On the role of Open Source in the Network Softwarization trend

Christian Esteve Rothenberg

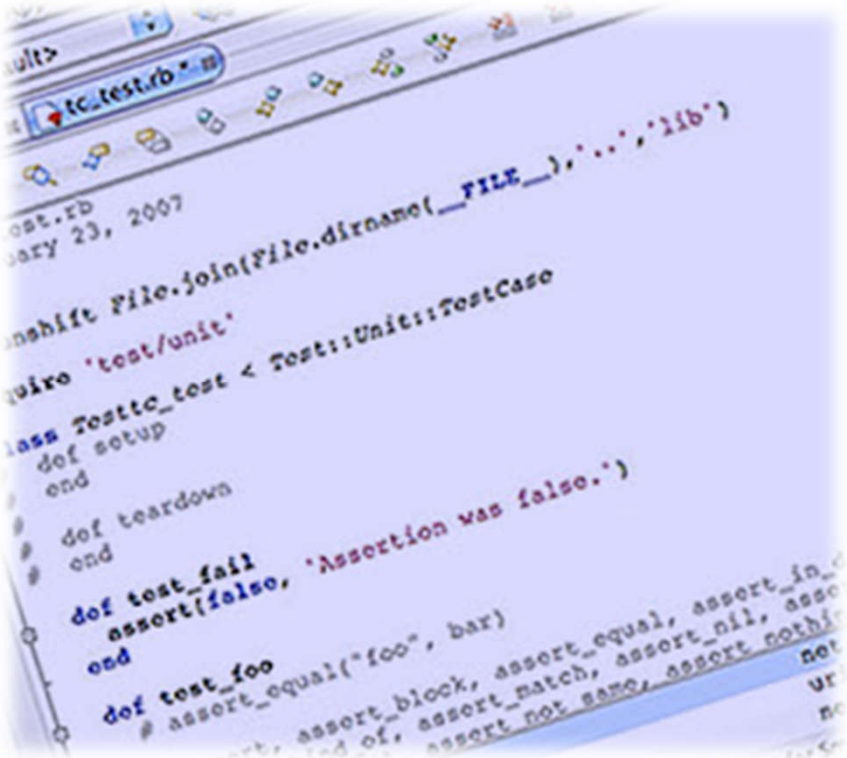
Palestra da Série de Seminários do IC/Unicamp

Nov 18, 2016

Agenda

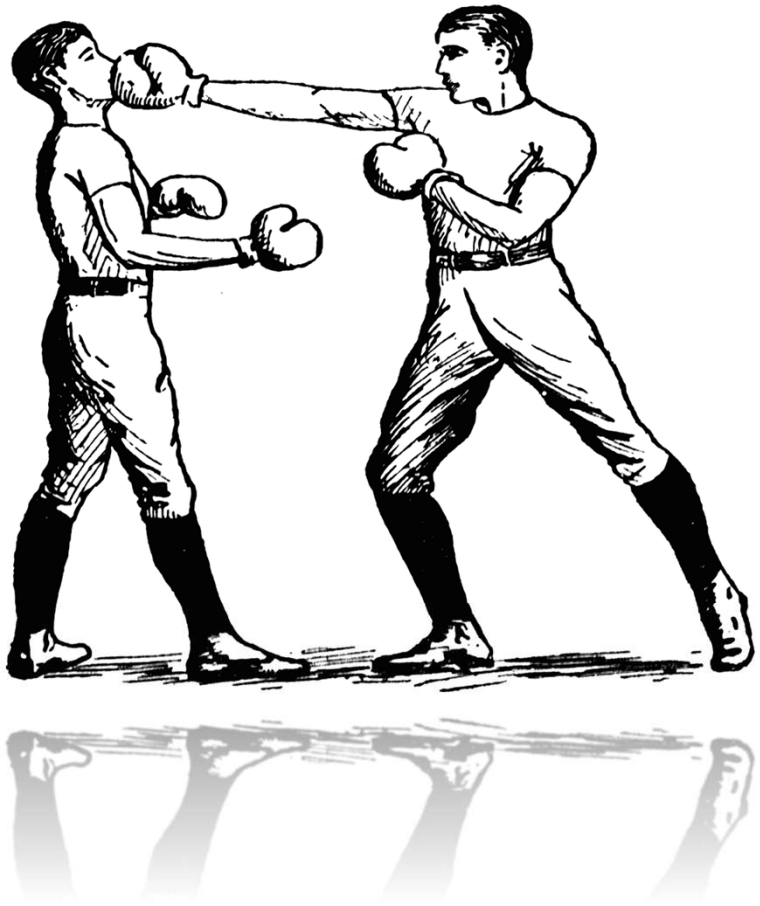
- What is Network Softwarization
- Open Source & Networking
- Some experiences and ongoing activities

Network Softwarization?



Source:
Image (right) http://stunlaw.blogspot.com.br/2013/08/softwarization-tentative-genealogy.html
Image (leftt) https://kc4lmd.files.wordpress.com/2015/01/computer-network-networking-austin.jpg

Network Softwarization



Existing

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

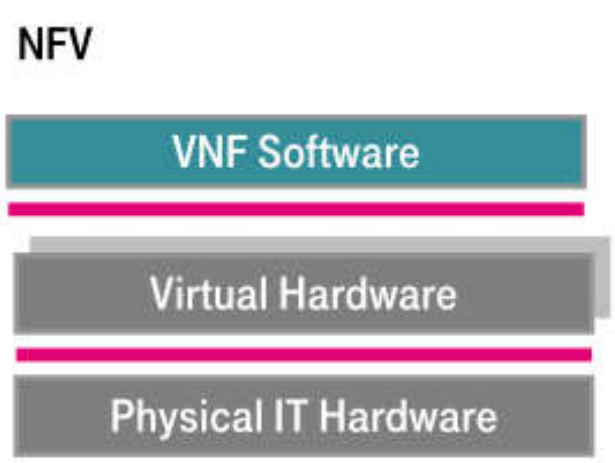
New

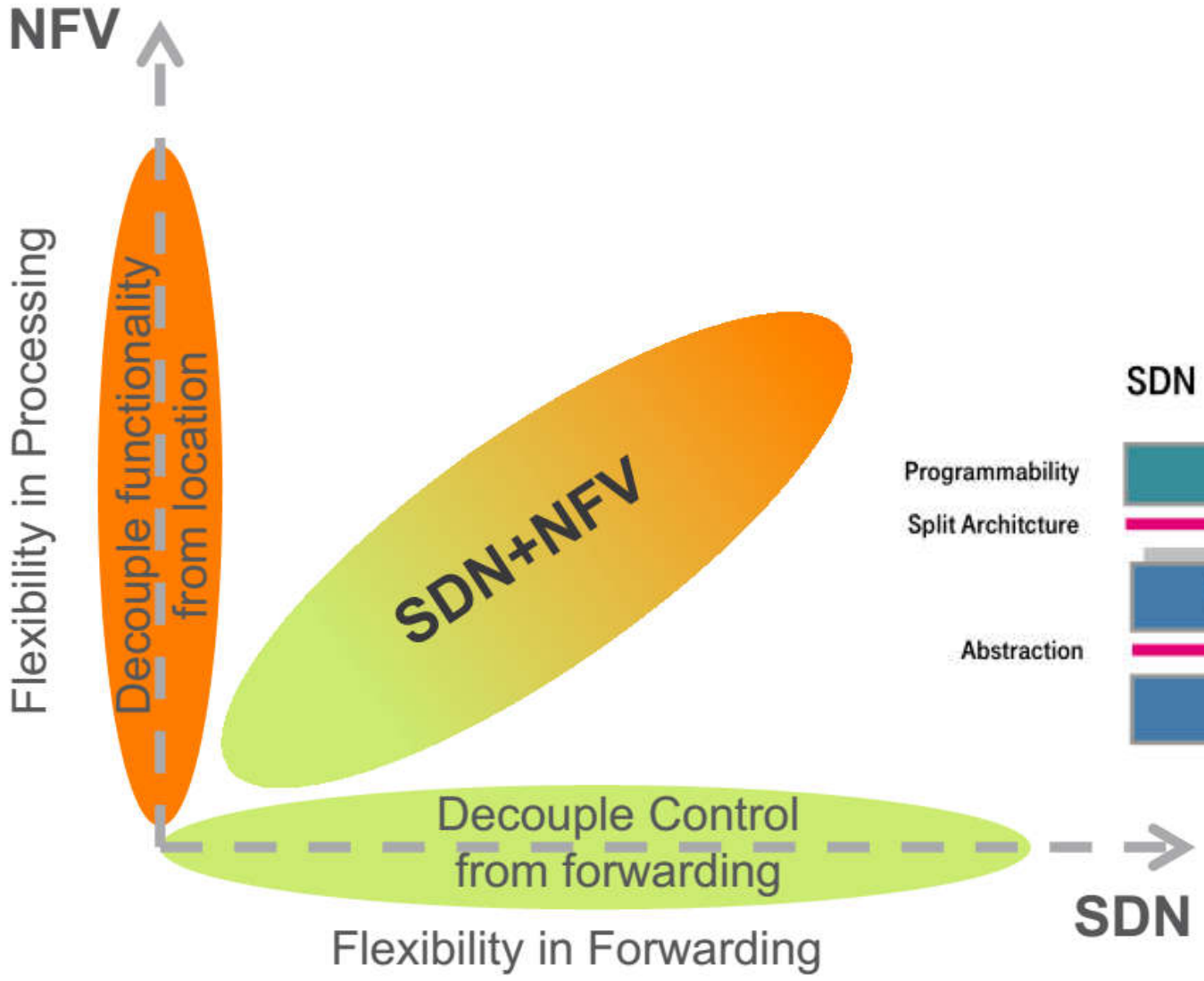
- APIs
- Open Source
- Customer Lead
- Network Function Virtualization (NFV)

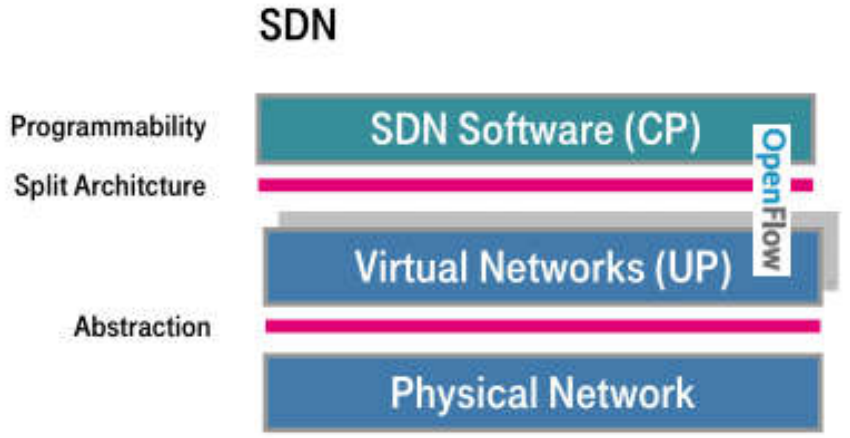
Adapted from: Kyle Mestery, Next Generation Network Developer Skills

Network Softwarization = SDN & NFV

Network Programmability /Flexibility



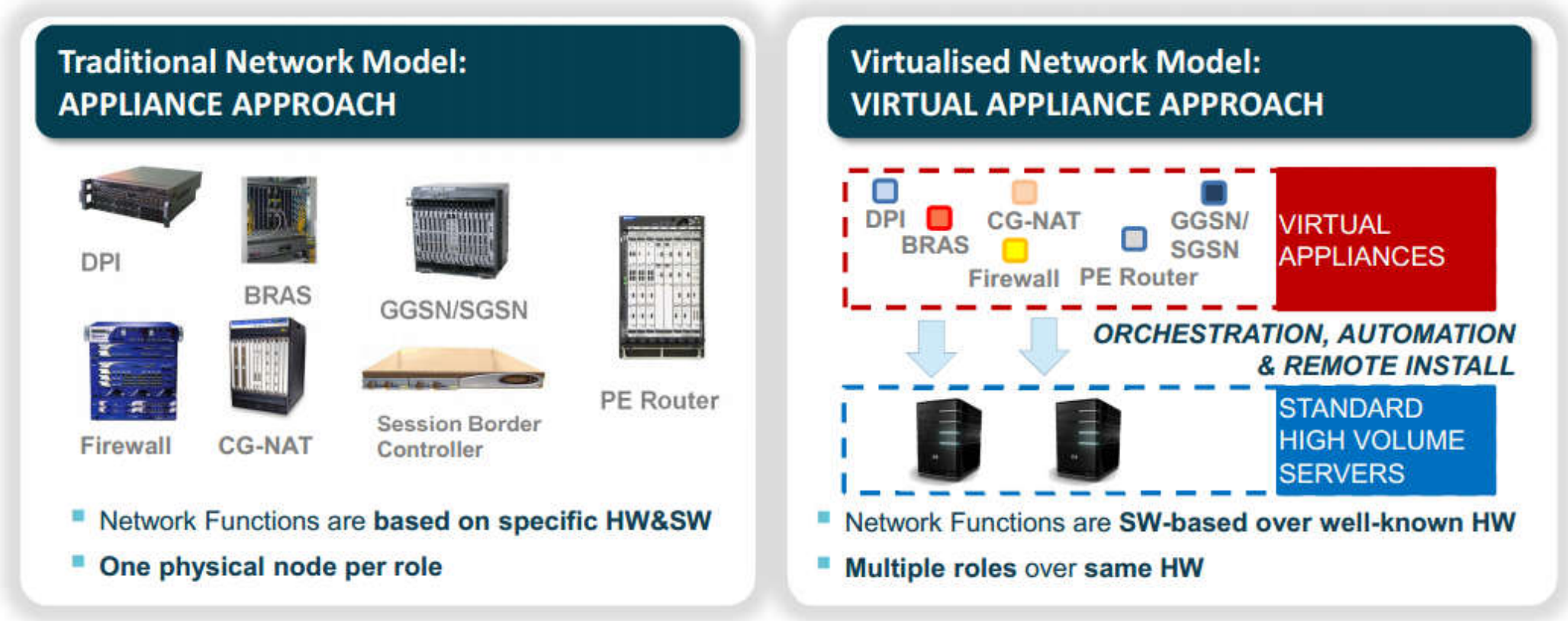




Sources: Ahmad Rostami, Ericsson Research (Kista): http://www.itc26.org/fileadmin/ITC26_files/ITC26-Tutorial-Rostami.pdf and Uwe Michel, T-Systems

Network Function Virtualization (NFV)

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



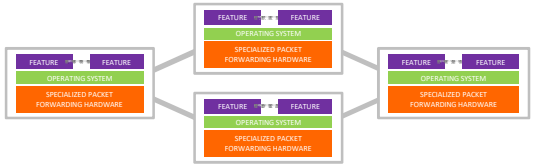
Source: Adapted from D. Lopez Telefonica I+D, NFV

Software Defined Networking (SDN)

Network equipment as Black boxes

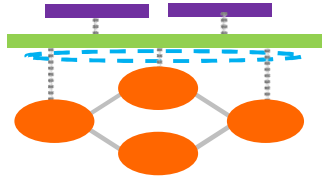
SDN

Open interfaces (OpenFlow) for instructing the boxes what to do

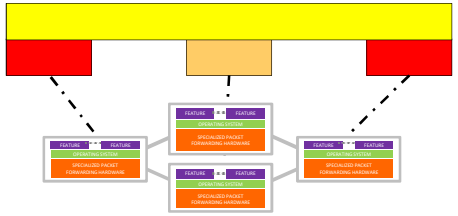


Boxes with autonomous behaviour

SDN

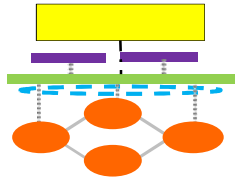


Decisions are taken out of the box



Adapting OSS to manage black boxes

SDN



Simpler OSS to manage the SDN controller

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

NFV vs. SDN

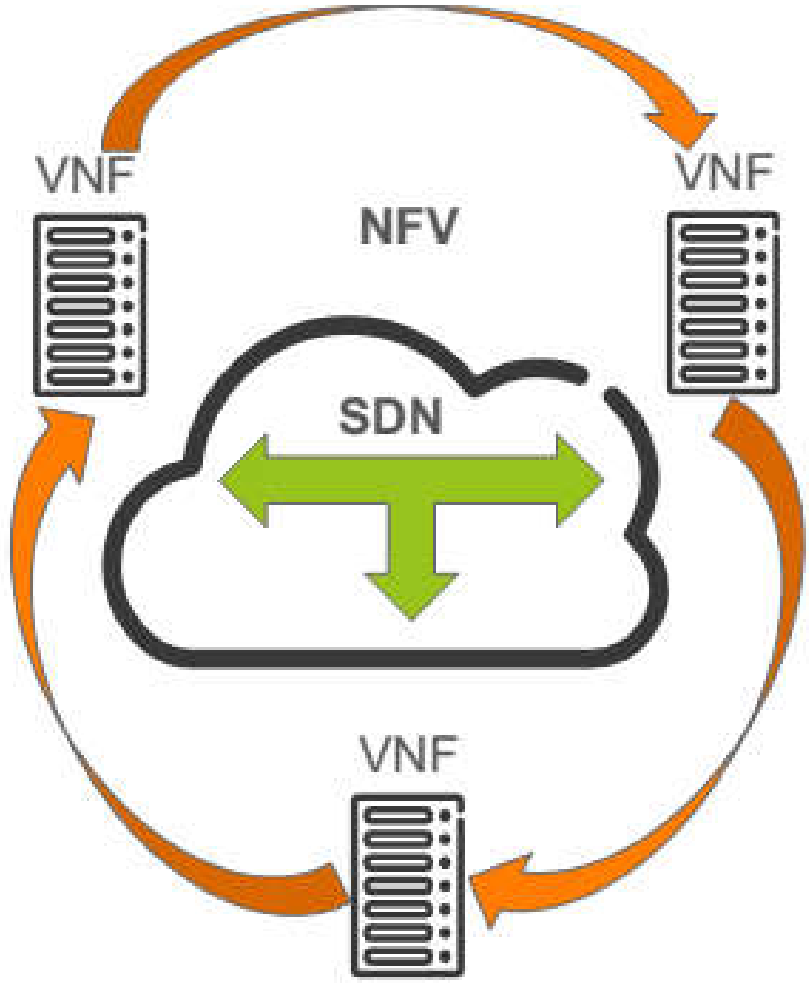
SDN ››› flexible forwarding & steering of traffic in a physical or virtual network environment

[Network Re-Architecture]

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

[Appliance Re-Architecture] (initially)

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



Why NFV/SDN?

