Computer Network

Lecture 7

Network Layer Part.2

2019. 03. 01

Sungwon Lee

Department of Software Convergence

경희대학교
KYUNG HEE UNIVERSITY

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)

Revisited

Software is eating up the world*

August 20, 2011

* Marc Andreessen
in Wall Sreet Journal

5

Ref: https://www.wsj.com/articles/SB10001424053111903480904576512250915629460

Software is eating the world, in all sectors

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

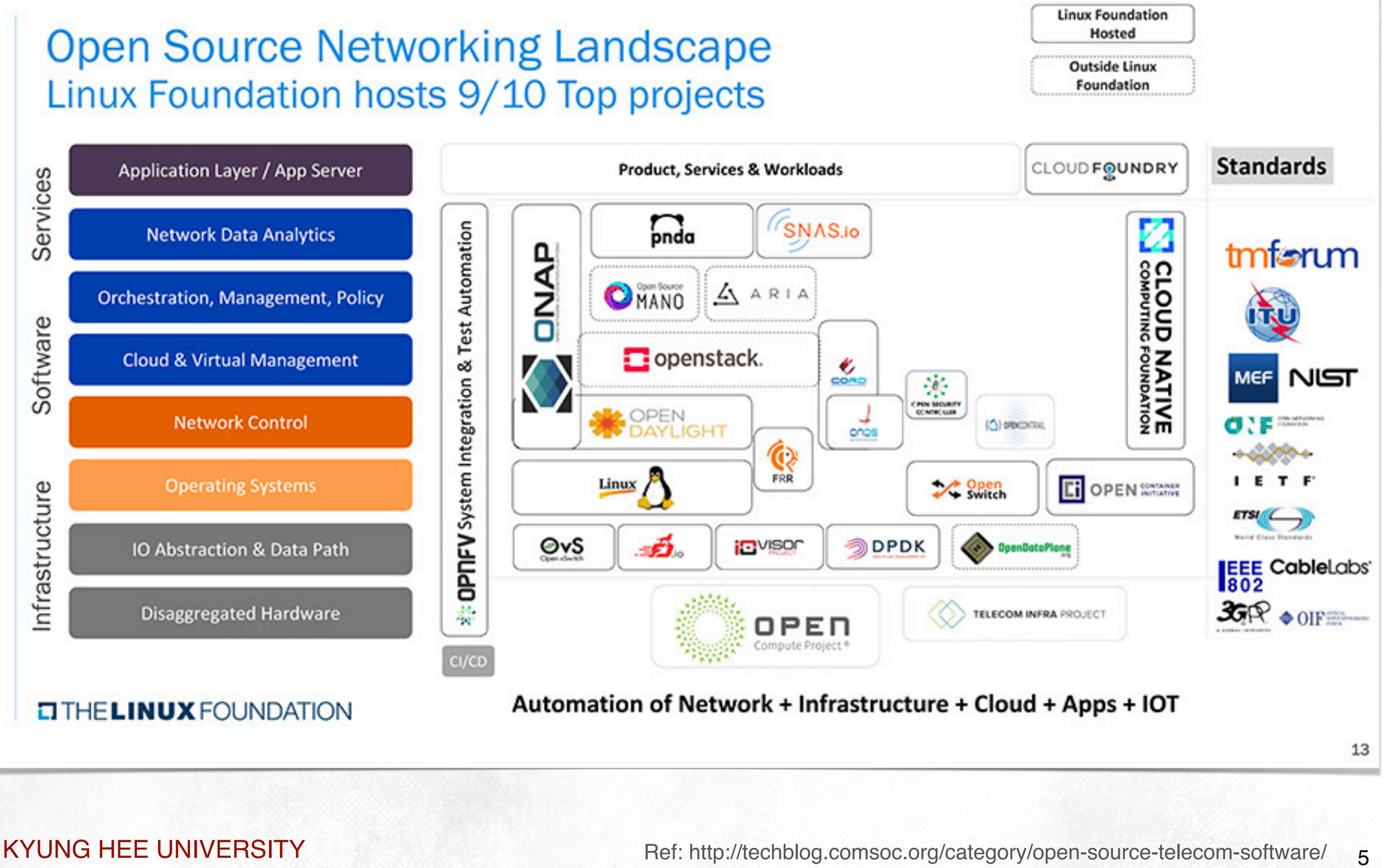
Mark Andreessen
founder of Netscape,
renowned Venture Capitalist
Andreessen-Horowitz

Ref: https://www.bigfootcap.com/a-review-of-why-software-is-eating-the-world-how-have-the-companies-fared/

Revisited

Linux Eating the Networking Industry

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



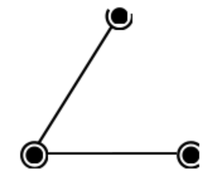
Revisited

Linux Eating the Networking Industry

AKRAINO EDGE STACK
CORD
DANOS
DPDK
.io
FRR
VISOR PROJECT
ONAP
ONOS Open Network Operating System
OPEN DAYLIGHT
OPEN SECURITY CONTROLLER
Open Switch
OPNFV
OvS Open vSwitch
pnda
SNAS.io
tungstenfabric
CLOUDFOUNDRY
CLOUD NATIVE COMPUTING FOUNDATION
OPEN CONTAINER INITIATIVE
Xen Project

Reference: https://en.wikipedia.org/wiki/Linux_Foundation

Open Source Networking Software for Legacy Routing Protocols

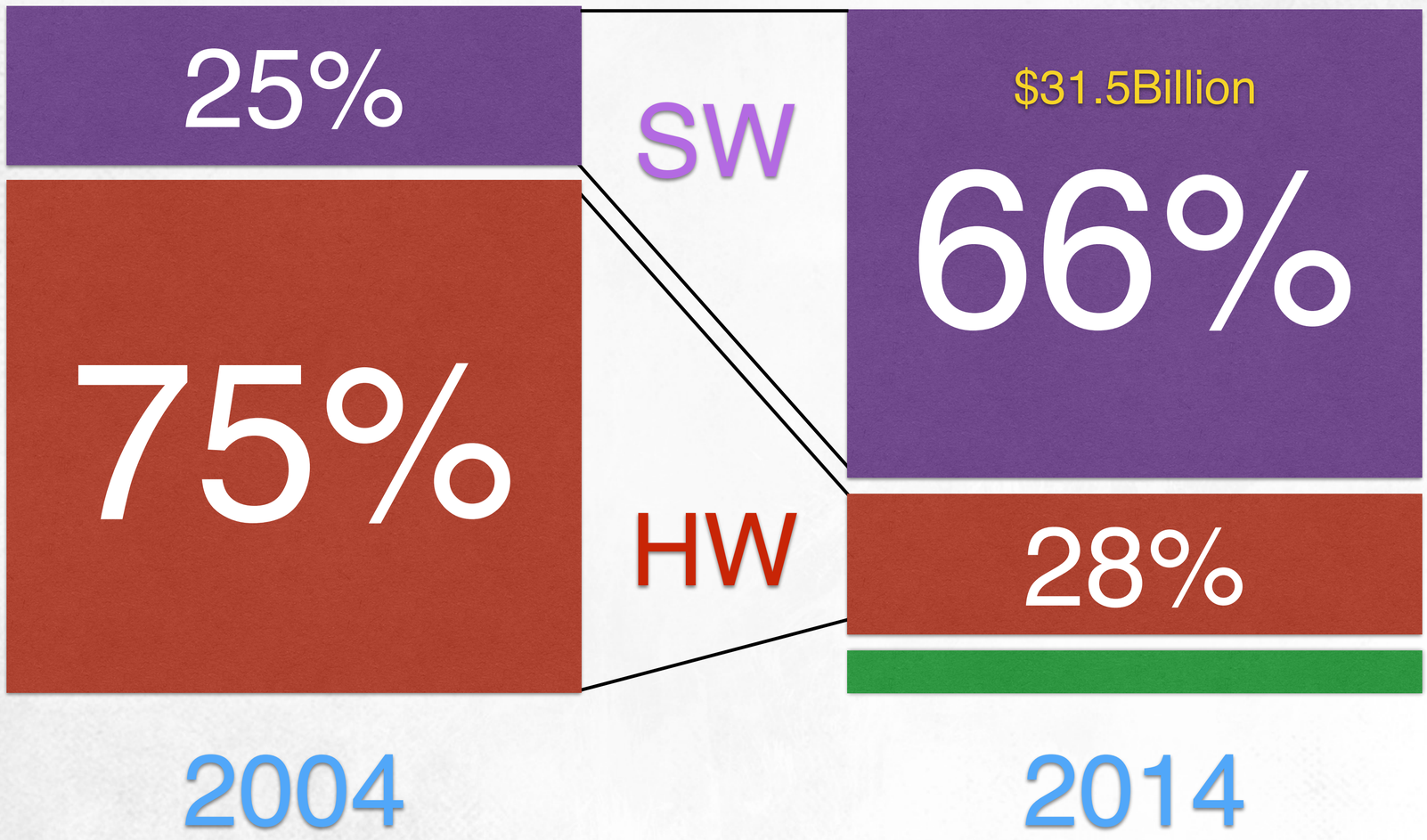


Babel
BIRD
OpenBGPD
OpenOSPFD
Quagga
Free Range Routing
XORP
Zebra
Optimized Link State Routing Protocol

Ref: https://en.wikipedia.org/wiki/List_of_open-source_routing_platforms

Ericsson's Annual Sales

"'Hardware' to 'Software and Service' within a decade"



참조: http://fortune.com/2014/09/25/connected-ericsson-ceo-hans-vestberg/

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."

참조: http://fortune.com/2014/09/25/connected-ericsson-ceo-hans-vestberg/

Purpose-Oriented Common Platform

"Three is Enough: Radio, Bearer, and Control"

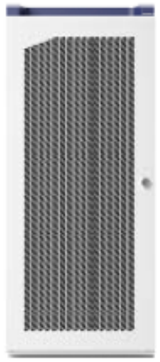
Purpose-Built for Performance

Ericsson RBS 6000

GSM

WCDMA/HSPA

LTE



IP Transport

(Radio Access)

Ericsson Blade Server

MSS / IMS

UDM

EVO Controller

SGSN-MME

Policy Control

(Control)

SSR 8000 Family

Service Awareness

Packet Gateway

Border Gateway

Caching / CDN

L2/L3 PE Router

3rd party app

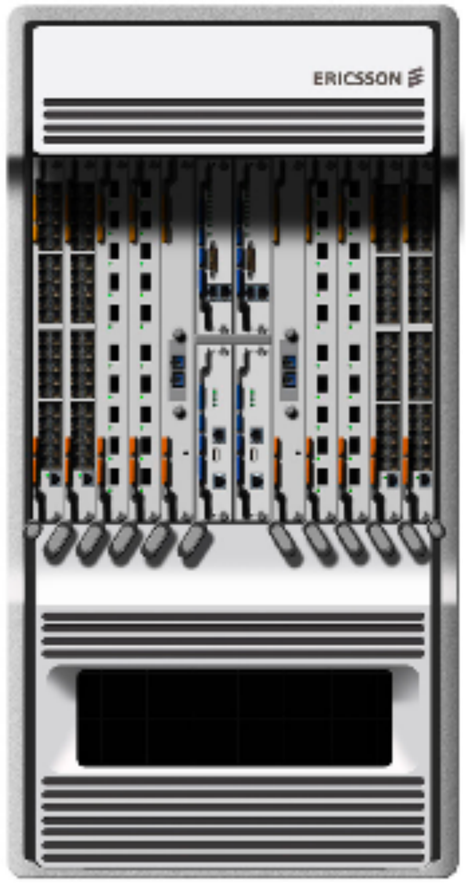
(Data and Networking)

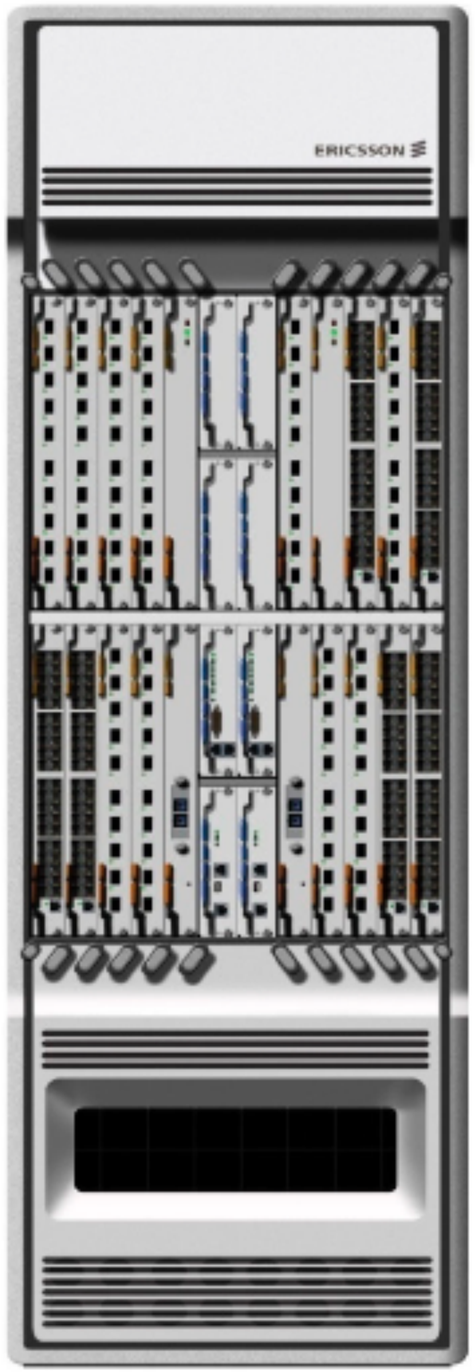
- User experience is dependent on network performance
- Ericsson systems deliver superior performance in mobile broadband
- Application separate from hardware in IP architectures
- True multi-standard, multi-application platforms deliver SDN benefits

Purpose-Oriented Common Platform

"Three is Enough: Radio, Bearer, and Control"

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





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

SSR 8000 Family

Service Awareness
Packet Gateway
Border Gateway

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

(Data and Networking)

ance
nobile broadband
ures
iver SDN benefits

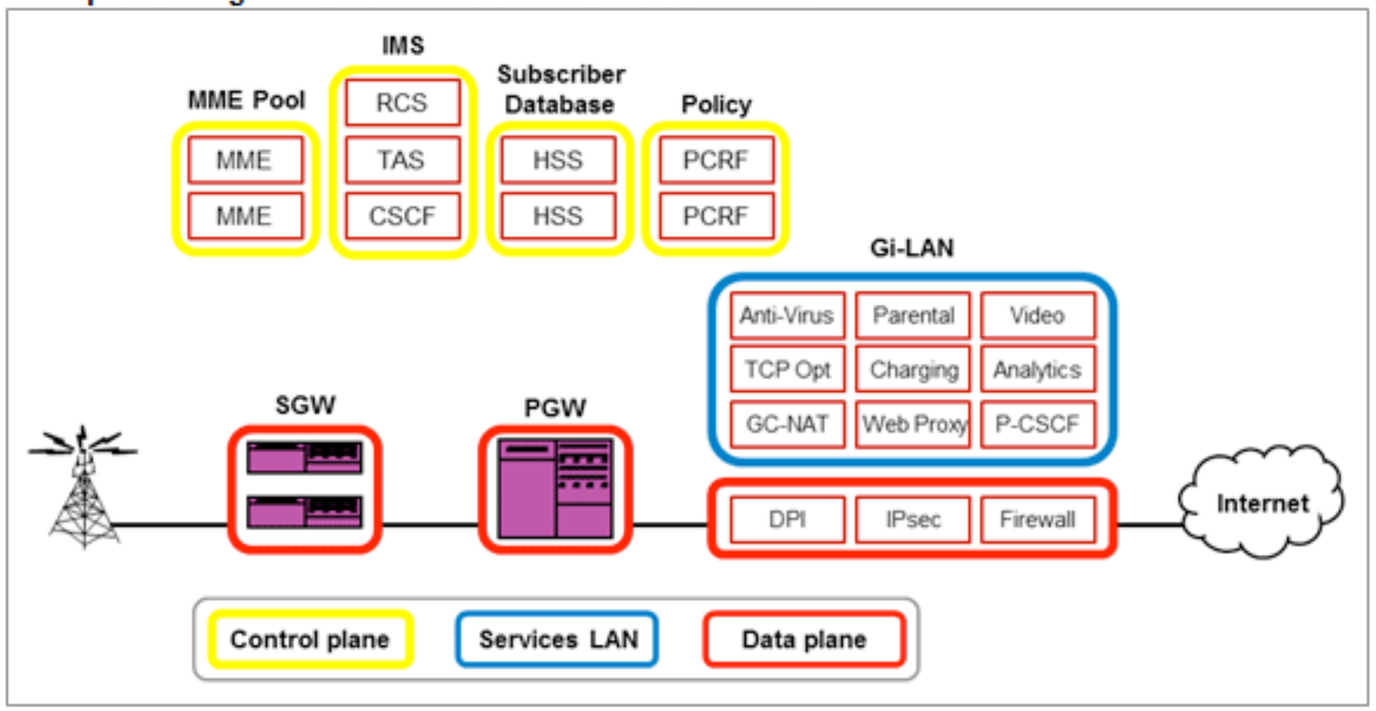
참조: http://www.ericsson.com/res/investors/docs/2012/ericsson-cloud-and-sdn.pdf
http://www.ericsson.com/res/region_RLAM/press-release/2011-09-13-family-smart-servers.pdf

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

참조: http://www.ericsson.com/res/thecompany/docs/press/backgrounders/cloud-press-backgrounder.pdf
http://www.heavyreading.com/details.asp?sku_id=3187&skuitem_itemid=1558

Big change in Current Network

"Softwarization & Virtualization"

