INTRIG

Information & Networking Technologies Research & Innovation Group

Prof. Christian Rothenberg

FEEC
Faculdade de Engenharia Elétrica e de Computação
UNICAMP

19-Apr, 2023

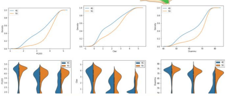
Agenda

- **About INTRIG @ UNICAMP**

INTRIG
INFORMATION & NETWORKING TECHNOLOGIES RESEARCH & INNOVATION GROUP

- **Fluid Network Control and Data Plane Research**
  - Concept Overview
  - Instances

- **Active Research:** Selected open-source fueled research threads
  - **Hybrid-P4-5G:** User Plane Functions in Programmable SW/HW Stacks
  - **P7** (P4 Programmable Patch Panel): 100G network emulation
  - **EFFECTOR:** DASH QoE Evaluation Framework with 5G Datasets
  - **QoEyes:** Virtual Reality Streaming QoE Estimation through ML and INT
  - **PoD acceleration:** Offloading container networking & security functions



- **The SMARTNESS 2030 research center (2023 - 2033)**
  - Vision
  - Research Collaboration opportunities with RedHat

Red Hat

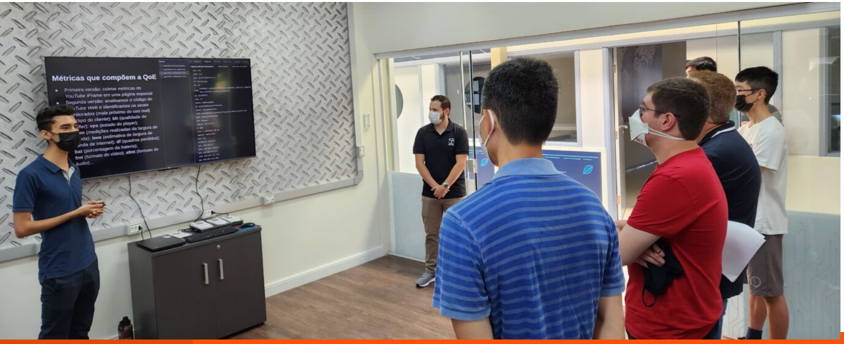
About

Prof. Christian Rothenberg & INTRIG research lab

About

INTRIG Research Lab Lead by Prof. Christian Rothenberg Since Sep/2013
@ FEEC / University of Campinas, Brazil

- **Team** (*on average*)
  - 1 postdoc, 7 PhD, 7 MSc, 5 undergrad students
- **Alumni**
  - 5 postdocs, 8 PhD, 25+ MSc, 25+ undergrads
- **Profile**
  - Systems-oriented, open-source, industry-friendly, multi-layer

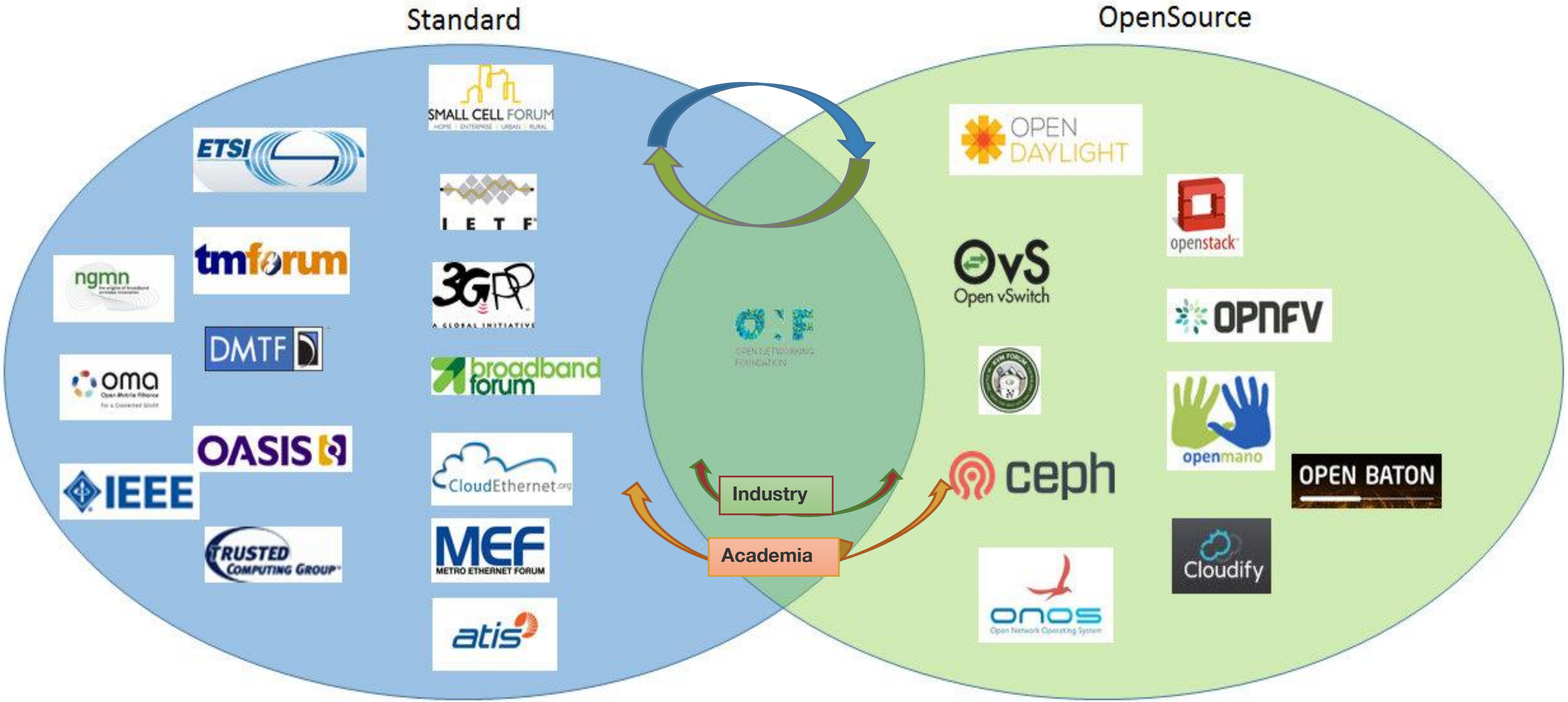


Open Source minded Research @ INTRIG

INFORMATION & NETWORKING TECHNOLOGIES RESEARCH & INNOVATION GROUP

- Extensive use of open source in support of research activities.
  - Etherpad, Owncloud, Docker, Gitlab, K3, etc.
- Use of research dev repository (github private->public)
- Publish versions of papers (in submission/accepted) at Arxiv, etc;
- Make “source code” of papers public available (e.g. Overleaf, github, etc.)
- Release all research data to allow third parties to re-use and reproduce
  - Github, wiki, how-to reproduce, Jupyter notebooks
- Create and contribute to community-oriented open source projects
  - https://github.com/intrig-unicamp/

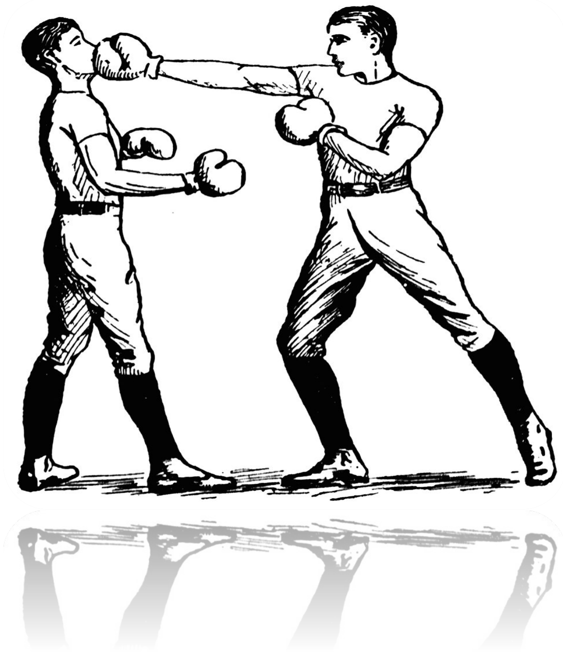
Networking Standard Developments & Open Source Organizations



Source: SDN IEEE Outreach, 2015: http://sdn.ieee.org/outreach

Network Softwarization

(i.e. SDN + NFV + IBN + xyz)



Old / Existing

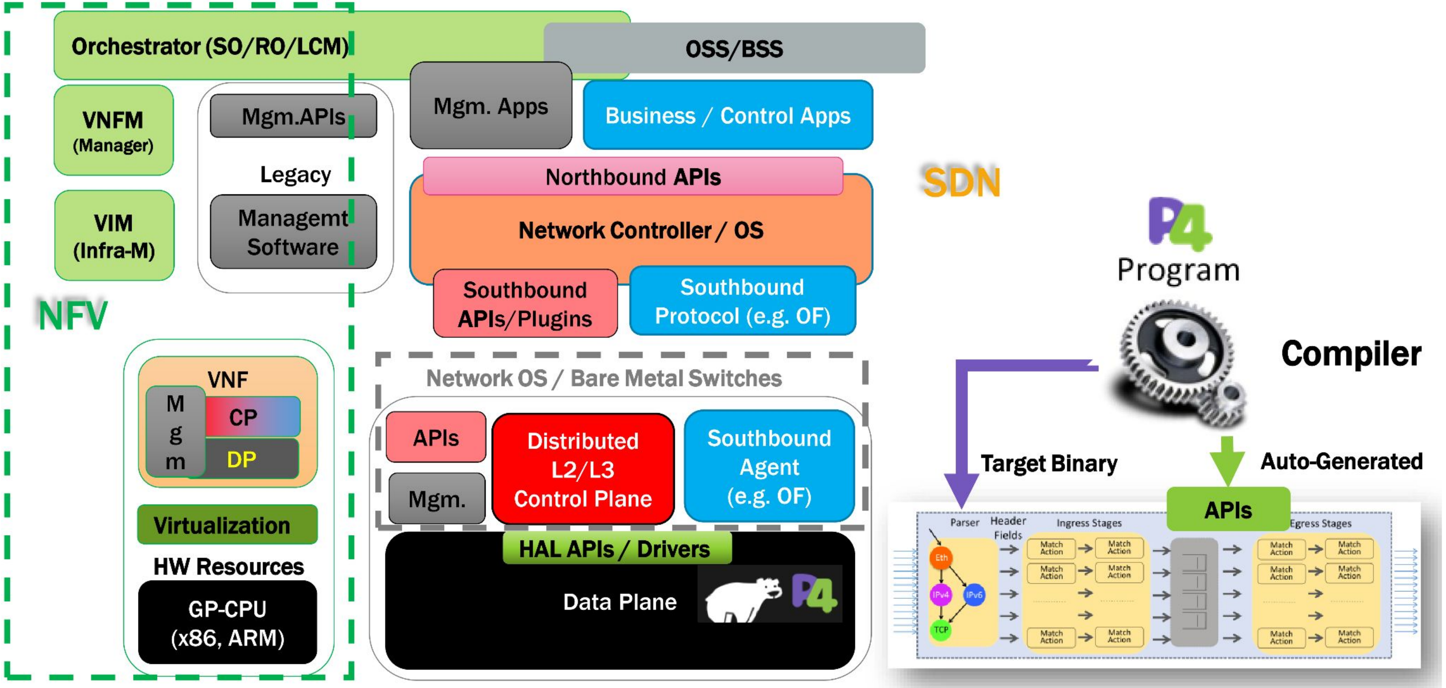
- CLIs & Manual labour
- Closed Source
- Vendor Lead
- Classic Network Appliances (HW)

New / Softwarized

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

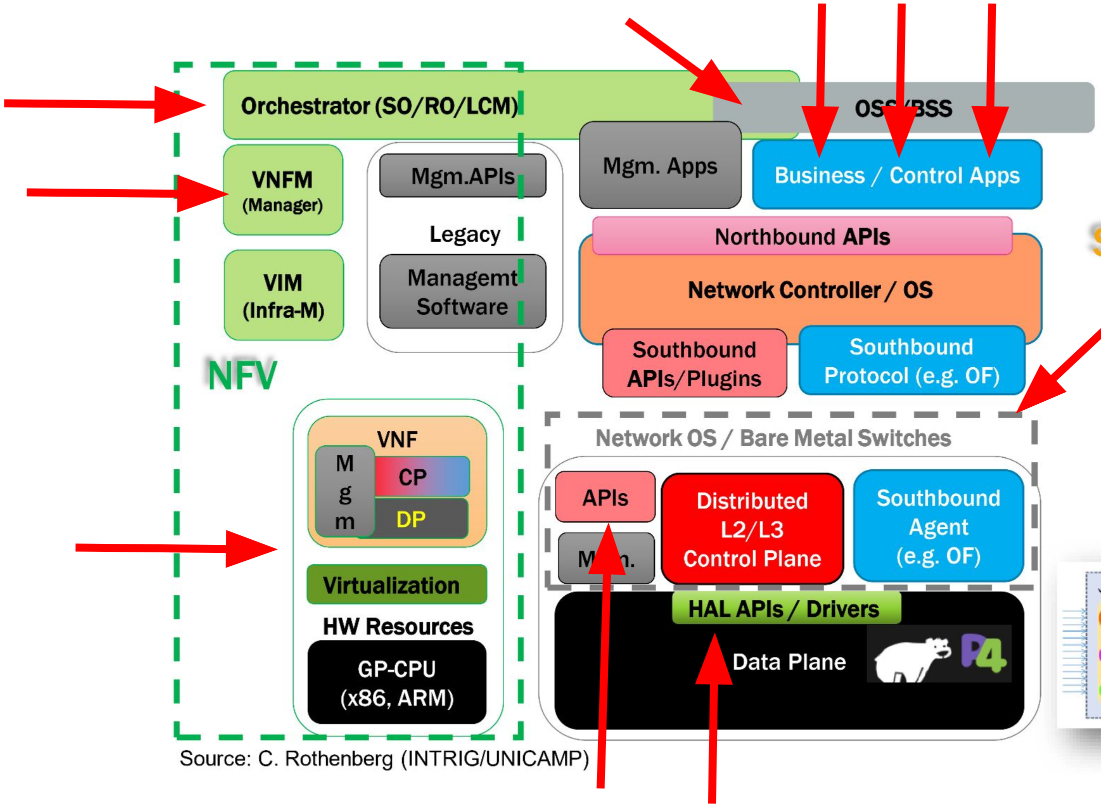
Source: Adapted from Kyle Mestery, Next Generation Network Developer Skills

Models & Approaches to Program / Refactor the Netsoft Stack



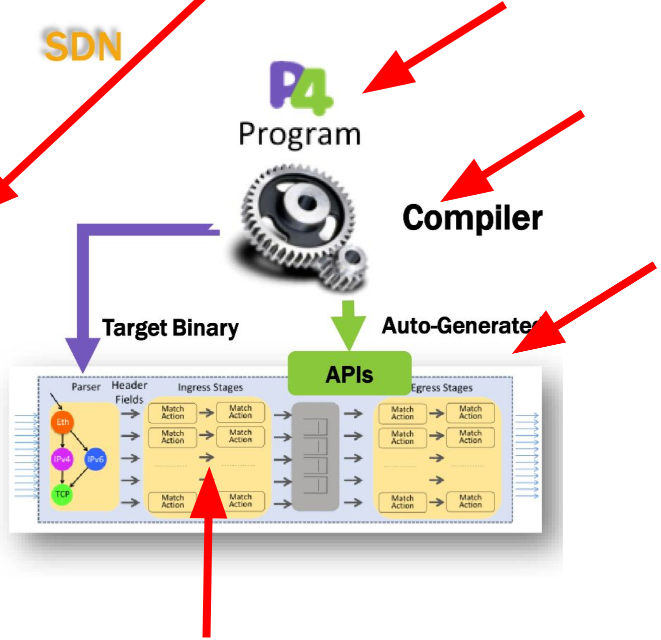
Source: C. Rothenberg (INTRIG/UNICAMP)

Models & Approaches to Program / Refactor the Netsoft Stack



Source: C. Rothenberg (INTRIG/UNICAMP)

e.g. libfluid ONF Driver Competition
http://opennetworkingfoundation.github.io/libfluid/



Research Projects and Open Source Results

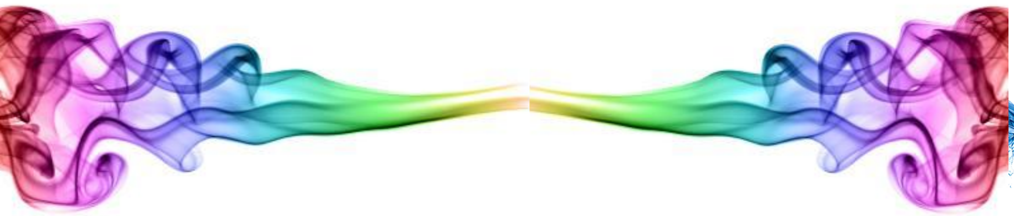
More info: https://github.com/intrig-unicamp/ https://github.com/chesteve/

(Selected) Technical lead of relevant open source projects:

- Gym, Testing Framework for Automated NFV Performance Benchmarking (2017 -)
  - https://github.com/intrig-unicamp/gym
  - IETF BMWG, Internet-Draft draft-rosa-bmwg-vnfbench
- Mininet-WiFi, Emulating Software Defined Wireless Networks (2015 -)
  - https://github.com/intrig-unicamp/mininet-wifi/
- libfluid, winner of the ONF Driver Competition (Mar/2014)
  - http://opennetworkingfoundation.github.io/libfluid/
- softswitch13, first OpenFlow 1.2 and 1.3 soft switch, controller, and testing (2011 - 2013)
  - https://github.com/CPqD/ofsoftswitch13
- Mini-CCNx, fast prototyping and experimentation of CCN networks (2013 -)
  - https://github.com/carlosmscabral/mn-ccnx
- RouteFlow, first IP routing architecture for SDN (2010 -). 3 x GSOC projects
  - https://github.com/routeflow/

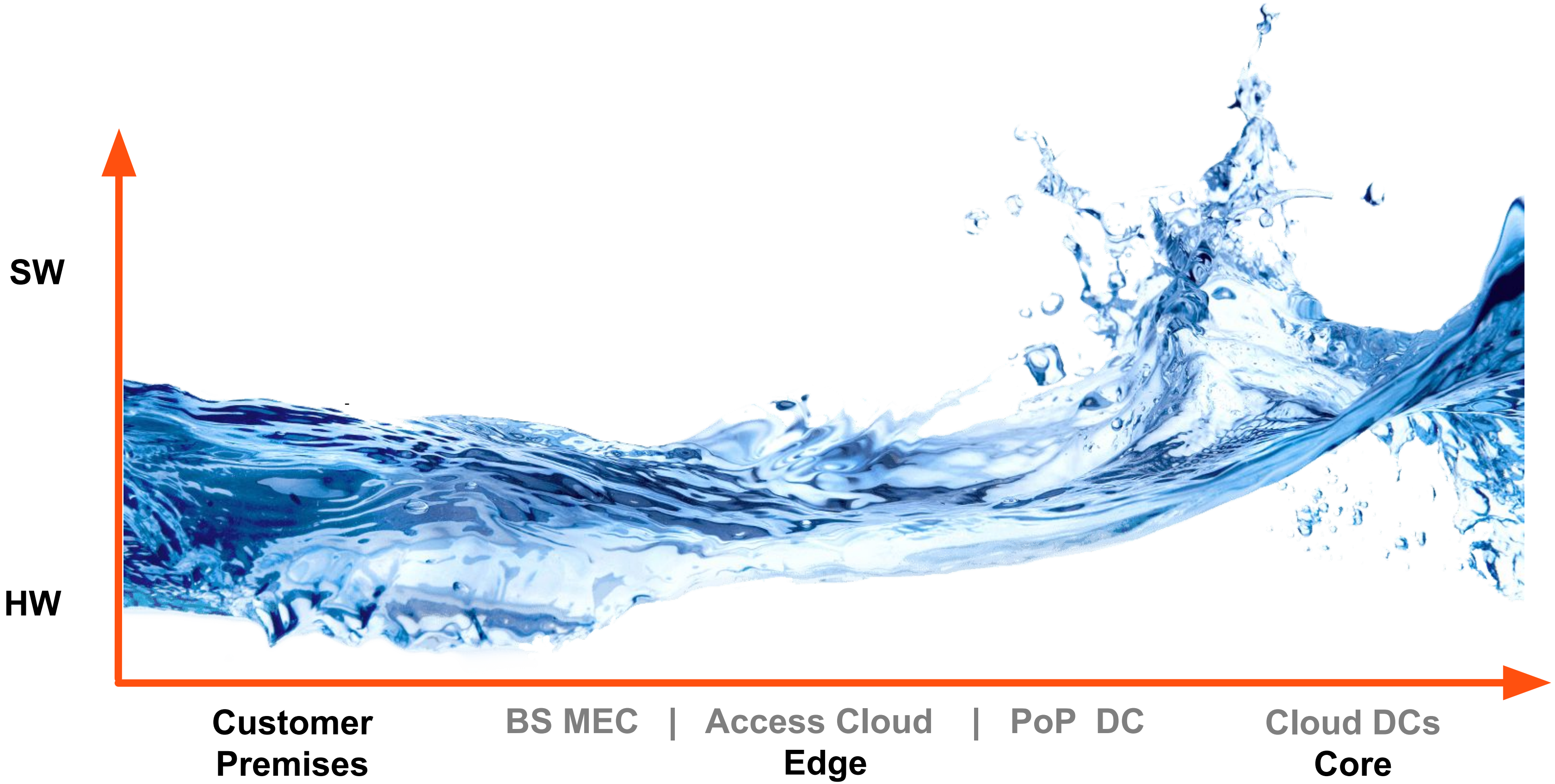
Fluid Network Control and Data Planes

An observation on ongoing technology and research trends

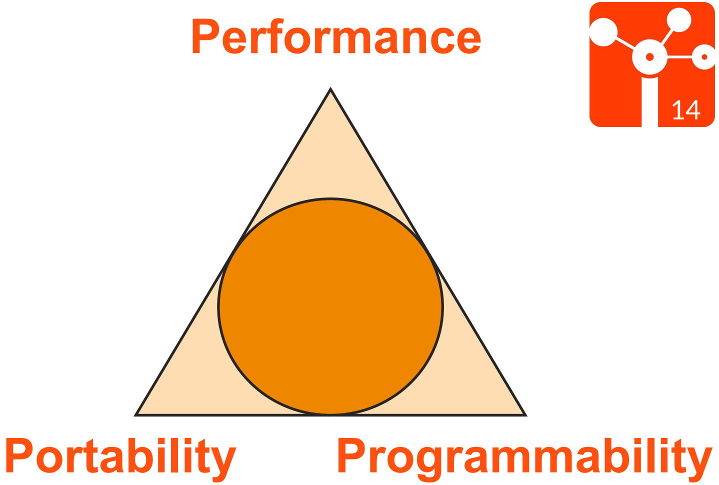


The **Fluid Networking** landscape

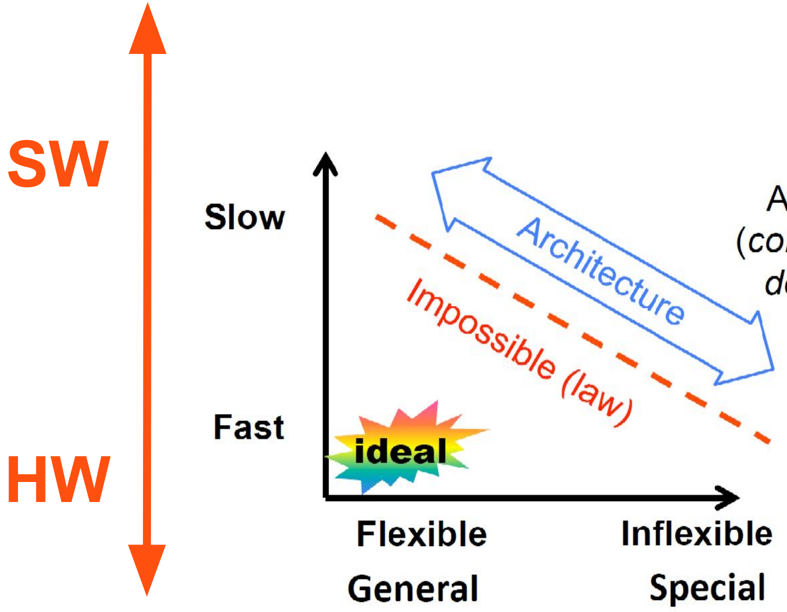
The **Fluid Networking** landscape



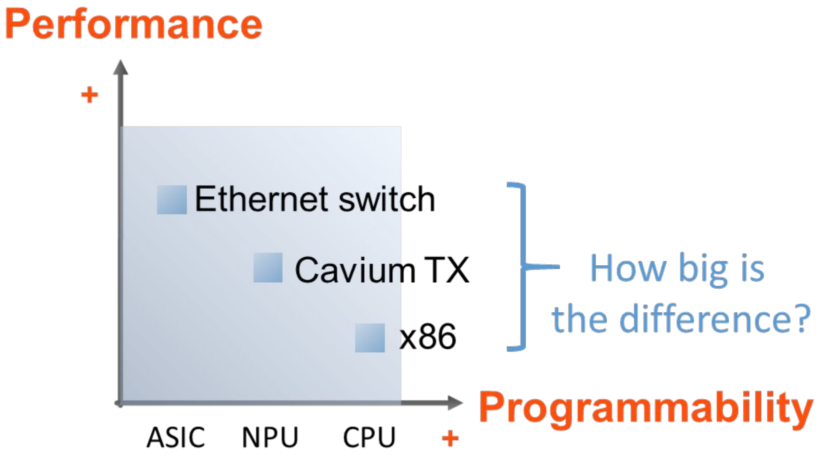
Fluid Networking:
HW-SW Continuum



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



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



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

Fluid Networking:
HW-SW Advances

Flexibility*
(programmability + portability)