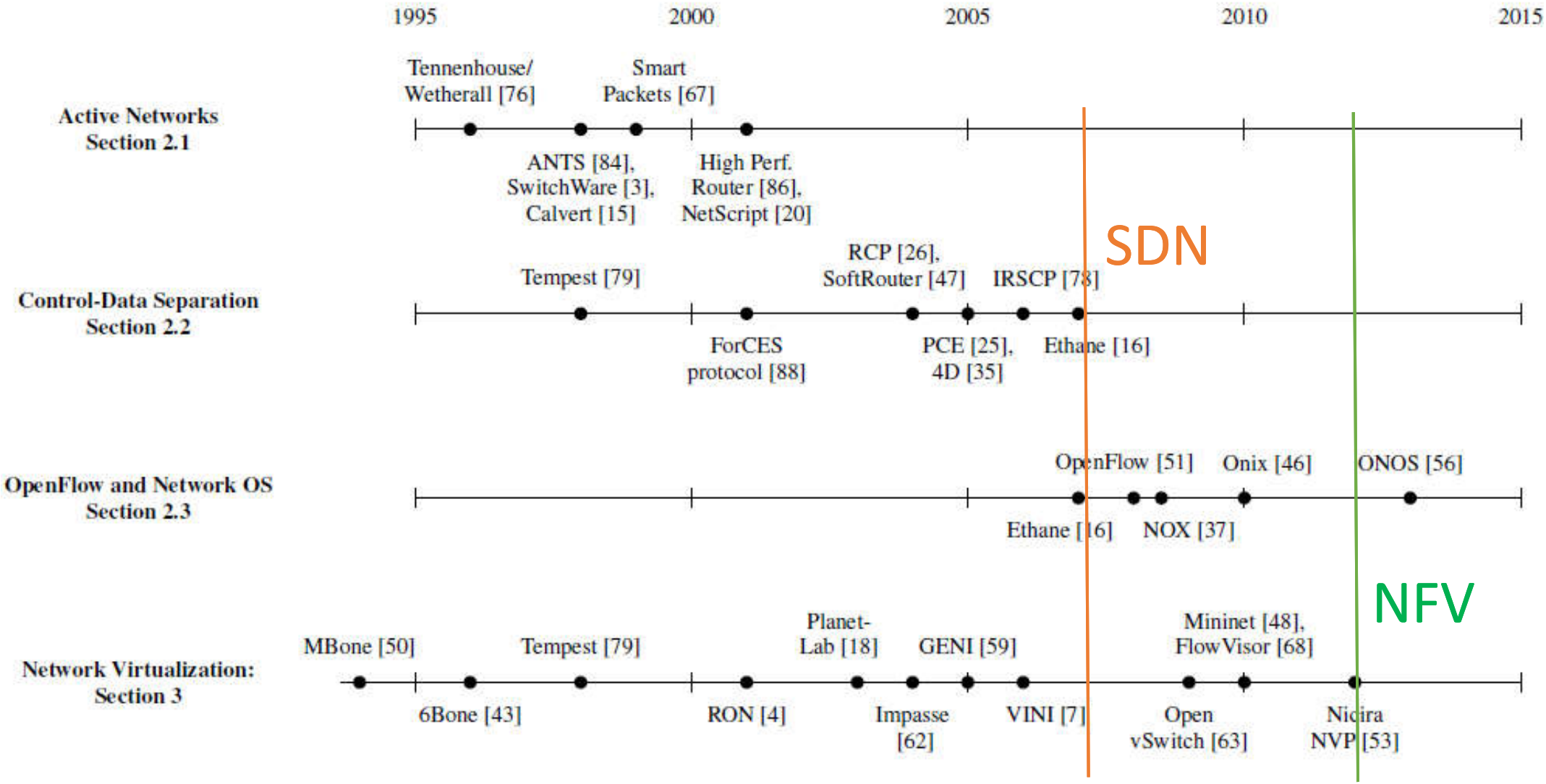
1. **Virtualization:** Use network resource without worrying about where it is physically located, how much it is, how it is organized, etc.
2. **Orchestration:** Manage thousands of devices
3. **Programmability:** Should be able to change behavior on the fly.
4. **Dynamic Scaling:** Should be able to change size, quantity, as a F(load)
5. **Automation:** Let machines / software do humans' work
6. **Visibility:** Monitor resources, connectivity
7. **Performance:** Optimize network device utilization
8. **Multi-tenancy:** Slice the network for different customers (as-a-Service)
9. **Service Integration:** Let network management play nice with OSS/BSS
10. **Openness:** Full choice of modular plug-ins

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

Obs: Core difference on the (complementary) SDN and NFV approaches (how?).
(SDN :: decoupling of control plane, NFV : decoupling of SW function from HW)

Source: Adapted from Raj Jain

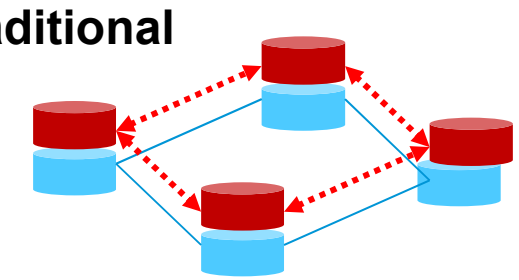
Intellectual History of Programmable Networks

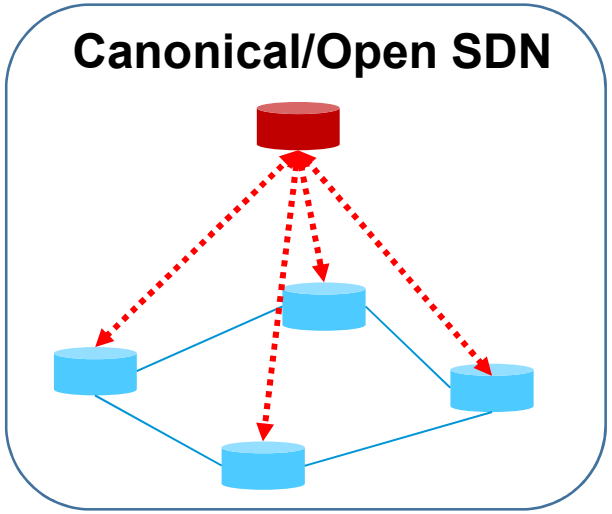


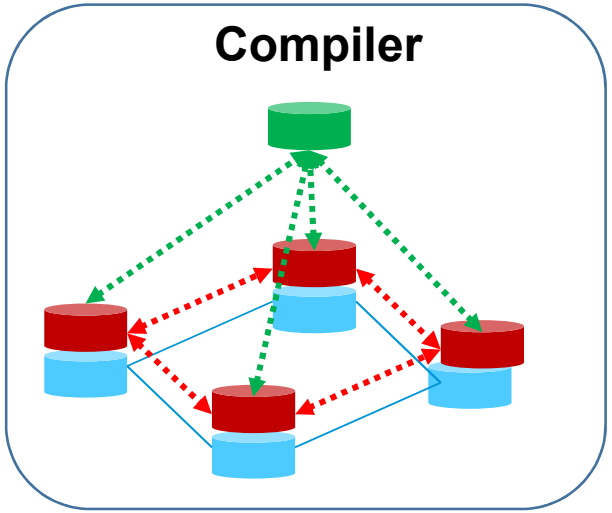
Source: N. Feamster, J. Rexford, E. Zegura. The Road to SDN: An Intellectual History of Programmable Networks. http://gtnoise.net/papers/drafts/sdn-cacm-2013-aug22.pdf

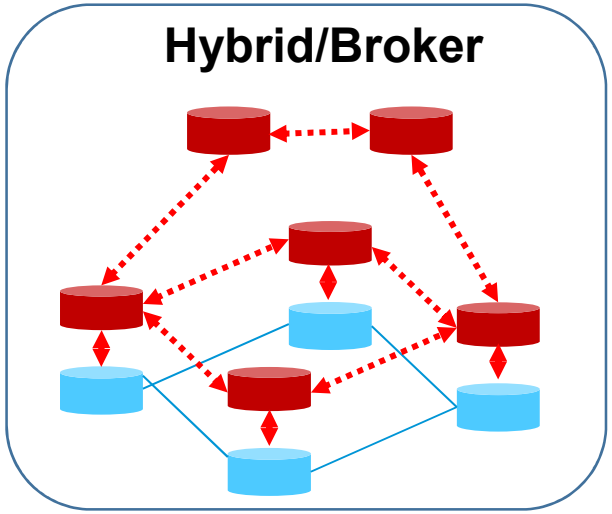
Different SDN Models

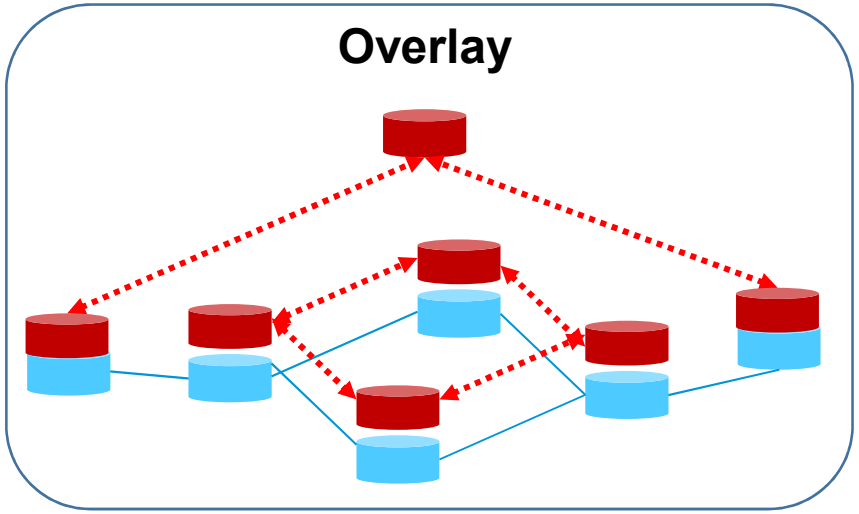
Control-plane component(s) Data-plane component(s)



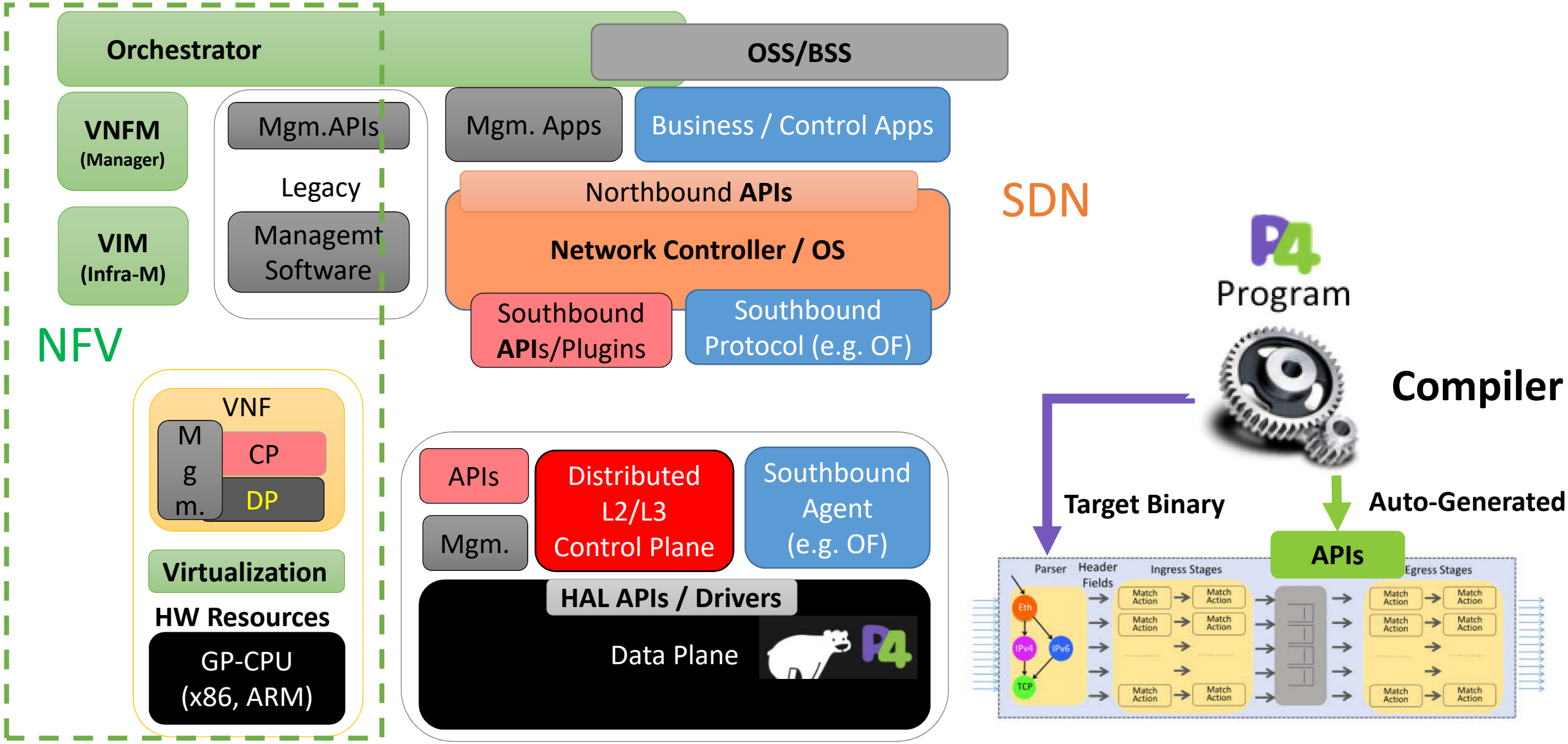








Different SDN Models to Program / Refactor the Stack



Open Source & Networking

Open Source Met Networking long time ago

Up until recently, mainly
(1) software appliances with limited scope;
(2) management plane

Functional level	FLOSS solutions	Overview	Legacy proprietary approach
Availability	Pacemaker, HAProxy, NGINX	Deployment and management of redundant, high-availability clusters	Network Load Balancing appliance
	Nagios, Cacti, Zabbix	Operation and performance monitoring software	Proprietary monitoring software
Security	IPCop, PFSense, Smoothwall, M0n0wall	Linux and BSD-based firewall systems	Firewall and IDS appliances
	OSSEC, Snort	Host and network-based Intrusion Detection Systems	
	SecurityOnion, OSSIM	Security Information Enterprise Management	
	PacketFence	Network Access Control (802.1x, captive portal, etc.)	NAC appliance
Management	cfEngine, Ansible, Puppet, Chef, mcollective, SaltStack	Configuration management and/or orchestration platforms	Proprietary, vendor-attached provisioning and management software
	SpaceWalk, Foreman	Server provisioning and configuration management	
Caching	Squid, Apache Traffic Server, Varnish Cache	Caching for performance at either client end or server end	Caching appliance
Storage	DRBD, GFS2, GlusterFS	Distributed and/or replicated filesystems	Storage appliance
	FreeNAS, OpenFiler	Open source storage management and sharing (iSCSI, NFS, etc.)	
Network Services (IPAM etc)	BIND, djbdns, TinyDNS, DNSmasq	Network services that provide directory services and address management services	---
Routing	Quagga, XORP, BIRD	Routing protocols implementation suites	Vendor lock between route engine and dedicated silicon router
Device OS / firmware	DD-WRT, OpenWRT, Gargoyle, PicOS	Open source firmware for wireless access points and Ethernet switches	Proprietary wireless firmware/protocols
Server virtualization	OpenVZ, LXC	OS kernel virtualization (container-based emulation)	Underutilized server hardware
	Xen, KVM	Server hardware full and/or para-virtualization	

Why Open Source in Networking?

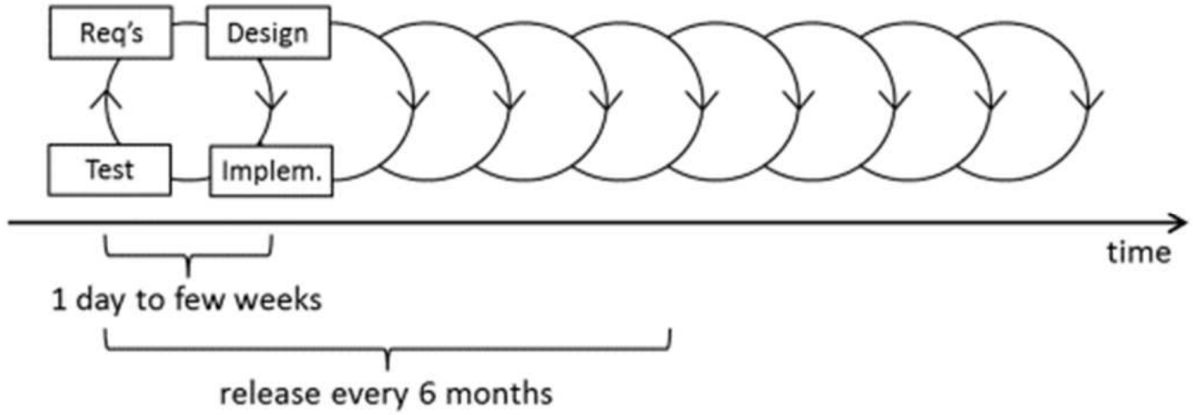
- Higher reliability, more flexibility
- Faster, lower cost, and higher quality development
- Collaborative decisions about new features and roadmaps
- A common environment for users and app developers
- Ability for users to focus resources on differentiating development
- Opportunity to drive open standards

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

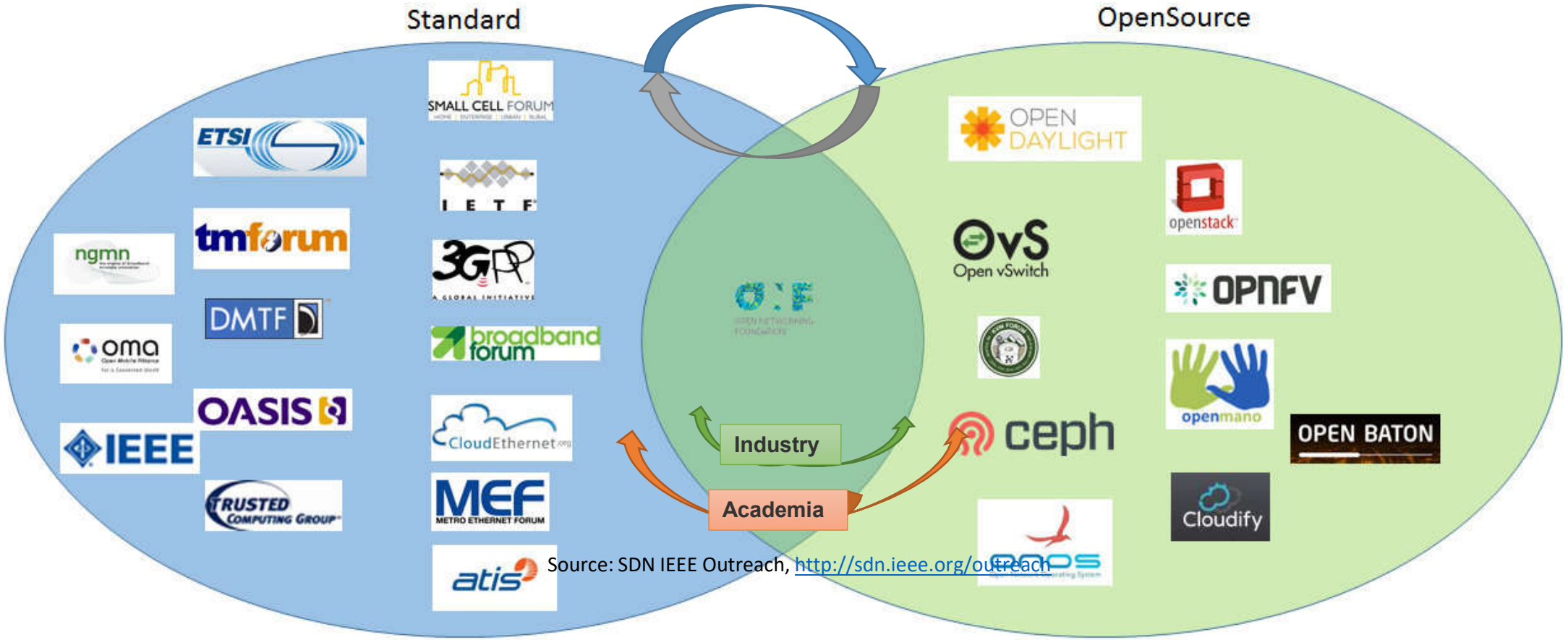
Source: Open Source in a Closed Network – Prodip Sen (OPNFV Summit 2015)

Why Open Source in Networking?