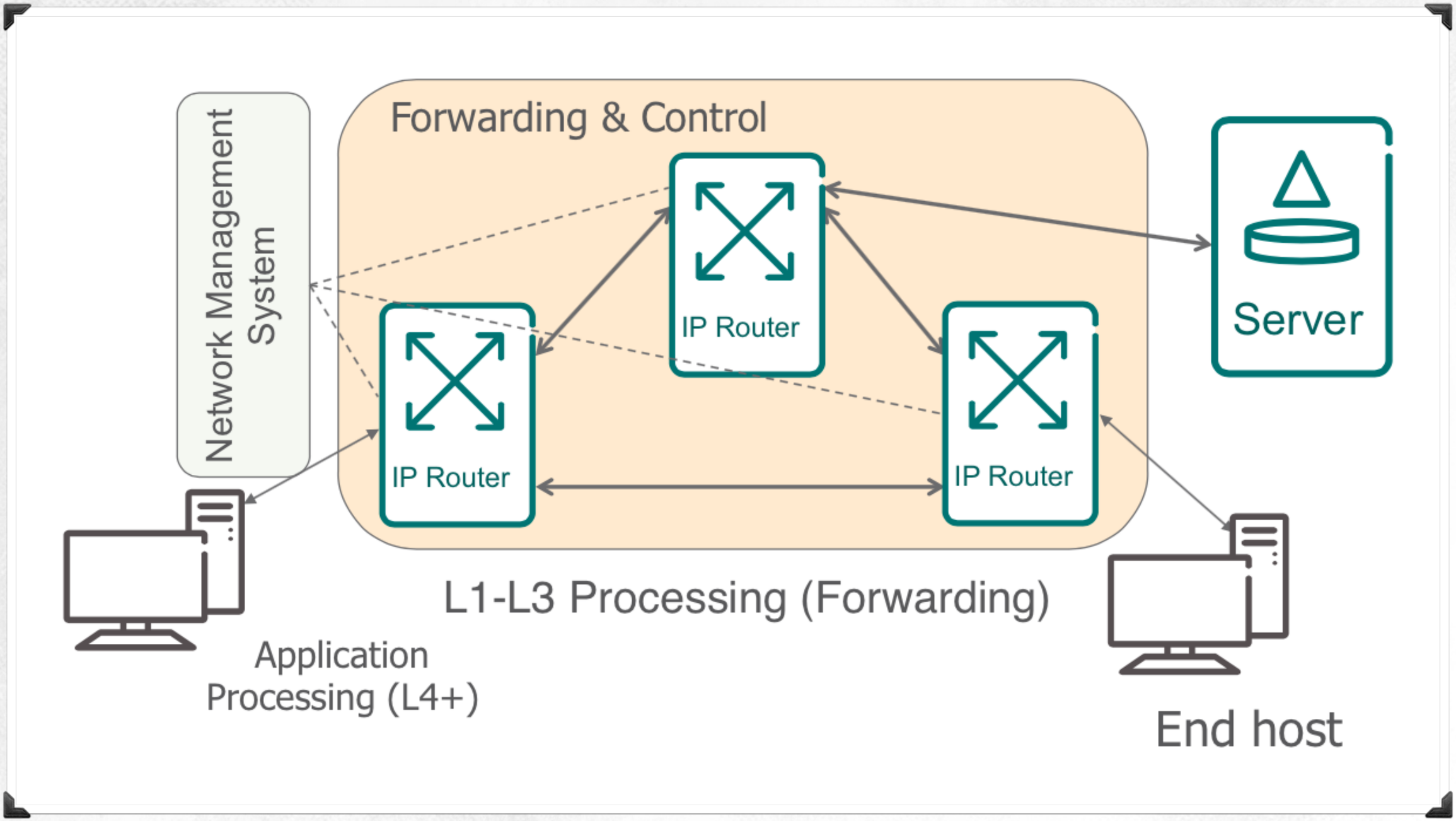
Past	**Today**
NE Oriented (eNB, SGW, PGW, MME)	Function Oriented (eNB & VNFs)
Standard (Only)	Software APIs (Major)
Uncontrollable Network	Controllable Network

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

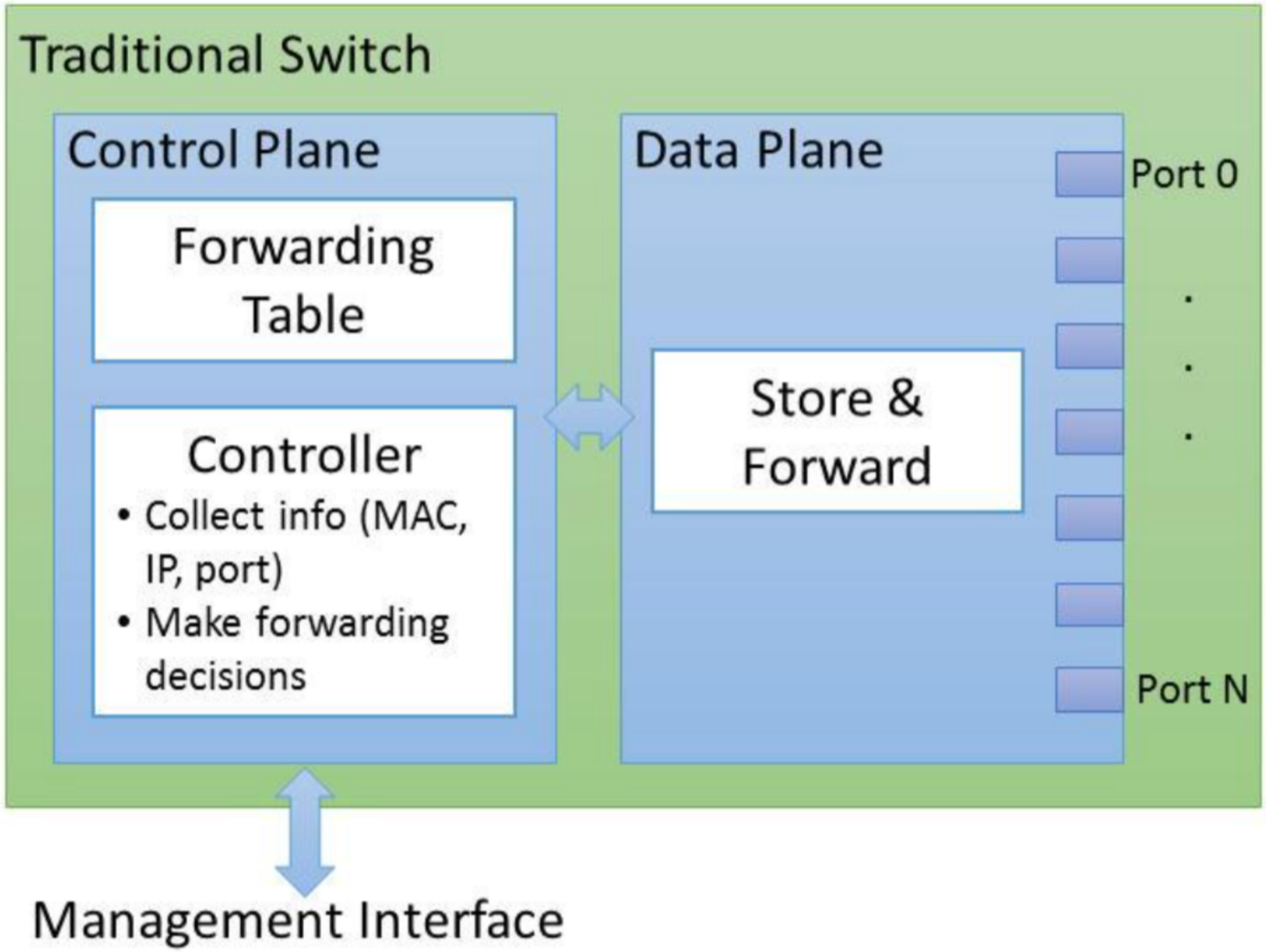
"네트워크 생존성이 목적이였던 {분산} 라우터 기반 네트워크"



출처: Ahmad Rostami, "The Evolution of Programmable Networks"

ROUTER

"제어 기능과 데이타 전송 기능의 통합 장치"



출처: Mauricio Tsugawa, "Tutorial on Software Defined Networking"

Problems on Router

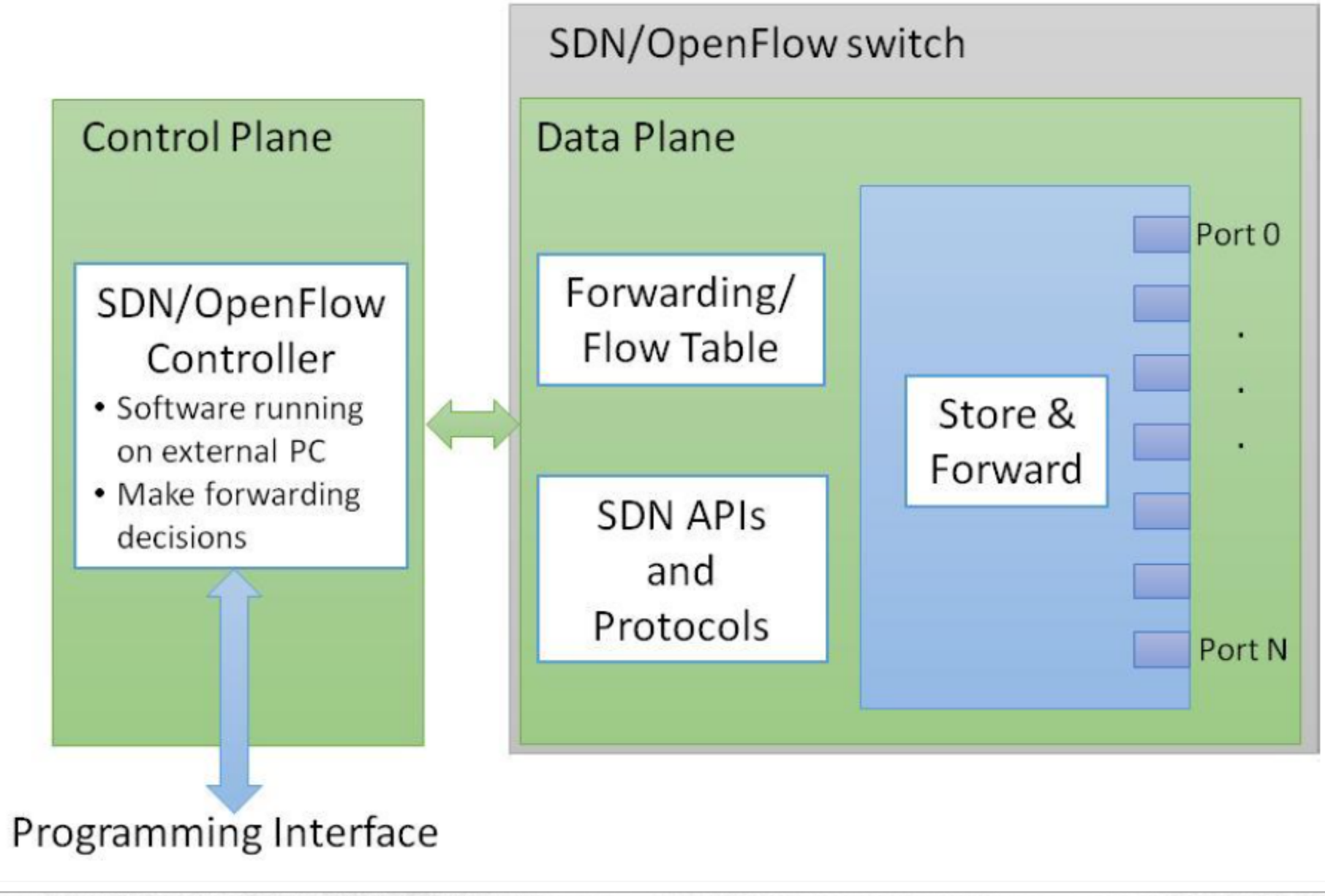
"네트워크는 단순하다. 라우터끼리 라우팅을 하니깐… That's All?"

- 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

출처: http://www.yunphoto.net/ko/photobase/yp9985.html

Why SDN?

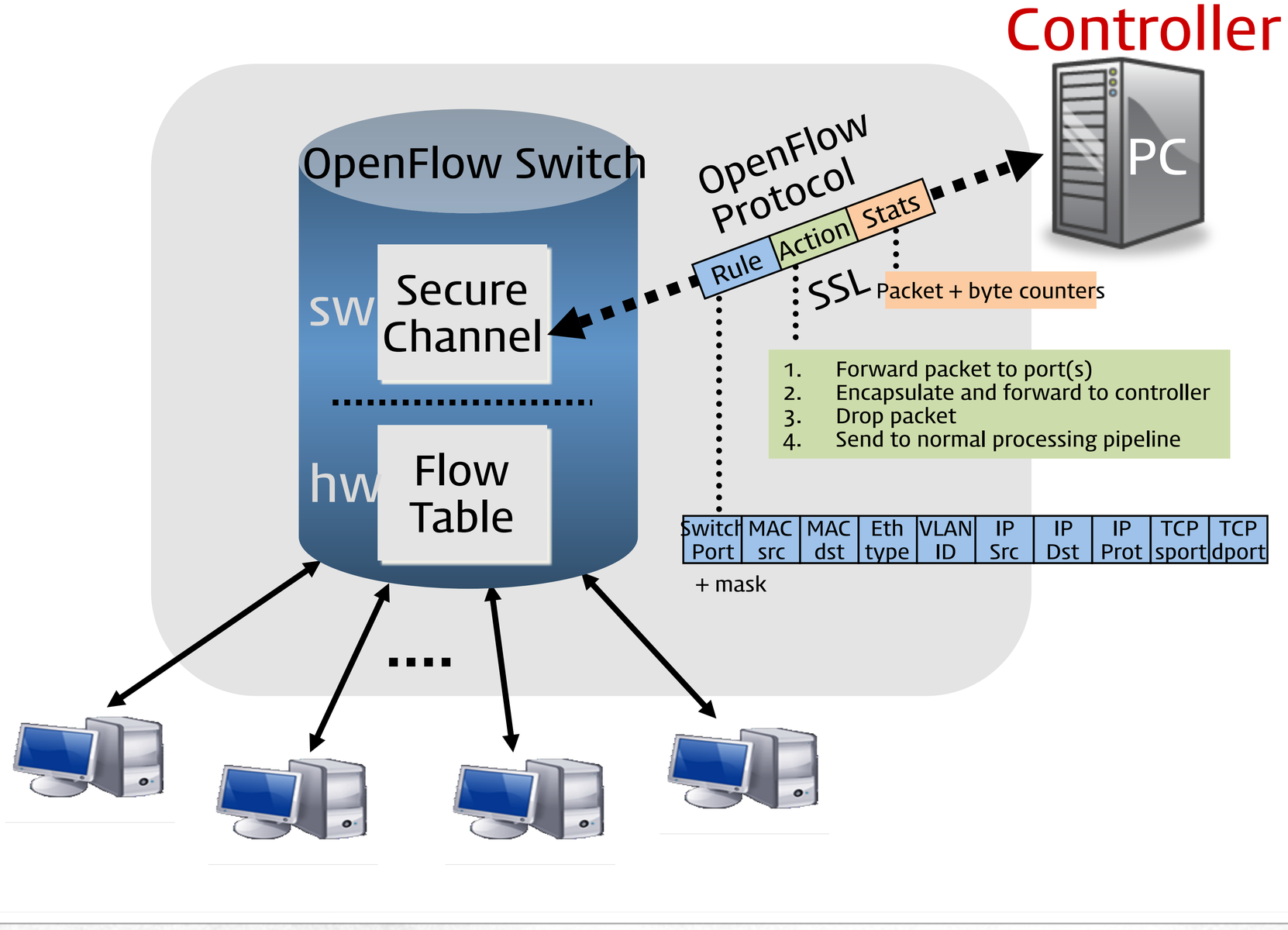
"제어와 전송의 분리 (Control & Bearer Separation)"



출처: Mauricio Tsugawa, "Tutorial on Software Defined Networking"

OpenFlow based SDN Network

"중앙 집중형 SDN 컨트롤러와 OpenFlow 프로토콜"



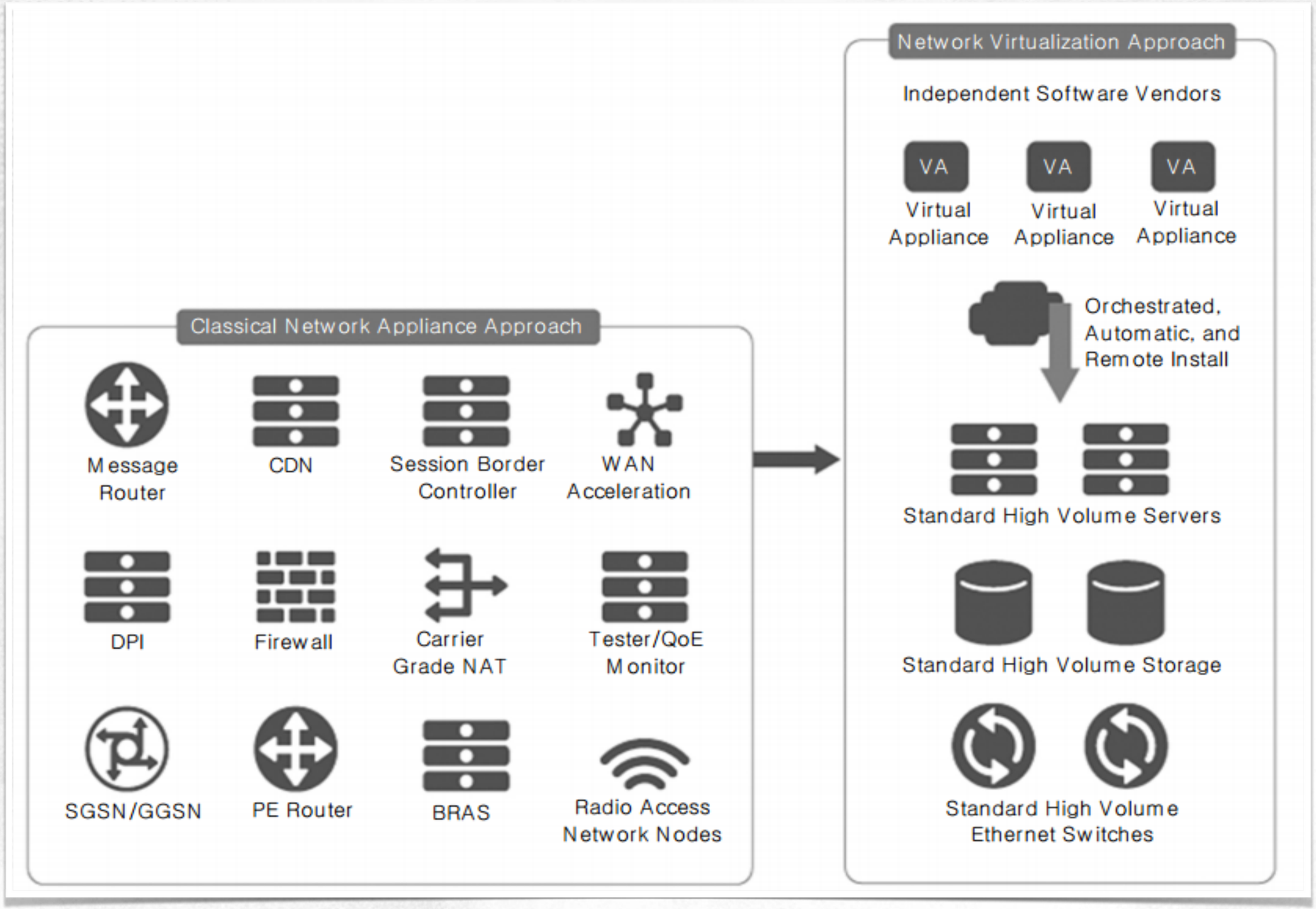
출처: Mauricio Tsugawa, "Tutorial on Software Defined Networking"