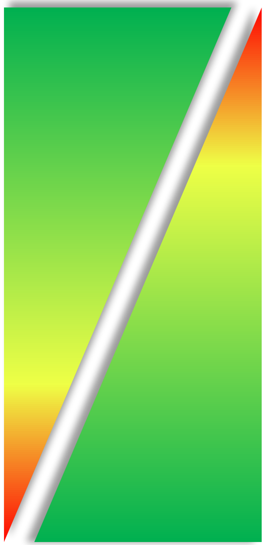
TABLE II
TECHNICAL CONCEPTS AND THEIR SUPPORT OF FLEXIBILITY IN NETWORKS. (✓: MAIN TARGET)

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	-	-	✓

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



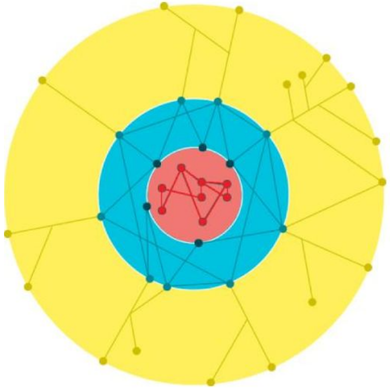
Performance***

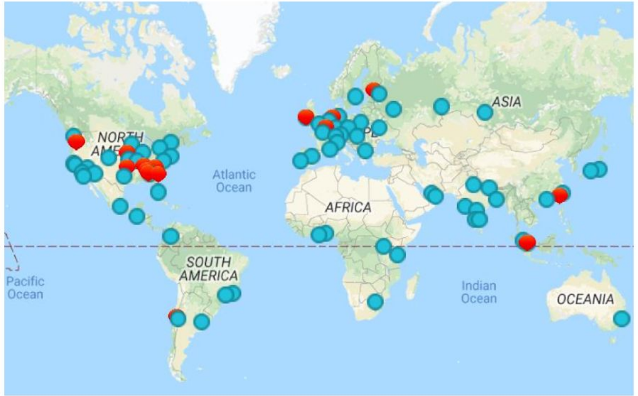
* M. He et al. **Flexibility in Softwarized Networks: Classifications and Research Challenges**. IEEE Survey & Tutorials, 2019

** 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 / Fog & Cloud Continuum



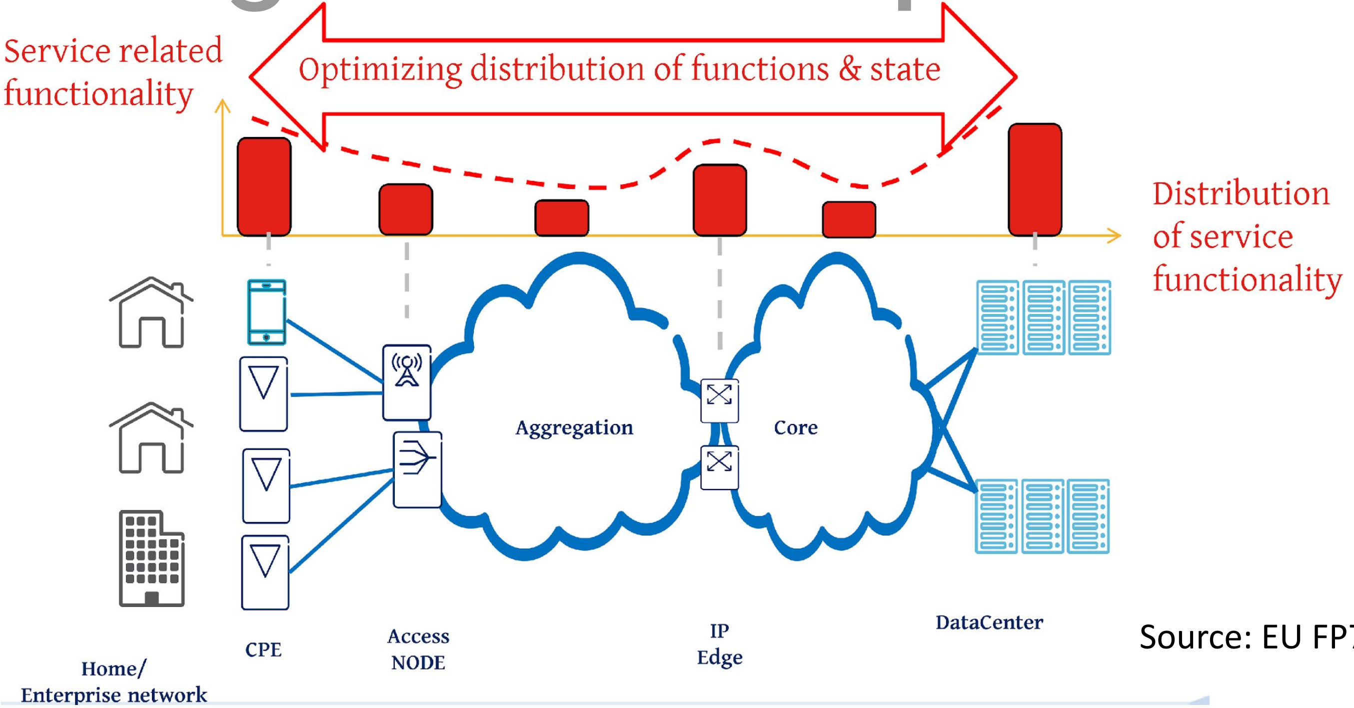


- 15 Data centers
- 100 Points of Presence (PoPs)
- 1000+ Edge nodes

Source: Google Cloud Infrastructure

Customer Premises	BS MEC | Access Cloud | PoP DC Edge	Cloud DCs Core

Fluid Networking:
Optimizing the E2E Compute Pools

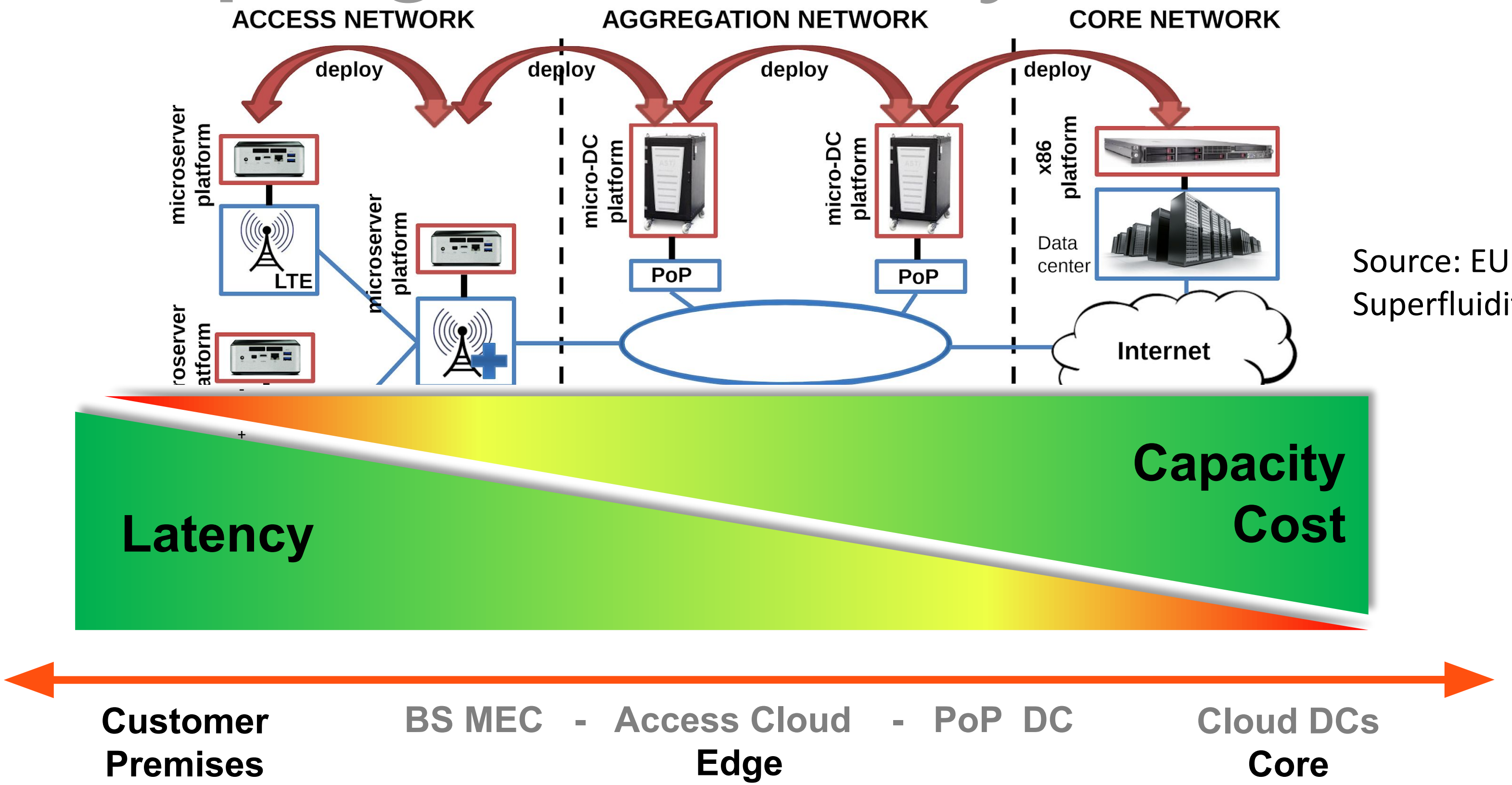


Source: EU FP7 UNIFY

Customer Premises | BS MEC | Access Cloud | PoP DC | Cloud DCs

Edge | **Core**

Fluid Networking:
Decoupling functionality / location

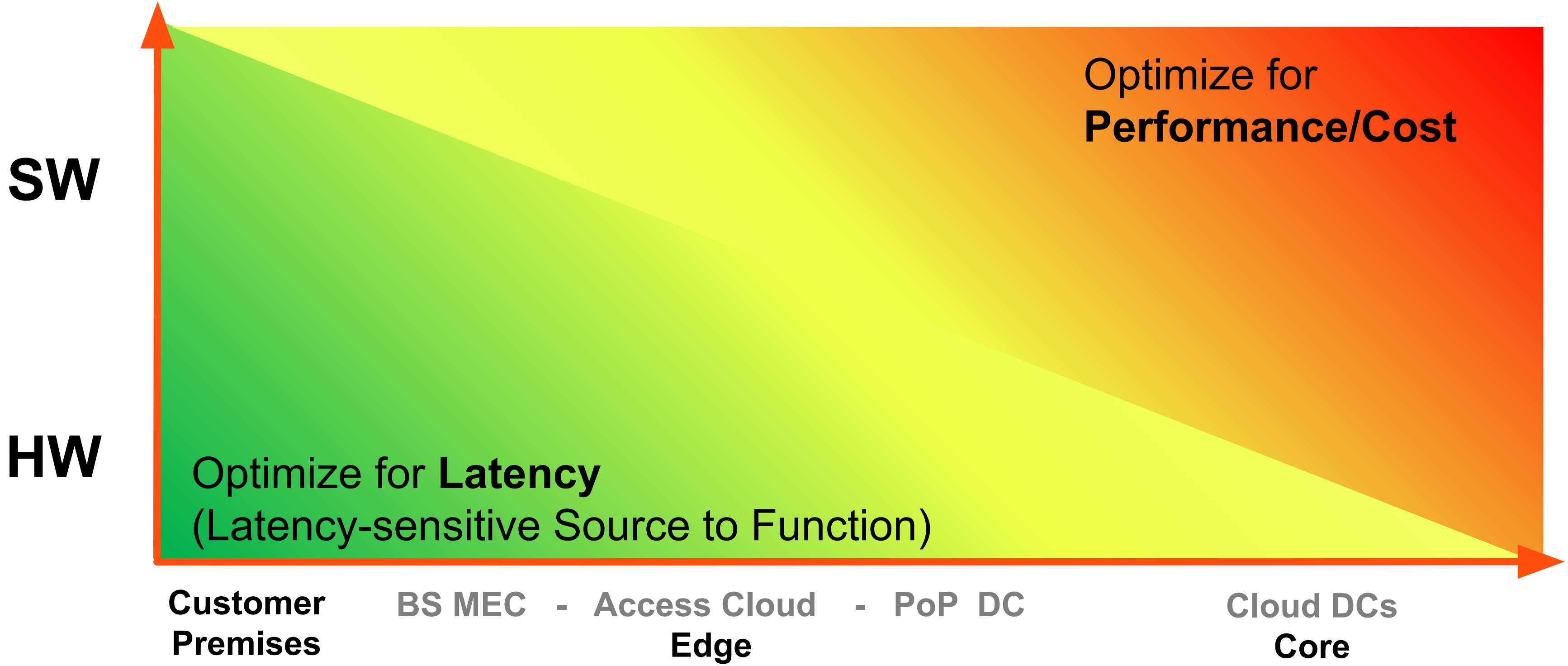


Source: EU Superfluidity

The **Fluid Networking** landscape

Control plane component(s)

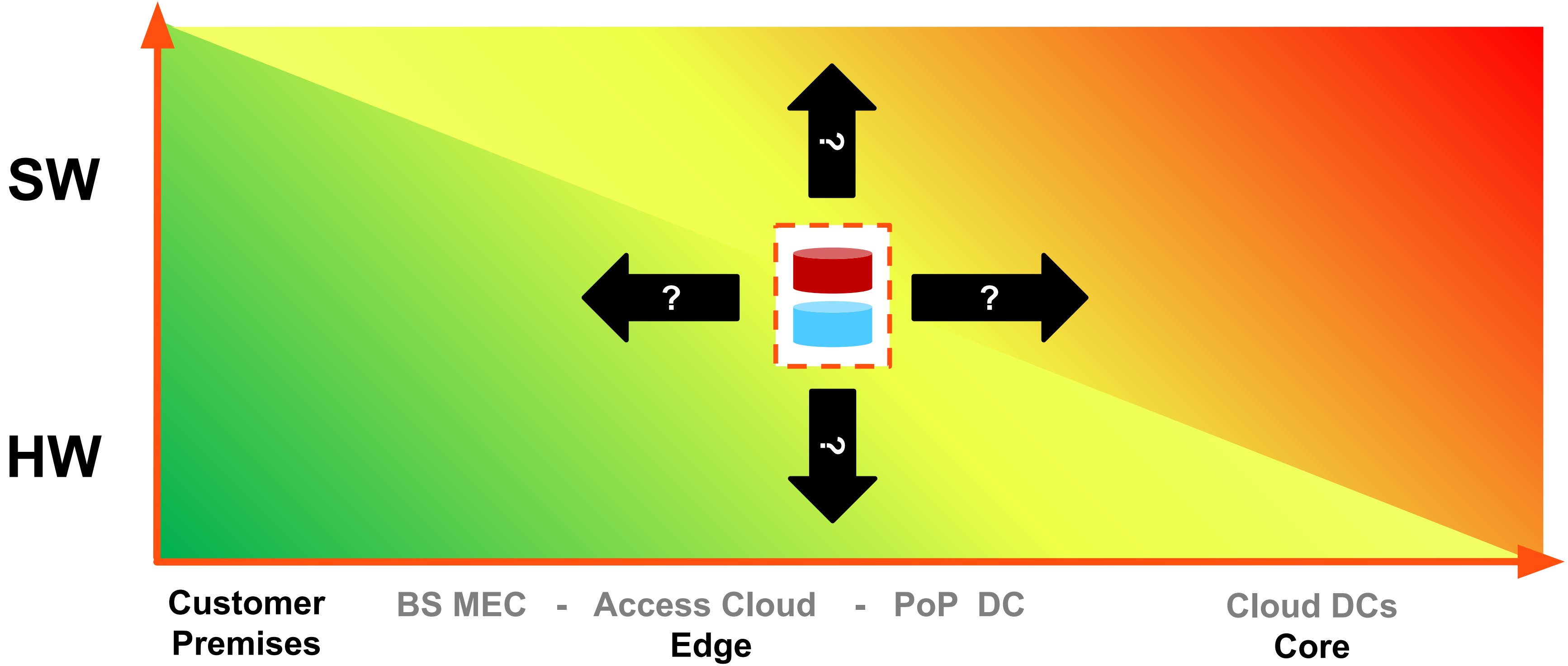
Data plane component(s)



The **Fluid Networking** landscape

Control plane component(s)

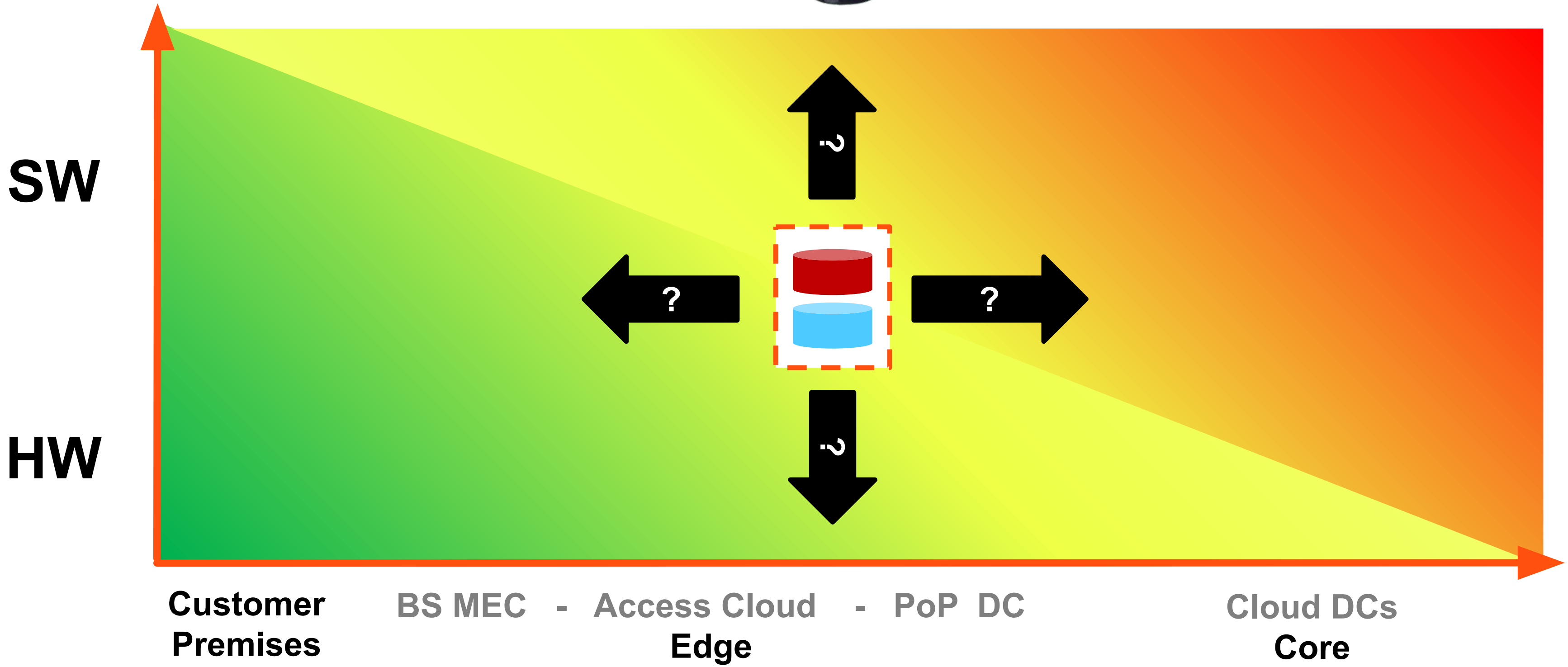
Data plane component(s)



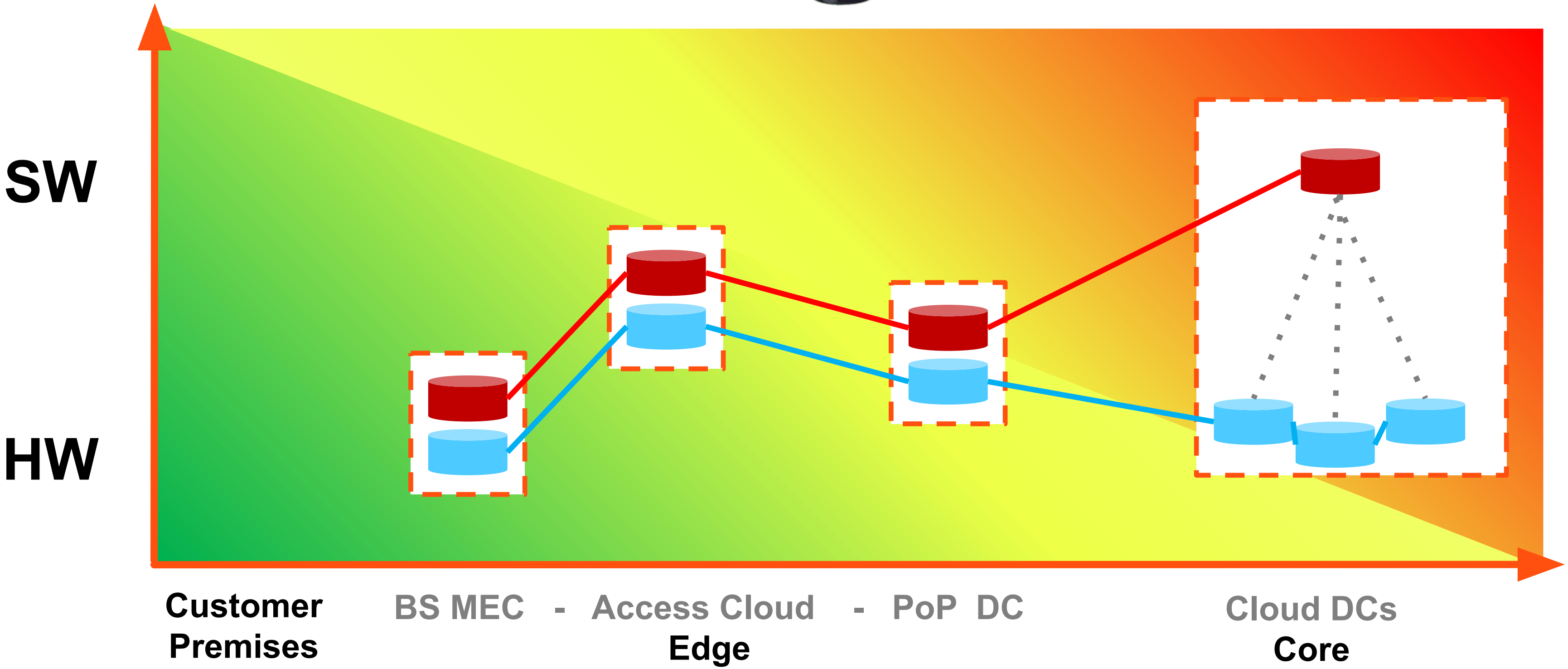
The **Fluid Networking** landscape

Control plane component(s)

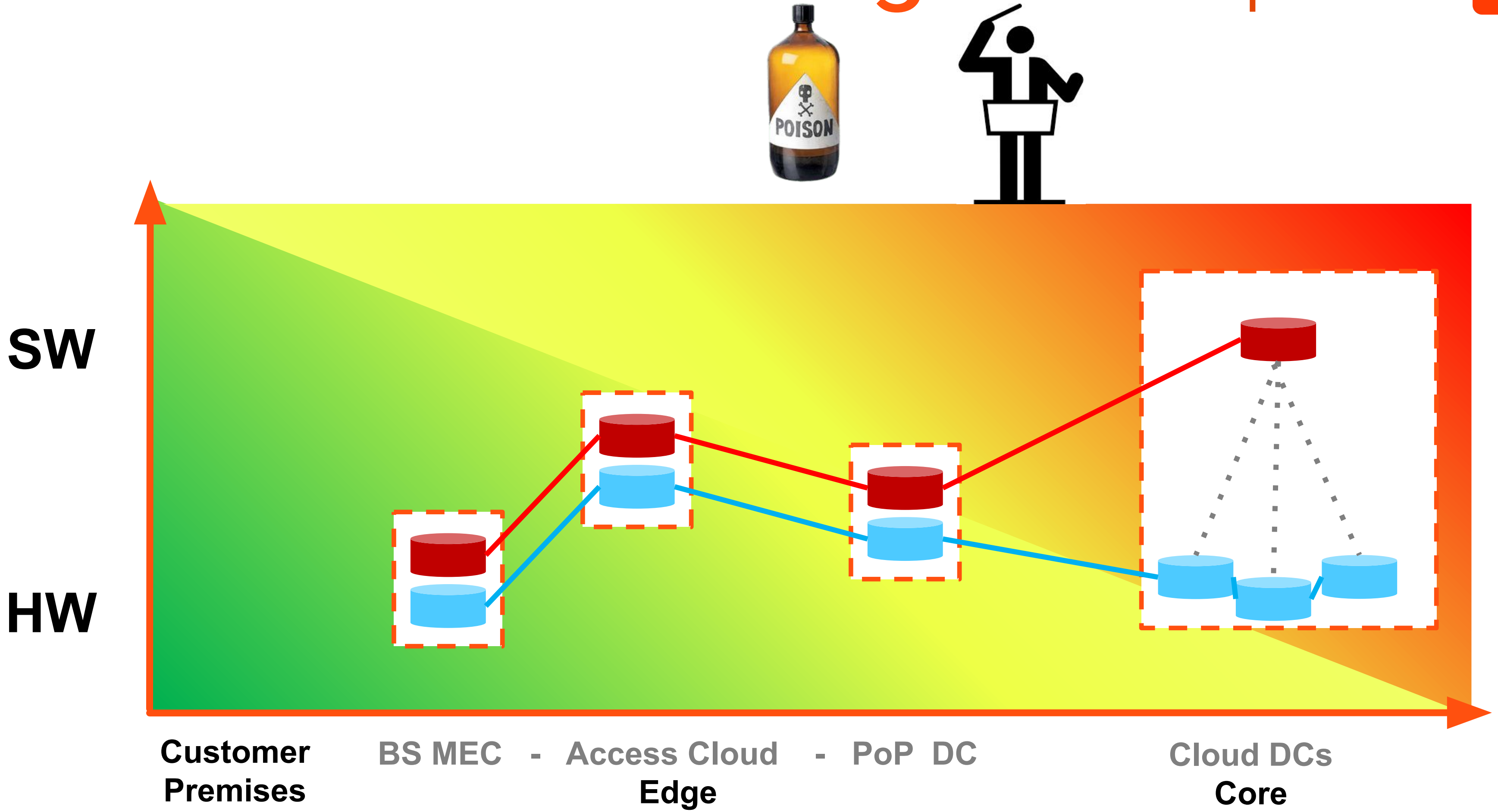
Data plane component(s)



