competitive issues and fragmentation
  - Security
  - Quality and robustness
  - Maintenance and support

Source: Survey Results: Bridging the Gap Between Open Standards and Open Source - Elizabeth Rose (OPNFV Summit 2016)