



Computer Network

# Lecture 7

## Network Layer Part.2

2019. 03. 01

Sungwon Lee  
Department of Software Convergence

# Contents

- **Softwarization**
- Software Defined Network(ing)
- Network Function Virtualization
- SDN/NFV
- SDN/NFV Softwares
- SDN/NFV Use Cases
- SDN/NFV into 5G
- P4 (programming language)

# Software is eating up the world\*

August 20, 2011



\* Marc Andreessen  
in Wall Street Journal



**Mark Andreessen**  
founder of Netscape,  
renowned Venture Capitalist  
Andreessen-Horowitz

**Software is eating the world, in all sectors**

In the future every company will become a **software** company

## Linux Eating the Networking Industry

### Open Source Networking Landscape Linux Foundation hosts 9/10 Top projects

Services

Application Layer / App Server

Network Data Analytics

Orchestration, Management, Policy

Cloud & Virtual Management

Network Control

Operating Systems

IO Abstraction & Data Path

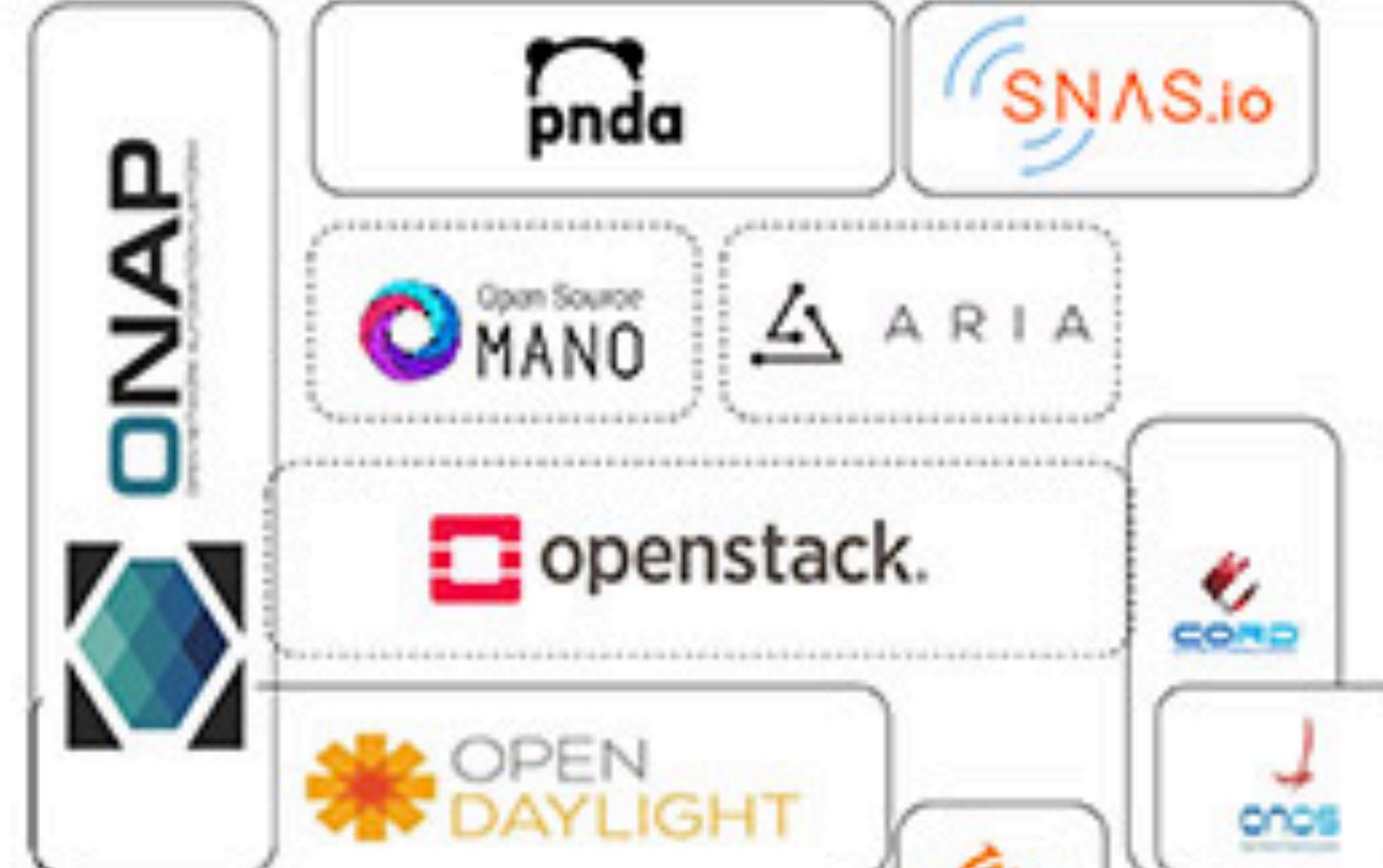
Disaggregated Hardware

THE LINUX FOUNDATION

OPNFV System Integration & Test Automation

CI/CD

Product, Services & Workloads



Linux Foundation Hosted

Outside Linux Foundation

CLOUD FOUNDRY

Standards

tmforum

ITU

MEF NIST

OIF

IETF

ETSI

IEEE 802 CableLabs

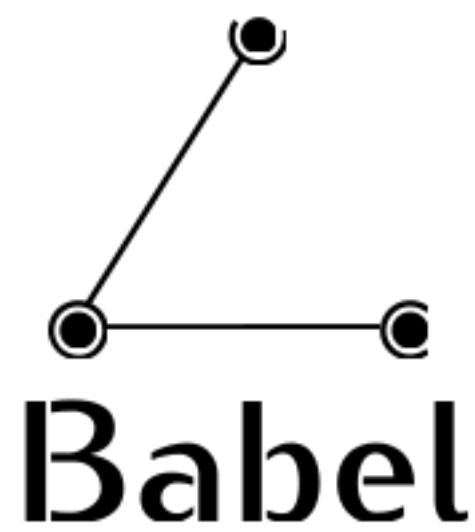
3GPP OIF

Automation of Network + Infrastructure + Cloud + Apps + IOT

## Linux Eating the Networking Industry



# Open Source Networking Software for Legacy Routing Protocols



Babel

BIRD

OpenBGPD

OpenOSPF

Quagga

Free Range Routing

XORP

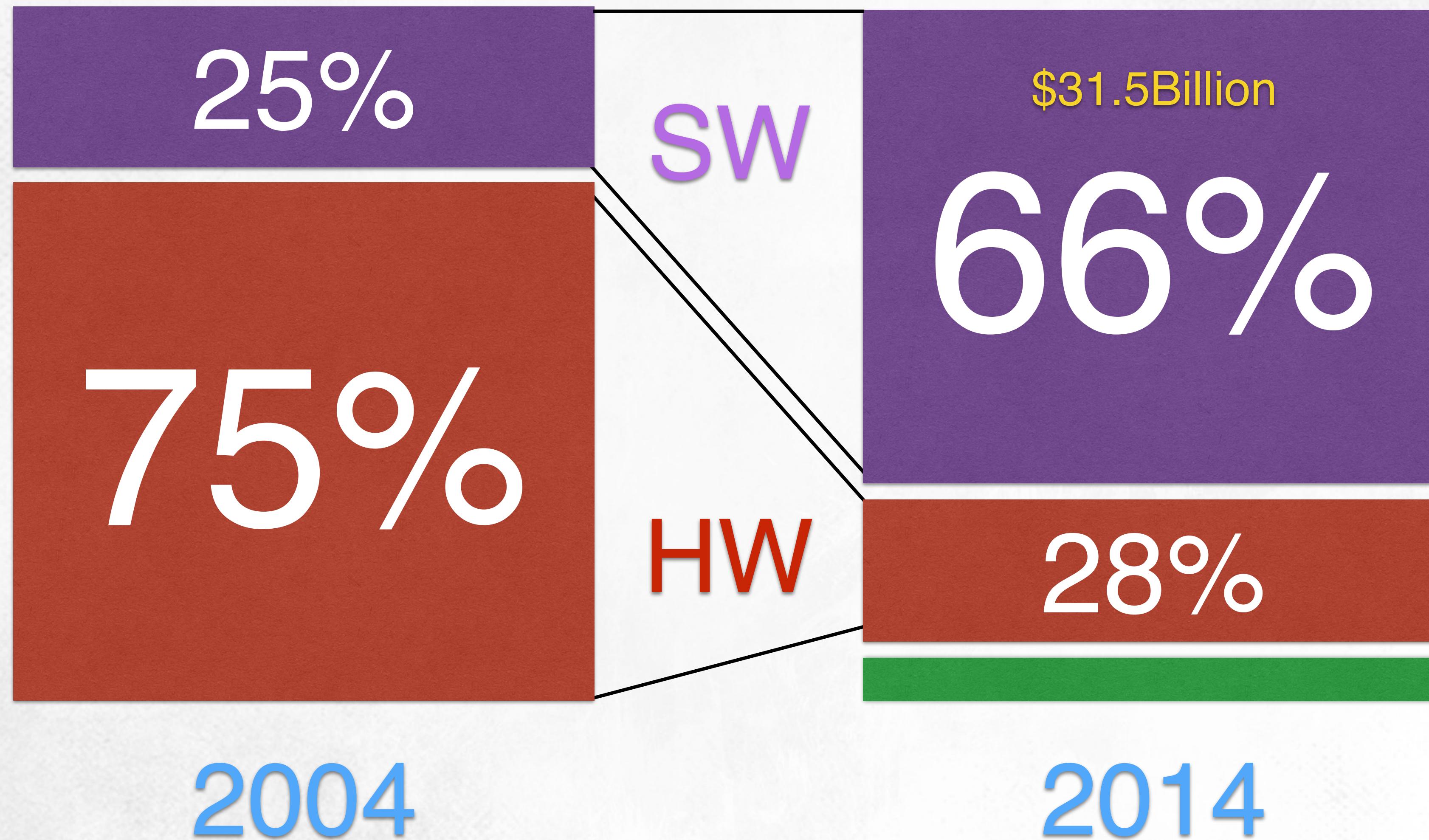
Zebra

Optimized Link State Routing Protocol



# Ericsson's Annual Sales

“‘Hardware’ to ‘Software and Service’ within a decade”



# IP & Cloud is Common

“Most 3.5G Systems are Deployed over IP Network in 2000s”

In 2010, Ericsson managed several dozen different hardware and software architectures, not an unusual model for the telecommunications industry. But the push to cloud services means it's time to simplify.

“We decided by 2015, we want to have three hardware platforms, two to three software stacks,” Vestberg told Fortune, while visiting New York for investor meetings.

“When we did that, we also decided that we wanted to virtualize all our products for the future. You can buy them per capacity, buy them per how much they are used. We couldn't do that in our industry before, it was impossible.”

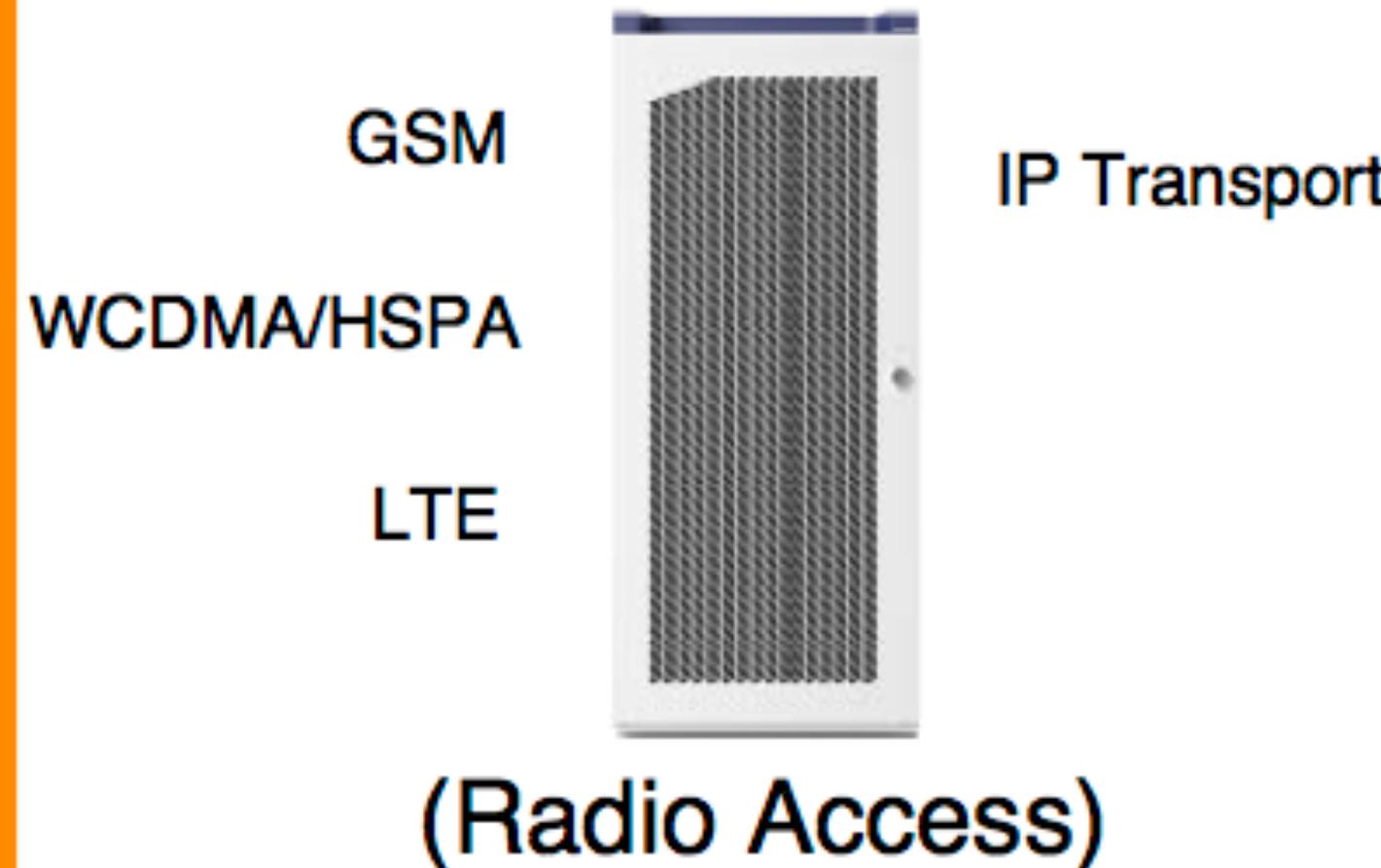
# Purpose-Oriented Common Platform

“Three is Enough: Radio, Bearer, and Control”