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



Standard Development & Open Source Organizations



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

Foundations : The New Player

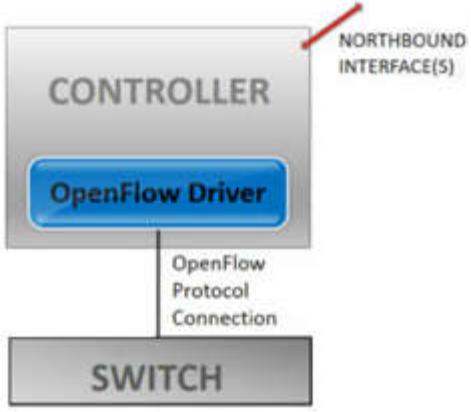
Target Collaboration

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

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

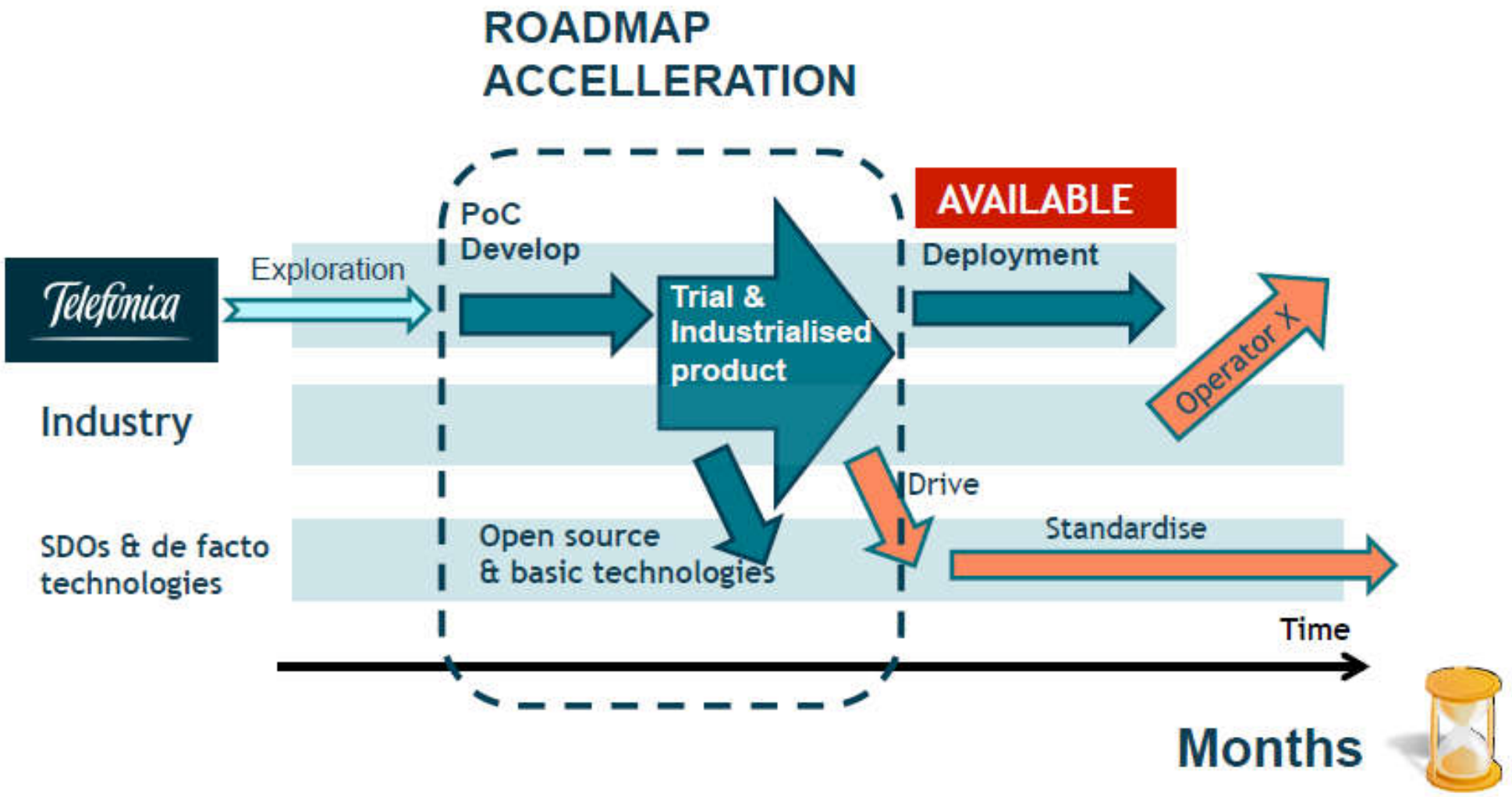
Recent Events involving Open Source

- The Battle for the Hypervisor Switch[1]
  - Open vSwitch vs Nexus1000v vs Hyper-V Virtual Switch
- The Battle for the Cloud: Stack Wars
  - Stack wars: OpenStack v. CloudStack v. Eucalyptus
- The Battle for the SDN control platform
  - OpenDaylight vs Floodlight vs etc. etc.
- The Battle for the SDN Northbound APIs
  - ONF vs OpenDaylight vs OpenStack vs etc.
- ONF OpenFlow Driver Competition
  - An open source driver to accelerate developments



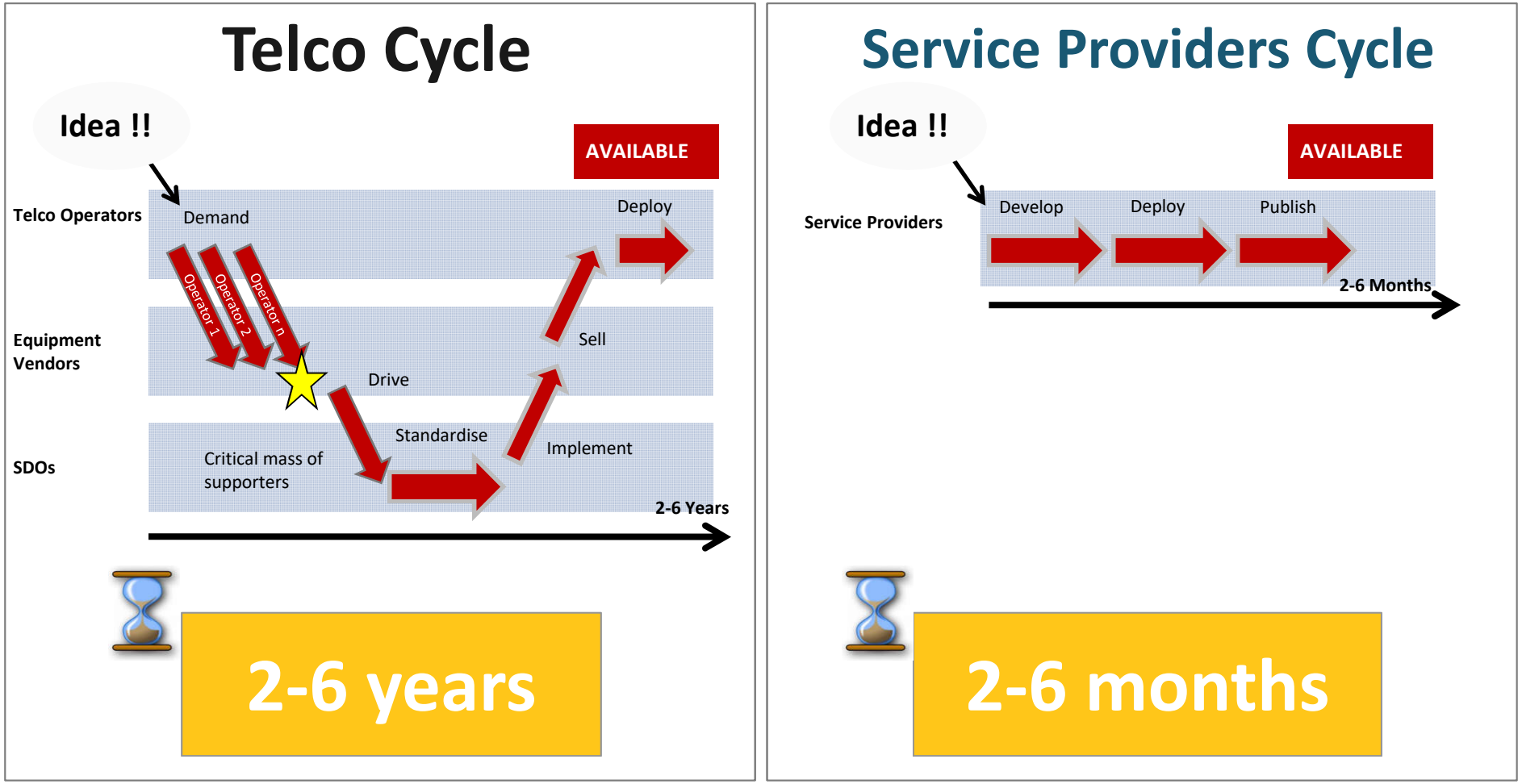
[1]http://www.networkworld.com/community/blog/battle-hypervisor-switch-and-future-networking

NFV >>> Accelerating Transformation



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

Sisyphus on Different Hills



Source: Telefonica I+D / ETSI NFV

Open Source SDN/NFV & Standardization

Evolving and accelerating the path to standardization

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

Further Reading:

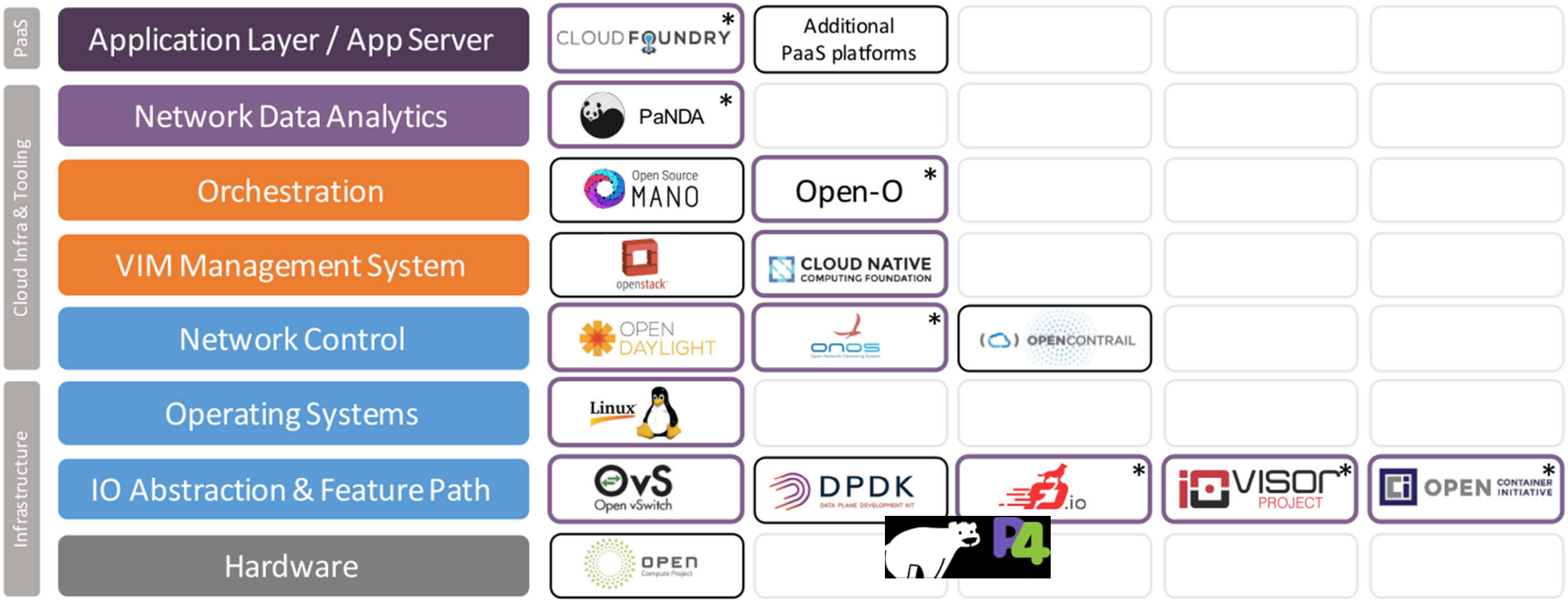
- IETF Trends and Observations *draft-arkko-ietf-trends-and-observations-00*
- Source of table: "*When Open Source Meets Network Control Planes*." In IEEE Computer (Special Issue on Software-Defined Networking), vol.47, Nov. 2014.
- Source of figure: A. Manzalini et al., "*Towards 5G Software-Defined Ecosystems*"

Open Source SDN Projects (2014 snapshot)

SDN Realm	Open Source Project Name
Benchmarking	Cbench (GPLv2, C/Perl/UNIX shell), OFLOPS (GPL, C)
Debugging / Testing / Simulation	ndb, OFRewind, STS (Apache, Python), OFDissector (BSD, C), liboftrace (BSD, C), OFTest (BSD, Python), Mininet (BSD, Python), fs-sdn (GPL, Python), ns-3 (GPL, C++), TestON (GPL, Python)
Verification	Hassel (GPL, Python), NetPlumber (GPL, Python), NICE (BSD, Python), FlowChecker, OFTEN
Control & Management Apps	Topology discovery (GPL, many), HostTracker (GPL, many), Plug-n-Serve (BSD, C++), Aster*x (BSD, C++), FlowScale (Apache, Java), SNAC (Python/C++), Odin (Apache, Python), PANE (BSD, Haskell), FRESCO (GPL, Python/C++), OSCARS (BSD, Java), RouteFlow (Apache, Python/C/C++/Java), Open DayLight (Eclipse, Java)
Programming Abstractions / Compilers / Isolation	FatTire, Flog, FML, Frenetic (GPL, OCaml), HFT, NetCore, Nettle, Procera, Pyretic (BSD, Python), Maple (Pyhon), FlowN, LibNetVirt (GPL, C/Python), OpenStack Neutron (Apache, Python)
Controller Platforms	NOX (GPL, C++); POX (GPL, Python); Maestro (LGPL, Java); Beacon (GPL, Java); Floodlight (Apache, Java); Ryu (Apache, Python); Trema (GPL, C/Ruby); FlowER (MIT, Erlang); NodeFlow (GPL, javascript); Mul (GPL, C); OpenDaylight (Eclipse, Java); ONOS (Java); OpenContrail (Apache, C++ / Python)
Data Plane Virtualization	FlowVisor (BSD, Java), PortVirt (BSD, C), Expedient (Apache, Python), OpenVirteX (Apache, Java)
OpenFlow Protocol Libraries / Southbound Driver	OFLib-Node (BSD, JavaScript), OpenFlowJ (BSD, Java), OpenFaucet (Apache, Python), Pylib-OpenFlow (BSD, Python/C++), freeflow (Apache, C/C++), xDPd/ROFL (MPL, C/C++), libfluid (Apache, C/C++)
Data Plane Implementations	NetFPGA (BSD, C/Verilog), Open vSwitch (Apache, C), Reference design (BSD, C), ofsoftswitch13 (BSD, C), OpenWRT/Pantou (GPL, C), Switch Light (Eclipse, C), Indigo Virtual Switch (Eclipse, C), LINC-Switch (Apache, Erlang)

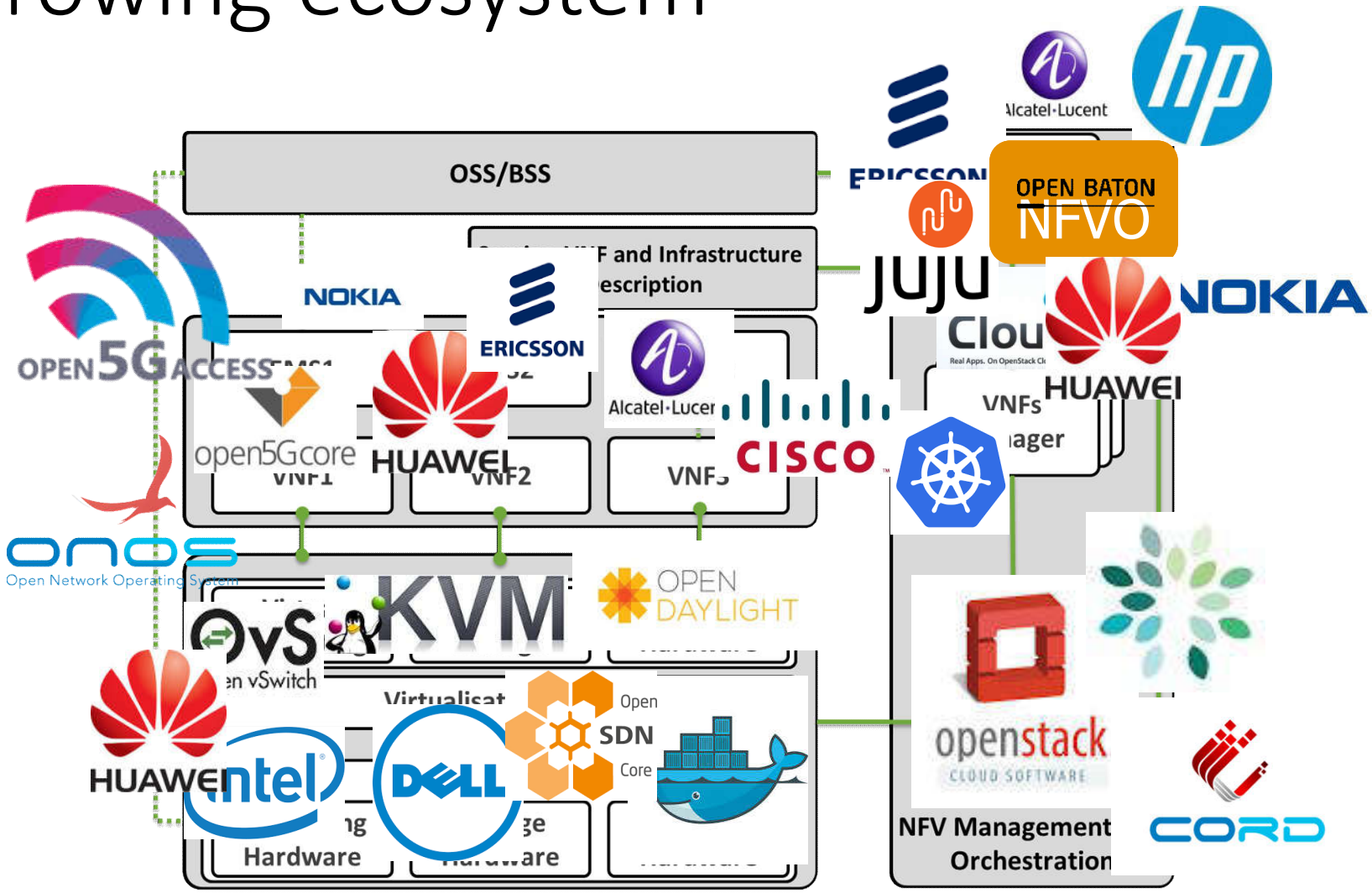
Source: "*When Open Source Meets Network Control Planes.*" In IEEE Computer (Special Issue on Software-Defined Networking). 2014.