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)

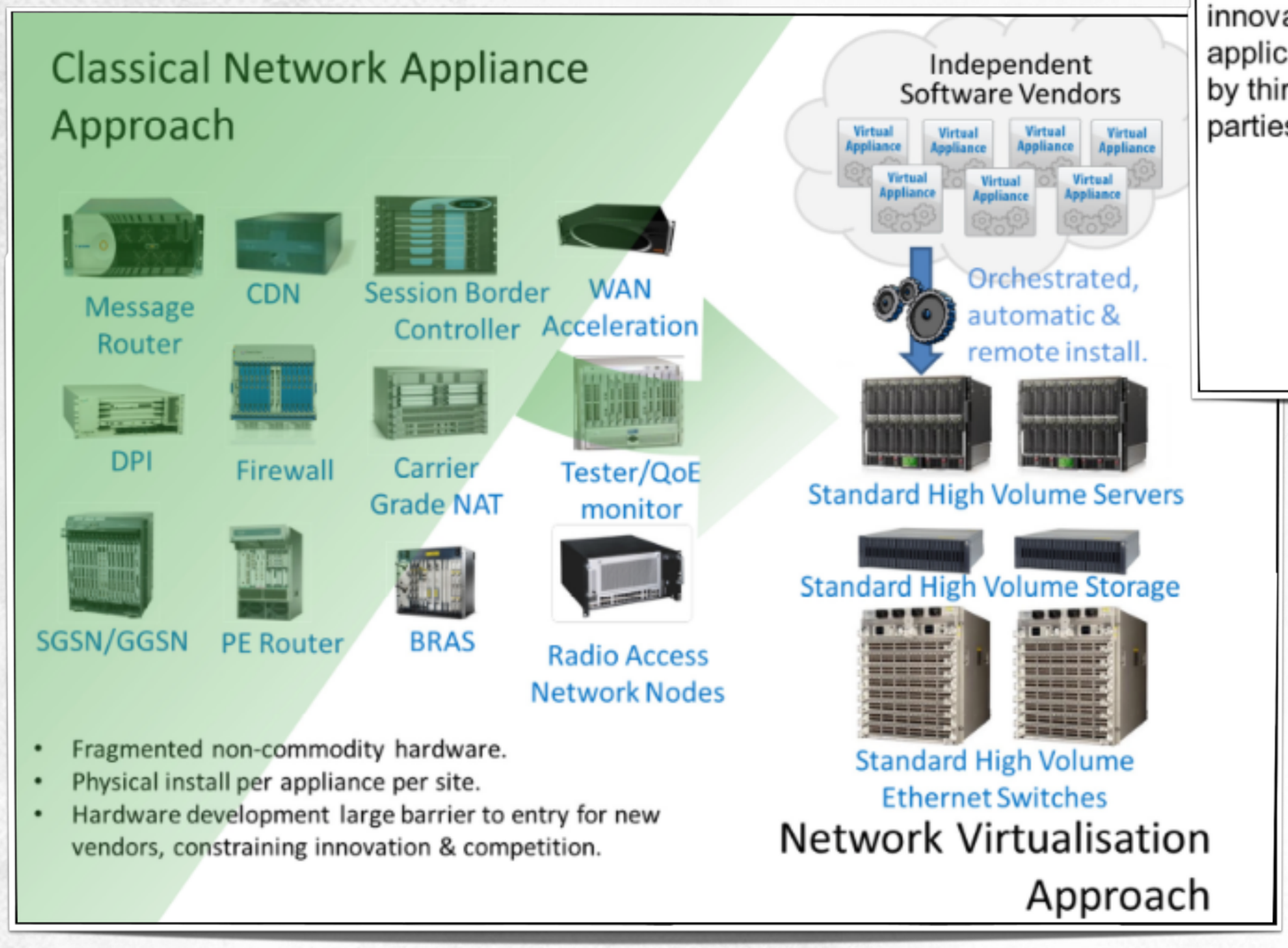
NFV

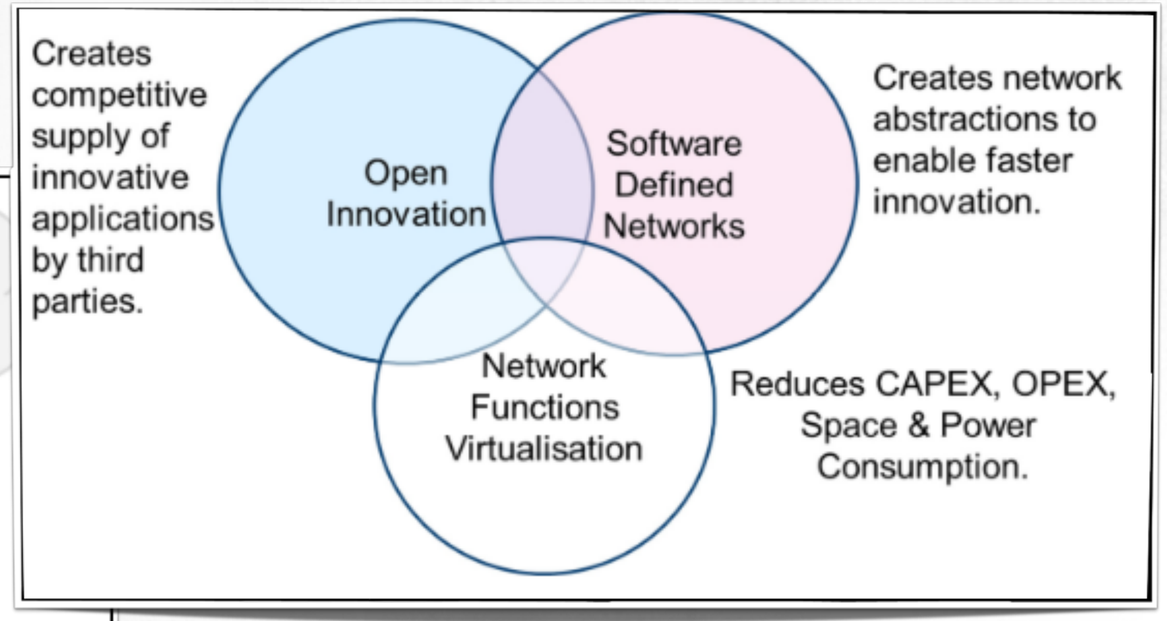
“Network Function Virtualization ?”



NFV ≠ SDN

"White Paper 1.0"

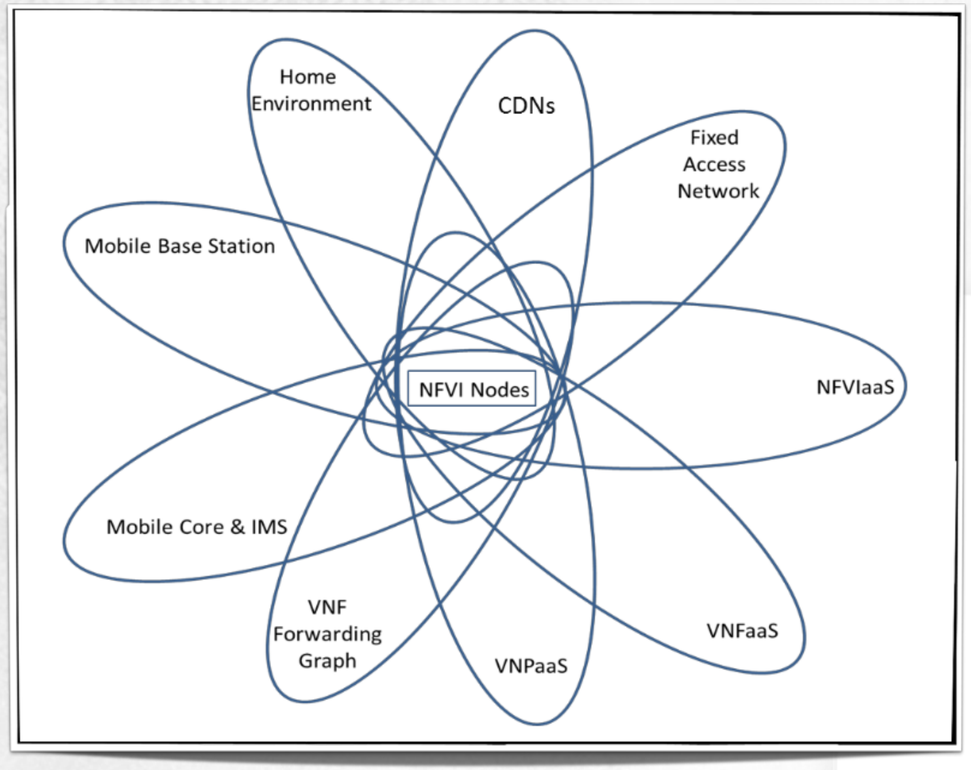


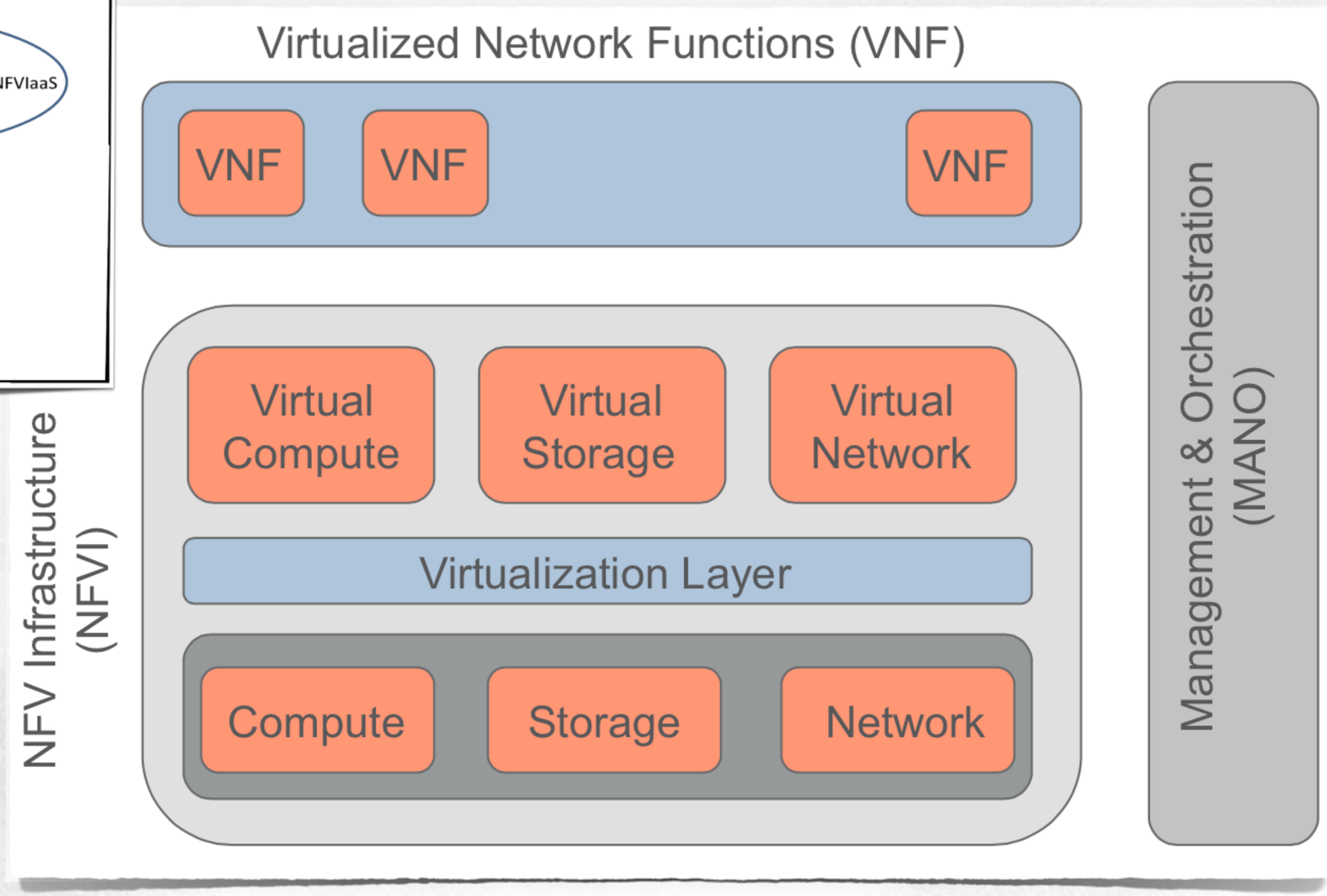


출처: http://www.etsi.org/technologies-clusters/technologies/nfv

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

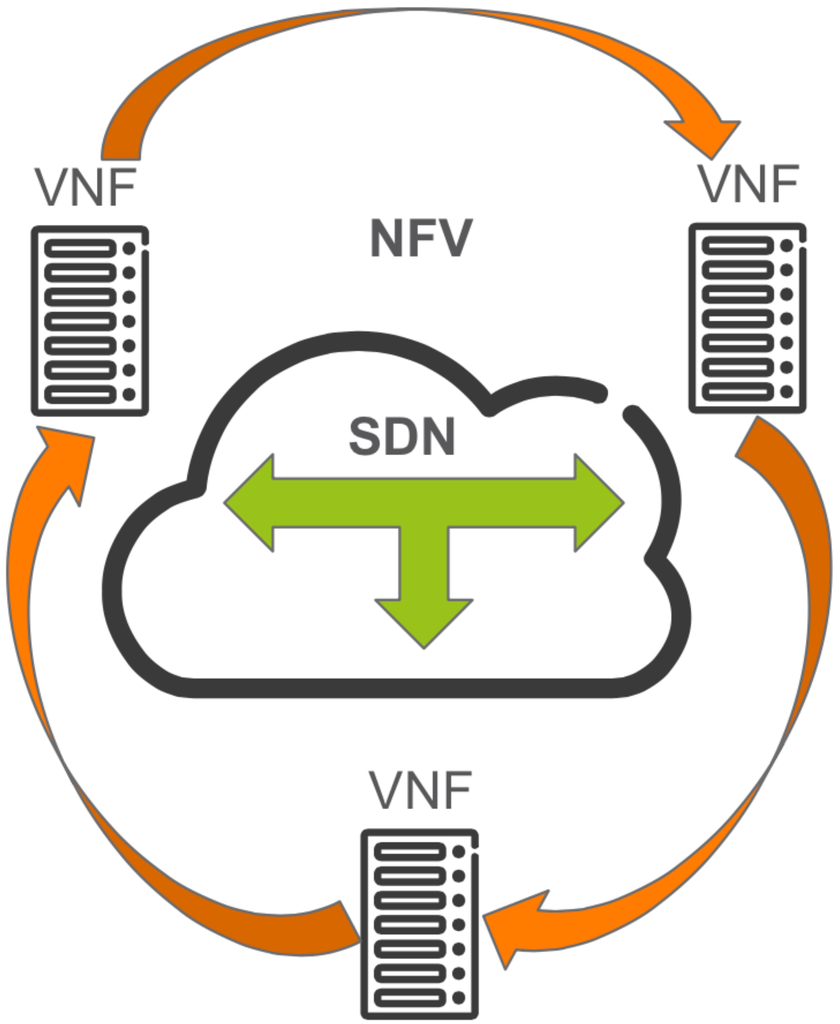
Contents

- 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"

