IEEE NETSOFT

24-28 June 2019 // Paris, France

Fluid Network Planes – *An overview of Network Refactoring and Offloading Trends*

Prof. Dr. Christian Esteve Rothenberg (University of Campinas), Brazil

chesteve@dca.fee.unicamp.br

INTRIG INFORMATION & NETWORKING TECHNOLOGIES RESEARCH & INNOVATION GROUP

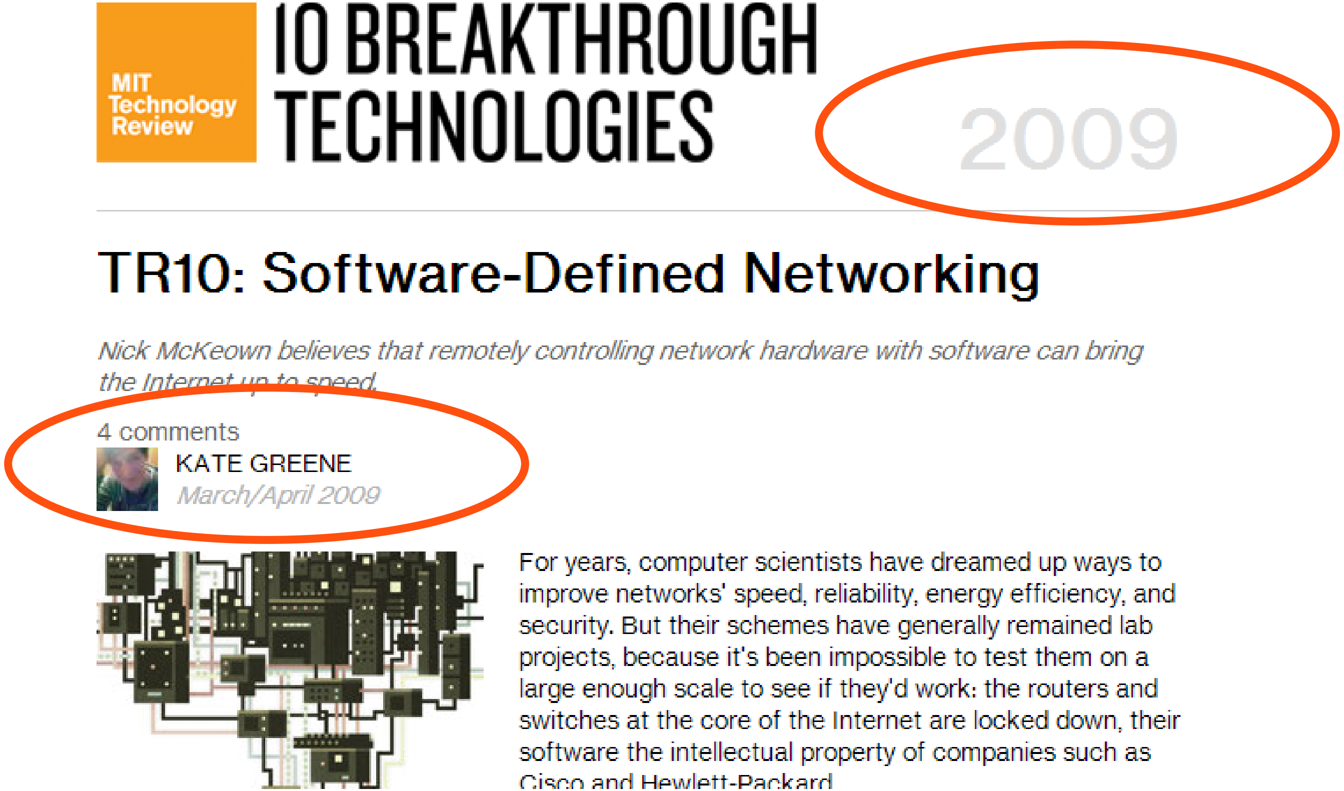
https://intrig.dca.fee.unicamp.br/christian

UNICAMP

Agenda

- A view on 10 years of SDN
- Fluid Network Planes
  - The ‘Concept’
  - Instances

The ‘origins’ of the SDN term



MIT Technology Review

10 BREAKTHROUGH TECHNOLOGIES

2009

TR10: Software-Defined Networking

Nick McKeown believes that remotely controlling network hardware with software can bring the Internet up to speed.

4 comments

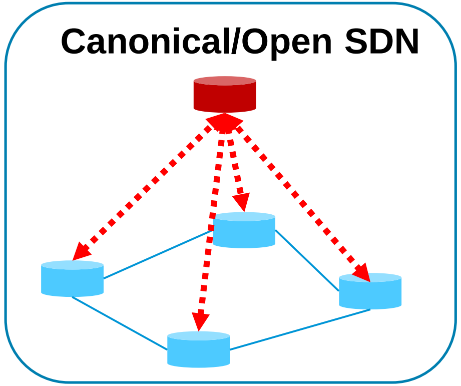
KATE GREENE

March/April 2009

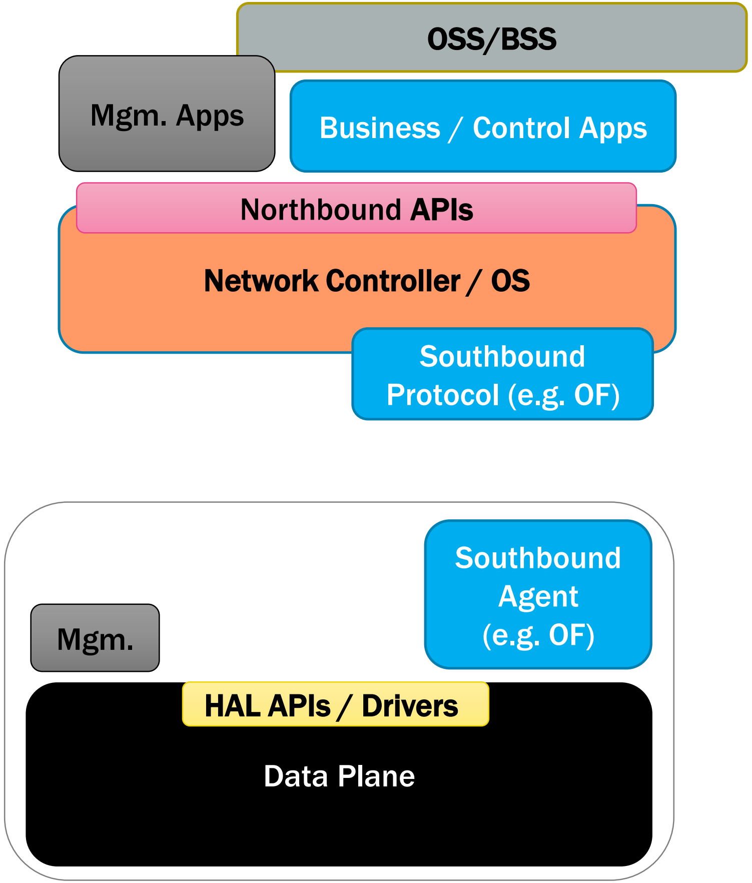
For years, computer scientists have dreamed up ways to improve networks' speed, reliability, energy efficiency, and security. But their schemes have generally remained lab projects, because it's been impossible to test them on a large enough scale to see if they'd work: the routers and switches at the core of the Internet are locked down, their software the intellectual property of companies such as Cisco and Hewlett-Packard

SDN in 2009 - 2010

🟥 Control-plane component(s) 🟦 Data-plane component(s)



Canonical/Open SDN



OSS/BSS

Mgm. Apps

Business / Control Apps

Northbound APIs

Network Controller / OS

Southbound Protocol (e.g. OF)

Mgm.

Southbound Agent (e.g. OF)

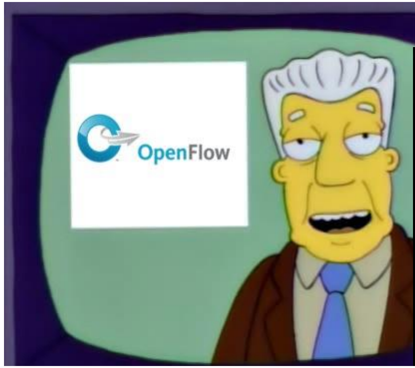
HAL APIs / Drivers

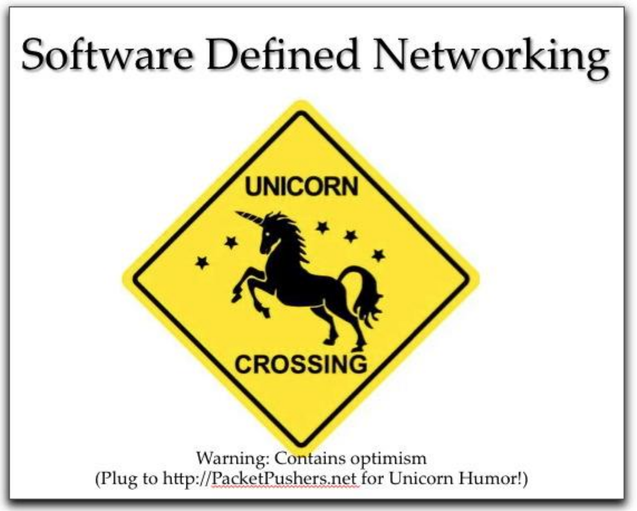
Data Plane

Source: C. Rothenberg (INTRIG/UNICAMP)

SDN in 2011 – 2012

SDN to the rescue!





Software Defined Networking

UNICORN
CROSSING

Warning: Contains optimism
(Plug to http://PacketPushers.net for Unicorn Humor!)

So, what is SDN?

"OpenFlow is SDN, but SDN is not OpenFlow"

(does not say much about SDN) – Networking community

"Don't let humans do machines' work"

(probably right…) – Networking Professional

"Let's call SDN whatever we can ship today"

(aka 'SDN washing') – Vendor X

"SDN is the magic buzzword that will bring us VC funding"

(hmmm… N/A, N/C) – Startup Y

"SDN is the magic that will get my paper/grant accepted"

(maybe, but not at IEEE Netsoft!) – Researcher Z

Headlines

"Google revamps networks with OpenFlow"
—ZDnet

"Prediction: OpenFlow Is Dead by 2014; SDN Reborn in Network Management"
—Mike Fratto, Network Computing

"Will OpenFlow commoditize networks? Impact Cisco margins?"
—Several media publications, Bloggers

"..We share a more pragmatic view, noting Cisco (for example) is likely to view SDN as a TAM expansion opportunity…" *—Deutsche Bank Research note, Wired, April 2012*

"SDN - Still Does Nothing"

"SDN - Smells Dollars Now"

"SDN - Software Defined Not-working"

"Hype around SDN/OpenFlow getting way out of Control. Where have I seen this before…" *—Ethereal mind, Blogger*

"SDN needs a bigger definition"
—Lippis report, 2012

Source: Adapted from A. Retana @ Lacnog'12

SDN in 2013 - 2015

Academia

Vendor A

Vendor B

Vendor C

Start-up 1

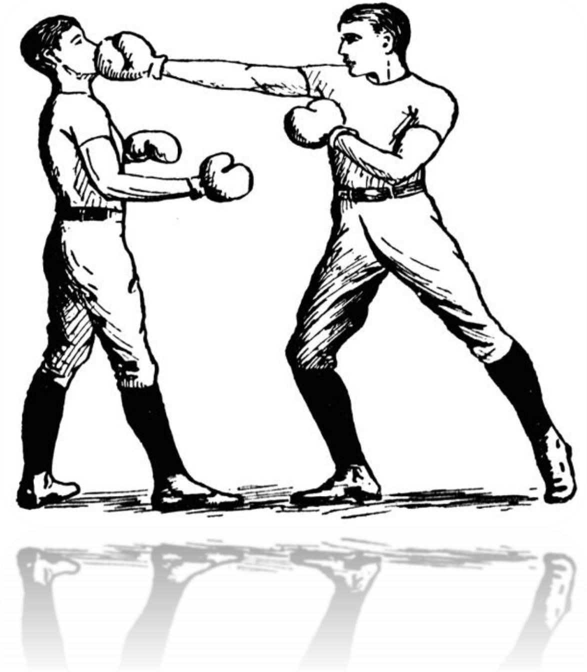
Start-up 2

…

Start-up n

© Robyn Graham Photography

SDN in 2015 – 2019 → Network Softwarization*
(i.e. NFV + SDN + IBN + *xyz*)



Old / Existing

- CLIs & Manual labour
- Closed Source
- Vendor Lead
- Classic Network Appliances (HW)

New / Softwarized

- APIs & Automation
- Open Source
- Customer Lead
- Virtual Network Functions (NFV/SW)

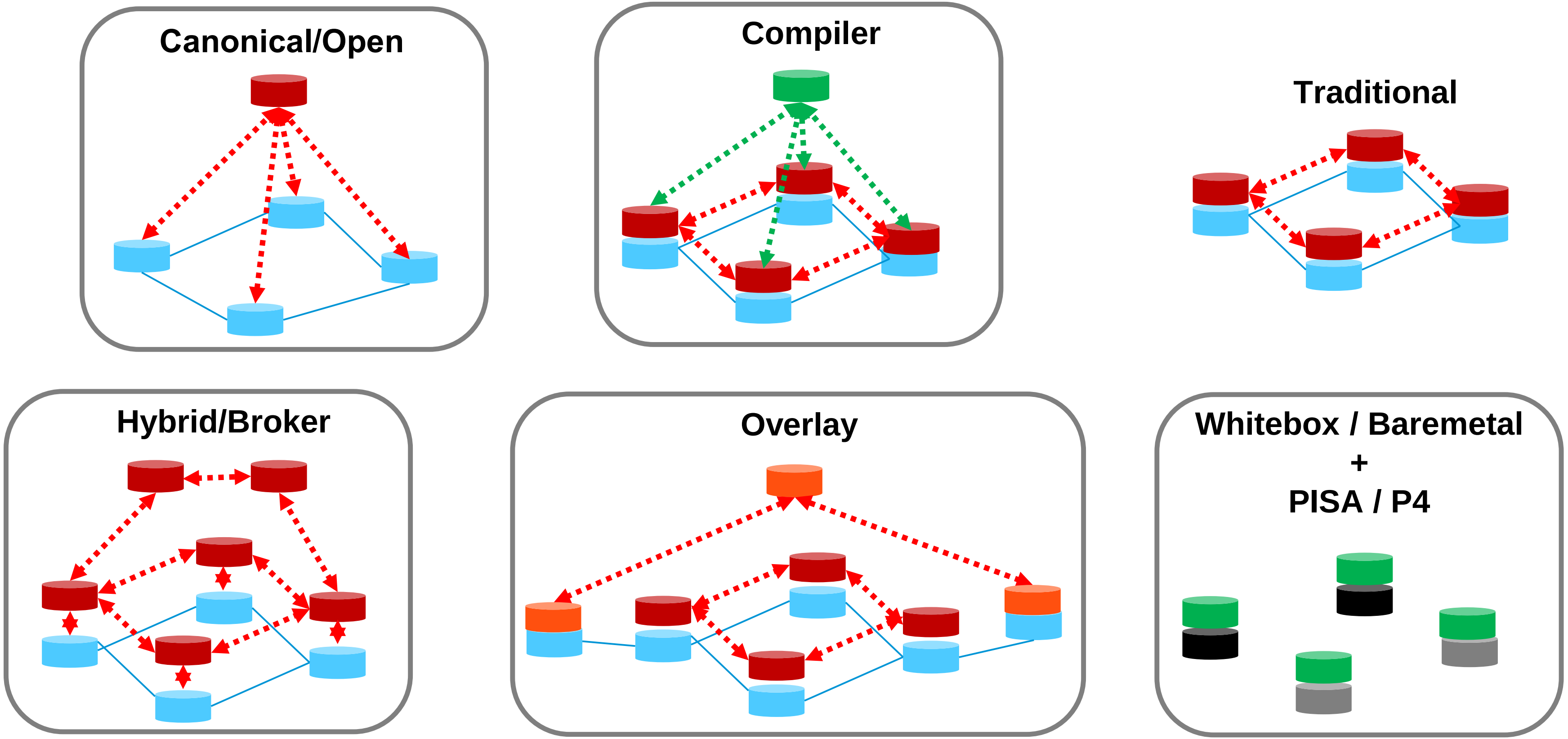
*1st IEEE Network Softwarization 2015 (NetSoft 2015)

Source: Adapted from Kyle Mestery, Next Generation Network Developer Skills

Different Network Softwarization Models

🟥 **Control** plane component(s)

🟦 **Data** plane component(s)



Canonical/Open

Compiler

Traditional

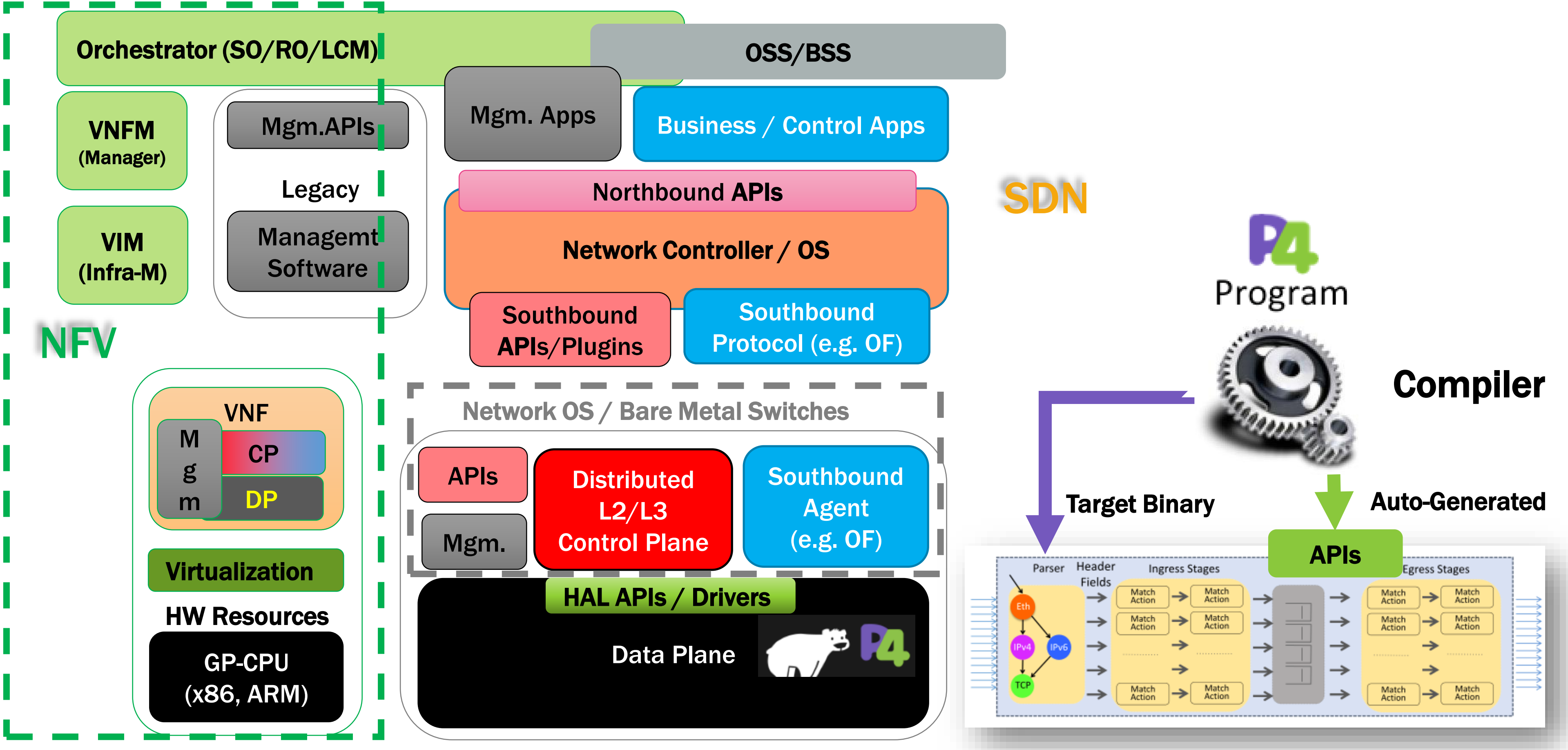
Hybrid/Broker

Overlay

Whitebox / Baremetal
+
PISA / P4

Source: C. Rothenberg (INTRIG/UNICAMP)

Models & Approaches to Program / Refactor the Netsoft Stack



Orchestrator (SO/RO/LCM)

OSS/BSS

VNFM (Manager)

VIM (Infra-M)

Mgm.APIs

Legacy

Managemt Software

Mgm. Apps

Business / Control Apps

Northbound APIs

Network Controller / OS

Southbound APIs/Plugins

Southbound Protocol (e.g. OF)

SDN

NFV

VNF

Mgm

CP

DP

Virtualization

HW Resources

GP-CPU (x86, ARM)

Network OS / Bare Metal Switches

APIs

Mgm.

Distributed L2/L3 Control Plane

Southbound Agent (e.g. OF)

HAL APIs / Drivers

Data Plane

P4 Program

Compiler

Target Binary

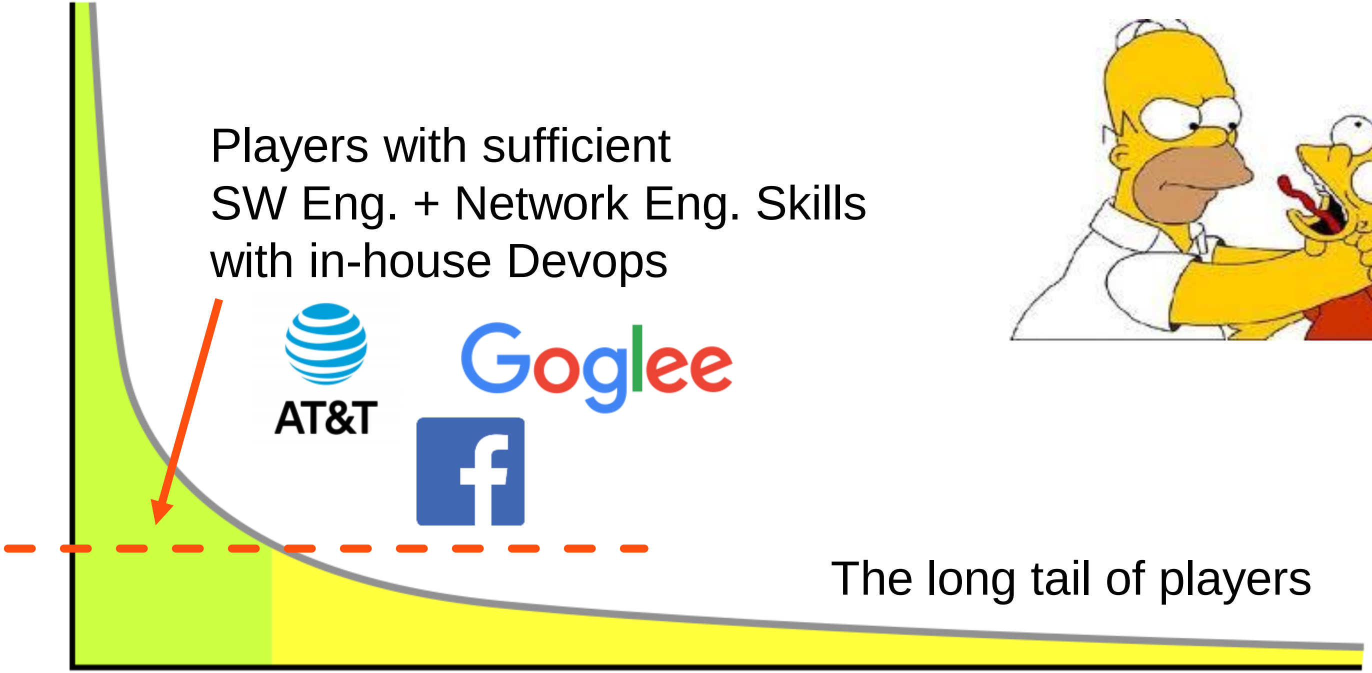
Auto-Generated

APIs

Source: C. Rothenberg (INTRIG/UNICAMP)

Network programmability? By who?