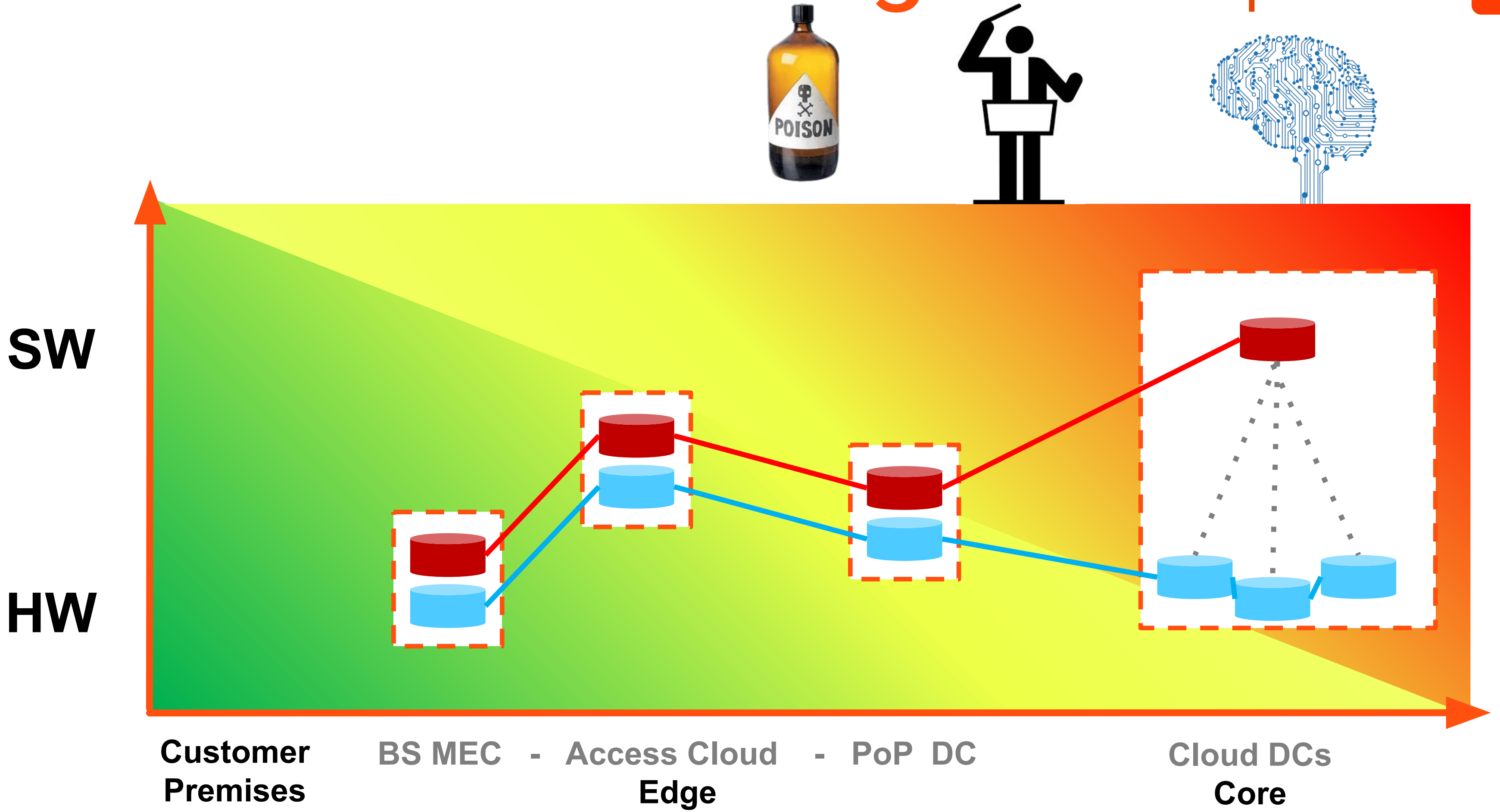
The **Fluid Networking** landscape



The **Fluid Networking** landscape



The **Fluid Networking** landscape

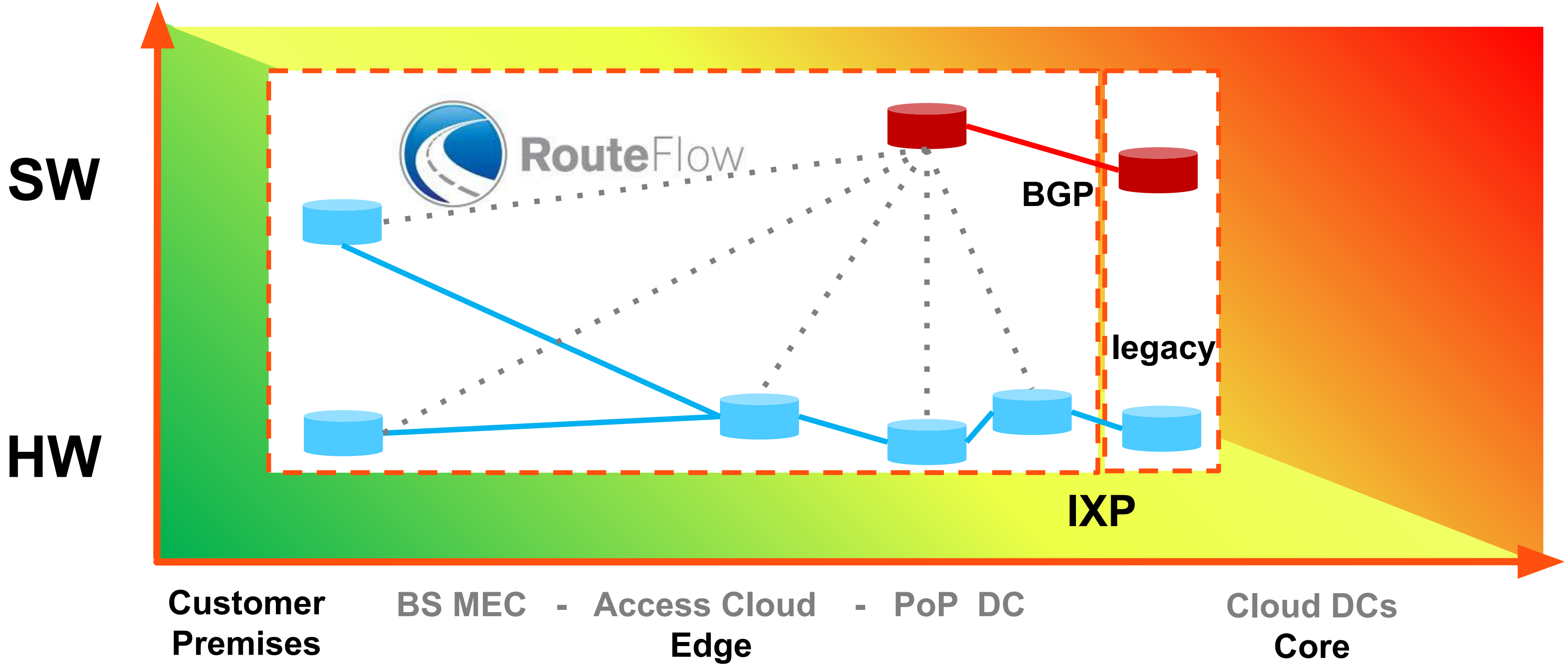


Featured research topics

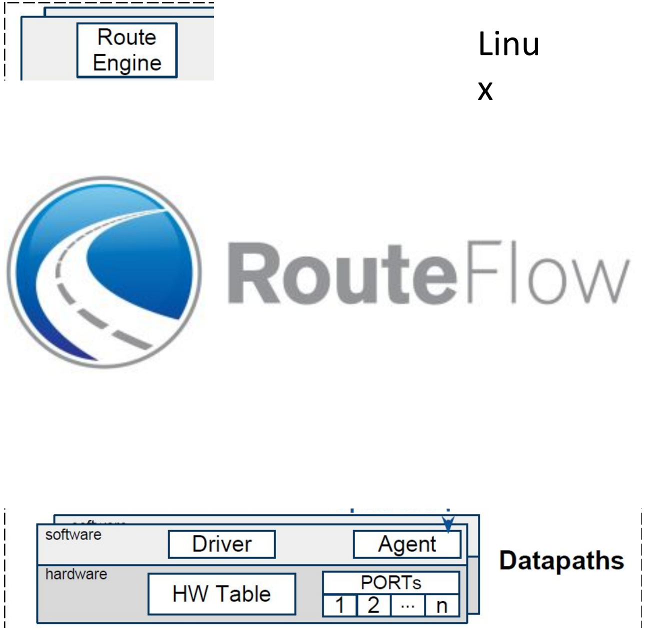
Conversation with RedHat

Instances of **Fluid Network Planes**

RouteFlow (2010 -)



RouteFlow: High-level Architecture



Control Plane

Glue

OpenFlow

Data Plane

Open Source SW | RouteFlow SDN/OpenFlow architecture

Quagga, XORP, BIRD

Linux

Open vSwitch, softswitch13

LXC, QEMU, XEN, libvirt

MongoDB, redis

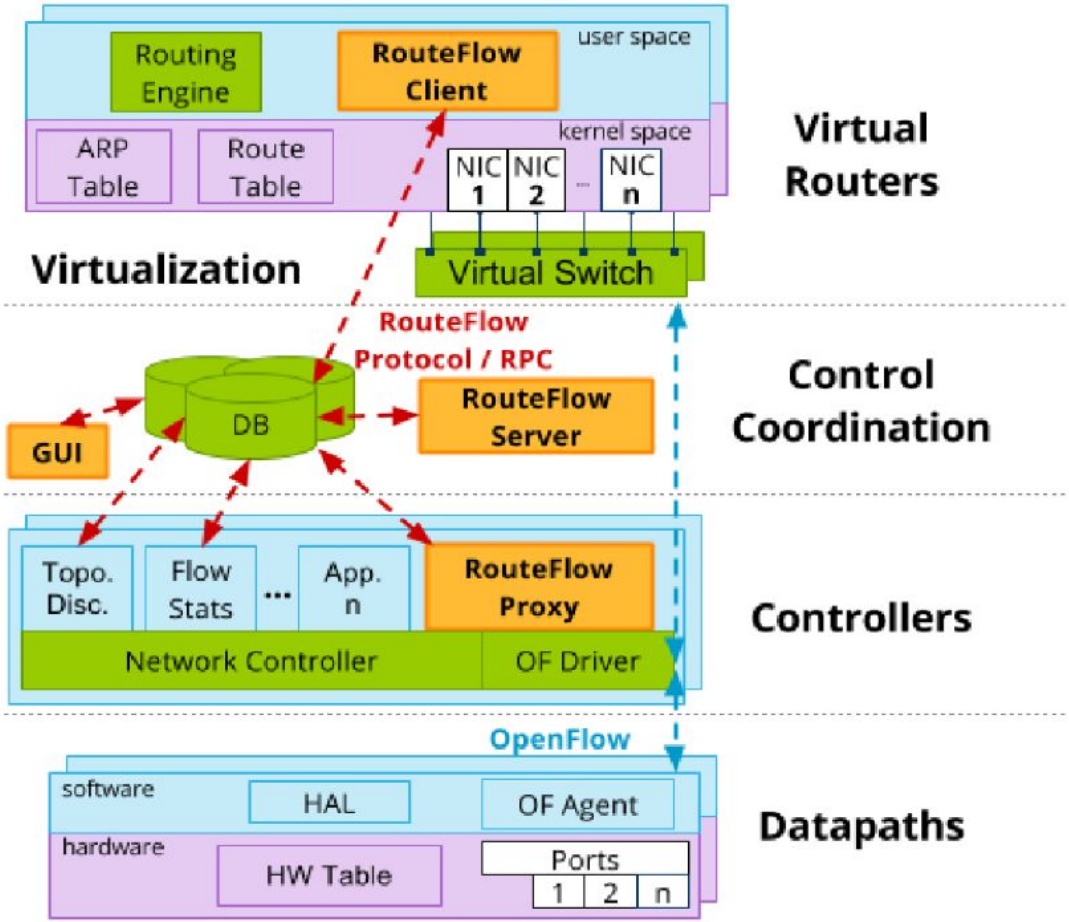
jQuery, JIT

NOX, POX, Floodlight, Ryu

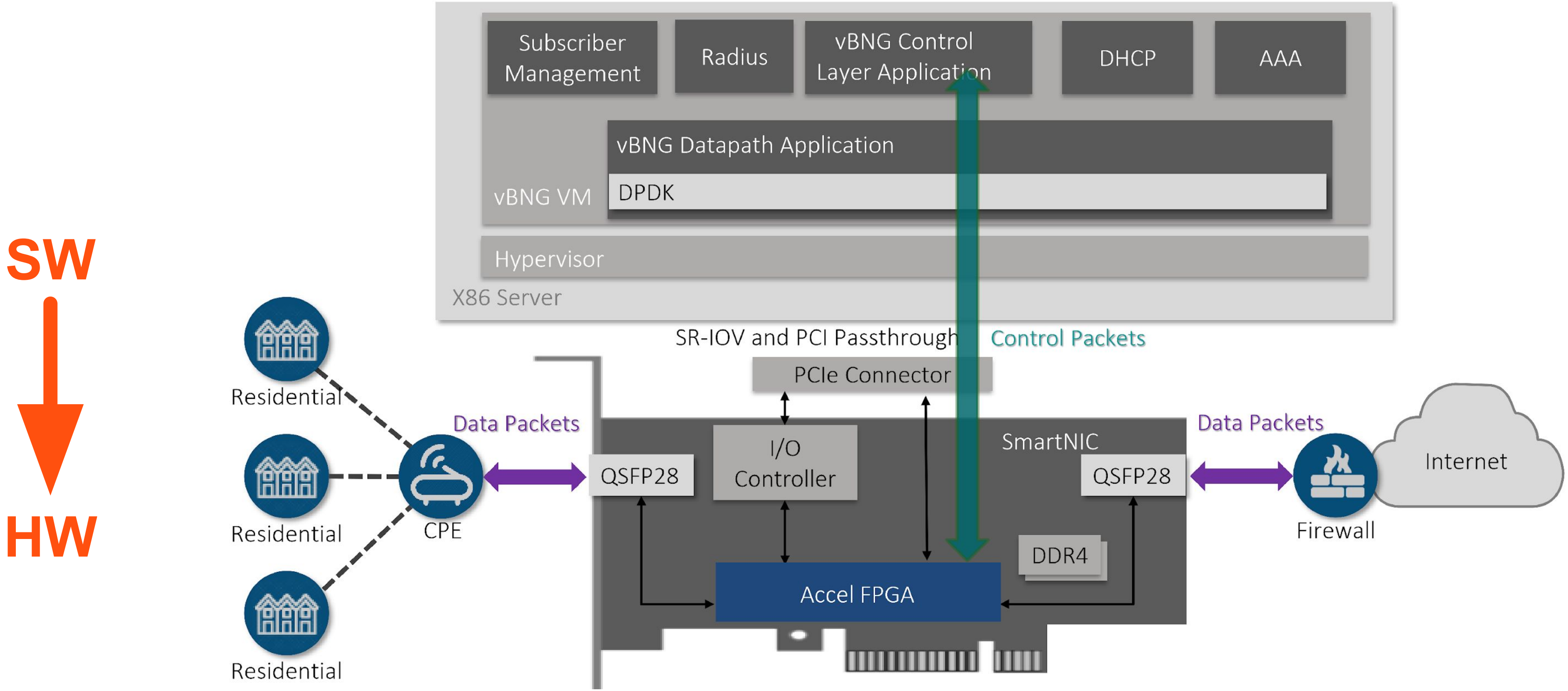
ONF driver

FlowVisor

Mininet, OVS, softswitch13, Pica8 OS

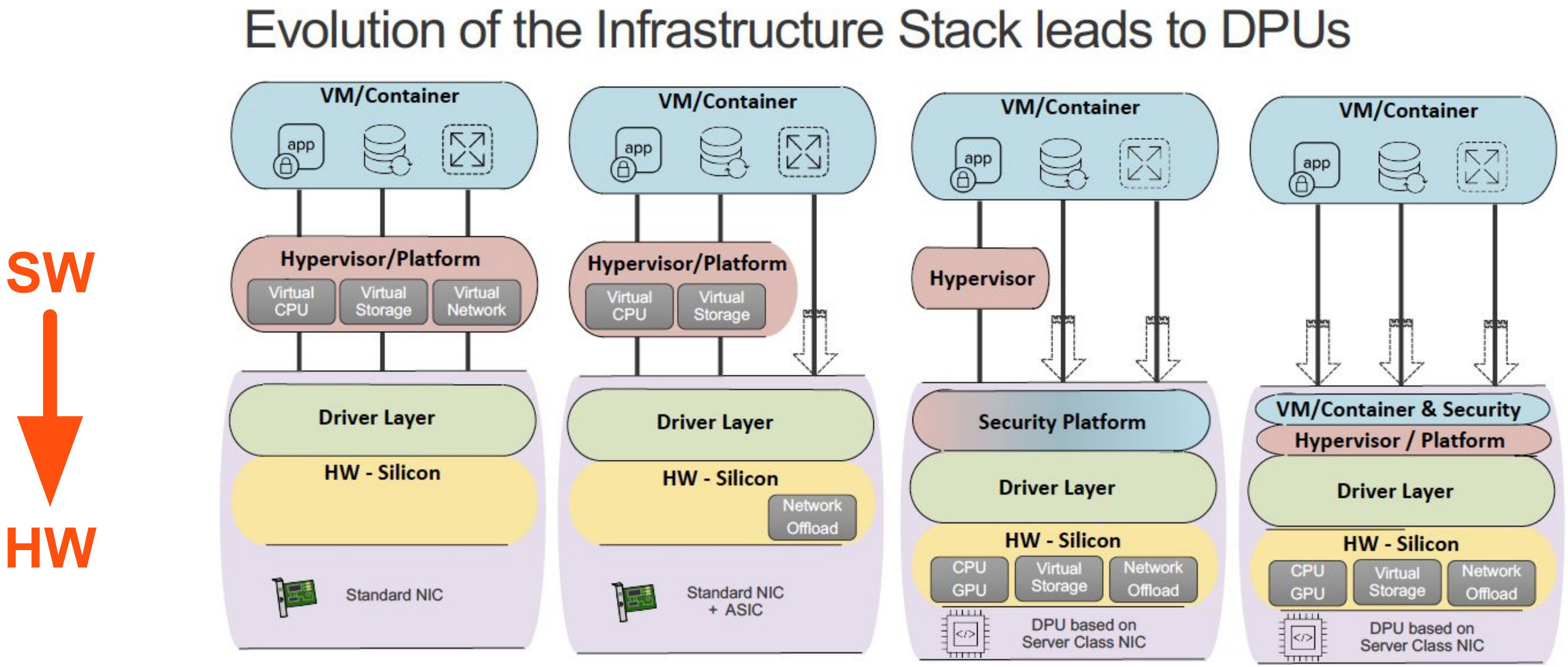


VNF offloading to Hardware



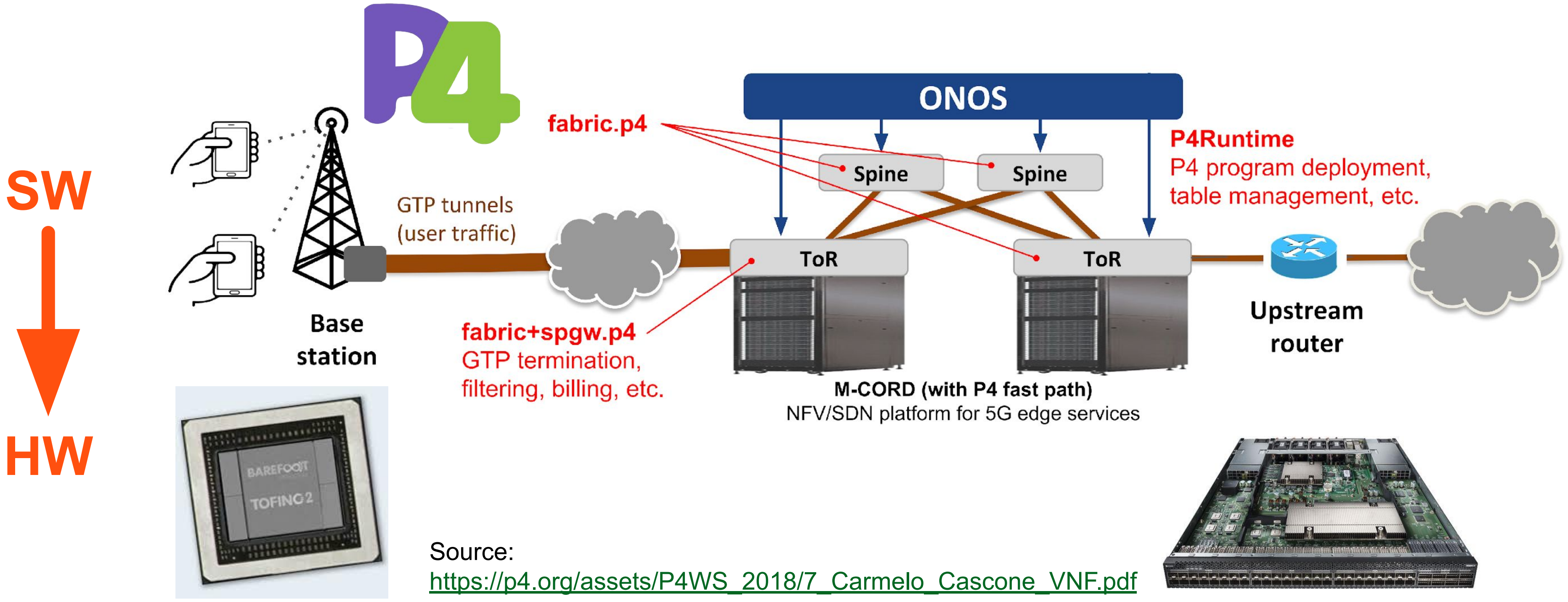
Source: https://www.dpdk.org/wp-content/uploads/sites/35/2018/12/Kalimani-and-Barak-Accelerating-NFV-with-DPDK-and-SmartNICs.pdf

SW/HW Evolution of the Infrastructure Stack

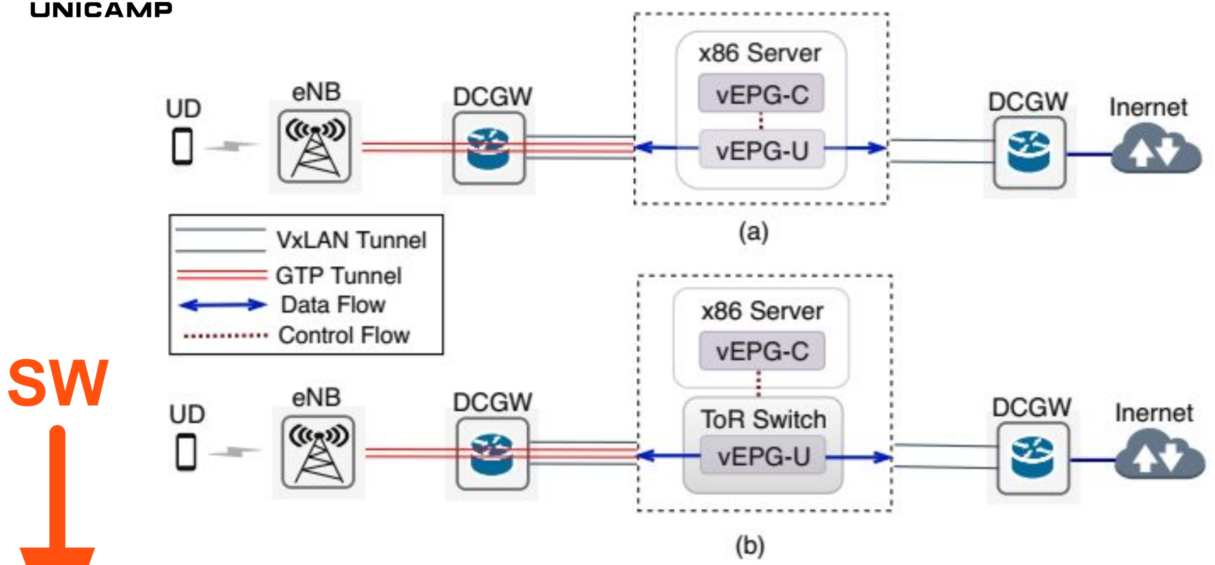


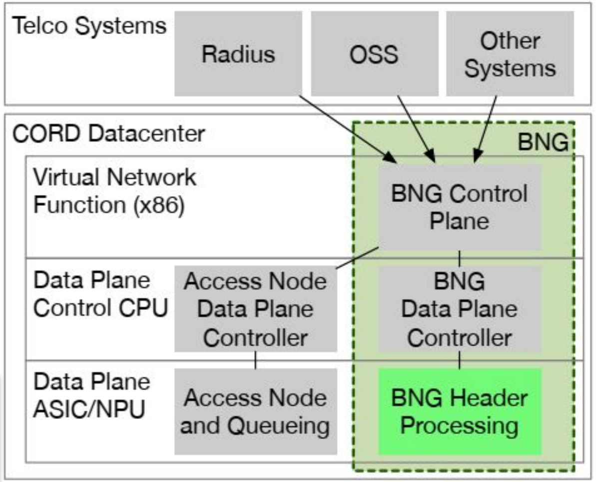
Source: **F5**

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



Source: https://p4.org/assets/P4WS_2018/7_Carmelo_Cascone_VNF.pdf





. 3: CORD-Service Edge component overview.