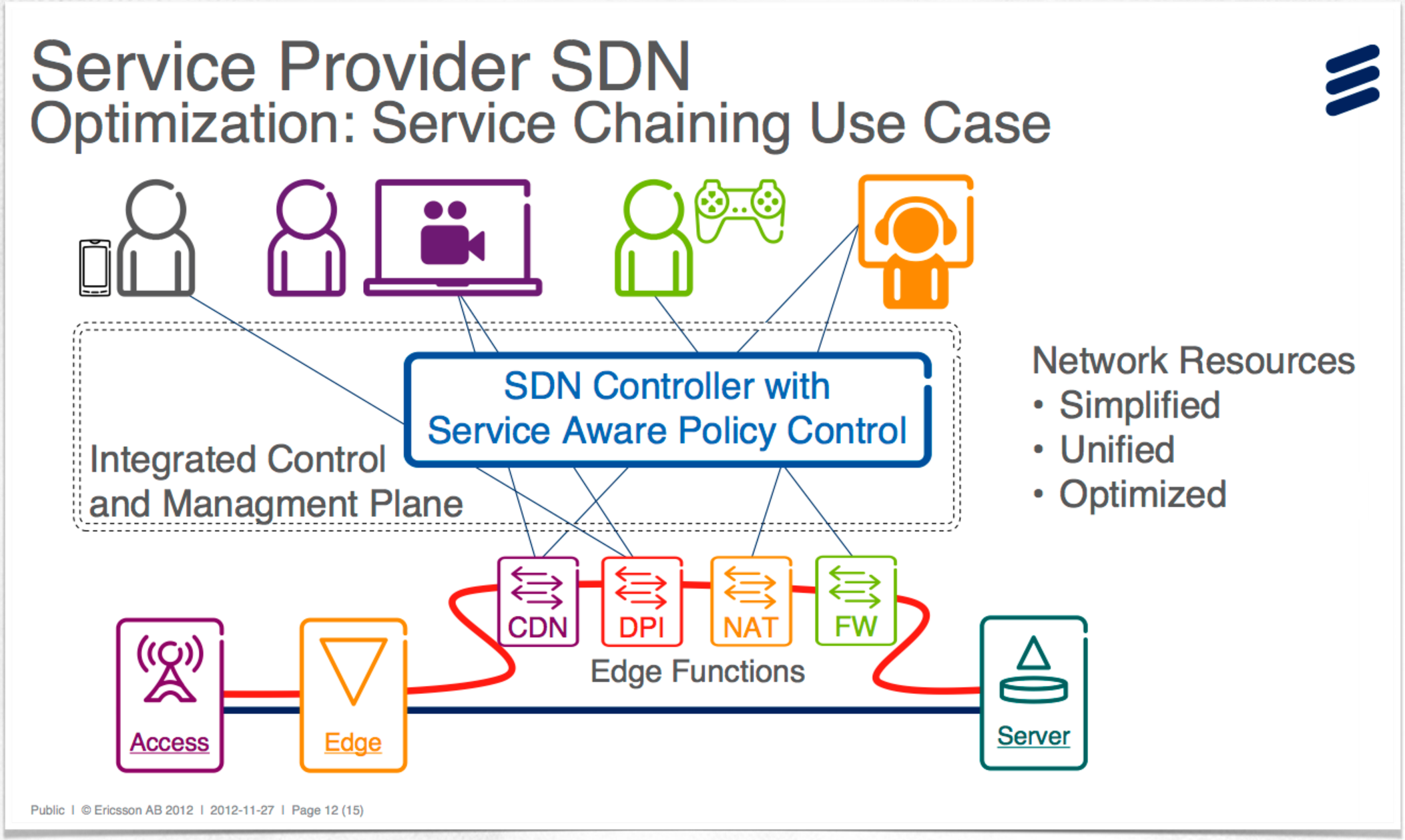
- SDN → flexible forwarding & steering of traffic in a physical or virtual network environment
- NFV → flexible placement of virtualized network functions across the network & cloud
- SDN & NFV are complementary tools for achieving full network programmability



출처: Ahmad Rostami, "The Evolution of Programmable Networks"

Service Chaining as a New Service

"SDN Deployment in Mobile Network"

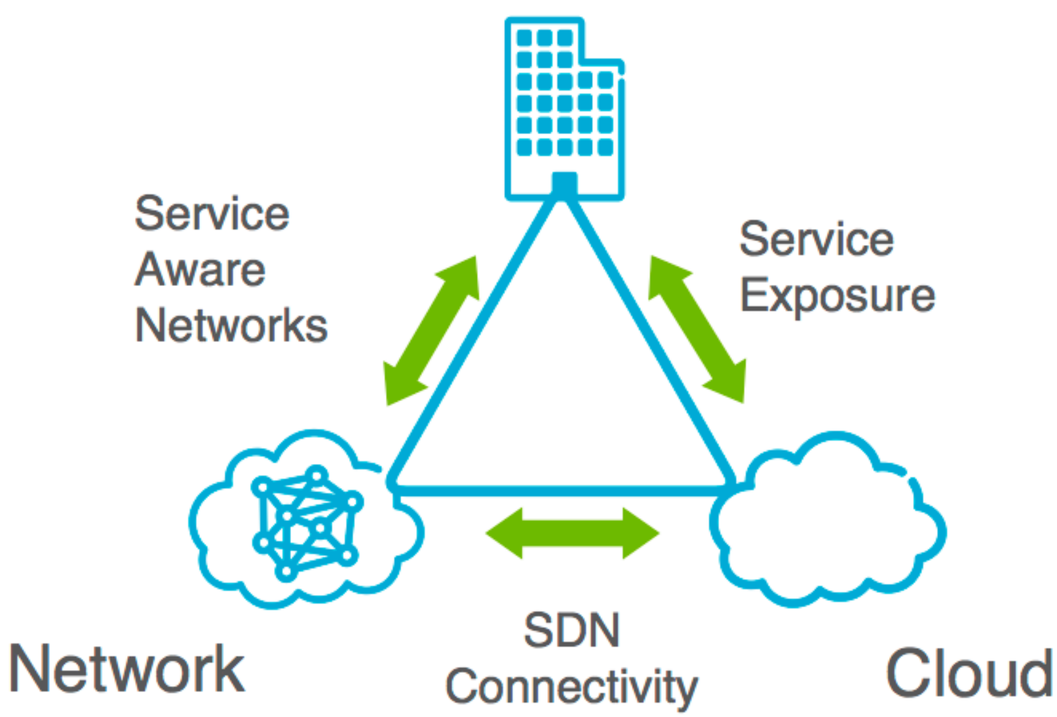


참조: http://www.ericsson.com/res/investors/docs/2012/ericsson-cloud-and-sdn.pdf

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

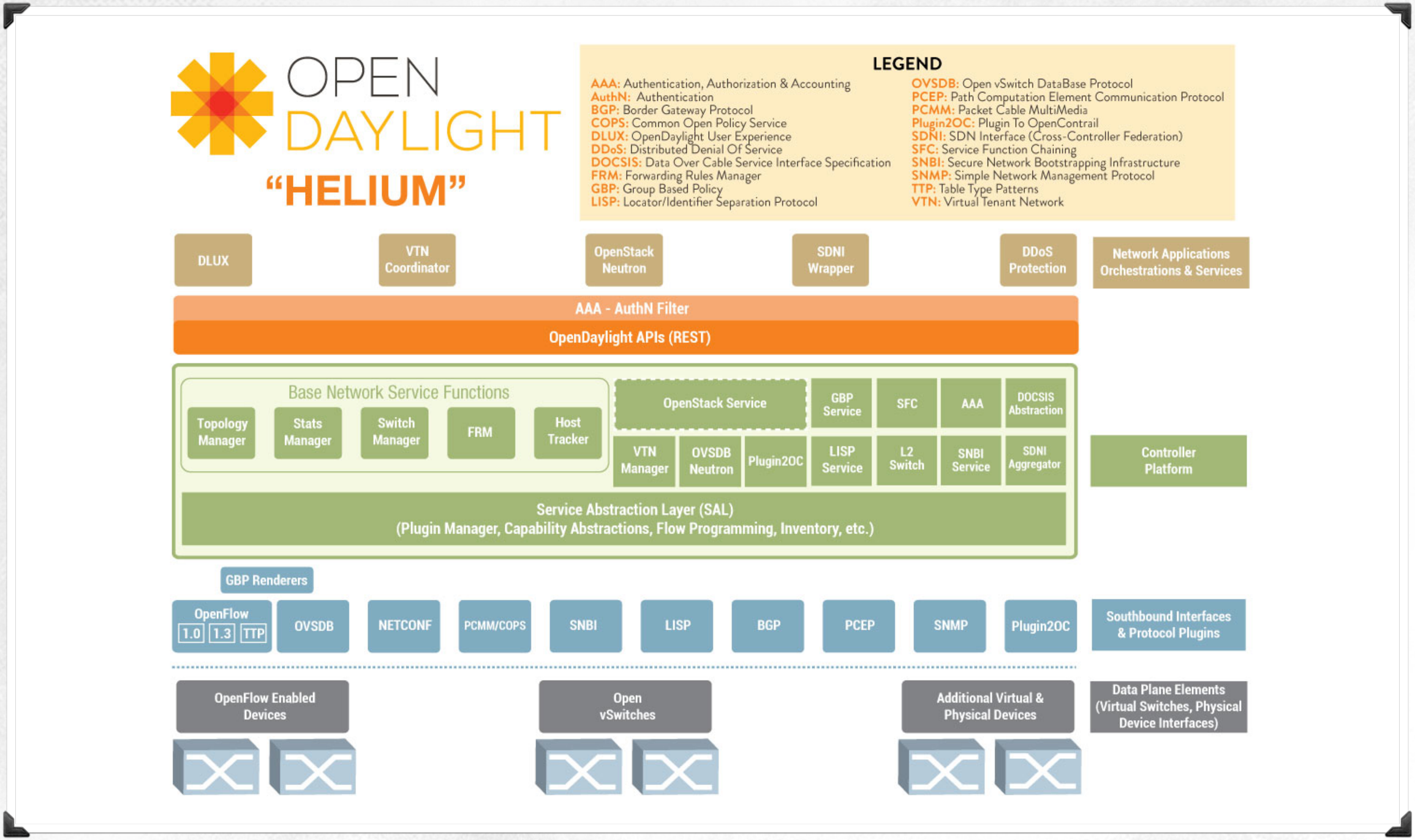
참조: http://www.ericsson.com/res/investors/docs/2012/ericsson-cloud-and-sdn.pdf

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)

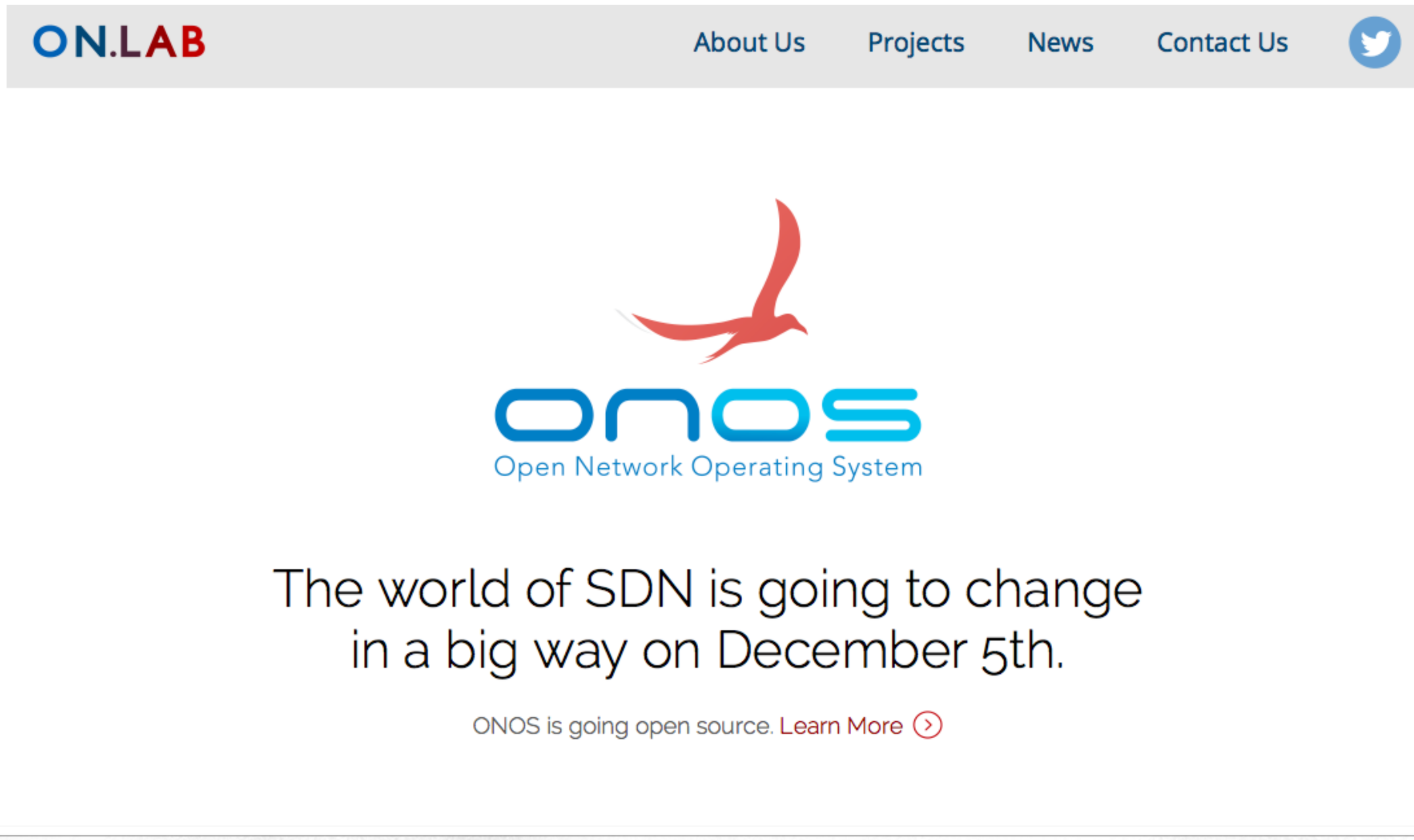
OpenDaylight

www.opendaylight.org



Open Network Operating System

onlab.us



OpenStack Networking

www.openstack.org/software/openstack-networking/

openstack CLOUD SOFTWARE

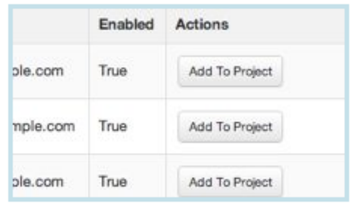
Home Software User Stories Community Foundation Profile Blog Wiki Documentation

OpenStack Networking

- Software
- Compute
- Storage
- Networking
- Dashboard
- Shared Services
- Getting Started
- Roadmap
- OpenStack Marketplace
- Latest Release

OpenStack Networking: Pluggable, scalable, API-driven network and IP management

Dashboard Demo



Projects & Users

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.

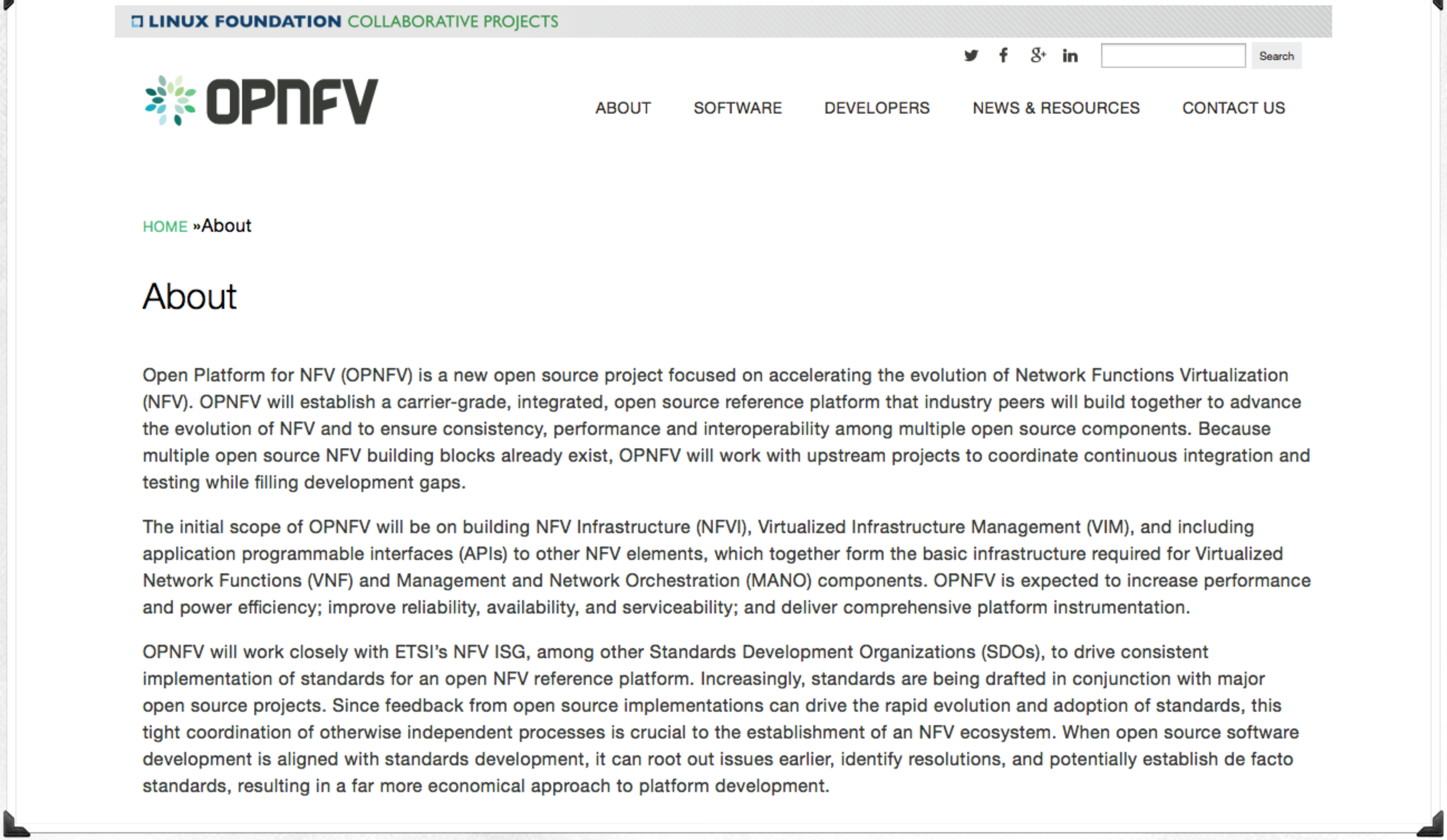
These IPAM capabilities enable you to:

- 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 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



Open Network Linux

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

GitHub This repository Search Explore Features Enterprise Blog Sign up Sign in

CPqD / ofsoftswitch13 ★ Star 81 Fork 75

OpenFlow 1.3 for OpenWRT

srujan edited this page on Oct 11, 2013 · 4 revisions

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.