Technical Expertise + Single Throat to Choke



Players with sufficient
SW Eng. + Network Eng. Skills
with in-house Devops

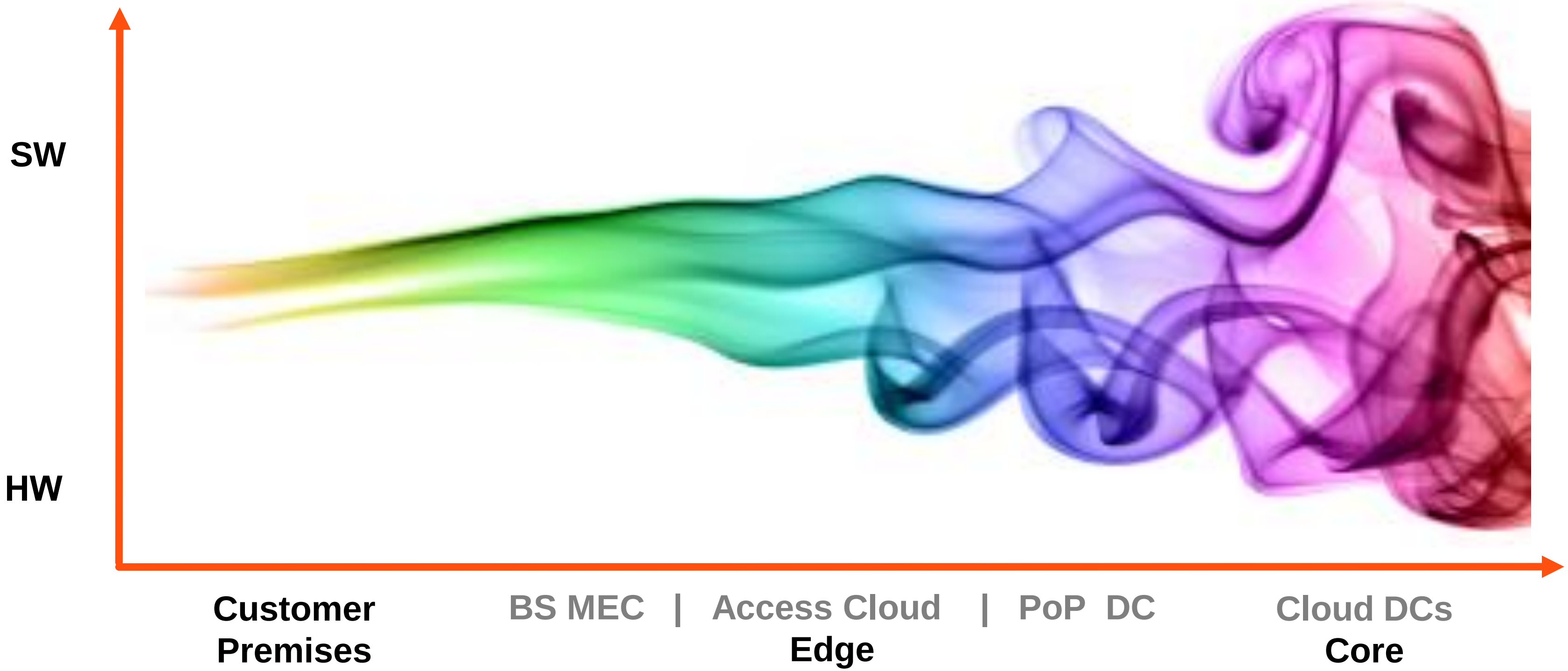
AT&T

Goglee

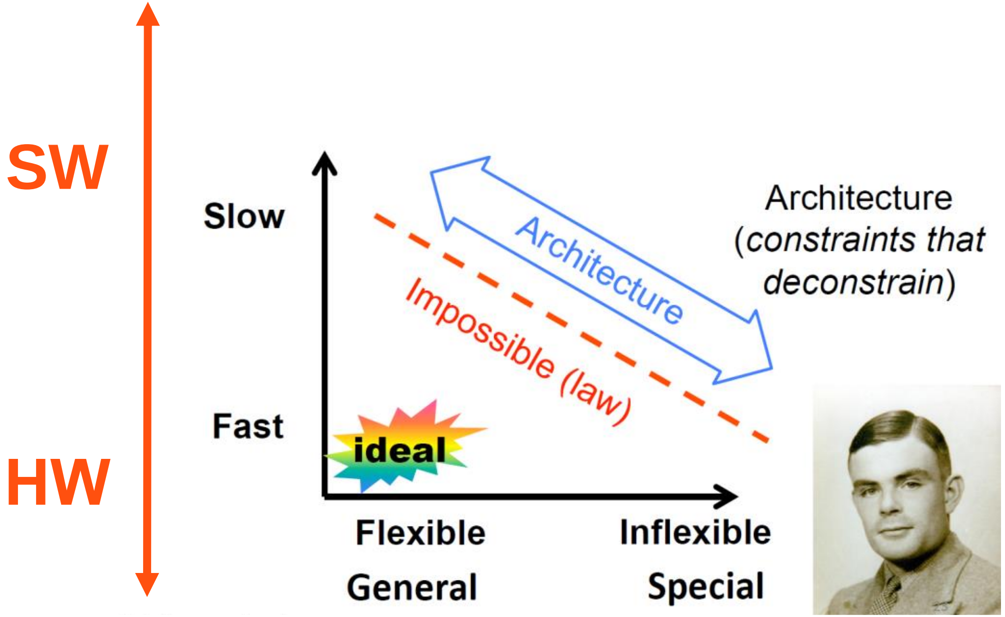
The long tail of players

The **Fluid Networking** landscape

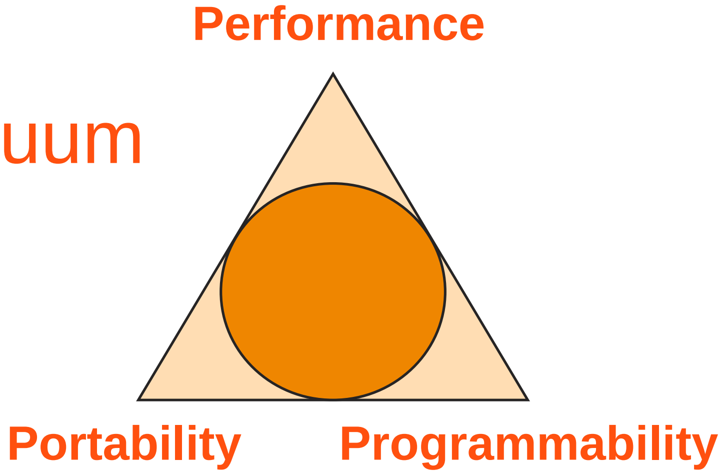
The **Fluid Networking** landscape



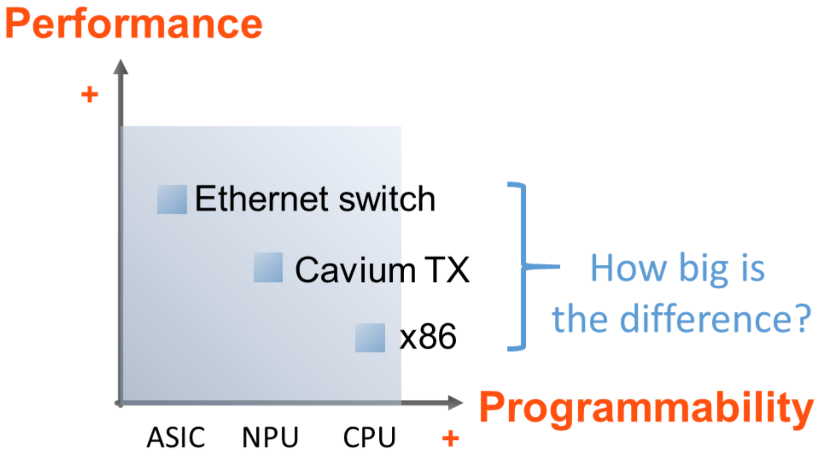
Fluid Networking: HW-SW Continuum



Source: D. Meyer (Courtesy by J. Doyle)



Source: C. Rothenberg. P3 Trade-offs. 2017



Source: G. Pongracz. "Cheap silicon". HotSDN13

Fluid Networking: HW-SW Continuum

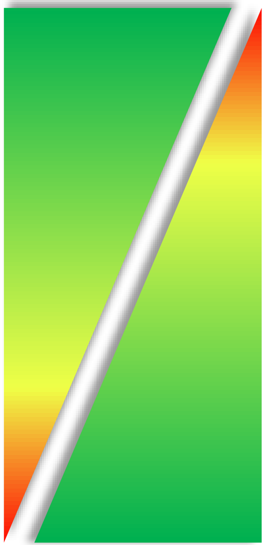
Flexibility*

(programmability + portability)

SW

- **Containers**
- **User space**
- **Kernel space**
- **Drivers, I/O SDKs**
- **General-purpose CPU**
- **HW-accelerated features****
- **FPGA**
- **GPU, TPU,**
- **Programmable NIC, ASIC**
- **Domain Specific Architectures (DSAS) e.g., P4 + PISA**

HW



Performance***

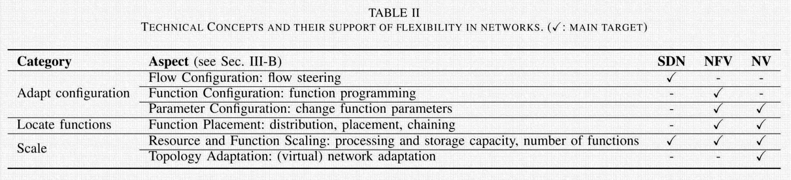


TABLE II
TECHNICAL CONCEPTS AND THEIR SUPPORT OF FLEXIBILITY IN NETWORKS. (✓: MAIN TARGET)

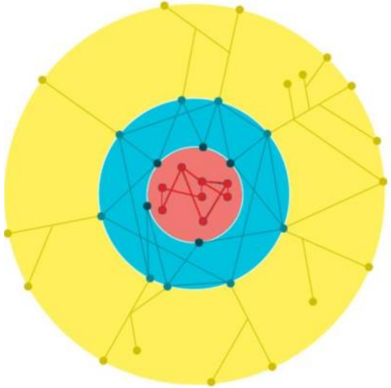
Category	**Aspect** (see Sec. III-B)	**SDN**	**NFV**	**NV**
Adapt configuration	Flow Configuration: flow steering	✓	-	-
	Function Configuration: function programming	-	✓	-
	Parameter Configuration: change function parameters	-	✓	✓
Locate functions	Function Placement: distribution, placement, chaining	-	✓	✓
Scale	Resource and Function Scaling: processing and storage capacity, number of functions	✓	✓	✓
	Topology Adaptation: (virtual) network adaptation	-	-	✓

* M. He et al. **Flexibility in Softwarized Networks: Classifications and Research Challenges**. IEEE Survey & Tutorials, 2019

** Linguaglossa et al. **Survey of Performance Acceleration Techniques for Network Function Virtualization.** Proc. of IEEE, 2019

*** G. Bianchi. **Back to the Future: Hardware-specialized Cloud Networking.** 2019

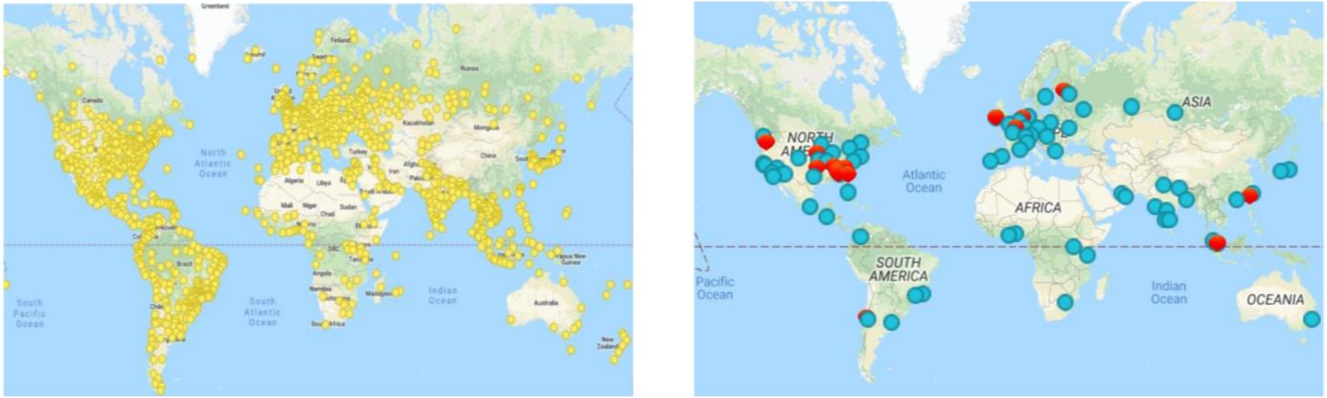
Fluid Networking: Quest for Latency



🔴 15 Data centers

🔵 100 Points of Presence (PoPs)

🟡 1000+ Edge nodes



Source: Google Cloud Infrastructure

Customer Premises

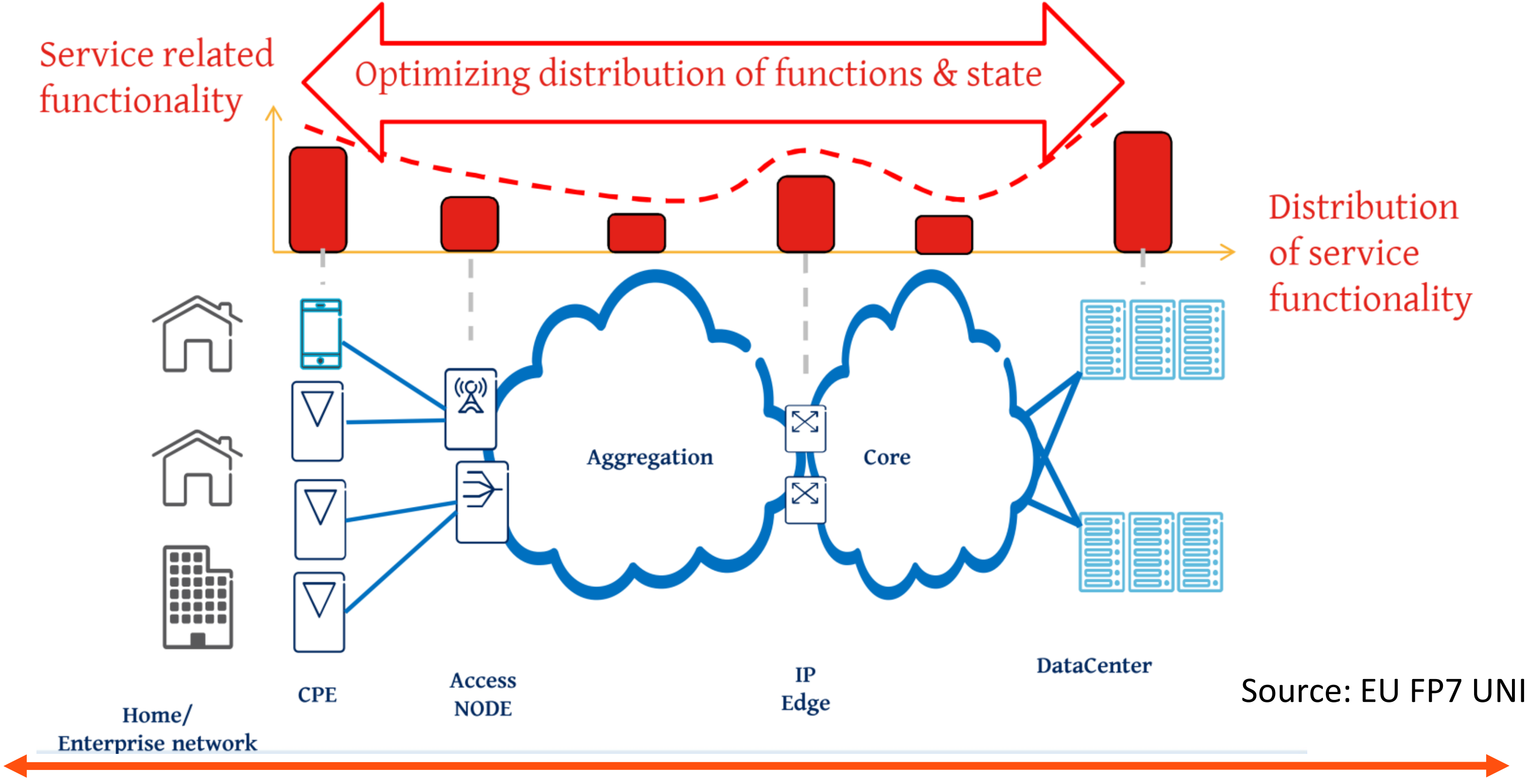
BS MEC | Access Cloud | PoP DC

Edge

Cloud DCs

Core

Fluid Networking: Optimizing the E2E Compute Pool


Service related functionality
Optimizing distribution of functions & state
Distribution of service functionality
Aggregation
Core
CPE
Access NODE
IP Edge
DataCenter
Home/ Enterprise network

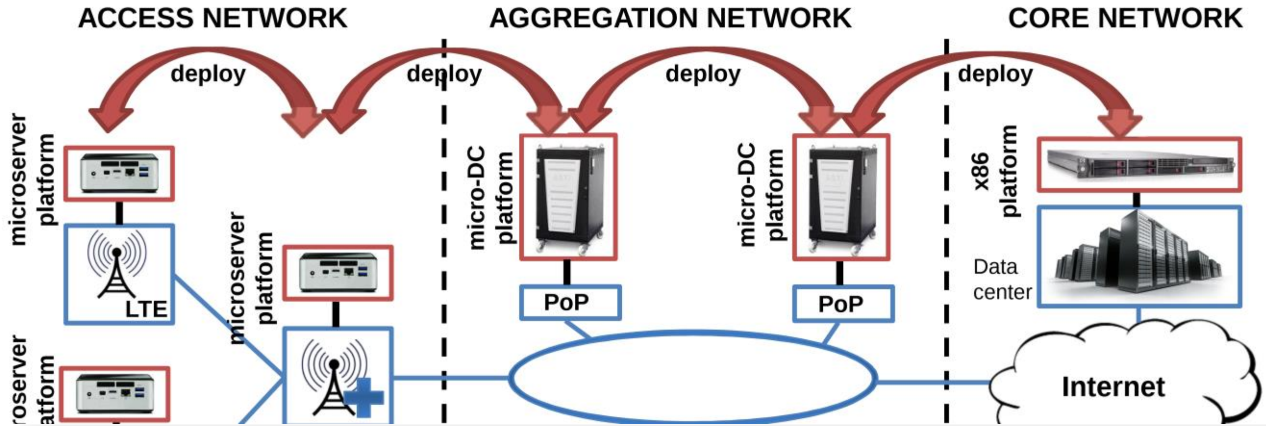
Source: EU FP7 UNIFY

Customer Premises

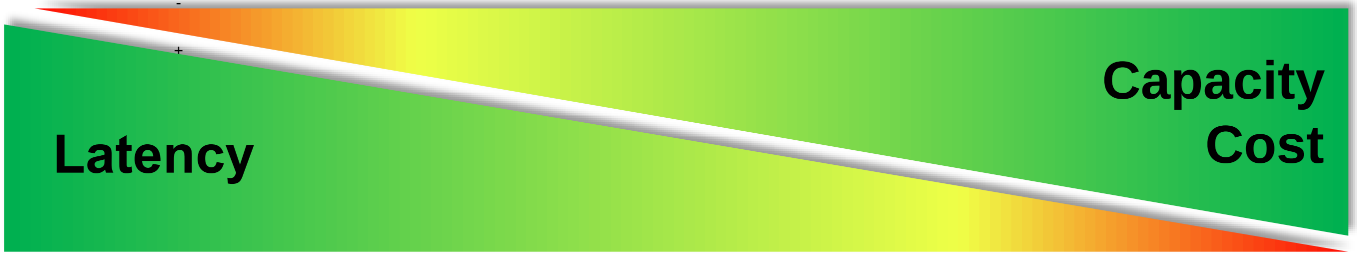
BS MEC | Access Cloud | PoP DC
Edge

Cloud DCs
Core

Fluid Networking: Decoupling functionality / location



Source: EU Superfluidity

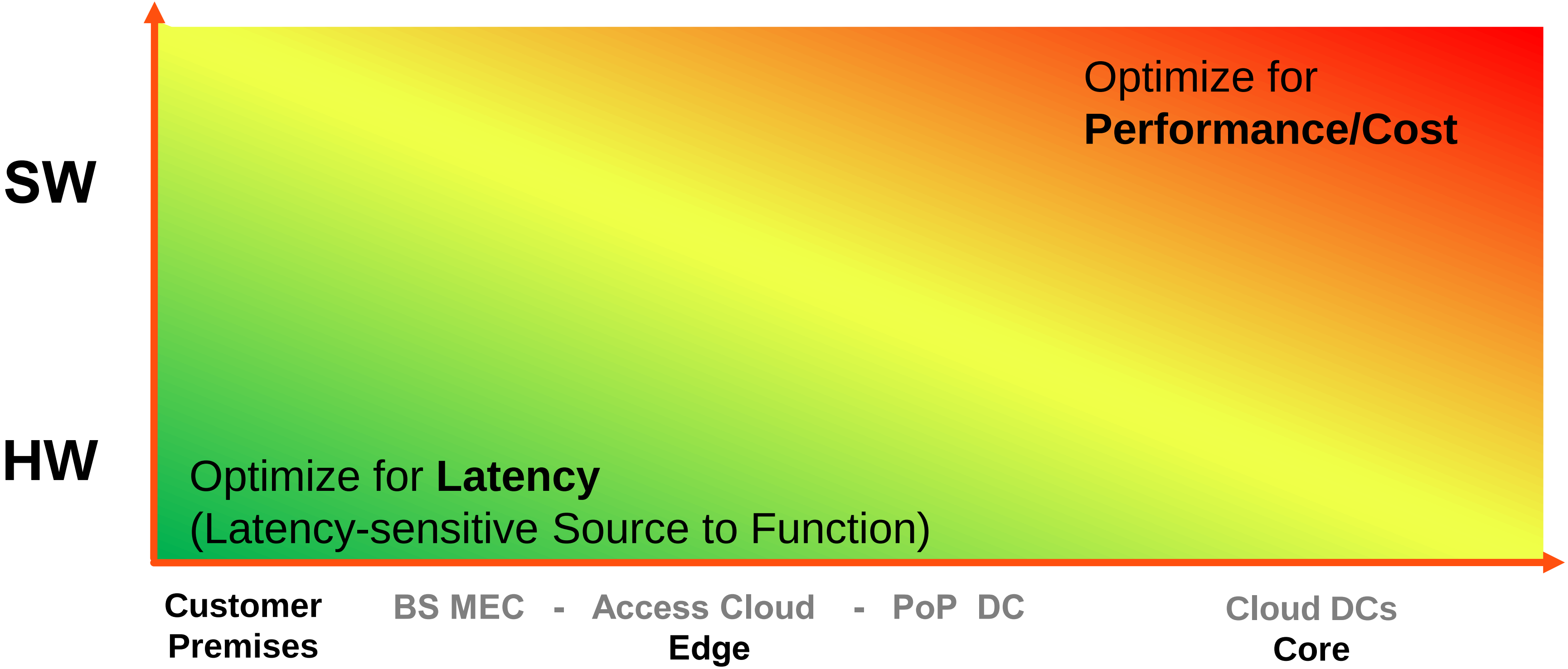


Customer Premises BS MEC - Access Cloud - PoP DC **Edge** Cloud DCs **Core**

The **Fluid Networking** landscape

🟥 **Control** plane component(s)

🟦 **Data** plane component(s)



SW

HW

Optimize for
Performance/Cost

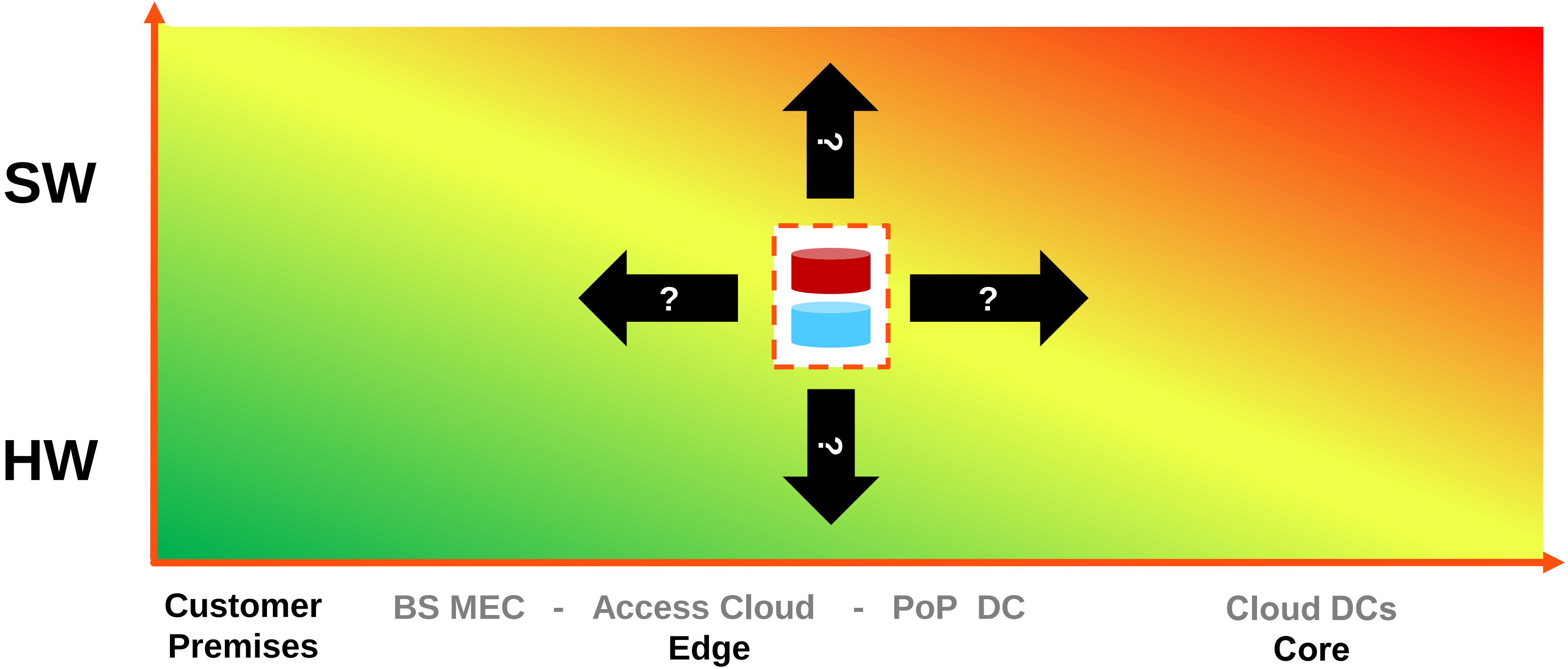
Optimize for **Latency**
(Latency-sensitive Source to Function)

Customer Premises

BS MEC - Access Cloud - PoP DC
Edge

Cloud DCs
Core

The **Fluid Networking** landscape



SW

HW

?

?

?

?