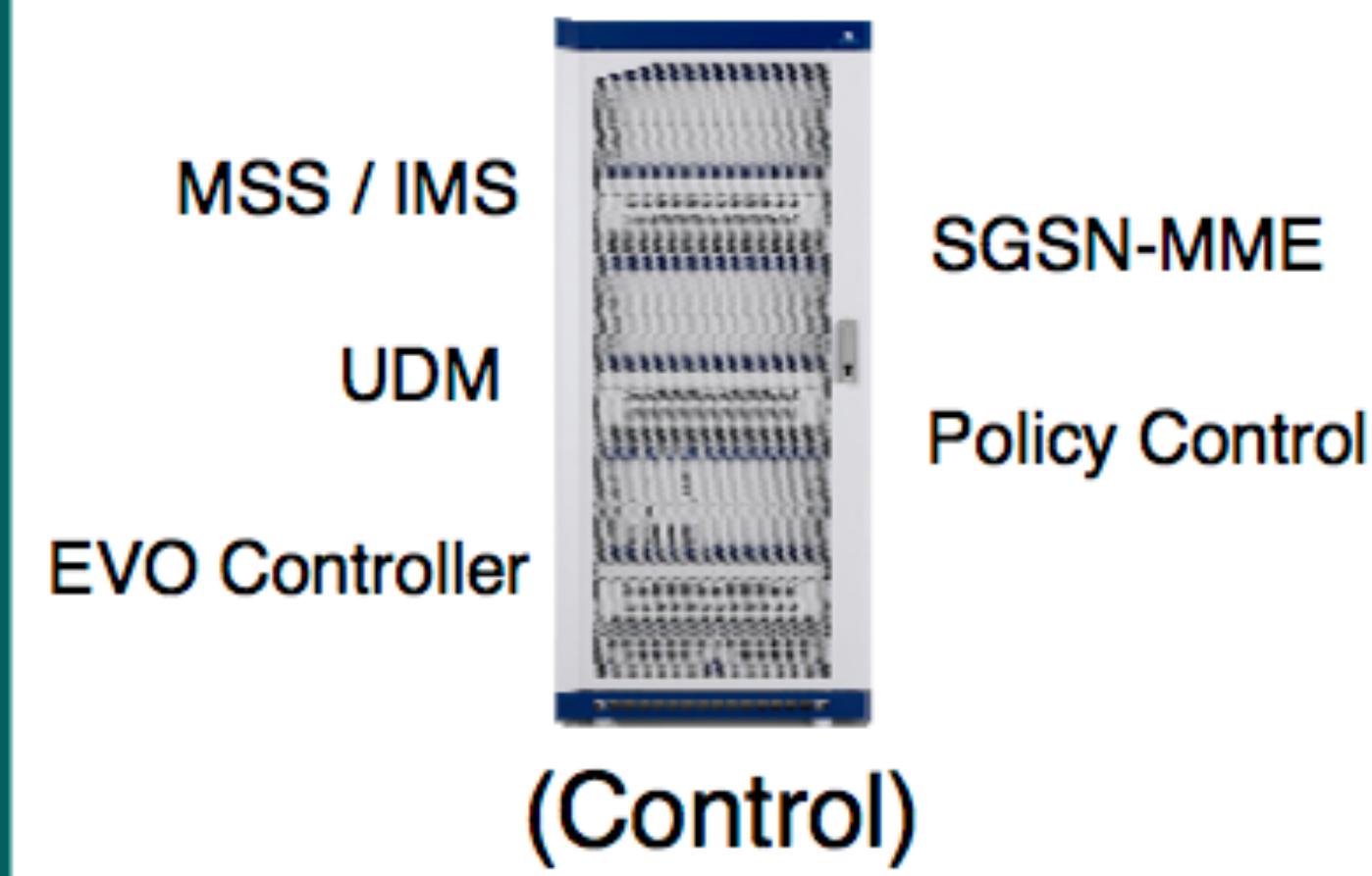
## Purpose-Built for Performance



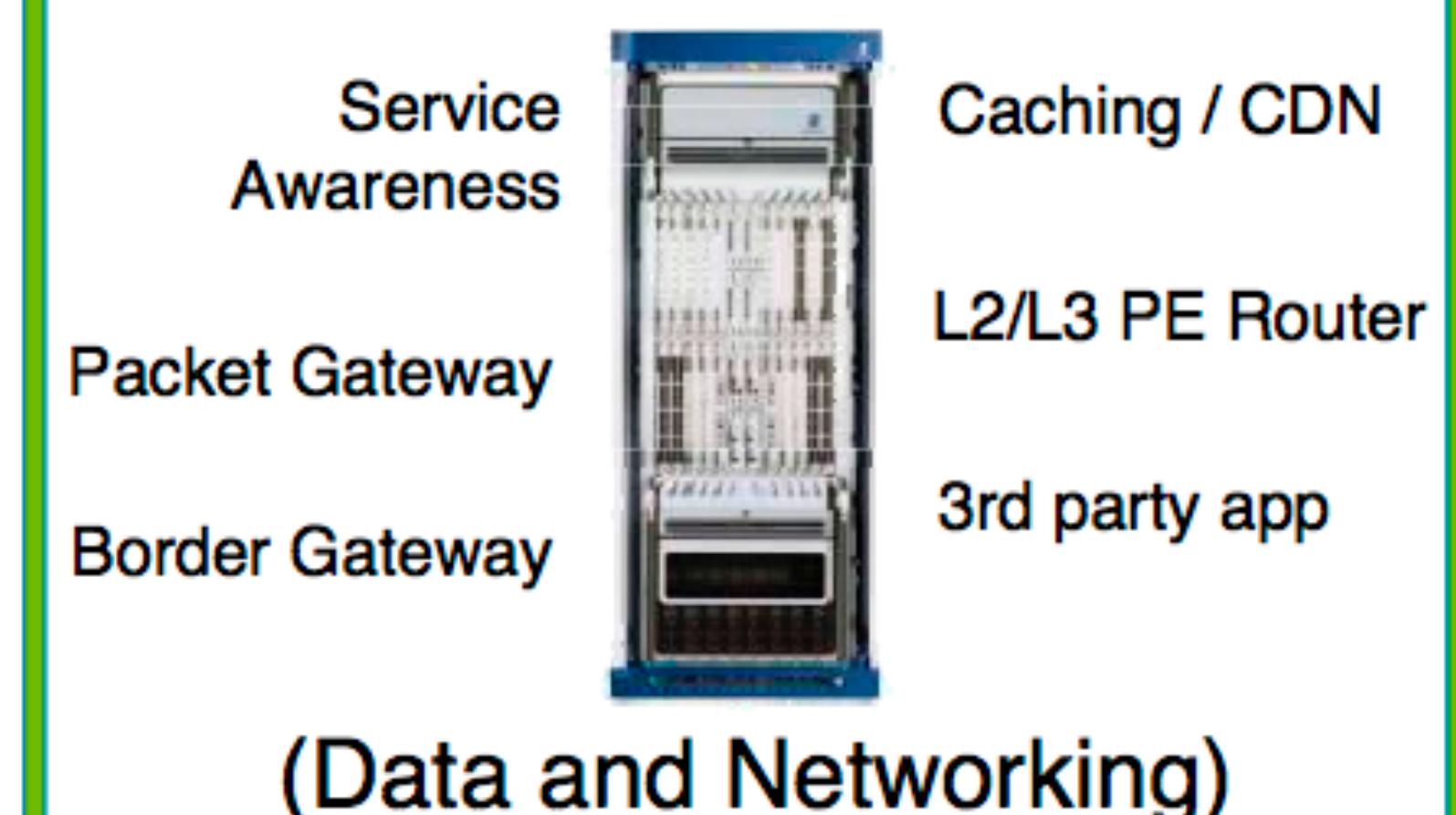
### Ericsson RBS 6000



### Ericsson Blade Server



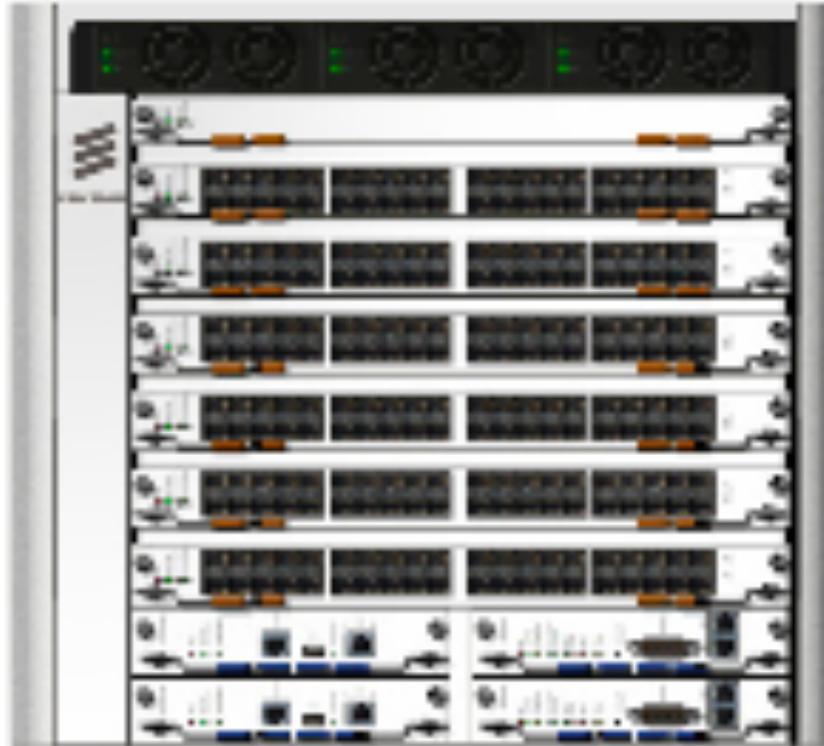
### SSR 8000 Family



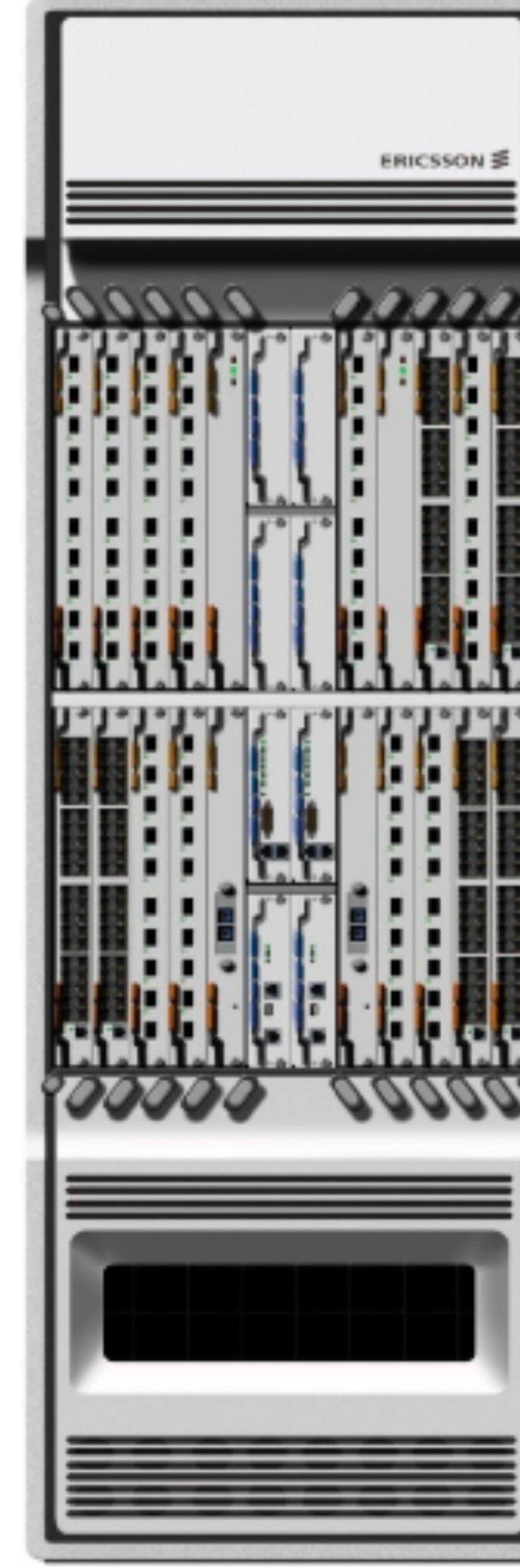
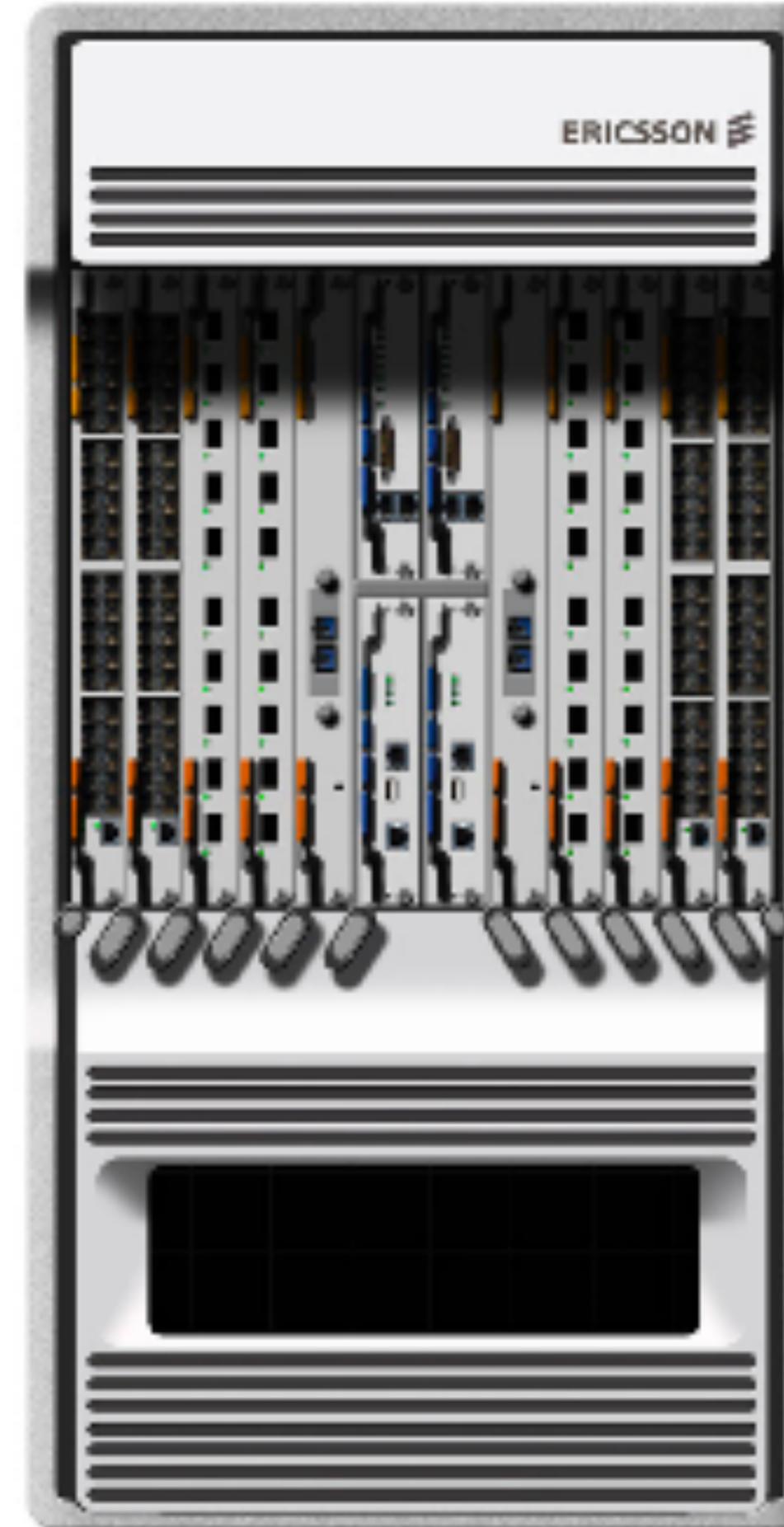
- User experience is dependent on network performance
- Ericsson systems deliver superior performance in mobile broadband
- Application separate from hardware in IP architectures
- True multi-standard, multi-application platforms deliver SDN benefits

# Purpose-Oriented Common Platform

“Three is Enough: Radio, Bearer, and Control”

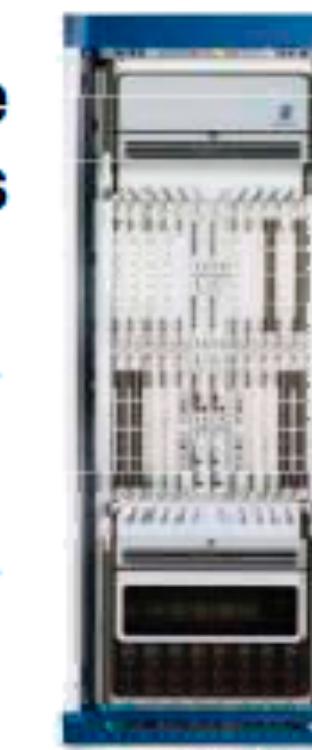


Left to right:  
Ericsson SSR 8006,  
SSR 8010, SSR 8020



## SSR 8000 Family

Service Awareness  
Packet Gateway  
Border Gateway



Caching / CDN  
L2/L3 PE Router  
3rd party app

(Data and Networking)

**Grow the capacity  
of the network in  
line with demand**

ance  
mobile broadband  
ires  
iver SDN benefits

# Purpose-Oriented Common Platform

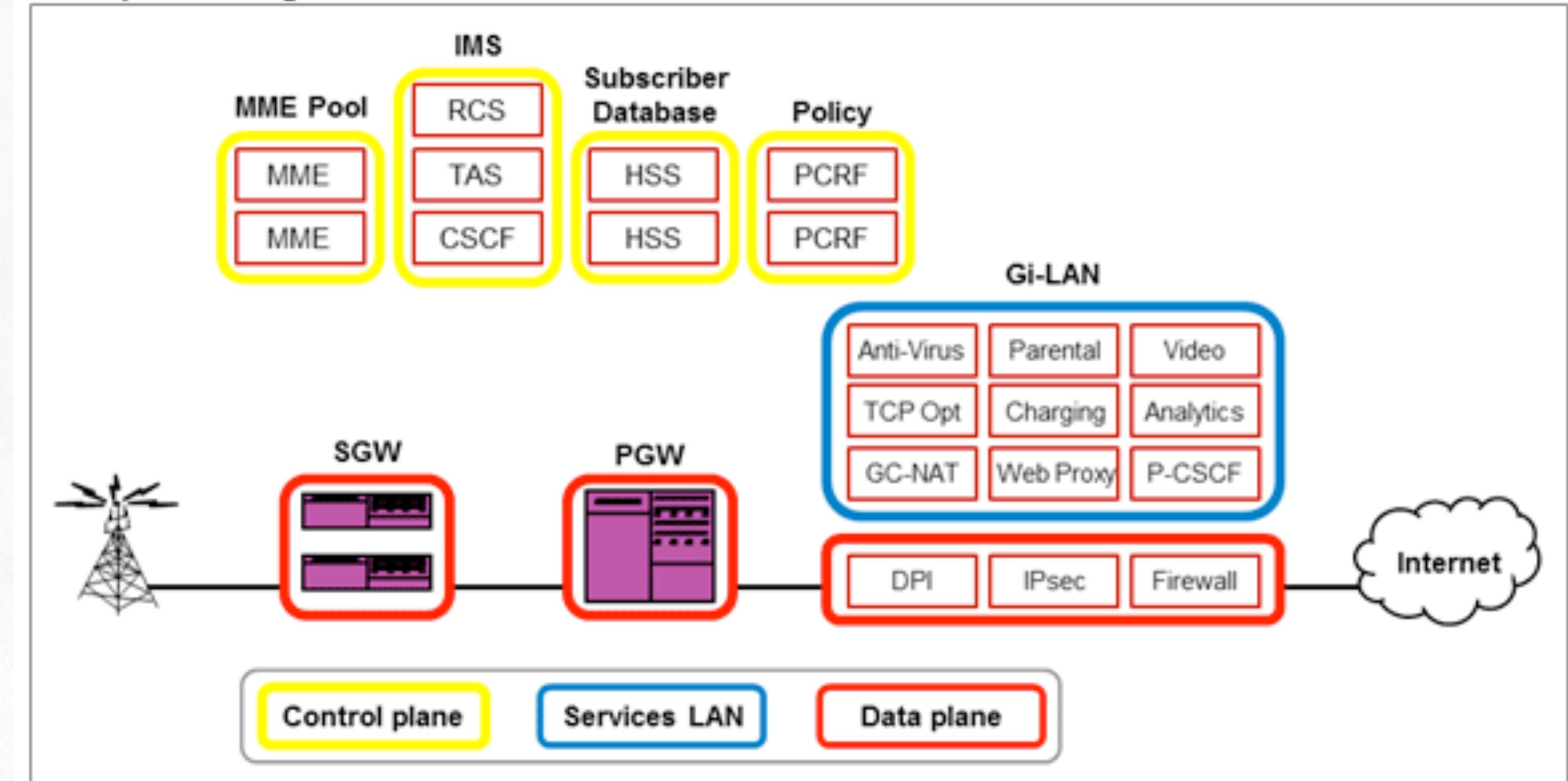
“Old Network Elements is ‘Virtualized Applications - Only’”

Applications: Ericsson shows full commitment to NFV and was at Mobile World Congress in Barcelona February 2014 launching a complete virtual Evolved Packet Core including vEPG, vSGSN-MME, vSAPC and vSASN.

Launch: Evolved Packet Core provided in a virtualized mode industrializes NFV

Since before Ericsson has a number of virtualized applications in commercial operation, e.g. virtual Media Delivery Network (vMDN), virtual Multi-Service Proxy (vMSP) and virtual Enriched Messaging (vEMe). Virtual MSP and virtual MDN provide key functionality such as Content Delivery Network (CDN), Transparent Internet Caching (TIC), video optimizations and advanced HTTP proxy features.

Excerpt 2: Categories of Mobile Packet Core Functions



Source: Heavy Reading

# Big change in Current Network

“Softwarization & Virtualization”

## Past

NE Oriented  
(eNB, SGW, PGW, MME)

Standard (Only)

Uncontrollable Network

## Today

Function Oriented  
(eNB & VNFs)

Software APIs (Major)

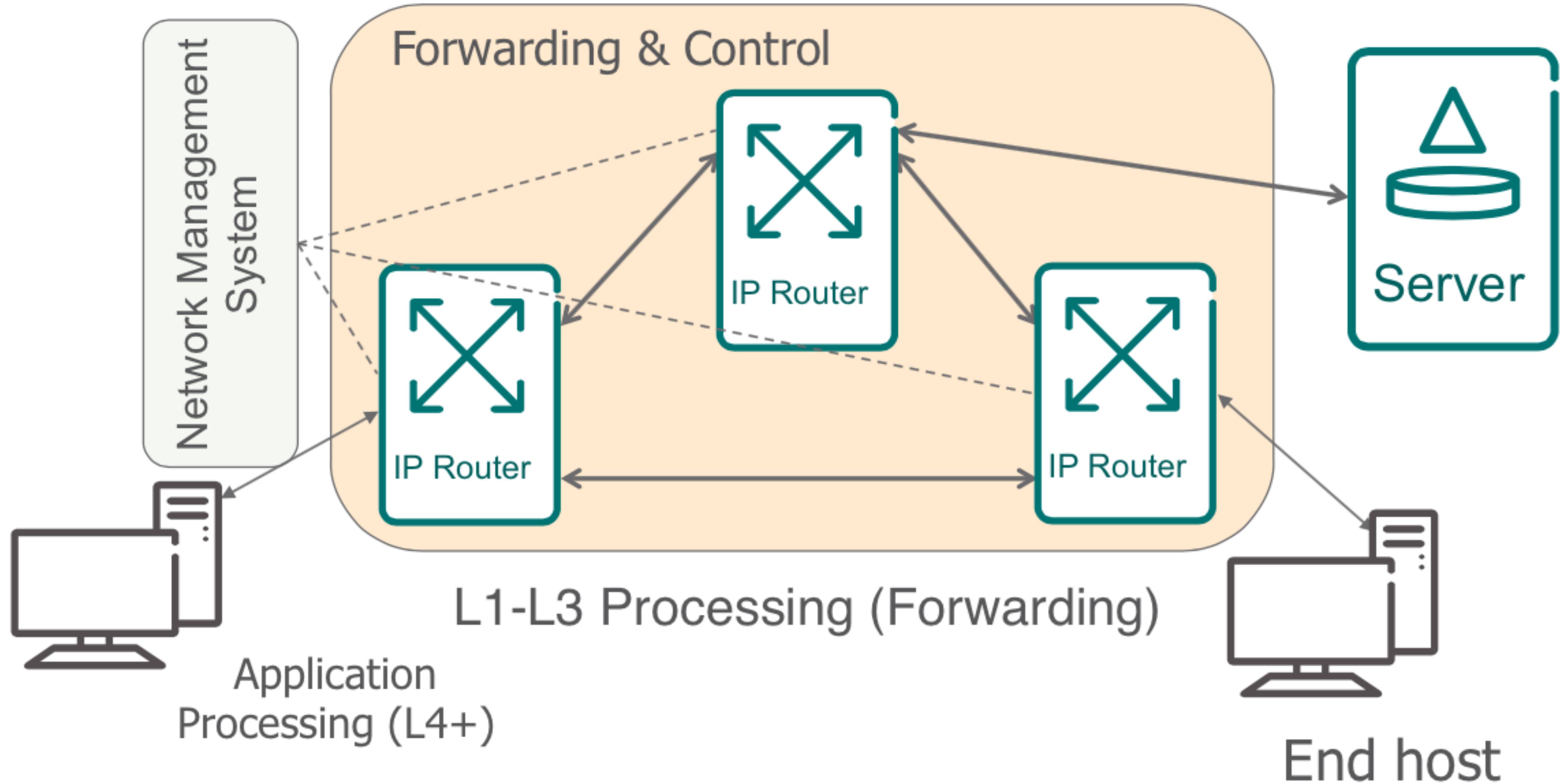
Controllable Network

# Contents

- Softwarization
- **Software Defined Network(ing)**
- Network Function Virtualization
- SDN/NFV
- SDN/NFV Softwares
- SDN/NFV Use Cases
- SDN/NFV into 5G
- P4 (programming language)