Pre-compiled image

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

Build instructions

Pages 5

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

Clone this wiki locally

https://github.com/CPqD/ofsof

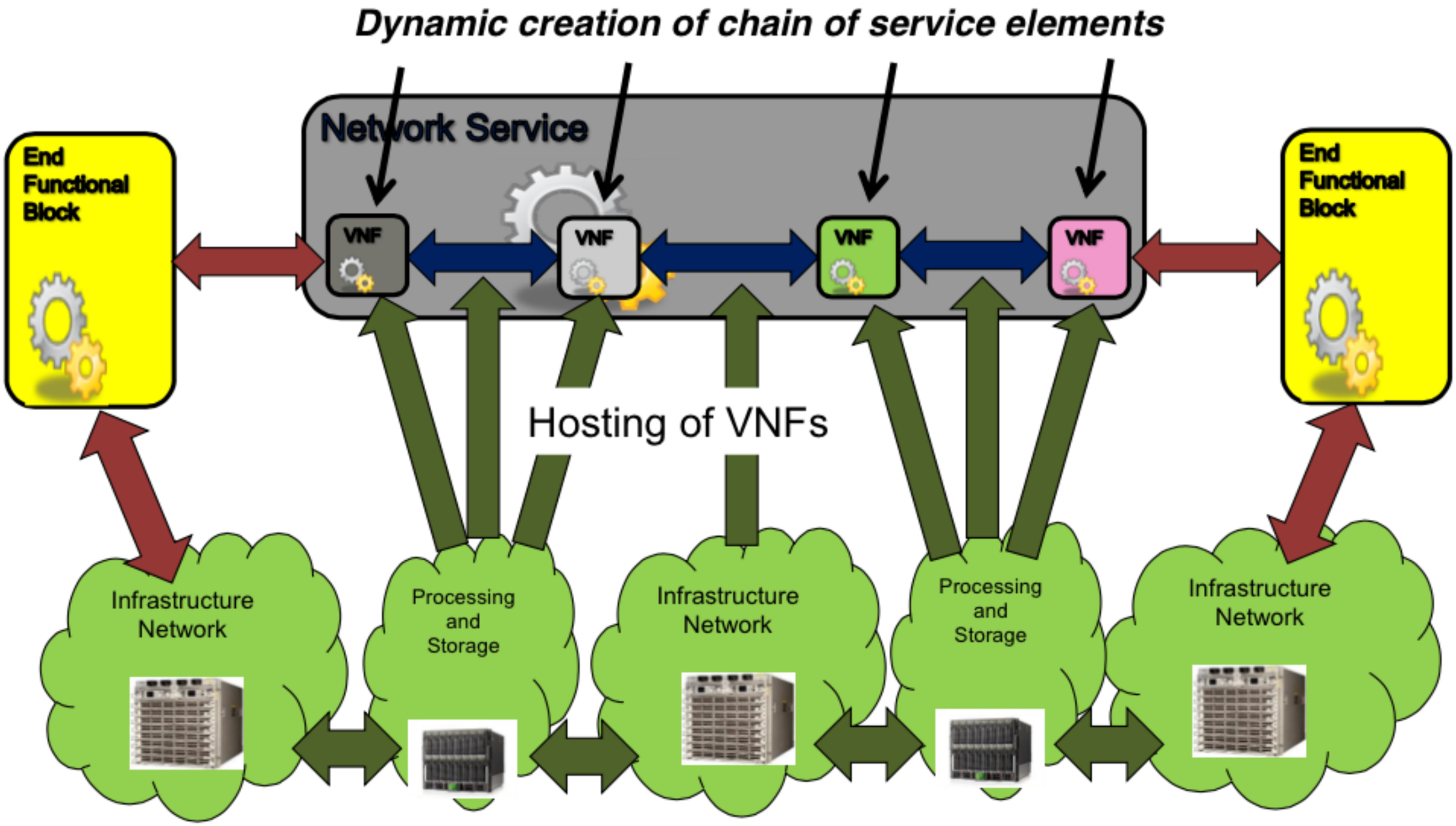
Clone in Desktop

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)

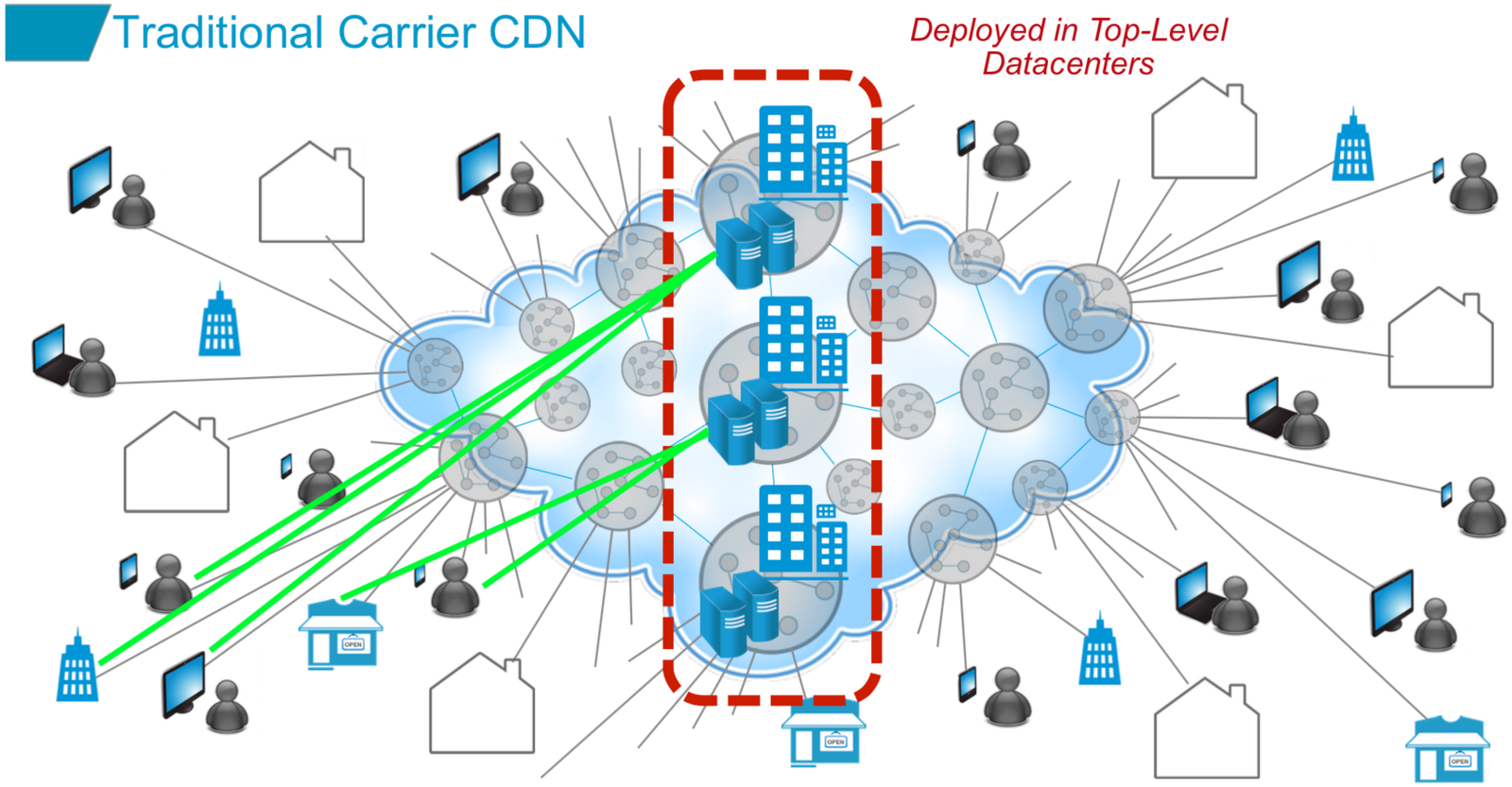
Use Cases in REAL

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

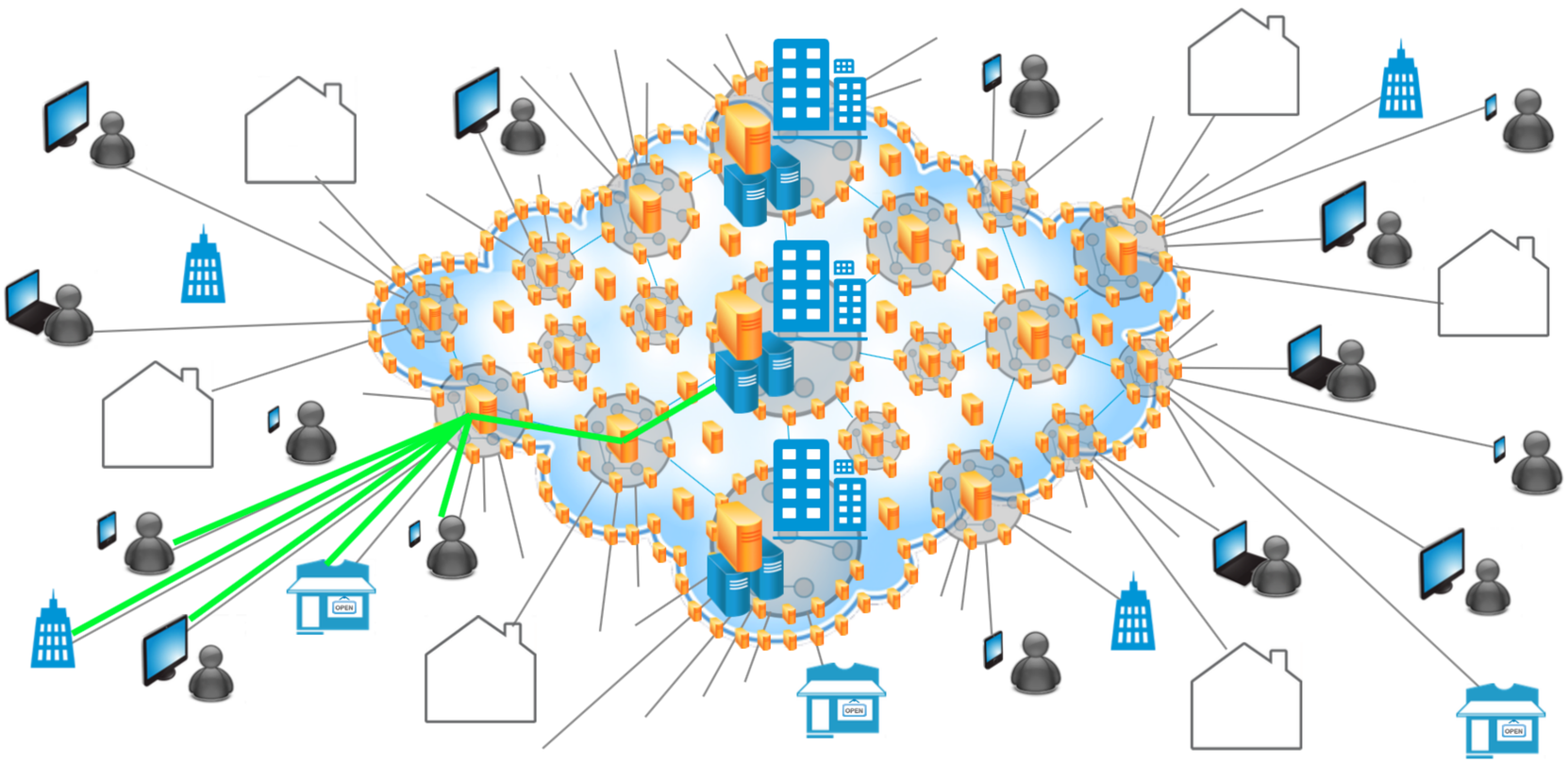


참조: BT, “NFV Orchestration by Layers and APIs”

Traditional Carrier CDN

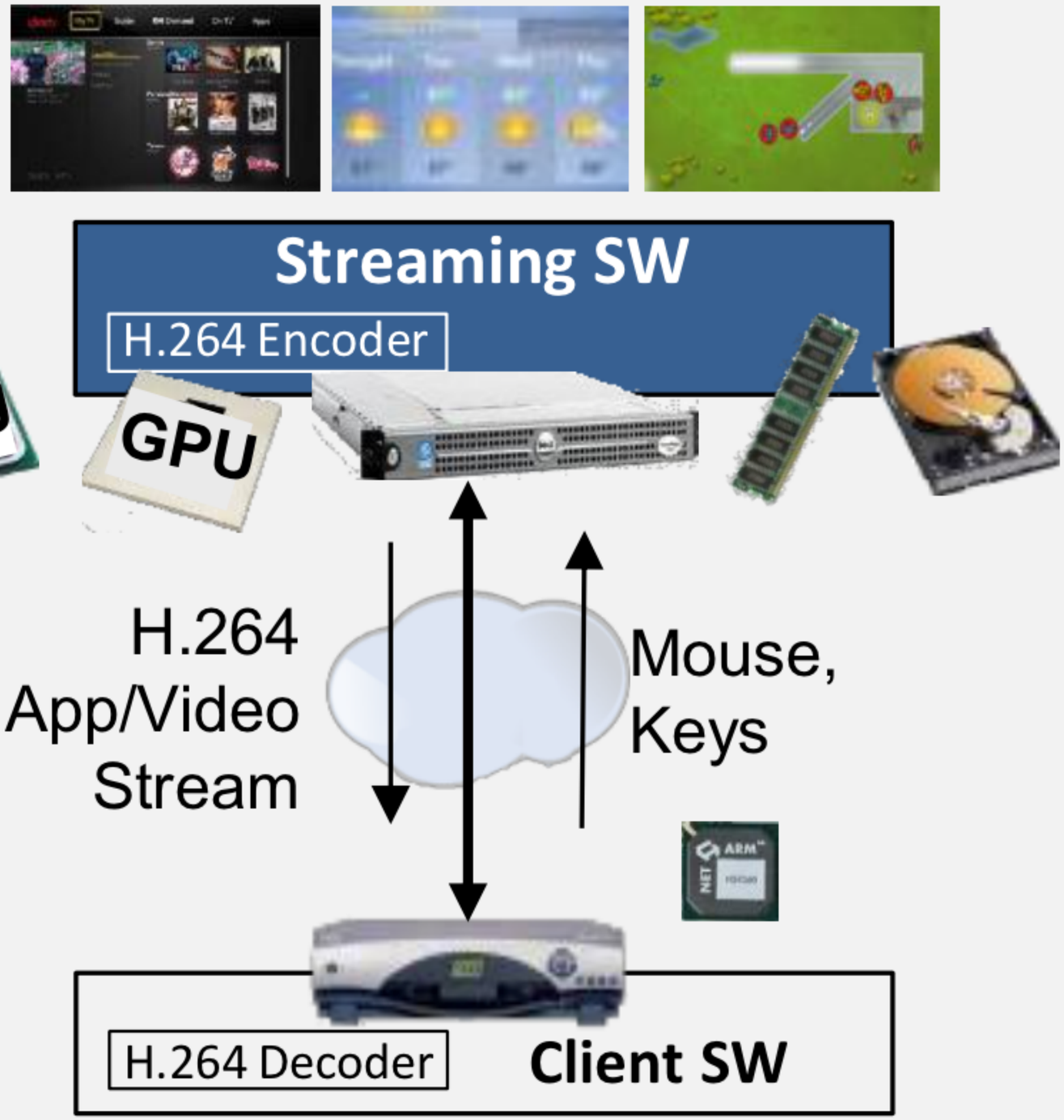


Scale Out and Deep with NFV



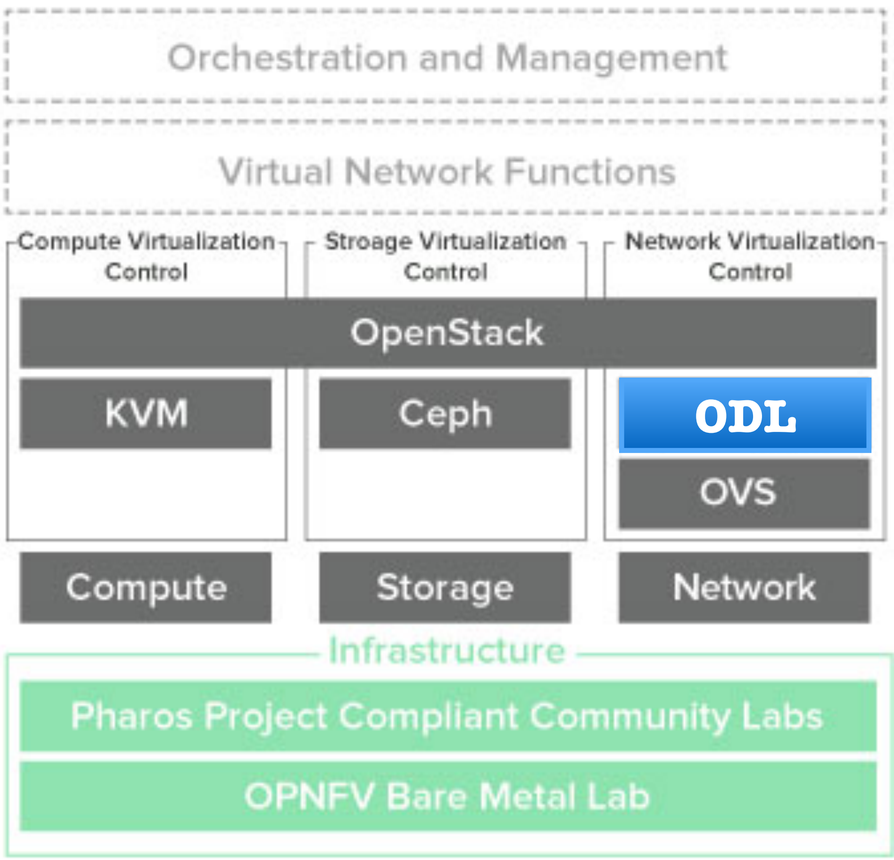
참조: Akamai, “Caches, Caches Everywhere Scaling for Big Video Events”