Customer Premises

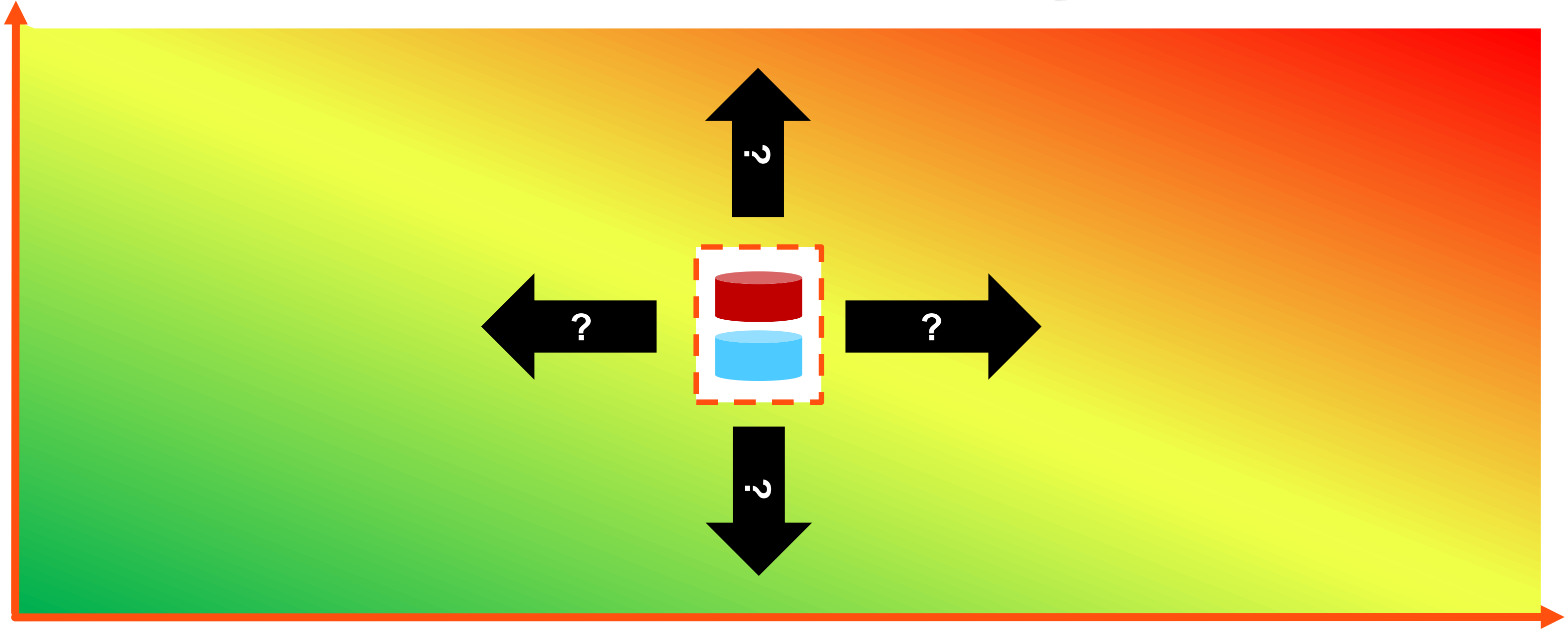
BS MEC - Access Cloud - PoP DC

Edge

Cloud DCs

Core

The **Fluid Networking** landscape



SW

HW

Customer Premises

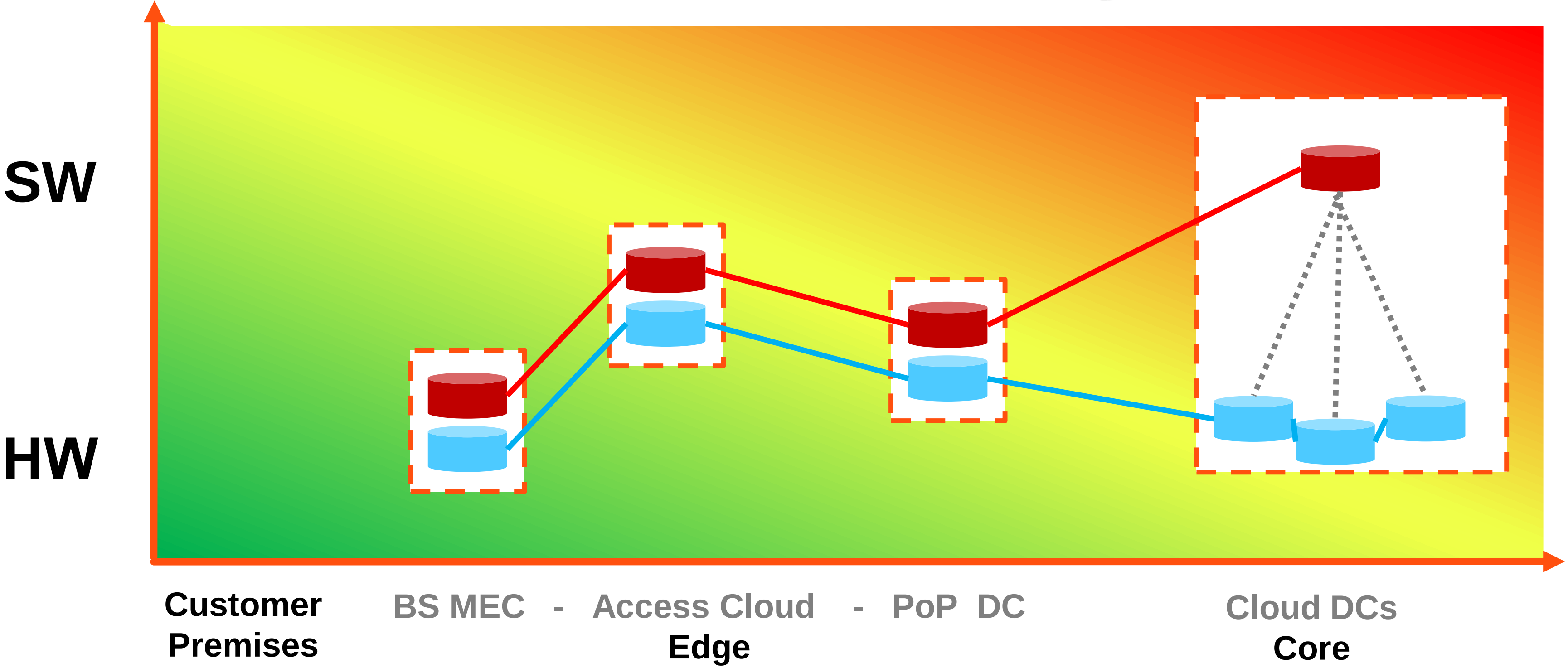
BS MEC - Access Cloud - PoP DC

Edge

Cloud DCs

Core

The **Fluid Networking** landscape



SW

HW

Customer Premises

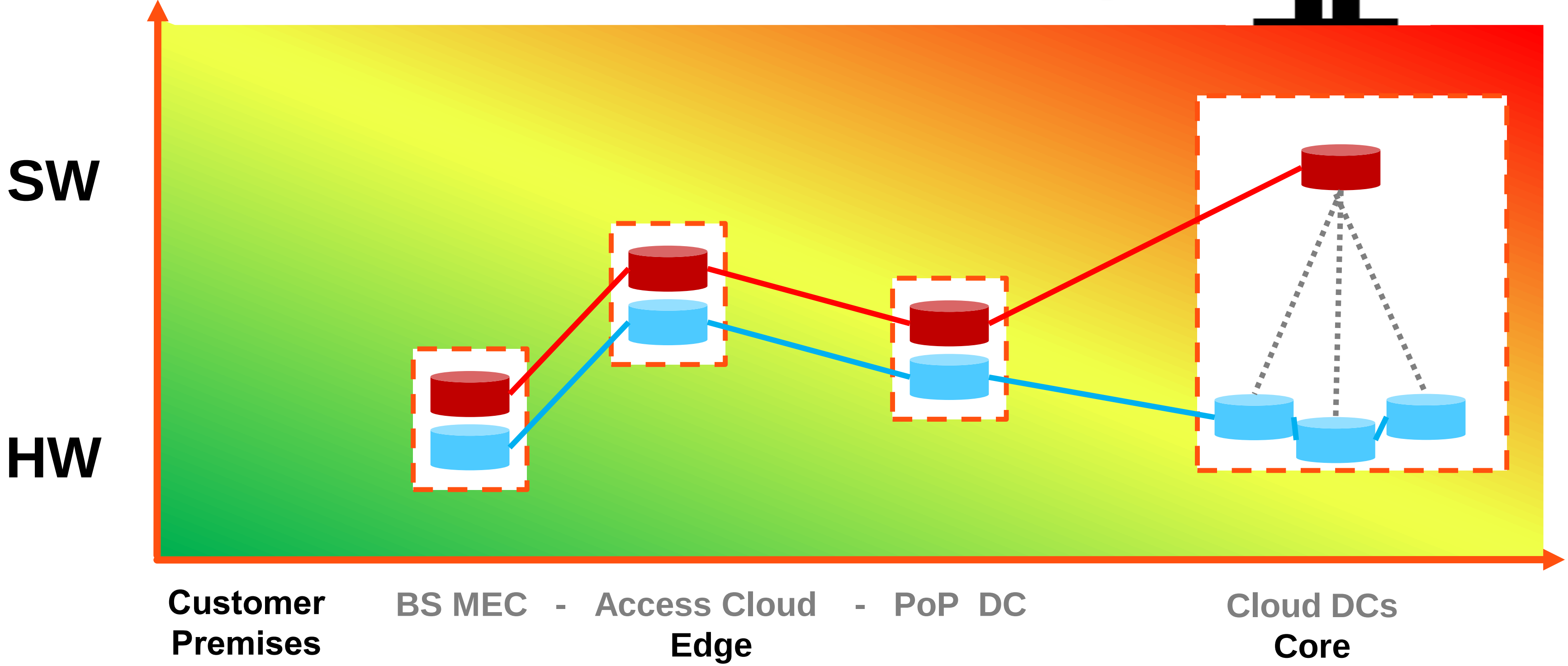
BS MEC - Access Cloud - PoP DC

Edge

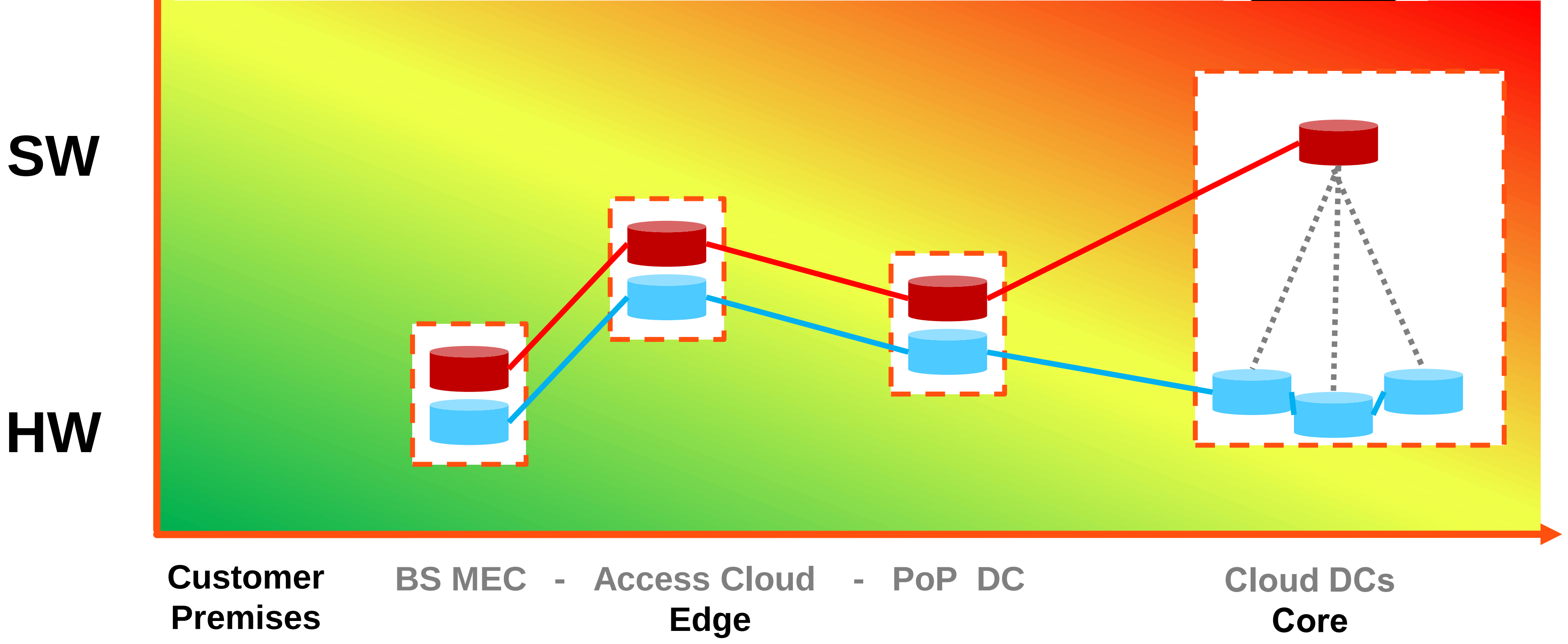
Cloud DCs

Core

The **Fluid Networking** landscape

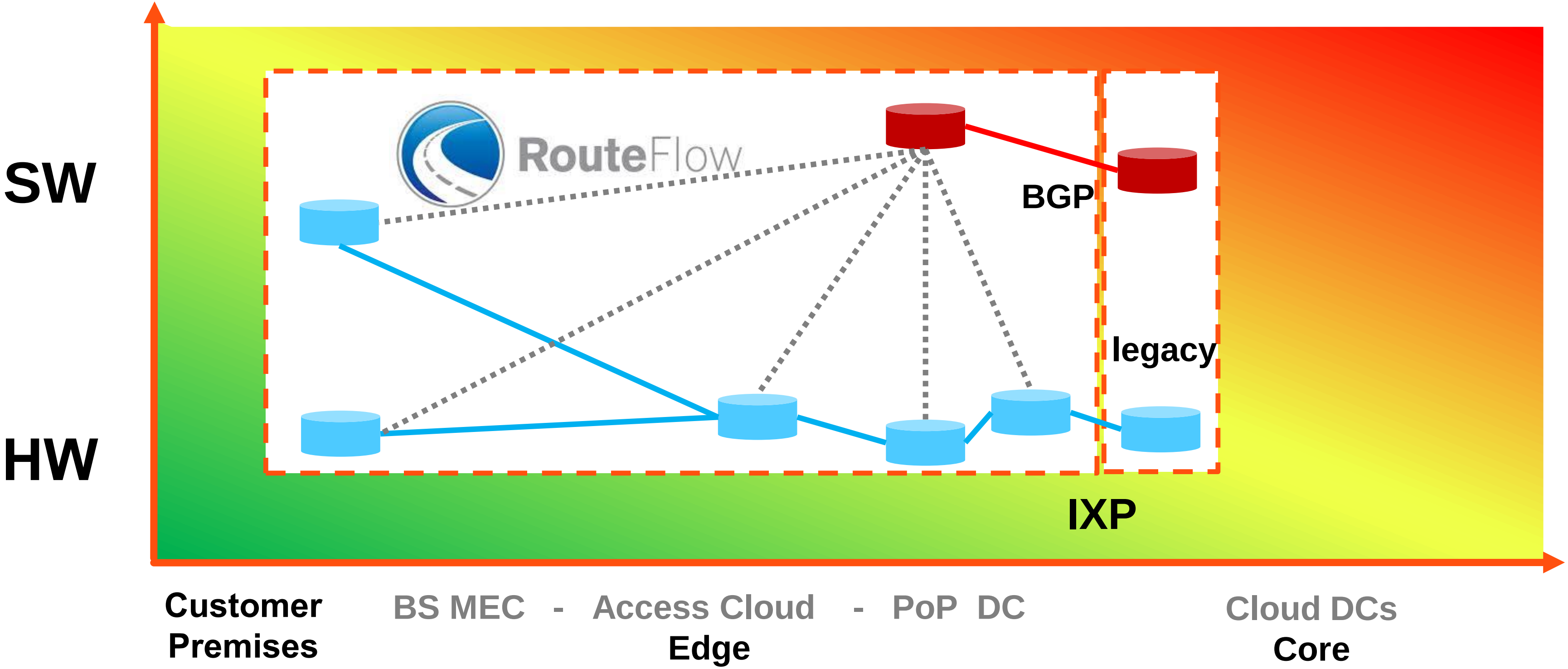


The **Fluid Networking** landscape

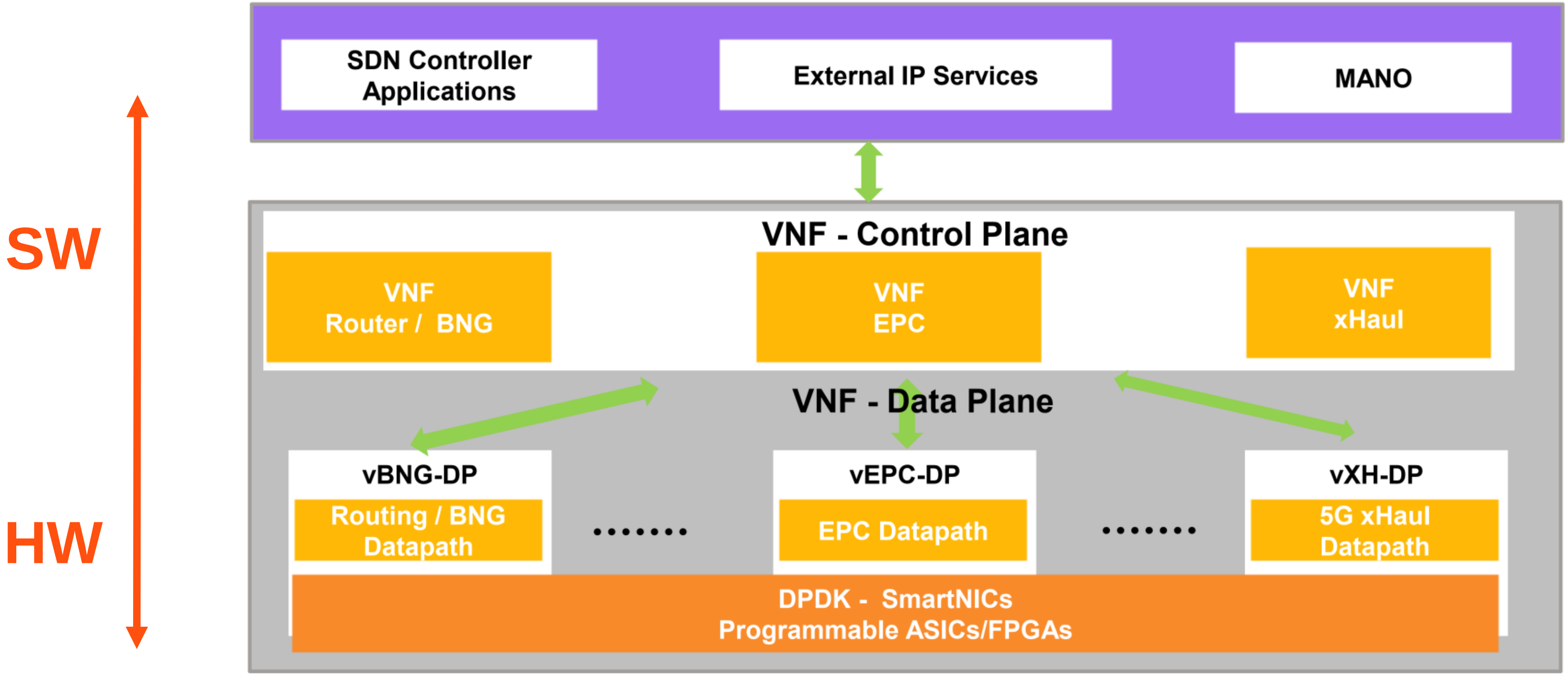


Instances of **Fluid Network Planes**

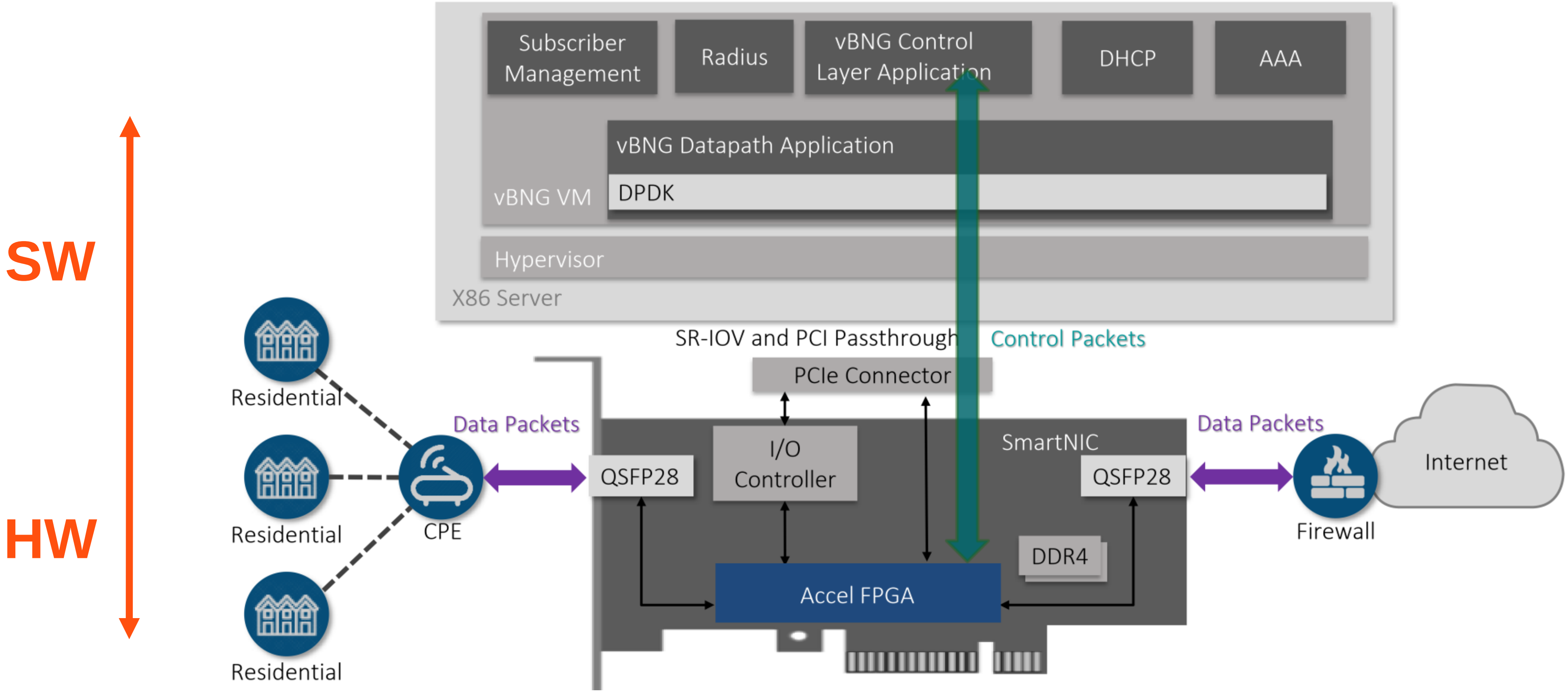
RouteFlow (2010 -)



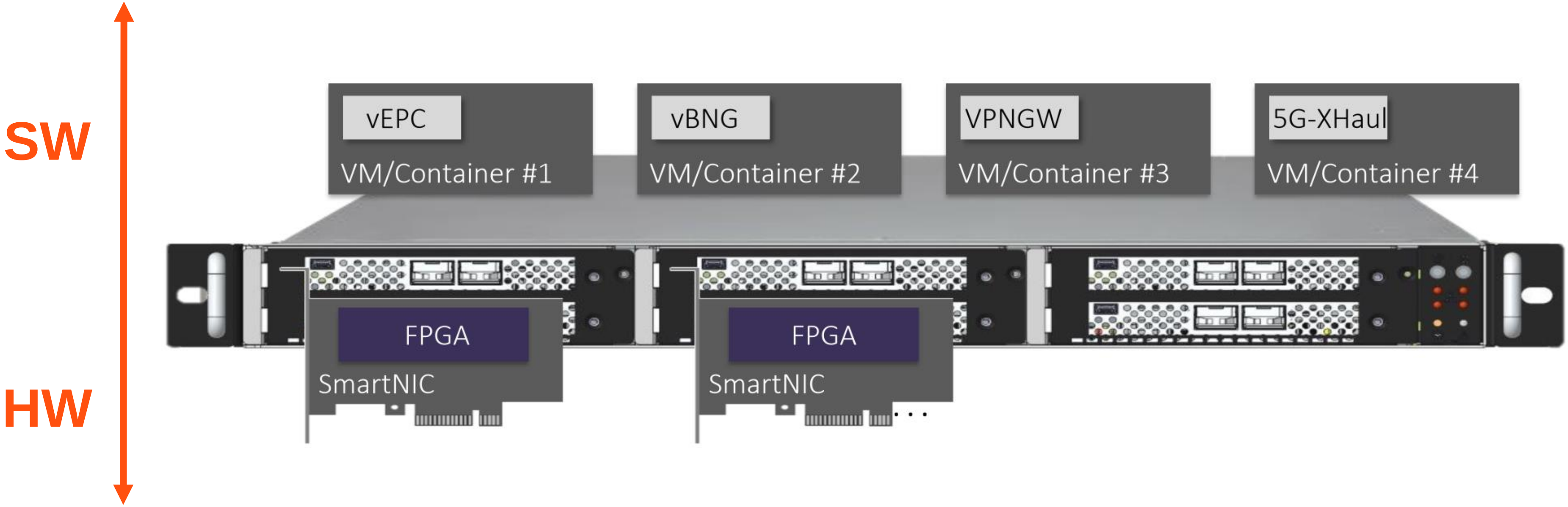
NFV layers of SW, Virtualization and HW platforms



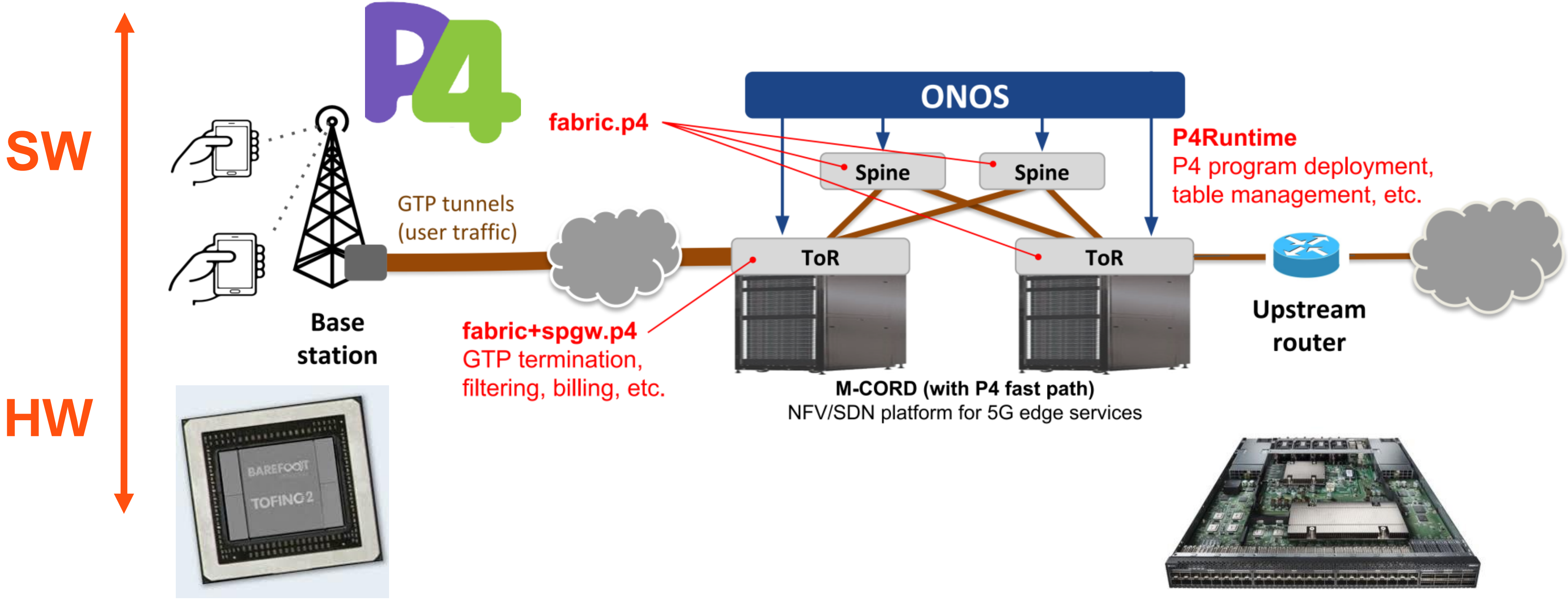
VNF offloading to Hardware



VNF offloading to Hardware



VNF offloading on multi-vendor P4 fabric controlled by ONOS via P4Runtime



SW

HW

ONOS

fabric.p4

Spine

Spine

ToR

ToR

P4Runtime
P4 program deployment, table management, etc.