Suneet Kumar Singh, Christian Esteve Rothenberg, Gyanesh Patra, Gergely Pongrácz. **Offloading Virtual Evolved Packet Gateway User Plane Functions to a Programmable ASIC**. In 1st ACM CoNEXT Workshop on Emerging in-Network Computing Paradigms (ENCP'19)

Pattam Gyanesh Patra, Fabricio Rodriguez, Juan Sebastian Mejia, Daniel Lazkani Feferman, Levente Csikor, Christian Esteve Rothenberg, Gergely Pongrácz. **Towards a Sweet Spot of Dataplane Programmability, Portability and Performance: On the Scalability of Multi-Architecture P4 Pipelines**. In IEEE JSAC, 2018

Computation in the Network

SW

HW

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

IRTF Computation in the Network (COIN)

Customer Premises

BS MEC - Access Cloud - PoP DC

Edge

Cloud DCs

Core

UNICAMP

INTRIG INFORMATION & NETWORKING TECHNOLOGIES RESEARCH & INNOVATION GROUP

ERICSSON

Fabricio Rodriguez, Levente Csikor, Carlos Recalde, Christian Esteve Rothenberg, Gergely Pongrácz. **Towards Low Latency Industrial Robot Control in Programmable Data Planes.** In IEEE NetSoft 2020.

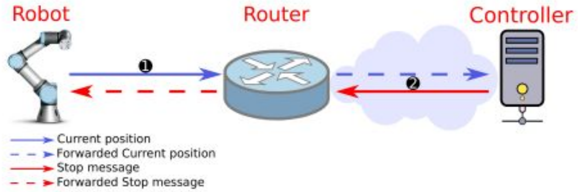


Fig. 4: Traditional scenario without in-network control.

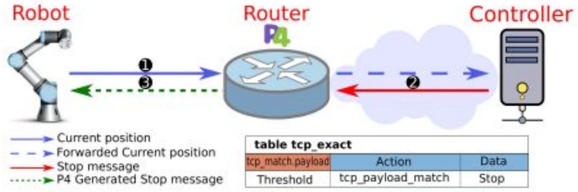


Fig. 5: In-network P4-based implementation.

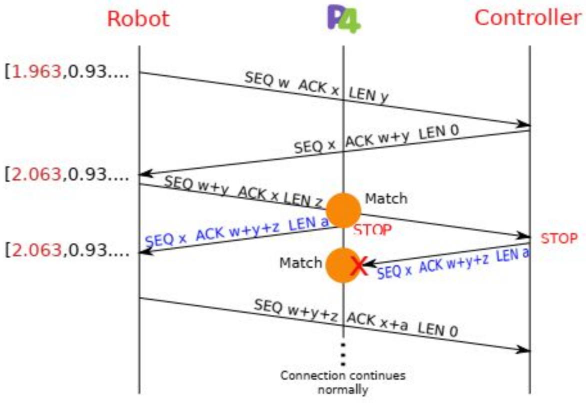
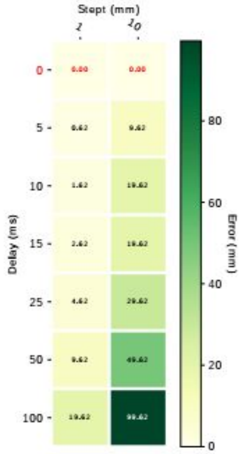


Fig. 6: TCP session approach.



(b) Steps of 1 - 10 mm

Fig. 8: Stop position error without in-network actions. Acceleration of $(30°/s^2)$.

Customer Premises

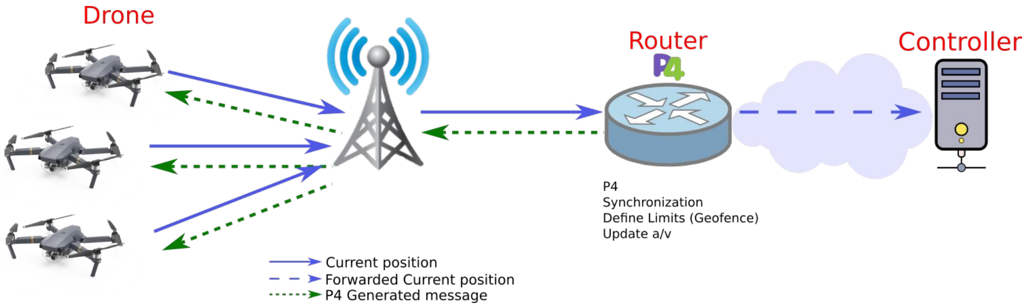
BS MEC - Access Cloud - PoP DC

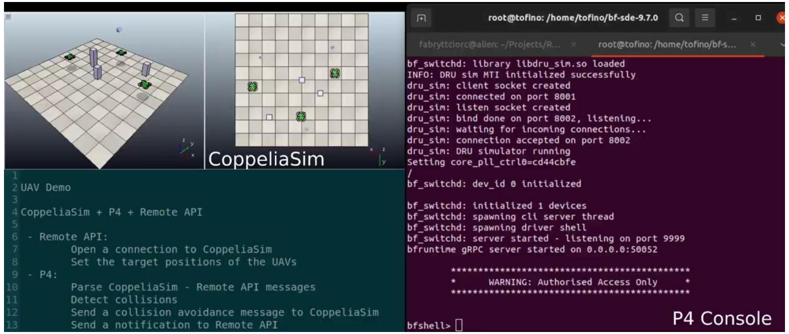
Edge

Cloud DCs

Core

In-Network Centralized Done Collision Avoidance Algorithm in Programmable Data Planes





https://www.youtube.com/watch?v=3ptCGeBXsvo

Fabricio Rodriguez et al. **P4 Workshop 2021** (Award Winner of most novel non-networking use of P4)
Fabricio Rodriguez et al. **GLOBECOMM 2022 Demo.**

Agenda

- **Fluid Network Control and Data Plane Research**
  - Concept Overview
  - Instances

- **Active Research:** Selected open-source fueled research threads
  - **Hybrid-P4-5G:** User Plane Functions in Programmable SW/HW Stacks
  - **P7** (P4 Programmable Patch Panel): 100G network emulation
  - **EFFECTOR:** DASH QoE Evaluation Framework with 5G Datasets
  - **QoEyes:** Virtual Reality Streaming QoE Estimation through ML and INT
  - **PoD acceleration:** Offloading container networking & security functions

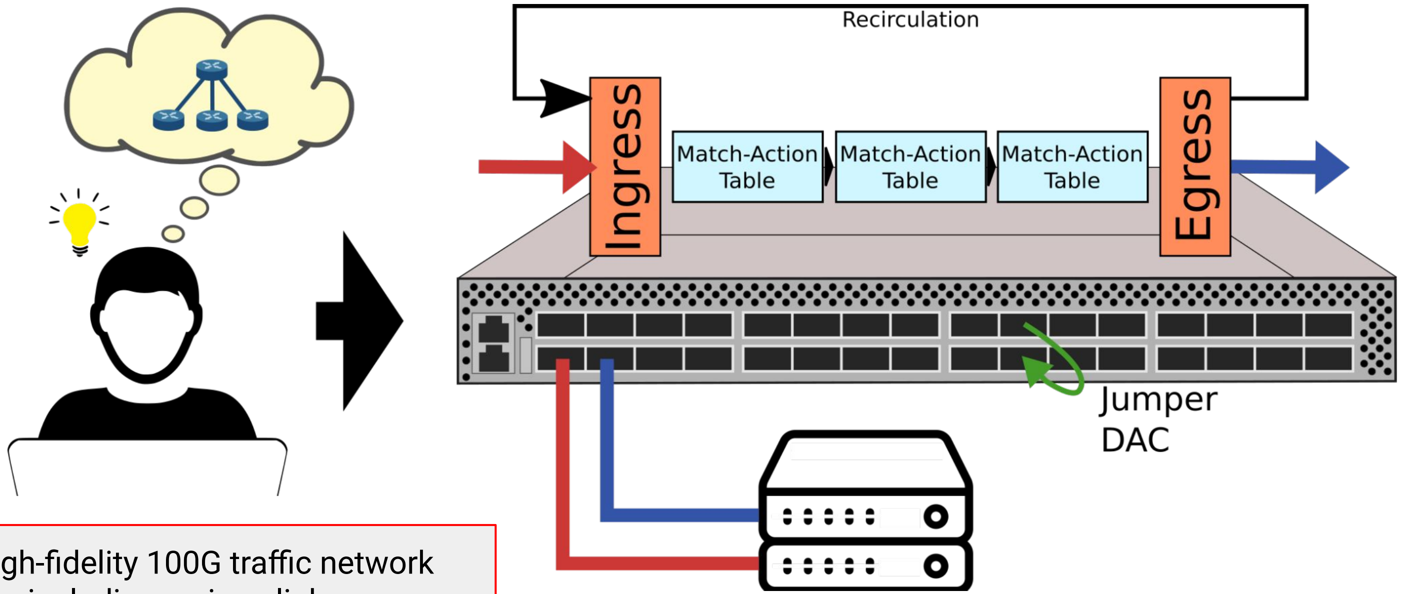
- **The SMARTNESS 2030 research center (2023 - 2033)**
  - Vision & Research Collaboration opportunities with RedHat

Featured research topics

Conversation with RedHat

P7 (P4 Programmable Patch Panel) Instant 100G Emulated Network Testbed

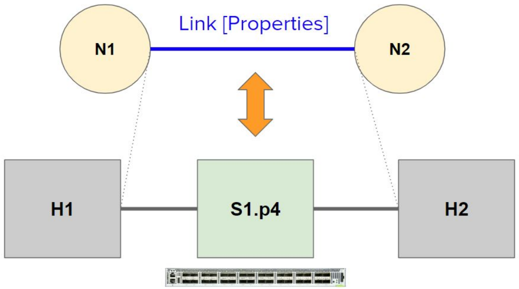
P7



P7 is a high-fidelity 100G traffic network emulation, including various link characteristics such as latency, jitter, packet loss, and bandwidth, as well as the option to customize network topologies.

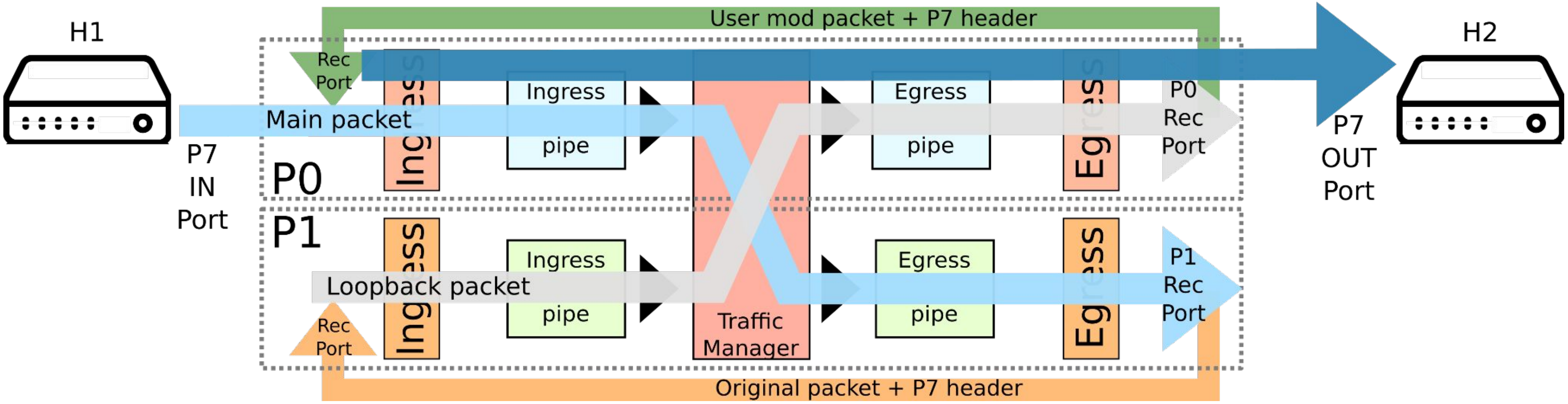
Everything implemented in a single P4 switch

Link characteristics and P4/TNA implementation approaches



Link Connectivity	Jumper cabling with internal Tag Intern Recirculation + internal Tag
Latency [ms]	Internal timer + recirculation TM + Pipelines recirculation
Jitter [ms]	Hash to determine recirculation times Lookup table with mathematical functions
Packet loss [%]	Random function to determine the probability to discard packets Realistic packet loss model
Bandwidth	Rate limit TNA TM feature Port configuration and shaping

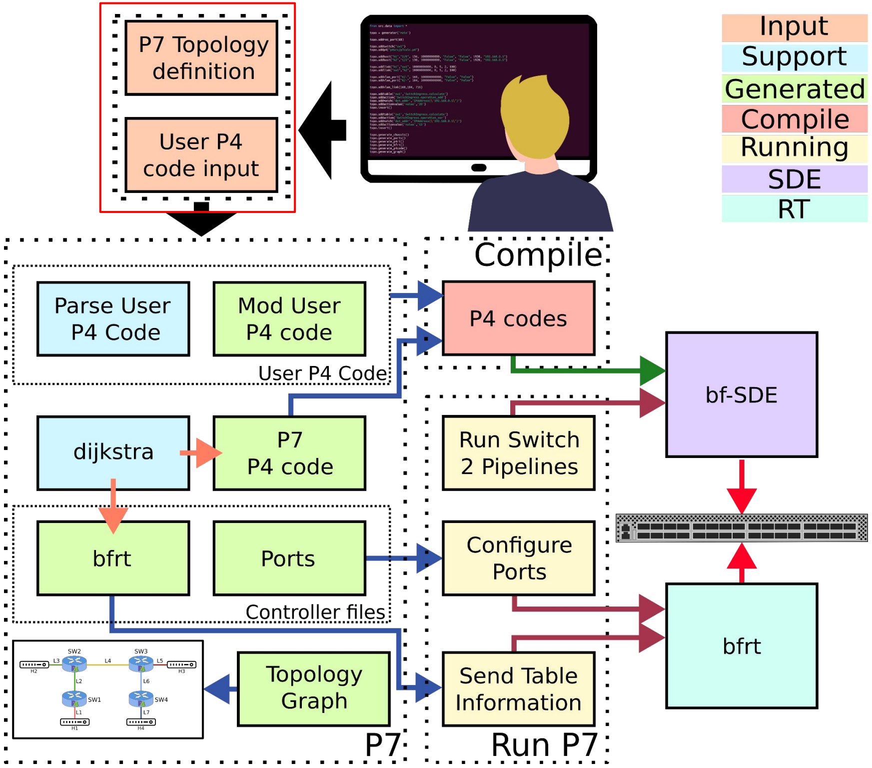
P7 multiple pipelines approach



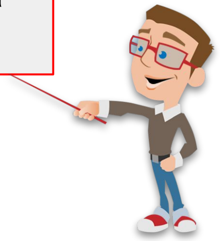
We propose a solution where a dedicated pipe runs the P7 P4 code, and a separate pipe runs the user-defined P4 code

We send the packet in the P7 pipe (P0) to the pipe where the user-defined P4 code is running (P1) using recirculation

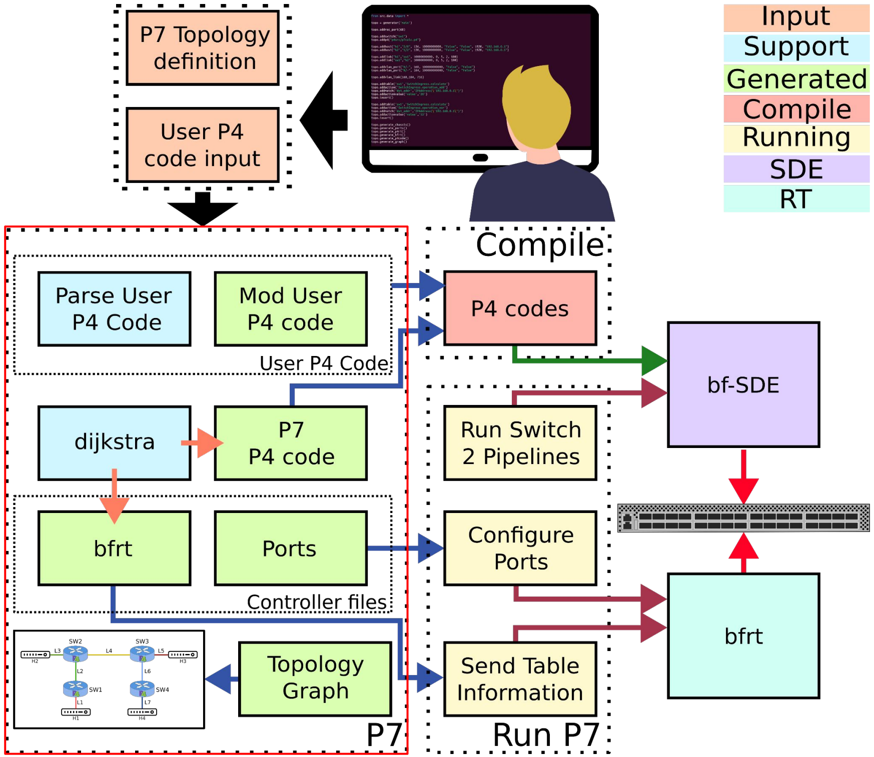
P7 Architecture & User workflow



The user defines the topology and sets a custom P4 code.



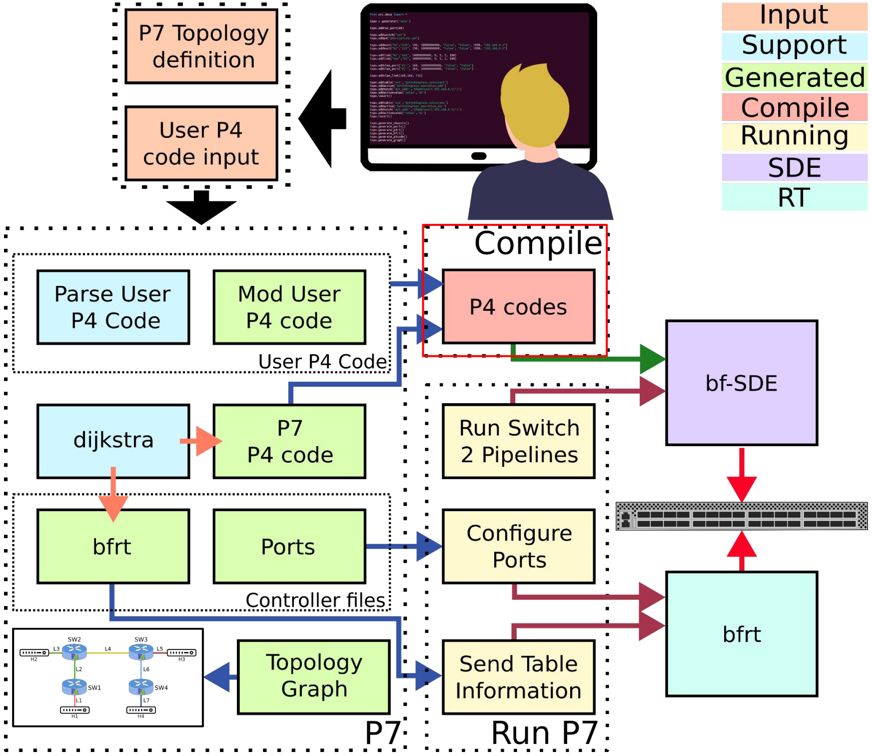
P7 Architecture & User workflow



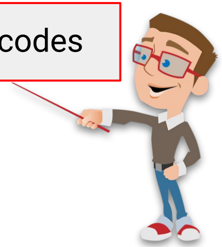
P7 processes the data and generates the necessary files:

- P7 P4 code
- User P4 code
- Tables information
- Ports configuration

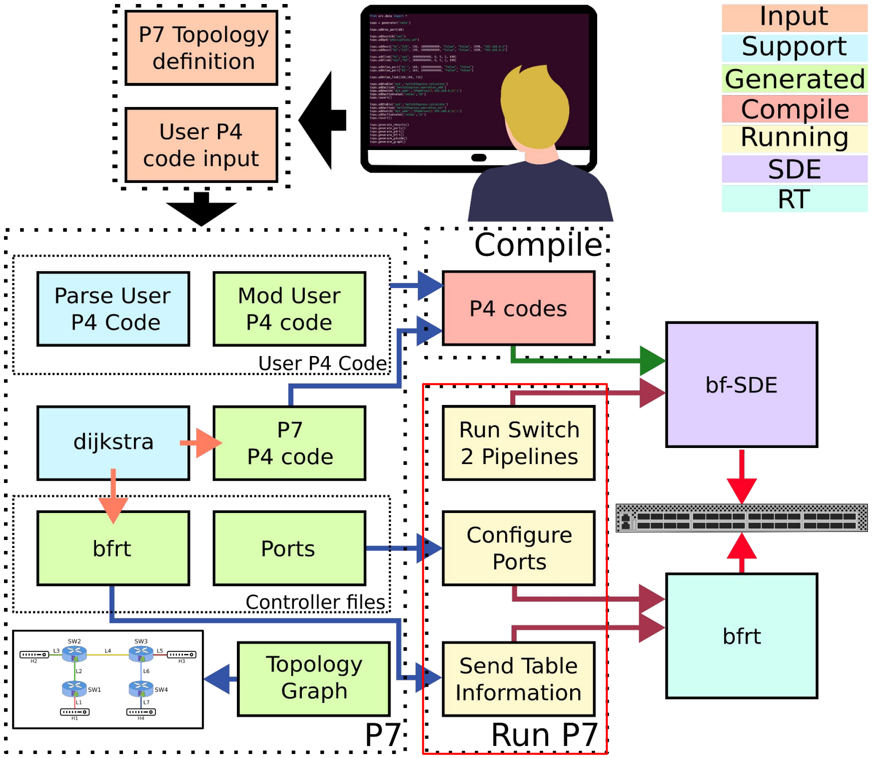
P7 Architecture & User workflow



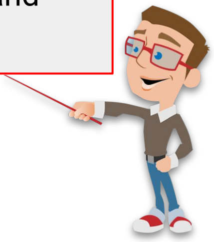
The user needs to compile both P4 codes



P7 Architecture & User workflow



Finally, the user can run the switch with both P4 codes and send the tables and ports configuration using the bfrt.



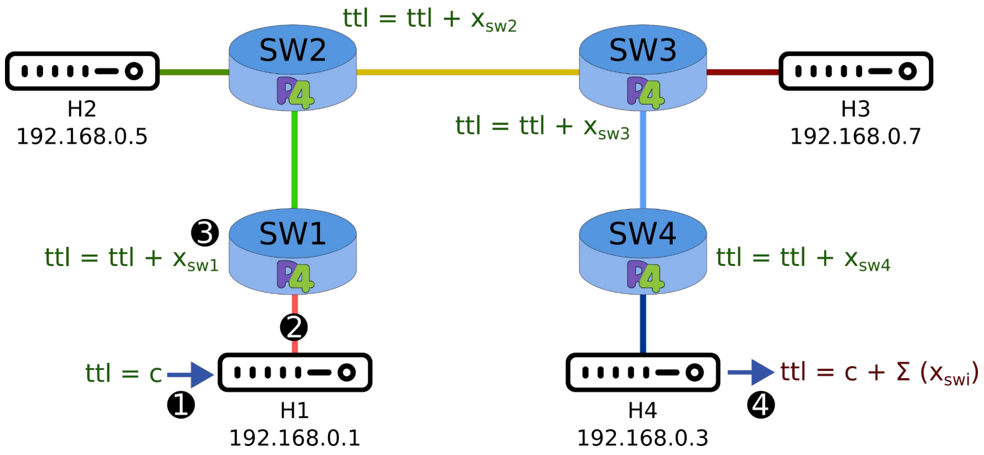
DEMO

ORGANIZAÇÃO SOCIAL DO MCTI

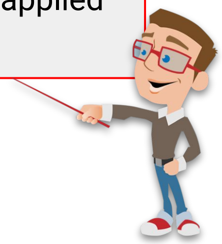
INFORMATION & NETWORKING
TECHNOLOGIES RESEARCH &
INNOVATION GROUP

UNICAMP

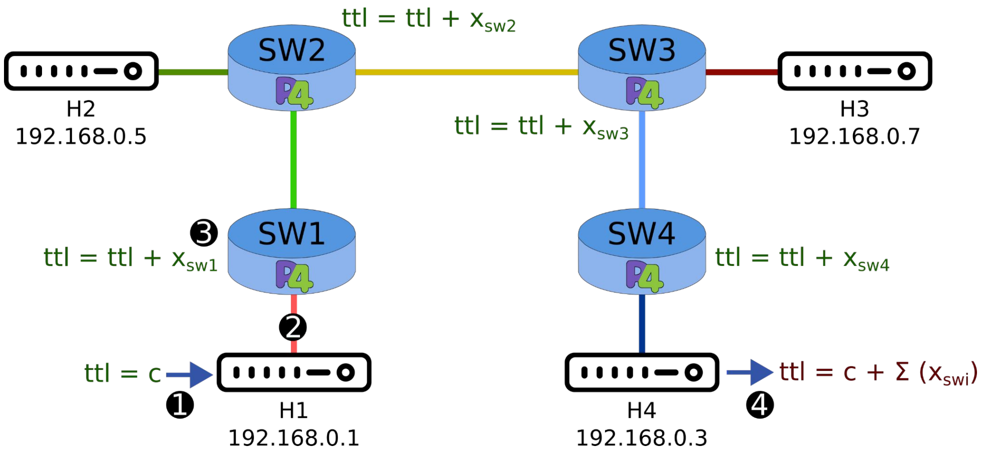
Network topology



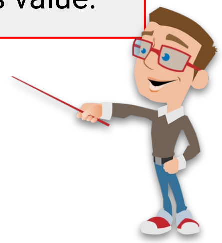
P4 code that contains different mathematical operations that are applied to the IP field ttl.