Virtualized STB Apps

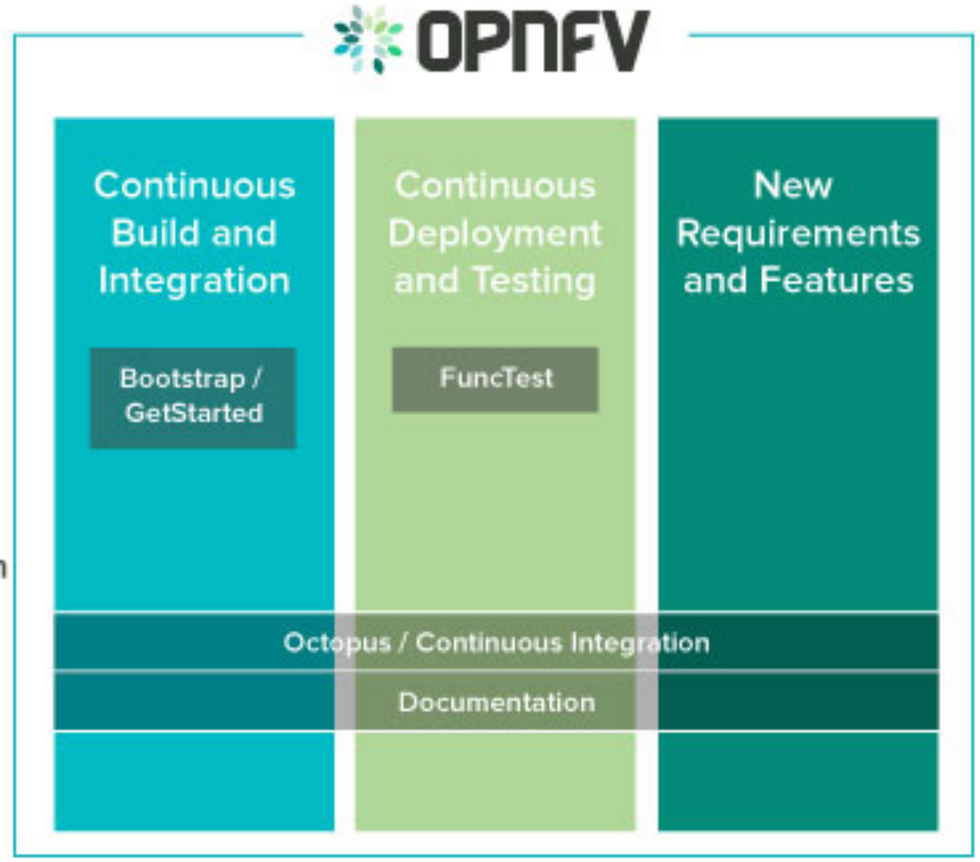


Today's SDN/NFV

TYPE #1: Telco Friendly Approach



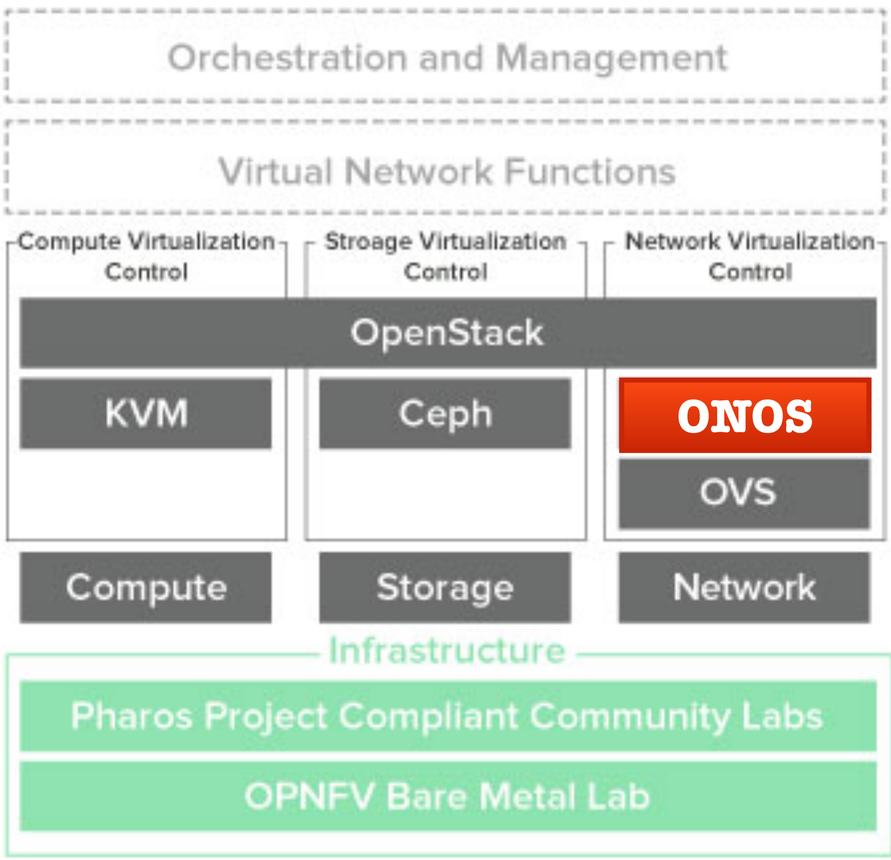
Upstream Project Collaboration



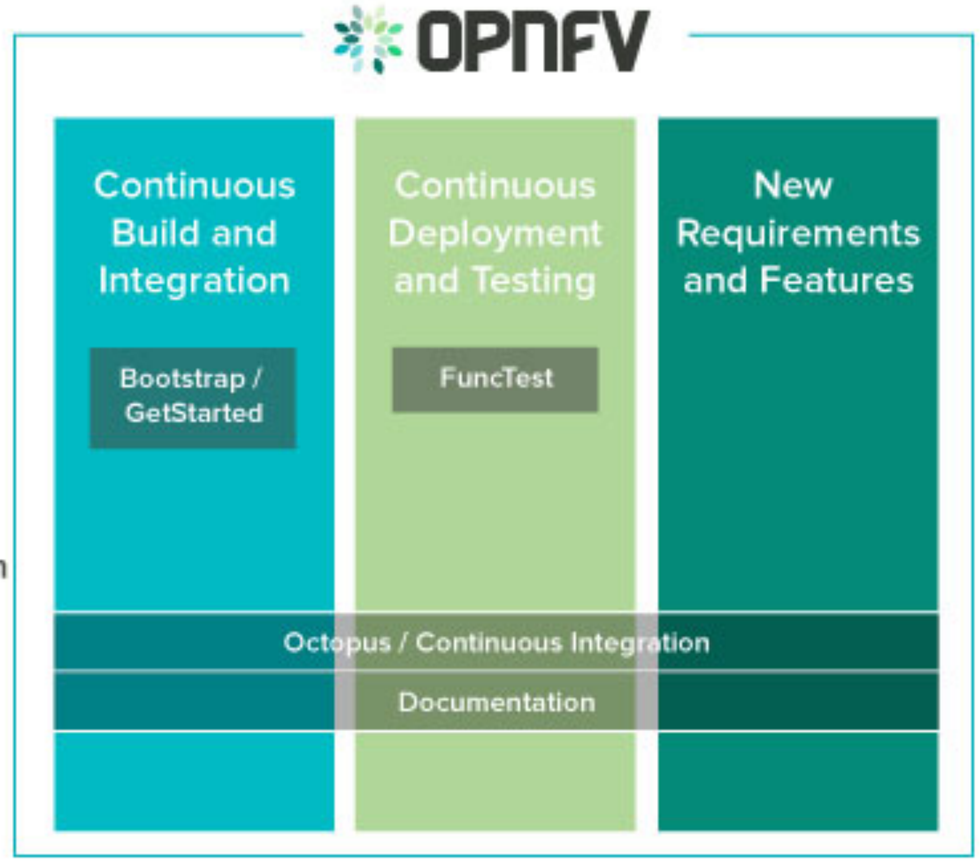
출처: www.opnfv.org

Today's SDN/NFV

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



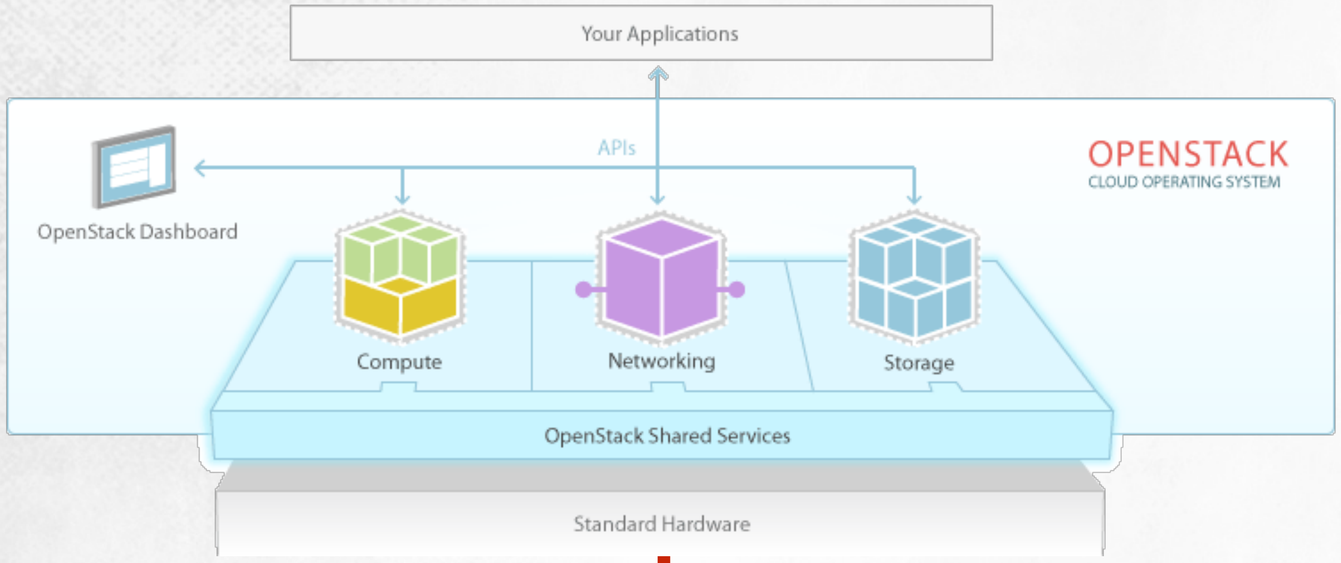
Upstream Project Collaboration

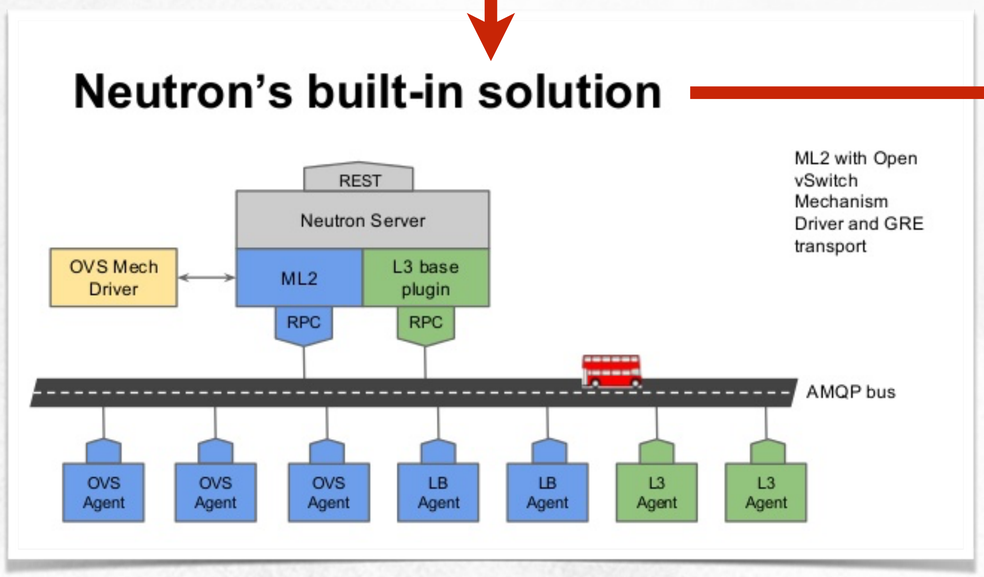


출처: www.opnfv.org

Today's SDN/NFV

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



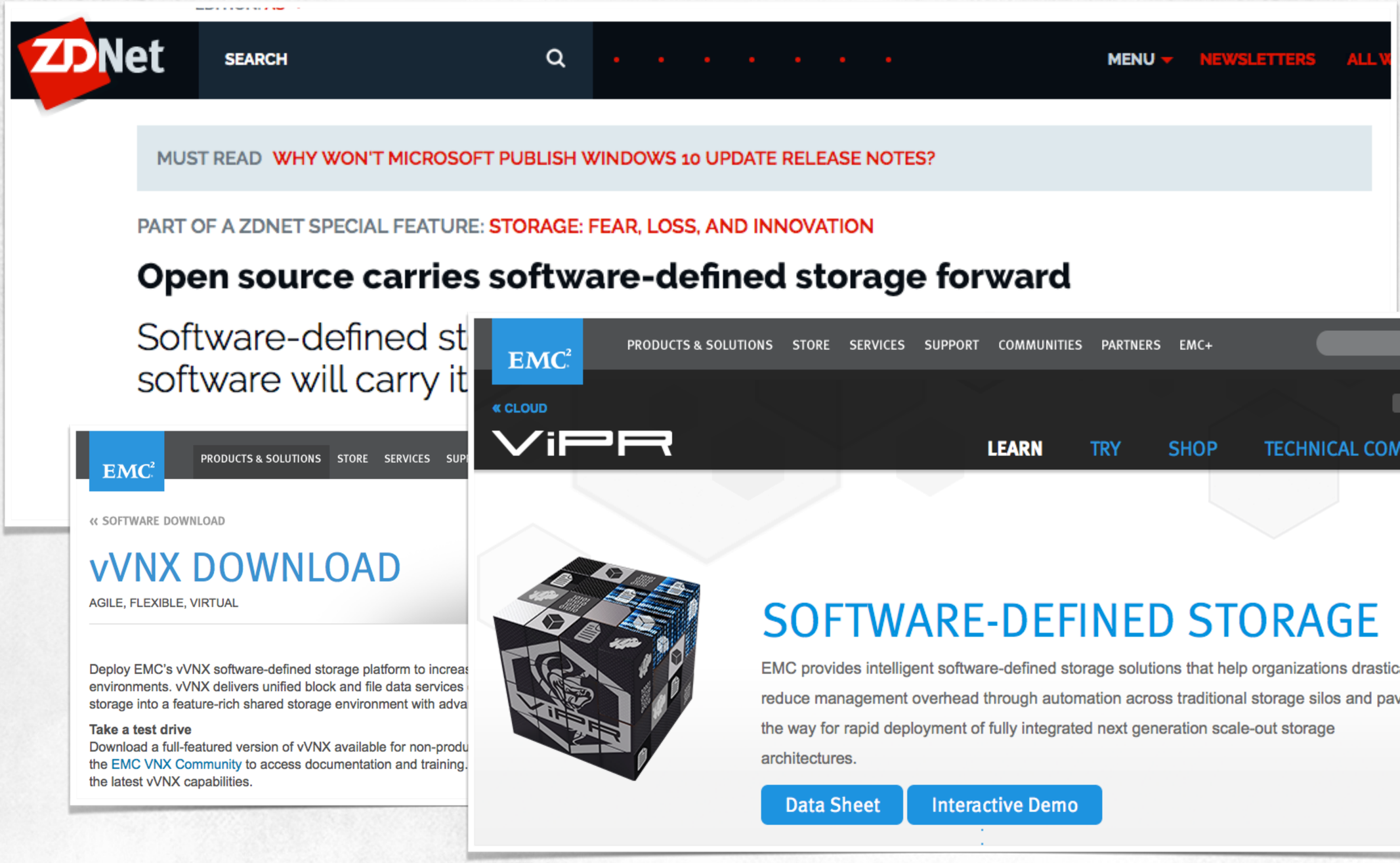


- Open vSwitch Plugin
- Cisco UCS/Nexus Plugin
- Cisco Nexus1000v Plugin
- Linux Bridge Plugin
- Modular Layer 2 Plugin
- Nicira Network Virtualization Platform (NVP) Plugin
- Ryu OpenFlow Controller Plugin
- NEC OpenFlow Plugin
- Big Switch Controller Plugin
- Cloudbase Hyper-V Plugin
- MidoNet Plugin
- Brocade Neutron Plugin Brocade Neutron Plugin
- PLUMgrid Plugin
- Mellanox Neutron Plugin Mellanox Neutron Plugin
- Embrane Neutron Plugin
- IBM SDN-VE Plugin
- CPLANE NETWORKS CPLANE NETWORKS
- Nuage Networks Plugin
- OpenContrail OpenContrail Plugin
- Extreme Networks Plugin
- Ruijie Networks Plugin
- Juniper Networks Neutron Plugin
- Calico Neutron Plugin (docs)

출처: openstack.org

Storage

Open Source Storage, Software Defined Storage



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”