GTP tunnels
(user traffic)

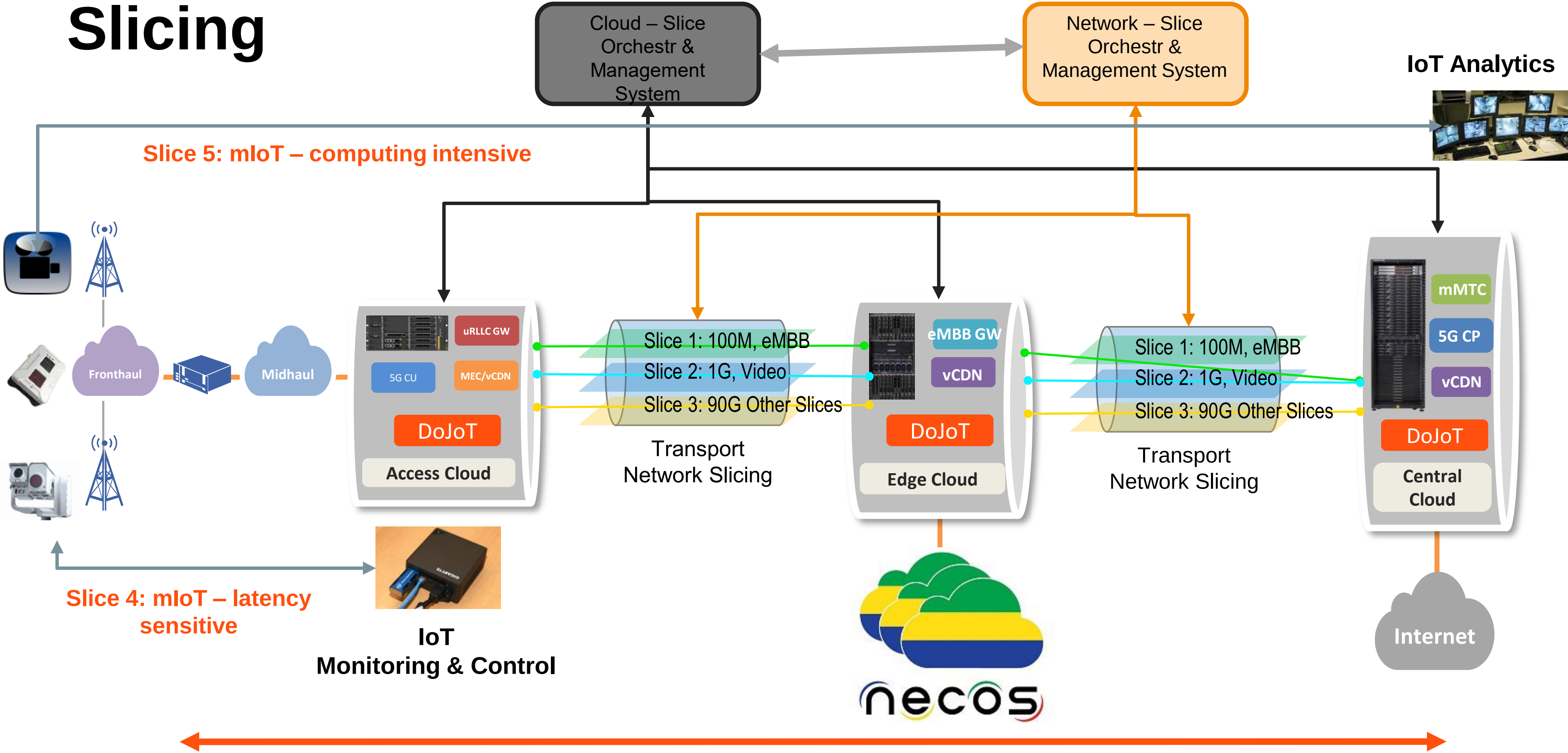
Base station

fabric+spgw.p4
GTP termination, filtering, billing, etc.

Upstream router

M-CORD (with P4 fast path)
NFV/SDN platform for 5G edge services

Slicing

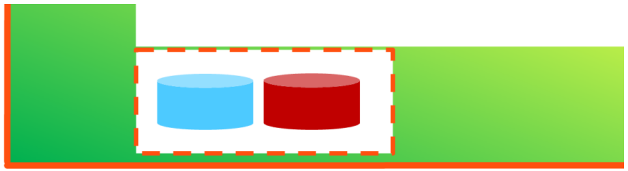


Cloud – Slice Orchestr & Management System
Network – Slice Orchestr & Management System
IoT Analytics
Slice 5: mIoT – computing intensive
Fronthaul
Midhaul
uRLLC GW
5G CU
MEC/vCDN
DoJoT
Access Cloud
Slice 1: 100M, eMBB
Slice 2: 1G, Video
Slice 3: 90G Other Slices
Transport Network Slicing
eMBB GW
vCDN
DoJoT
Edge Cloud
Slice 1: 100M, eMBB
Slice 2: 1G, Video
Slice 3: 90G Other Slices
Transport Network Slicing
mMTC
5G CP
vCDN
DoJoT
Central Cloud
Internet
Slice 4: mIoT – latency sensitive
IoT Monitoring & Control
necos

* D. Ports and J. Nelson. **When Should The Network Be The Computer?**. HotOS'19

IRTF Computation in the Network (COIN)

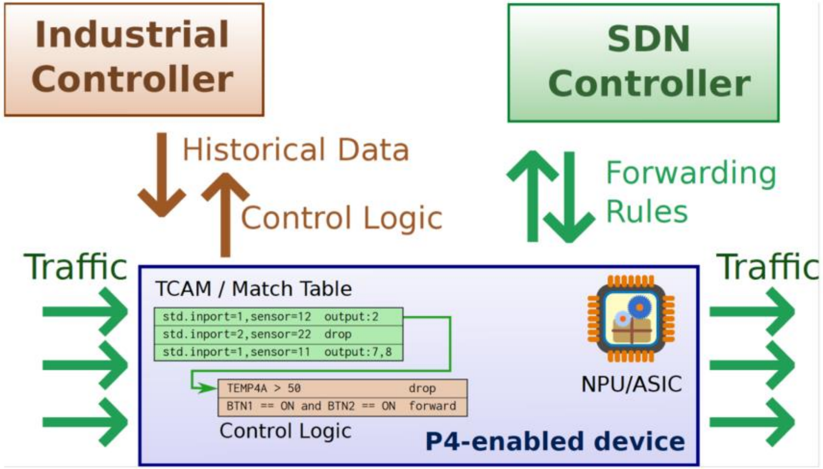
HW



Customer Premises

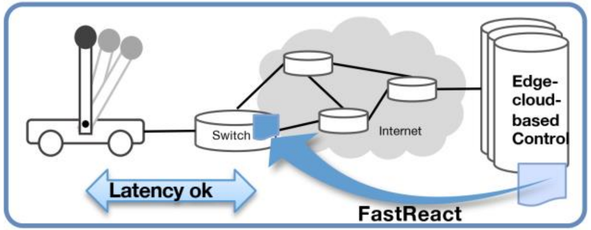
BS MEC - Access Cloud Edge

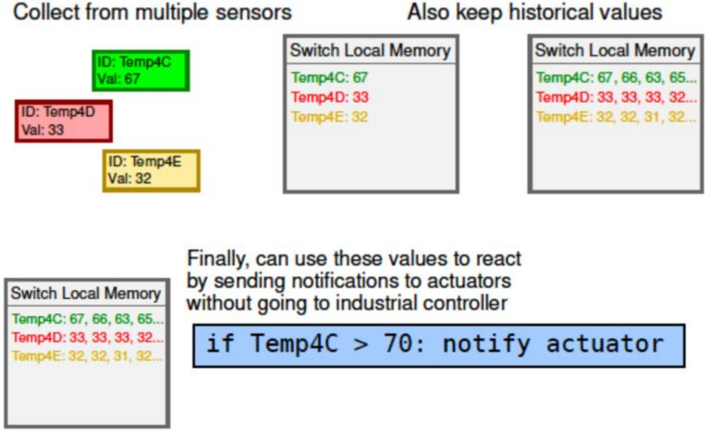
J. Vestin et al. **In-Network Control and Caching for Industrial Control Networks using Programmable Data Planes**. 2018



```
std.inport=1,sensor=12  output:2
std.inport=2,sensor=22  drop
std.inport=1,sensor=11  output:7,8

TEMP4A > 50                  drop
BTN1 == ON and BTN2 == ON    forward
```





Switch Local Memory
Temp4C: 67
Temp4D: 33
Temp4E: 32

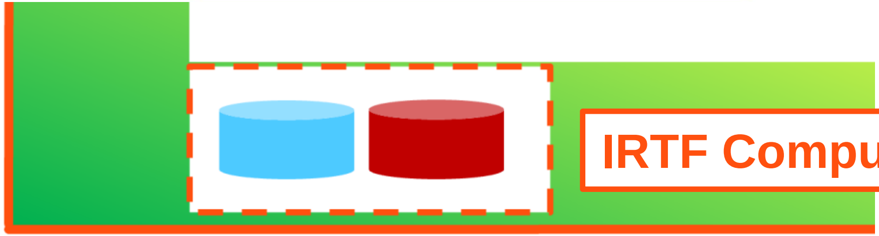
Switch Local Memory
Temp4C: 67, 66, 63, 65...
Temp4D: 33, 33, 33, 32...
Temp4E: 32, 32, 31, 32...

Switch Local Memory
Temp4C: 67, 66, 63, 65...
Temp4D: 33, 33, 33, 32...
Temp4E: 32, 32, 31, 32...

Finally, can use these values to react by sending notifications to actuators without going to industrial controller

```
if Temp4C > 70: notify actuator
```

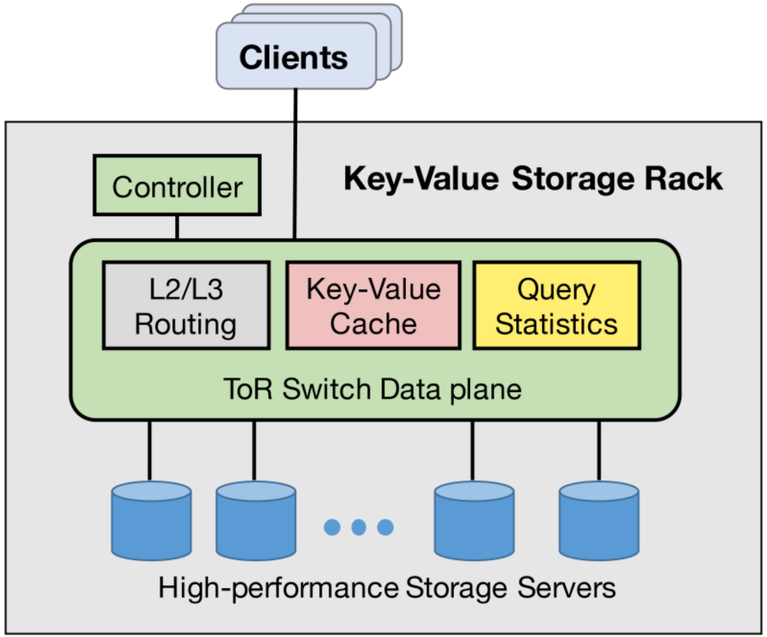
HW



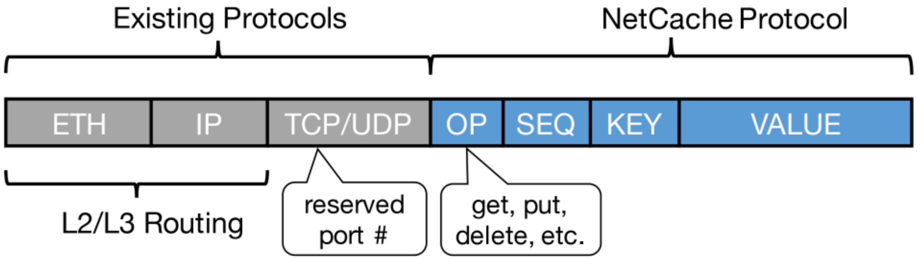
IRTF Computation in the Network (COIN)

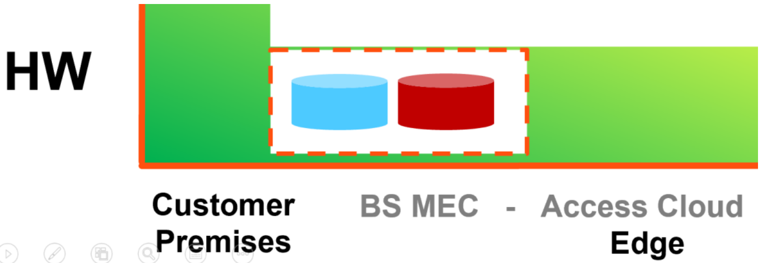
Customer Premises **BS MEC - Access Cloud Edge**

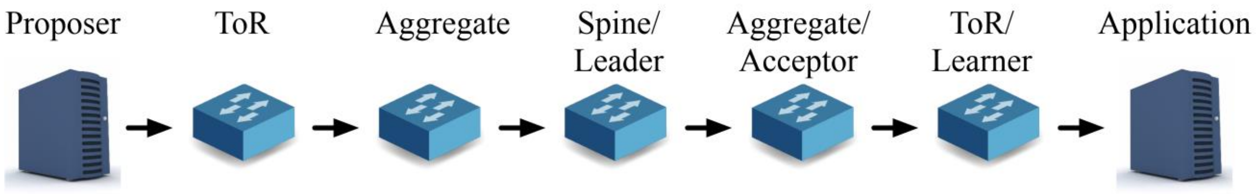
X. Jin et al. **Netcache: Balancing key-value stores with fast in-network caching**. SOSP'17



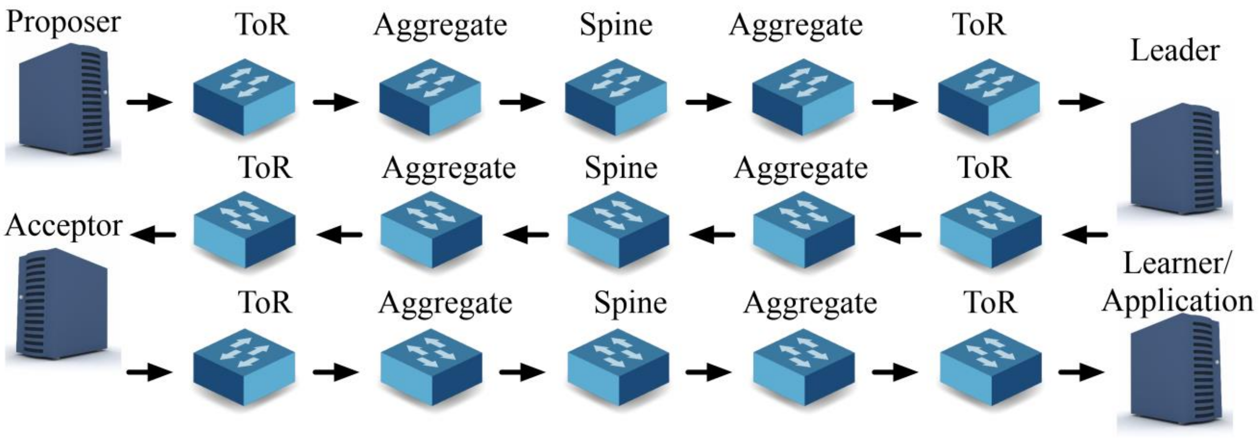
(a) NetCache architecture.



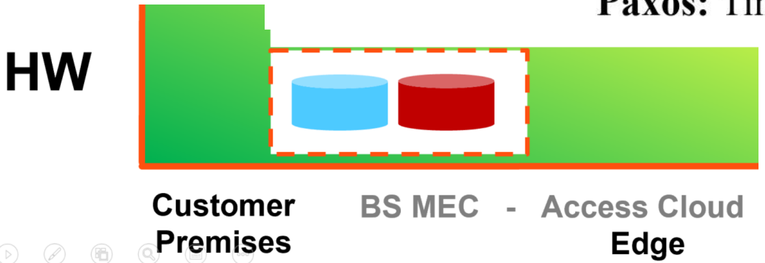




P4xos: Time to reach consensus: RTT/2

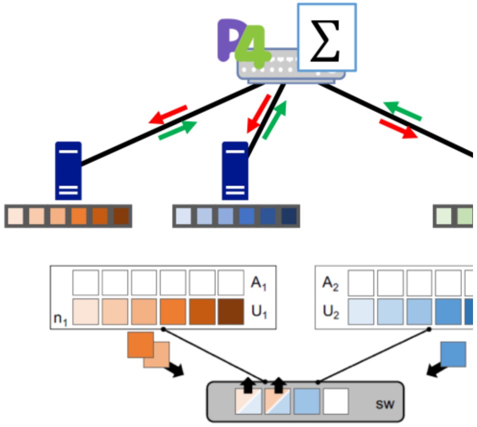


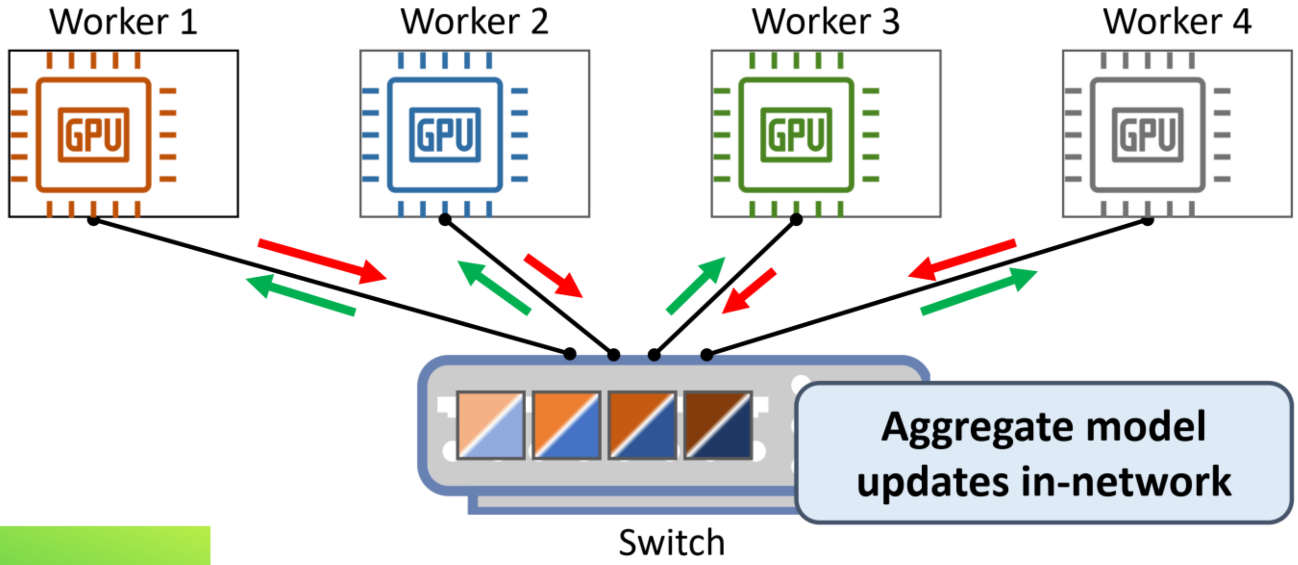
Paxos: Time to reach consensus: RTT x 3/2



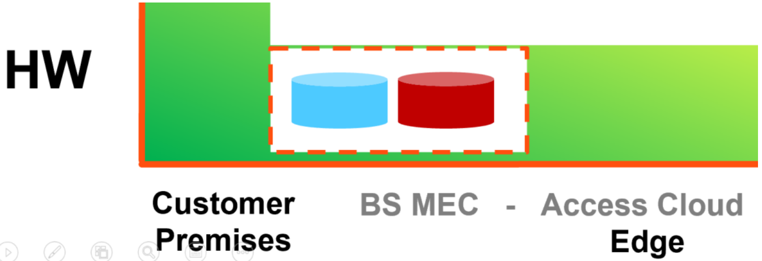
H. Tu Dang et al. **P4xos: Consensus as a Network Service**. 2018

SwitchML: the network is the ML accelerator

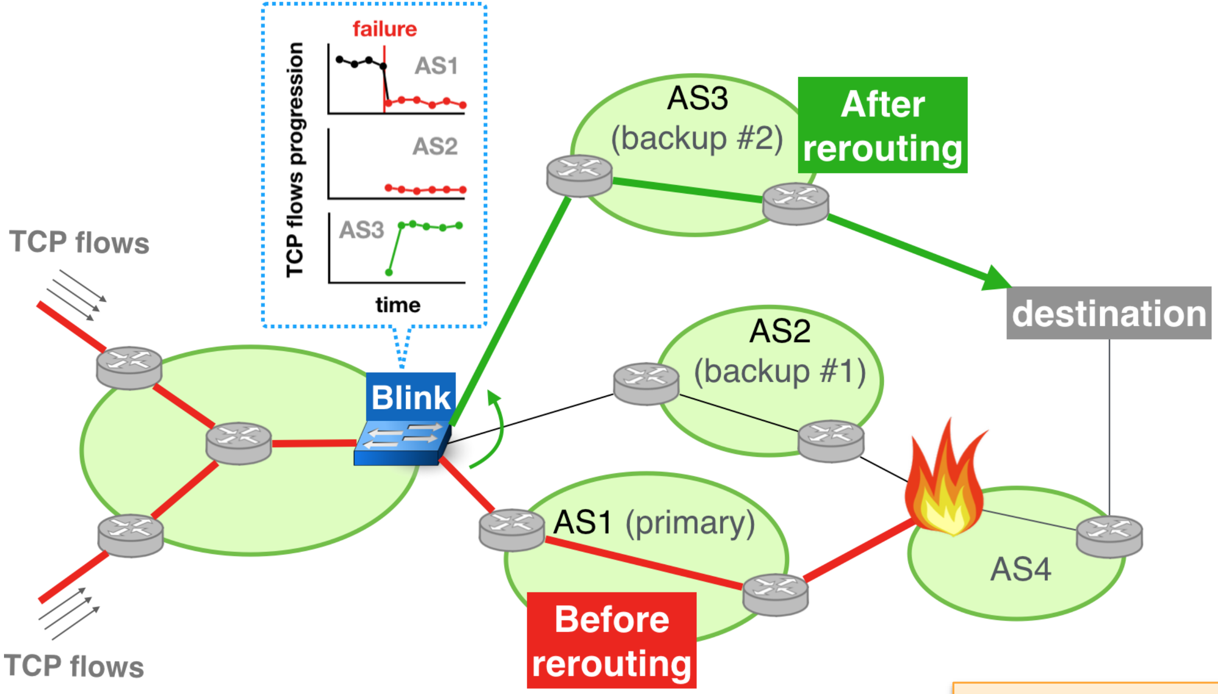


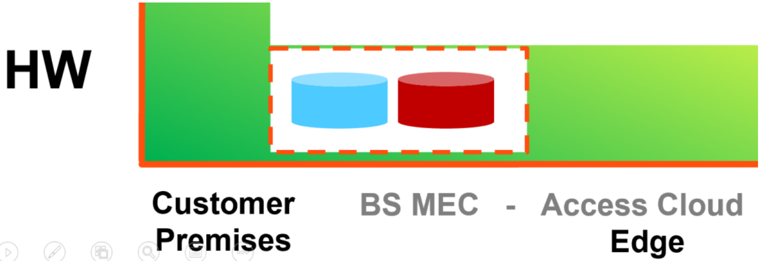


Aggregate model updates in-network



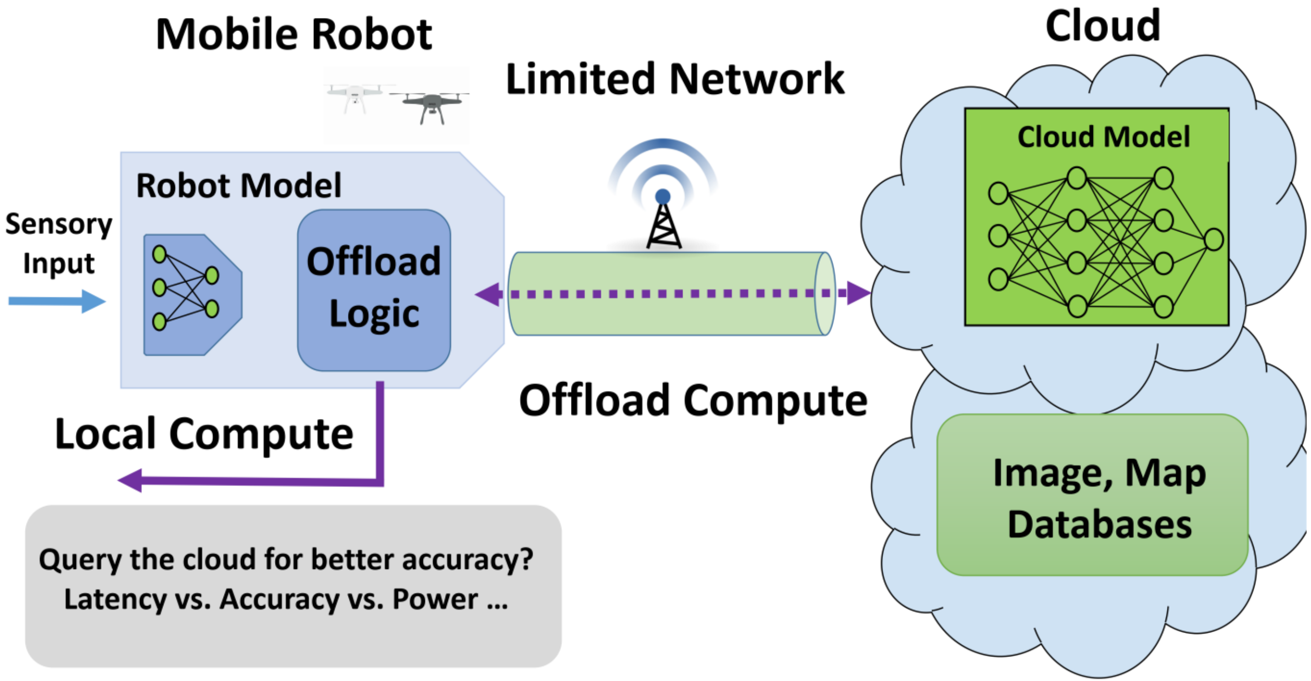
A. Sapio et al. **Scaling Distributed Machine Learning with In-Network Aggregation**. 2019



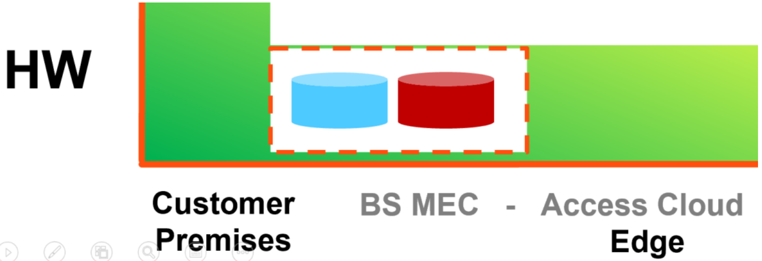


* T. Holterbach et al. **Blink: Fast Connectivity Recovery Entirely in the Data Plane**. NSDI'19