Open Source Building Blocks
2015 – 2016: Several New Projects



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

NFV Growing ecosystem



A growing ecosystem…

5G Related Open Source

Experimentation

Testbeds

5G Berlin

Open SDN Core

5G playground
Fraunhofer

open5Gcore

open5Gphotonics

OPEN 5G ACCESS

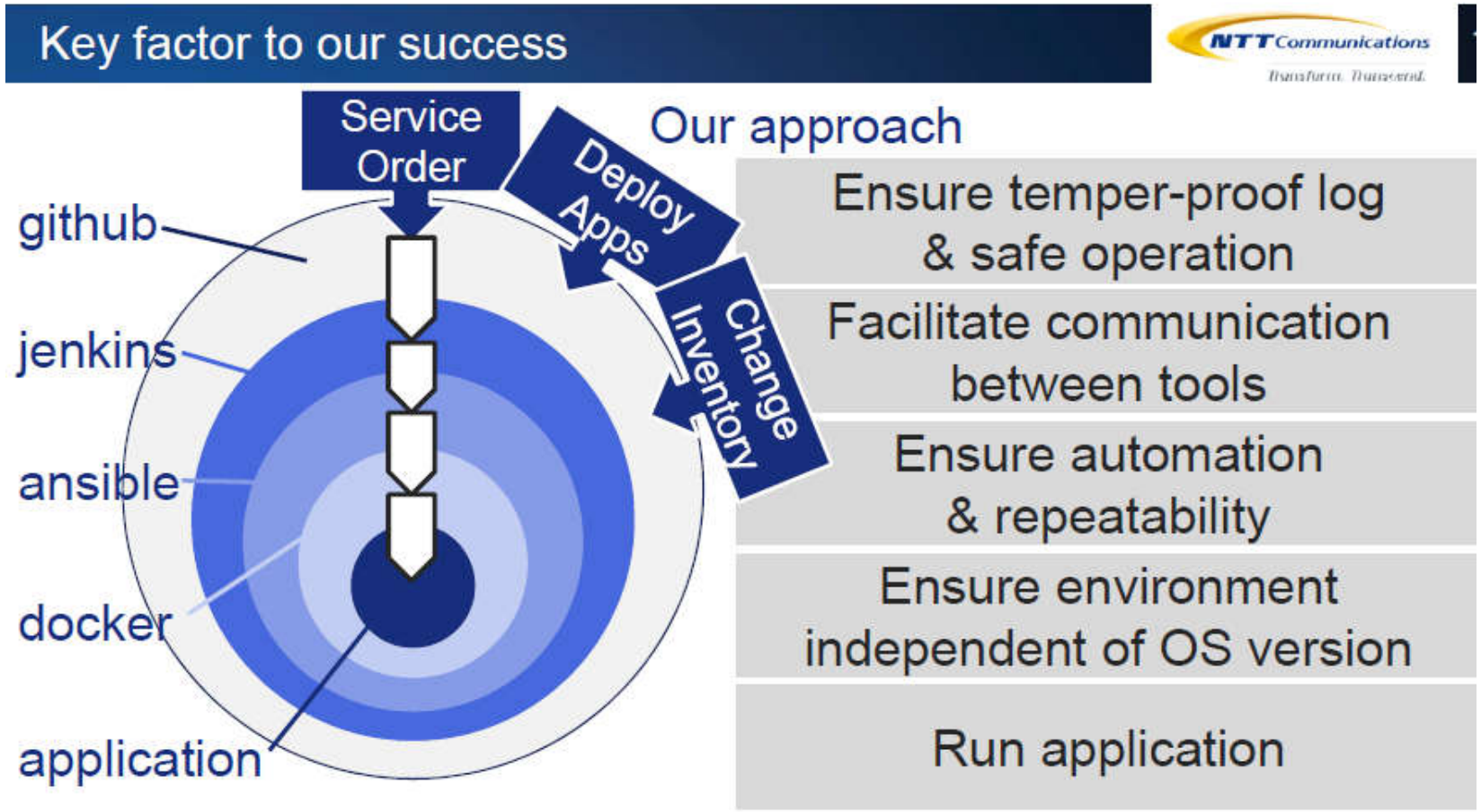
Source: http://www.5g-berlin.org/

Continuous Integration Environment

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

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

Example: NTT (TM FORUM 2016)



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

All of This Leads Us To …

Software Defined Networking

DevOps

DevOps Defined Networking

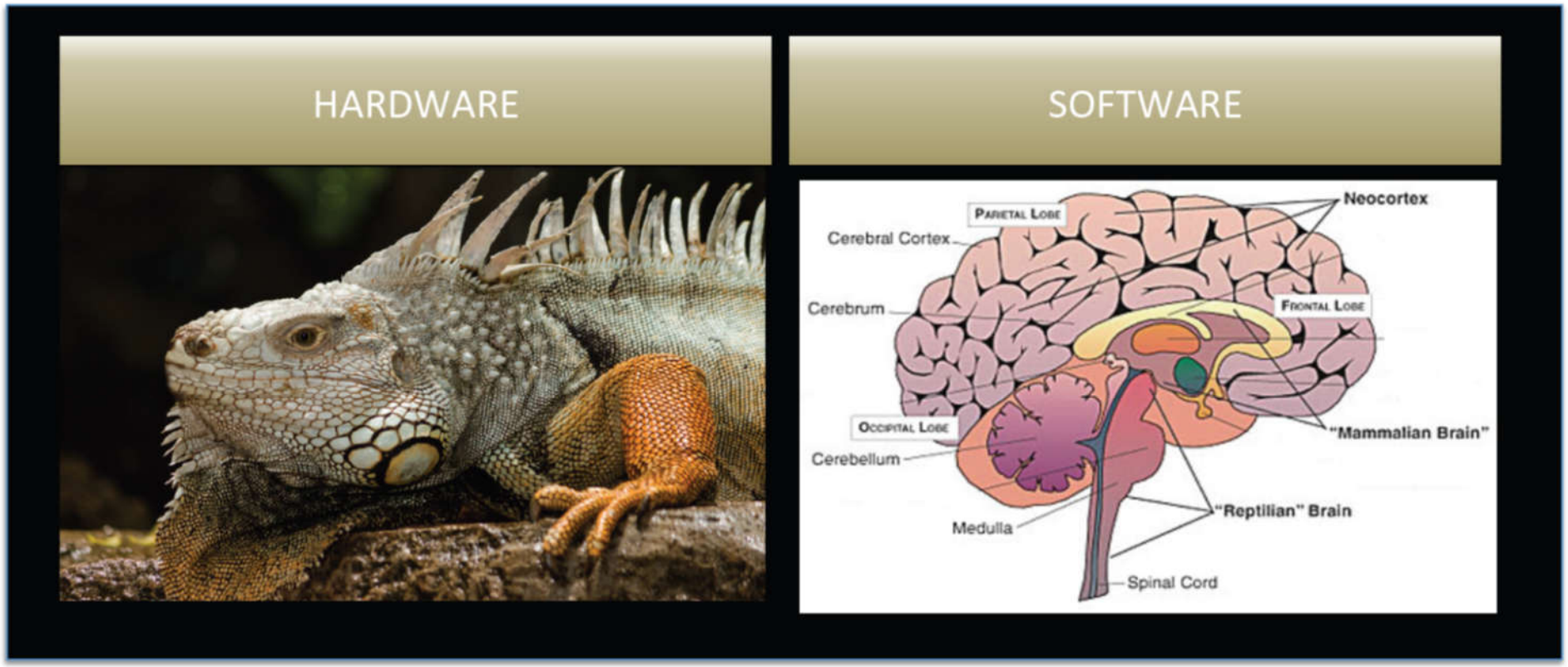
Slide courtesy: Kyle Mestery

Detour on Layered Architectures and Complexity

And what does this have to do with SDN and Open Source

Trend: The Evolution of Intelligence

Precambrian (Reptilian) Brain to Neocortex → Hardware to Software



- Key Architectural Features of Scalable/Evolvable Systems
  - RYF-Complexity (behavior)
  - Layered Architecture
  - Bowties and Hourglasses
  - Horizontal Transfer (H*T)
  - Protocol Based Architectures

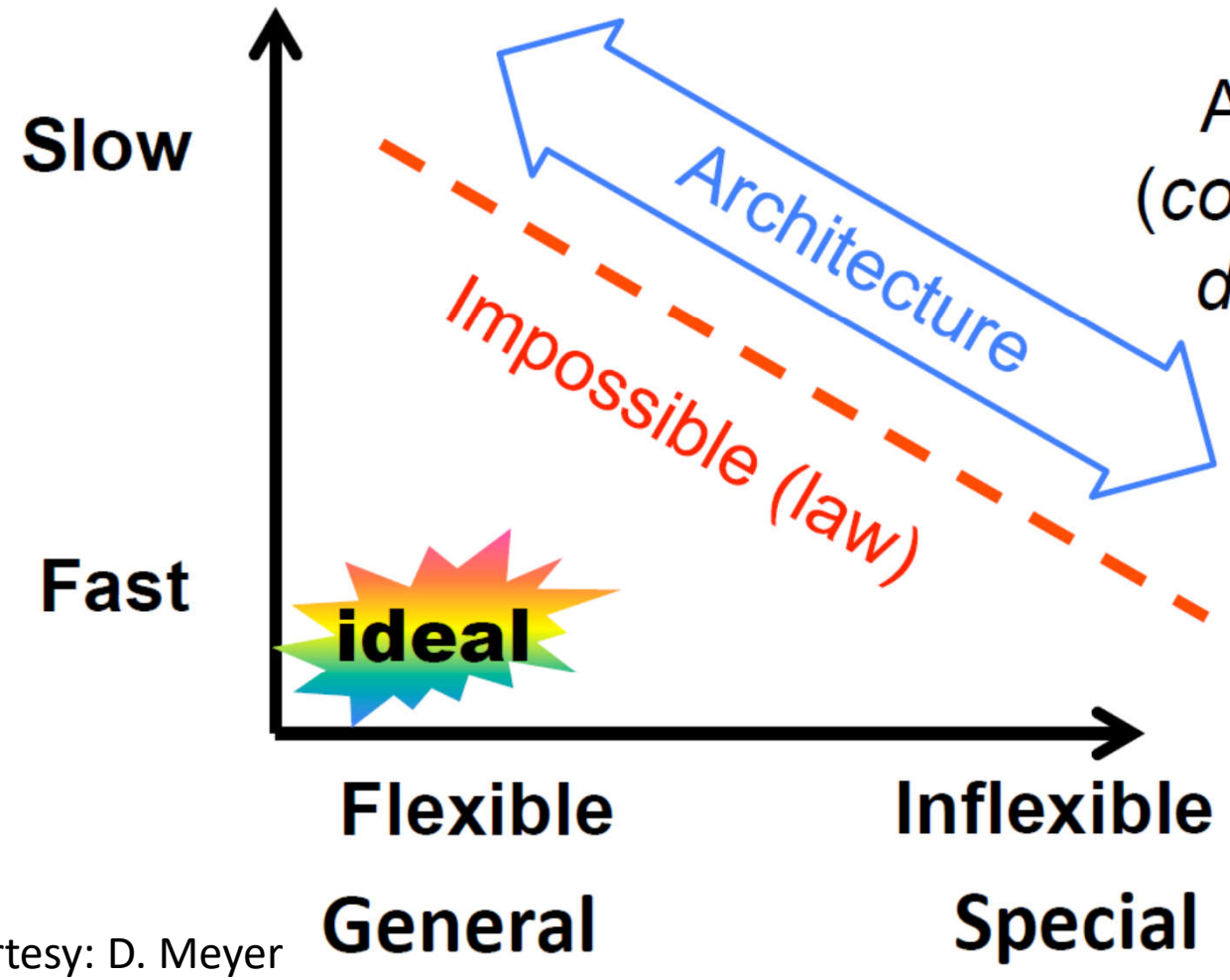
Once you have HW
its all about code[1]…

Slide courtesy: D. Meyer

1 http://www.joelonsoftware.com/articles/Biculturalism.html

Universal Laws and Architectures (Turing)

Layering, Formal Systems, Hard Tradeoffs

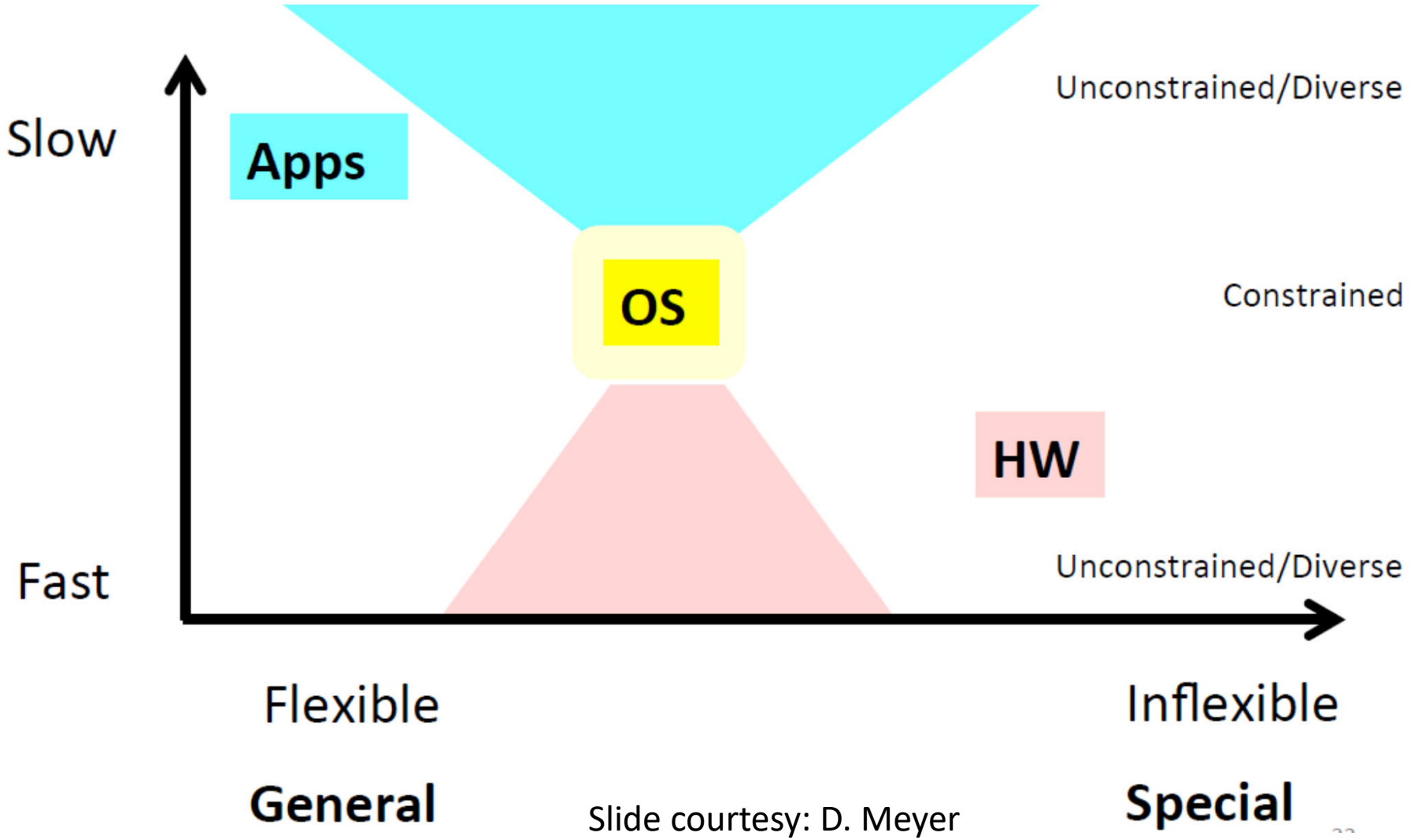


Architecture
(*constraints that deconstrain*)

Slide courtesy: D. Meyer

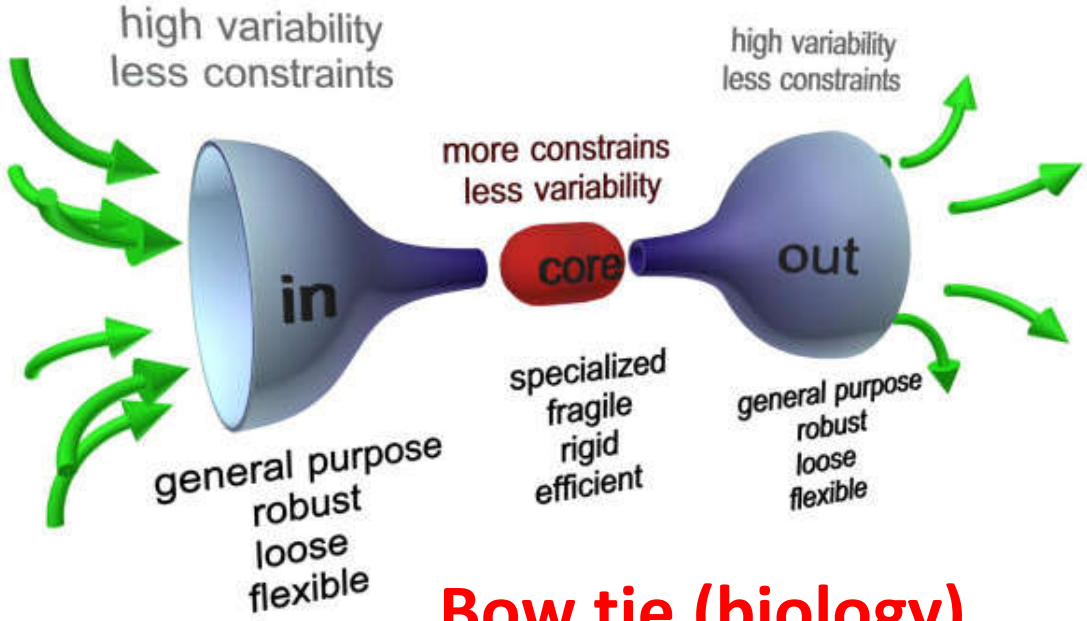
Original slide courtesy John Doyle

Overlaying Tradeoffs

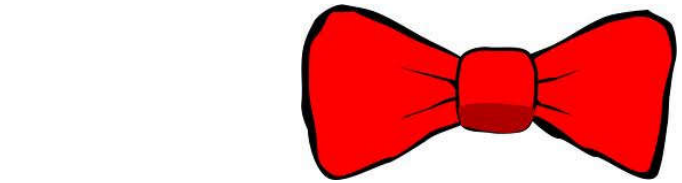


Slide courtesy: D. Meyer

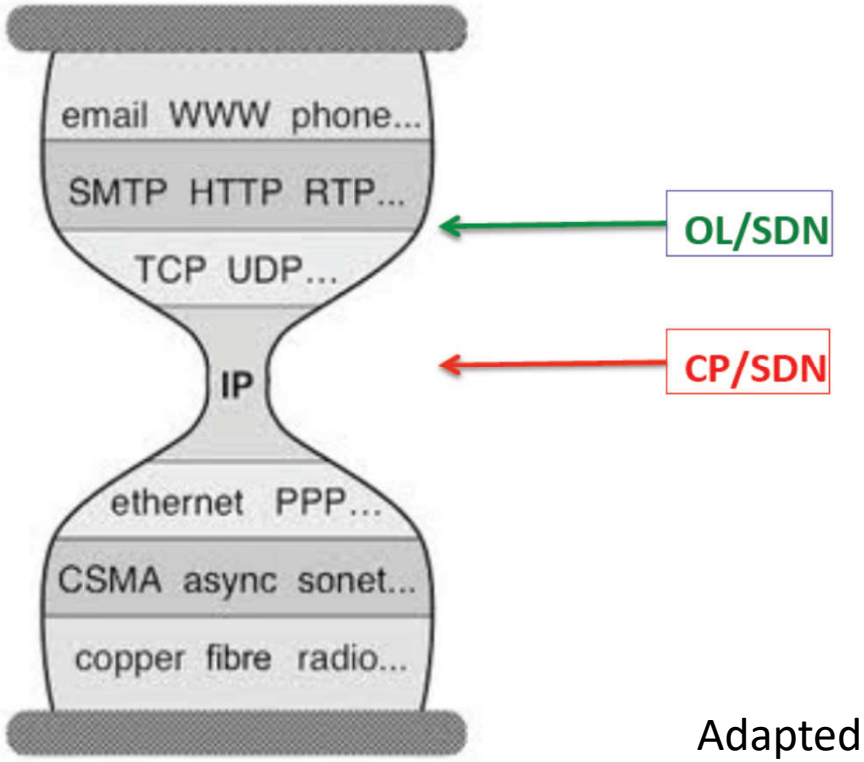
Bowties /Hourglasses?