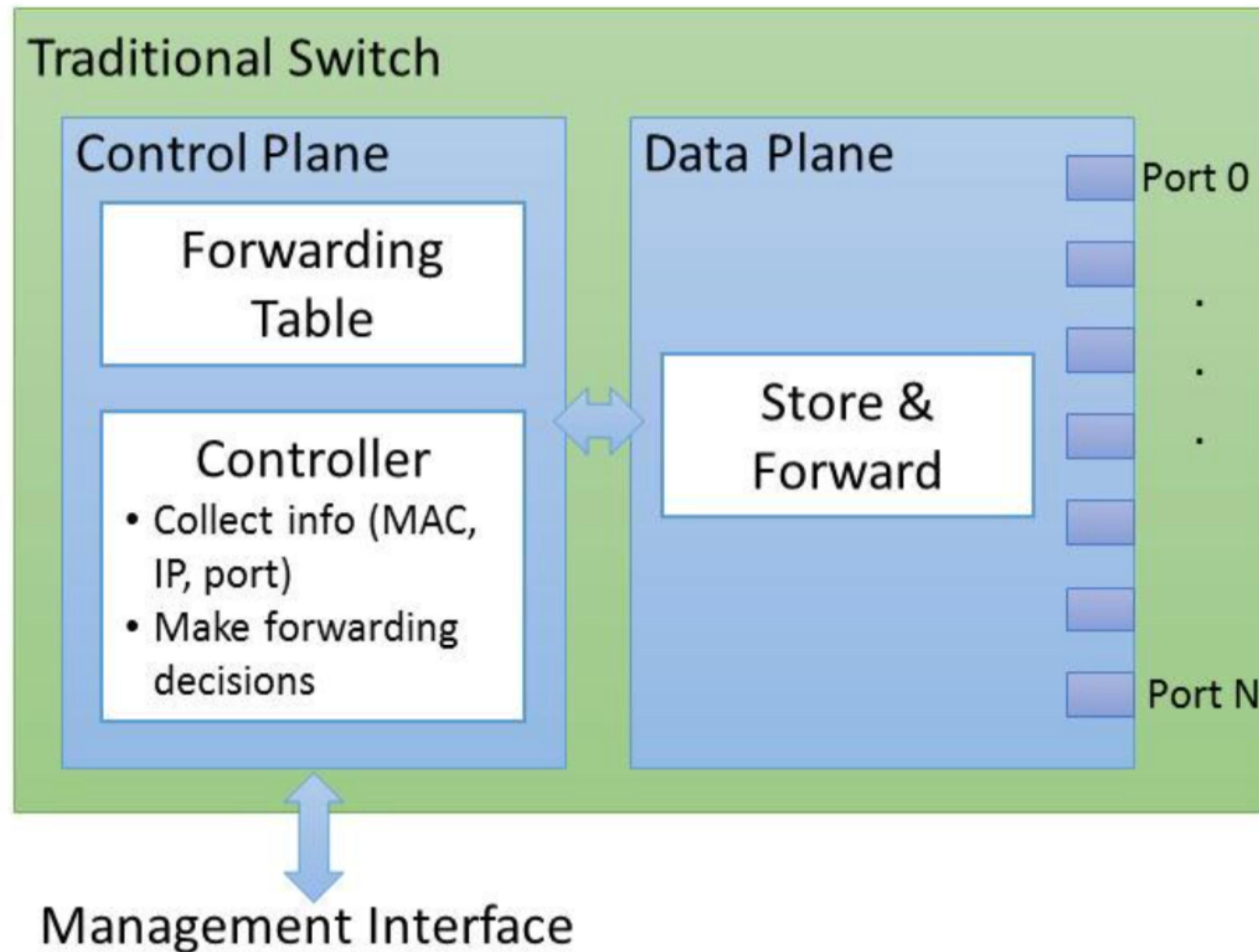
# INTERNET

“네트워크 생존성이 목적이었던 {분산} 라우터 기반 네트워크”



# ROUTER

“제어 기능과 데이터 전송 기능의 통합 장치”



# Problems on Router

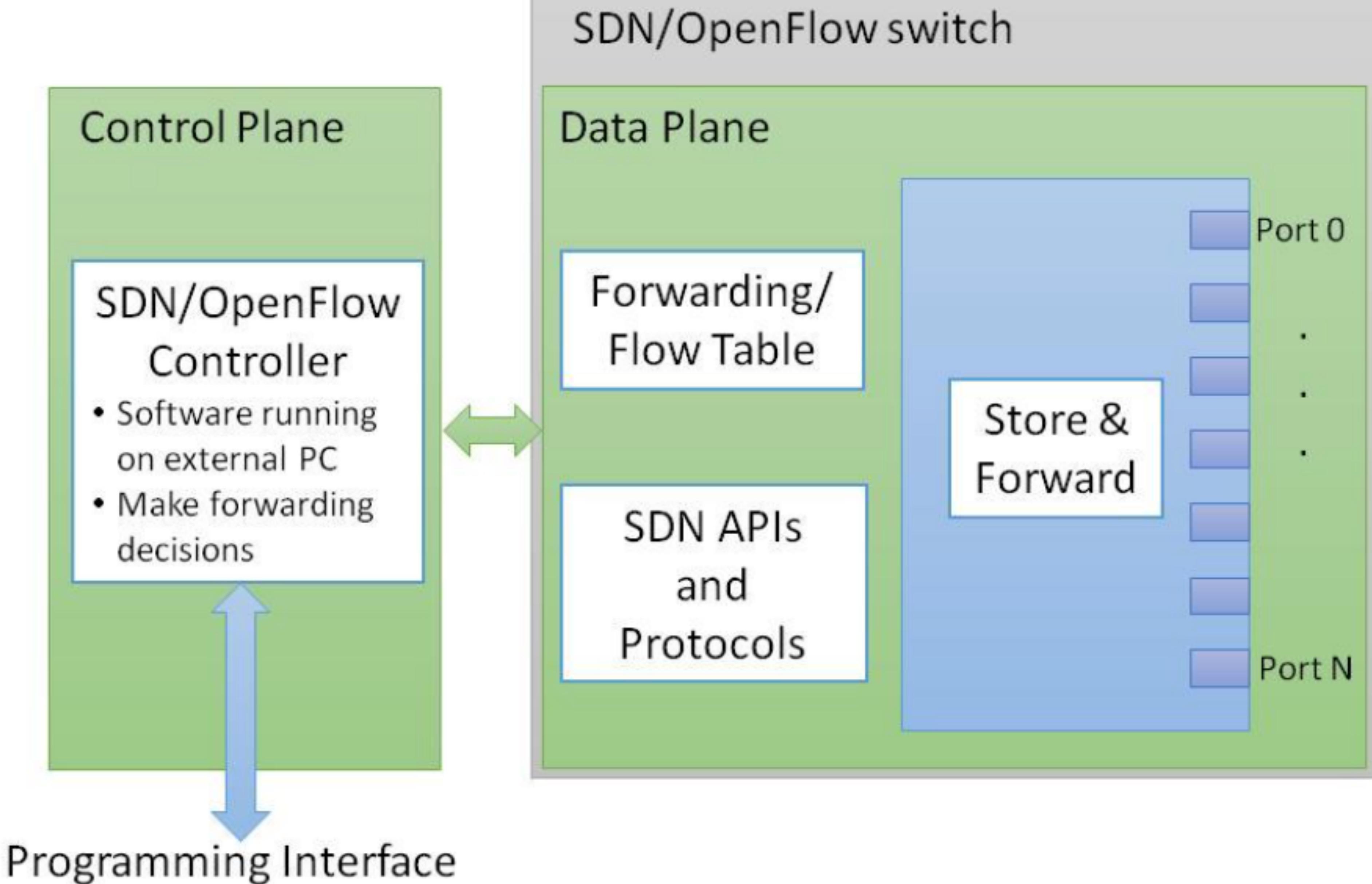
“네트워크는 단순하다. 라우터끼리 라우팅을 하니깐... That's All?”



- 새로운 통신 프로토콜
- 사용자 전용 Customized 라우팅
- 패킷 레벨 전송 경로 제어
- 서비스 인식 기반 전송 기술
- 서비스 인식 기반 패킷 처리 기술
- Cross Layer 서비스 지원
- 멀티 벤더 장비의 일관된 제어/관리

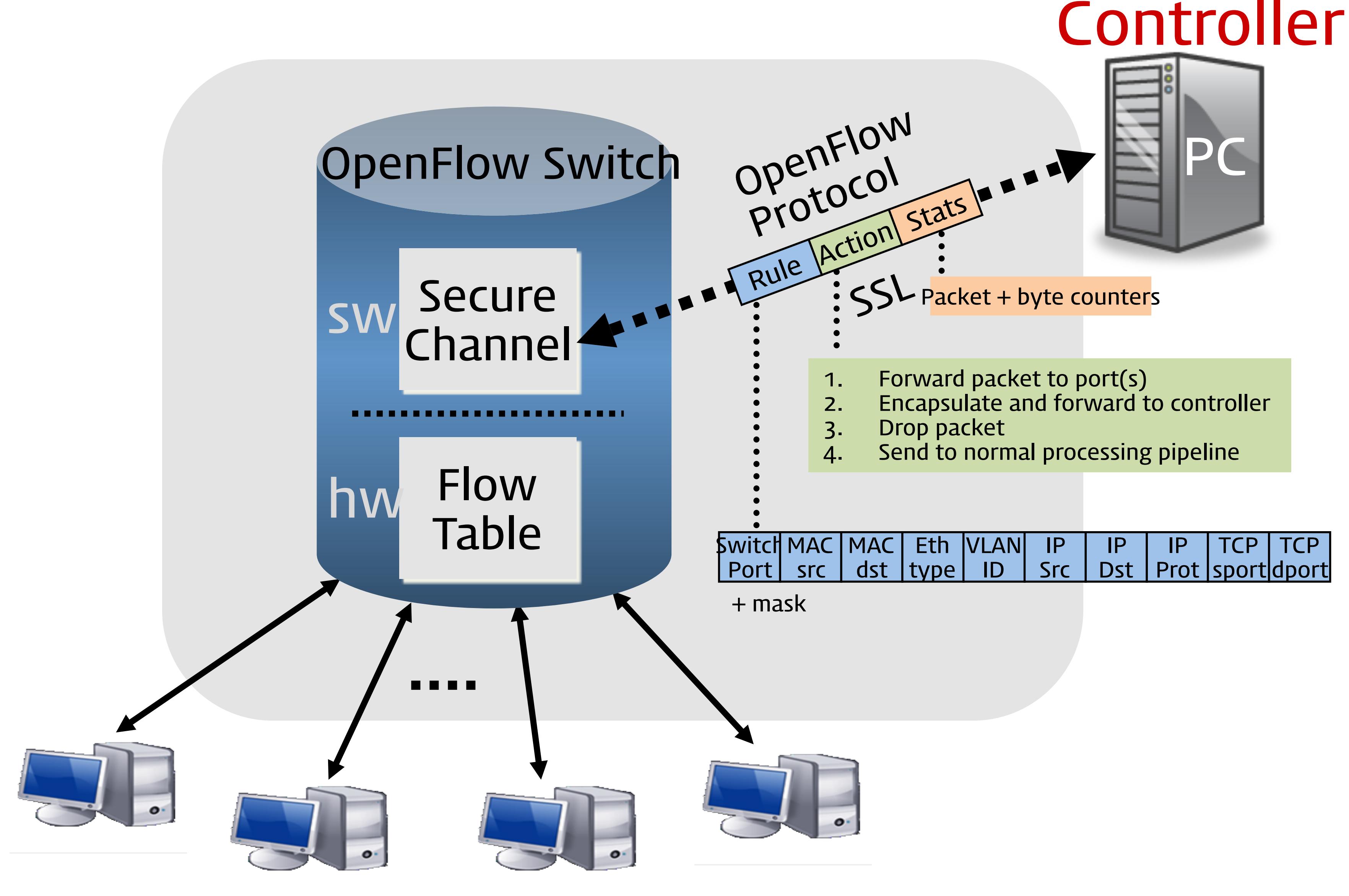
# Why SDN?

“제어와 전송의 분리 (Control & Bearer Separation)”



# OpenFlow based SDN Network

“중앙 집중형 SDN 컨트롤러와 OpenFlow 프로토콜”

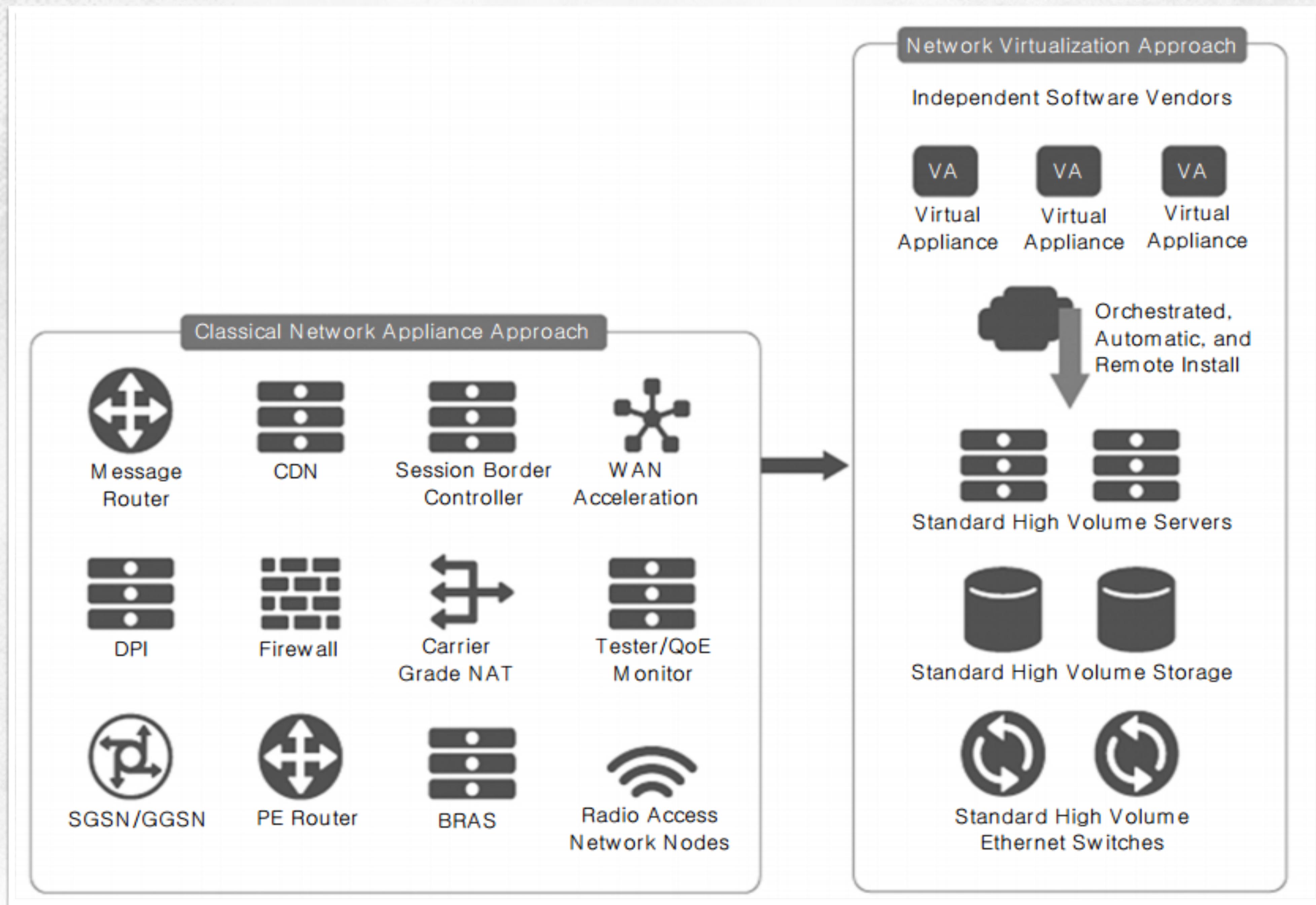


# Contents

- Softwarization
- Software Defined Network(ing)
- **Network Function Virtualization**
- SDN/NFV
- SDN/NFV Softwares
- SDN/NFV Use Cases
- SDN/NFV into 5G
- P4 (programming language)

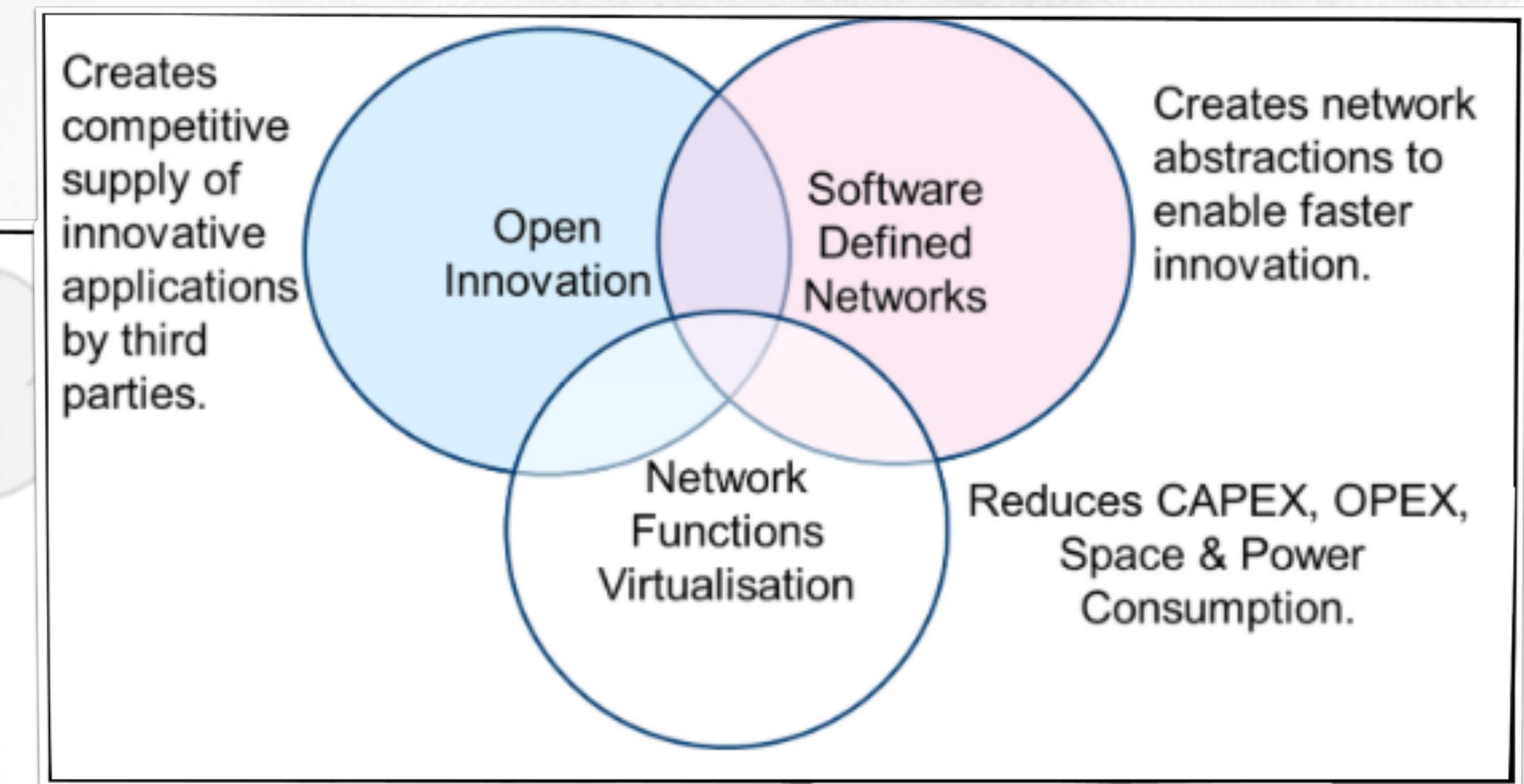
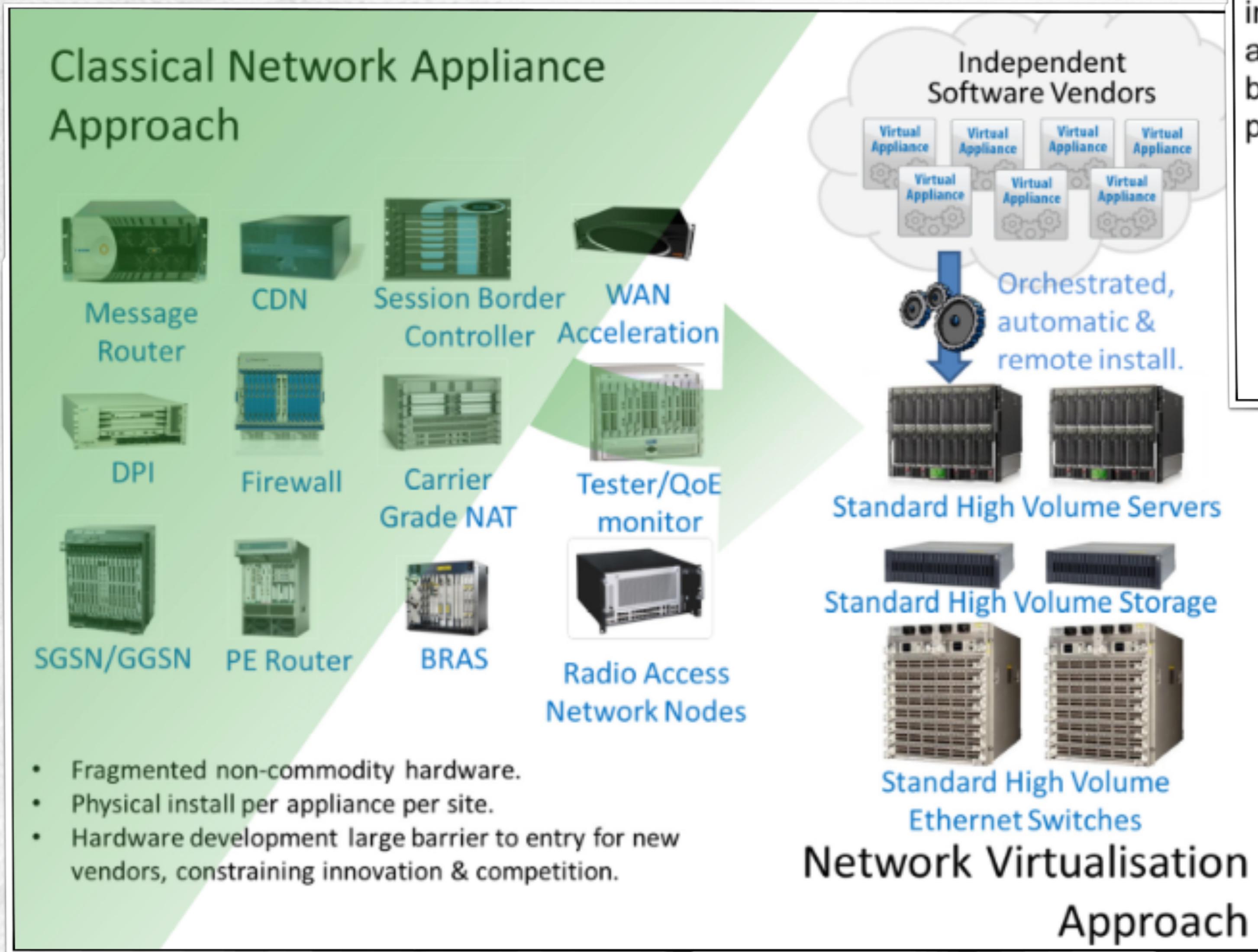
# NFV

“Network Function Virtualization ?”