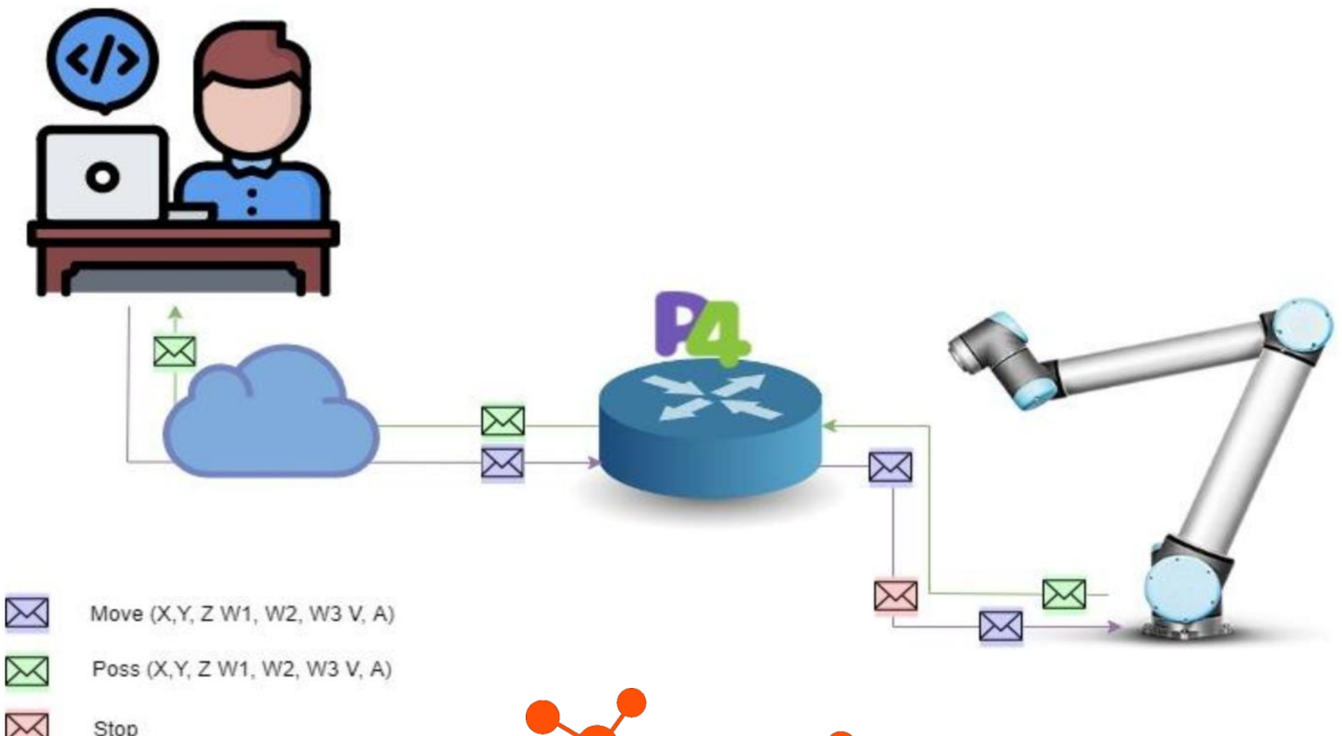
E. Costa Molero et al. **Hardware-Accelerated Network Control Planes**. HotNets'18



Query the cloud for better accuracy?
Latency vs. Accuracy vs. Power …



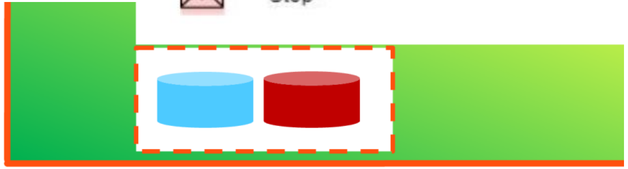
Network Offloading Policies for Cloud Robotics: a Learning-based Approach



🟪 Move (X,Y, Z W1, W2, W3 V, A)

🟩 Poss (X,Y, Z W1, W2, W3 V, A)

🟥 Stop



HW

Customer Premises

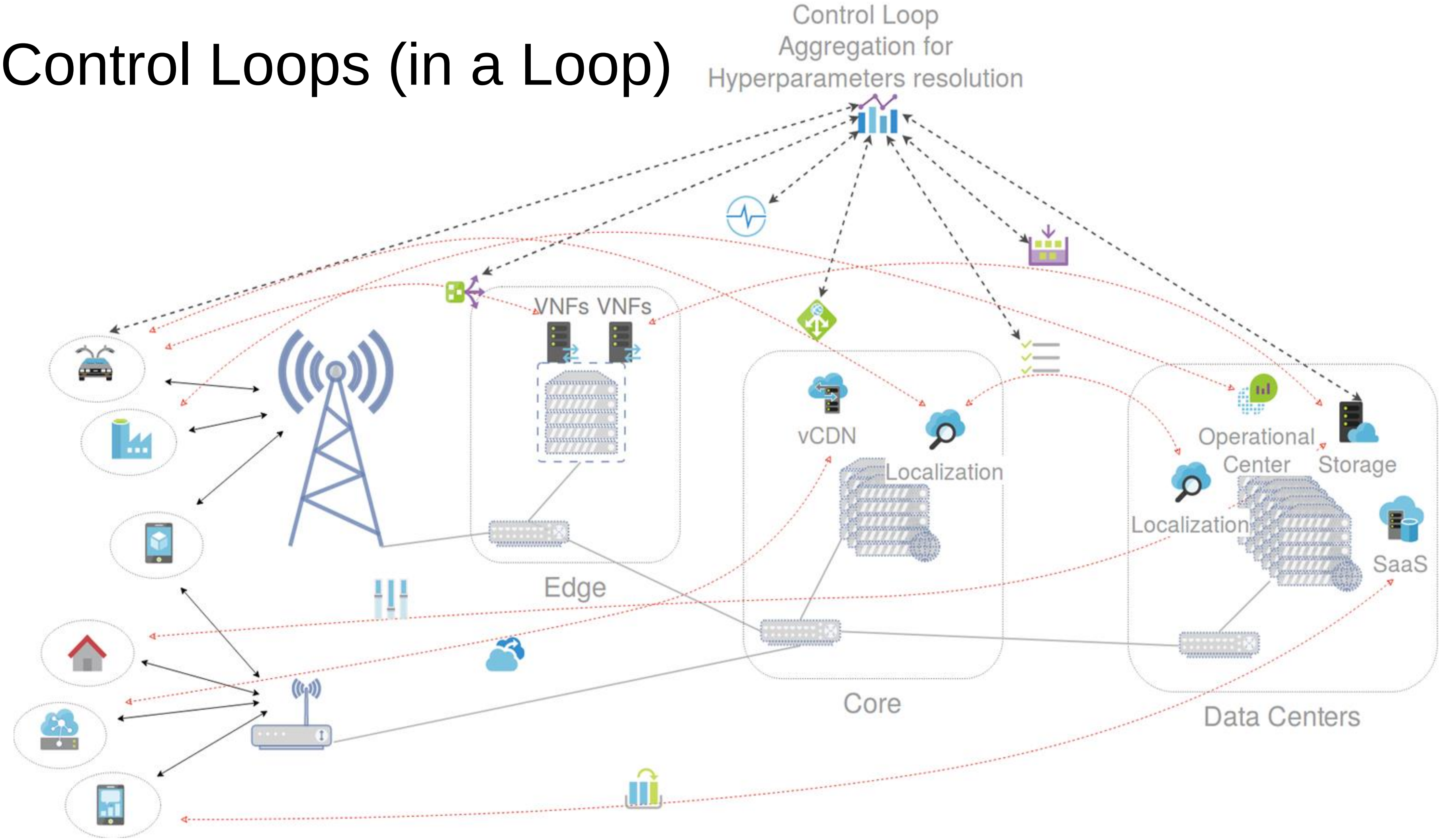
BS MEC - Access Cloud Edge

INTRIG INFORMATION & NETWORKING TECHNOLOGIES RESEARCH & INNOVATION GROUP

ERICSSON

UNICAMP

Control Loops (in a Loop)



References & Credits

- Kaljic, Enio, et al. "A Survey on Data Plane Flexibility and Programmability in Software-Defined Networking." arXiv preprint arXiv:1903.04678 (2019).
- L. Linguaglossa et al., "Survey of Performance Acceleration Techniques for Network Function Virtualization," in Proceedings of the IEEE, vol. 107, no. 4, pp. 746-764, April 2019.
- Edgar Costa Molero, Stefano Vissicchio, and Laurent Vanbever. 2018. Hardware-Accelerated Network Control Planes. In Proceedings of the 17th ACM Workshop on Hot Topics in Networks (HotNets '18). ACM, New York, NY, USA, 120-126.
- Huynh Tu Dang, Marco Canini, Fernando Pedone, and Robert Soulé. “Paxos Made Switch-y.” In ACM SIGCOMM Computer Communication Review (CCR). April 2016.
- JIN, Xin et al. Netcache: Balancing key-value stores with fast in-network caching. In: Proceedings of the 26th Symposium on Operating Systems Principles. ACM, 2017
- Yuta Tokusashi, Huynh Tu Dang, Fernando Pedone, Robert Soulé, and Noa Zilberman. “The Case For In-Network Computing On Demand.” In European Conference on Computer Systems (EuroSYS). March 2019.

References & Credits

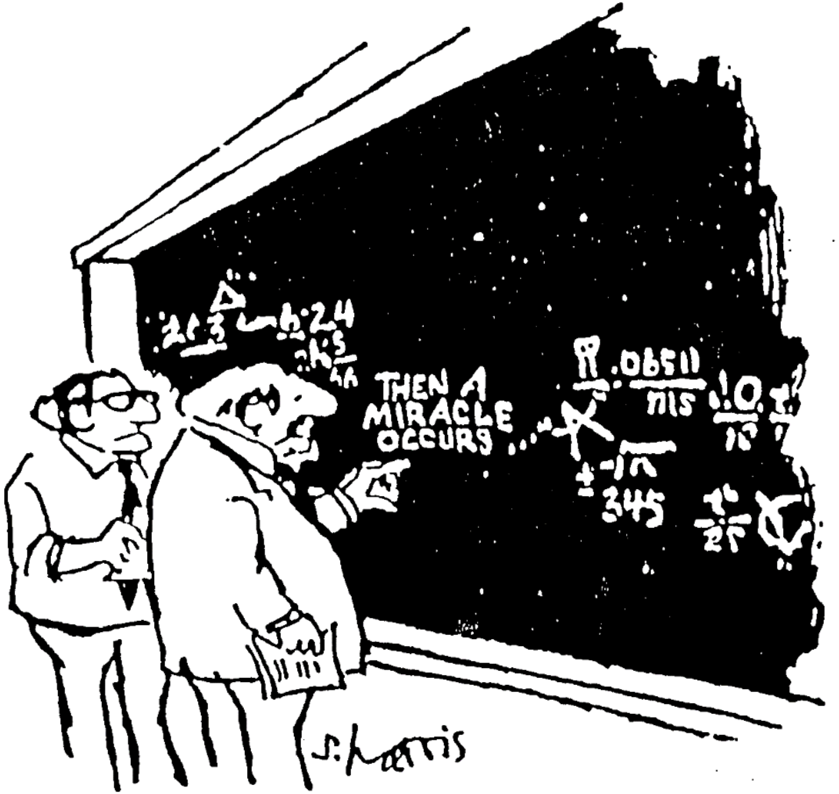
- D. Ports and J. Nelson. When Should The Network Be The Computer?. In Proceedings of the Workshop on Hot Topics in Operating Systems (HotOS '19)
- Atul Adya, Robert Grandl, Daniel Myers, and Henry Qin. 2019. Fast key-value stores: An idea whose time has come and gone. In Proceedings of the Workshop on Hot Topics in Operating Systems (HotOS '19)
- Theophilus A. Benson. 2019. In-Network Compute: Considered Armed and Dangerous. In Proceedings of the Workshop on Hot Topics in Operating Systems (HotOS '19)
- Theo Jepsen, Daniel Alvarez, Nate Foster, Changhoon Kim, Jeongkeun Lee, Masoud Moshref, and Robert Soulé. 2019. Fast String Searching on PISA. In Proceedings of the 2019 ACM Symposium on SDN Research (SOSR '19)
- Thomas Holterbach, Edgar Costa Molero, Maria Apostolaki, Alberto Dainotti, Stefano Vissicchio, Laurent Vanbever. Blink: Fast Connectivity Recovery Entirely in the Data Plane. NSDI 2019.
- A. Sapio, M. Canini, C.-Y. Ho, J. Nelson, P. Kalnis, C. Kim, A. Krishnamurthy, M. Moshref, D. R. K. Ports, P. Richtarik. Scaling Distributed Machine Learning with In-Network Aggregation. KAUST technical report, Feb 2019

References & Credits

- A. Sapio et al. Scaling Distributed Machine Learning with In-Network Aggregation. 2019.
- Huynh Tu Dang, Pietro Bressana, Han Wang, Ki Suh Lee, Hakim Weatherspoon, Marco Canini, Fernando Pedone, Noa Zilberman, Robert Soulé, "P4xos: Consensus as a Network Service", Tech Report, University of Lugano 2018/01, May 2018
- H. Tu Dang et al. P4xos: Consensus as a Network Service. 2018
- Raphael Rosa and Christian Esteve Rothenberg. "The Pandora of Network Slicing: A Multi-Criteria Analysis". ETT. 2019
- J. Vestin, A. Kassler, J. Åkerberg, FastReact: In-Network Control and Caching for Industrial Control Networks using Programmable Data Planes. In 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation September 4th - 7th, 2018, Torino, Italy.

Merci!

Questions?



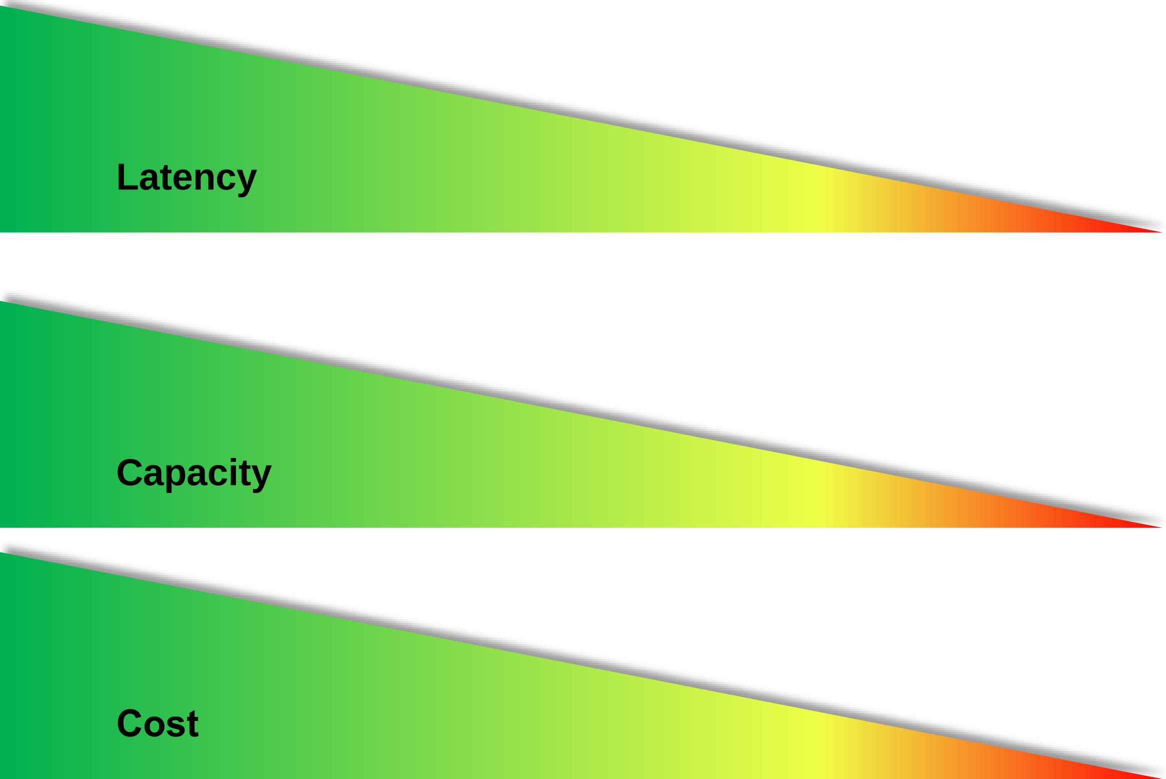
INTRIG INFORMATION & NETWORKING TECHNOLOGIES RESEARCH & INNOVATION GROUP

UNICAMP

FEEC Faculdade de Engenharia Elétrica e de Computação UNICAMP

BACKUP

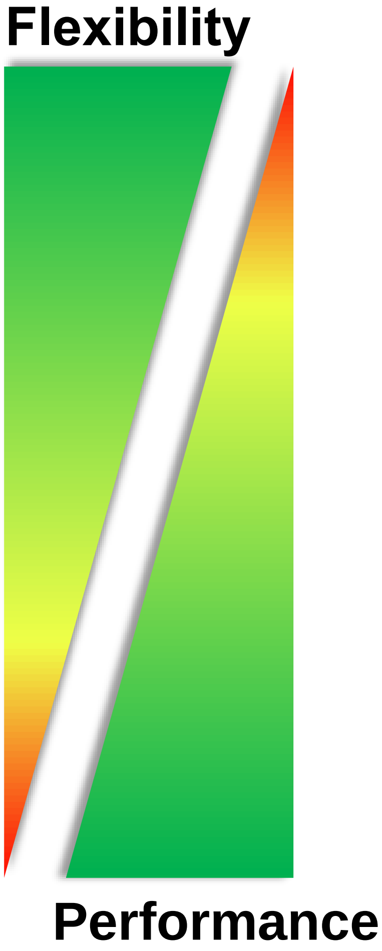
Flexibility



Latency

Capacity

Cost



Flexibility

Performance

Flexibility

- HW cloud talk
- M. He et al. “Flexibility in Softwarized Networks: Classifications and Research Challenges”

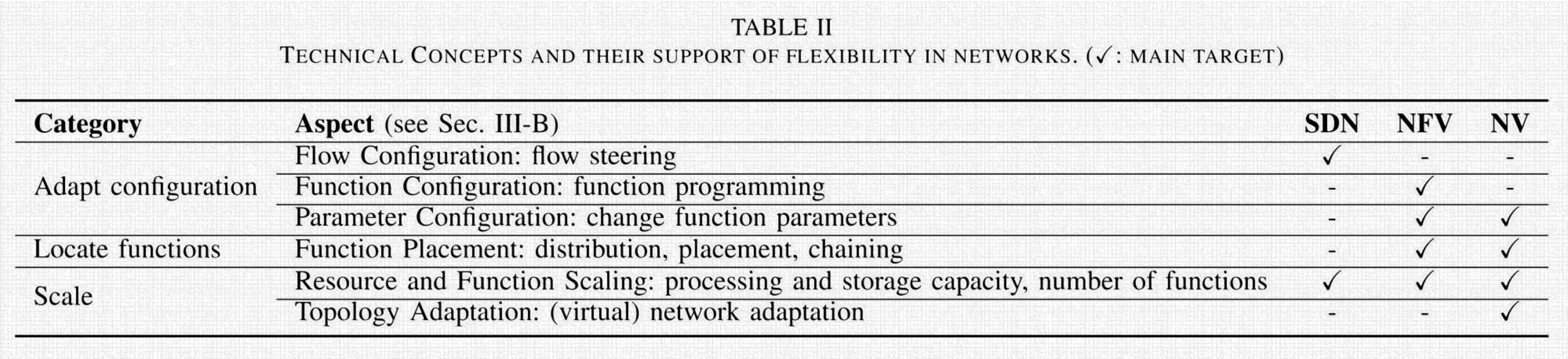
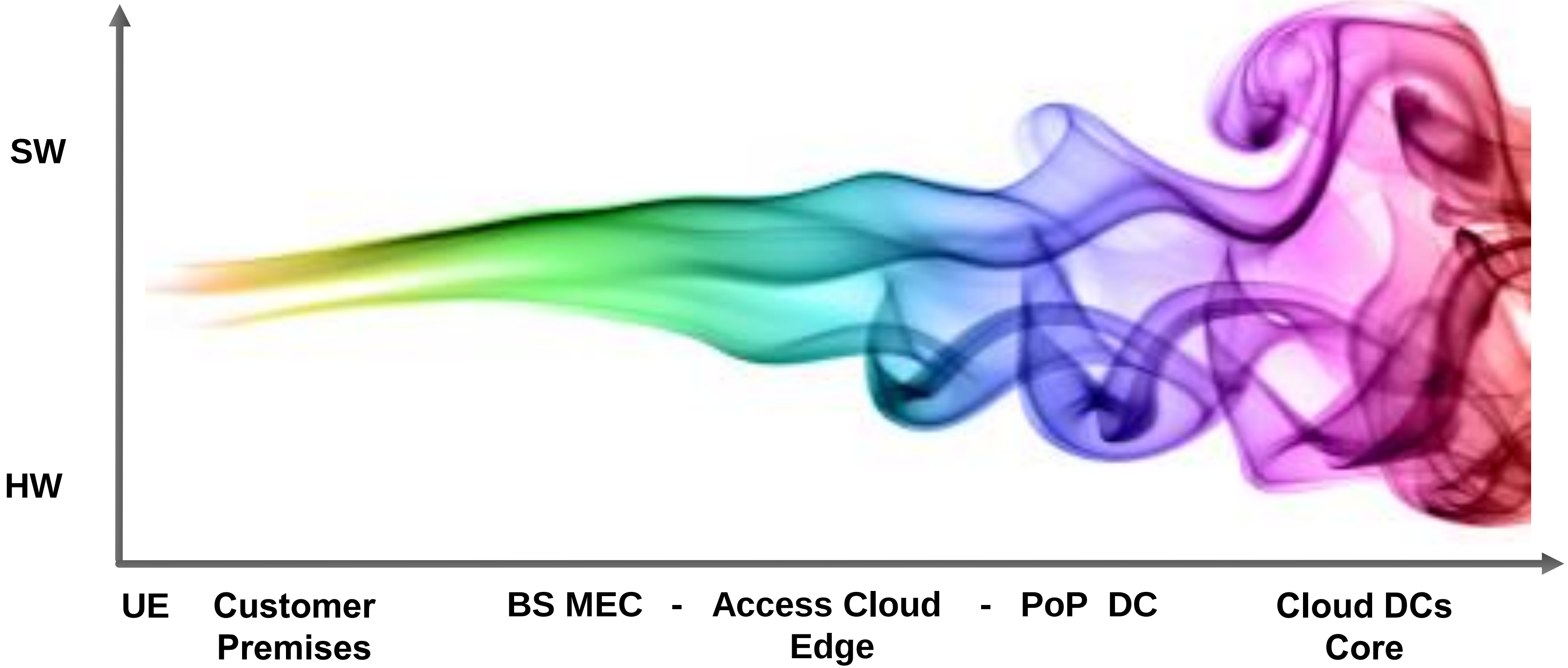


TABLE II

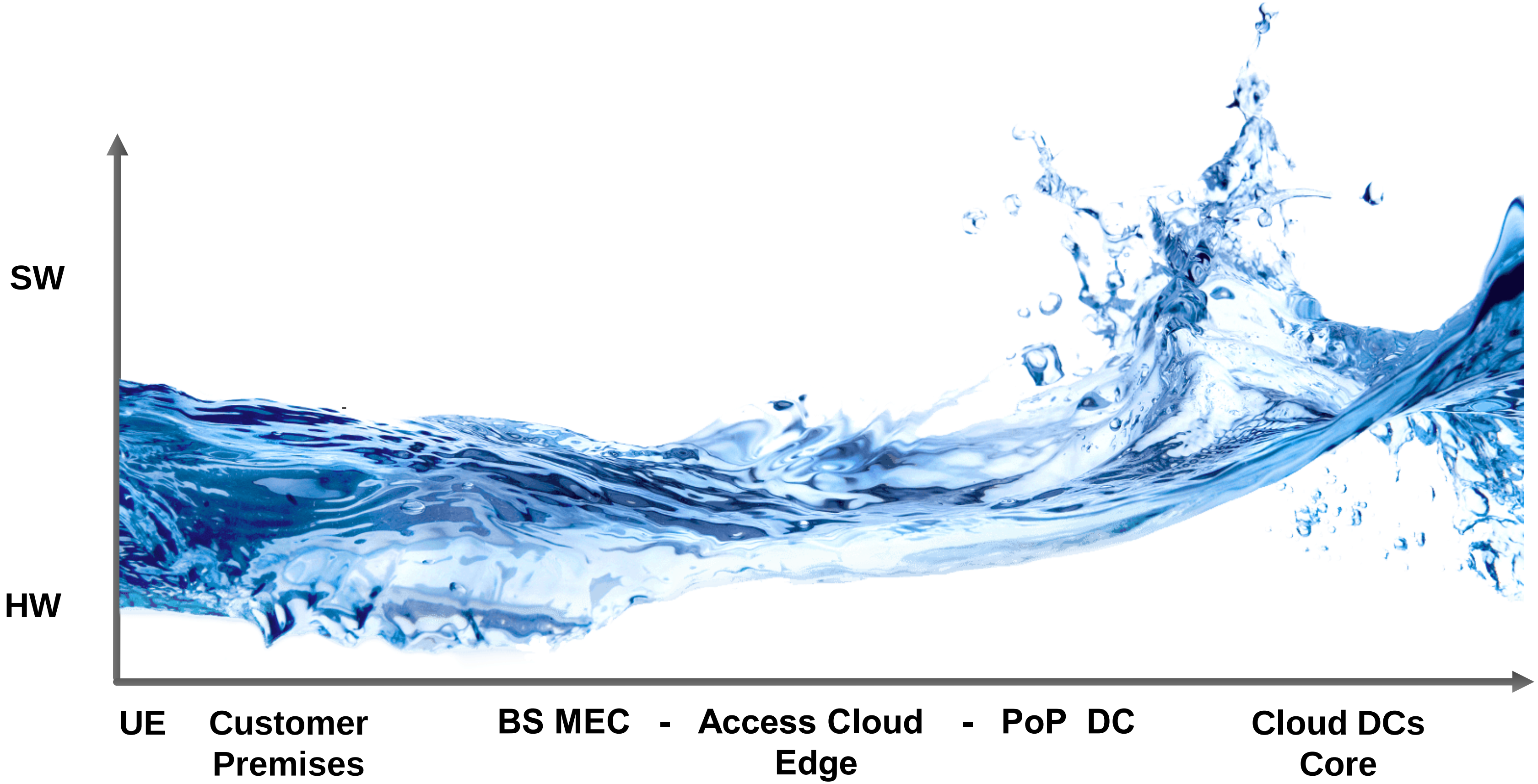
TECHNICAL CONCEPTS AND THEIR SUPPORT OF FLEXIBILITY IN NETWORKS. (✓: MAIN TARGET)

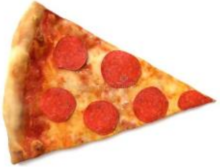
Category	**Aspect** (see Sec. III-B)	**SDN**	**NFV**	**NV**
Adapt configuration	Flow Configuration: flow steering	✓	-	-
	Function Configuration: function programming	-	✓	-
	Parameter Configuration: change function parameters	-	✓	✓
Locate functions	Function Placement: distribution, placement, chaining	-	✓	✓
Scale	Resource and Function Scaling: processing and storage capacity, number of functions	✓	✓	✓
	Topology Adaptation: (virtual) network adaptation	-	-	✓

The Fluid Networking landscape



The Fluid Networking landscape





What is a Slice?