Bow tie (biology)



Open Source is a wildcard



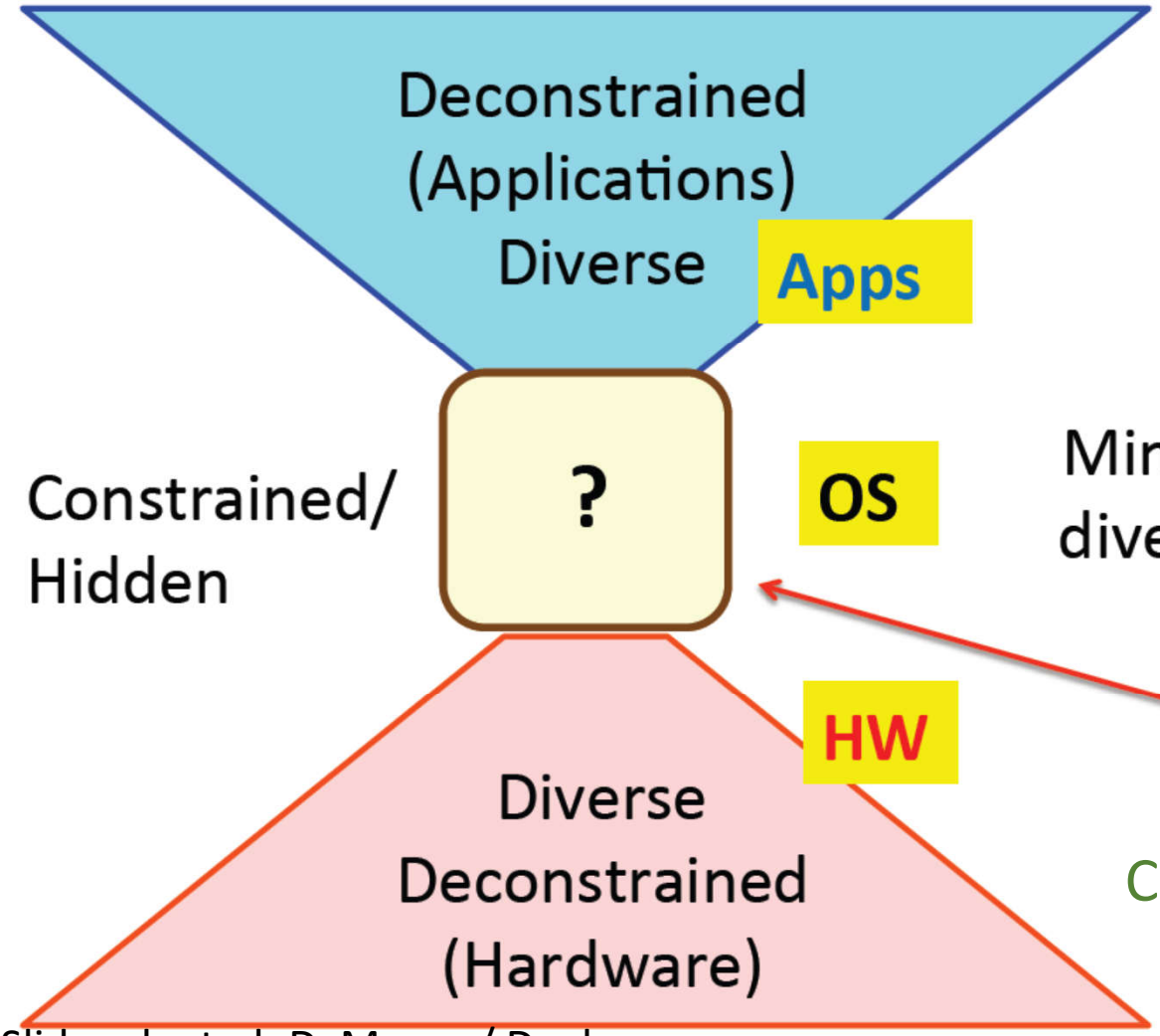
OL/SDN

CP/SDN

OF/SDN?

Adapted Slide courtesy: D. Meyer

Layered architectures make Robustness and Evolvability *compatible*



Minimal diversity

Think OS Kernel + Libraries and Drivers

SDN / NFV Orchestrator?

SDN Controller?

Plug-in Architectures

Control Plane Drivers + APIs

Model-Driven Everything

Slide adapted: D. Meyer / Doyle

Some experiences and ongoing activities

About: Christian Esteve Rothenberg

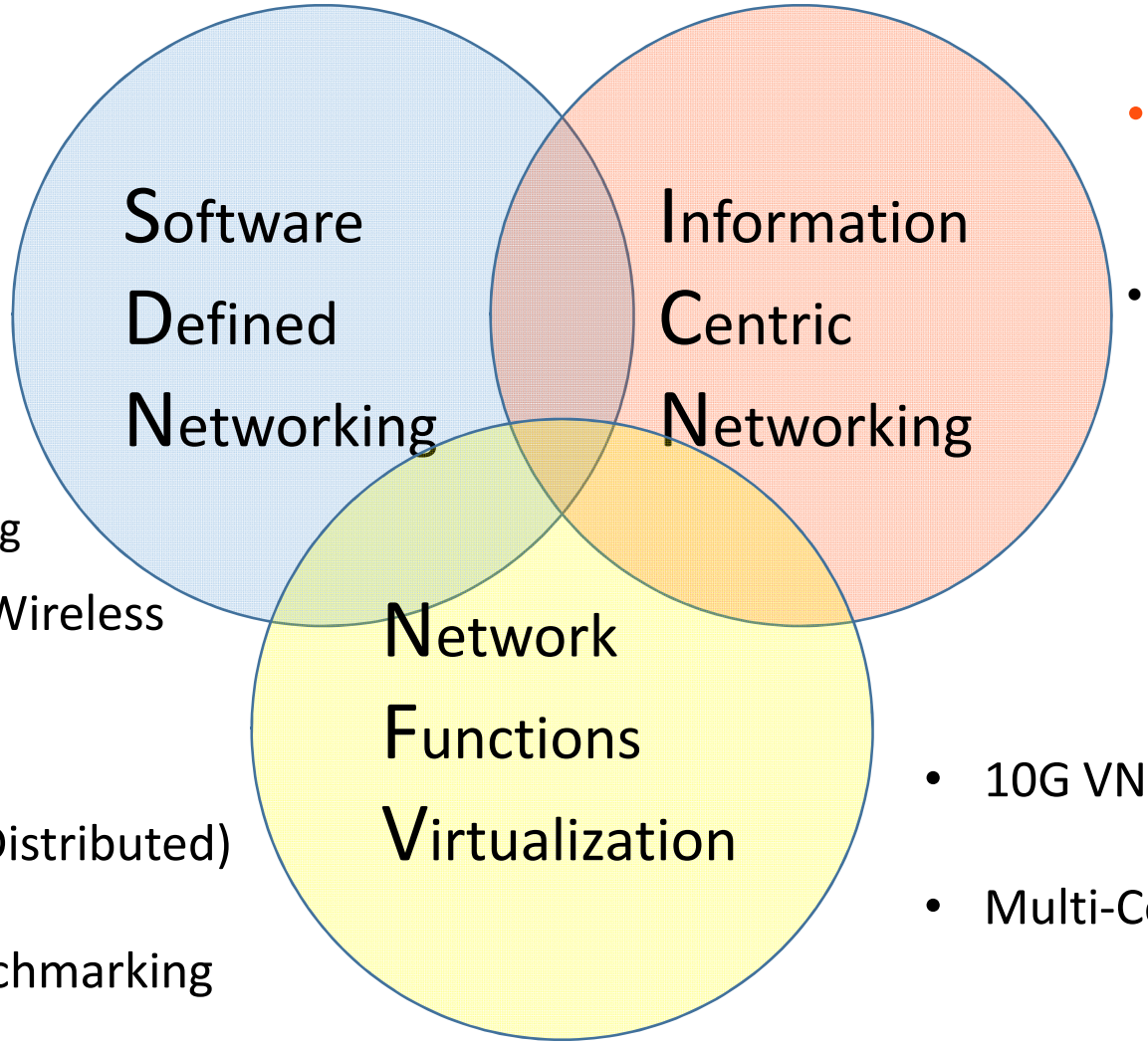
- Assistant Professor (tenure track) at FEEC/UNICAMP (since 2013)
  - Leading the INTRIG lab at DCA/FEEC/UNICAMP
    INTRIG: Information & Networking Technologies Research & Innovation Group
  - Currently, supervising 7 PhD, 5 MSc candidates, and 4 undergrad students
- PhD in Electrical and Computer Engineering (FEEC/UNICAMP, 2010),
  MSc in Electrical Eng and Information Technology (Darmstadt University, 2006),
  Telecommunication Eng (Universidad Politécnica de Madrid, 2004)
  - Visiting researcher at Ericsson Research Nomadic Lab, Jorvas, Finland, 2008, participated in EU Publish/Subscribe Internet Routing Paradigm (PSIRP).
- Research Scientist at CPqD R&D Center in Telecommunication (2010-2013)
  - Technical Lead of SDN activities in the Converged Networking Division
- ONF Research Associate (since Apr/2013)

FEEC
Faculdade de Engenharia Elétrica e de Computação
UNICAMP

INTRIG
INFORMATION & NETWORKING TECHNOLOGIES RESEARCH & INNOVATION GROUP

Research Interests and Main Goals & Results

- RouteFlow
  (hybrid IP-SDN)
- softswitch13
- libfluid
  (ONF Driver)
- SDN-2-SDN Peering
- Mininet-WiFi :: Wireless / Mobility
- MD2-NFV
  (Multi-Domain Distributed)
- VNF-a-a-S / Benchmarking



- Mini-CCNx -> Mini-NDNx
- LIPSIN
  (in-packet Bloom filters)

- 10G VNF
- Multi-Core Architectures

Open Source minded Research @ INTRIG

- Extensive use of open source in support of research activities.
  - Etherpad, Owncloud, Docker, Gitlab, etc.
- Publish versions of papers (in submission/accepted)
  - Arxiv.org
- Make “source code” of papers available
  - Overleaf, github
- Release all research data to allow third parties to re-use and reproduce
  - Github, wiki + readme (instructions on how to reproduce the paper experiments)
  - Highlight the reproducibility aspect in the papers (today: plus, tomorrow: requirement/norm?)
- Students use github (private->public) as the research dev repository
  - Even if no paper accepted or in early
- Create community-oriented open source projects

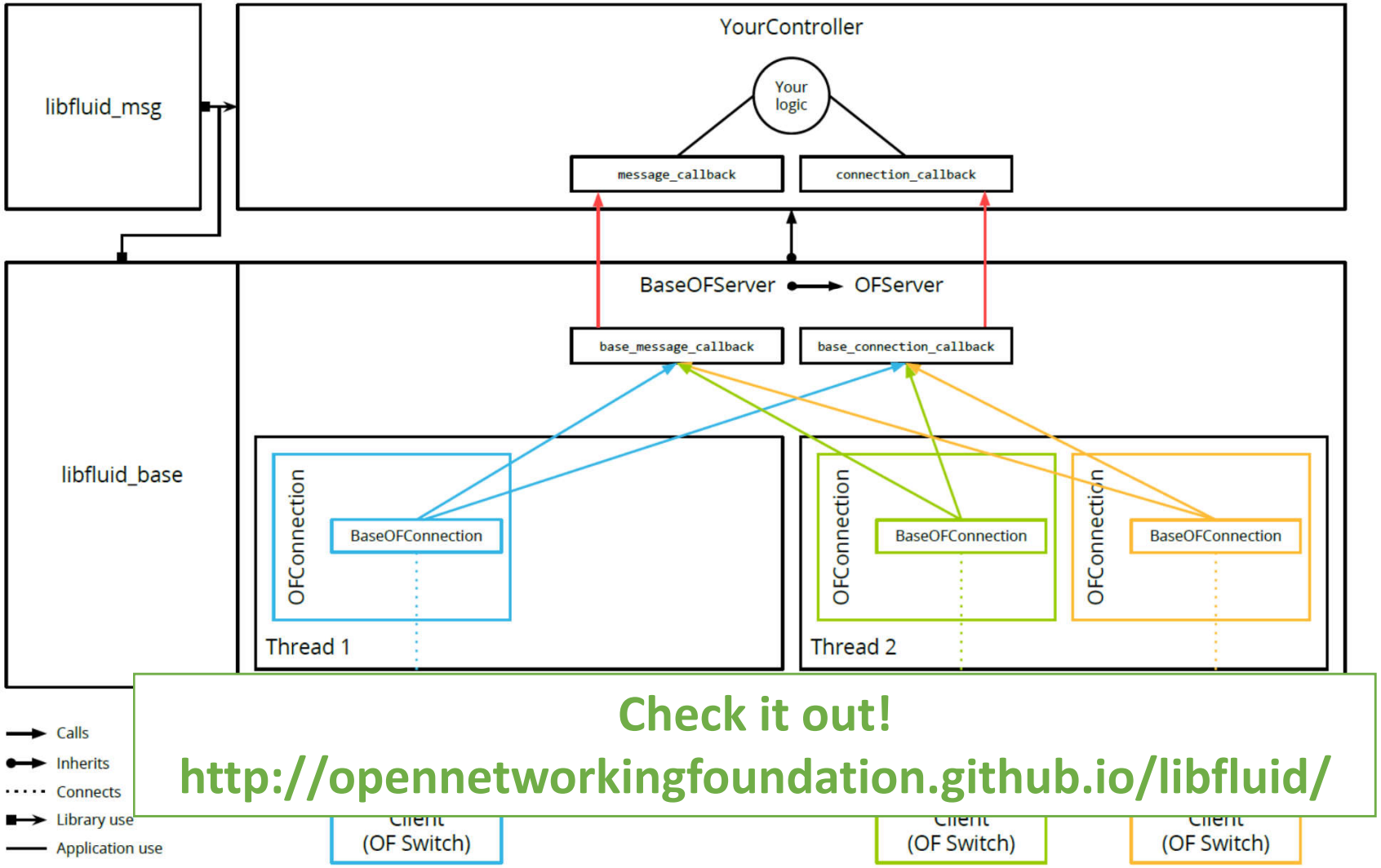
Research Projects and Open Source Results

Technical lead of successful open source projects:

- 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 framework [funded and technical collaboration with Ericsson] (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/

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

libfluid ONF Driver Implementation



Check it out!

http://opennetworkingfoundation.github.io/libfluid/

*Joint work with E. Fernandes (CPqD), A. Vidal (CPqD), M. Salvador (Lenovo), F. Verdi (Ufscar)

RouteFlow: Introduction

RouteFlow

FUNTTEL

Ministério das Comunicações

GOVERNO FEDERAL BRASIL PAÍS RICO É PAÍS SEM POBREZA

FINEP AGÊNCIA BRASILEIRA DA INOVAÇÃO

Background

Glue of IP routing stacks with OpenFlow

Controller-centric hybrid IP networks

Migration path to SDN

Architecture

Modular (3 components)

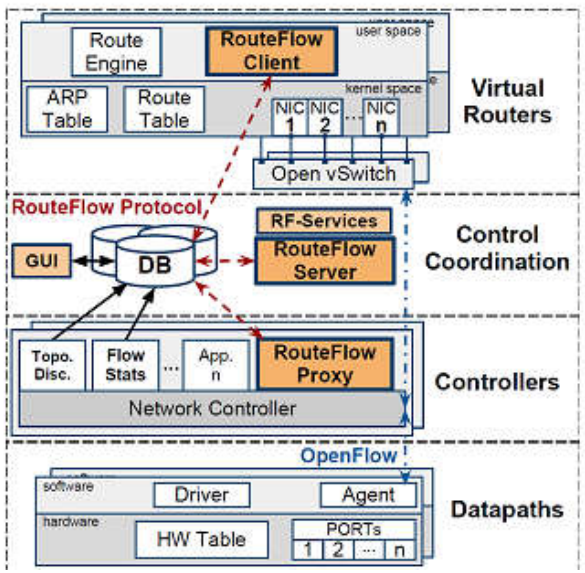
Hierarchical, distributed

Multi-controller support

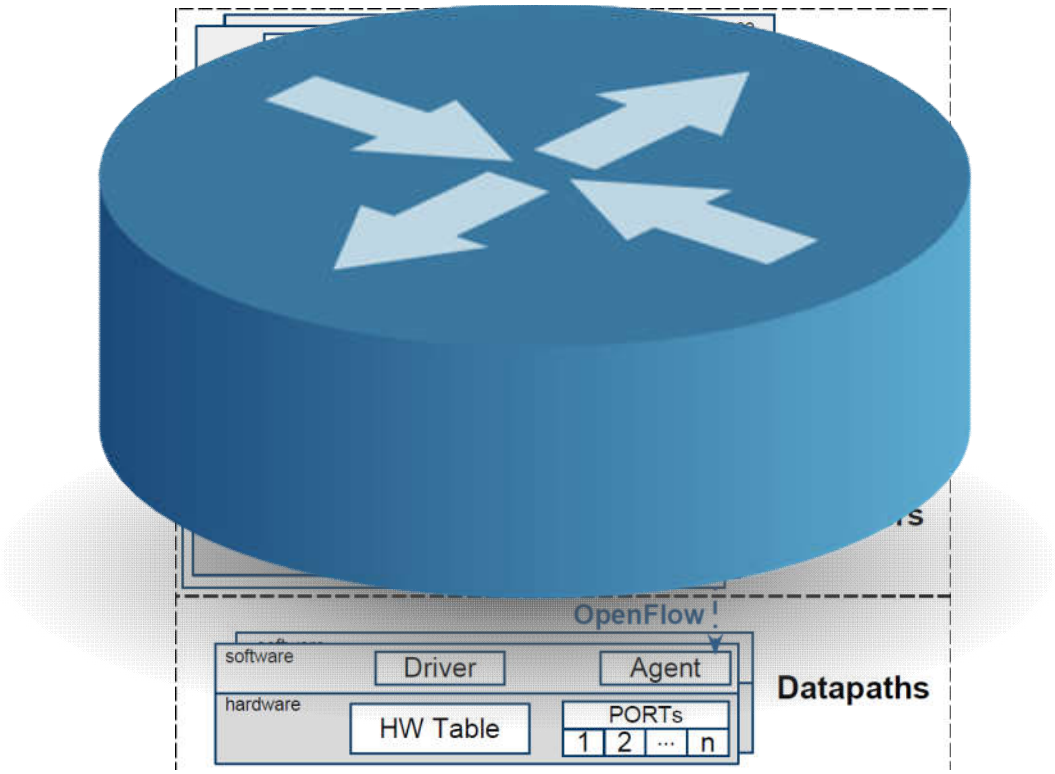
(POX, NOX, Floodlight, Ryu)

Any Linux-based routing stack

(Quagga, XORP, BIRD)