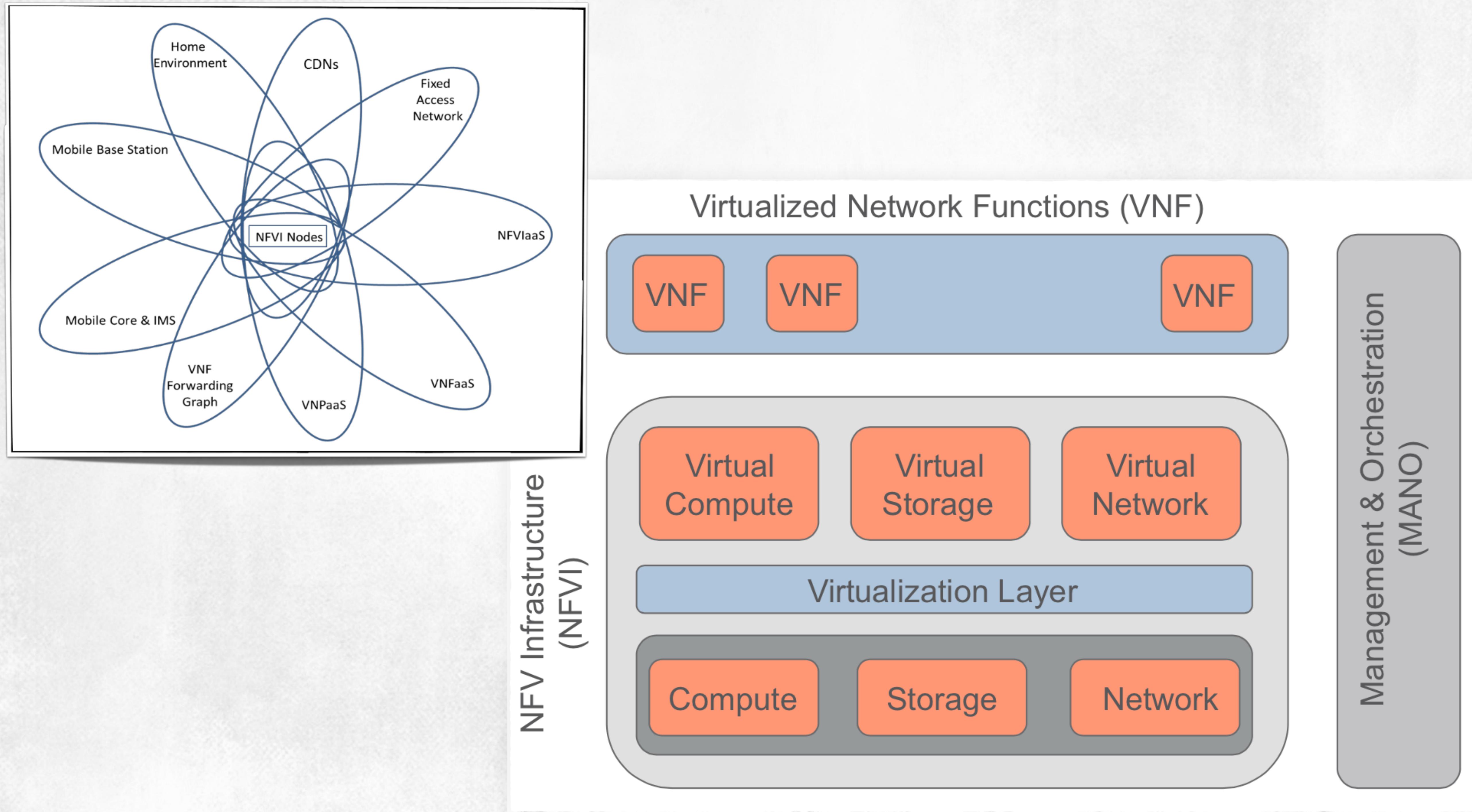
# NFV ≠ SDN

## “White Paper 1.0”



# NFV ≠ SDN

“White Paper 2.0 : NFVI Supports NFVs and Fields of Application”



출처: Ahmad Rostami, “The Evolution of Programmable Networks”

출처: <http://www.etsi.org/technologies-clusters/technologies/nfv>

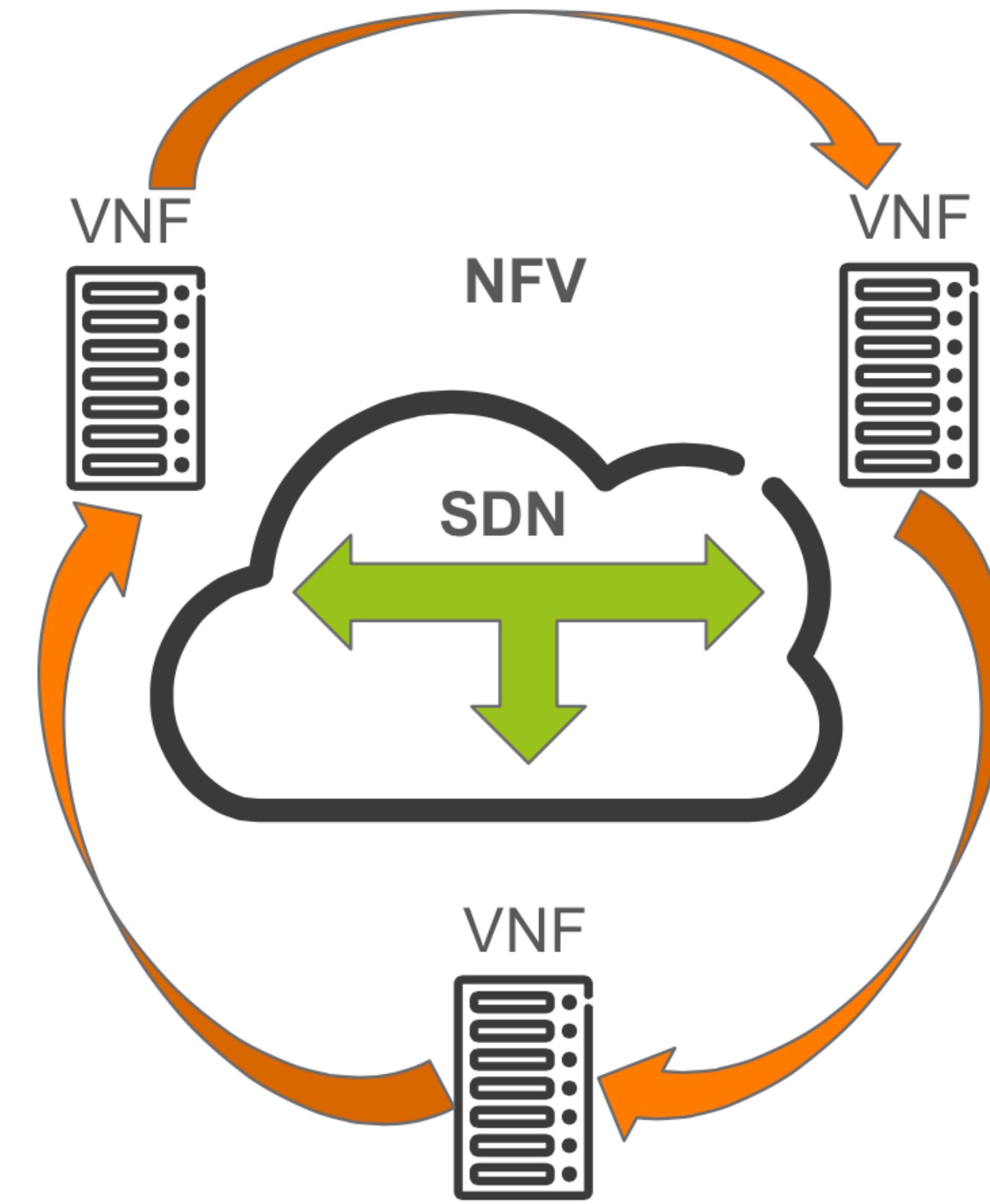
# Contents

- Softwarization
- Software Defined Network(ing)
- Network Function Virtualization
- **SDN/NFV**
- SDN/NFV Softwares
- SDN/NFV Use Cases
- SDN/NFV into 5G
- P4 (programming language)

# SDN & NFV

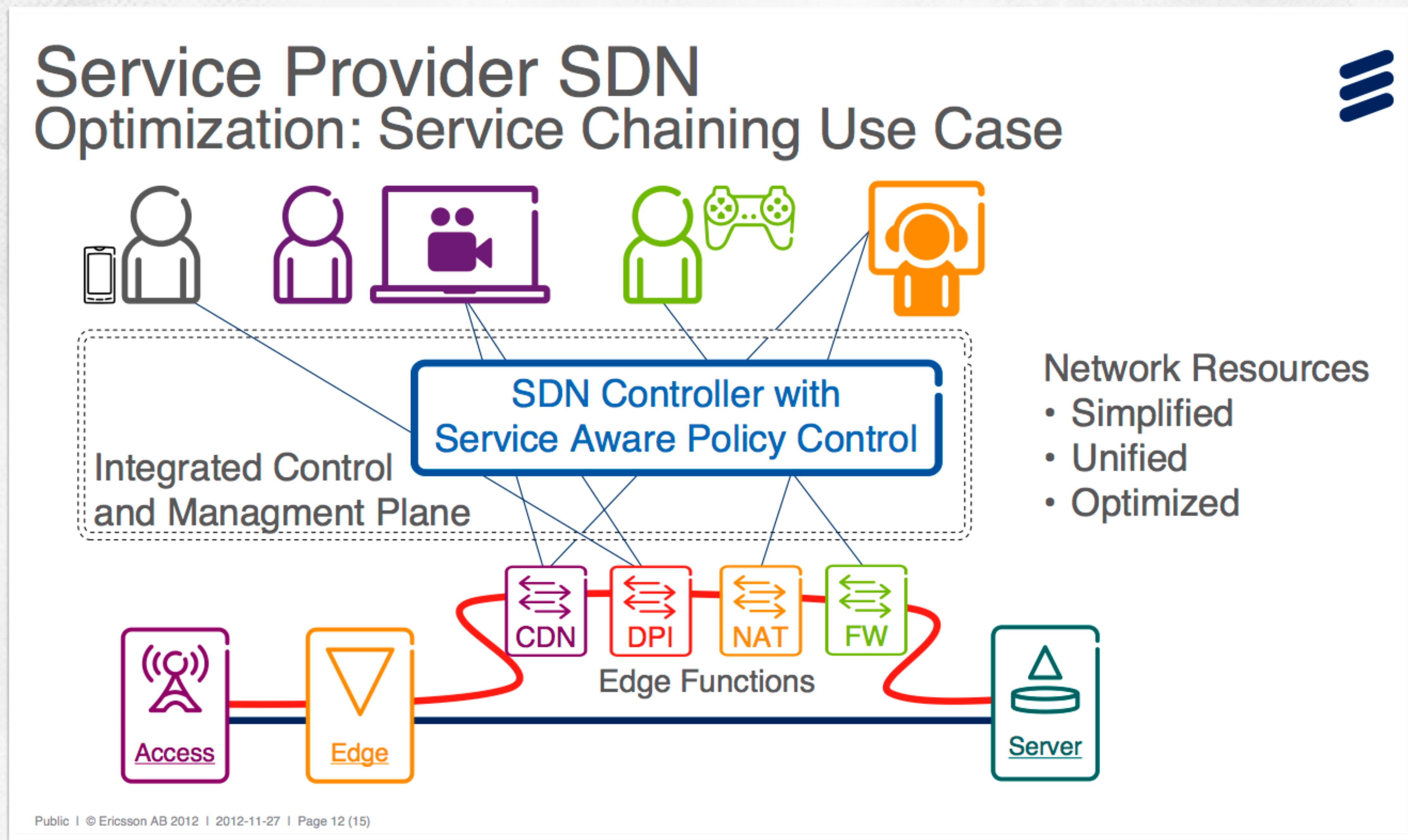
“SDN is traffic forwarding (in original) & NFV is about Box”

- › SDN → flexible forwarding & steering of traffic in a physical or virtual network environment
- › NFV → flexible placement of virtualized network functions across the network & cloud
- › SDN & NFV are complementary tools for achieving full network programmability



# Service Chaining as a New Service

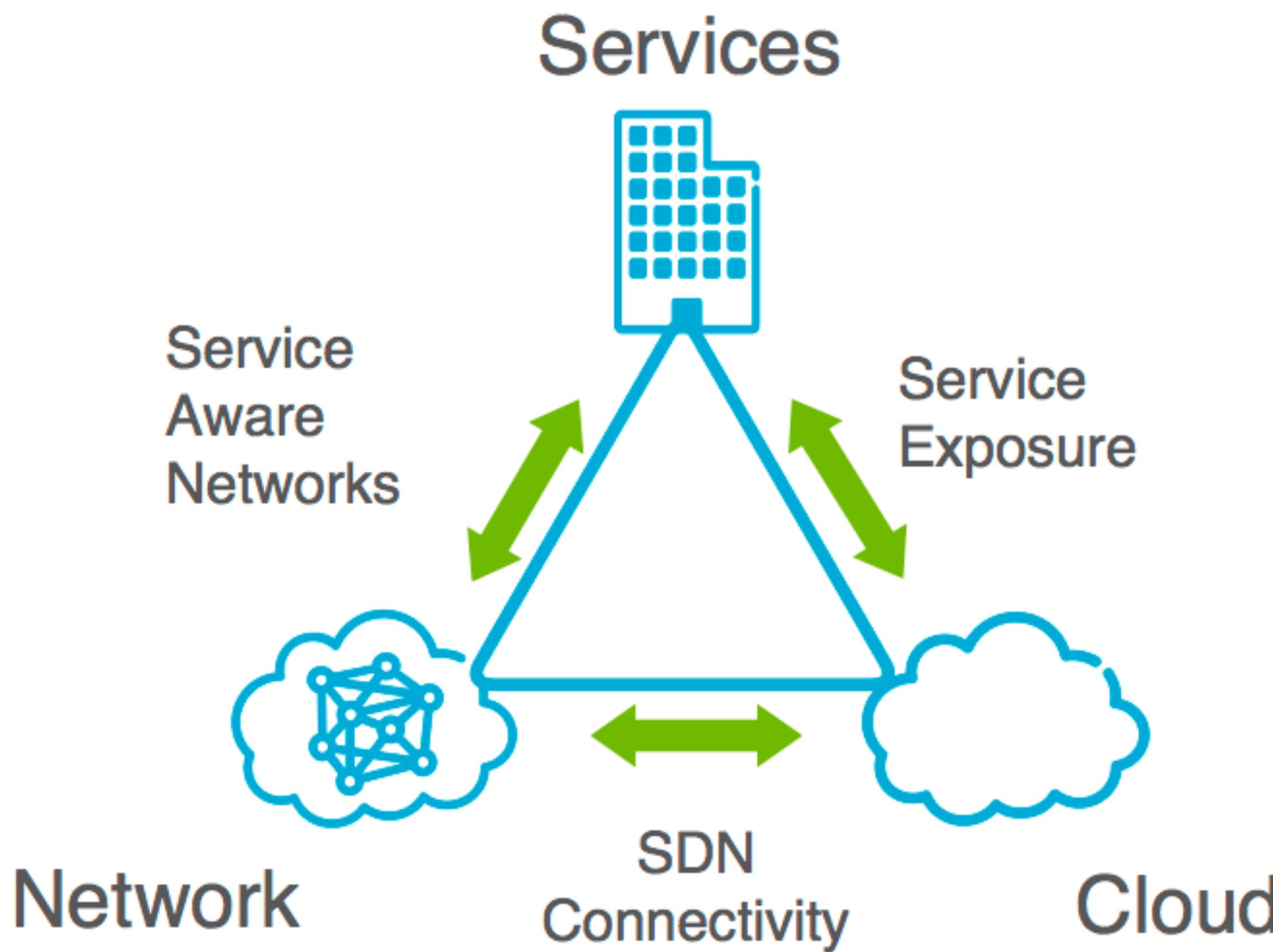
“SDN Deployment in Mobile Network”



# What is NETWORK?

Software, Service, Service ...