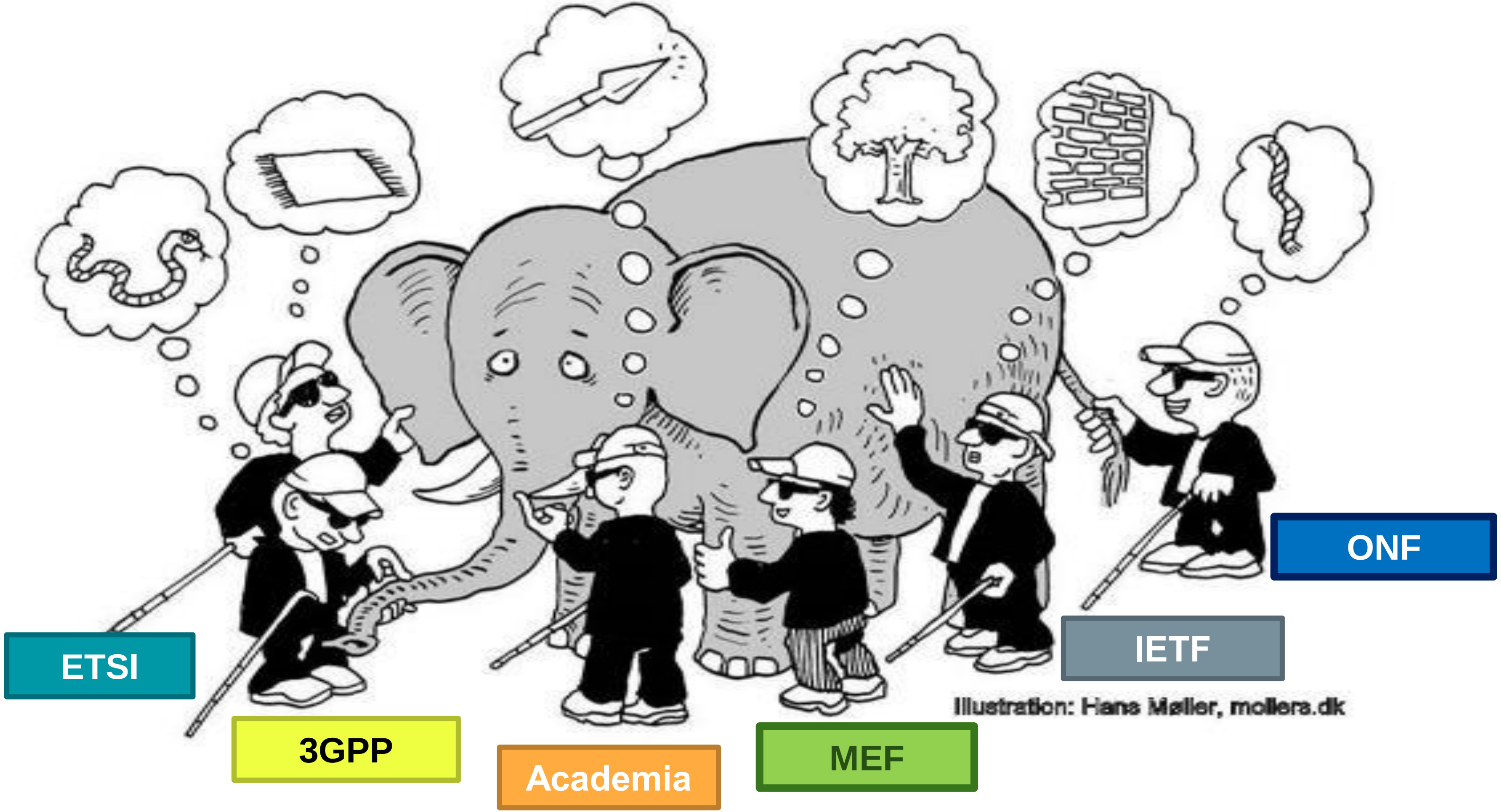
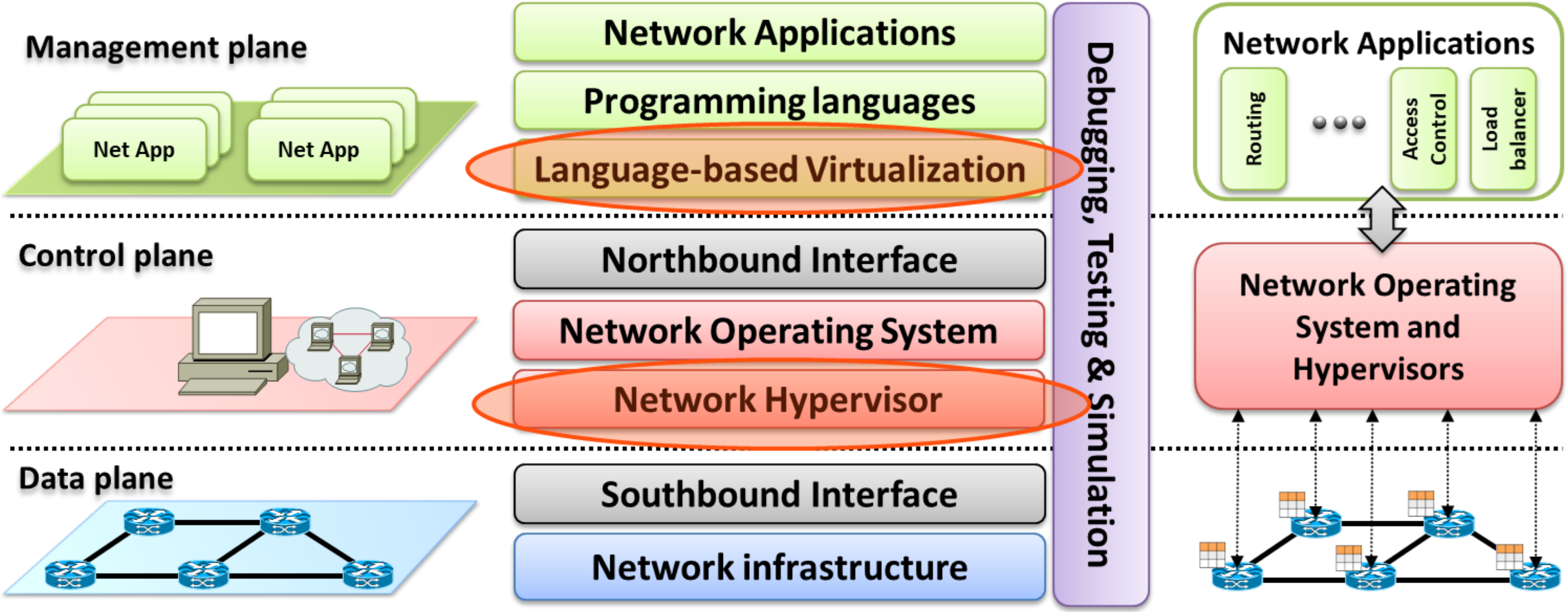


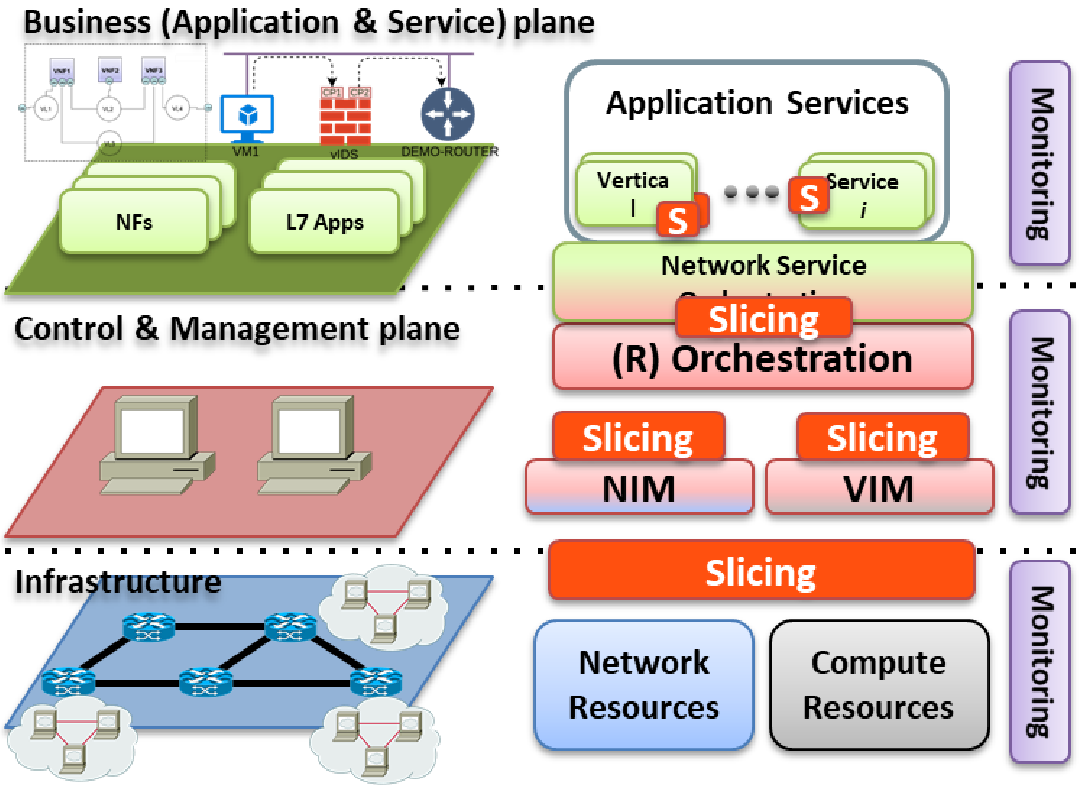
Illustration: Hans Møller, mollers.dk

SDN & Virtualization vs Slicing



Source: The NECOS project, Novel Enablers for Cloud Slicing. http://www.h2020-necos.eu/

Different Slicing Models & Approaches



Mode 3: Service-based
[Service Slice aaS]

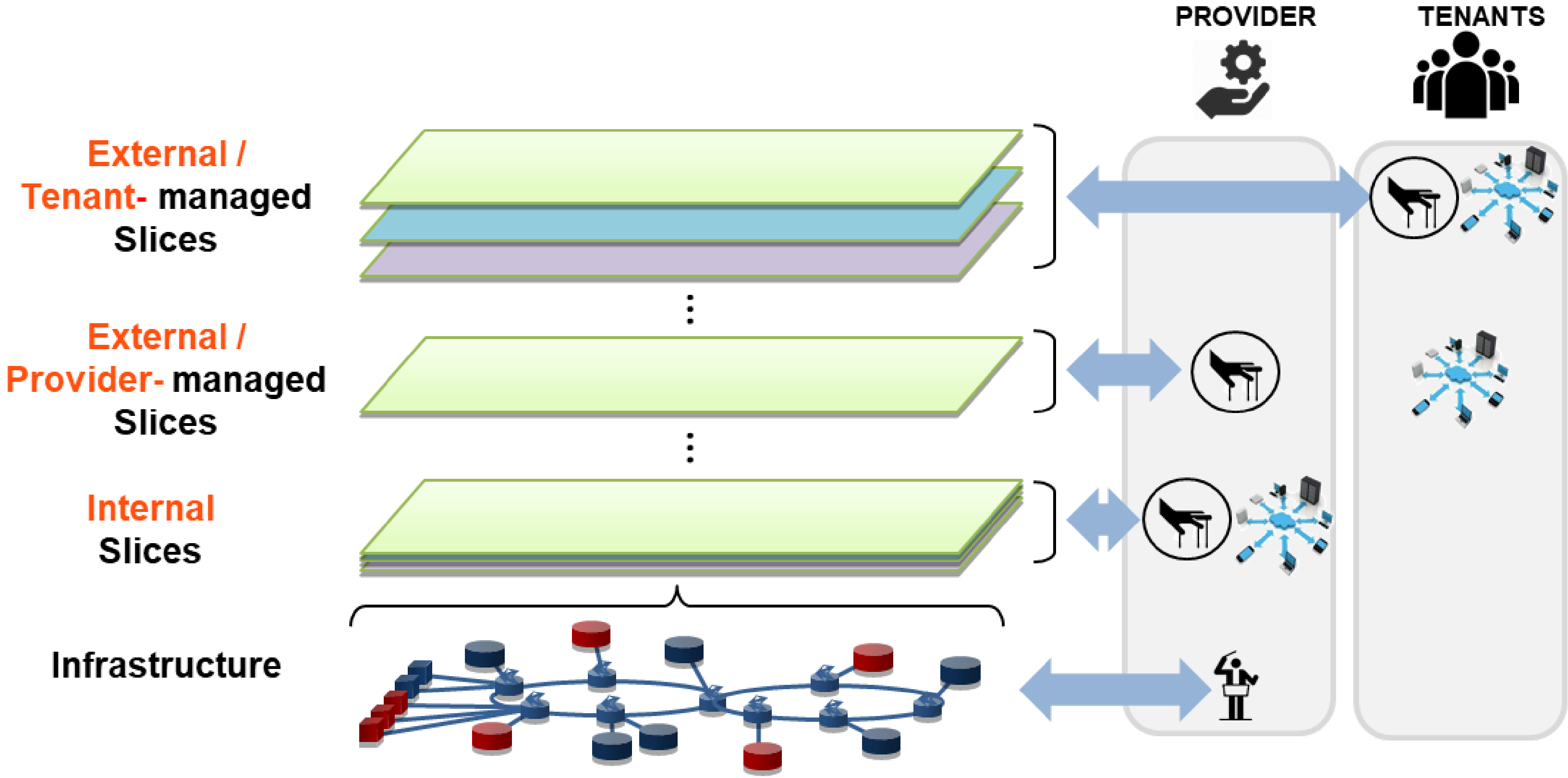
Mode 2: MANO-based
[NFV aaS]

Mode 1: VIM-dependent
[Platform Slice aaS]

Mode 0: VIM-independent
[Infra Slice aaS]
[Bare-metal Slice]

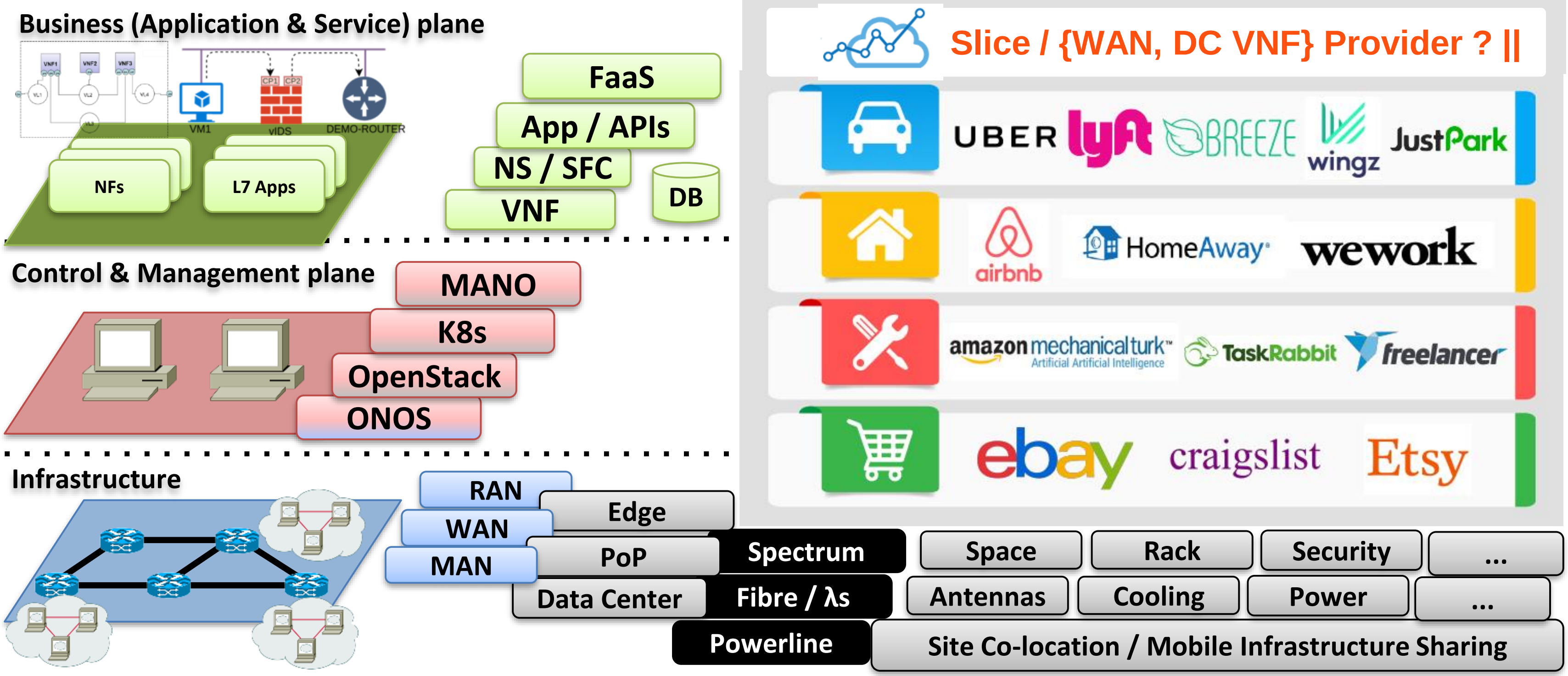
Source: The NECOS project, Novel Enablers for Cloud Slicing. http://www.h2020-necos.eu/

Types of Slices and Control Responsibilities


PROVIDER
TENANTS
External / Tenant- managed Slices
External / Provider- managed Slices
Internal Slices
Infrastructure

Source: A Network Service Provider Perspective on Network Slicing. Luis M. Contreras and Diego R. López. IEEE Softwarization, January 2018

Slicing under massive any resource multi-tenancy (gone wild) … or when sharing economy meets cloud network slicing



Business (Application & Service) plane

FaaS

App / APIs

NS / SFC

VNF

DB

NFs

L7 Apps

Control & Management plane

MANO

K8s

OpenStack

ONOS

Infrastructure

RAN

WAN

MAN

Edge

PoP

Data Center

Spectrum

Fibre / λs

Powerline

Space

Rack

Security

...

Antennas

Cooling

Power

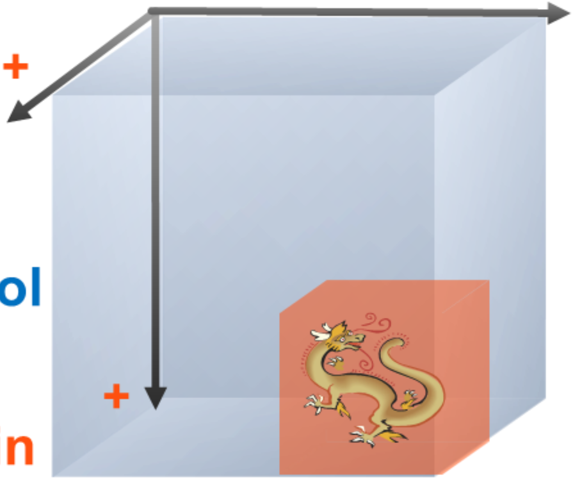
...

Site Co-location / Mobile Infrastructure Sharing

Slice / {WAN, DC VNF} Provider ? ||

Source: http://www.h2020-necos.eu/ Source (image "sharing economy"): https://www.kreezalid.com/blog/78403-what-is-sharing-economy

Deep Slicing: Concept and Challenging Trade-offs



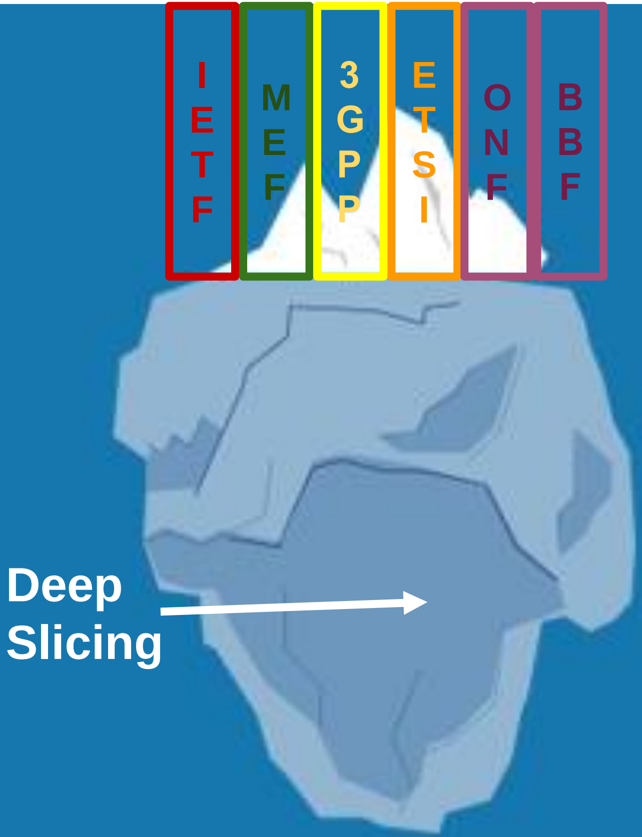
+ Isolation under massive multi-tenancy

+ Resources / Functions Protocol stack choice & control

+ Multi-Domain Administrative & Technological

Source: Inspired by the author (C. Rothenberg) P^3 trade-offs: Programmability, Performance, Portability.
https://www.slideshare.net/chesteve/ieee-hpsr-2017-keynote-softwarized-dataplanes-and-the-p3-tradeoffs-programmability-performance-portabiilty

Towards Deep Slices



Fragmented Standardization

Business challenges & Technological challenges

From infrastructure sharing to any-layer any-resource sharing (from PHY to APP)

Deep

End-to-End, Multi-Domain (tech + admin)

Tenant Choice & Control

Isolation + Scaling

any resource, any function anywhere

Deep Slicing: Challenges up front

Standardization gap goes hand by hand with a series of **key challenges from provider's perspective** on (i) scalability, (ii) arbitration, (iii) slice planning and dimensioning, and (iv) multi-domain (cf. [FG-NET-Contribution]). Both business and technical implications can be deemed necessary for such multi-operator slice provisioning context.

From the **business** side, some key implications include: (i) coordination models, (ii) inter-provider SLAs, (iii) pricing schemes, (iv) service specification, and (v) customer facing advertisement.

From a **technical** perspective we highlight (i) slice decomposition, (ii) discovery of domains, (iii) common abstraction models, (iv) standard interfaces/protocols, APIs.