Network topology

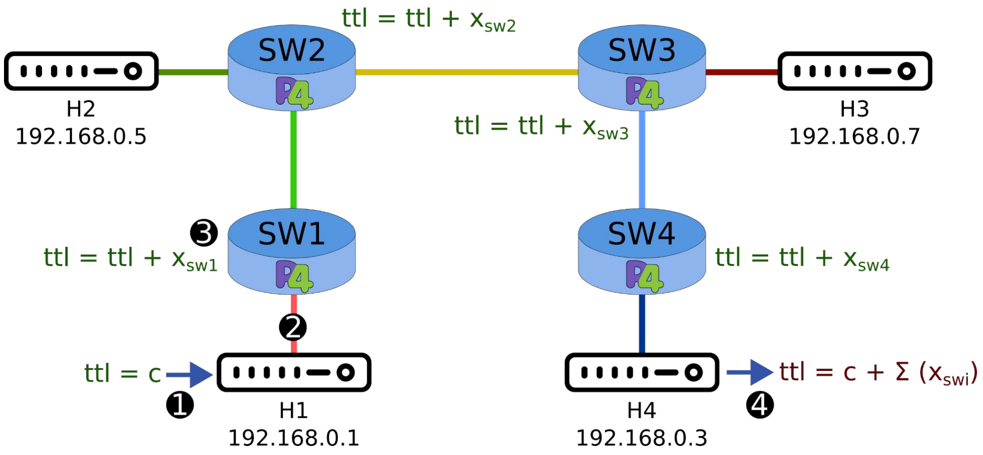


These operations are defined by a P4 table that contains the operation and its value.

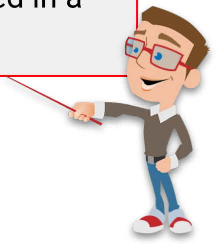


```
# add table entry sw1
topo.addtable('sw1','SwitchIngress.calculate')
topo.addaction('SwitchIngress.operation_add')
topo.addmatch('dst_addr','IPAddress(\'192.168.0.1\')')
topo.addactionvalue('value','5')
topo.insert()
```

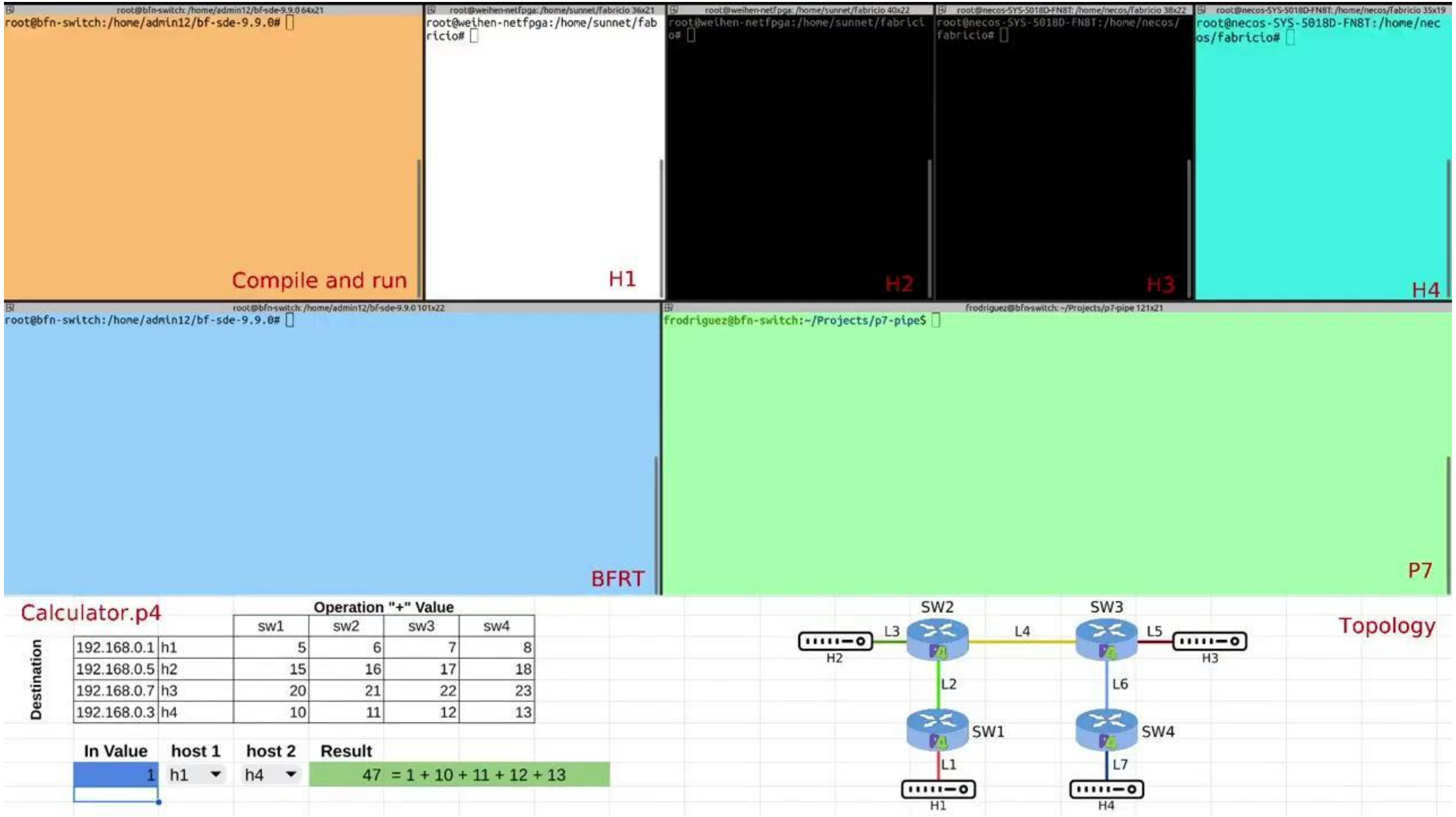
Network topology



The P4 code will perform a specific operation based on the destination IP of the packet and the information filled in a table for each switch.



```
# add table entry sw1
topo.addtable('sw1','SwitchIngress.calculate')
topo.addaction('SwitchIngress.operation_add')
topo.addmatch('dst_addr','IPAddress(\'192.168.0.1\')')
topo.addactionvalue('value','5')
topo.insert()
```



Demo running P7 with the calculator P4 code

https://drive.google.com/file/d/1EYjpcuwoSguL9yO2cnliIYH7vW-mRFMQ/view?usp=share_link

Future of P7

- Address scalability challenges
  - Topology Size
  - Buffers consumption
- New features
  - In-band Network Telemetry (INT)
  - Dynamic link behaviors
  - Trace base link characteristics
- Embed into disaggregated network testbed initiatives
  - e.g Open RAN Brasil
  - Facilitate reproducible experiments based on use case scenarios (e.g. congestion, heavy-hittters, DDoS, bufferbloat, slicing, etc.)

Further info + Q&A

P7 (P4 Programmable Patch Panel): HW Emulated Network Testbed

Open Source repository

- https://github.com/intrig-unicamp/p7

Publications

- F. Rodriguez et al. "**P4 Programmable Patch Panel (P7): An Instant 100G Emulated Network on Your Tofino-based Pizza Box**". In ACM SIGCOMM'22 Poster/Demo Session, Aug. 2022. (SIGCOMM SRC'22 Award Winner)

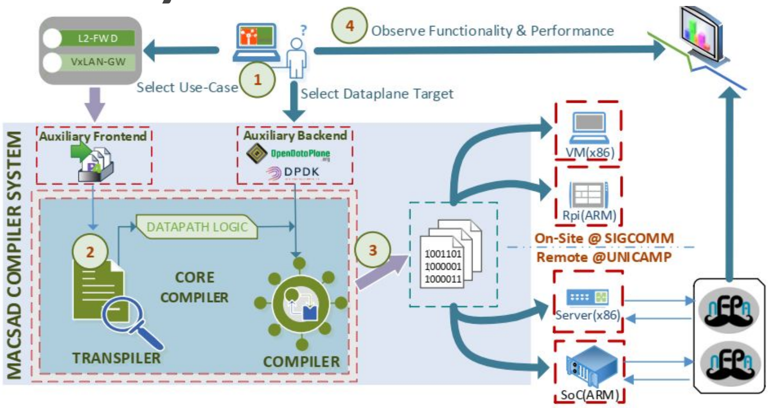
Demos

- F. Rodriguez et al. "**P7 (P4 Programmable Patch Panel): An instant 100G emulated network testbed in a pizza box**". In P4 Workshop 2022. (Award Winner of most novel networking use of P4)
  - https://www.youtube.com/watch?v=aRYxPvlvo_Q
- F. Rodriguez et al. "**Towards Multiple Pipelines Network Emulation with P7**". To appear in IEEE NetSoft, Jun. 2023.
  - https://drive.google.com/file/d/1EYjpcuwoSguL9yO2cnliIYH7vW-mRFMQ/view
- F. Rodriguez et al. "**Network Emulation with P7: A P4 Programmable Patch Panel on Tofino-based Hardware.**" To appear in SBRC Salão de Ferramentas, May 2023.
  - https://www.youtube.com/watch?v=dAhy8R34vHU

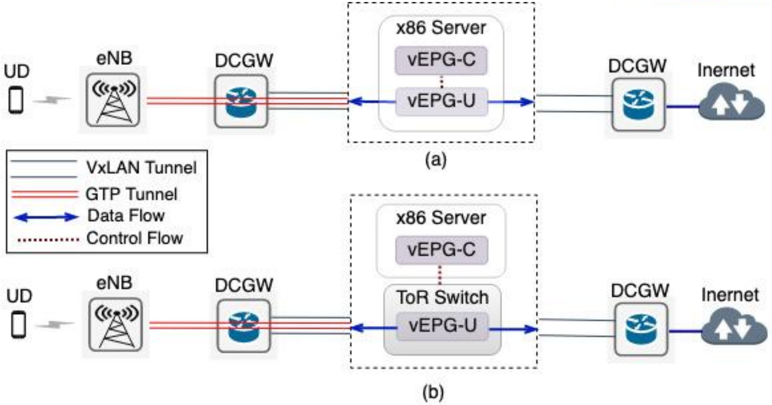
Hybrid-P4-5G: User Plane Functions in Programmable SW/HW Stacks

Recent research

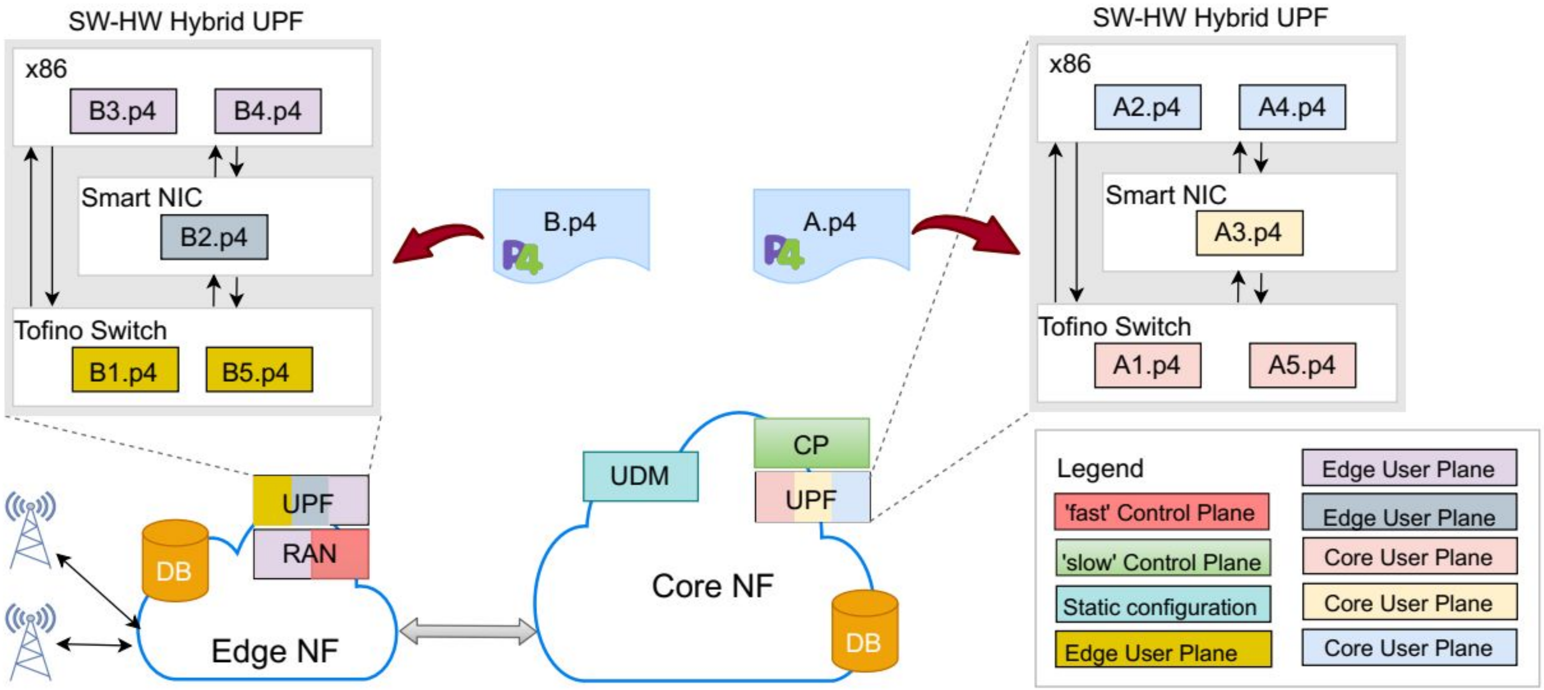
- **MACSAD: Multi-Architecture Compiler System for Abstract Dataplanes**
  - Partnering P4 with ODP



- **vEPG: 4G/5G user plane functions**
  - x86 / ODP / DPDK
  - Tofino HW



Hybrid P4 Programmable Pipelines for 5G gNodeB and User Plane Functions



Hybrid P4 Programmable Pipelines for 5G gNodeB and User Plane Functions

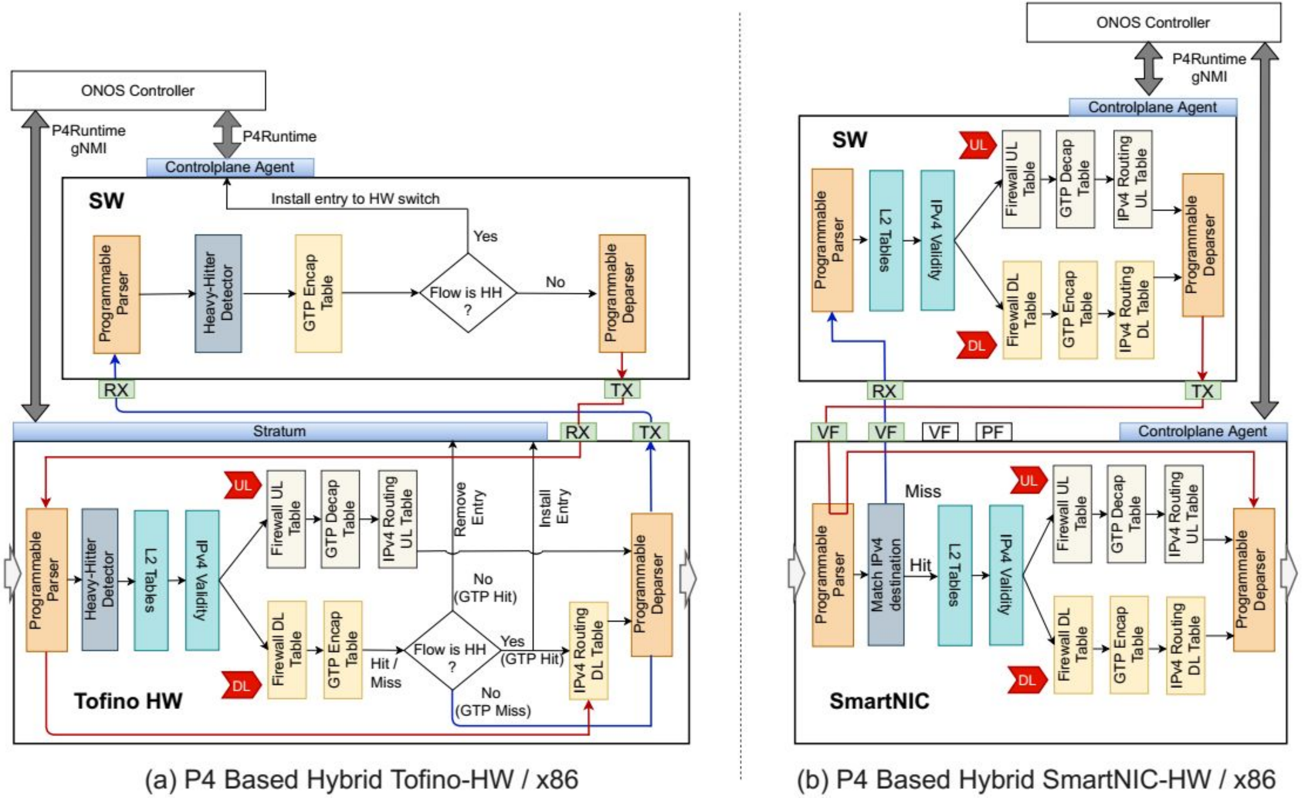
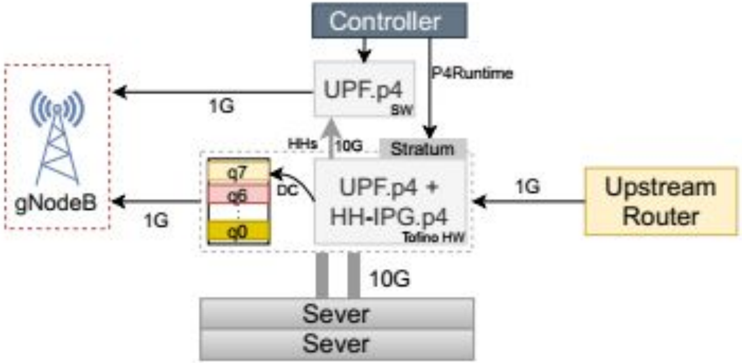
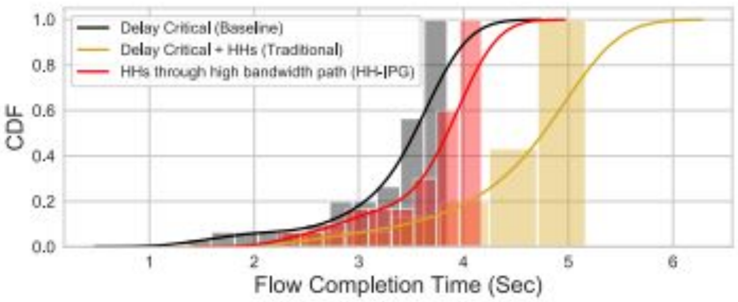


Figure 5: P4-based hybrid Tofino-HW/x86 and SmartNIC/x86 architectures for UPF.

Hybrid P4 Programmable Pipelines for 5G gNodeB and User Plane Functions



(a) QoS-HH Use-case Scenario



(b) Flow completion time with and without HHs offloading

Further info + Q&A

Hybrid-P4-5G: User Plane Functions in Programmable SW/HW Stacks

Open Source repository

- https://github.com/intrig-unicamp/macsad

Publications

- S. Singh et al. "**Hybrid P4 Programmable Pipelines for 5G gNodeB and User Plane Functions**". In IEEE TMC. 2022
- S. Singh et al. "**HH-IPG: Leveraging Inter-Packet Gap Metrics in P4 Hardware for Heavy Hitter Detection**". In IEEE TNSM, 2022.

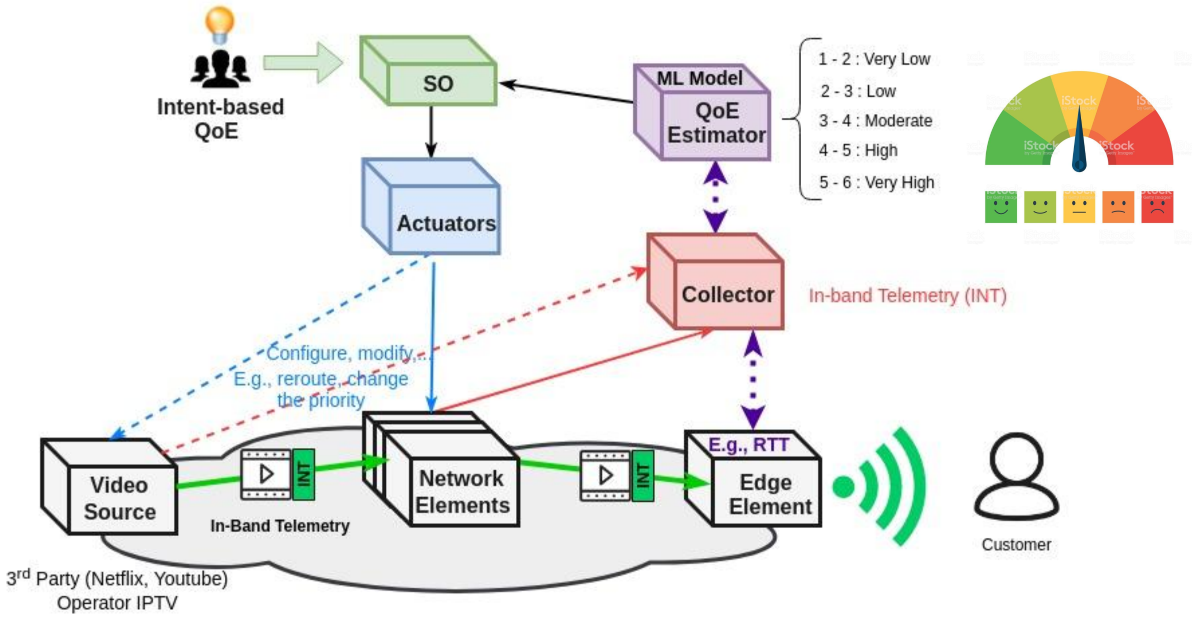
Demos

- https://www.youtube.com/watch?v=pskxeMETYGY

EFFECTOR: DASH QoE Evaluation Framework with 5G Datasets

Recent research

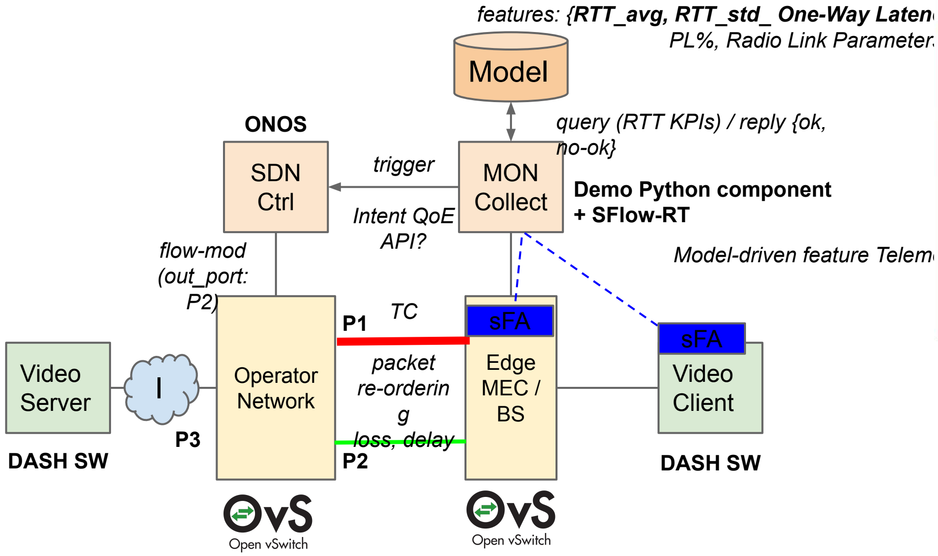
Intent-based CCL for DASH Video SLA using ML-based Edge QoE Estimation

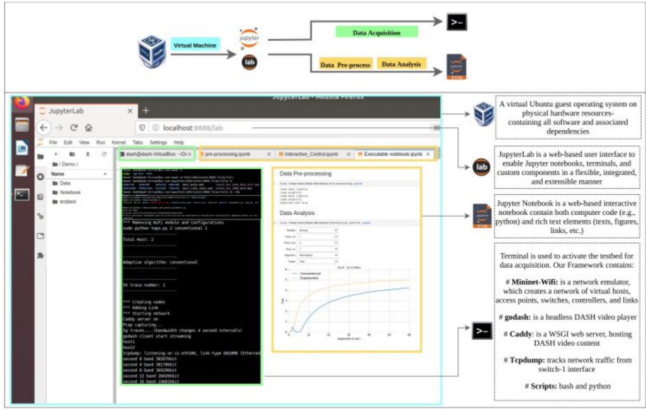


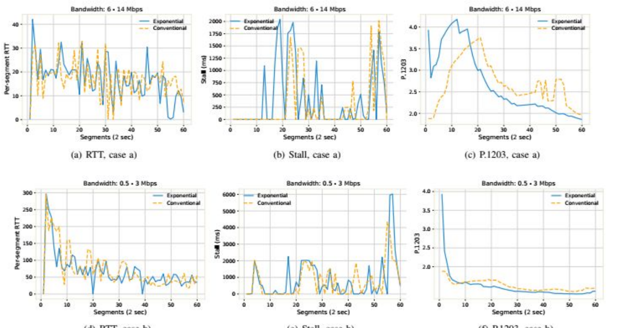
Christian Esteve Rothenberg, Danny Lachos Perez, Nathan Saraiva, Raphael Rosa, Raza Ul Mustafa, Md Tariqul Islam, Pedro Henrique Gomes. "**Intent-based Control Loop for DASH Video Service Assurance using ML-based Edge QoE Estimation**". In IEEE NetSoft 2020 Demo Session, Ghent, Belgium, June 2020.