## From Connectivity to Experience Provider



Service providers expand the definition of SDN:

- Enterprise cloud services
- Network and Cloud management
- Service exposure
- Purpose-built performance
- Integrated network control

# Contents

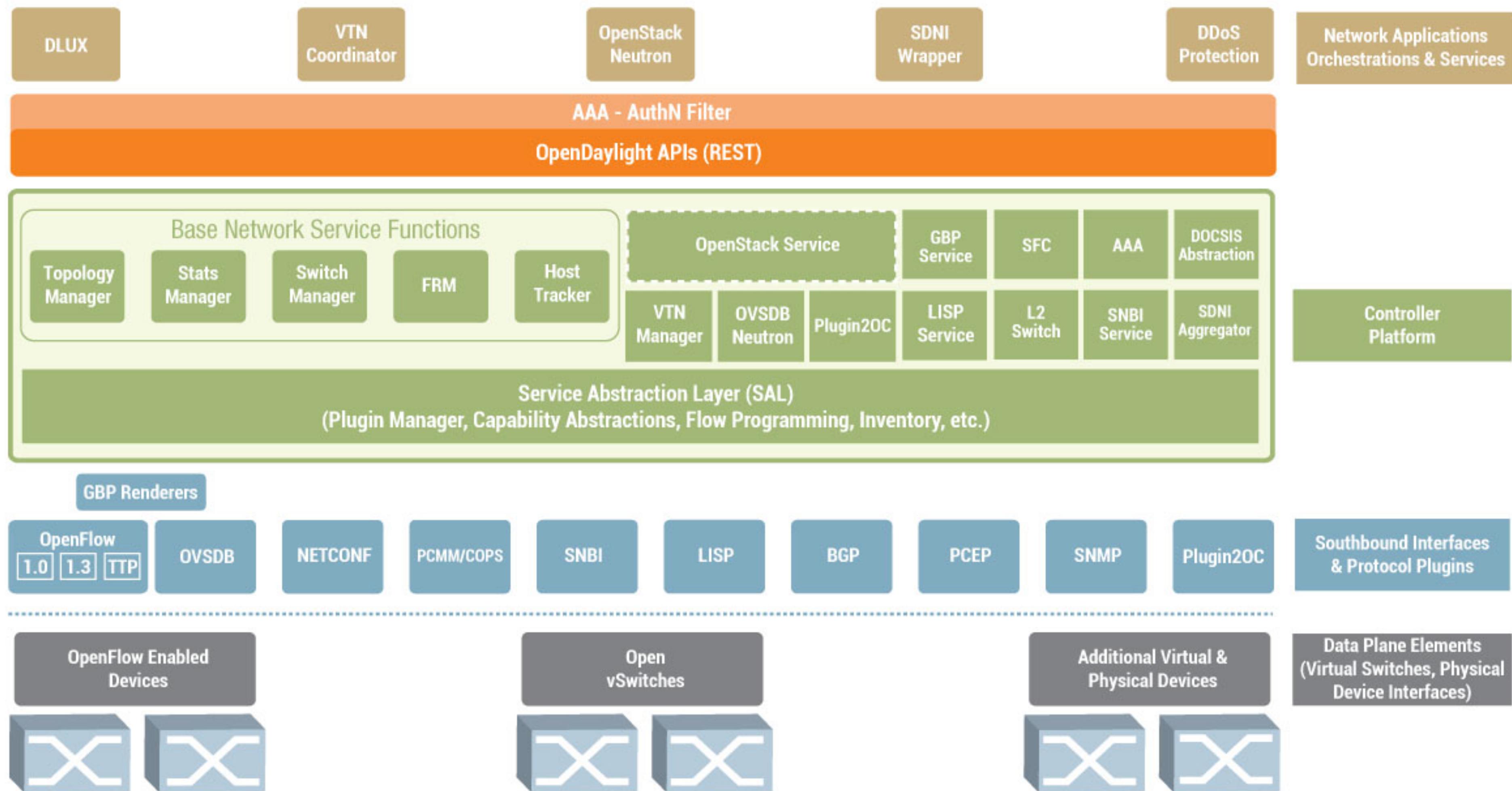
- Softwarization
- Software Defined Network(ing)
- Network Function Virtualization
- SDN/NFV
- **SDN/NFV Softwares**
- SDN/NFV Use Cases
- SDN/NFV into 5G
- P4 (programming language)

# OpenDaylight

[www.opendaylight.org](http://www.opendaylight.org)



LEGEND	
AAA:	Authentication, Authorization & Accounting
AuthN:	Authentication
BGP:	Border Gateway Protocol
COPS:	Common Open Policy Service
DLUX:	OpenDaylight User Experience
DDoS:	Distributed Denial Of Service
DOCSIS:	Data Over Cable Service Interface Specification
FRM:	Forwarding Rules Manager
GBP:	Group Based Policy
LISP:	Locator/Identifier Separation Protocol
OVSDB:	Open vSwitch DataBase Protocol
PCEP:	Path Computation Element Communication Protocol
PCMM:	Packet Cable MultiMedia
Plugin2OC:	Plugin To OpenContrail
SDNI:	SDN Interface (Cross-Controller Federation)
SFC:	Service Function Chaining
SNBI:	Secure Network Bootstrapping Infrastructure
SNMP:	Simple Network Management Protocol
TTP:	Table Type Patterns
VTN:	Virtual Tenant Network



# Open Network Operating System

[onlab.us](http://onlab.us)

**ON.LAB**

About Us

Projects

News

Contact Us

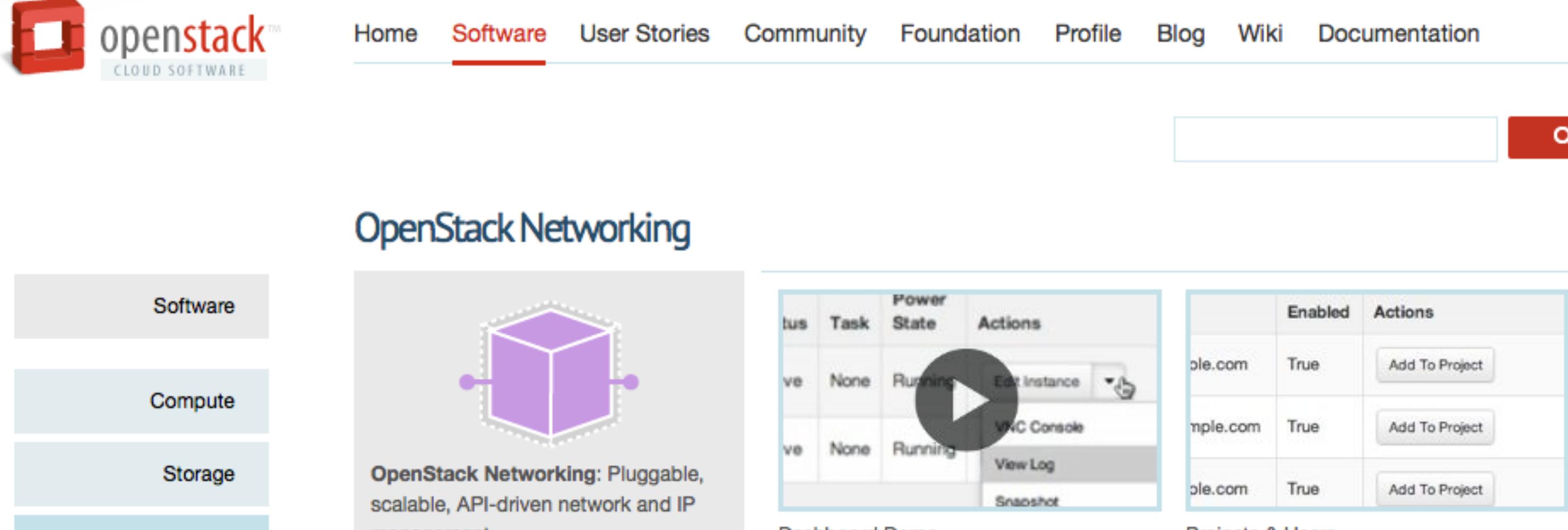


The world of SDN is going to change  
in a big way on December 5th.

ONOS is going open source. [Learn More](#)

# OpenStack Networking

[www.openstack.org/software/openstack-networking/](http://www.openstack.org/software/openstack-networking/)



The screenshot shows the OpenStack Networking page. On the left, there's a sidebar with links: Software, Compute, Storage, Networking (which is highlighted), Dashboard, Shared Services, Getting Started (highlighted in green), Roadmap, OpenStack Marketplace, and Latest Release. The main content area has a title "OpenStack Networking". It features three main sections: "OpenStack Networking: Pluggable, scalable, API-driven network and IP management" with a purple cube icon, "Dashboard Demo" showing a table of instances (with one instance selected), and "Projects & Users" showing a table of projects. Below these sections, there's a text block about the Neutron project providing virtual networking services, mentioning the Networking API v2.0, Quantum API v1.1, and Melange API. It also lists IPAM capabilities and describes the introduction of the subnet entity in the Networking API v2.0.

The Neutron project provides virtual networking services among devices that are managed by the [OpenStack](#) compute service.

The Networking API v2.0 combines the [Quantum API v1.1](#) with some essential Internet Protocol Address Management (IPAM) capabilities from the [Melange API](#).

These IPAM capabilities enable you to:

- Associate IP address blocks and other network configuration settings required by a network device, such as a default gateway and dns-servers settings, with an OpenStack Networking network.
- Allocate an IP address from a block and associate it with a device that is attached to the network through an OpenStack Networking port.

To do this, the Networking API v2.0 introduces the subnet entity. A subnet can represent either an IP version 4 or version 6 address block. Each OpenStack Networking network commonly has one or more subnets. When you create a port on the network, an available fixed IP address is allocated to it from one of the designated subnets for each IP version. When you delete the port, the allocated addresses return to the pool of available IPs on the subnet. Networking API v2.0 users can choose a specific IP address from the block or let OpenStack Networking choose the first available IP address.

# Open Platform for NFV

[www.opnfv.org](http://www.opnfv.org)

The screenshot shows the official website for the Open Platform for NFV (OPNFV). At the top, there's a header with the Linux Foundation logo and the text "COLLABORATIVE PROJECTS". Below the header is the OPNFV logo. The main navigation menu includes links for "ABOUT", "SOFTWARE", "DEVELOPERS", "NEWS & RESOURCES", and "CONTACT US". On the right side of the header is a search bar. The main content area features a large heading "About" followed by several paragraphs of text describing the project's mission, initial scope, and collaboration with ETSI's NFV ISG.

**LINUX FOUNDATION COLLABORATIVE PROJECTS**

**OPNFV**

ABOUT SOFTWARE DEVELOPERS NEWS & RESOURCES CONTACT US

Search

[HOME](#) »About

## About

Open Platform for NFV (OPNFV) is a new open source project focused on accelerating the evolution of Network Functions Virtualization (NFV). OPNFV will establish a carrier-grade, integrated, open source reference platform that industry peers will build together to advance the evolution of NFV and to ensure consistency, performance and interoperability among multiple open source components. Because multiple open source NFV building blocks already exist, OPNFV will work with upstream projects to coordinate continuous integration and testing while filling development gaps.

The initial scope of OPNFV will be on building NFV Infrastructure (NFVI), Virtualized Infrastructure Management (VIM), and including application programmable interfaces (APIs) to other NFV elements, which together form the basic infrastructure required for Virtualized Network Functions (VNF) and Management and Network Orchestration (MANO) components. OPNFV is expected to increase performance and power efficiency; improve reliability, availability, and serviceability; and deliver comprehensive platform instrumentation.

OPNFV will work closely with ETSI's NFV ISG, among other Standards Development Organizations (SDOs), to drive consistent implementation of standards for an open NFV reference platform. Increasingly, standards are being drafted in conjunction with major open source projects. Since feedback from open source implementations can drive the rapid evolution and adoption of standards, this tight coordination of otherwise independent processes is crucial to the establishment of an NFV ecosystem. When open source software development is aligned with standards development, it can root out issues earlier, identify resolutions, and potentially establish de facto standards, resulting in a far more economical approach to platform development.

# Open Network Linux

[opennetlinux.org](http://opennetlinux.org)

**Open Network Linux** is a Linux distribution for "bare metal" switches, that is, network forwarding devices built from commodity components. ONL uses **ONIE** to install onto on-board flash memory. Open Network Linux is a part of the **Open Compute Project** and is a component in a growing collection of open source and commercial projects.



## Hardware Support

Because of the HTML formatting, this page may be best viewed from  
<http://opennetlinux.org/hcl>

### Quanta

Device	Ports	CPU	Forwarding	Support Status
QuantaMesh T1048-LB9	48x1G + 4x10G	P2020	Broadcom BCM56534 (Firebolt3)	Supported and Tested
QuantaMesh T1048-LB9A	48x1G + 4x10G	P2020	Broadcom BCM56534 (Firebolt3)	Supported and Tested
QuantaMesh T3048-LY2	48x10G + 4x40G	P2020	Broadcom BCM56846 (Trident+)	Supported and Tested

### Accton/Edge-Core

Device	Ports	CPU	Forwarding	Support Status
Accton AS4600-54T	48x1G + 4x10G	P2020	Broadcom BCM56540 (Apollo2)	Supported and Tested
Accton AS5600-52X	48x10G + 4x40G	P2020	Broadcom BCM56846 (Trident+)	Supported and Tested
Accton AS5610-52X	48x10G + 4x40G	P2020	Broadcom BCM56846 (Trident+)	Supported and Tested
Accton AS5710-54X	48x10G + 6x40G	P2041	Broadcom BCM56854 (Trident2)	Supported and Tested
Accton AS6700-32X	32x40G	P2041	Broadcom BCM56850 (Trident2)	Supported and Tested

### DNI/Agema

Device	Ports	CPU	Forwarding	Support Status
AG-7448CU	48x10G + 4x40G	P2020	Broadcom BCM56845 (Trident)	Supported and Tested

# OpenFlow enabled OpenWRT

[github.com/CPqD/ofsoftswitch13/wiki/OpenFlow-1.3-for-OpenWRT](https://github.com/CPqD/ofsoftswitch13/wiki/OpenFlow-1.3-for-OpenWRT)

The screenshot shows a GitHub wiki page for the repository CPqD/ofsoftswitch13. The page title is "OpenFlow 1.3 for OpenWRT". It contains text about the project, sections for "Pre-compiled image" and "Build instructions", and a sidebar with navigation links and clone options.

**GitHub Header:** GitHub, This repository, Search, Explore, Features, Enterprise, Blog, Sign up, Sign In

**Repository Header:** CPqD / ofsoftswitch13, Star 81, Fork 75

**Page Title:** OpenFlow 1.3 for OpenWRT

**Text Content:** srujan edited this page on Oct 11, 2013 · 4 revisions

**Text Content (continued):** Following the steps of Pantou, which turns OpenWrt router into OpenFlow switches, we developed a version using our OpenFlow 1.3 software switch, enabling to test the most recent version of the protocol on a real device, for a low cost.

We build images for the TP-LINK TL-WR1043ND. If you have interest on build for other devices and want to share it, please contact us, so we could keep an images repository.

**Section: Pre-compiled image**

If you do not want to build the code you can download pre-compiled images from

**Section: Build instructions**

**Sidebar:**

- Pages 5
  - Dpctl Documentation
  - Dpctl Flow Mod Cases
  - Home
  - OpenFlow 1.3 for OpenWRT
  - OpenFlow 1.3 Tutorial

**Clone Options:**

- Clone this wiki locally
  - <https://github.com/CPqD/ofsoftswitch13>
  - Clone in Desktop

# Contents

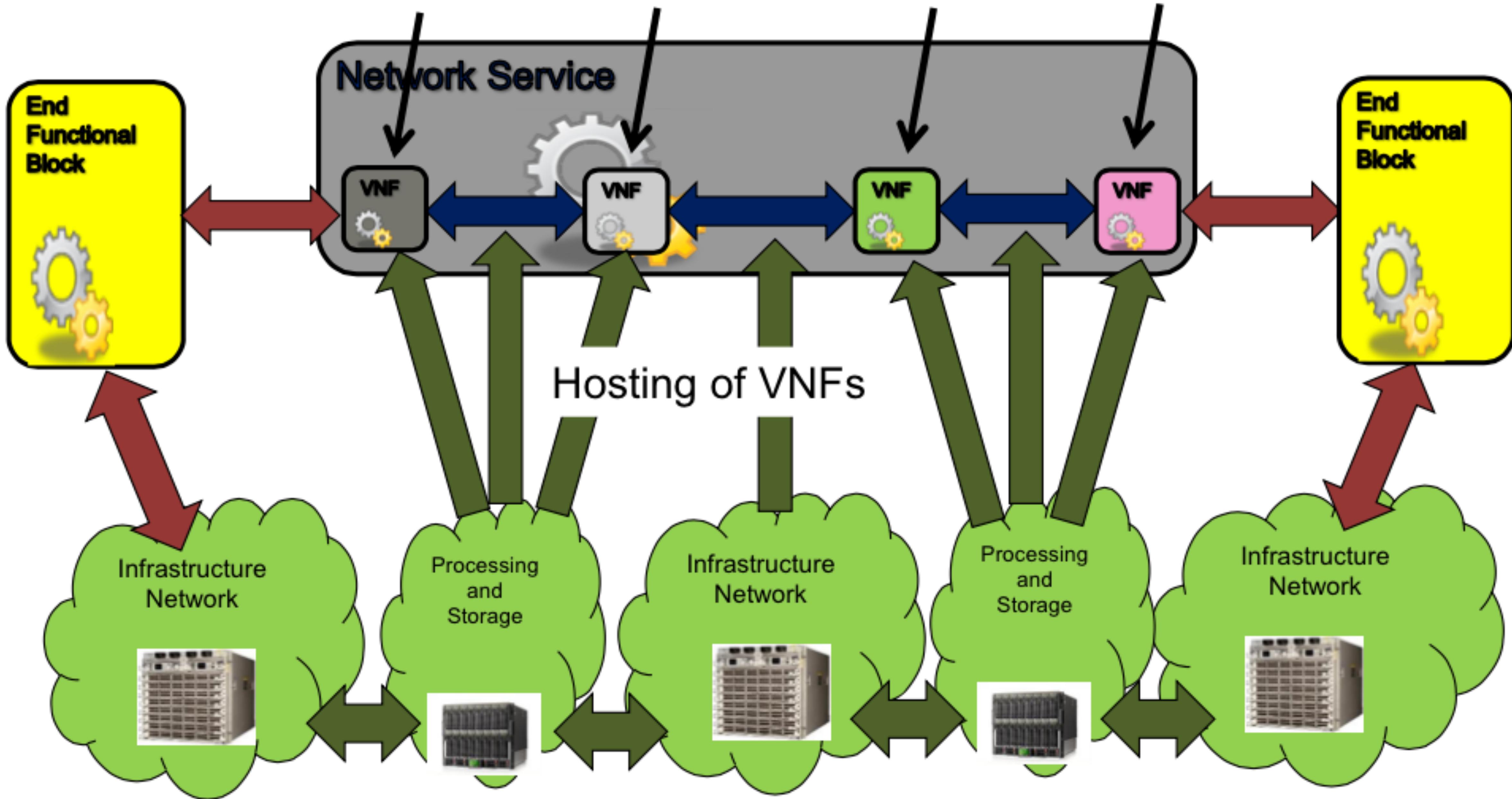
- Softwarization
- Software Defined Network(ing)
- Network Function Virtualization
- SDN/NFV
- SDN/NFV Softwares
- **SDN/NFV Use Cases**
- SDN/NFV into 5G
- P4 (programming language)