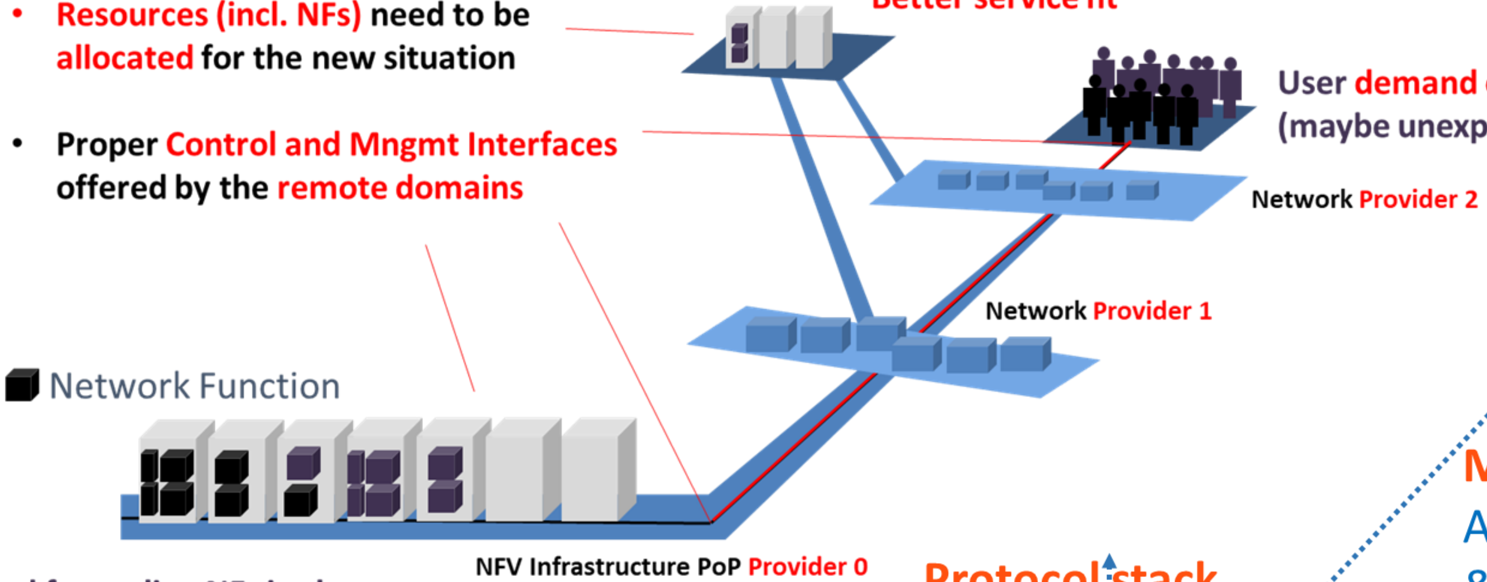
Source & further reading: Doc.6 ITU-T FG 2030 contribution: Network 2030 Challenges and Opportunities in Network Slicing
https://extranet.itu.int/sites/itu-t/focusgroups/net-2030/_layouts/15/WopiFrame.aspx?sourcedoc=%7bC4E9266E-1058-4035-AA25-451ABCB5C07B%7d&file=NET2030-I-006.docx&action=default

- **Resources (incl. NFs) need to be allocated for the new situation**
- **Proper Control and Mngmt Interfaces offered by the remote domains**

Opportunity for instantiating NFs in proximity

Better service fit

User demand changes (maybe unexpectedly or bursty)



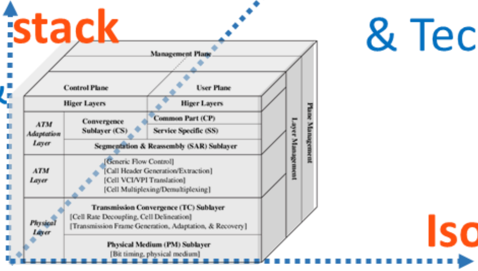
Need for scaling NFs in the origin domain could not be sufficient

Protocol stack

Choice & Control

Multi-Domain

Administrative & Technological


Management Plane
Control Plane
User Plane
Higer Layers
Higer Layers
ATM Adaptation Layer
Convergence Sublayer (CS)
Common Part (CP)
Service Specific (SS)
Segmentation & Reassembly (SAR) Sublayer
ATM Layer
[Generic Flow Control]
[Call Header Generation/Extraction]
[Cell VCI/VPI Translation]
[Cell Multiplexing/Demultiplexing]
Physical Layer
Transmission Convergence (TC) Sublayer
[Cell Rate Decoupling, Cell Delineation]
[Transmission Frame Generation, Adaptation, & Recovery]
Physical Medium (PM) Sublayer
[Bit timing, physical medium]
Layer Management
Plane Management

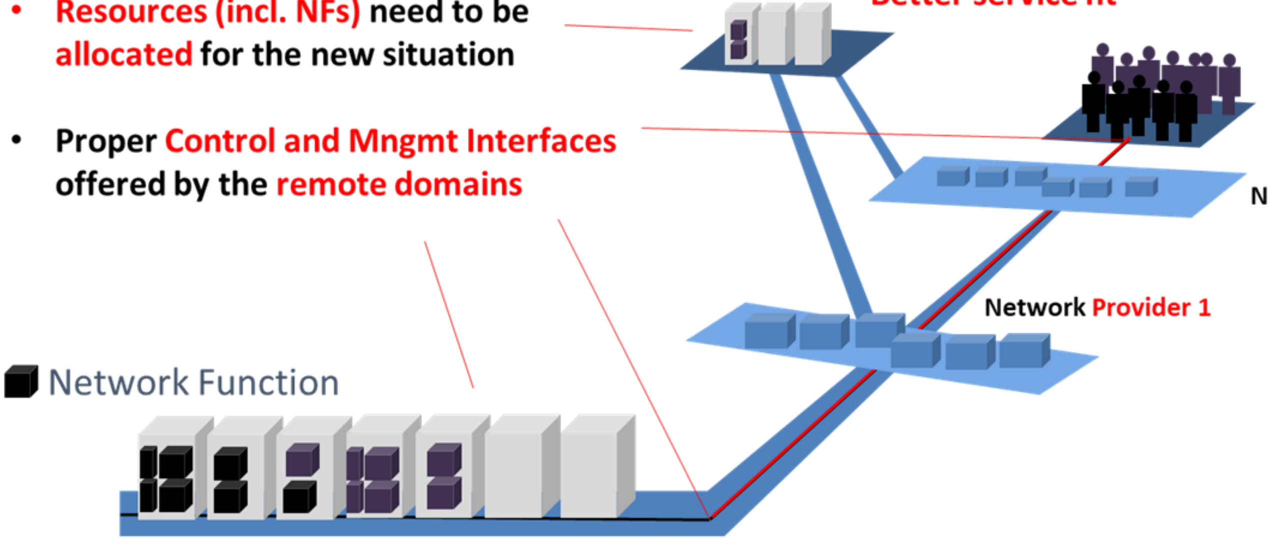
Isolation

under massive multi-tenancy

Source: Adapted from slide courtesy by Luis M. Contreras, Telefonica

Multi-Domain Slicing Scenario

- **Resources (incl. NFs) need to be allocated for the new situation**
- **Proper Control and Mngmt Interfaces offered by the remote domains**



Opportunity for instantiating NFs in proximity
Better service fit

User demand changes
(maybe unexpectedly or bursty)

Need for scaling NFs in the
origin domain could not be sufficient

Source: Adapted from slide courtesy by Luis M. Contreras, Telefonica

Why slice-ready federation is needed?

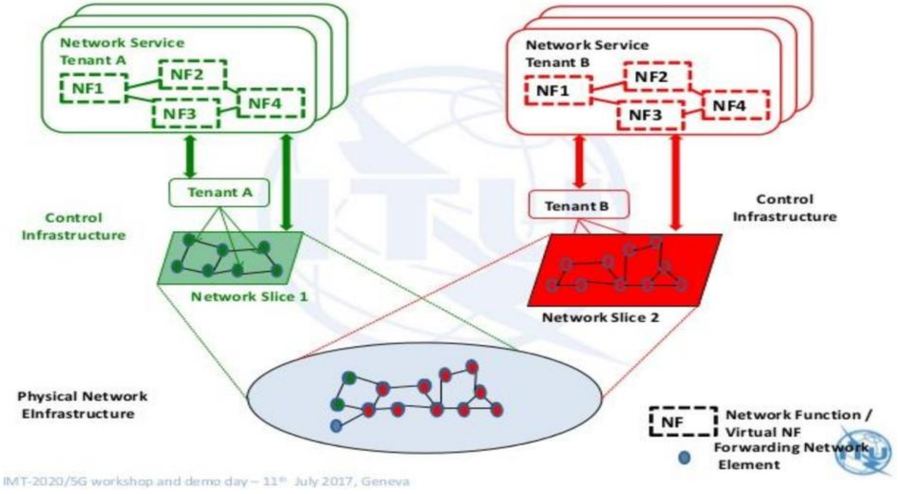
- Vertical customers can request **services** that lay **outside the footprint** of their **primary provider**
- Interaction with other providers are needed but …
  - How we can **charge** and bill for that service?
  - How we can **ensure SLAs** among providers?
  - How we can **know about the capabilities** of other providers for a comprehensive e2e service provision?
- The current interconnection models is **not aware of peer's** network **resources** (i.e., load conditions, etc)
- All these **environments are static**, requiring long interactions for setting up any inter-provider connection
- **Automation** for both the **interconnection** sessions and the **service deployment** on top of that is needed to reach the goal of **flexibility and dynamicity**

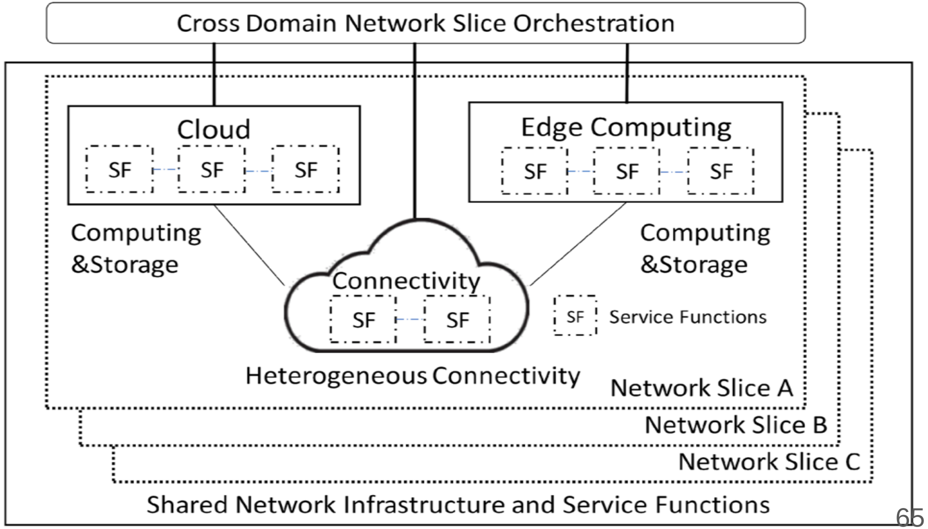
Source: Luis M. Contreras, Telefonica

Slicing in Scope

Network Slice – A Network Slice is a **jointly managed group of subsets of resources, network functions / network virtual functions at the data, control, management/orchestration, and service planes at any given time.**

Cross-domain management of network slices in network infrastructure and service functions





Acknowledgments

Work by Christian Rothenberg was supported by the Innovation Center, Ericsson Telecomunicações S.A., Brazil under grant agreement UNI.64.

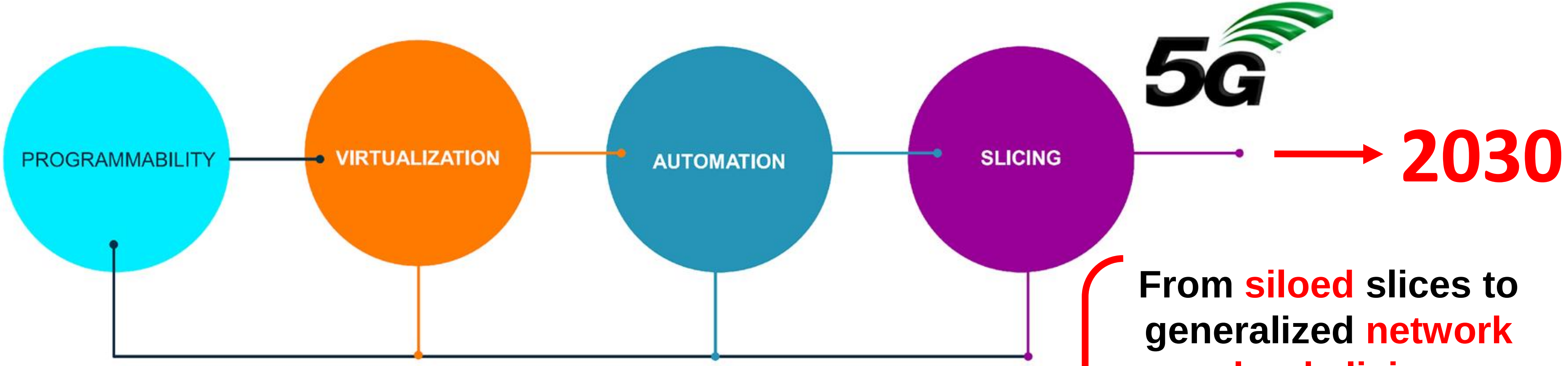
Ack. Mateus Santos and Pedro Gomes for input insights

This work includes contributions funded was partially funded by the EU-Brazil NECOS project under grant agreement no. 777067.

Luis M. Contreras and Alex Galis, co-authors of ITU-T FG 2030 input Doc.6: Network 2030 Challenges and Opportunities in Network Slicing.

Raphael Rosa (PhD candidate at UNICAMP), for his contributions to the vision around Unfolding Slices, Control Loops (in a Loop), Disaggregated Metrics/Prices, and Smart Peering

Slicing Journey: from 5G towads 2030



From **siloed** slices to generalized **network cloud slicing**

Deep, massive resource **sharing & multi-tenancy**

New **Tenant-Provider** relationships and **power of choices**

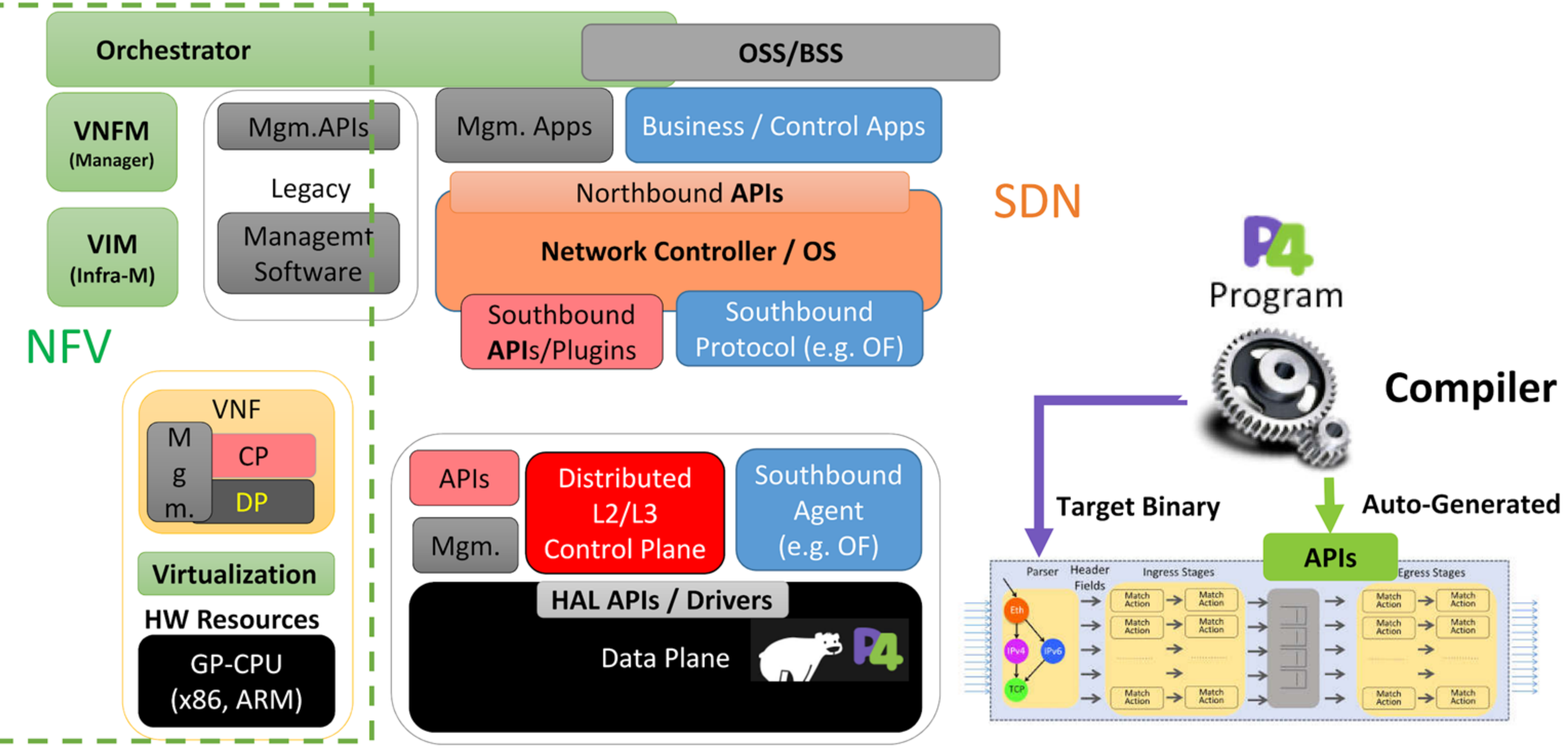
Executive Summary

Source. Adapted from slide courtesy by Luis M. Contreras, Telefonica

Main relevant standardization related activities to Slicing

- **NGMN** Slices - consist of 3 layers: 1) Service Instance Layer, 2) Network Slice Instance Layer, and 3) Resource layer (2016).
- **3GPP** - SA2 23.799 Study Item “Network Slicing’ (2016);
  SA5 TR 28.801Study Item “Network Slicing (2017)
- **ITU-T** IMT2020 - Recommendations: 5G Architecture, Management of 5G, Network Softwarisation and Slicing - (2016 – 2017)
- **ONF** - Recommendation TR-526 “Applying SDN architecture to Network Slicing” (2016)
- **BBF** - Requirements / architecture of transport network slicing SD-406: E2E Network Slicing (2017)
- **ETSI** - NFV priorities for 5G (white paper) (2017). ZSM ISG automation technology for network slice management (2018). MEC support for network slicing (2018)
- **IETF** - No specific WG (despite attempts in 2017-2018).
  draft-galis-netslices-revised-problemstatement-03, draft-geng-netslices-architecture-02, draft-geng-coms-architecture-01, draft-netslices-usecases-01, draft-qiang-coms-use-cases-00, draft-qiang-coms-netslicing-information-model-02, draft-galis-anima-autonomic-slice-networking-04, draft-defoy-coms-subnetinterconnection-03, draft-homma-coms-slicegateway-01

Different SDN Models to Program / Refactor the Stack

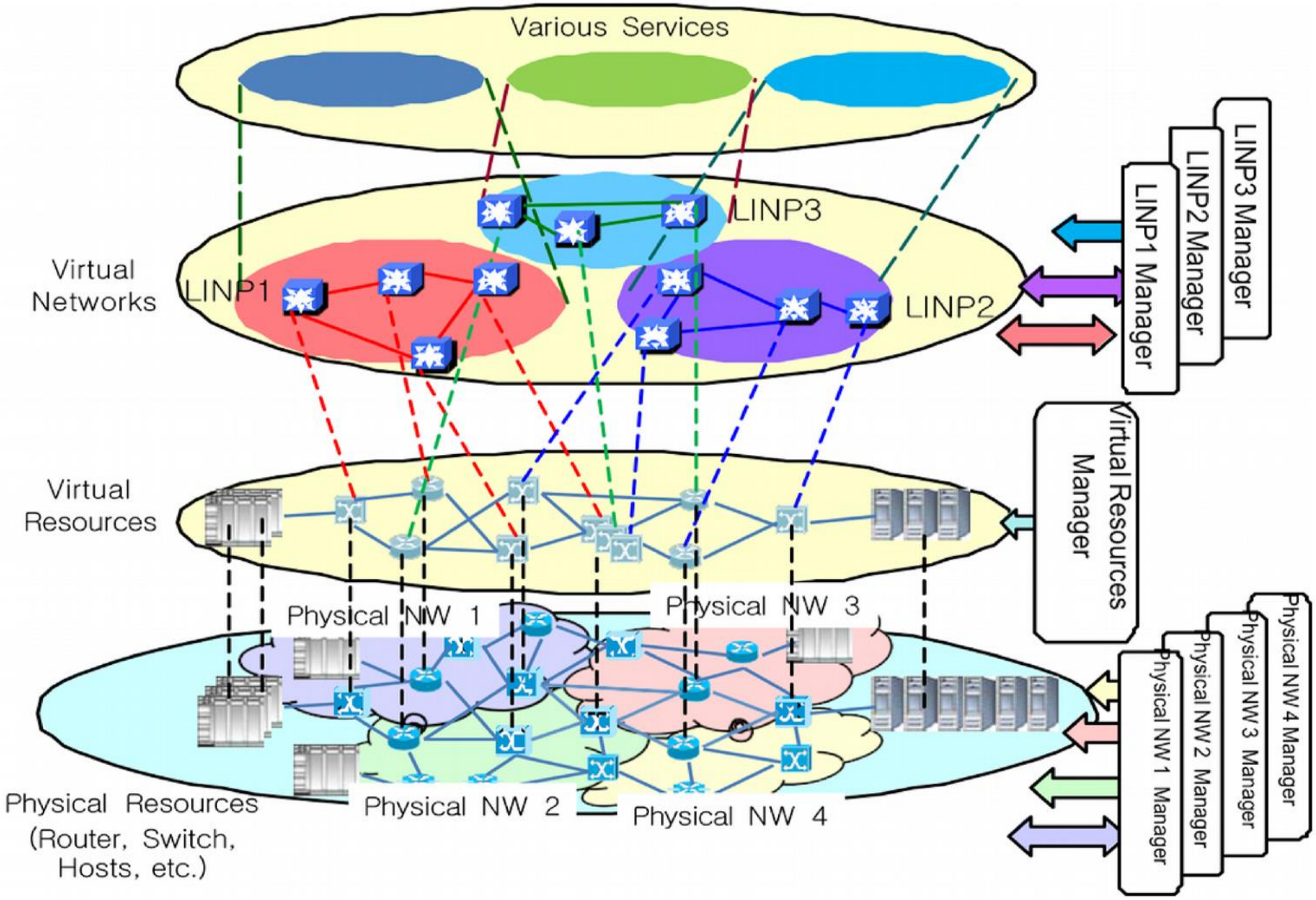


History of Network Slicing

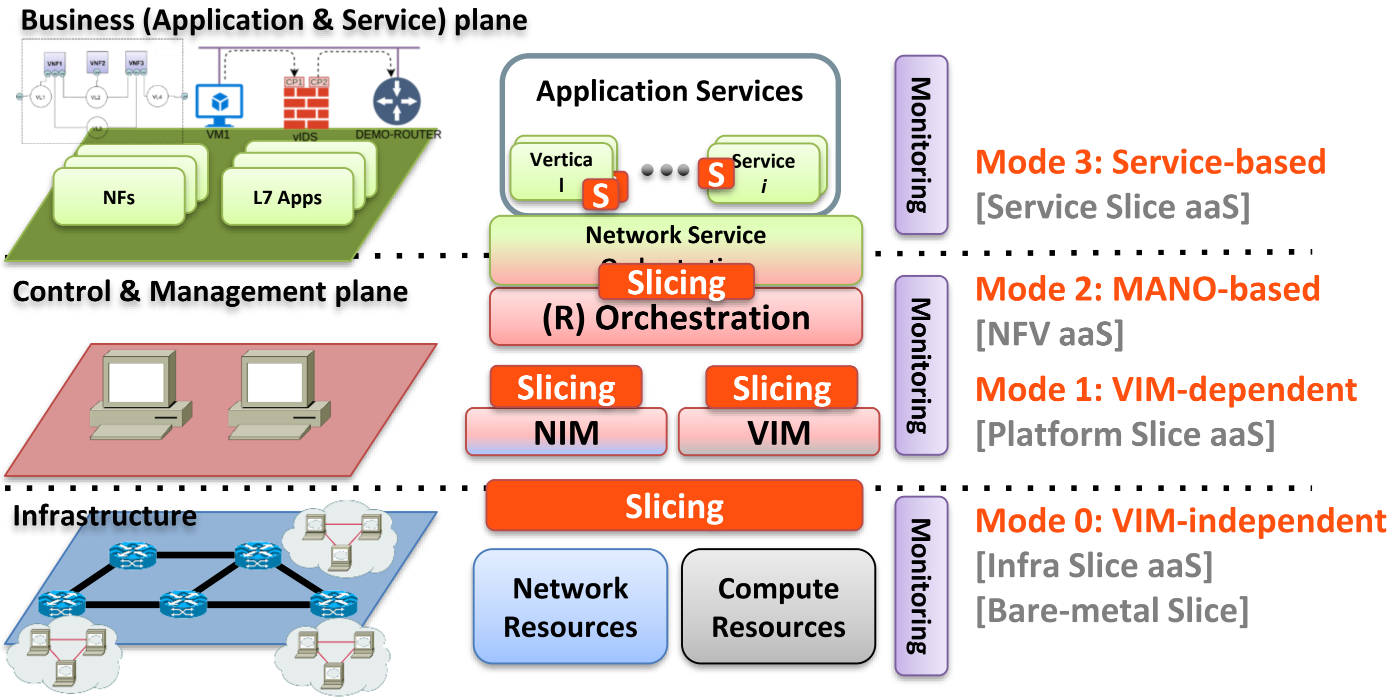
[Not today
towards 2030]

- Early references: Programmable Networks research & Federated Testbed research (1995 -2012)
- GENI Slice (2008): “A GENI slice is the unit of isolation for experiments. A container for resources used in an experiment; A unit of access control
- **ITU-T Slicing** (2011) as defined in [ITU-T Y.3011], [ITUTY.3012] Slicing allows logically isolated network partitions (LINP) with a slice being considered as a unit of programmable resources such as network, computation and storage
- Many more...
  - See: Alex Galis, Netsoft 2018 Tutorial:
    "Network Slicing Landscape: A holistic architectural approach"
    http://www.maps.upc.edu/public/presentations/netsoft18_slicingtutorial_v1.0.pdf

ITU-T
LINP


Various Services
Virtual Networks
LINP1
LINP2
LINP3
LINP3 Manager
LINP2 Manager
LINP1 Manager
Virtual Resources
Virtual Resources Manager
Physical NW 1
Physical NW 2
Physical NW 3
Physical NW 4
Physical NW 4 Manager
Physical NW 3 Manager
Physical NW 2 Manager
Physical NW 1 Manager
Physical Resources
(Router, Switch, Hosts, etc.)