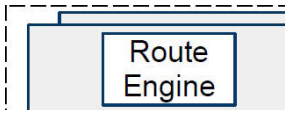
RouteFlow: Basics



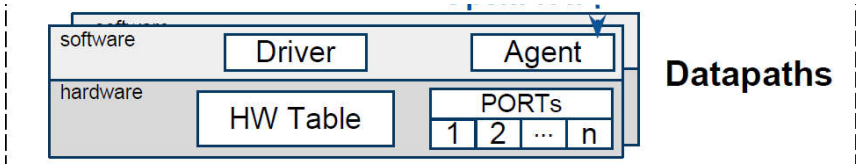
Control Plane

Data Plane

RouteFlow: High-level Architecture



Linux



Datapaths

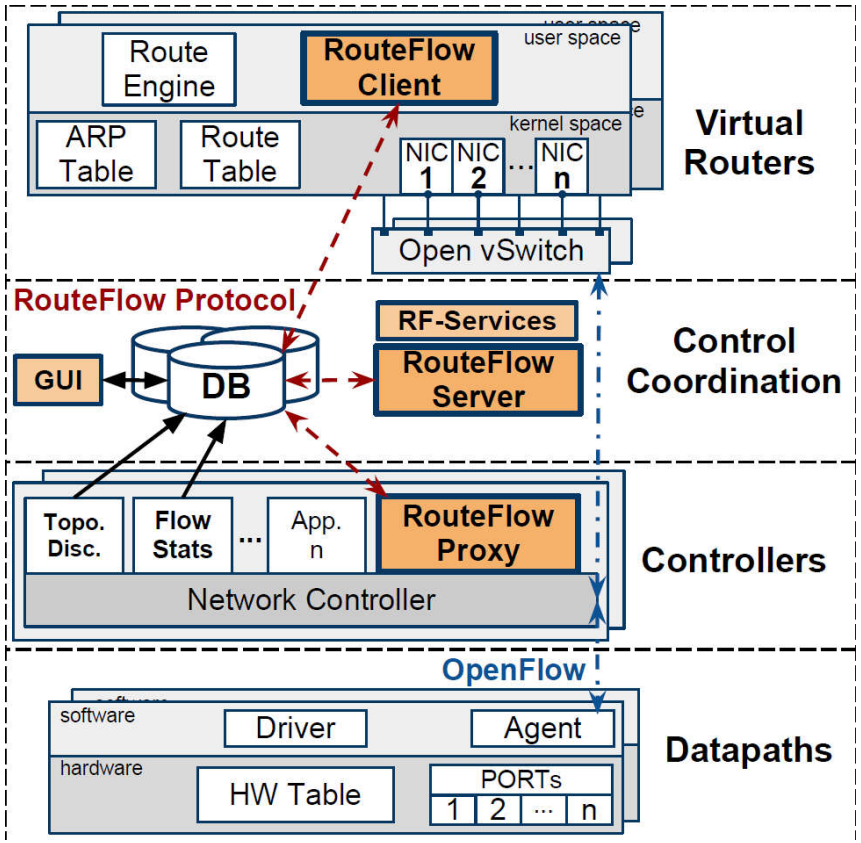
Control Plane

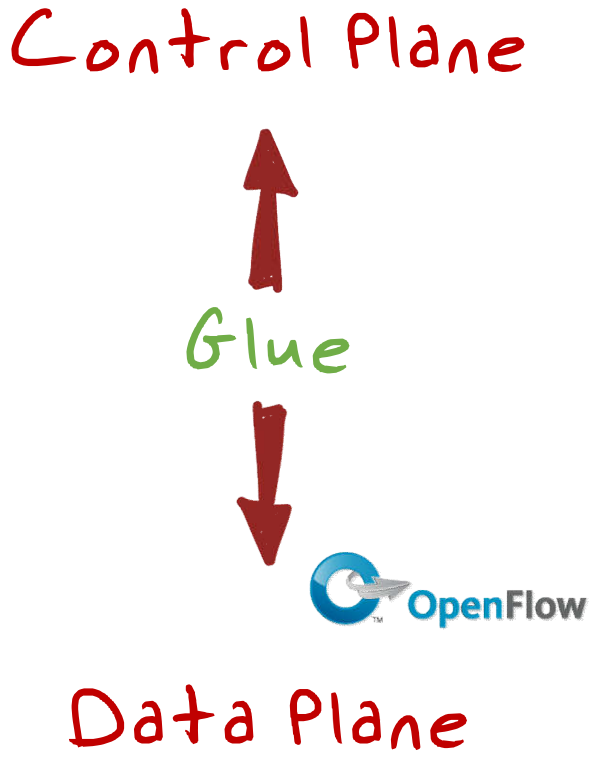
Glue

OpenFlow

Data Plane

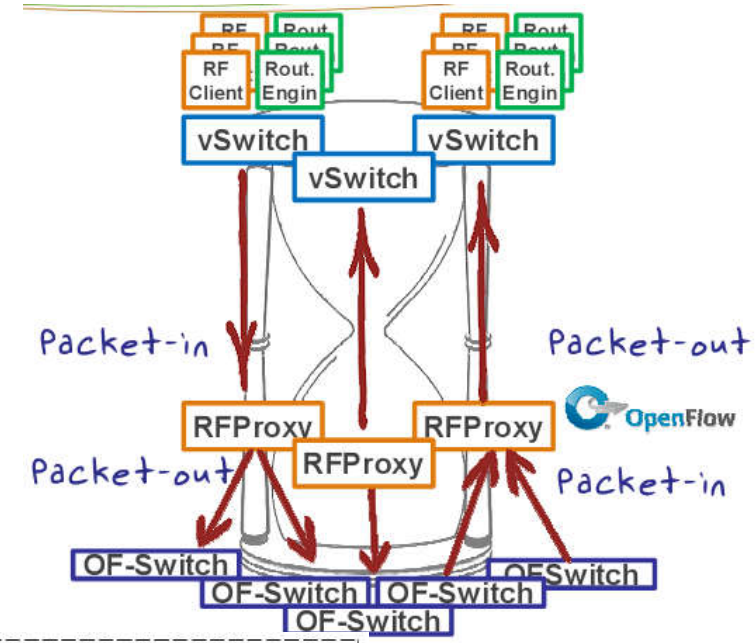
RouteFlow: High-level Architecture



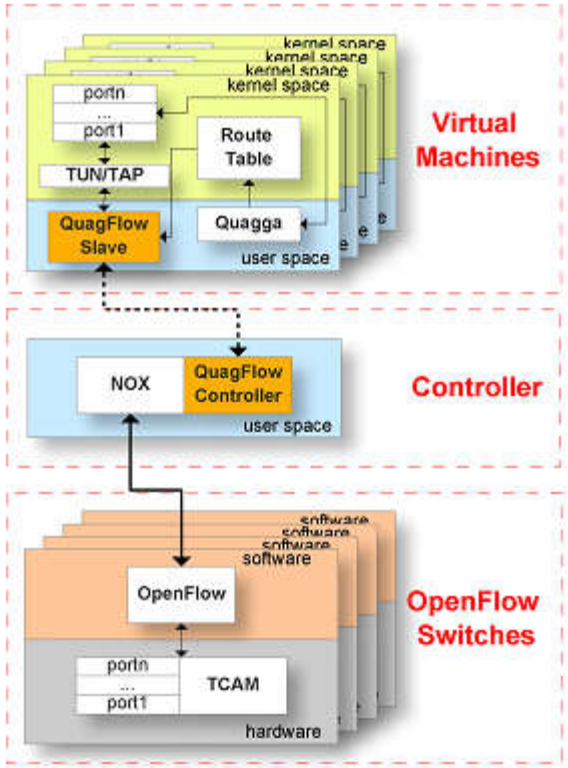


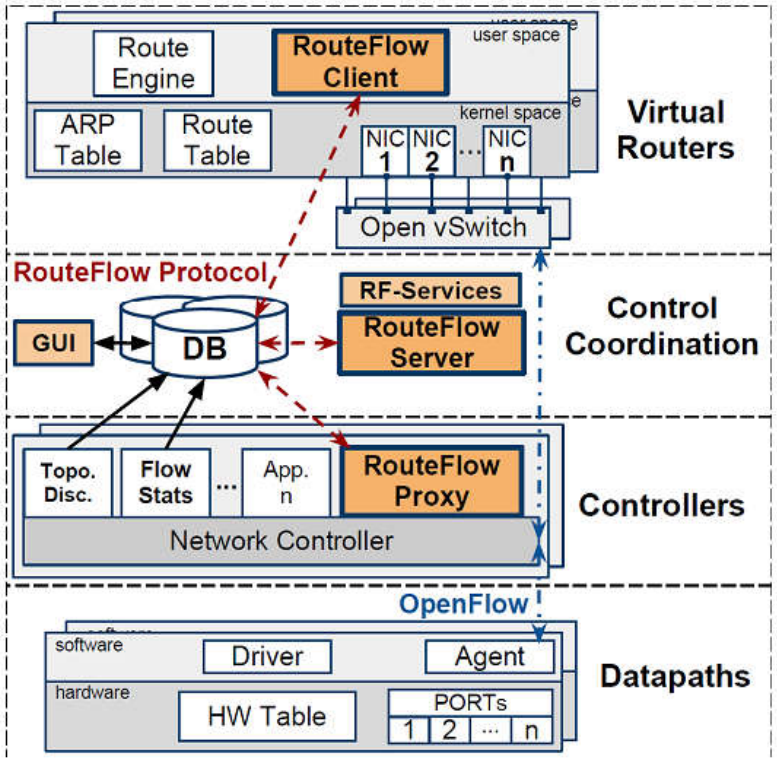
RouteFlow architecture

- Evolution since QuagFlow PoC



Dev(Ops)





Open Source SW | RouteFlow SDN/OpenFlow architecture

Quagga, XORP, BIRD

Linux

Open vSwitch, softswitch13

LXC, QEMU, XEN, libvirt

MongoDB, redis

jQuery, JIT

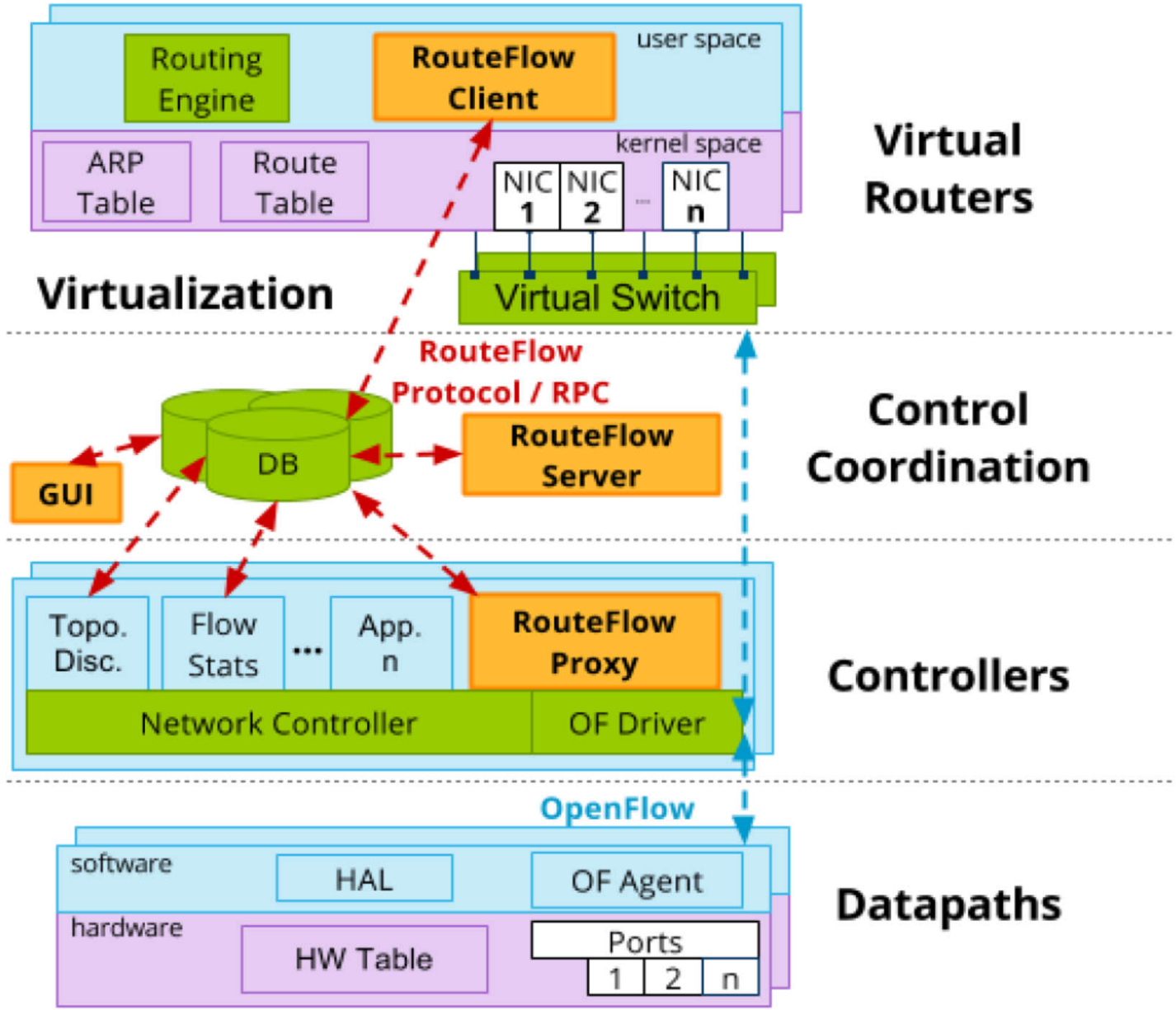
NOX, POX, Floodlight, Ryu

ONF driver

FlowVisor

Mininet, OVS,

softswitch13, Pica8 OS



RouteFlow Project History

- Jan / 2010
- Aug / 2010
- Dez / 2010
- May / 2011
- Oct / 2011
- Nov / 2011
- Apr / 2012
- Aug / 2012

- Start Msc. Thesis work by Marcelo N.
- First Prototype
- First Short-Paper @ WPEIF
- Evaluation on NetFPGA testbed
- QuagFlow Poster @ SIGCOMM
- Open-Source Release
- Demos @ ONS11
- Tutorial & Demo @ OFELIA/CHANGE SS
- Indiana University - Pronto OF switches + BGP peering with Juniper MX
- Demo @ SuperComputing 11
- Demos @ ONS12
- HotSDN Paper
- Running on FIBRE / OFELIA testbed
- Collaboraion with NTT

Open Innovation

Visitors: 40,000+ (20,000+ unique)

From 3,000+ cities from 130+ countries!

Warning!
Outdated
[2014]

http://go.cpqd.com.br/routeflow/

RouteFlow

900

days since
project start

1000s downloads!

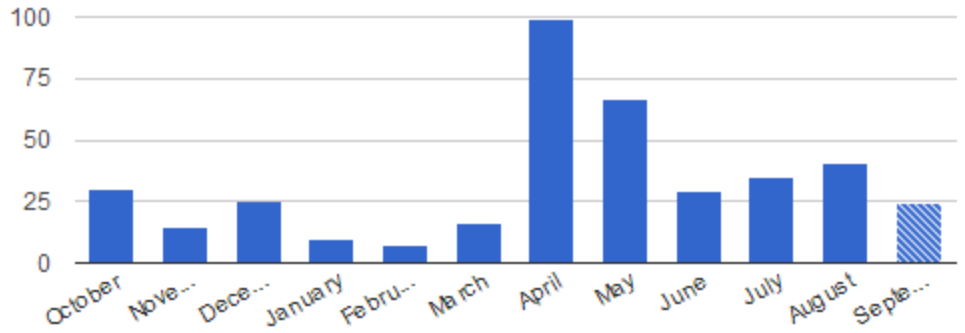
github SOCIAL CODING

Apache

Community development

Warning! Outdated [2014]

- Mailing List:
  - 200 Members
  - 160 Topics

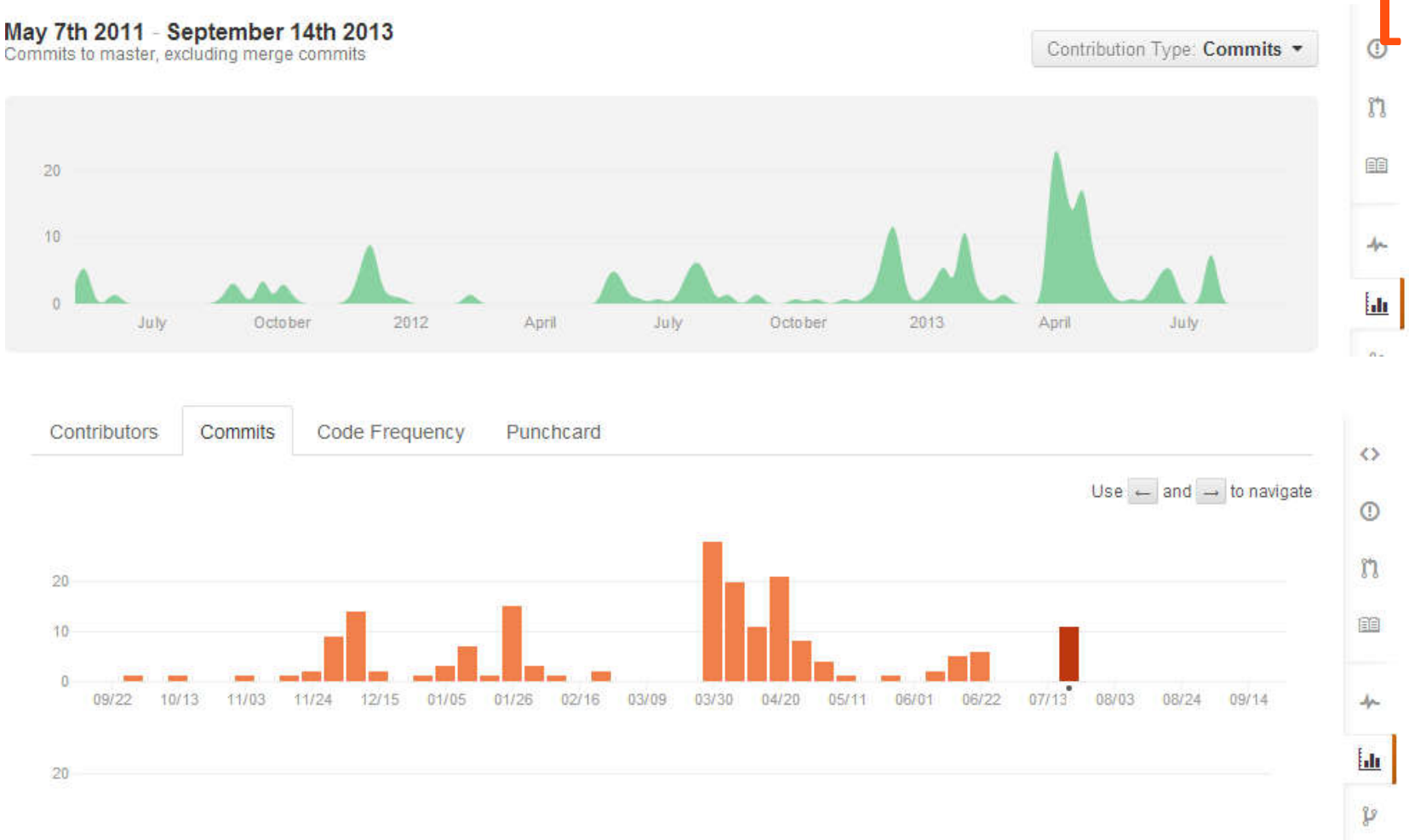


- Code contributions:

  (5K – 10K LOC, many testing hours, bug reports, etc)

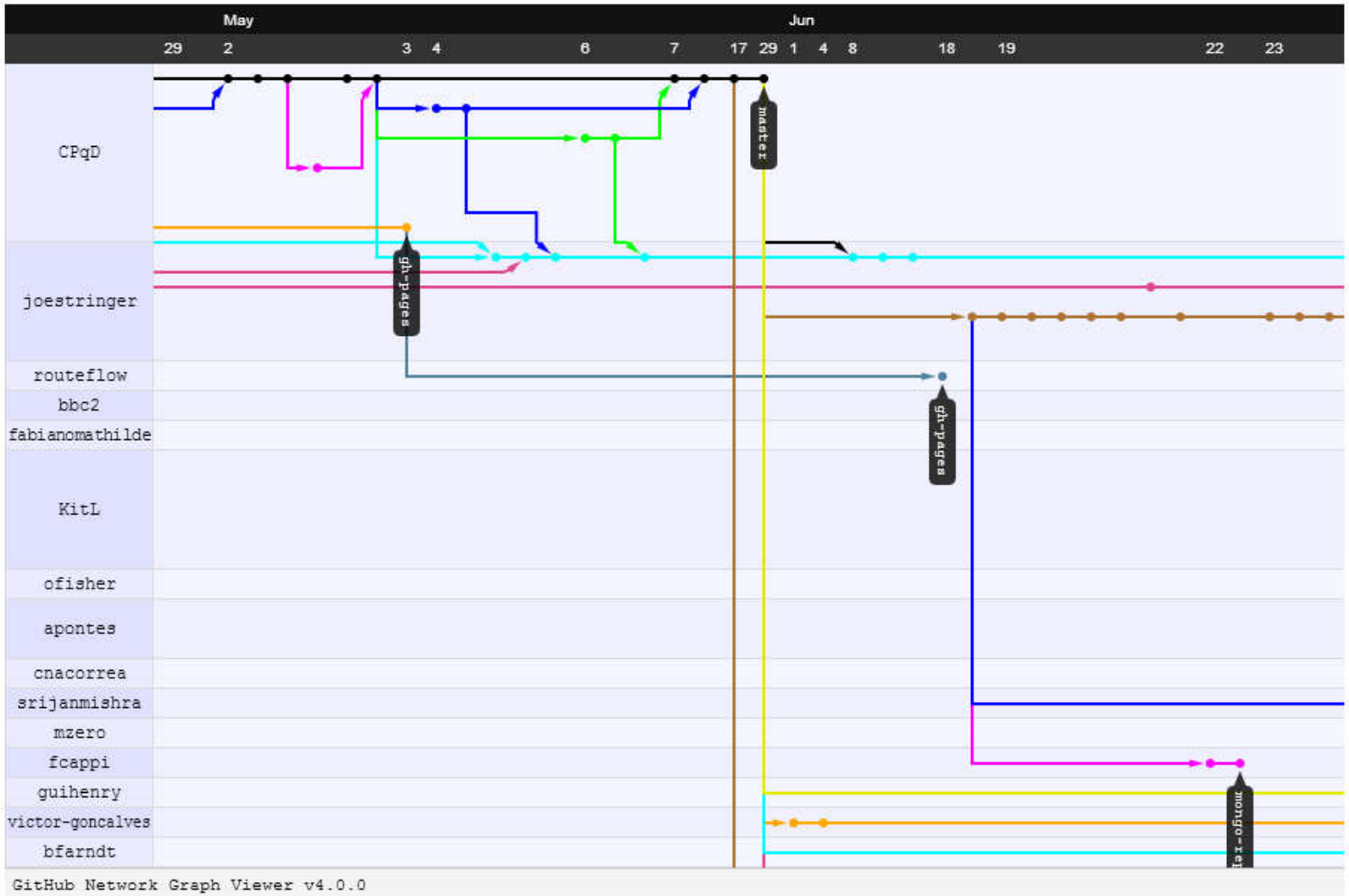
Github Activity

Warning!
Outdated
[2014]