# Use Cases in REAL

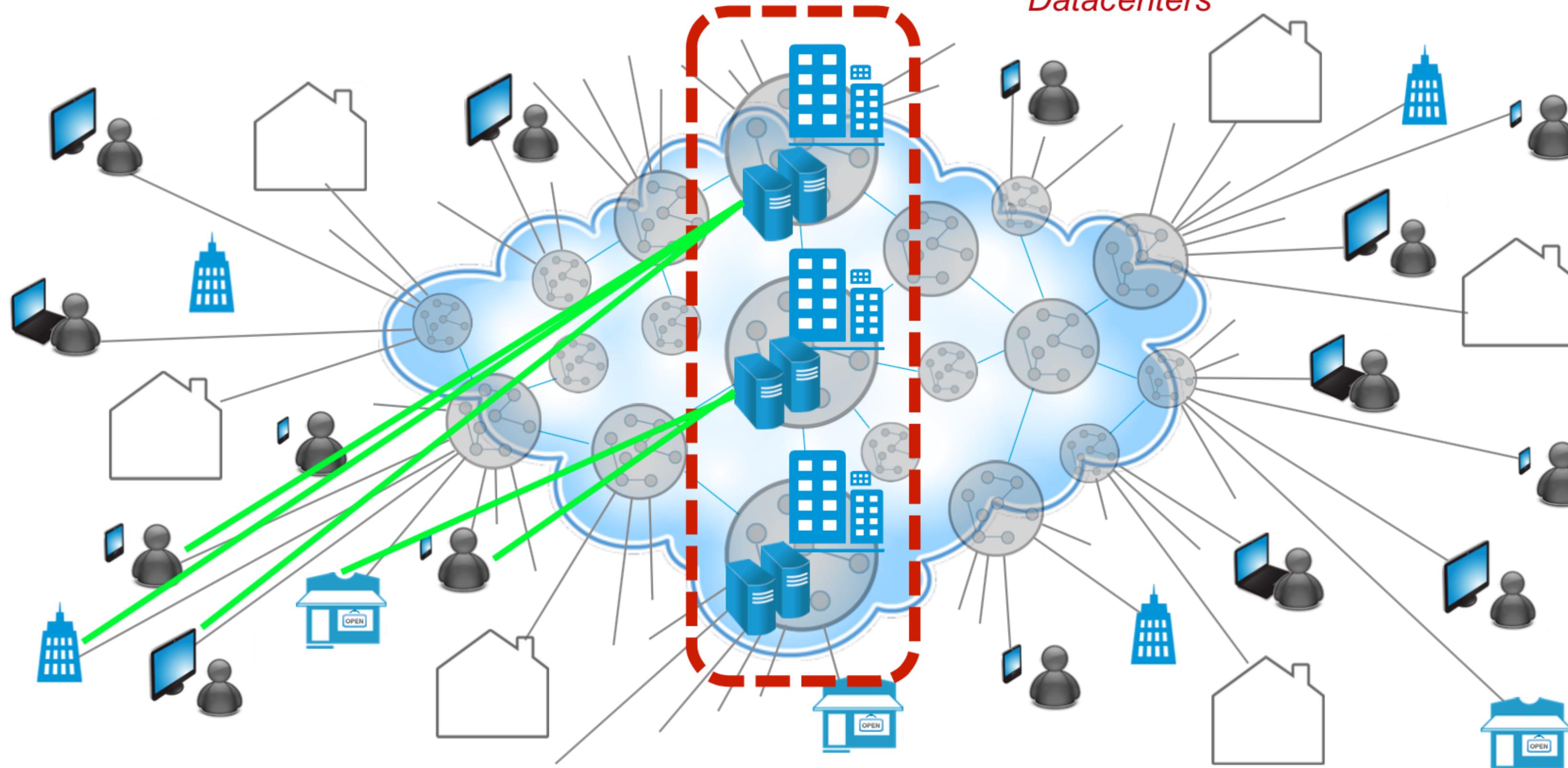


vCPE/FW/NAT/Encryption/Virus-Vaccine/DPI/etc.,

*Dynamic creation of chain of service elements*

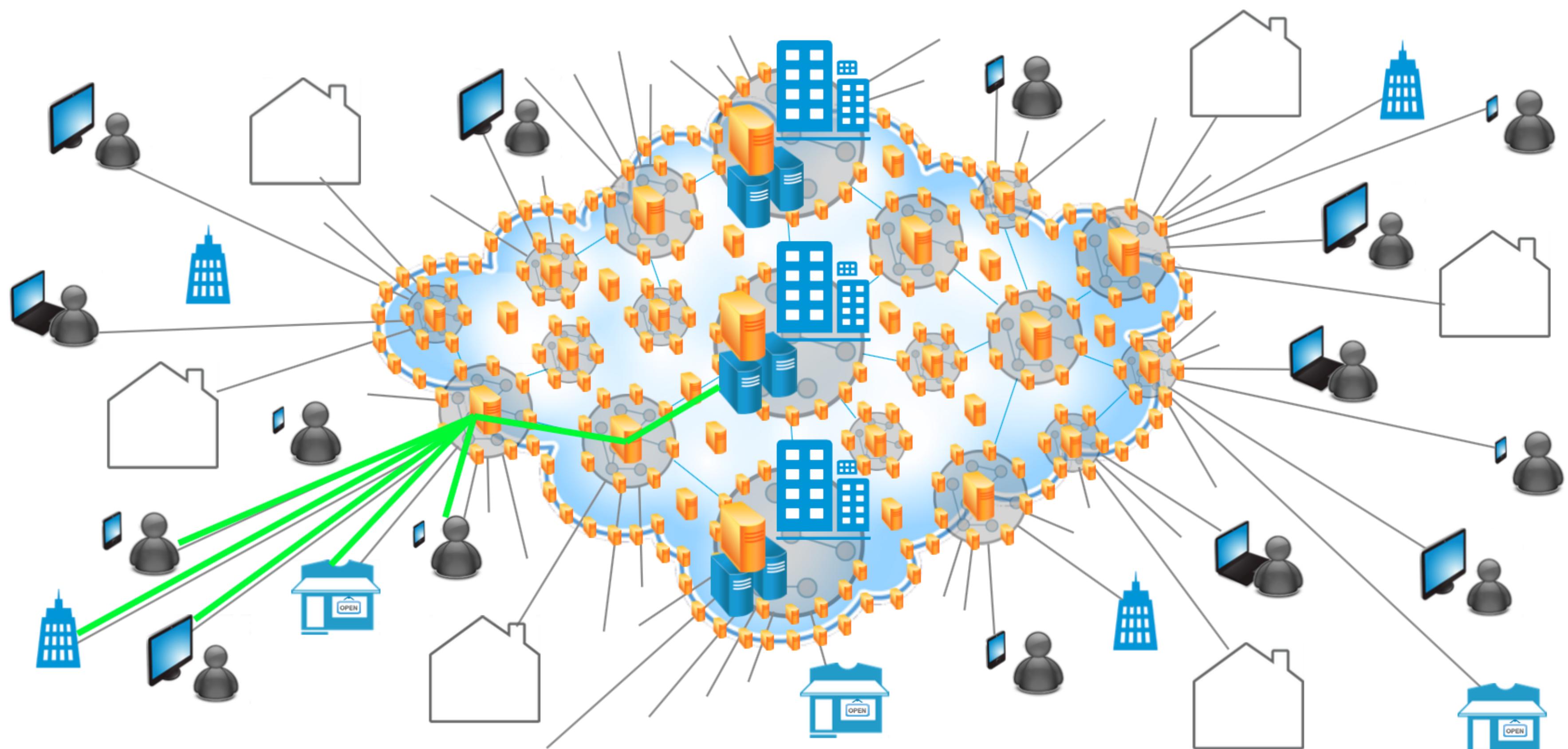


## Traditional Carrier CDN

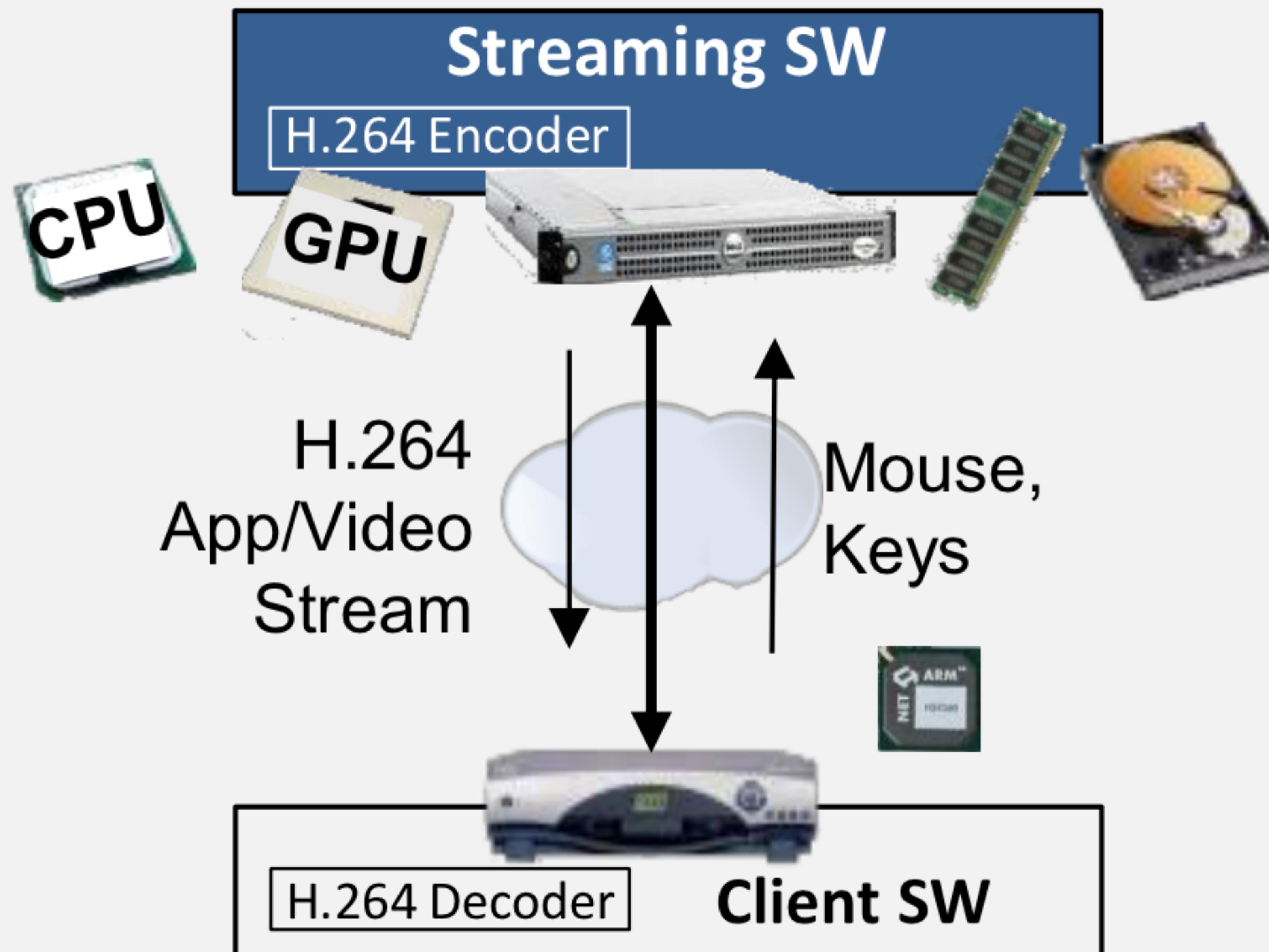
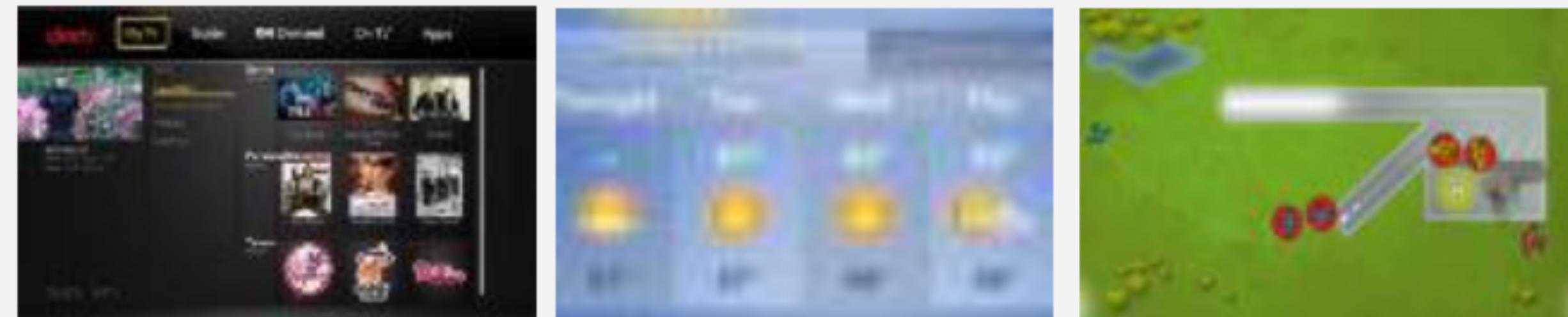


*Deployed in Top-Level Datacenters*

## Scale Out and Deep with NFV



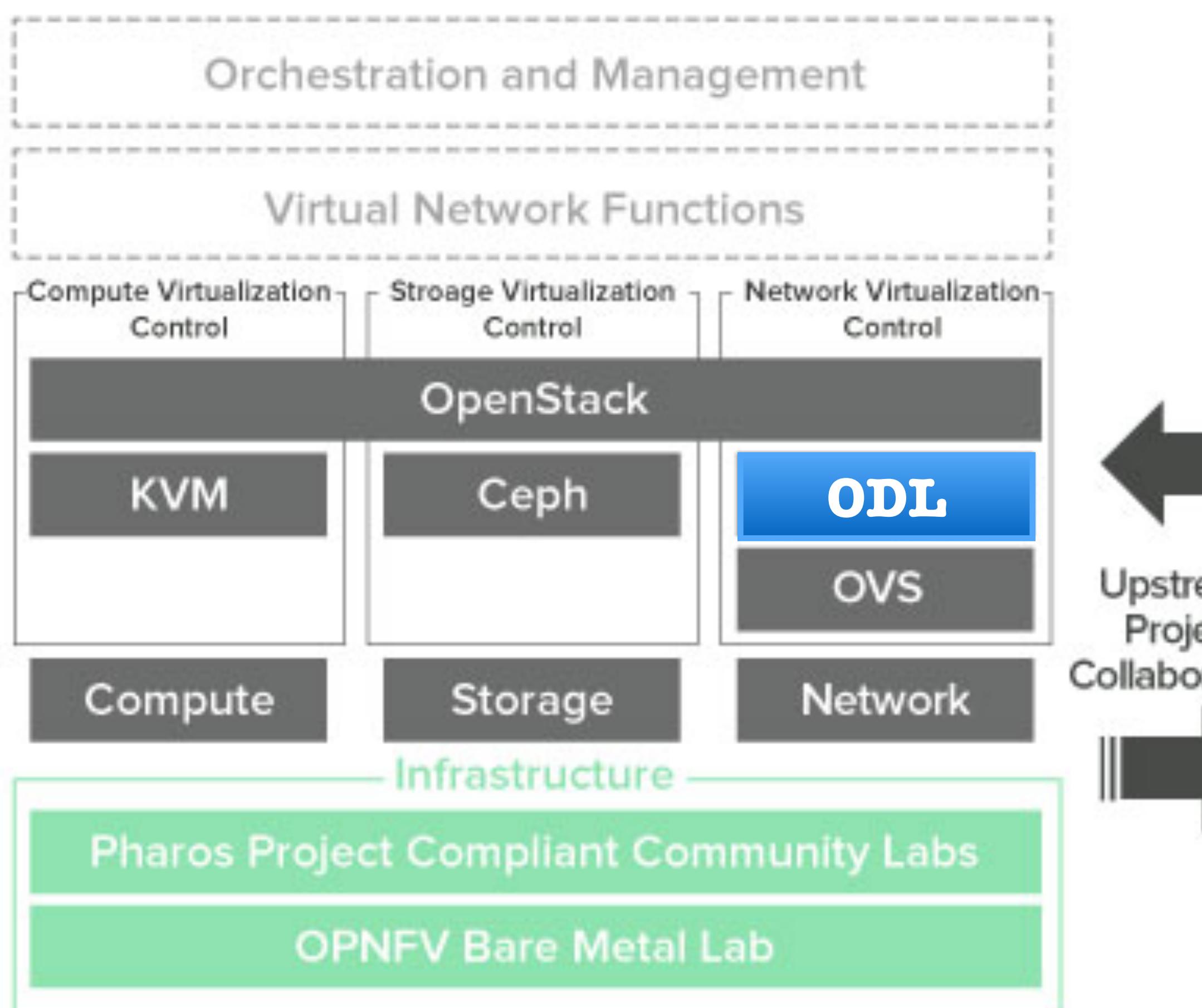
# Virtualized STB Apps



# Today's SDN/NFV

## TYPE #1: Telco Friendly Approach

### OPNFV Arno Overview



 **OPNFV**

Continuous Build and Integration  
Bootstrap / GetStarted

Continuous Deployment and Testing  
FuncTest

New Requirements and Features

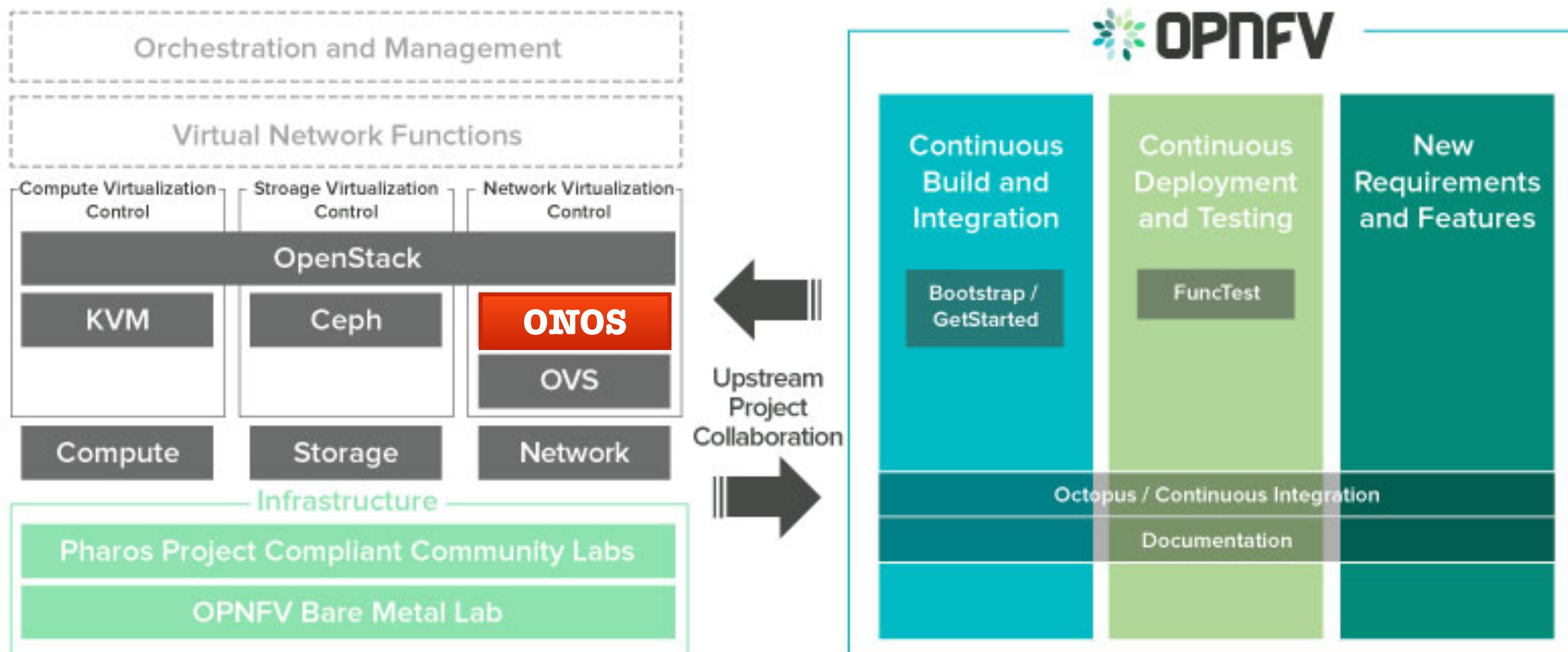
Octopus / Continuous Integration

Documentation

# Today's SDN/NFV

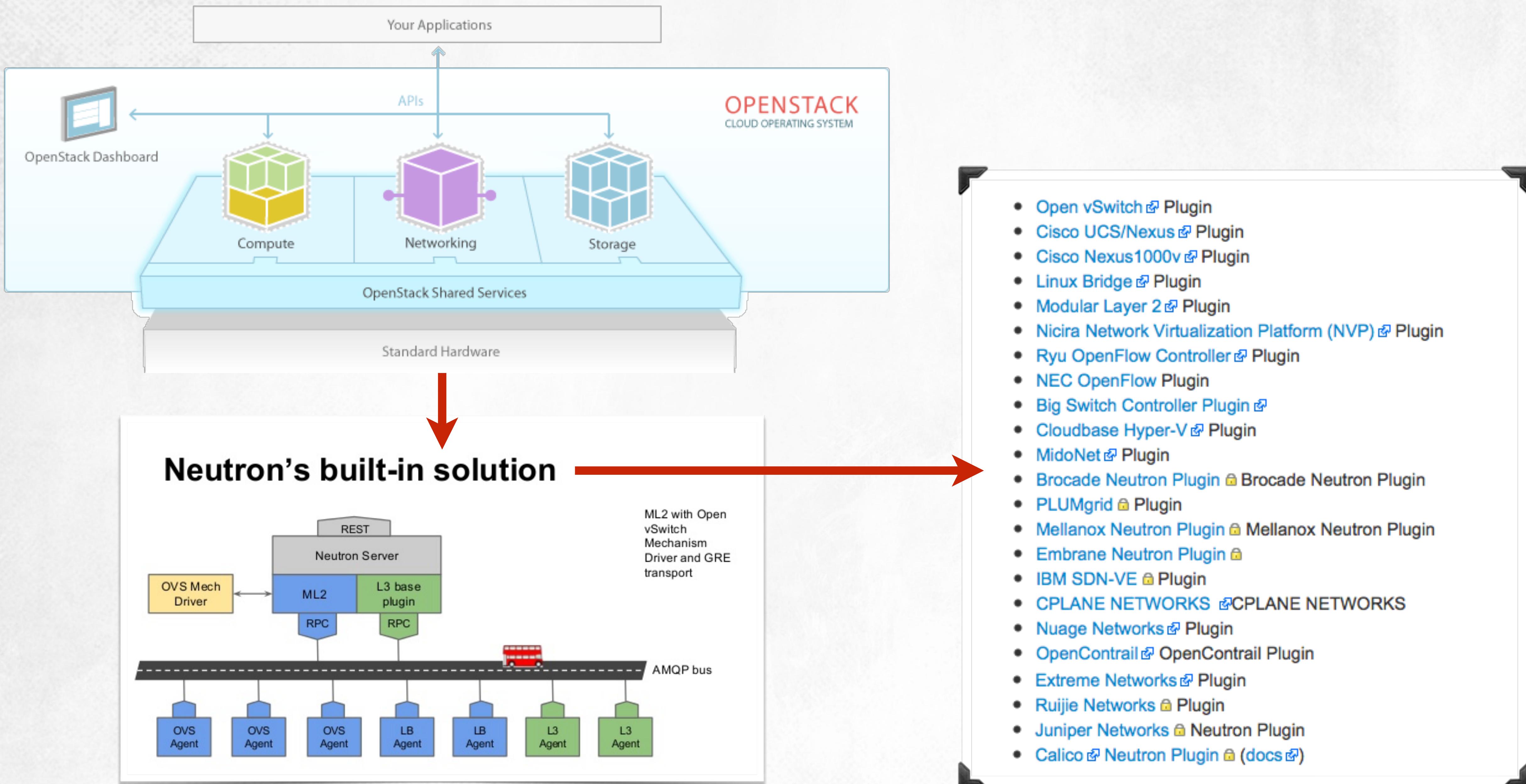
TYPE #2: Telco Friendly Approach (**ONOS** replacing[?] ODL)

## OPNFV Arno Overview



# Today's SDN/NFV

## TYPE #3: IT Friendly Approach (Network as a Service)



# Storage

## Open Source Storage, Software Defined Storage

**ZDNet Article Screenshot:** A screenshot of a ZDNet article titled "Open source carries software-defined storage forward". The headline is bolded. Below it, a snippet reads: "Software-defined storage is here to stay, but open source software will carry it forward." A "MUST READ" callout at the top says "WHY WON'T MICROSOFT PUBLISH WINDOWS 10 UPDATE RELEASE NOTES?".