Recent research

Intent-based CCL for DASH Video SLA using ML-based Edge QoE Estimation



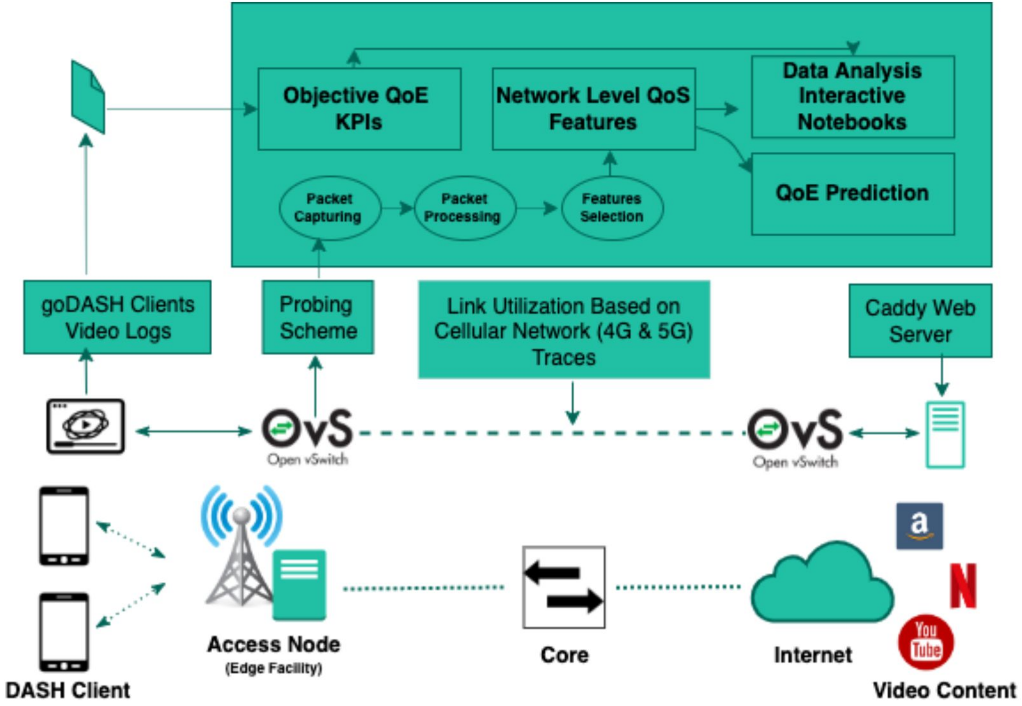


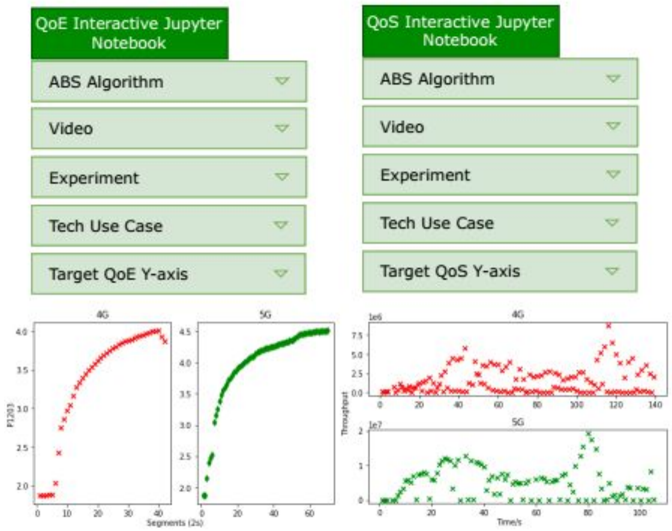


Raza Ul Mustafa, Md Tariqul Islam, Christian Esteve Rothenberg, Simone Ferlin, Darijo Raca, and Jason J. Quinlan. "**DASH QoE performance evaluation framework with 5G datasets**". In IEEE CNSM AnServApp. Nov. 2020.

Recent and ongoing research

DASH QoE estimation from QoS-level metrics in 4G/5G scenarios

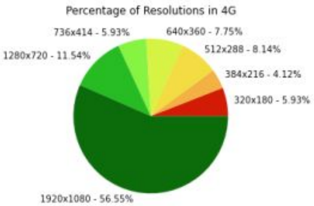


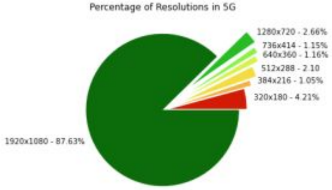


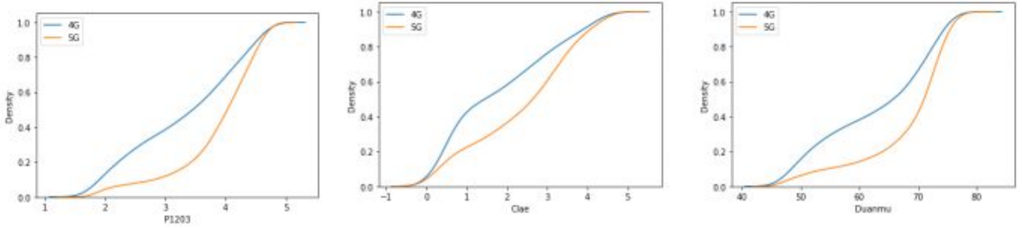
https://github.com/razaulmustafa852/EFFECTOR

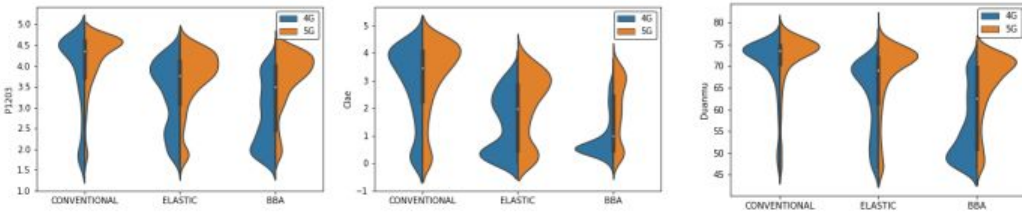
Recent and ongoing research

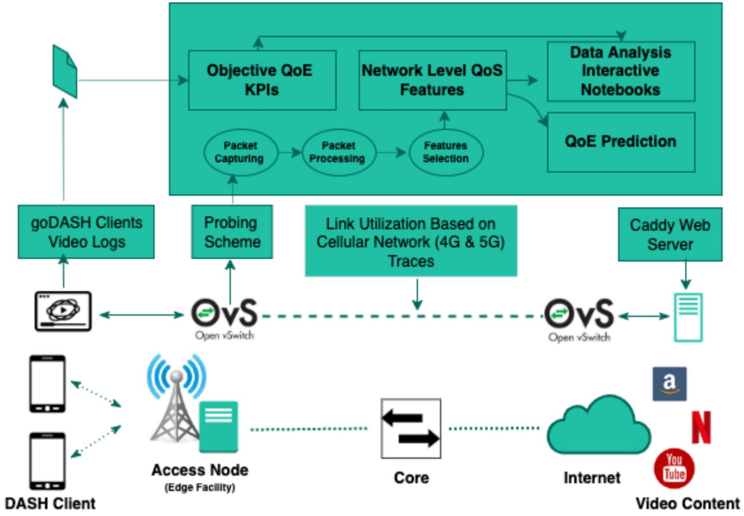
DASH QoE estimation from QoS-level metrics in 4G/5G scenarios

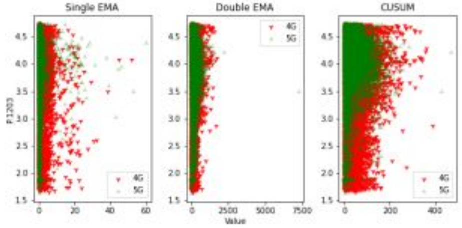












Throughput Sintel - TCP

Technology	Mean	STD	Min	25 %	50 %	75 %	Max
4G	3.26	3.90	0.000416	0.17	1.41	5.60	19.39
5G	5.10	4.40	0.000416	0.625736	4.90	7.88	19.76

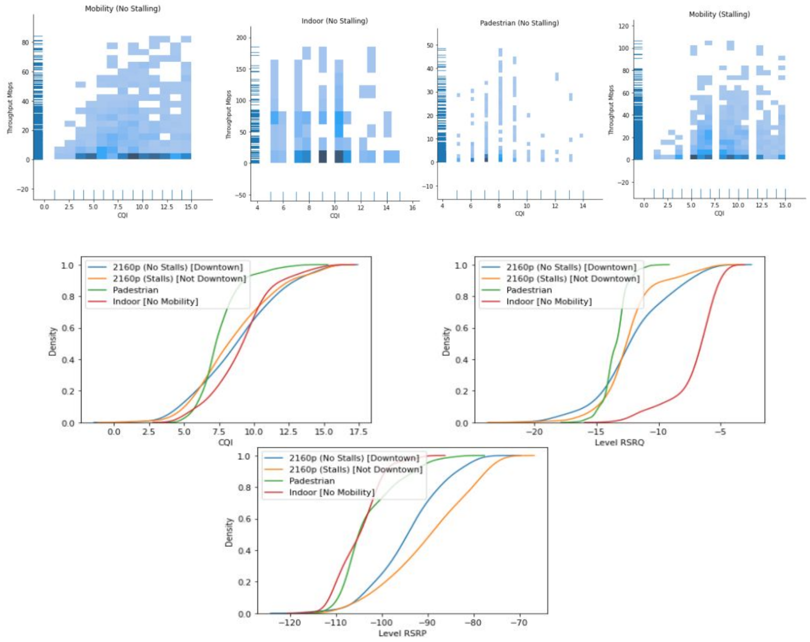
Throughput Tears - QUIC

Technology	Mean	STD	Min	25 %	50 %	75 %	Max
4G	1.75	2.17	0.000416	0.39	1.05	2.20	15.08
5G	6.32	6.03	0.000416	0.08	5.02	11.99	25.40

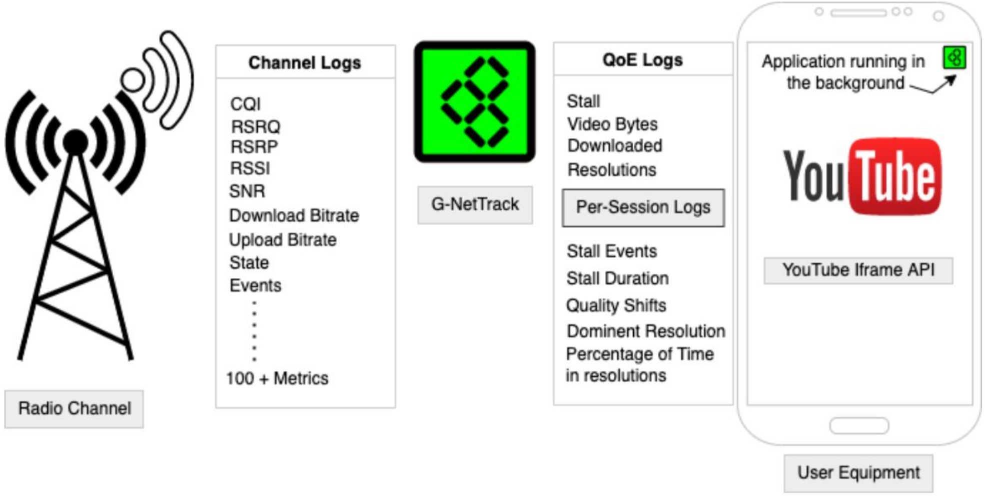
Recent and ongoing research

Real 4G/5G trace collection in Youtube streaming

- Nice France, Campinas (5G DSS) and Sao Paulo (5GSA)

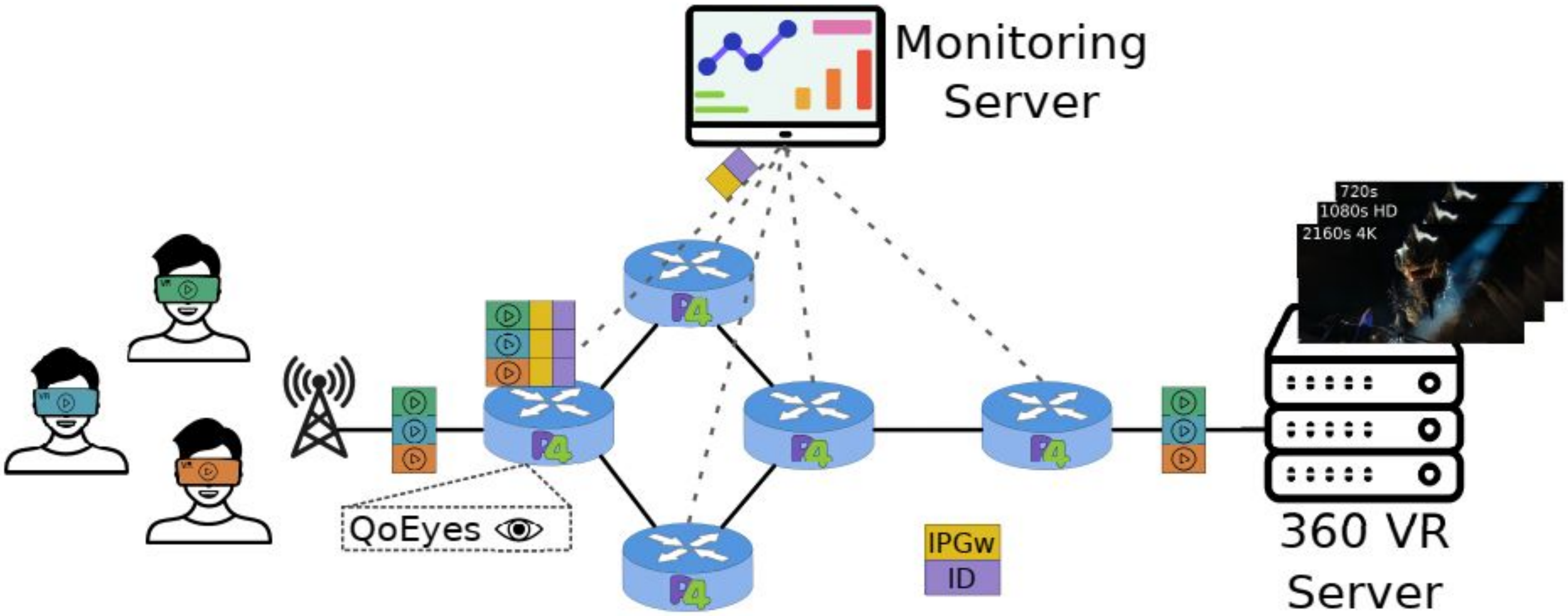


- YouTube IFRAME API for YouTube QoE Logs
- G-NetTrack Pro - Wireless network monitor and drive test tool

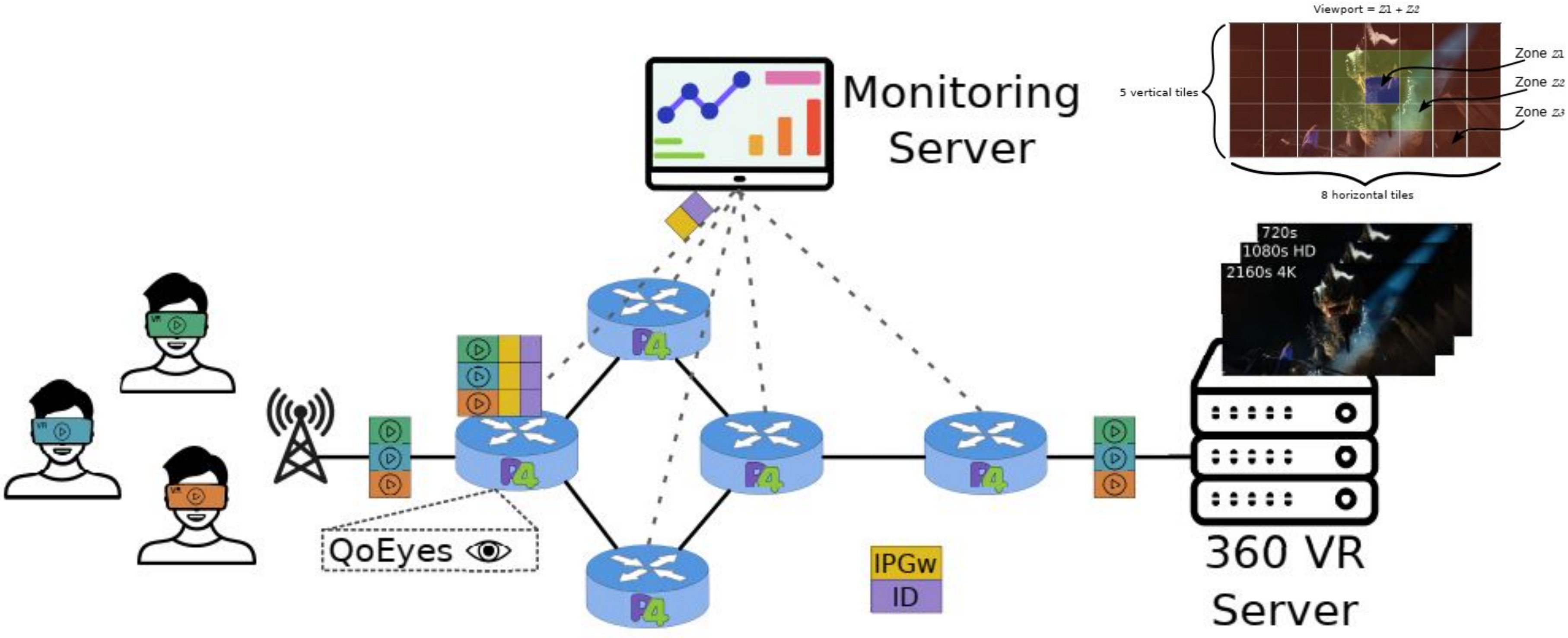


QoEyes: Virtual Reality Streaming QoE Estimation through ML and INT

QoEyes


Monitoring Server
720s
1080s HD
2160s 4K
QoEyes
IPGw
ID
360 VR Server

QoEyes


Monitoring
Server
Viewport = z1 + z2
5 vertical tiles
8 horizontal tiles
Zone z1
Zone z2
Zone z3
720s
1080s HD
2160s 4K
QoEyes
IPGw
ID
360 VR
Server

QoEyes

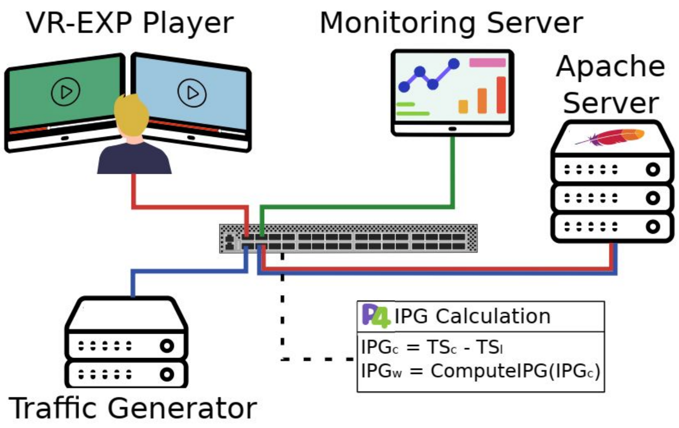


Fig. 3. Experimental environment.

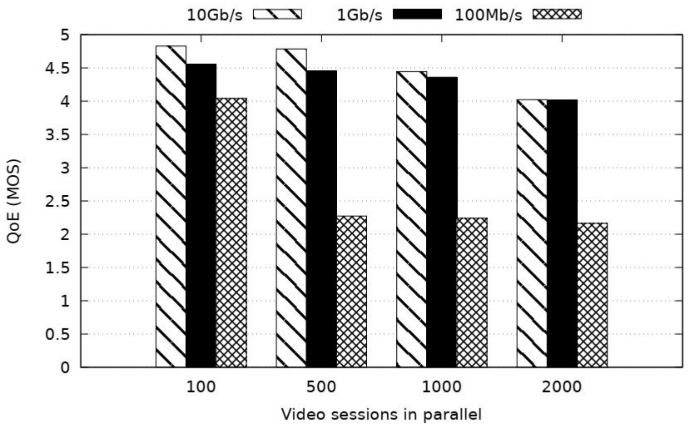
TABLE I
HARDWARE RESOURCE UTILIZATION

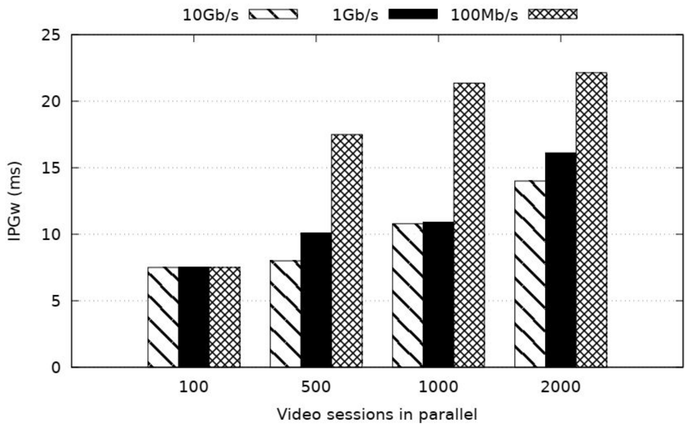
Resource	**Switch.p4**	**QoEyes**
Hash Bits	32.3%	34.2%
SRAM	29.8%	30.6%
TCAM	28.4%	28.4%
VLIW Actions	34.6%	38.8%
Stateful ALUs	15.6%	15.6%

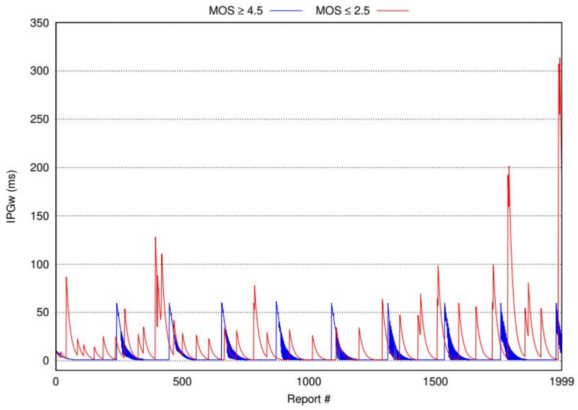
TABLE II
NUMBER OF TILES RECEIVED WITH 100 SESSIONS IN PARALLEL

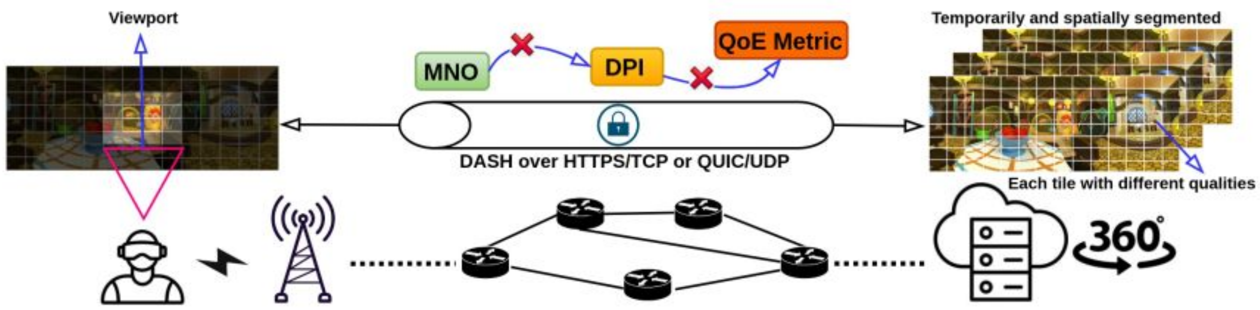
	10Gb			1Gb			100Mb		
	z1	z2	z3	z1	z2	z3	z1	z2	z3
720p	1	12	1375	1	12	1375	1	12	1375
4K	59	468	5	59	468	5	59	468	5

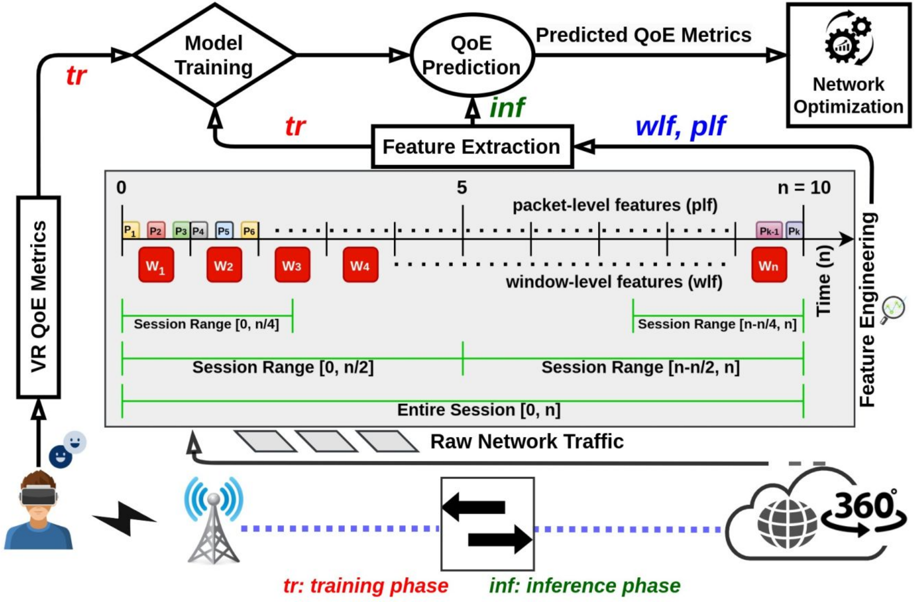
QoEyes

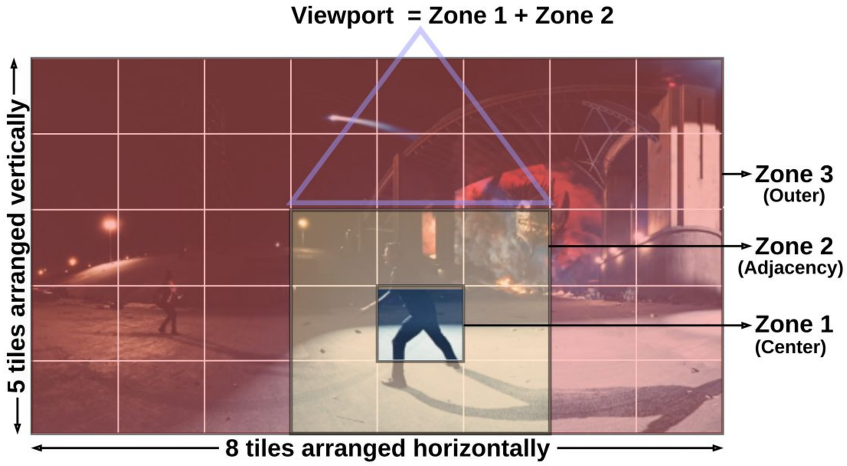

10Gb/s
1Gb/s
100Mb/s
QoE (MOS)
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
100
500
1000
2000
Video sessions in parallel