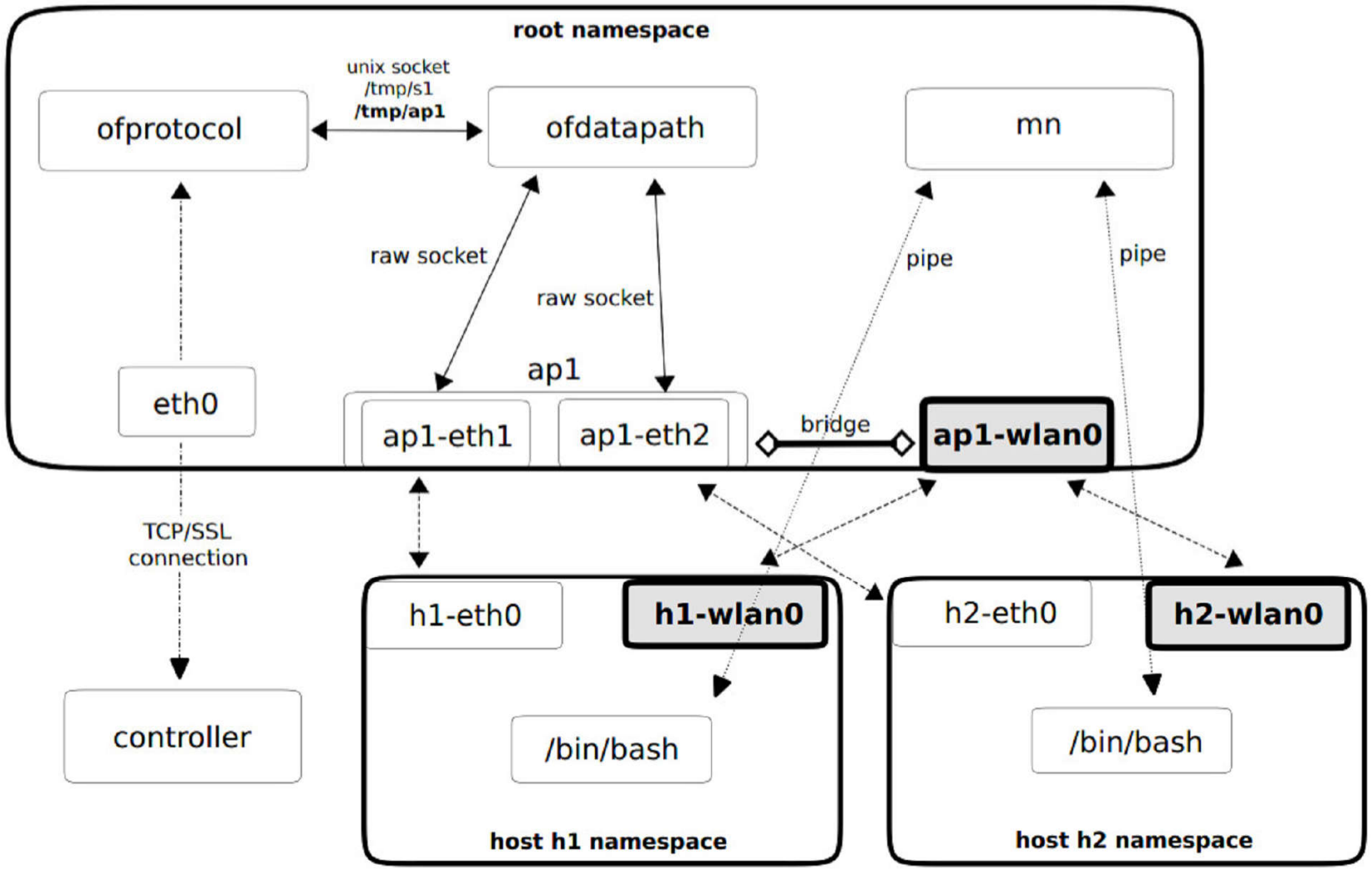
Github Activity

Warning!
Outdated
[2014]



Mininet-WiFi

Emulator for Software-Defined Wireless Networks



More info: Mininet-WiFi: Emulating Software-Defined Wireless Networks

https://github.com/intrig-unicamp/mininet-wifi

Mininet-WiFi :: Use Cases

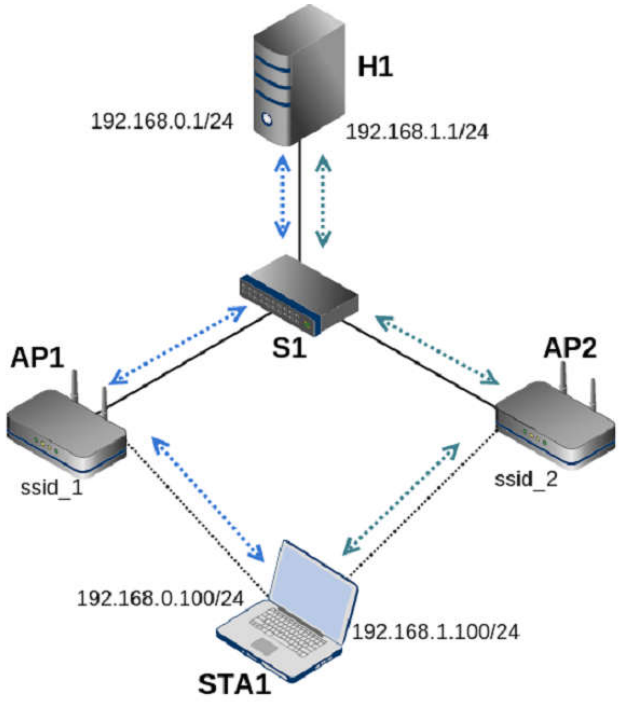


Figure: Topology.

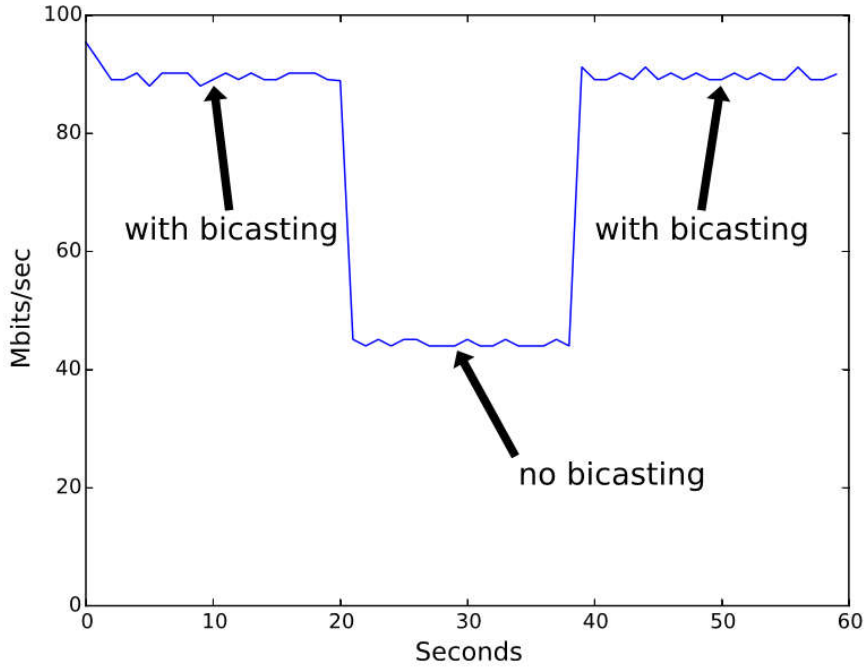
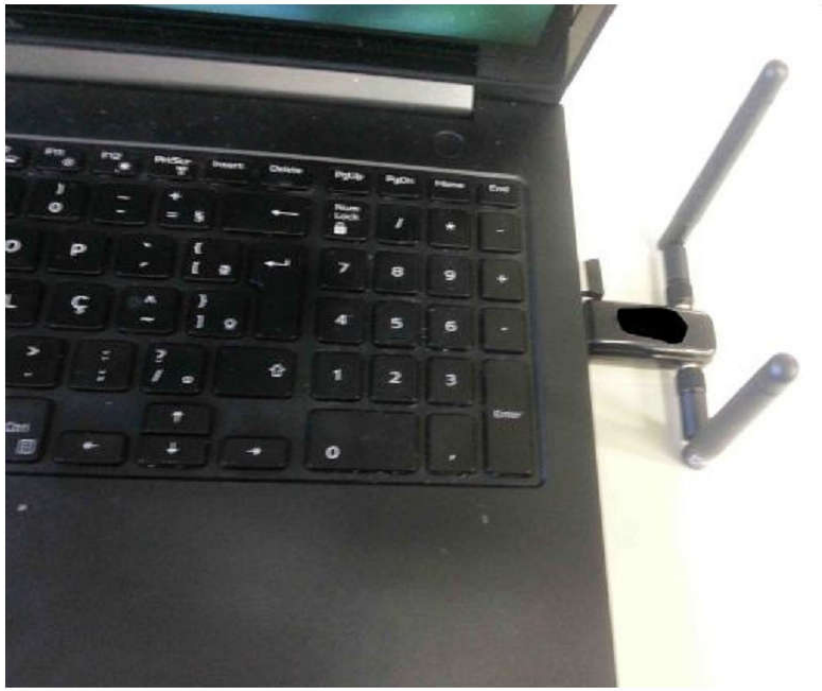


Figure: Bicasting.

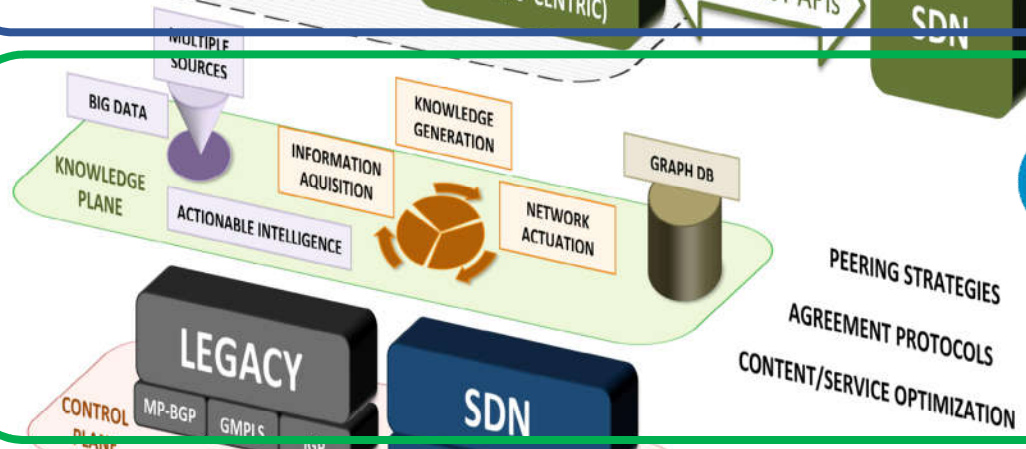


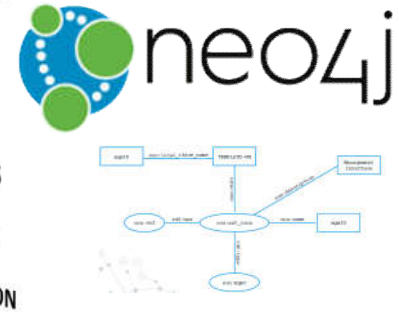
Physical wireless NIC integrated into Mininet-WiFi.

More info: **Demos**
Video 01: https://www.youtube.com/watch?v=_PtSmhf7Z8s
Video 02: https://www.youtube.com/watch?v=H46EPuJDJhc
Video 03: https://www.youtube.com/watch?v=WH6bSOKC7Lk

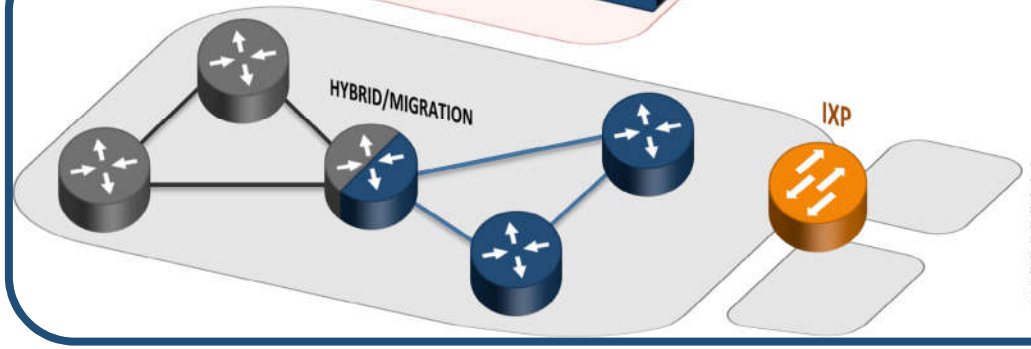
Advanced Peering with a Software-Defined Knowledge Plane

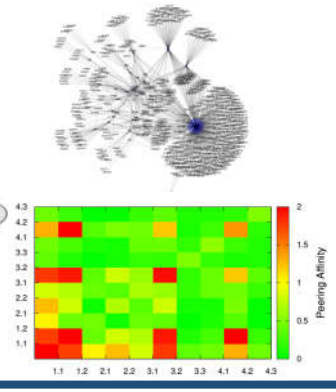
- ALTO as East/West "Peering" APIs between SDN domains
  - Exchange Topology and Cost Maps
  - Optimize Placement & Routing towards Services/Content





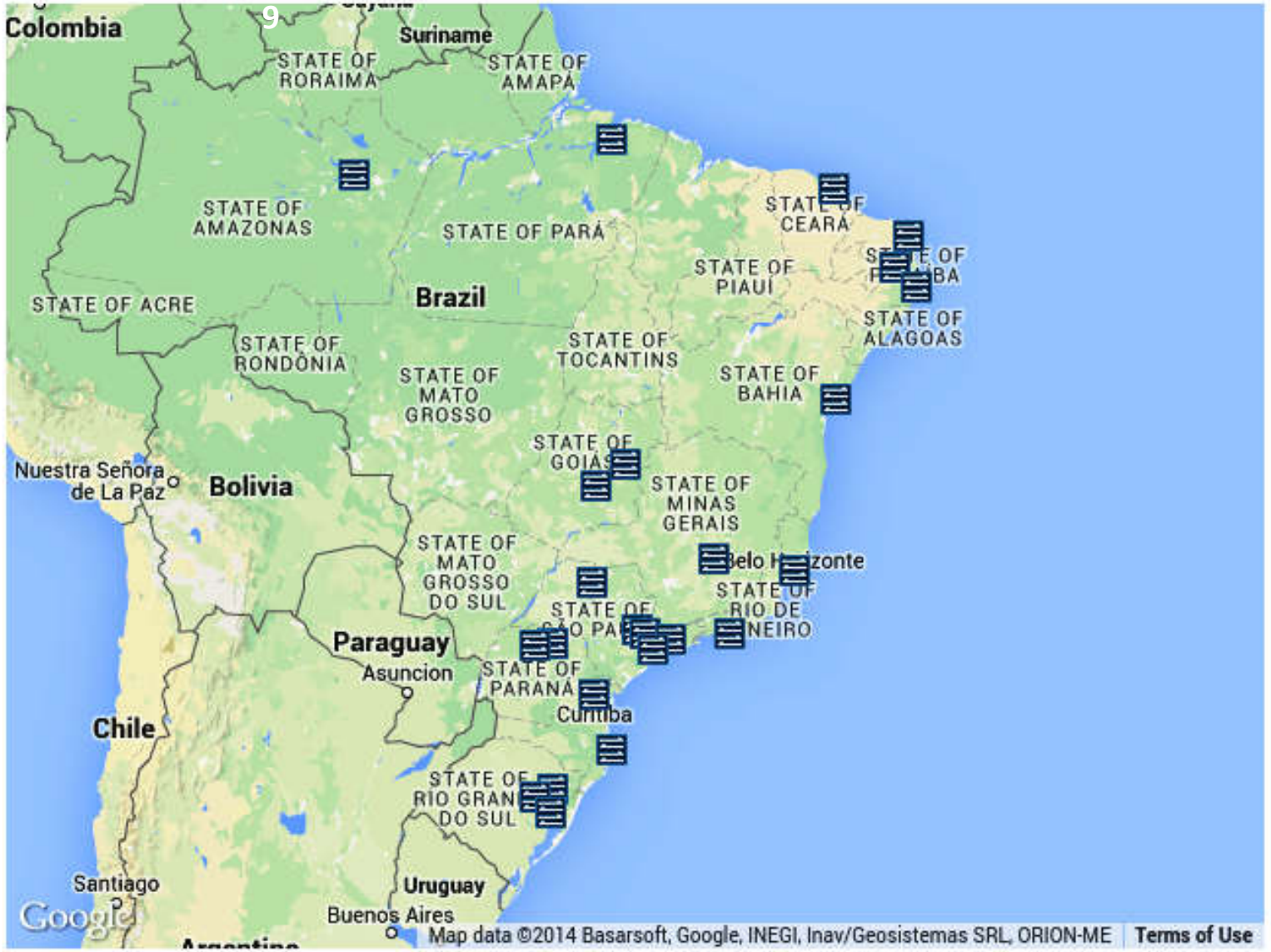
- Semantic Models / APIs based on NML (Network Markup Language)
- Knowledge generation based on Machine Learning & Big Data





- Anatomy of current IXP Ecosystem
- Extracting requirements for SDN peering and realistic topologies for inter-domain SDN

PTTMetro in Brazil (PTT.br)



- Americana
- Belém
- Belo Horizonte
- Brasília
- Campina Grande
- Campinas
- Cuiabá
- Caxias do Sul
- Curitiba
- Florianópolis
- Fortaleza
- Goiânia
- Lajeado
- Londrina
- Manaus
- Maringa
- Natal
- Porto Alegre
- Recife
- Rio de Janeiro
- Salvador
- Paulista Central (São Carlos)
- São José dos Campos
- São José do Rio Preto
- São Paulo
- Vitória

Location of Each Brazilian IXP (figure extracted from http://www.ptt.br)

Methodology: AS-level Graphs

Data Sources

- PTT.br Official Data at www.ptt.br
- PeeringDB (we found it was unreliable)
- Telnet Access to IXP's Looking Glasses (BGP Table, Paths Summary, Communities List)

Graph analysis on BGP adjacency matrix of all IXPs using NetworkX & Neo4j)