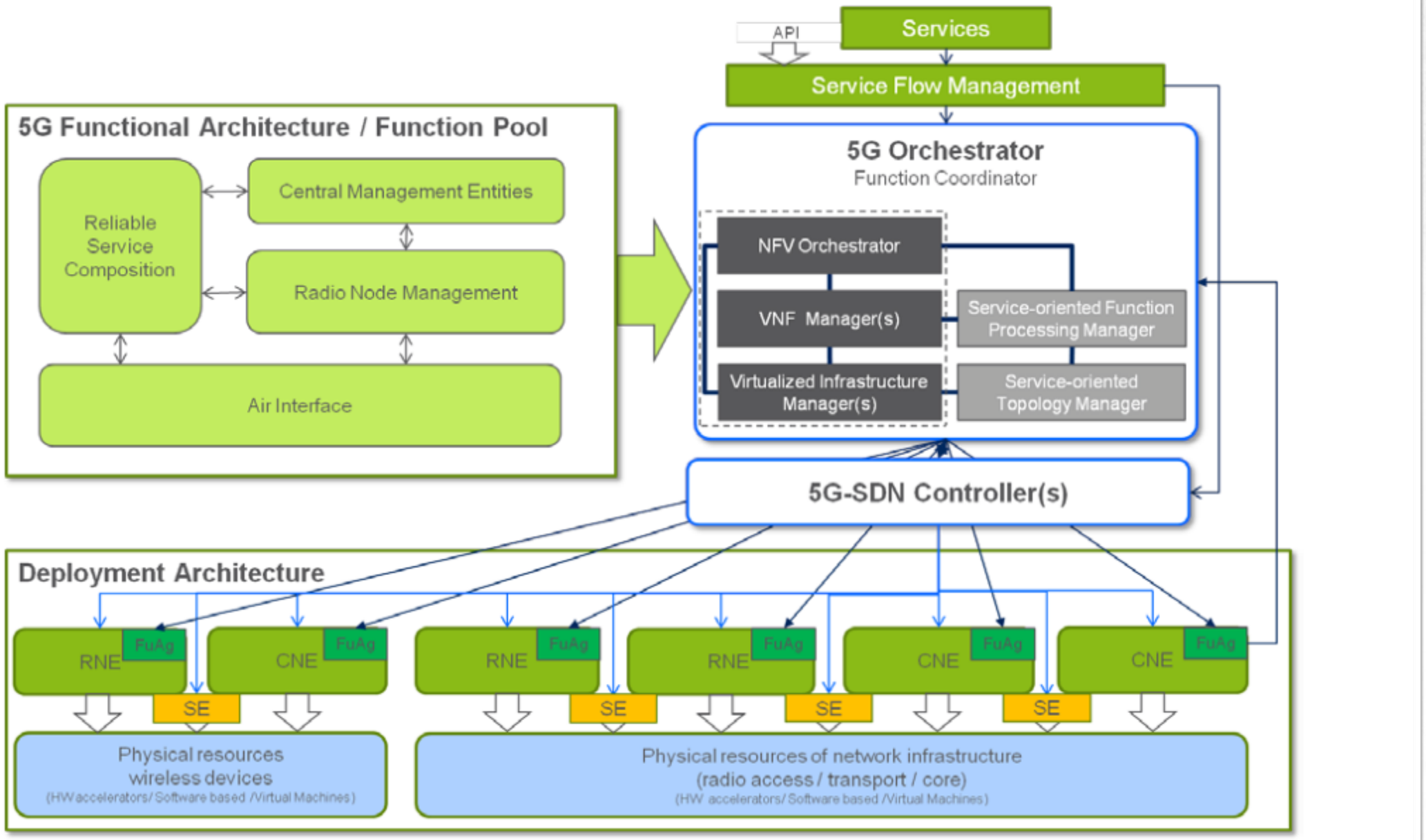


Figure 3-6: Orchestration & Control Architecture View of METIS 5G System.

‘Telco SDN/NFV’ for 5G Mobile Network

Computing in everywhere

(단말 포함)

출처: Ericsson Technology Review

‘Telco SDN/NFV’ for 5G Mobile Network

“METIS2020 - Network & Device Communication”

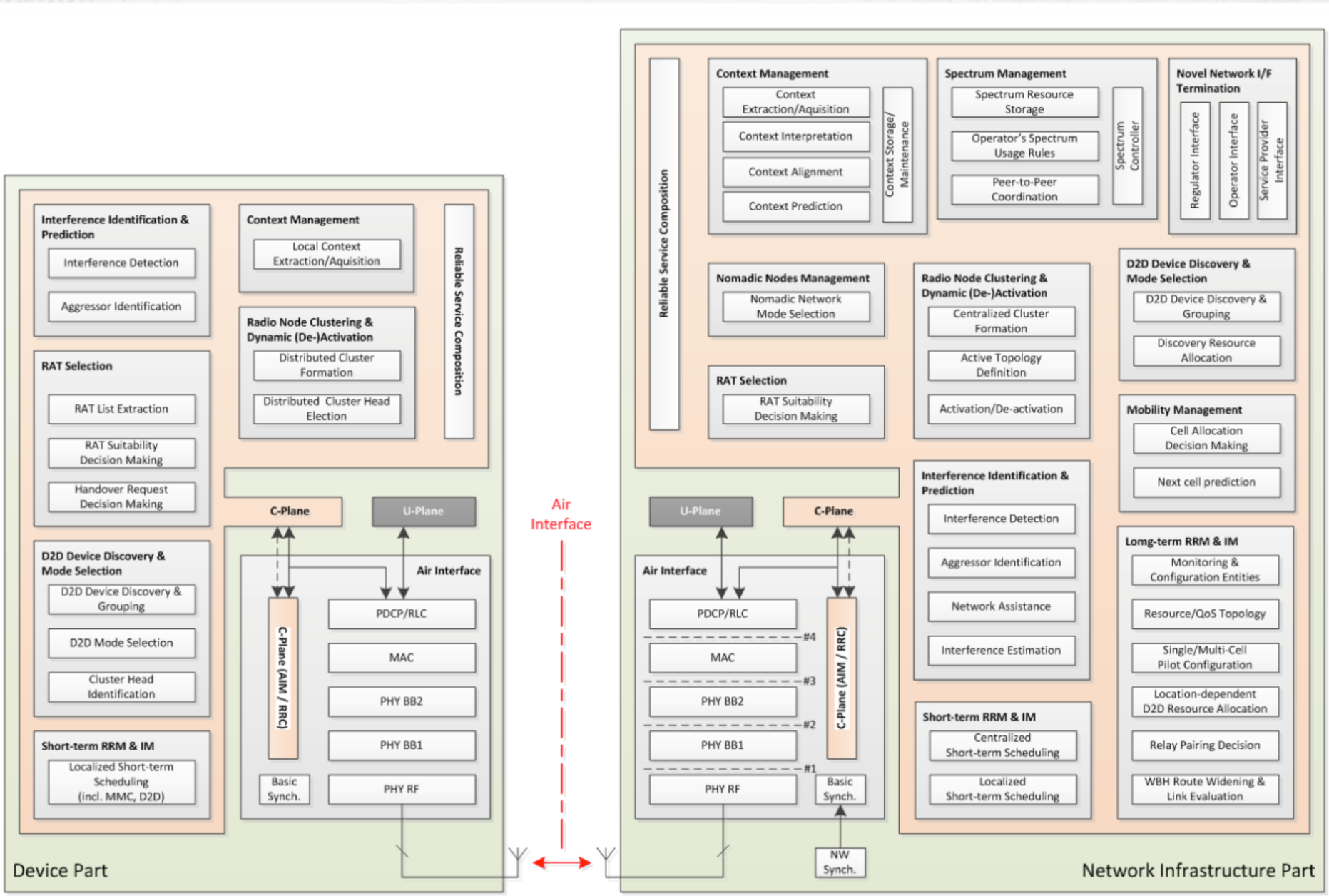


Figure 3-4: Generic functional architecture of METIS 5G System considering interrelations between infrastructure and device part (network-to-device communication).

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: Recogniti

- Collaboration with the M
  students to push bounda
- "Identity module" to be jo
  preferences and behavio
- Establishes innovation-b

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

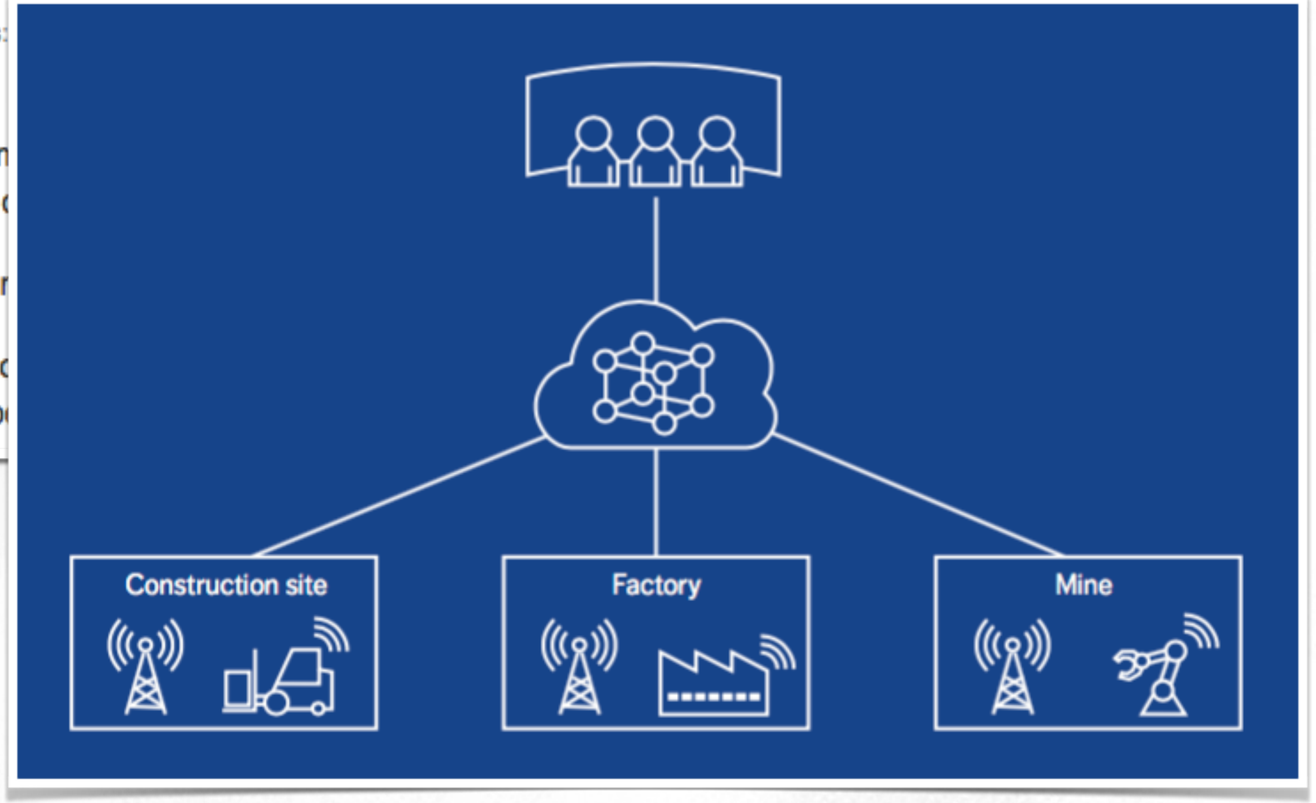
2016-03-31 Categories: In

- Presents the ultima
  customers to conti
- New product and s
  demonstrate Erics
- Highlights how Eri
  the transformation
  era of television

Ericsson and Telefónica demonstrate live delivery of LTE-U

2016-03-30 Categories:

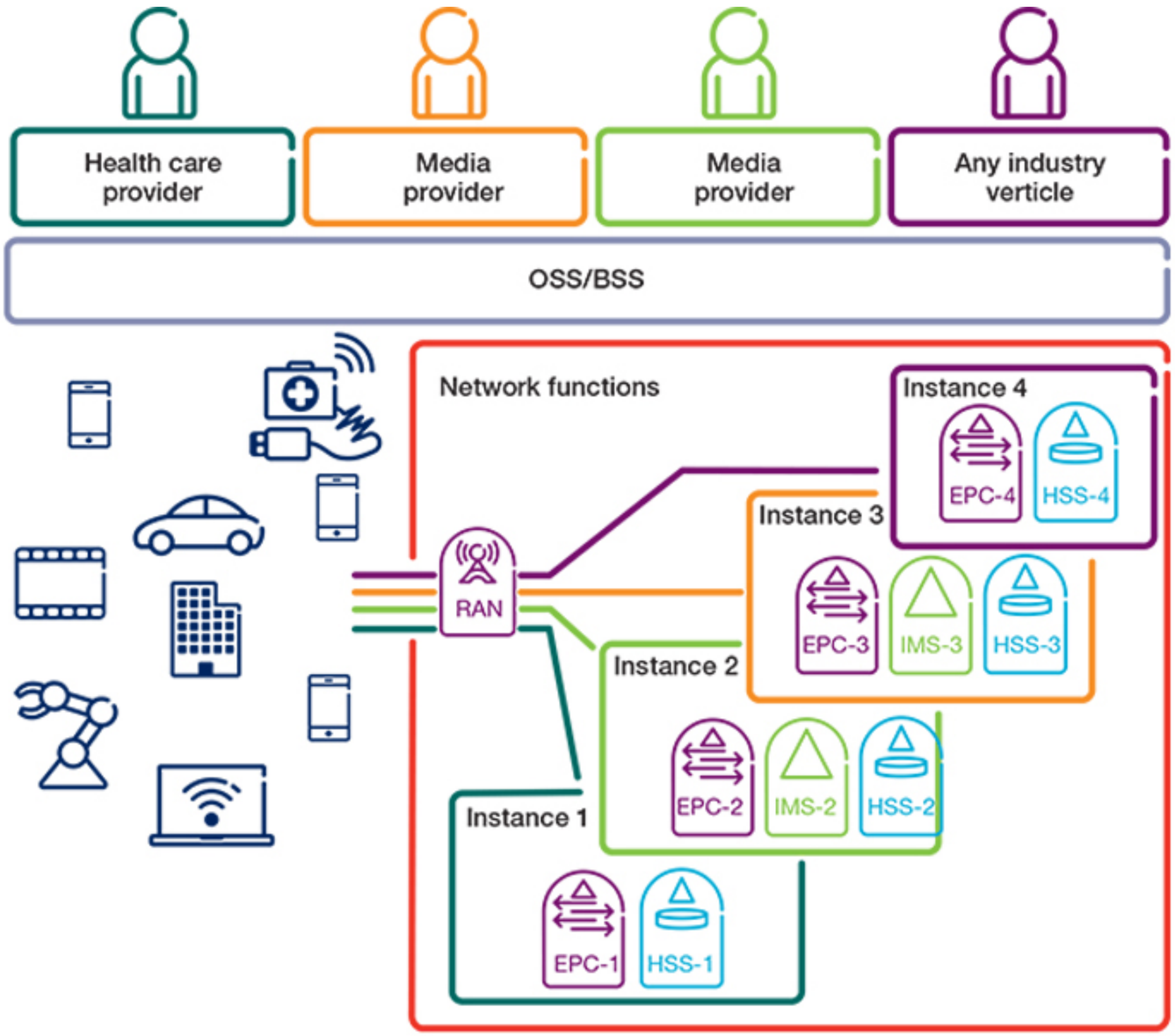
- Over-the-air den
  RBS 6402 pico
- Ericsson annour
- LTE-U enables
  licensed LTE sp

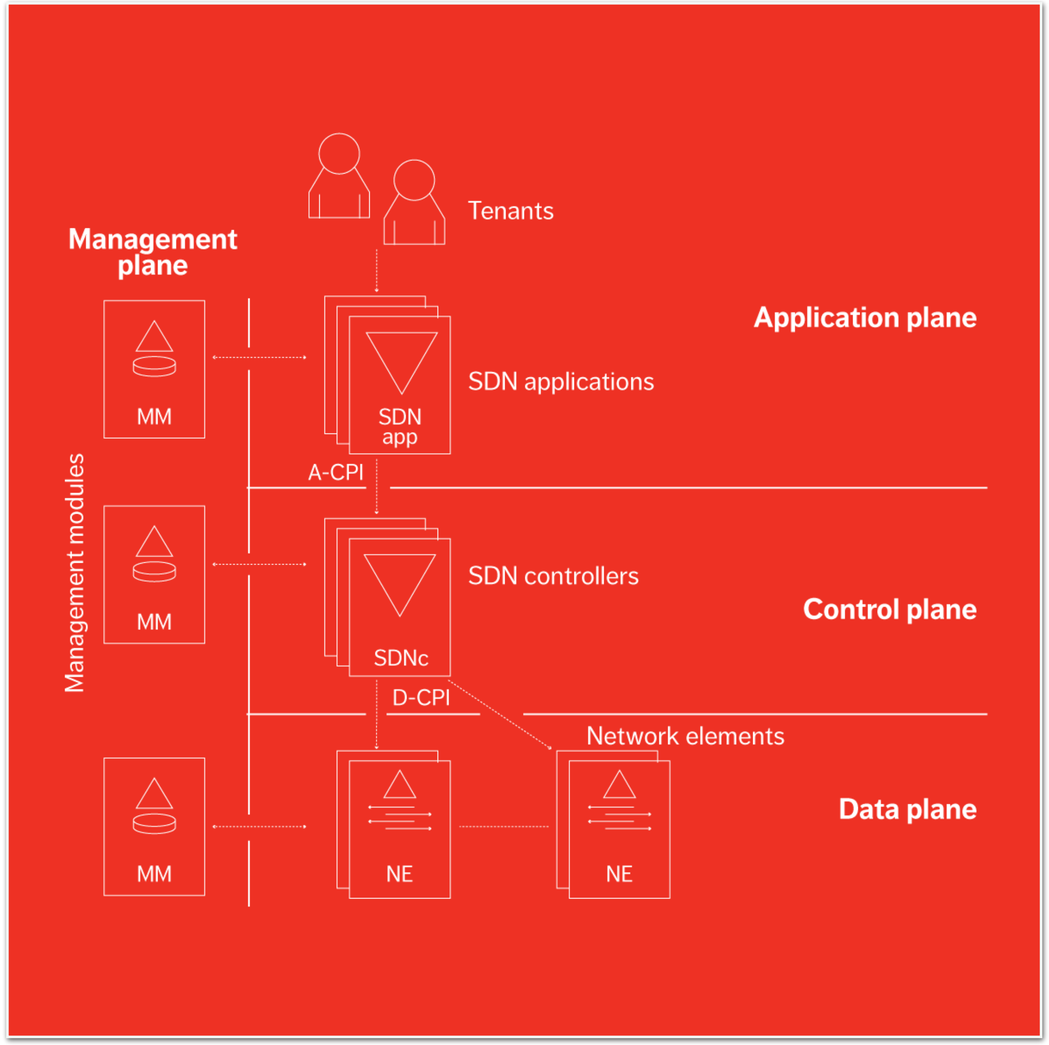


출처: Ericsson Technology Review

Design Principle of 5G Networking

"일반 사용자에서 새로운 사용자(사업)를 받아들이는 그릇"