10Gb/s
1Gb/s
100Mb/s
IPGw (ms)
25
20
15
10
5
0
100
500
1000
2000
Video sessions in parallel


MOS ≥ 4.5
MOS ≤ 2.5
IPGw (ms)
350
300
250
200
150
100
50
0
0
500
1000
1500
1999
Report #


Viewport
MNO
DPI
QoE Metric
DASH over HTTPS/TCP or QUIC/UDP
Temporarily and spatially segmented
Each tile with different qualities
360°


Model Training
QoE Prediction
Predicted QoE Metrics
Network Optimization
tr
tr
inf
wlf, plf
Feature Extraction
VR QoE Metrics
0
5
n = 10
packet-level features (plf)
P1 P2 P3 P4 P5 P6
Pk-1 Pk
W1 W2 W3 W4 Wn
window-level features (wlf)
Time (n)
Feature Engineering
Session Range [0, n/4]
Session Range [n-n/4, n]
Session Range [0, n/2]
Session Range [n-n/2, n]
Entire Session [0, n]
Raw Network Traffic
360°
tr: training phase
inf: inference phase


Viewport = Zone 1 + Zone 2
5 tiles arranged vertically
8 tiles arranged horizontally
Zone 3
(Outer)
Zone 2
(Adjacency)
Zone 1
(Center)

QoS KPI	Applied Statistics: Entire Session	Applied Statistics: F25, L25, F50, L50
Throughput (TP)	avg, max, min, medn, std, 10-90p	avg
Packet Count (PC)	total, avg, max, min, medn, std, 10-90p	total
Interarrival Time (IAT)	avg, max, min, medn, std, 10-90p	avg
Packet Size (PS)	avg, max, min, medn, std, 10-90p	avg

Abbreviation	Details
avg, max, min, medn, std	average, maximum, minimum, median, standard deviation
10-90p	the distribution of the 10th to 90th percentile (in steps of 10)
F25, L25, F50, L50	first and last 25% and 50% of session

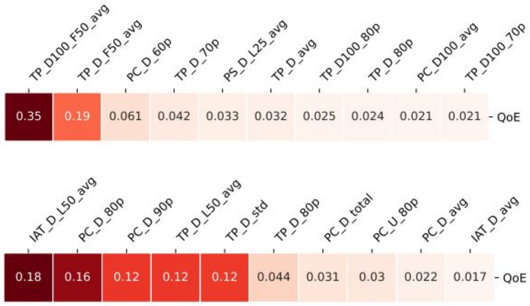
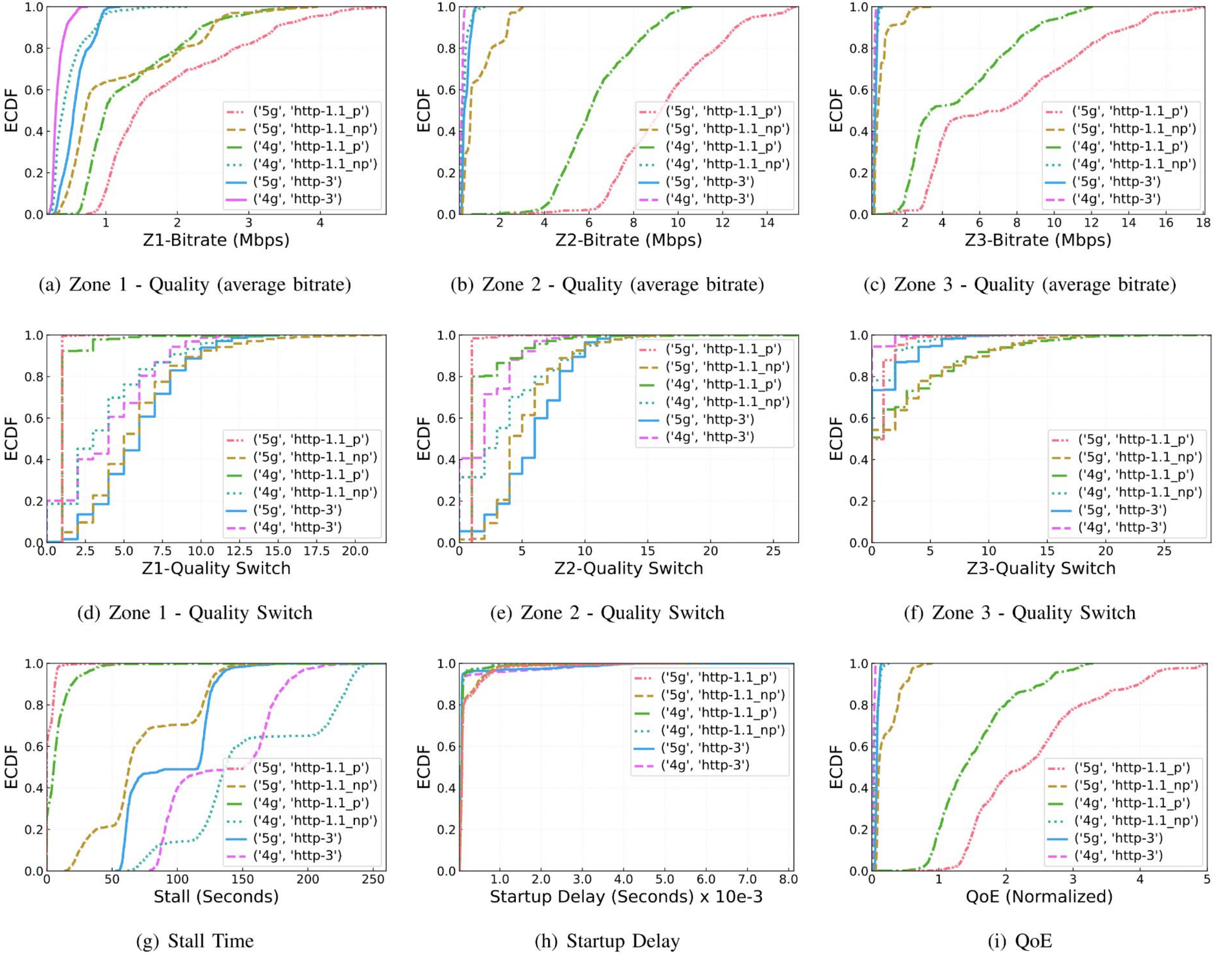


Fig. 6: The Ranking of the Top 10 Most Significant Features for QoE Value Prediction Across HTTPS (Top) and QUIC (Bottom) Datasets

TABLE II: RMSE and r^2 (highlighted in blue) Across Different Objective Metrics and QoE Value Prediction Models

	Random Forest		Extra Trees		K Nearest Neighbors		Light GBM		Cat Boost		XG Boost		Neural Networks		Weighted Ensemble	
	H	Q	H	Q	H	Q	H	Q	H	Q	H	Q	H	Q	H	Q
Quality-Z1	-0.25	-0.05	-0.26	-0.05	-0.34	-0.06	-0.22	-0.05	-0.22	-0.05	-0.22	-0.05	-0.28	-0.05	0.93 -0.22	0.93 -0.05
Quality-Z2	-0.34	-0.02	-0.34	-0.02	-0.72	-0.03	-0.32	-0.02	-0.32	-0.02	-0.32	-0.02	-0.36	-0.02	0.99 -0.32	0.99 -0.02
Quality-Z3	-0.24	-0.01	-0.20	-0.01	-0.48	-0.01	-0.18	-0.01	-0.18	-0.01	-0.22	-0.01	-0.19	-0.01	0.99 -0.17	0.99 -0.01
Quality Shifts-Z1	-1.39	-2.02	-1.38	-2.00	-1.57	-2.16	-1.36	-2.00	-1.38	-2.01	-1.42	-2.02	-1.43	-2.02	0.81 -1.37	0.55 -2.00
Quality Shifts-Z2	-1.26	-1.57	-1.23	-1.55	-1.52	-1.85	-1.26	-1.55	-1.24	-1.56	-1.25	-1.57	-1.35	-1.58	0.81 -1.27	0.77 -1.53
Quality Shifts-Z3	-1.16	-0.86	-1.13	-0.84	-1.66	-0.96	-1.14	-0.84	-1.12	-0.84	-1.14	-0.86	-1.27	-0.87	0.87 -1.16	0.69 -0.84
Stall Time	-2.49	-2.85	-1.33	-2.14	-6.13	-8.15	-1.40	-1.84	-1.44	-2.68	-2.11	-1.76	-1.71	-1.77	0.99 -1.18	0.99 -1.51
Startup Delay	-0.31	-0.60	-0.29	-0.56	-0.31	-0.56	-0.27	-0.50	-0.27	-0.50	-0.29	-0.55	-0.26	-0.50	0.05 -0.26	-0.02 -0.50
QoE	-0.08	-0.11	-0.07	-0.10	-0.18	-0.14	-0.06	-0.09	-0.05	-0.09	-0.07	-0.10	-0.06	-0.09	0.99 -0.06	0.99 -0.09



(a) Zone 1 - Quality (average bitrate)

(b) Zone 2 - Quality (average bitrate)

(c) Zone 3 - Quality (average bitrate)

(d) Zone 1 - Quality Switch

(e) Zone 2 - Quality Switch

(f) Zone 3 - Quality Switch

(g) Stall Time

(h) Startup Delay

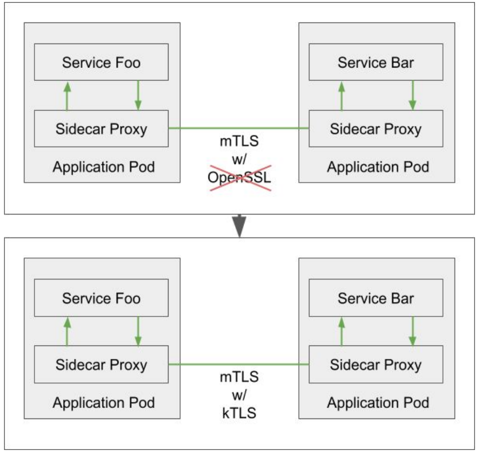
(i) QoE

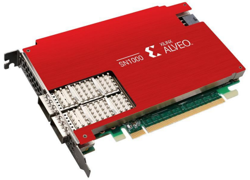
Fig. 4: The Distribution for Quality, Quality Shifts, Stall Time, Startup Delay Metrics and Calculated QoE Value

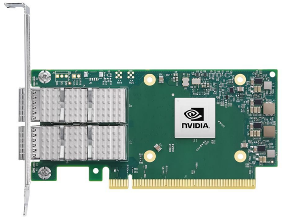
PoD acceleration: Offloading container networking & security functions

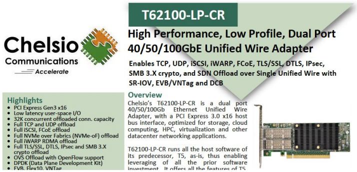
PoD acceleration:
Joint work with Prof. Verdi (UFSCAR) & ERICSSON

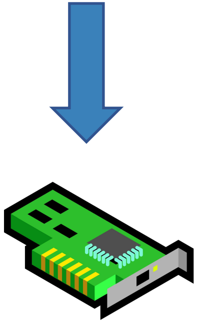
- Offloading of Kubernetes networking functions
- Offloading mesh crypto functions (eg. kTLS)
- Targets: smartnics and/or IPU (eg. Intel Mount Evans, Xilinx SN1000, Nvidia ConnectX-6 DX, Chelsio T62100 LP-CR)





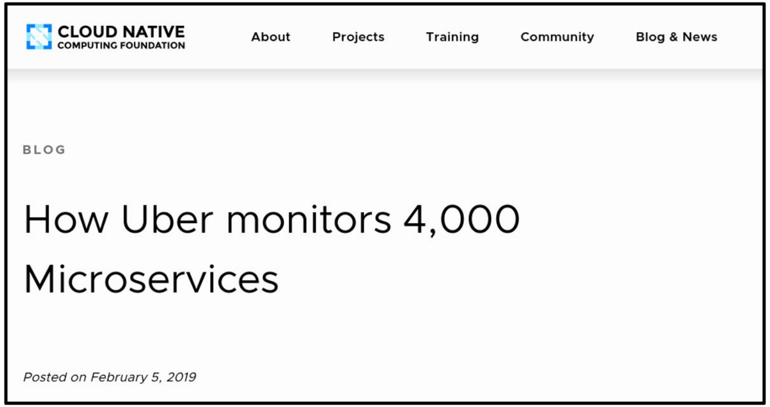






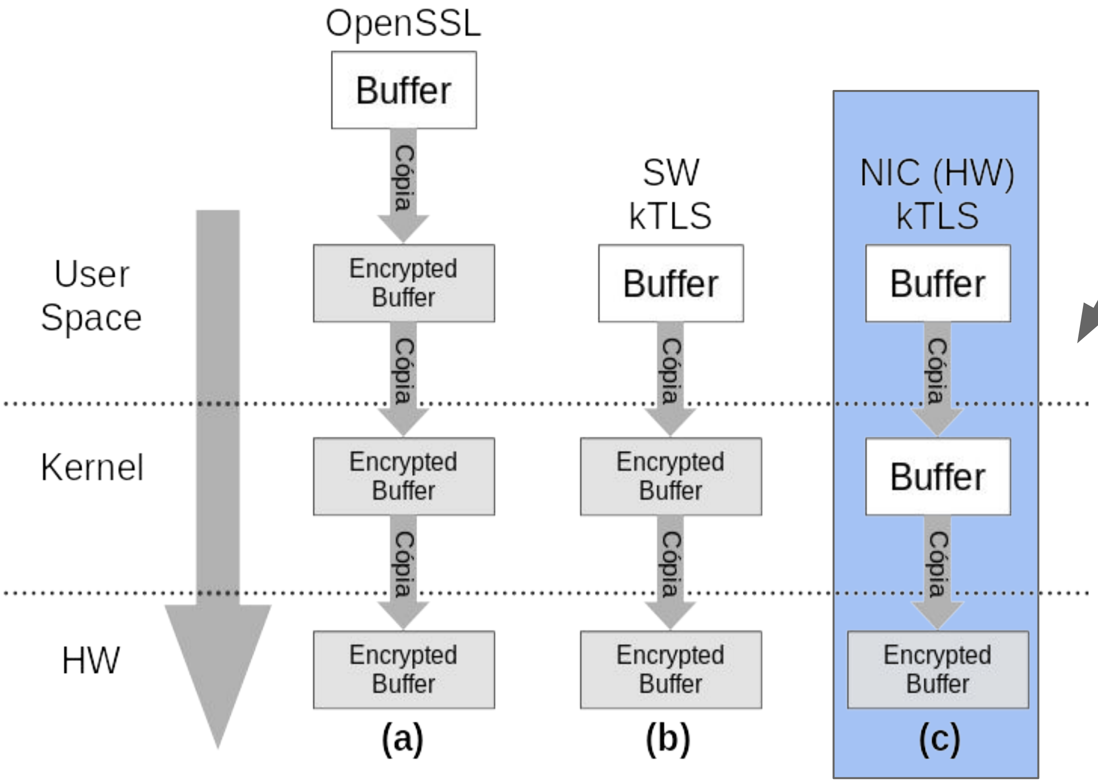
Motivation

Applications are mostly dependent on crypto functions

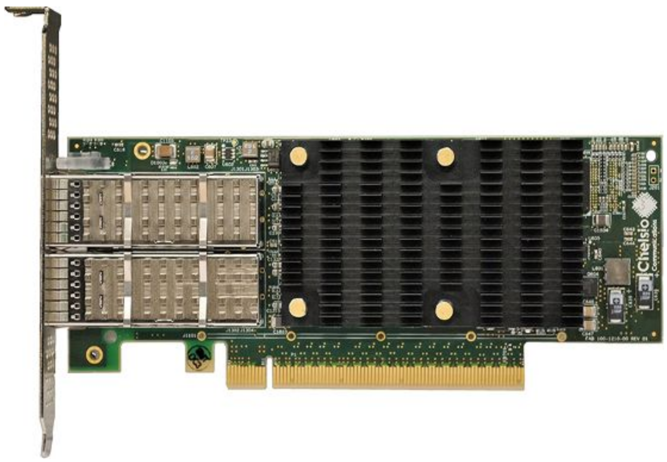


HTTP/2

kTLS and its modes



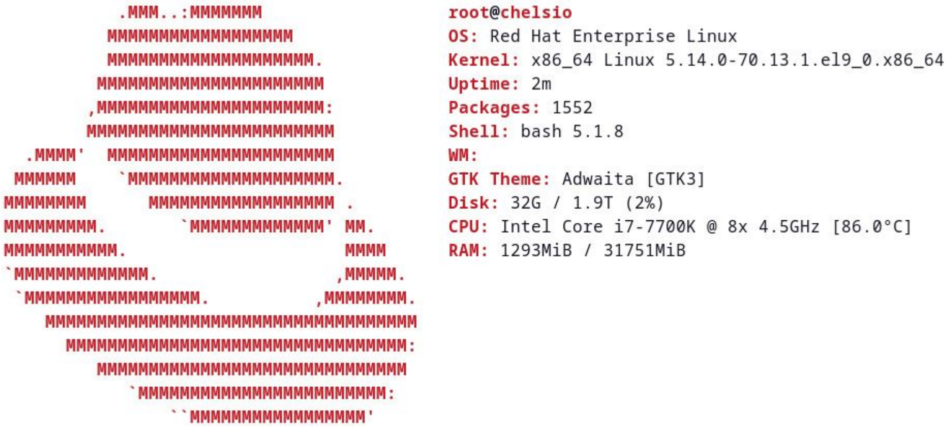
This is what we want



Chelsio's full offload TLS/SSL is uniquely capable of 100Gb line rate performance

Ongoing: Setting up the experiments

- Current setup: Red Hat Enterprise Linux release 9.1 (Plow)
- Preliminary, we found the following results for average CPU usage per each approach:



No kTLS	SW kTLS	NIC kTLS (Inline)	NIC kTLS (Co-processor)
87.7 %	70.6 %	26.7 %	52.8 %

Next: Virtualization

- We began our tests with containers because kTLS relies too much on Linux Kernel, and containerization would likely be more plausible with NIC kTLS.

podman

SMARTNESS 2030

Conversation with RedHat

SMARTNESS Engineering Research Center

Focus: how to engineer (i.e. plan, design, build and operate) cloud computing and network infrastructures with the adequate capabilities to empower next generation internet services and applications.

Founders:

- Universities: UNICAMP, USP and UFSCar
- Company: Ericsson

Duration: Up to ten years (5+5)

Funding model

- Fapesp:Industry:University – 1:1:2

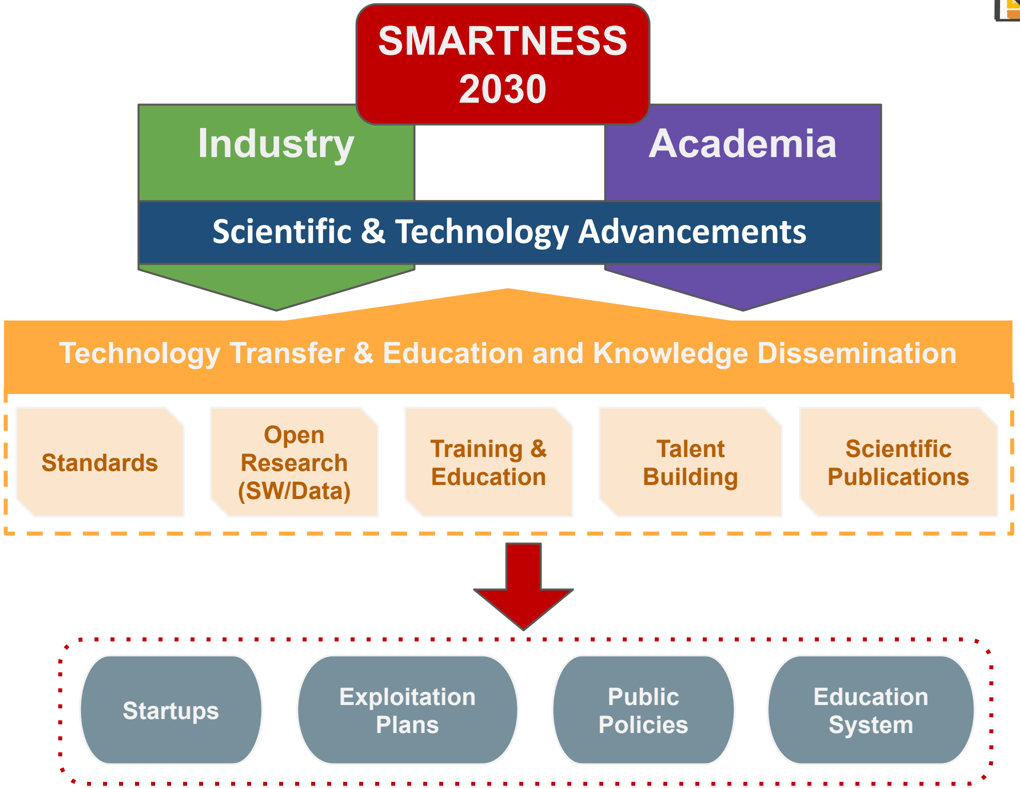
Location: hub center at UNICAMP with Satellite locations at USP, UFSCAR and Ericsson

Size: Approximately 25 FTE

https://smartness2030.tech/

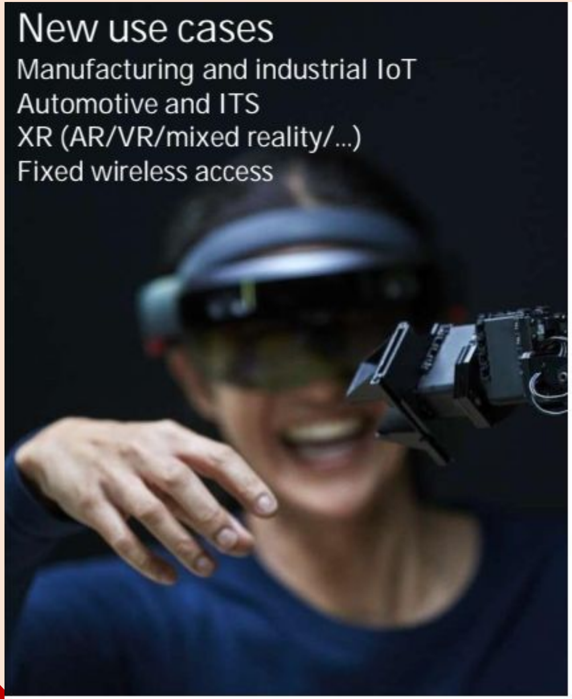
UNICAMP ERICSSON FAPESP ufscar USP

SMARTNESS 2030* Impact Setting



*The networking-centric FAPESP ERC

Towards 6G
(Ericsson Vision vs. Scope of SMARTNESS)



Integrated connectivity and edge compute

Zero-touch

Trusted networking

SMARTNESS 2030 Scope

XR (VR/AR/MR)
Virtual, augmented, and mixed reality
Multi-modal **Man-Machine** interaction
Cloud **Gaming**
Tactile Internet
Immersive Media & 3D **Holograms**
Holographic Telepresence
...

Augmented Society

Industrial Internet

Robotics & Intelligent **Industry 4.0** process automation
Industrial **IoT** & Smart manufacturing
V2X & Internet of Vehicles (**IoV**)
Intelligent Transport Systems (**ITS**)
5G/6G-based **high-precision positioning**
Domain-specific **APIs** for 3rd party apps
...