Different Slicing Models & Approaches



Business (Application & Service) plane

NFs

L7 Apps

Control & Management plane

Infrastructure

Application Services

Vertical

Service *i*

S

S

Network Service Orchestration

Slicing

(R) Orchestration

Slicing

NIM

Slicing

VIM

Slicing

Network Resources

Compute Resources

Monitoring

Monitoring

Monitoring

Mode 3: Service-based
[Service Slice aaS]

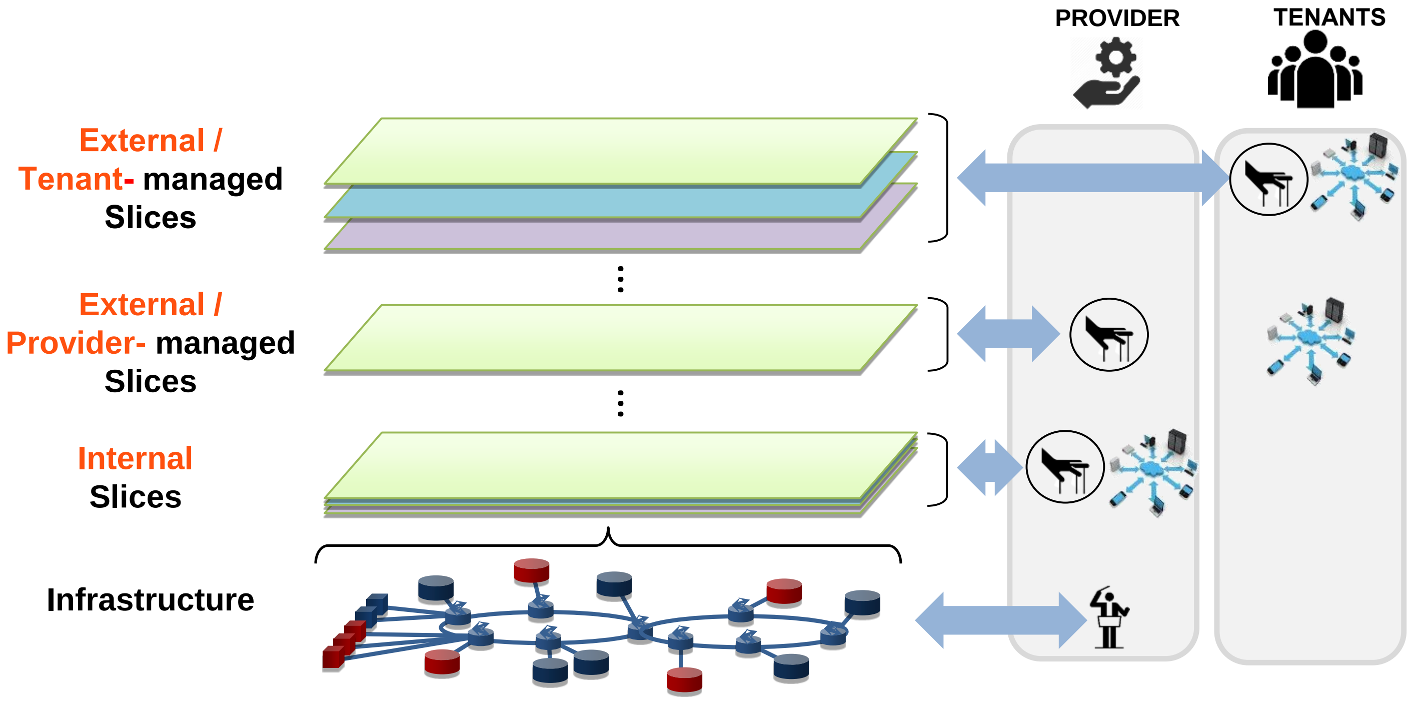
Mode 2: MANO-based
[NFV aaS]

Mode 1: VIM-dependent
[Platform Slice aaS]

Mode 0: VIM-independent
[Infra Slice aaS]
[Bare-metal Slice]

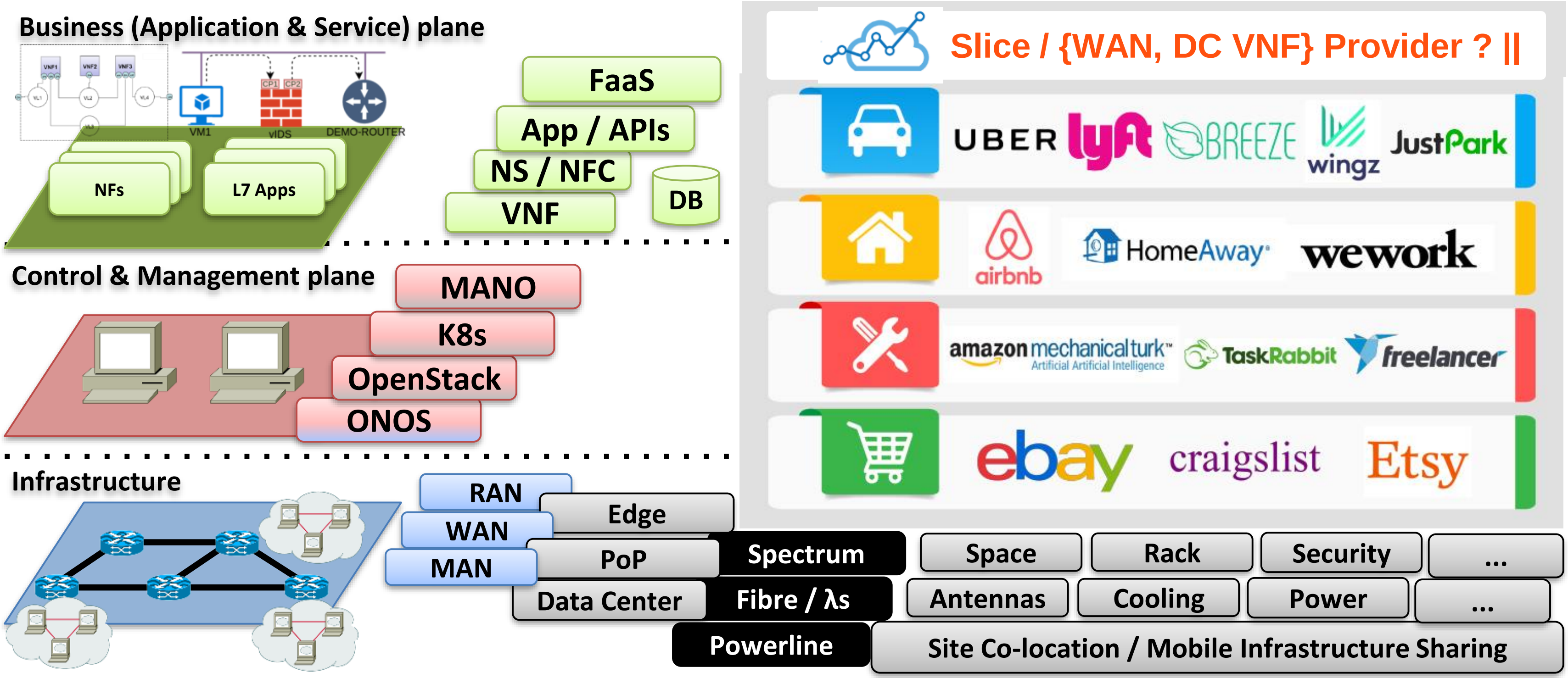
Source: The NECOS project, Novel Enablers for Cloud Slicing. http://www.h2020-necos.eu/

Types of Slices and Control Responsibilities


PROVIDER
TENANTS
External / Tenant- managed Slices
External / Provider- managed Slices
Internal Slices
Infrastructure

Source: A Network Service Provider Perspective on Network Slicing. Luis M. Contreras and Diego R. López. IEEE Softwarization, January 2018

Slicing under massive any resource multi-tenancy (gone wild) … or when sharing economy meets cloud network slicing



Business (Application & Service) plane

NFs

L7 Apps

FaaS

App / APIs

NS / NFC

VNF

DB

Control & Management plane

MANO

K8s

OpenStack

ONOS

Infrastructure

RAN

WAN

MAN

Edge

PoP

Data Center

Spectrum

Fibre / λs

Powerline

Slice / {WAN, DC VNF} Provider ? ||

Space

Rack

Security

...

Antennas

Cooling

Power

...

Site Co-location / Mobile Infrastructure Sharing

Source: http://www.h2020-necos.eu/

Source (image “sharing economy”): https://www.kreezalid.com/blog/78403-what-is-sharing-economy

What do we mean by Network Slices?

Network Slice – A Network Slice is a **managed group of subsets of resources, network functions / network virtual functions at the data, control, management/orchestration, and service planes at any given time.**

The behaviour of the network slice is realized via network slice instances (i.e. activated network slices, dynamically and non-disruptively re-provisioned).

A network slice is programmable and has the ability to expose its capabilities.

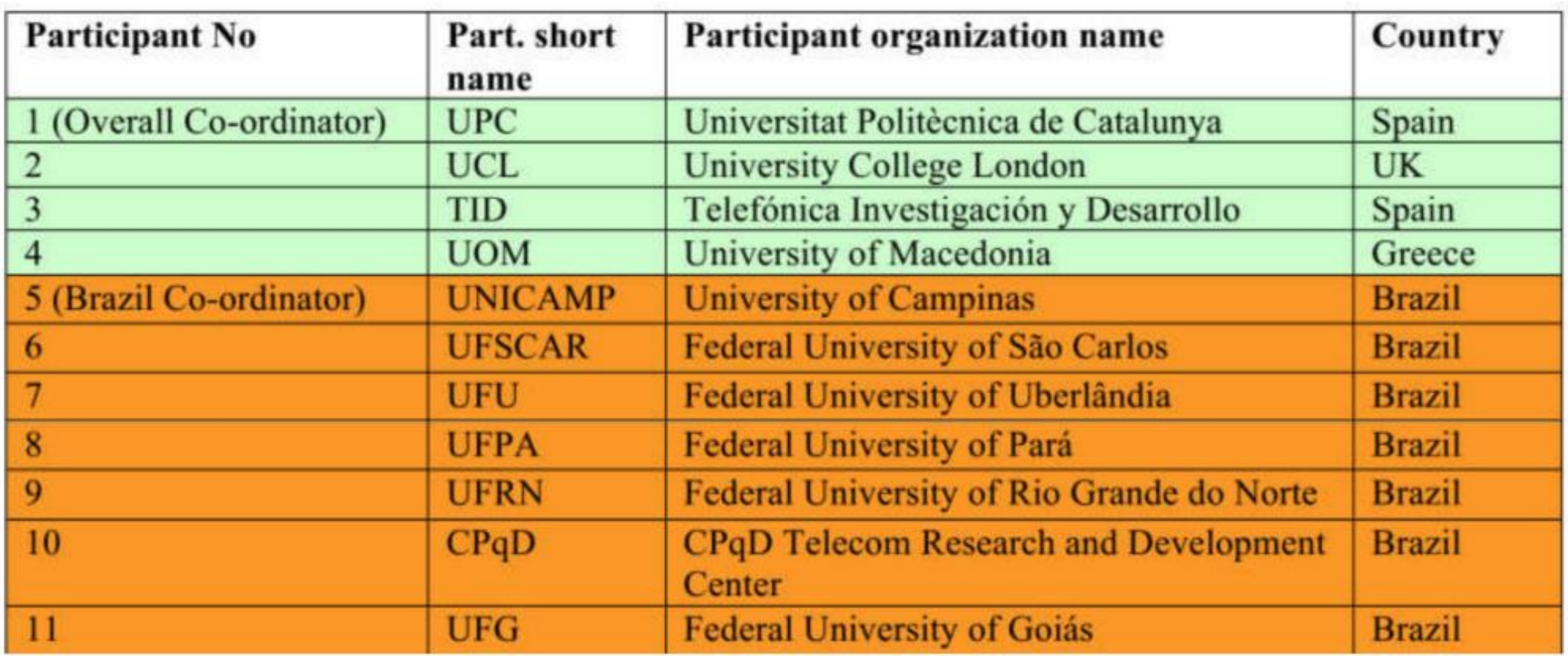
→ A network slice supports at least one type of service.

→ A network slice may consist of cross-domain components from separate domains in the same or different administrations, or components applicable to the access network, transport network, core network, and edge networks.

→ A resource-only partition is one of the components of a Network Slice, however on its own does not fully represent a Network Slice.

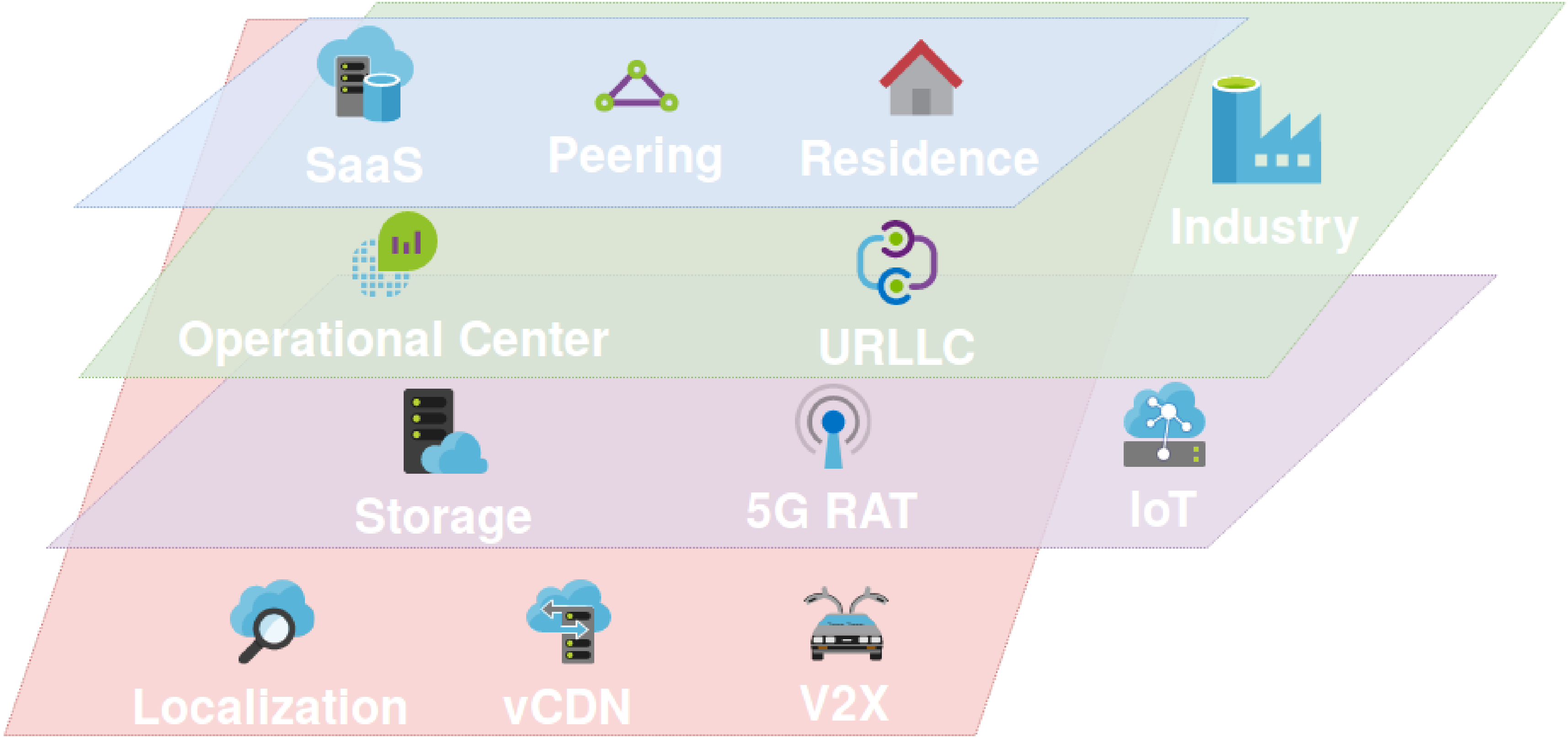
→ Underlays / overlays supporting all services equally (‘best effort” support) are not fully representing a Network Slice

Consortium

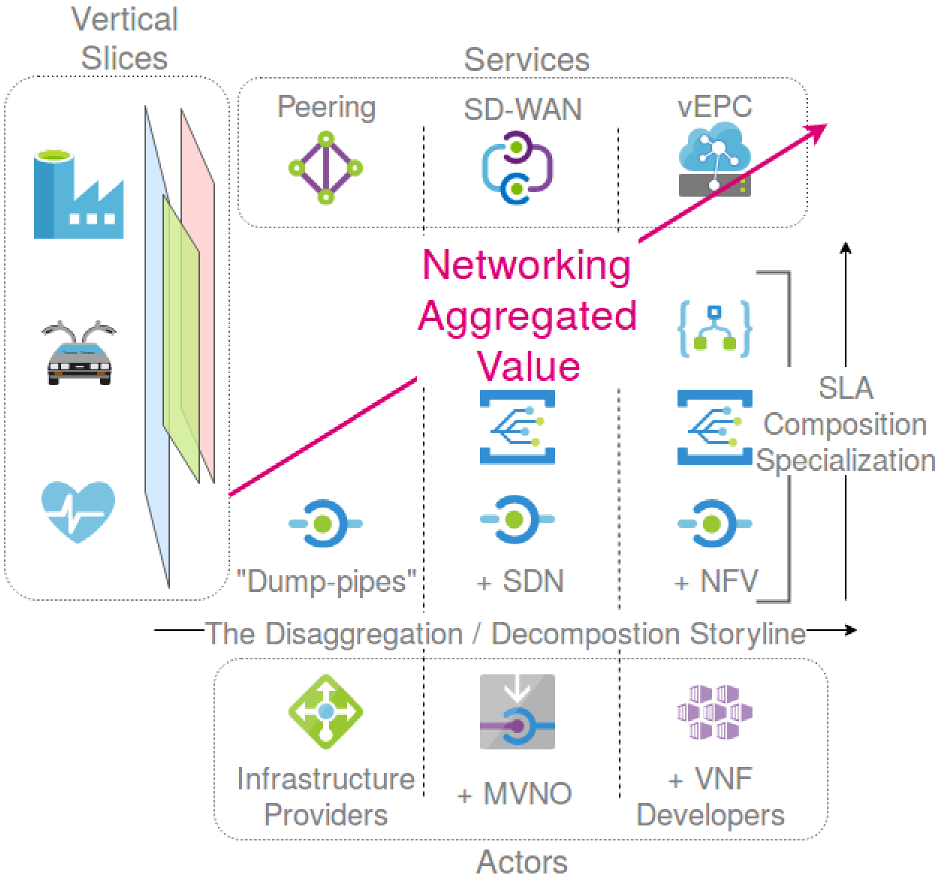


Participant No	Part. short name	Participant organization name	Country
1 (Overall Co-ordinator)	UPC	Universitat Politècnica de Catalunya	Spain
2	UCL	University College London	UK
3	TID	Telefónica Investigación y Desarrollo	Spain
4	UOM	University of Macedonia	Greece
5 (Brazil Co-ordinator)	UNICAMP	University of Campinas	Brazil
6	UFSCAR	Federal University of São Carlos	Brazil
7	UFU	Federal University of Uberlândia	Brazil
8	UFPA	Federal University of Pará	Brazil
9	UFRN	Federal University of Rio Grande do Norte	Brazil
10	CPqD	CPqD Telecom Research and Development Center	Brazil
11	UFG	Federal University of Goiás	Brazil

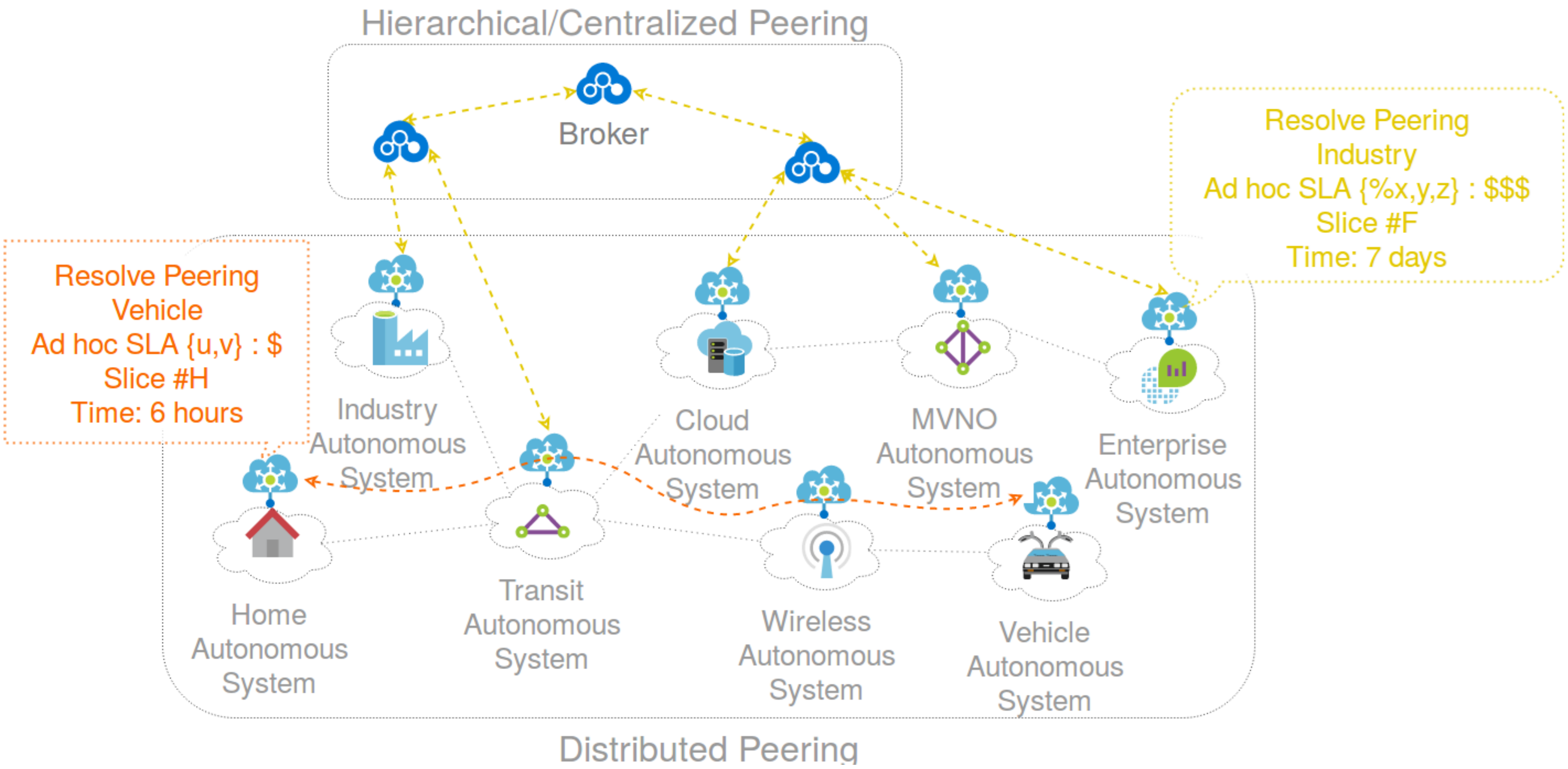
Unfolding Slices through Massive Multi-Tenancy



Disaggregated Metrics/Prices: SLA Hazards



Smart Peering for Multi-Domain NS-as-a-S

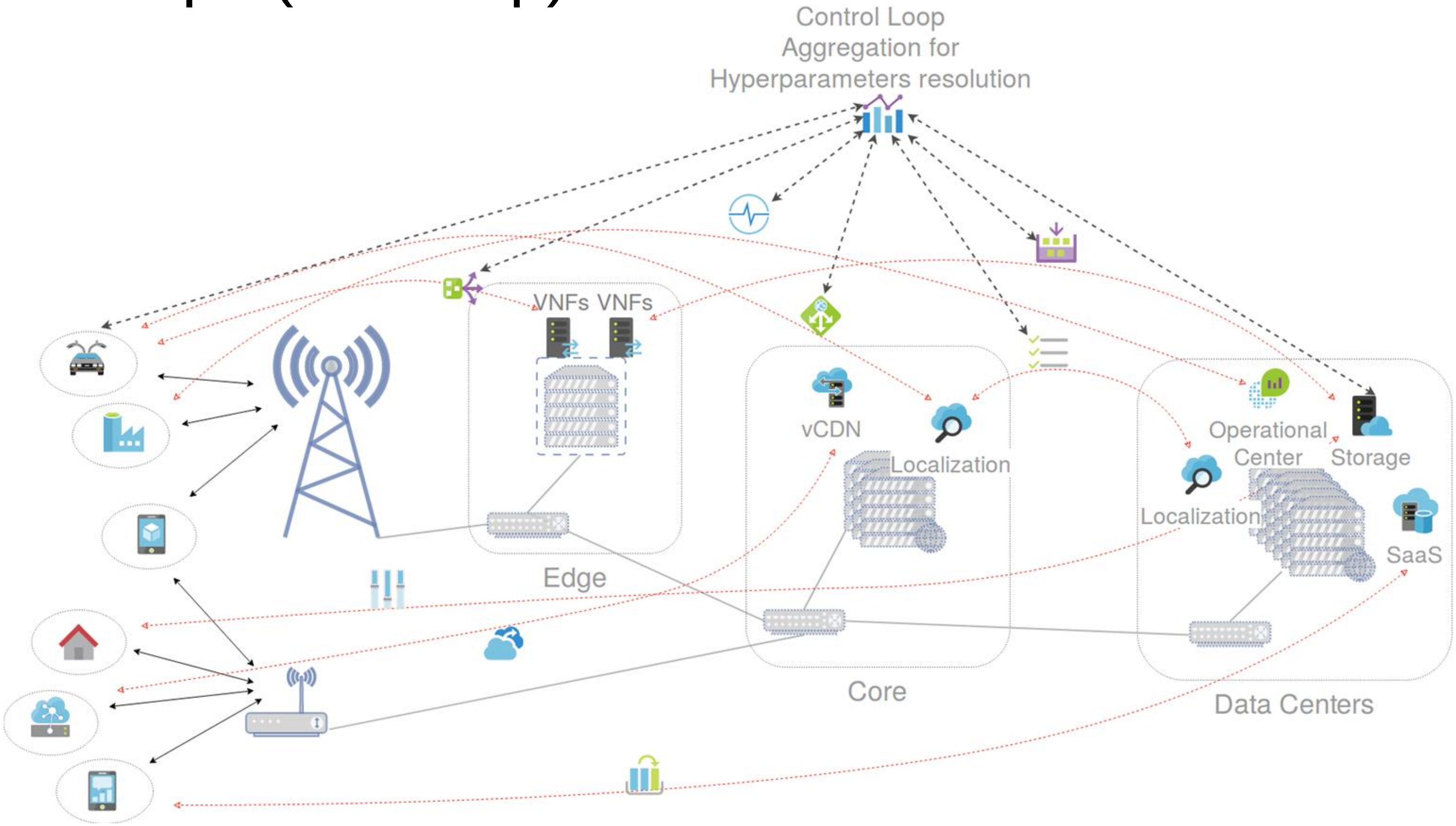


Hierarchical/Centralized Peering
Broker
Industry Autonomous System
Cloud Autonomous System
MVNO Autonomous System
Enterprise Autonomous System
Home Autonomous System
Transit Autonomous System
Wireless Autonomous System
Vehicle Autonomous System
Distributed Peering

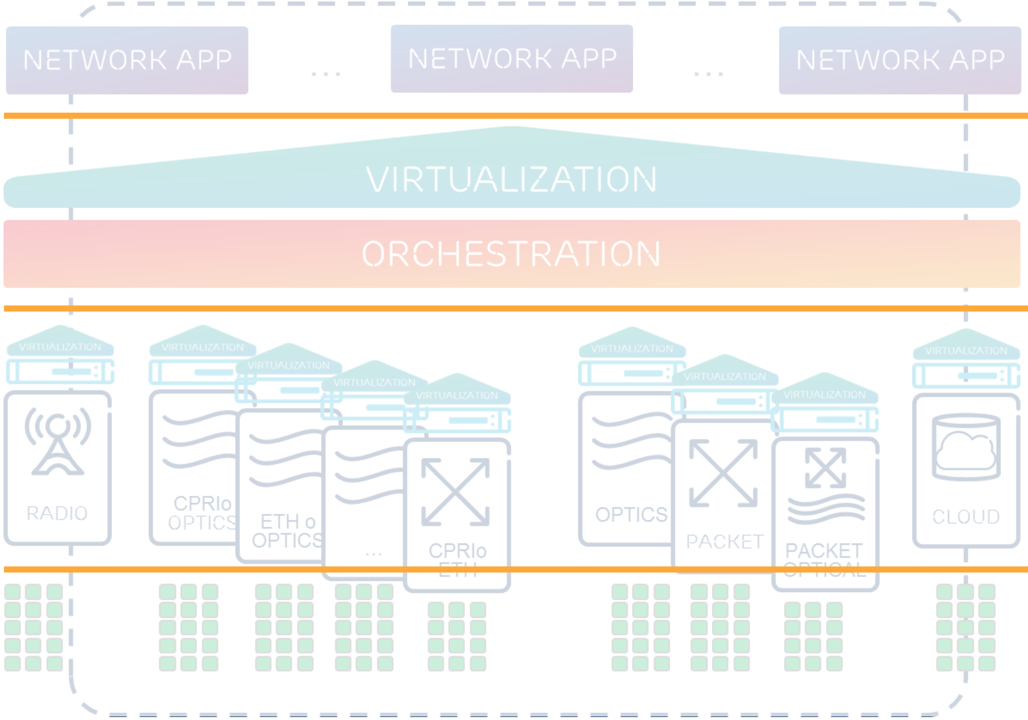
Resolve Peering
Vehicle
Ad hoc SLA {u,v} : $
Slice #H
Time: 6 hours

Resolve Peering
Industry
Ad hoc SLA {%x,y,z} : $$$
Slice #F
Time: 7 days

Control Loops (in a loop)



Expose just enough information to make optimal resource orchestration.



Provide service

Orchestration Layer: ~1

Simplified view
Relevant data

Domain Controllers: 10s

Detailed control
Technology dependent

Network nodes: 1000s

Source: Netsoft 2017 Tutorial: End-to-End Programmability and Orchestration in 5G Networks.