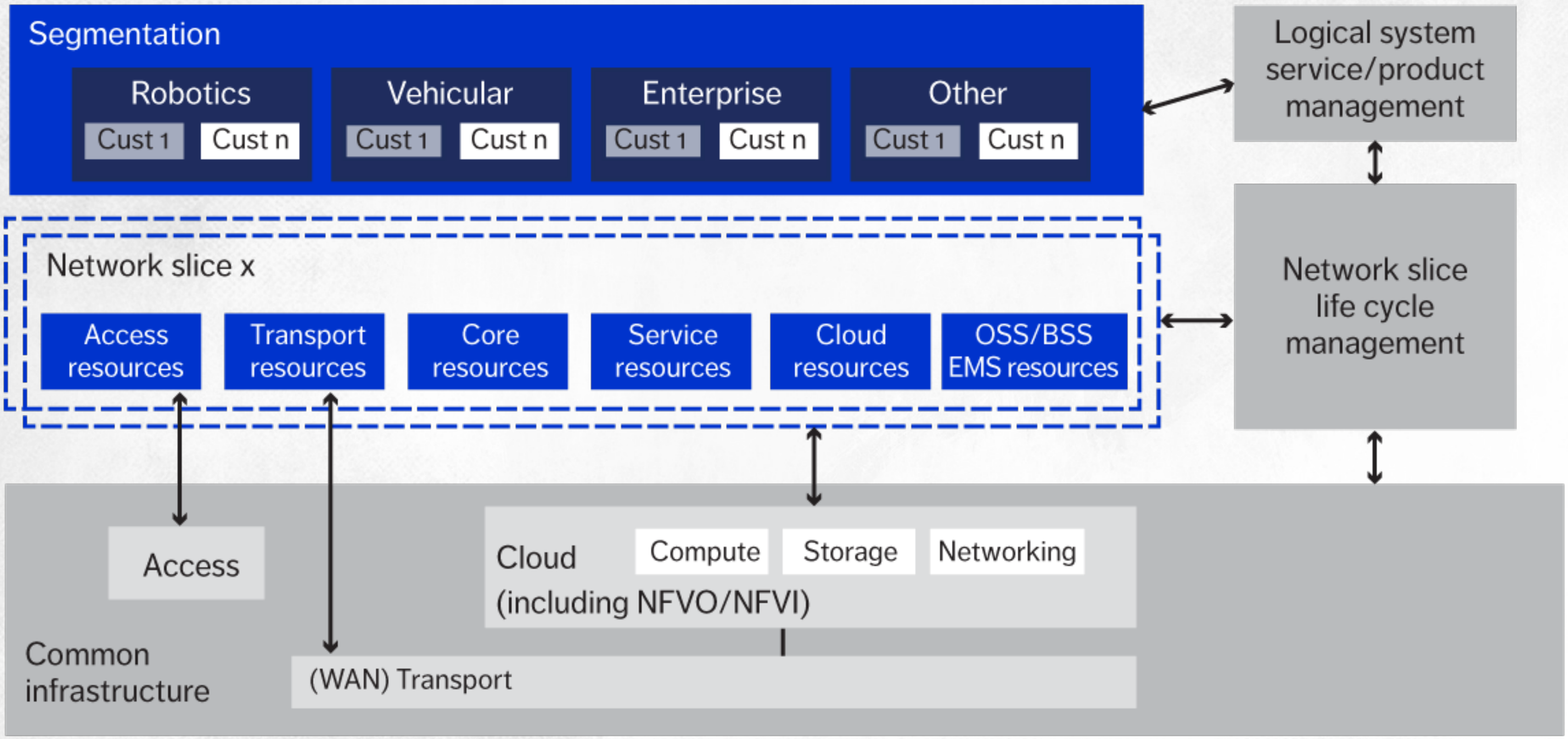




출처: Ericsson Technology Review

Design Principle of 5G Networking

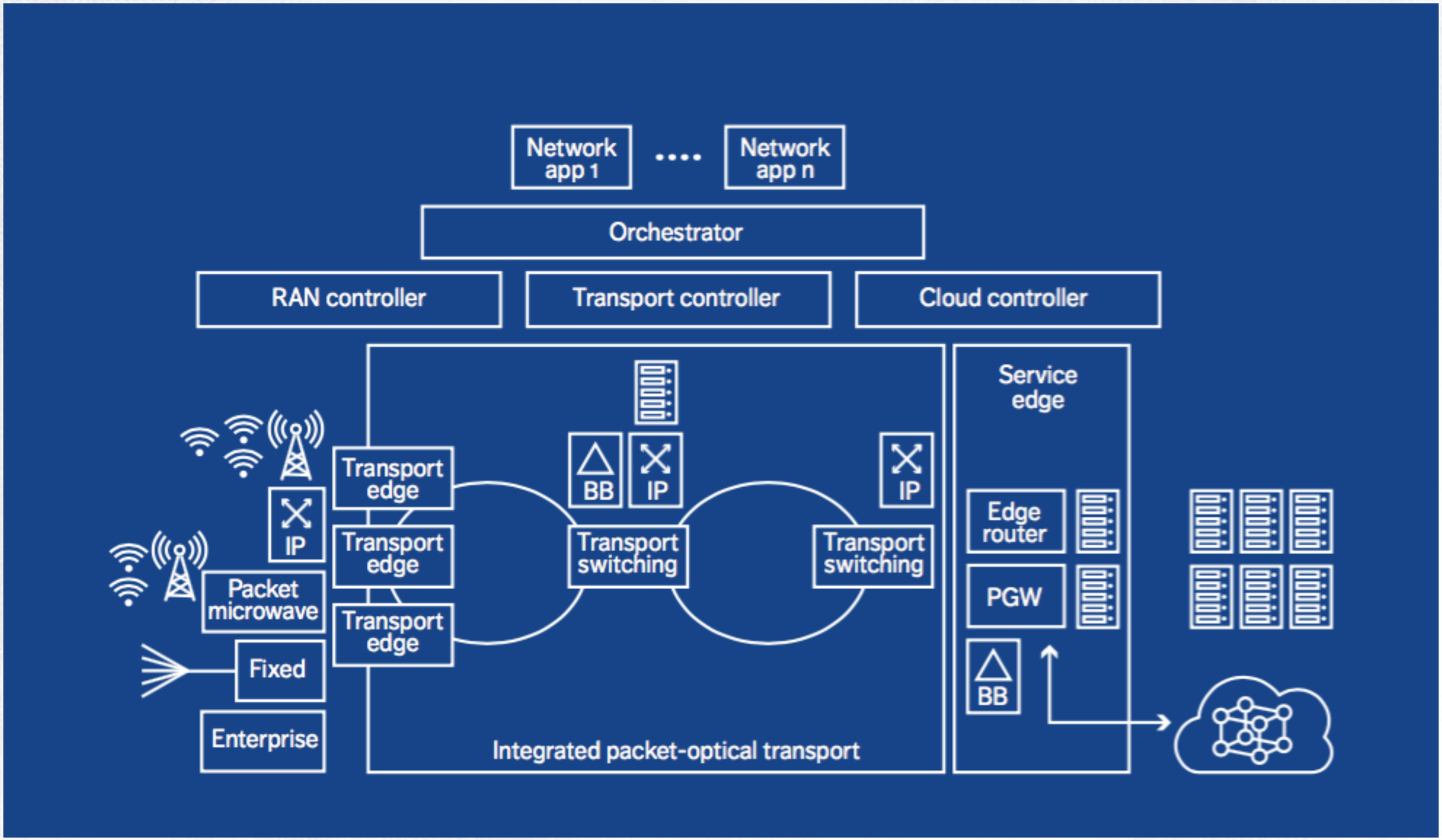
"일반 사용자에서 새로운 사용자(사업)를 받아들이는 그릇"



출처: Ericsson Technology Review

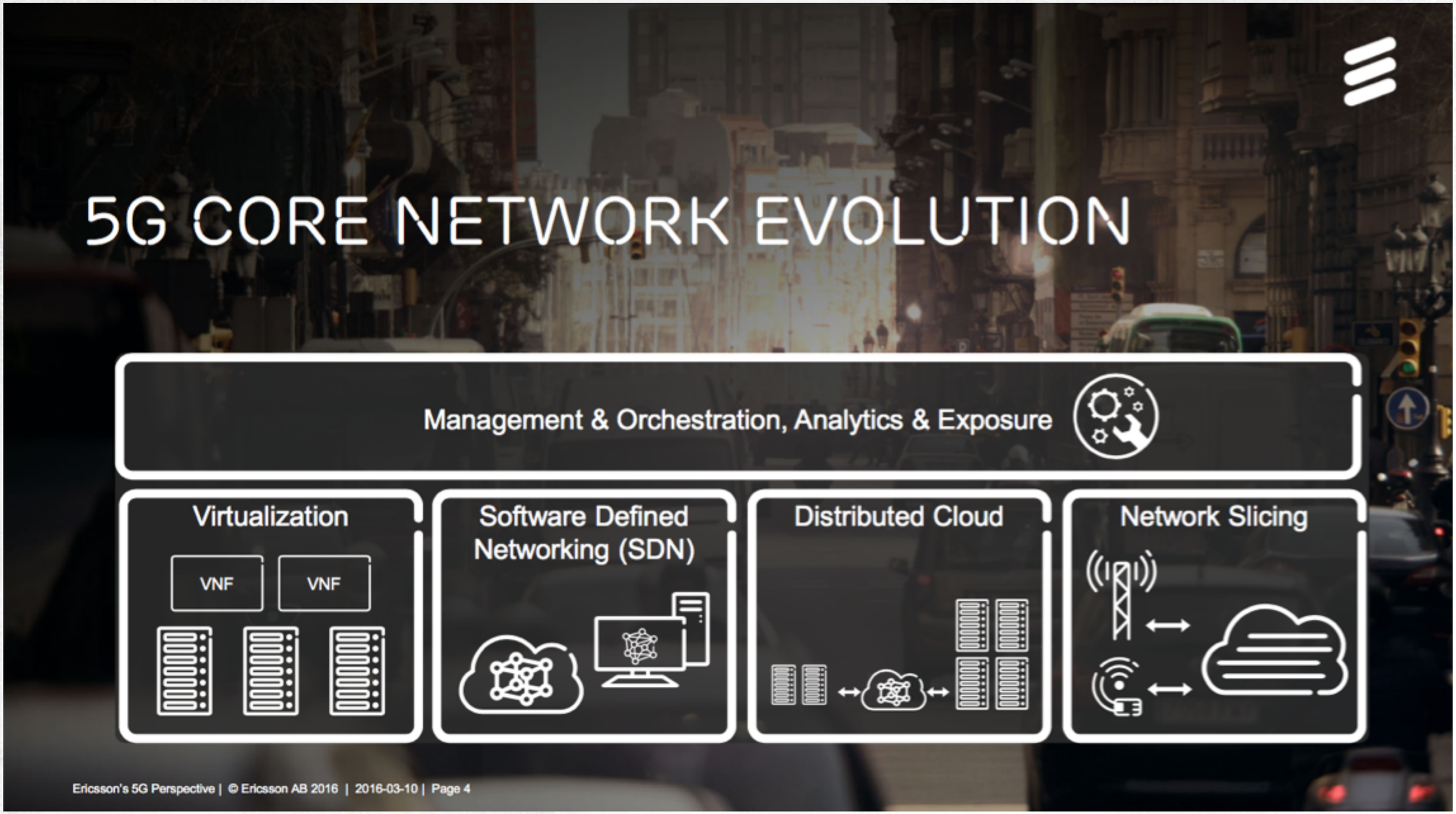
Design Principle of 5G Networking

"일반 사용자에서 새로운 사용자(사업)를 받아들이는 그릇"



출처: Ericsson Technology Review

Design Principle of 5G Networking



출처: Ericsson Technology Review

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)

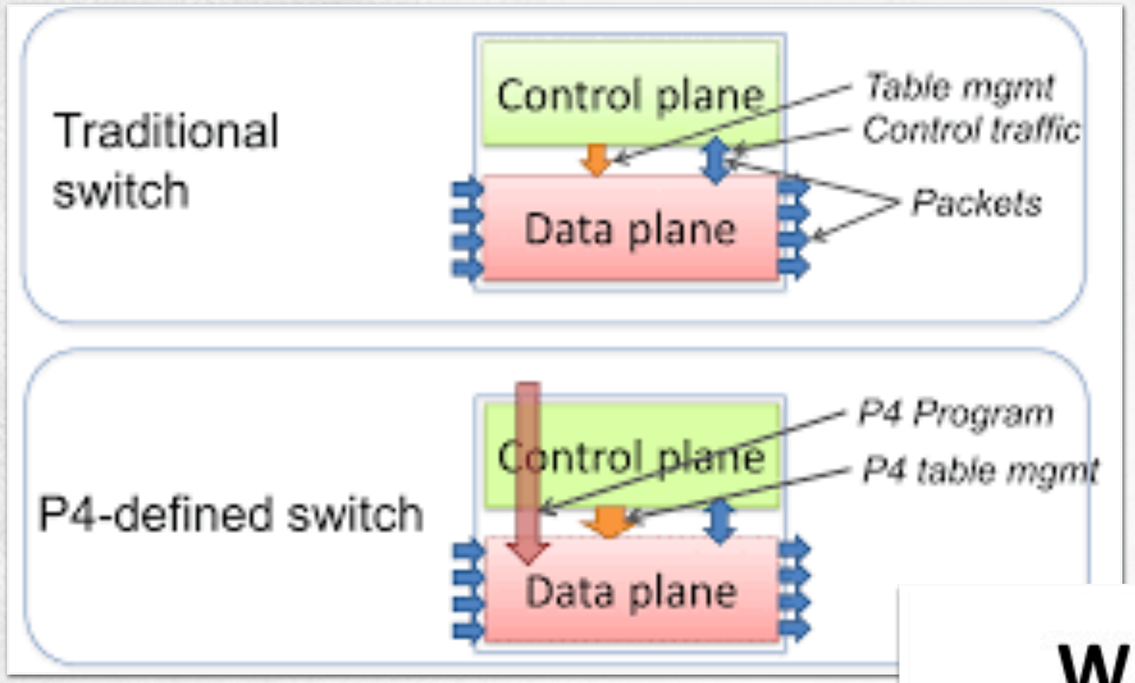
Concept

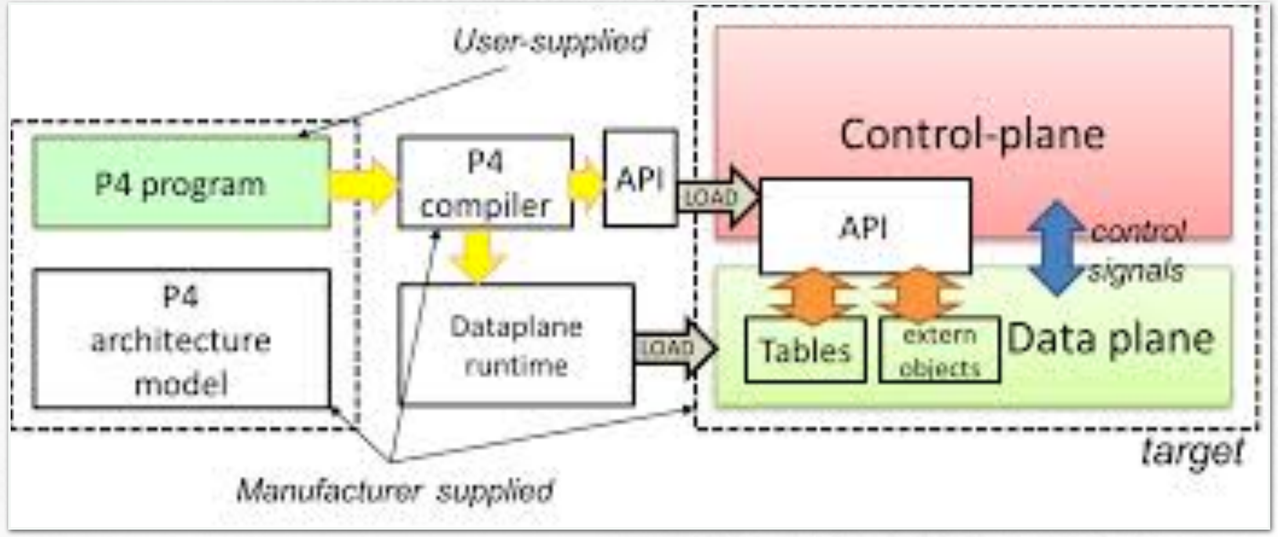
- P4 is a programming language designed to allow programming of packet forwarding planes. In contrast to a general purpose language such as C or Python, P4 is a domain-specific language with a number of constructs optimized around network data forwarding. P4 is an open-source, permissively licensed language 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”.

Reference: https://en.wikipedia.org/wiki/P4_(programming_language)

P4 (programming language)

Implementation





What does a P4 program look like?

```
/* Example: A typical IPv4 routing table */
table ipv4_lpm {
    reads {
        ingress_metadata.vrf     : exact;
        ipv4.dstAddr             : lpm;
    }
    actions {
        nop;
        l3_l2_switch;
        l3_multicast;
        l3_nexthop;
        l3_ecmp;
        l3_drop;
    }
    size : 65536;
}
```

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

Reference: https://www.slideshare.net/opennetsummit/p4-webinarintrodemooct2015chang

Thank you

경희대학교
KYUNG HEE UNIVERSITY