**EMC ViPR Page Screenshot:** A screenshot of the EMC ViPR website. The main navigation bar includes links for PRODUCTS & SOLUTIONS, STORE, SERVICES, SUPPORT, COMMUNITIES, PARTNERS, and EMC+. The page features a large image of a Rubik's cube with storage-related icons.

**vVNX Download Page Screenshot:** A screenshot of the EMC vVNX download page. It features a large "vVNX DOWNLOAD" button and a "Data Sheet" button. The page also includes a "Take a test drive" section with a link to the EMC VNX Community.



# Contents

- Softwarization
- Software Defined Network(ing)
- Network Function Virtualization
- SDN/NFV
- SDN/NFV Softwares
- SDN/NFV Use Cases
- **SDN/NFV into 5G**
- P4 (programming language)

# ‘Telco SDN/NFV’ for 5G Mobile Network

“SDN/NFV-based METIS2020 Architecture”

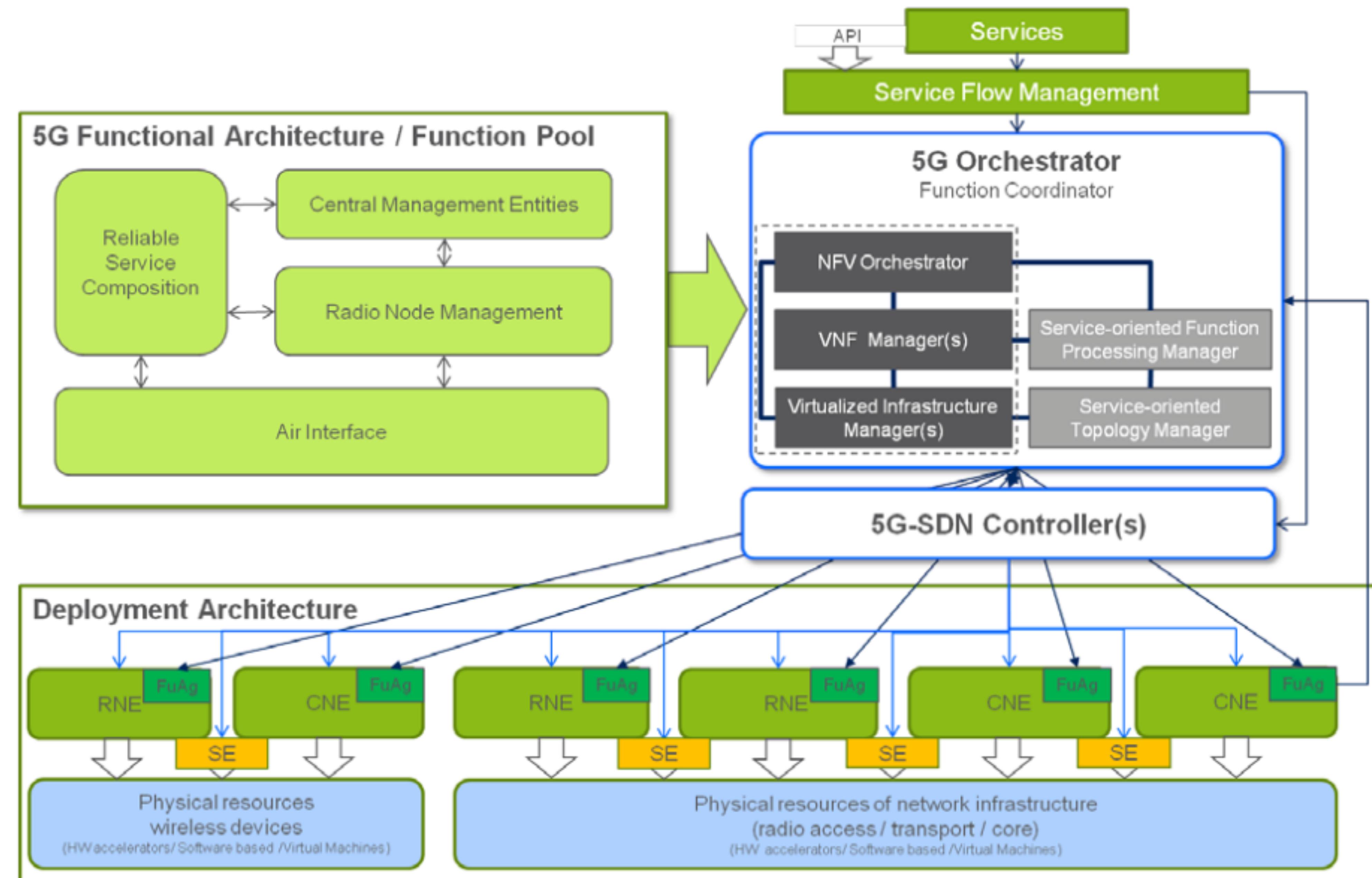


Figure 3-6: Orchestration & Control Architecture View of METIS 5G System.

# 'Telco SDN/NFV' for 5G Mobile Network



Computing in  
everywhere  
(단말 포함)

# ‘Telco SDN/NFV’ for 5G Mobile Network

“METIS2020 - Network & Device Communication”

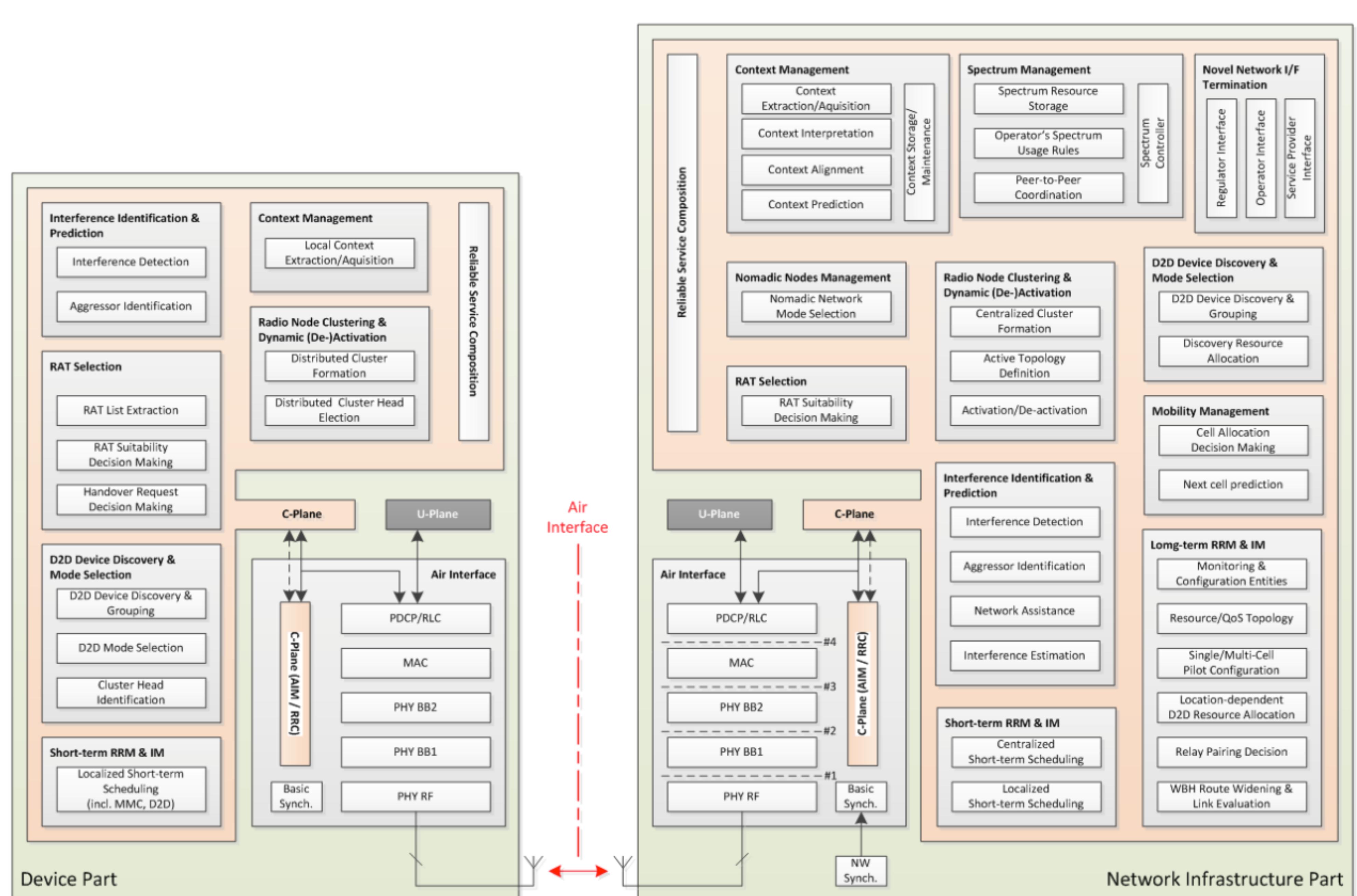


Figure 3-4: Generic functional architecture of METIS 5G System considering interrelations between infrastructure and device part (network-to-device communication).

# Design Principle of 5G Networking

“일반 사용자에서 새로운 사용자(사업)를 받아들이는 그릇”

Ericsson teams with MIT students on autonomous car innovation

2016-03-23 Categories: Recognition

- Collaboration with the MIT students to push boundaries
- “Identity module” to be job preferences and behavior
- Establishes innovation-based

Ericsson puts the TV consumer at center of agenda for NAB show 2016

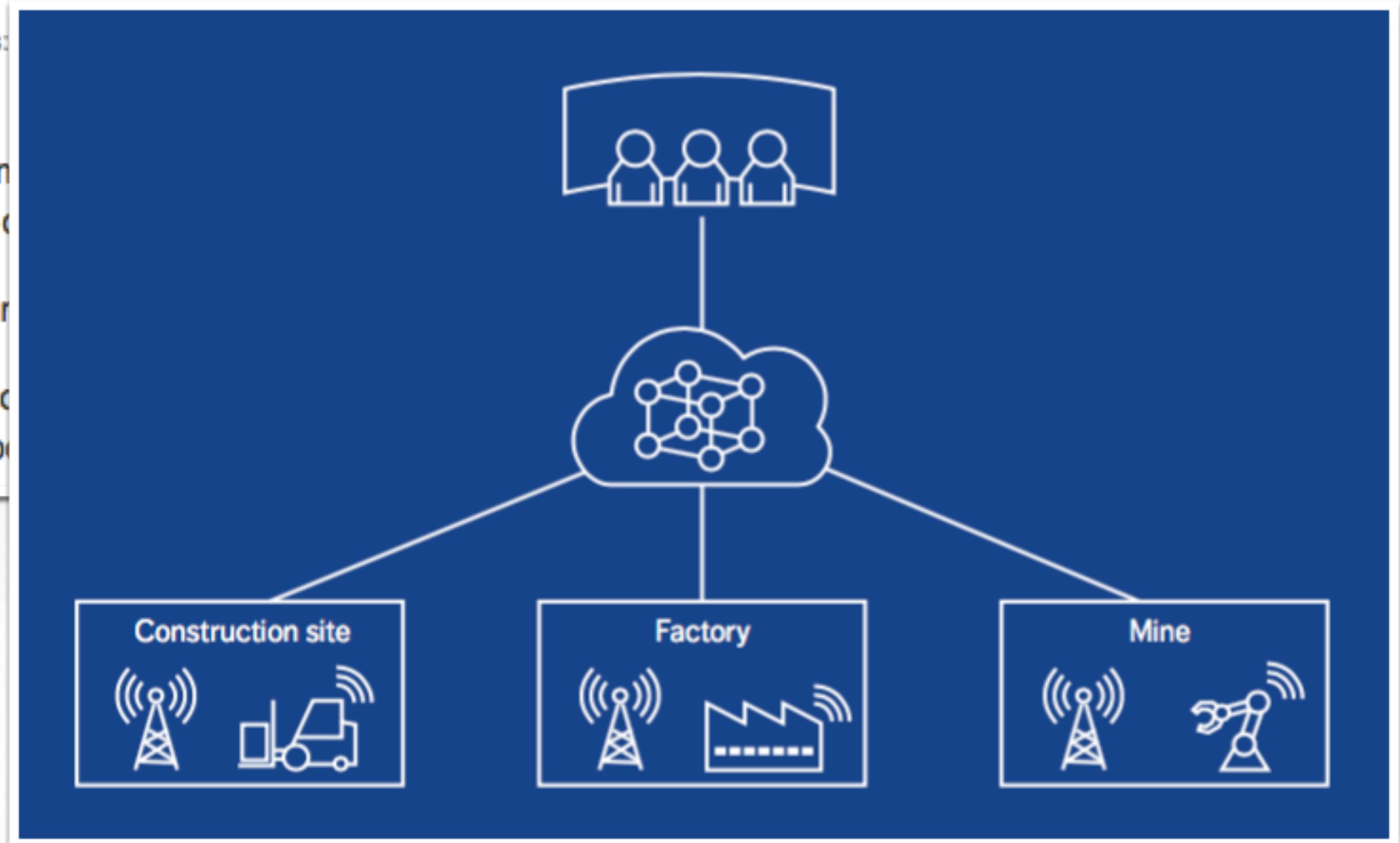
2016-03-31 Categories: Industry

- Presents the ultimate customers to continue
- New product and services demonstrate Ericsson
- Highlights how Ericsson the transformation era of television

Ericsson and Telefónica demonstrate live delivery of LTE-U

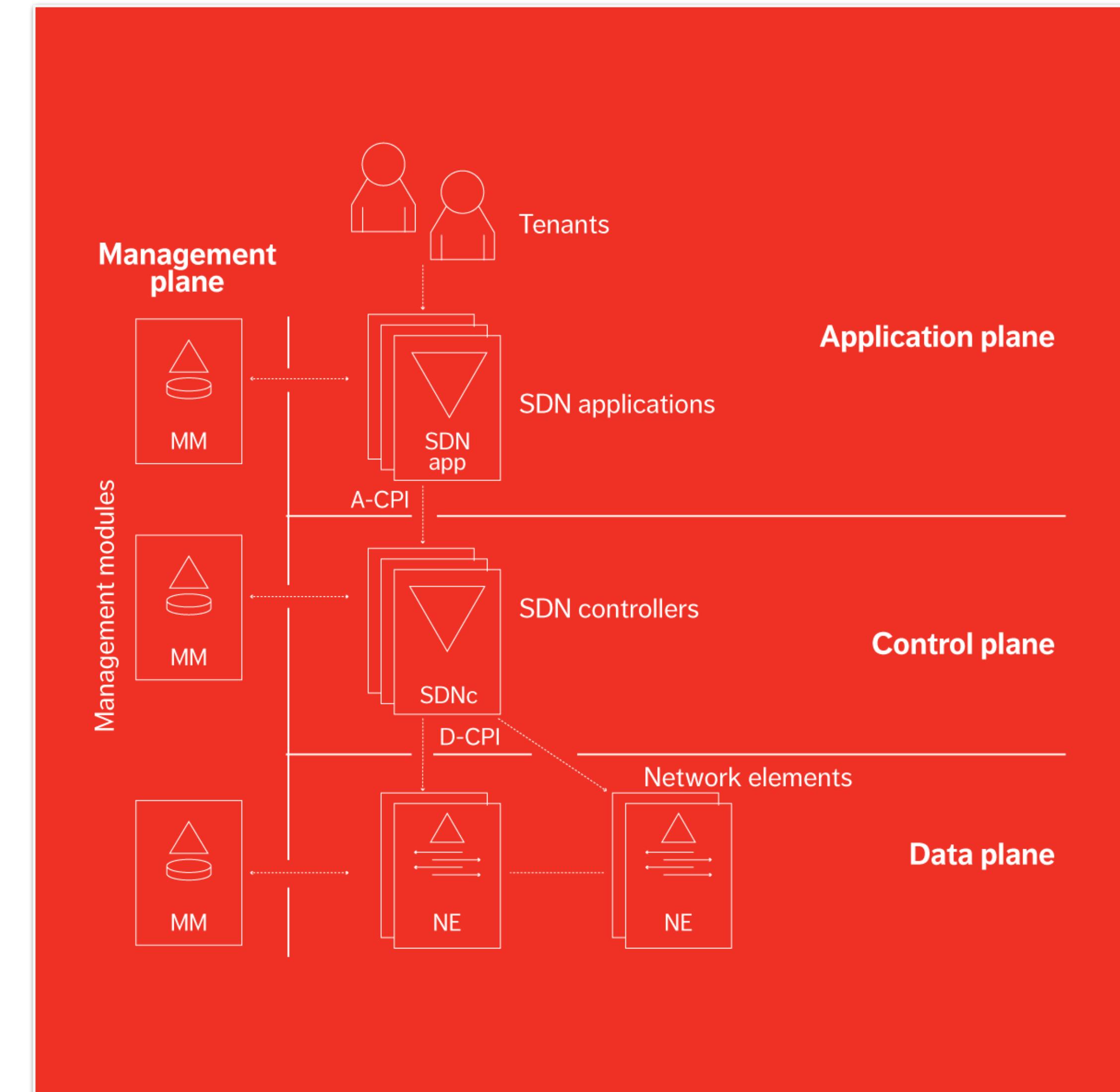
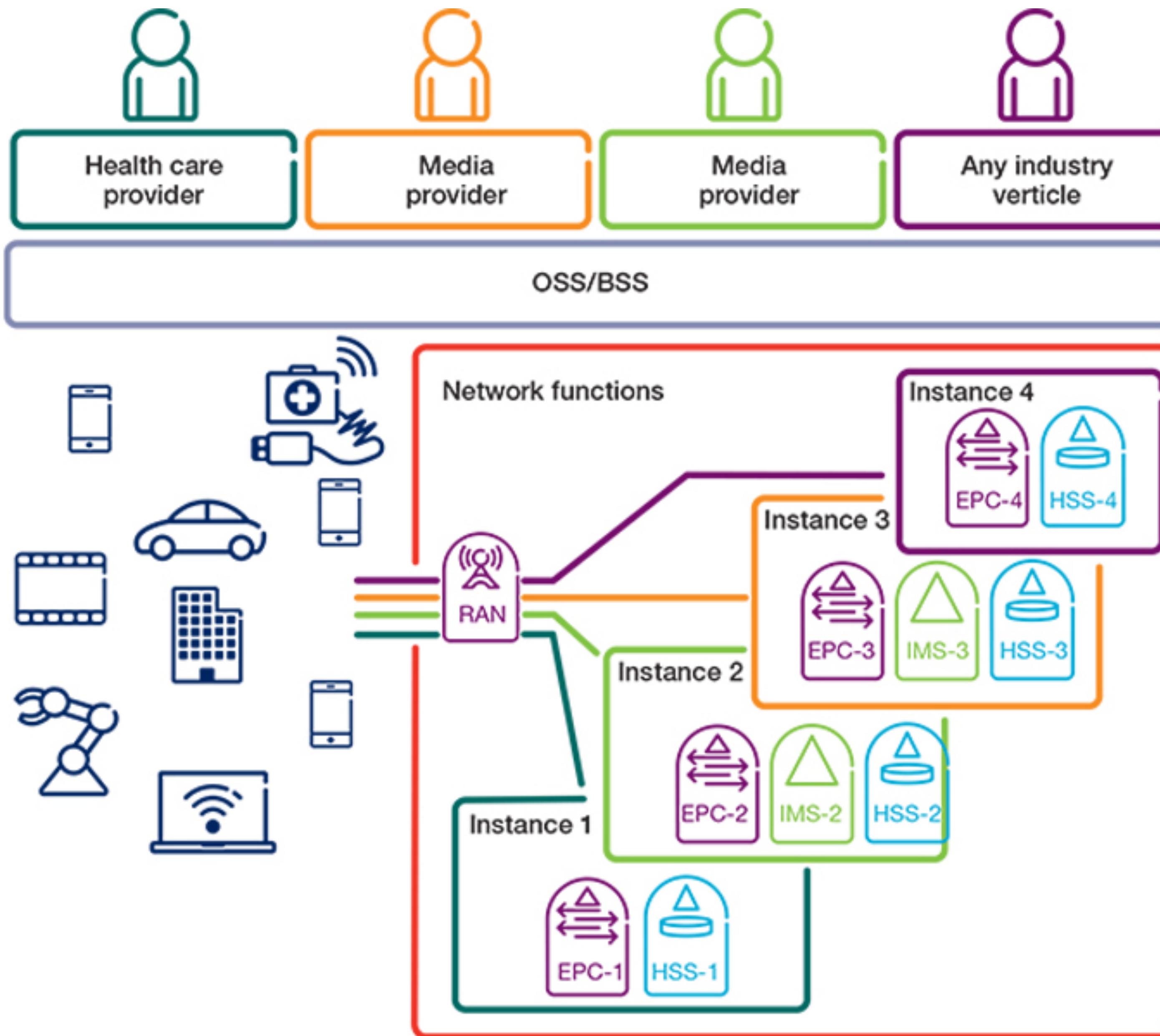
2016-03-30 Categories:

- Over-the-air deployment of RBS 6402 pico cells
- Ericsson announces
- LTE-U enables commercial licensed LTE spectrum



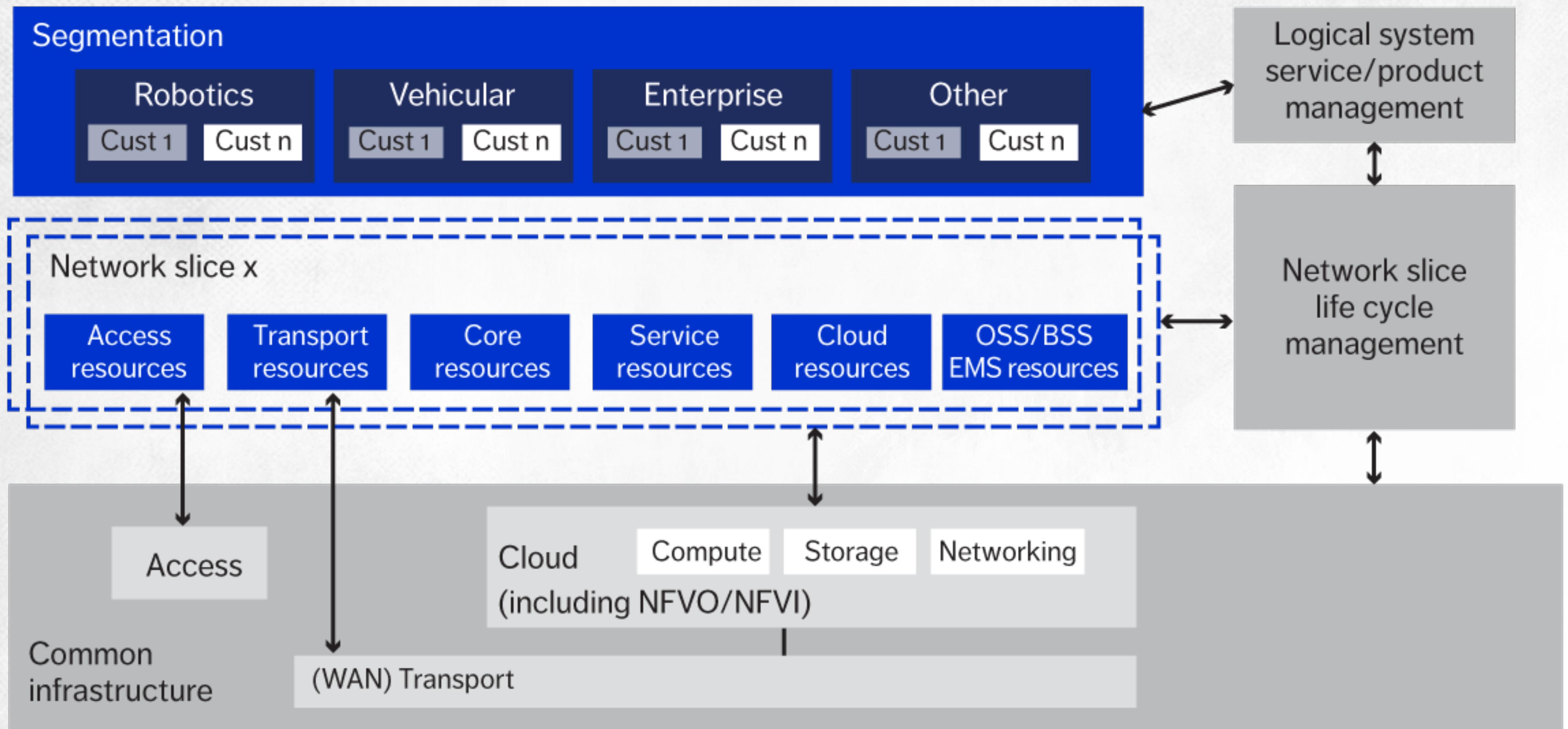
# Design Principle of 5G Networking

“일반 사용자에서 새로운 사용자(사업)를 받아들이는 그릇”



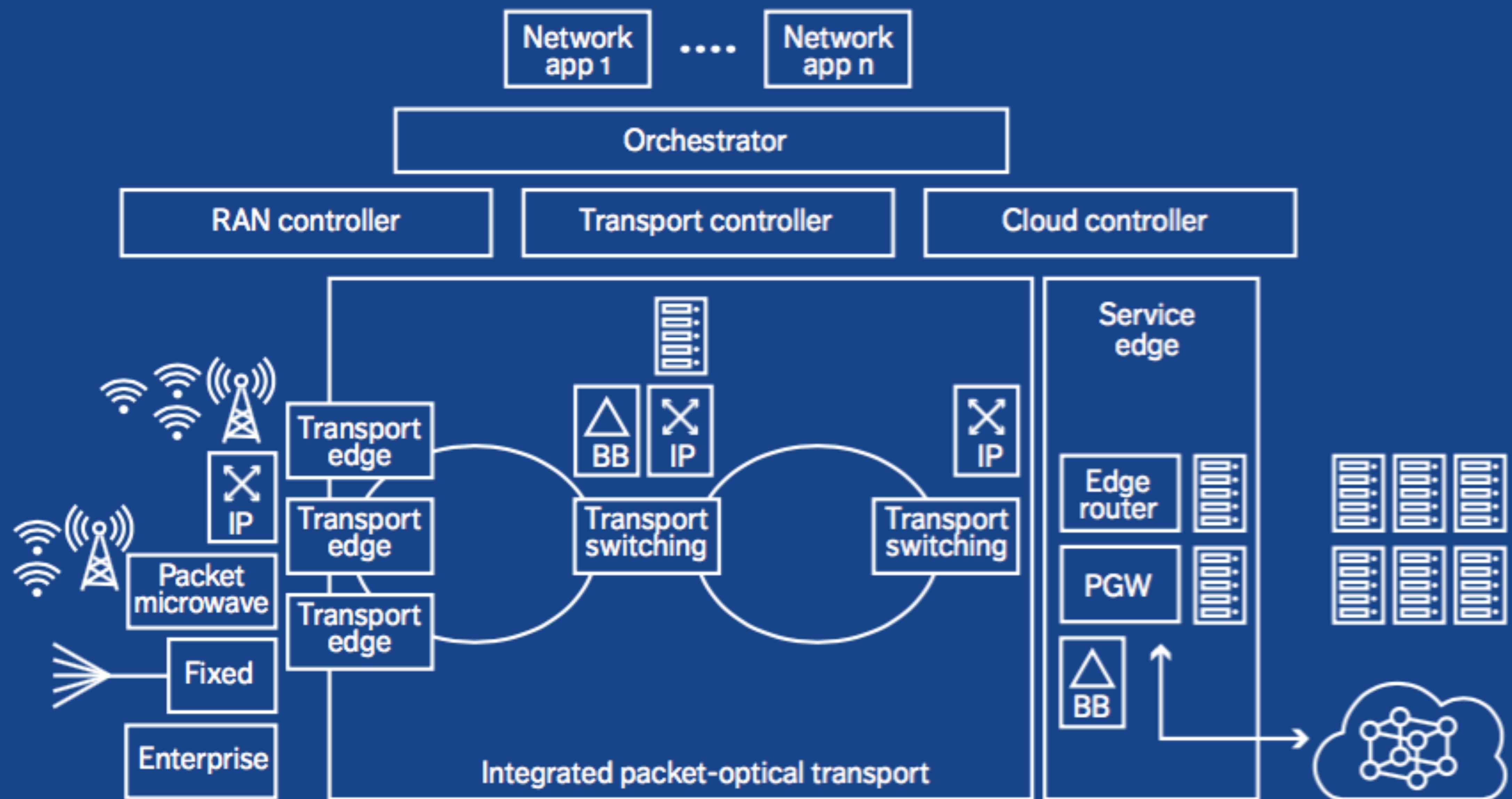
# Design Principle of 5G Networking

“일반 사용자에서 새로운 사용자(사업)를 받아들이는 그릇”



# Design Principle of 5G Networking

“일반 사용자에서 새로운 사용자(사업)를 받아들이는 그릇”



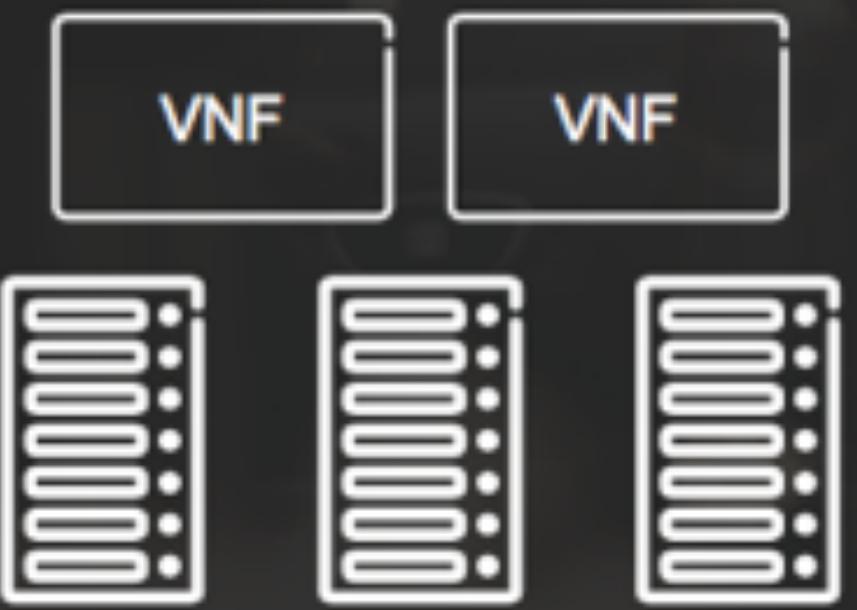
# Design Principle of 5G Networking

## 5G CORE NETWORK EVOLUTION

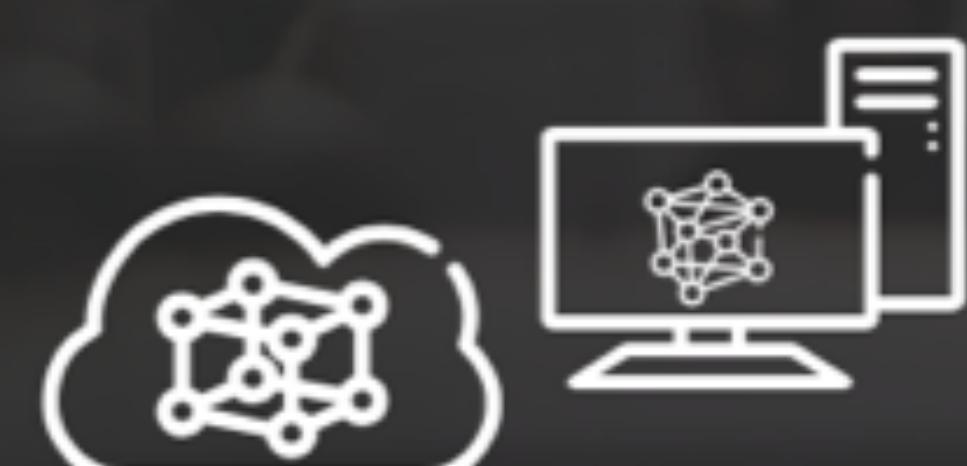
Management & Orchestration, Analytics & Exposure



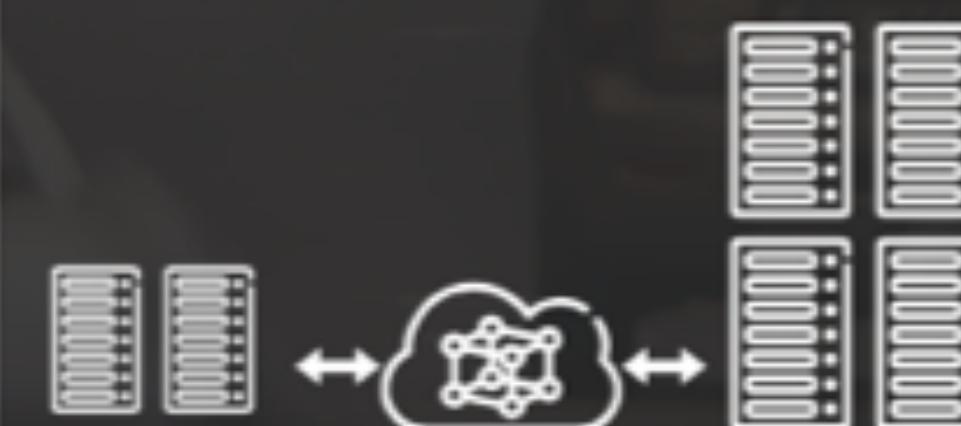
Virtualization



Software Defined  
Networking (SDN)



Distributed Cloud



Network Slicing



# Contents

- Softwarization
- Software Defined Network(ing)
- Network Function Virtualization
- SDN/NFV
- SDN/NFV Softwares
- SDN/NFV Use Cases
- SDN/NFV into 5G
- P4 (programming language)

# P4 (programming language)

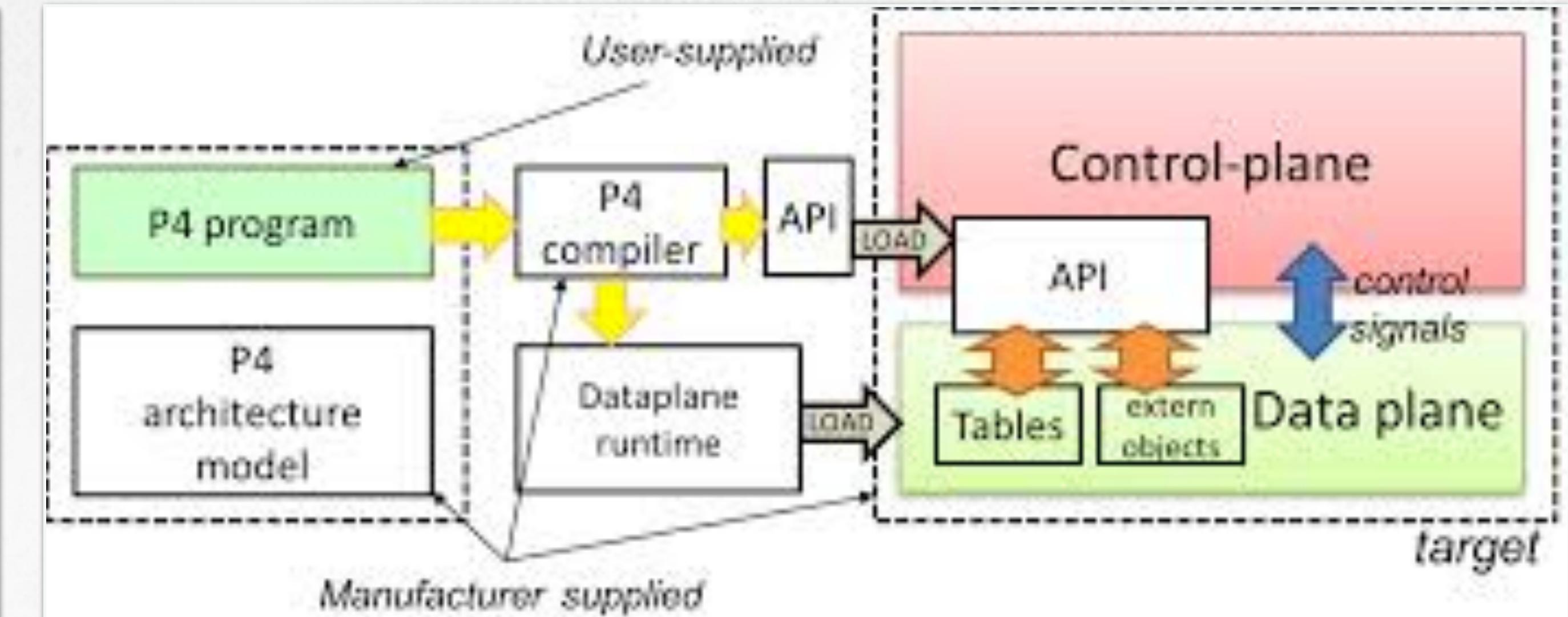
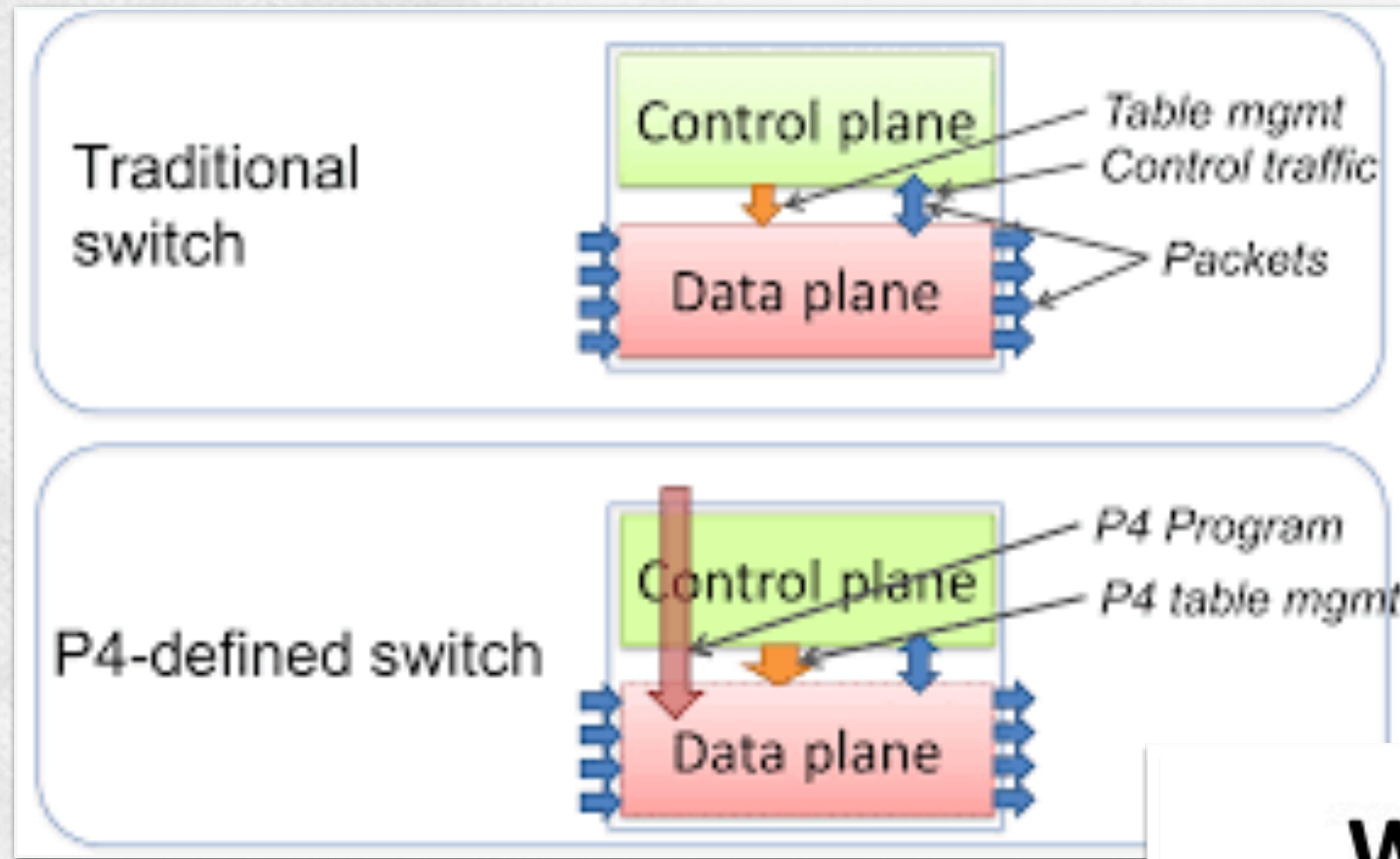
## Concept

---

- P4 is a programming language designed to allow programming of packet forwarding planes. In contrast to a general purpose language such as C or Python, P4 is a domain-specific language with a number of constructs optimized around network data forwarding. P4 is an open-source, permissively licensed language and is maintained by a non-profit organization called the P4 Language Consortium. The language was originally described in a SIGCOMM CCR paper in 2014 titled “Programming Protocol-Independent Packet Processors” – the alliterative name shortens to “P4”.

# P4 (programming language)

## Implementation



### What does a P4 program look like?

```
/* Example: A typical IPv4 routing table */





```

vrf	ipv4.dstAddr / prefix	action	data
1	192.168.1.0 / 24	l3_l2_switch	port_id=64
10	10.0.16.0 / 22	l3_ecmp	ecmp_index=12
1	192.168.0.0 / 16	l3_nexthop	nexthop_index=451
1	0.0.0.0 / 0	l3_nexthop	nexthop_index=1

14



# Thank you