Future Applications

SMARTNESS 2030

Scientific & Technology Advancements

- **Telco Cloud & Edge Native** OS, SW/HW Architectures
- Edge compute systems across **devices & distributed cloud**
- **Decentralized data** mgm: ingestion, processing, routing, storage, streaming
- **Lightweight** Virtualization
- **Network Service Mesh**

Customized Edge Computing

Cognitive Architectures

- **ML/AI** 4 Networking
- Distributed **Machine Intelligence**
- Network + Service **BigData Repository**

Fluid Control & Data Planes

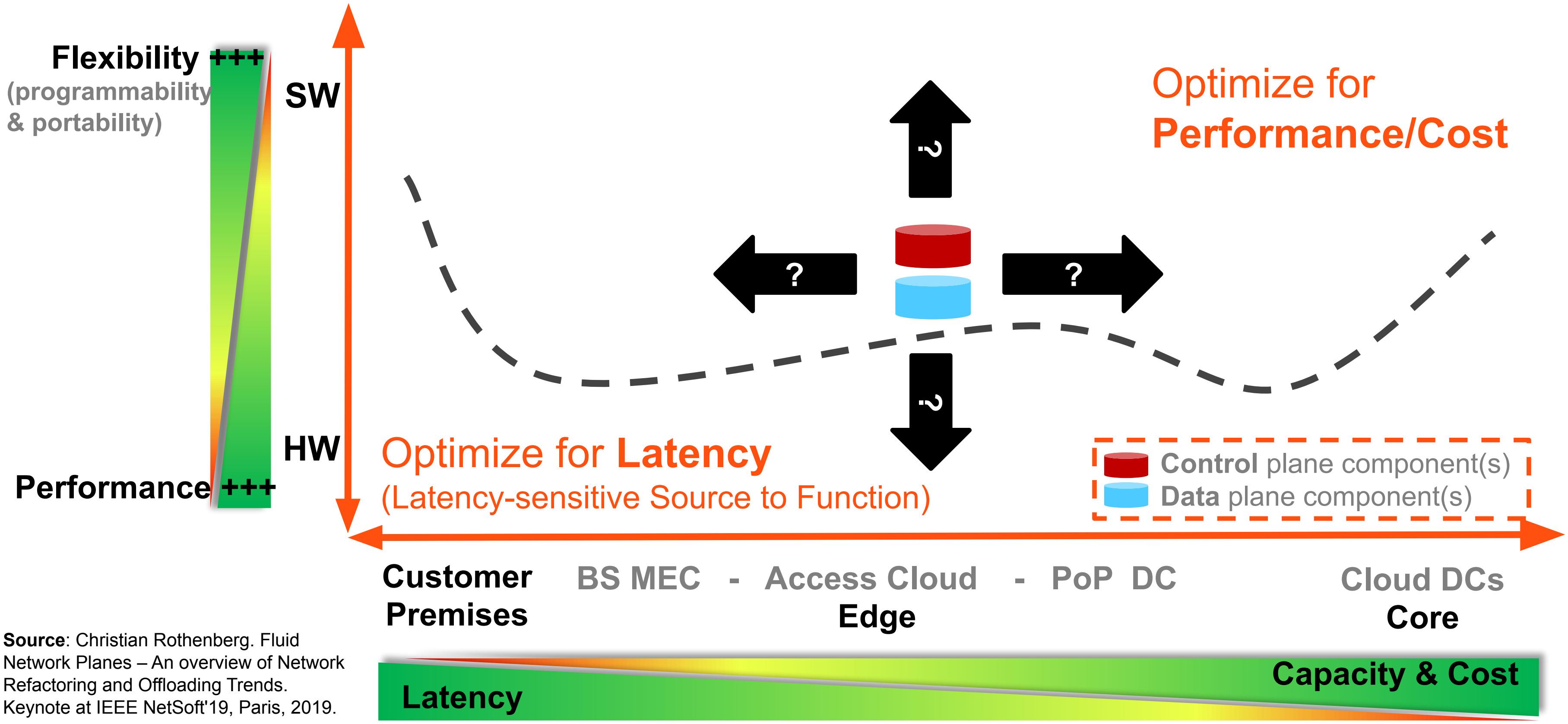
- Network **Architecture** matches what **Applications** need
- **Intent**-based Networking
- **Slicing** / Network Slicing 2.0
- **Programmability** & **Automation**
- **Flexible**, Adaptable CP/DP
- **In-Network Processing** (e.g., IETF Computing in the Network)
- **Time-Sensitive** (TSN) & **Deterministic** Net. (detnet)

Trustworthiness: Security, Privacy, Safety, Ethics

Sustainability

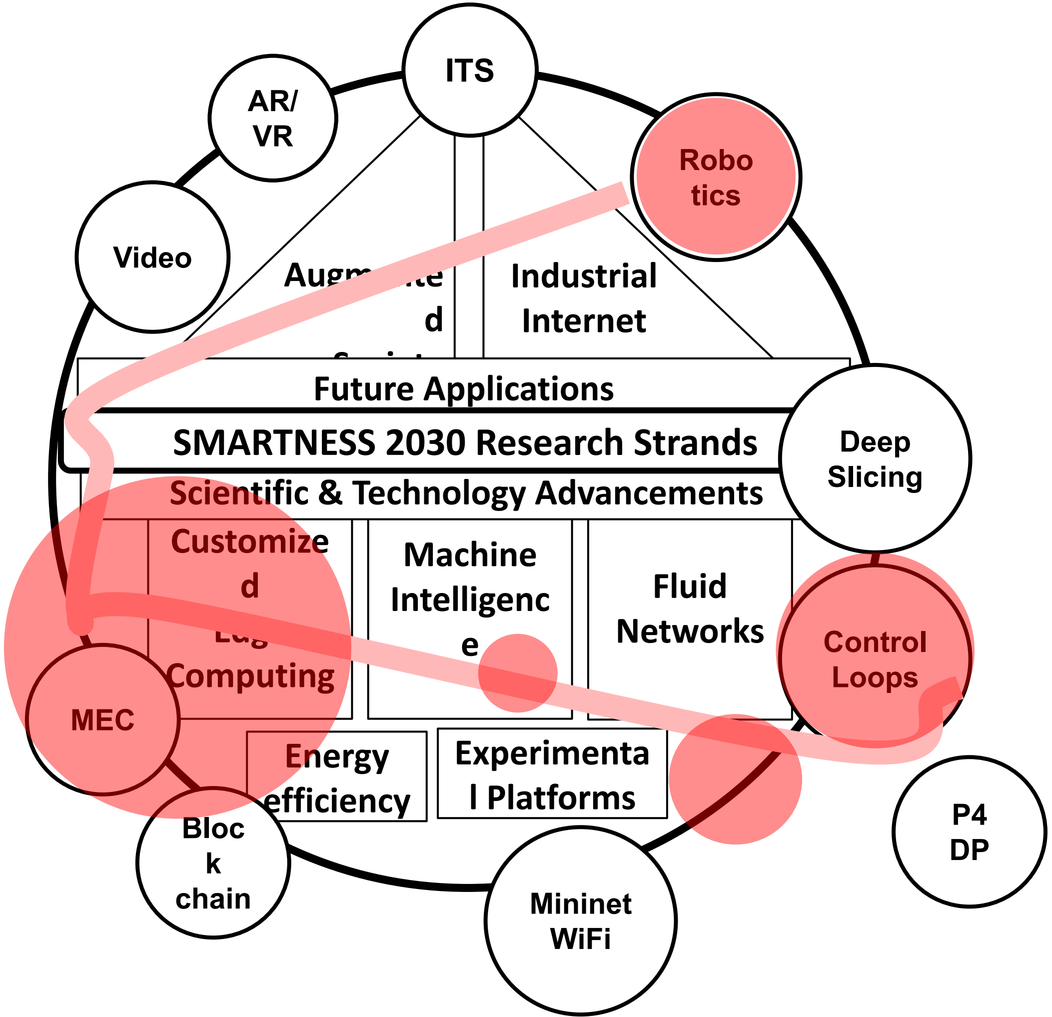
SMARTNESS 2030

Fluid network architectures tailored to Sliced Services

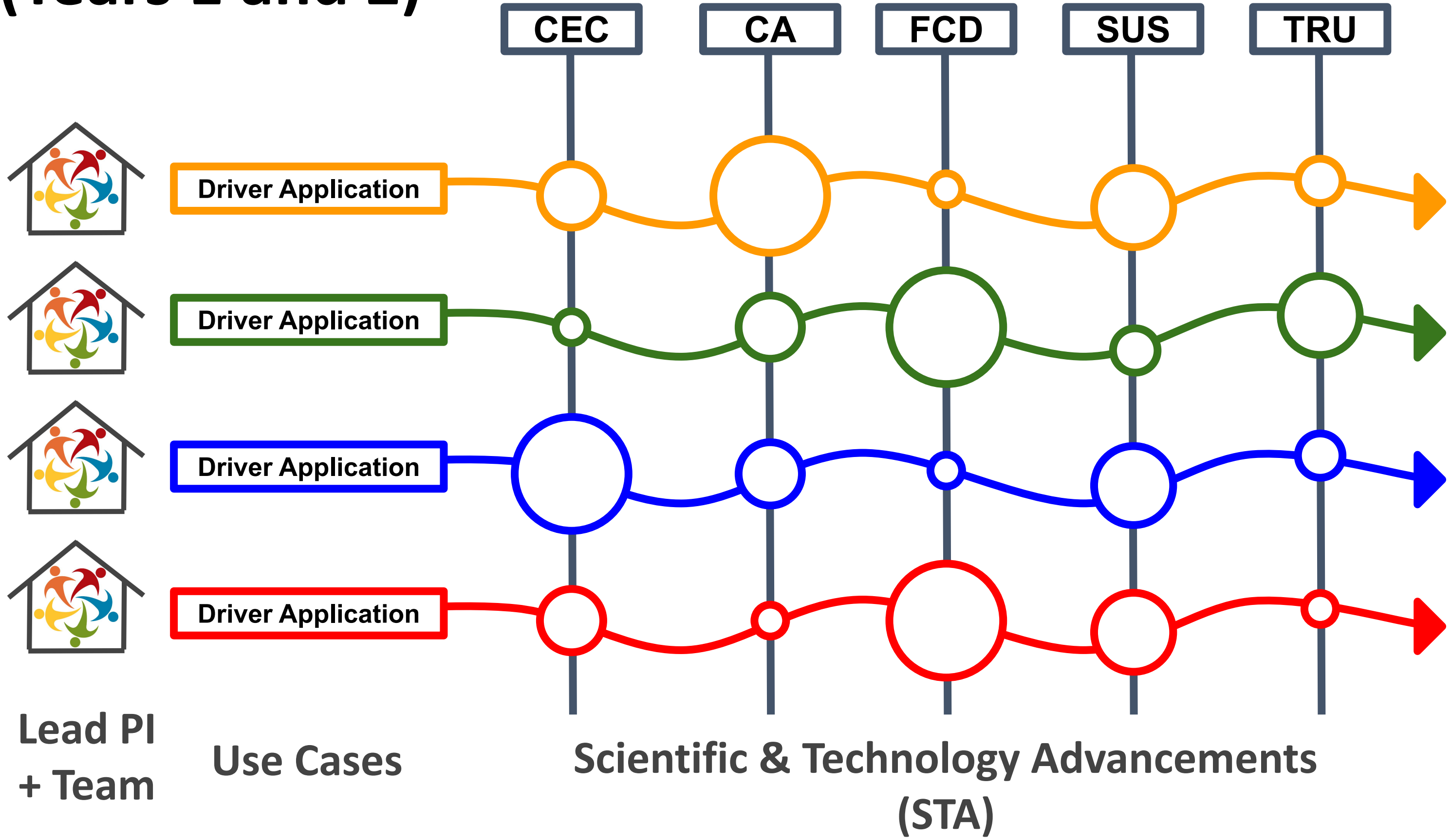


Source: Christian Rothenberg. Fluid Network Planes – An overview of Network Refactoring and Offloading Trends. Keynote at IEEE NetSoft'19, Paris, 2019.

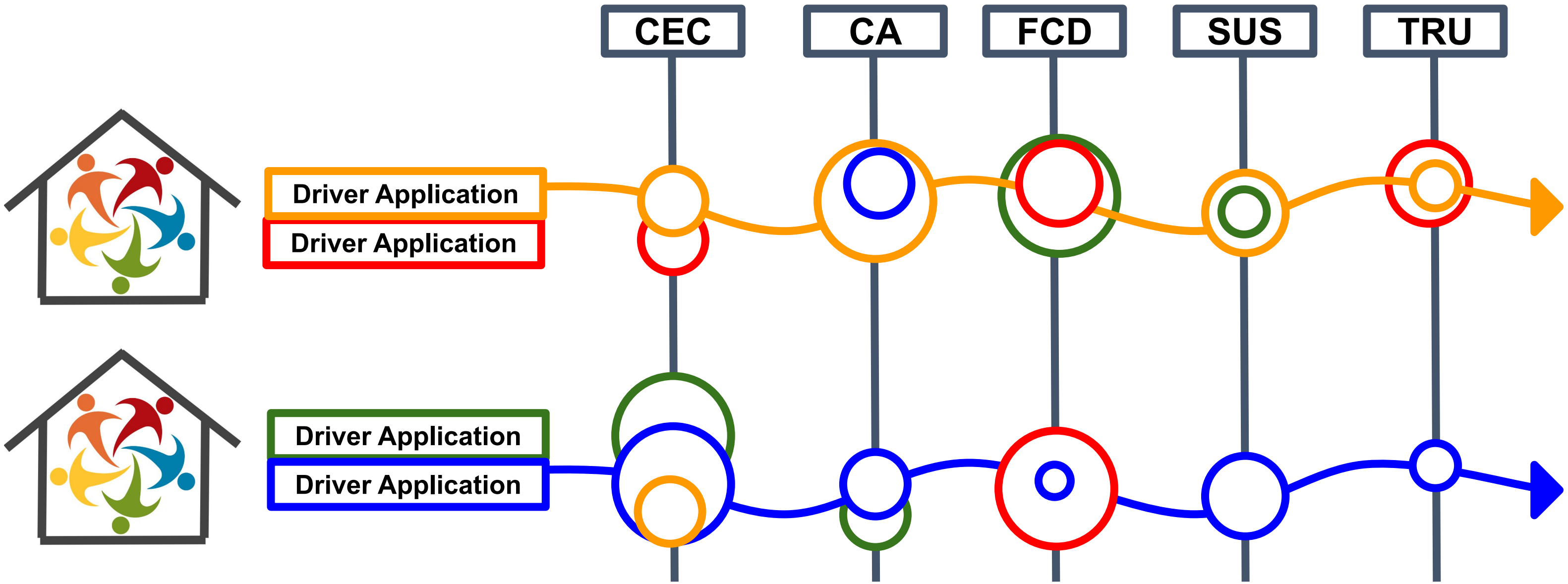
Research Strands Approach



Schedule & Methodology (Years 1 and 2)



Schedule & Methodology (Years 3 to 5)

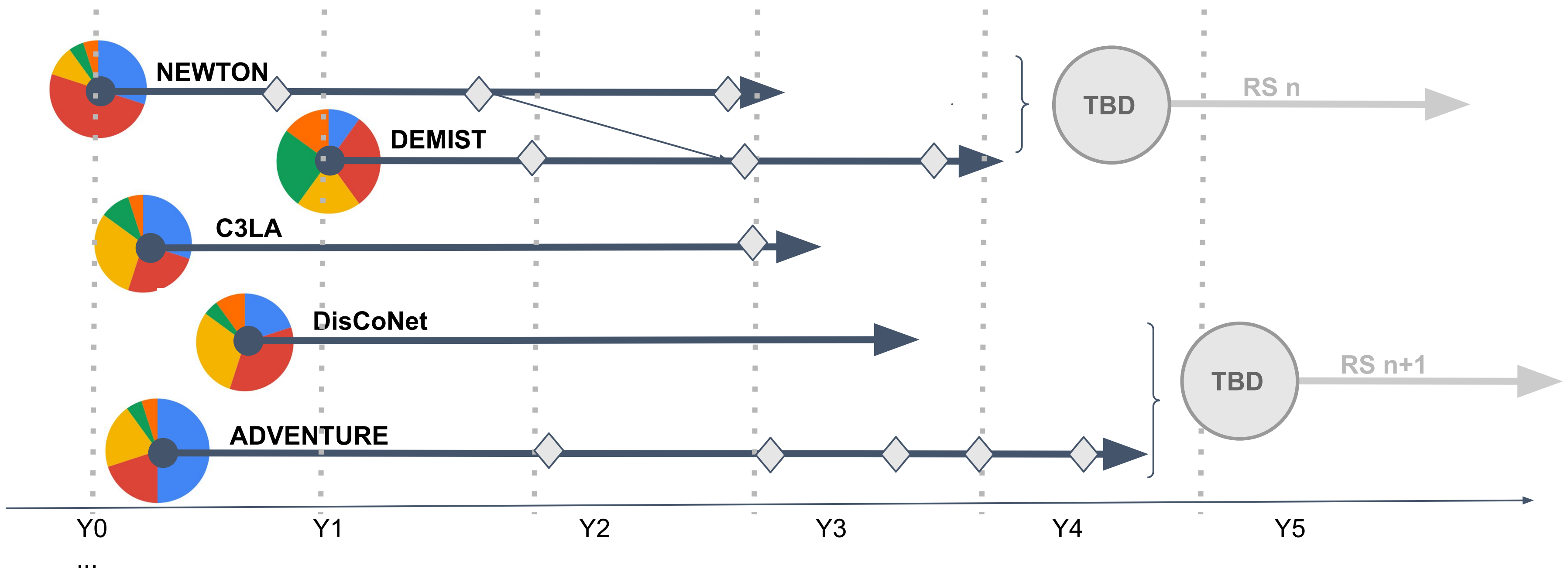


Lead PI
+ Team

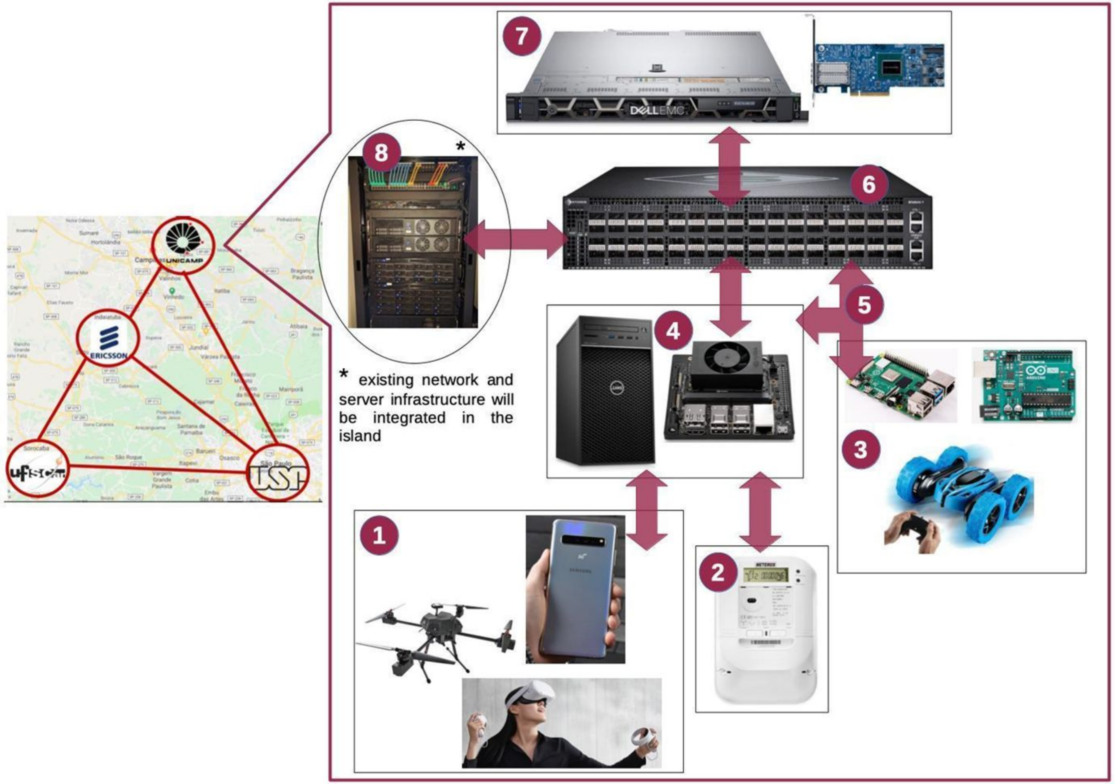
Integrated
Use Case
Pilots

Aggregation of Multidisciplinary
Scientific & Technology Advancements

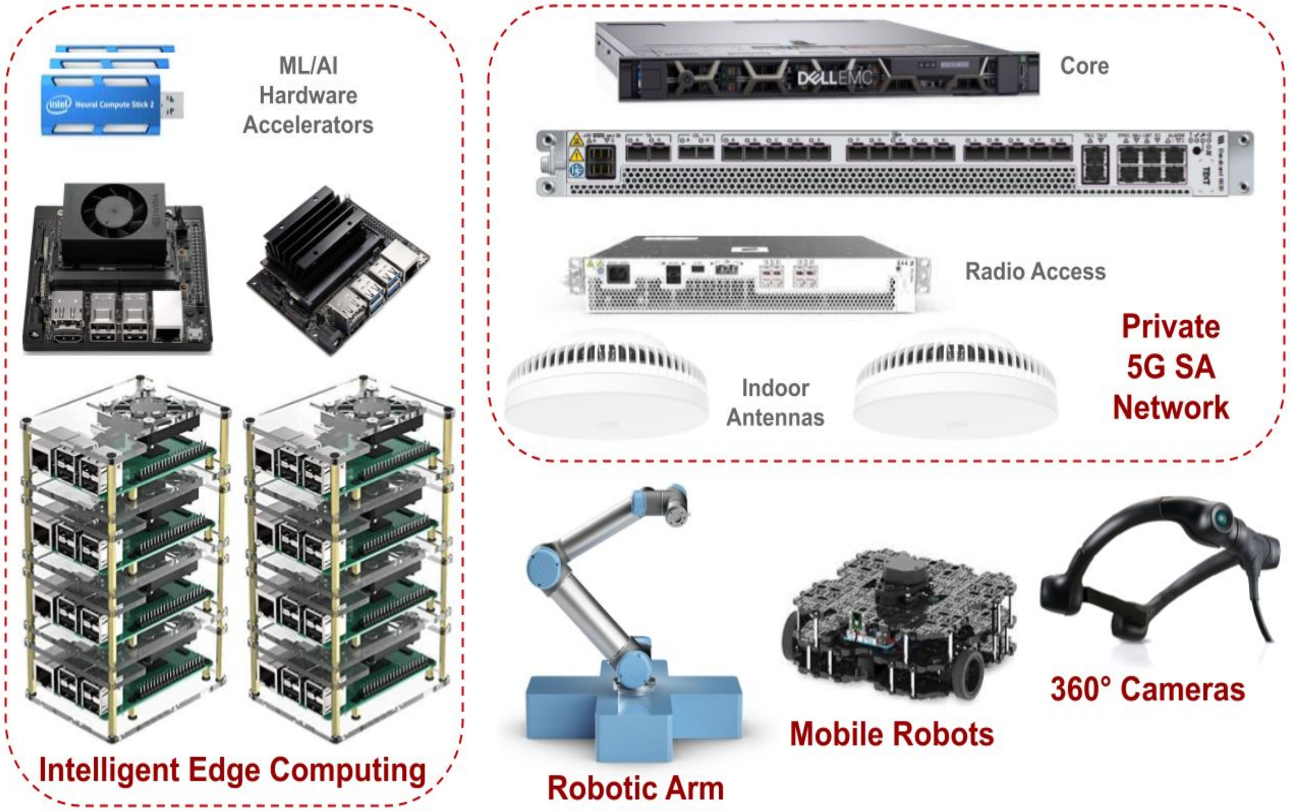
Research Strands vs Topics & Lifecycle



Lab Infrastructure



Lab Infrastructure



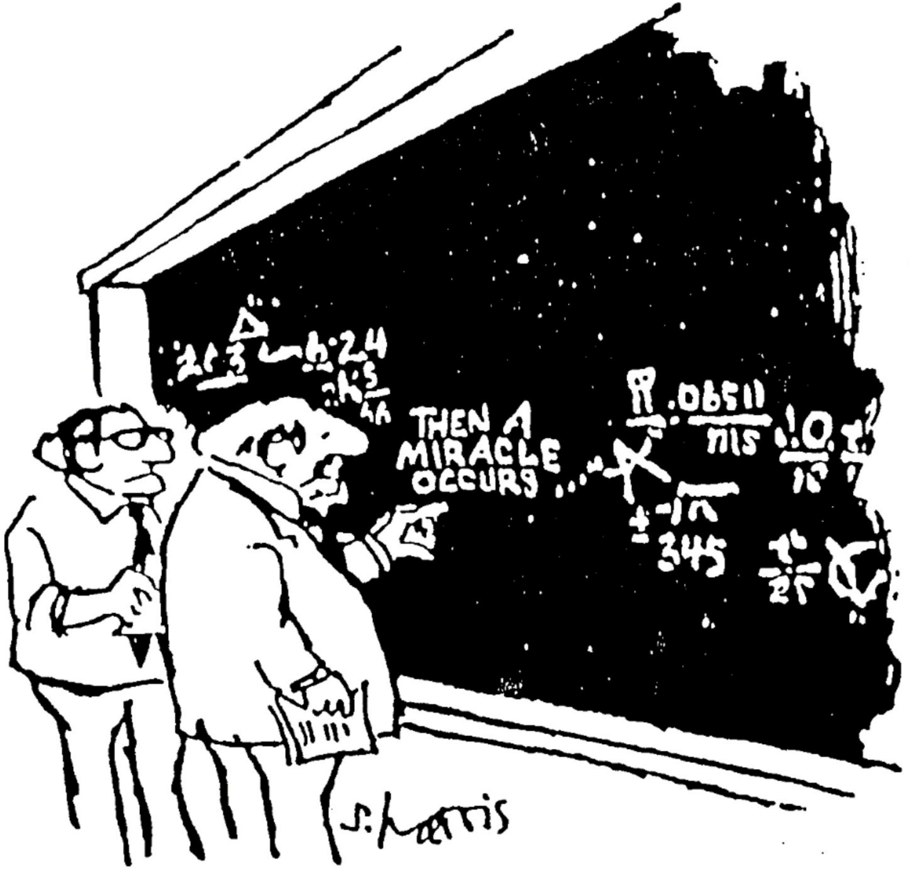
Opportunities for Collaboration

- We (INTRIG + SMARTNESS) will keep leveraging open source projects as research artifacts in support of our research agendas
  - Benefits from potential Red Hat collaboration:
    - Steering decisions on the choice of project, branch, etc.
    - Technical advice for faster time to MVP software setups, doubts, communication, etc.
- We look forward to
  - receiving open research questions for impact contributions to open source projects
  - research internships by our students
    (thinks GSOC like experiences but research-centric)
- We are (also) hiring!
  - MSc, PhD, Post-doc FAPESP scholarships
  - Application deadline: Apr-30 (see FEEC/CPG)

SMARTNESS 2030

Thanks!

More Q&A ?



Credits

For example: 'image: Flaticon.com'. This cover has been designed using images from Flaticon.com

https://www.flaticon.com/free-icon/hiring_5434643