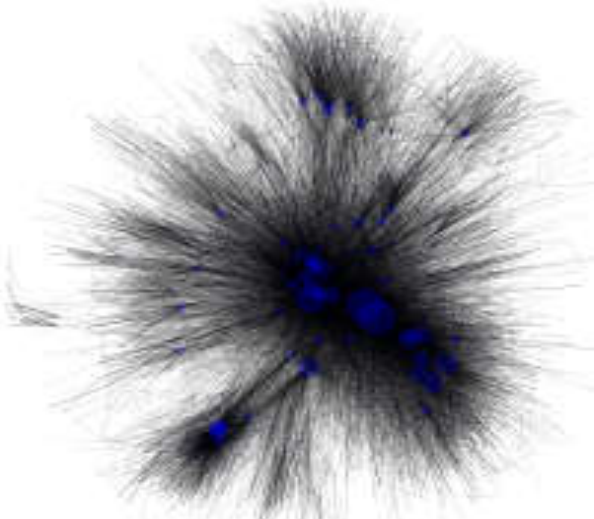
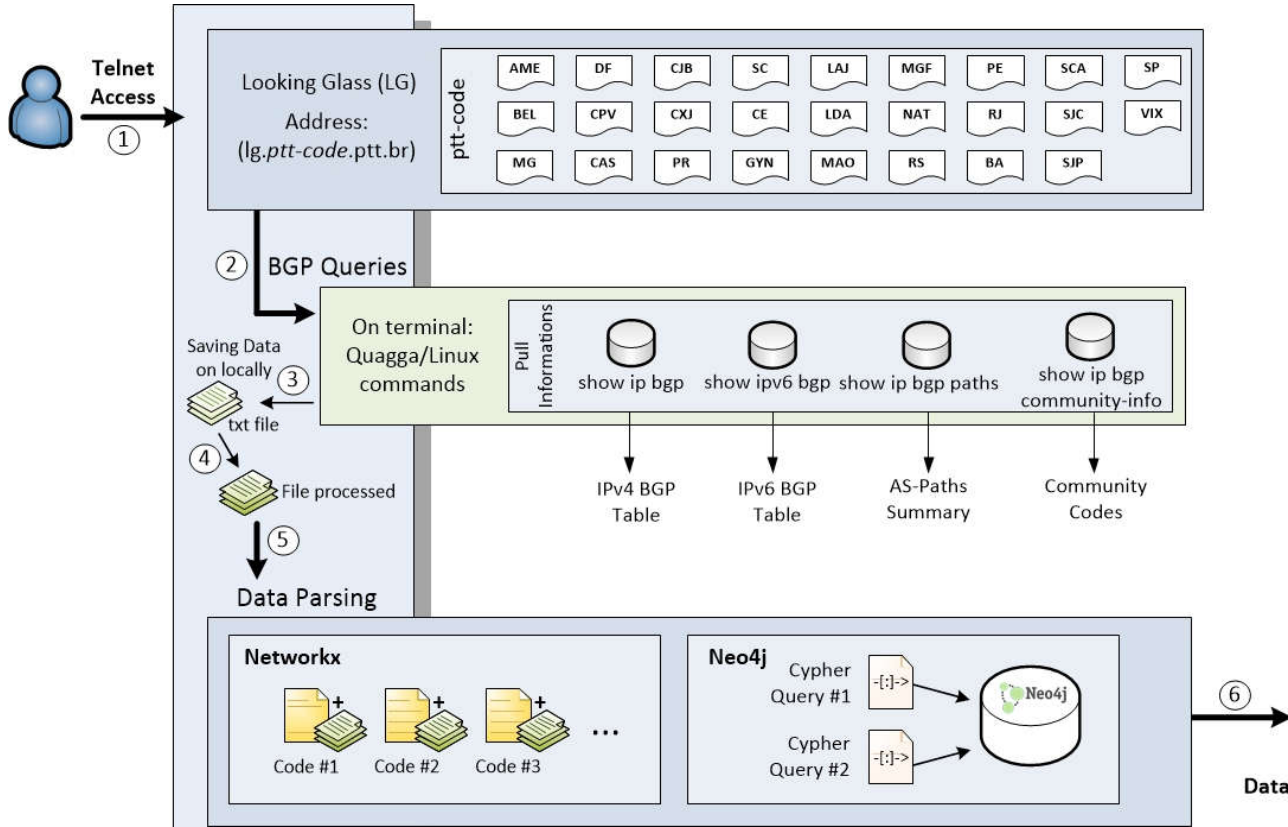
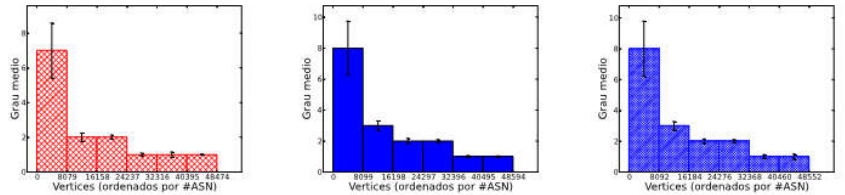


Figure 1b. Example Graph of PTT-VIX (Vitória, ES)



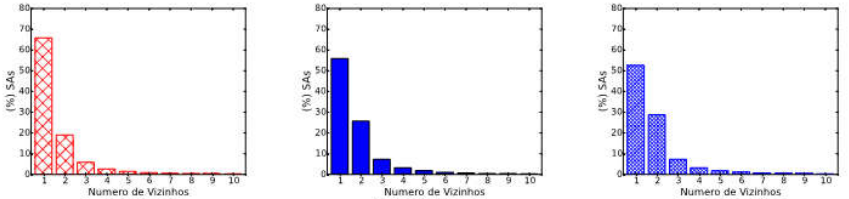
Results (see details in SBRC´15 & on-demand)

- Profile and Classification of IXP's Members
  - Who is currently where? Who should be there?
- AS Vertices's Degree / Depth / Diameter
  - How many? How far?
- Density of Peering
  - How many unexpoited peering relationships?
    –> Peering Recommender system
    –> SDN-enabled peering opportunities
- AS-Prepend for TE
  - How much indirect TE through BGP hacks?
    –> SDN-enabled QoS between peers
- k-Clique Communities & Machine Learning (ML)
  - Organization and Classification of Customer/Client/Peer
  - –> Peering Recommender system?



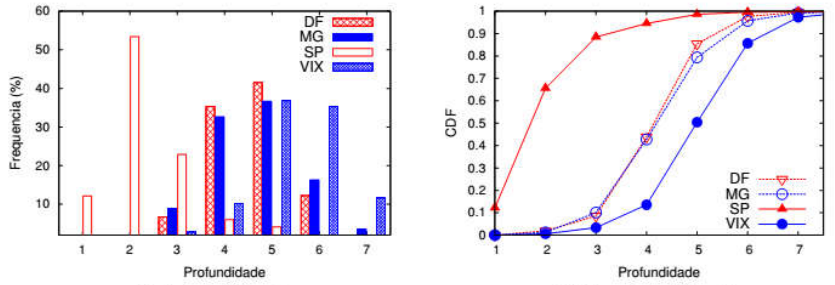
(a) PTT-DF (b) PTT-MG (c) PTT-VIX

Figure 3: Average Degree of Graphs by ASN



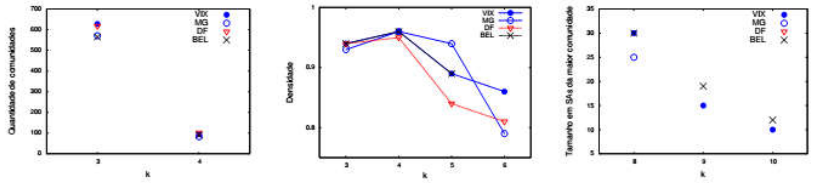
(a) PTT-DF (b) PTT-MG (c) PTT-VIX

Figure 4: Graphs's Degree Distribution



(a) Isolated Depth (b) Cumulated Depth

Figure 5: Depth of AS-PATH



(a) k3 and k4 Communities (b) Density (c) Higher Communities

Figure 6: k-Clique Communities

Research challenges (1/2)

Research and development around Software Defined Networking (SDN) and Network Function Virtualization (NFV) is vast and testbeds and related toolkits in academia addressing SDN, NFV, Mobile Edge Computing (MEC) and 5G technologies are being set-up

- A strong impact on the industry is anticipated

Many open source initiatives with high potentials are not known by other researchers due to the lack of visibility

- Scientific publications are a good mean for those researchers to get some visibility, but that’s not enough

Research challenges (2/2)

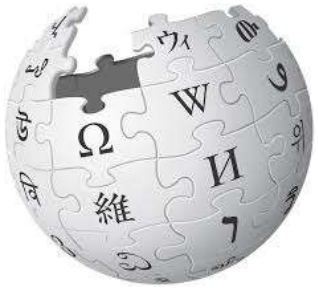
Many researchers, students, product developers have no idea how to get started

- The large number of existing activities makes very difficult the selection process of the tools needed for supporting their requirements

How to get started???

Image Source:
https://www.linkedin.com/pulse/how-technology-has-become-blessing-curse-paul-hollington

Towards a “Wikipedia” of Toolkits & Testbeds



NEW: Please see and contribute to

- http://bit.do/oss-sdn-nfv

Open Source
Toolkits & Testbeds

Name	Organization	Main Contribution / Focus [SHORT DESC: 160 char.]	Link-Project	Link-Repo-Code	OpenSource-License	Mailing-list	Link-to-Mailing-List (Subscribe)	Link-to-Mailing-List (Archives)	Status	Link-to-Scientific-Page	Remarks	Tag 1	Tag 2	Tag 3	Tag 4
OpenvSwitch (OVS)	Linux Foundation	Production quality, multilayer virtual switch designed to enable massive network automation through programmatic extension, while still supporting standard management interfaces and protocols.	http://openvswitch.org/	https://github.com/openvswitch/ovs	Apache 2.0	announce@openvswitch.org	http://mail.openvswitch.org/mailma	http://openvswitch.org/pipermail/	Active			SW Dataplane / Switch	vSwitch	Cloud	Network Programmability
OpenSwitch (OSP)	Linux Foundation	OpenSwitch is a network operating system for disaggregated switches that are built around OCP compliant hardware and that utilizes the ONIE boot loader to install and uninstall network operating systems	http://www.openswitch.net/	https://archive.openswitch.net/artifacts/	Apache 2.0	ops-dev@lists.openswitch.net	http://lists.openswitch.net/cgi-bin/m	http://lists.openswitch.net/piperm	Active		good project governance	SW Dataplane / Switch	HW Dataplane / Switch		Network Programmability
Indigo	Big Switch Networks	The Indigo agent includes core libraries aimed at enabling support for OpenFlow on physical and hypervisor switches.	http://www.projectfloodlight.org/indigo/	https://github.com/floodlight/indigo	Eclipse Public License version 1	indigo-announce@openflowhub.	http://groups.google.com/a/openflo	https://groups.google.com/a/ope	Active			SW Dataplane / Switch	HW Dataplane / Switch	SDK, APIs, Libraries	SDN Dataplane Agent
Oflsoftswitch13	CPqD	Research-friendly User-space OpenFlow 1.x software switch forked from Stanford's original reference switch design used for prototyping and experimentation. Integrated into Mininet. Used for open source implementation of new OpenFlow features by ONF members.	http://cpqd.github.io/ofsoftswitch13/	https://github.com/CPqD/ofsoftswitch1	BSD license	openflow-discuss@openflowswit	https://mailman.stanford.edu/mailm	https://mailman.stanford.edu/mai	Active			SW Dataplane / Switch			
LINC-switch	FlowForwarding	LINC is a pure OpenFlow software switch written in Erlang	http://flowforwarding.github.io/LINC-S	https://github.com/FlowForwarding/LIN	Apache 2.0	linc-dev@flowforwarding.org	http://groups.google.com/a/flowfor	https://groups.google.com/a/flow	Inactive			SW Dataplane / Switch			
Protocol Oblivious Forwarding (POF)	Huawei	SDN southbound protocol designed for high flexibility.	http://www.poforwarding.org	https://www.poforwarding.org/pofstd	BSD license	pofforwarding@Huawei.com			Active			SW Dataplane / Switch	HW Dataplane / Switch	SDK, APIs, Libraries	DSL - Domain Specific Langu
Lagopus	NTT	High-performance software OpenFlow 1.3 switch leveraging DPDK.	http://lagopus.github.io/	https://github.com/lagopus/lagopus	Apache 2.0	lagopus-devel@lists.sourceforge	http://lists.sourceforge.net/lists/list	http://sourceforge.net/mailarchiv	Active	http://www.usenix.org/event/atc14/te		SW Dataplane / Switch			
Berkeley Extensible Software Switch (BES	Berkeley University	Modular framework for high-performance software switches allowing to configure custom packet processing datapath by composing small "modules".	http://github.com/NetSys/bess/wiki	https://github.com/NetSys/bess	BSDv3				Active	http://www2.eecs.berkeley.edu/Pubs/TechRpts/2		SW Dataplane / Switch	SDK, APIs, Libraries	DSL - Domain Specific Language	
ClickOS	NEC	A minimalistic, tailor-made, virtualized operating system to run Click-based middleboxes.	http://cnp.neclab.eu/clickos/	https://github.com/cnplab/clickos	BSDv3	cnplab@listserv.neclab.nec.de.	http://listserv.neclab.nec.de/mailm	http://listserv.neclab.nec.de/mail	Active	http://www.usenix.org/system/files/conference/ns		SW Dataplane / Switch			
Click	MIT, UCLA, and others	The Click modular router: fast modular packet processing and analysis	http://www.read.cs.ucla.edu/click/	https://github.com/kohler/click	MIT	click@librelist.com	http://librelist.com/browser/click	http://librelist.com/browser/click	Active	http://pdos.csail.mit.edu/papers/click:tocs00/paper		SW Dataplane / Switch	SDK, APIs, Libraries	DSL - Domain Specific Language	
Snabb Switch	Snabb	Snabb (formerly "Snabb Switch") is a simple and fast packet networking toolkit.	http://snabb.co/	https://github.com/snabbco/snabb	Apache 2.0	snabb-devel@googlegroups.com	https://groups.google.com/forum/#!	https://groups.google.com/forum/	Active			SW Dataplane / Switch	SDK, APIs, Libraries		
OpenNetVM	GW, UCR	High-performance NFV platform for running service chains through Docker NFs	http://sdnfv.github.io/onvm/	https://github.com/sdnfv/openNetVM	BSD	timwood@gwu.edu			Active	http://faculty.cs.gwu.edu/~timwood/papers/16-Hot		SW Dataplane / Switch	DPDK		
Open Network Install Environment (ONIE)	Open Compute Project	Open Compute Project open source initiative contributed by Cumulus Networks that defines an open "install environment" for bare metal network switches	http://onie.opencompute.org/	https://github.com/opencomputeproject	GNU GPL v2	opencompute-onie@lists.openco	http://lists.opencompute.org/mailm	http://lists.opencompute.org/pipe	Active			HW Dataplane / Switch	SDK, APIs, Libraries		
Open Network Linux (ONL)	Open Compute Project	Linux distribution for "bare metal" switches, that is, network forwarding devices built from commodity components.	http://opennetlinux.org/	https://github.com/OpenComputeProje	GNU GPL v2	opennetworklinux@googlegroup	https://groups.google.com/forum/#!	https://groups.google.com/forum/	Active			HW Dataplane / Switch	SDK, APIs, Libraries		
Facebook Open Switching System (FBOS	Facebook	Facebook's software stack (user-space applications, libraries, and utilities) for controlling and managing network switches.	https://github.com/facebook/fboss	https://github.com/facebook/fboss	BSD license				Active			HW Dataplane / Switch	SW Dataplane / Switch	SDK, APIs, Libraries	
OpenDaylight (ODL)	Linux Foundation	Production-ready open SDN platform containing features, protocols and plug-ins that can be integrated in a number of ways to deliver a broad set of SDN use cases	https://www.opendaylight.org	https://github.com/opendaylight	Apache 2.0	controller-users@lists.opendayli	https://lists.opendaylight.org/mailm	https://lists.opendaylight.org/pipermail/controller-users/				SDN Controller	Virtualization Platform		
ONOS	Linux Foundation	Carrier-grade SDN network operating system designed for high availability, performance, scale-out	http://onosproject.org/	https://github.com/opennetworkinglab/	Apache 2.0	onos-discuss@onosproject.org	http://groups.google.com/a/onospr	https://groups.google.com/a/ono	Active			SDN Controller	Virtualization Platform		
Floodlight	Big Switch Networks	Java-based OpenFlow 1.0 controller	http://www.projectfloodlight.org/floodl	https://github.com/floodlight/floodlight	Eclipse Public License version 1	floodlight-dev@openflowhub.org	http://groups.google.com/a/openflo	https://groups.google.com/a/openflowhub.org/forum/?fromgroups#!forum/floodlight-dev				SDN Controller	Virtualization Platform		
Ryu	NTT	Python-based OpenFlow 1.x controller	https://osrg.github.io/ryu/	https://github.com/osrg/ryu	Apache 2.0	ryu-devel@lists.sourceforge.net	https://lists.sourceforge.net/lists/list	http://sourceforge.net/mailarchiv	Active			SDN Controller	Virtualization Platform		
Trema	NEC	Trema is a full-stack framework for developing OpenFlow controllers in Ruby and C.	https://trema.github.io/trema/	https://github.com/trema/trema	GNU GPL v2	trema-dev@googlegroups.com	https://groups.google.com/group/t	https://groups.google.com/forum/#!forum/trema-dev				SDN Controller	Virtualization Platform		
OpenMul	OpenMUL Foundation	Base SDN/OpenFlow controller platform written almost entirely in C (from scratch) and provides top performance in terms of flow handling (download rate and latency) as well as a very stable application development platform.	http://www.openmul.org/openmul-com	https://github.com/openmul/openmul	GNU GPL v2			http://www.openmul.org/discussion.html#				SDN Controller	Virtualization Platform		
POX	Stanford University	Python-based OpenFlow 1.0 controller used for research and experimentation	https://github.com/noxrepo/pox	https://github.com/noxrepo/pox	Apache 2.0	pox-dev@lists.noxrepo.org	http://lists.noxrepo.org/listinfo.cgi/	http://lists.noxrepo.org/pipermail/	Closed			SDN Controller	Virtualization Platform		
Beacon	Stanford University	Java-based OpenFlow 1.0 controller	https://openflow.stanford.edu/display/	git@gitolite.stanford.edu:/beacon.git	BSD License			https://openflow.stanford.edu/list	Closed			SDN Controller			
SNAC	Stanford University	OpenFlow 1.0 controller with network access control application							Closed			SDN Controller	SDN Application	Security	
NOX	Stanford University	First OpenFlow 1.0 controller implementation	https://github.com/noxrepo/nox	https://github.com/noxrepo/nox	Apache 2.0	nox-dev@lists.noxrepo.org	http://lists.noxrepo.org/listinfo.cgi/	http://lists.noxrepo.org/pipermail/	Closed			SDN Controller			
IRIS	ETRI	The Recursive SDN Openflow Controller by ETRI is an open source version of IRIS. IRIS is an Openflow-based SDN controller designed to solve scalability and availability issues of SDN.	http://openiris.etri.re.kr/	https://github.com/openiris/IRIS								SDN Controller			
ExaBGP	Exa Networks	ExaBGP provides a convenient way to implement Software Defined Networking by transforming BGP messages into friendly plain text or JSON, which can then be easily handled by simple scripts or your BGP/DOSS.	https://github.com/Exa-Networks/exab	https://github.com/Exa-Networks/exab	BSD	exabgp-users@googlegroups.co	https://groups.google.com/forum/#!	https://groups.google.com/forum/	Active			vRouter	BGP	WAN	DCP
GoBGP	NTT	GoBGP is an open source BGP implementation designed from scratch for modern environment and implemented in a modern programming language, the Go Programming Language.	https://osrg.github.io/gobgp/	https://github.com/osrg/gobgp	Apache 2.0	gobgp-devel@lists.sourceforge.n	https://lists.sourceforge.net/lists/list	http://sourceforge.net/mailarchiv	Active			vRouter	BGP	WAN	DCP
Bird	CZ.NIC	IP Routing Stack	http://bird.network.cz/	git://git.nic.cz/bird.git	GNU GPL	bird-users@network.cz	http://bird.network.cz/mailman/listin	http://bird.network.cz/pipermail/bi	Active			vRouter	BGP	WAN	DCP
Quagga	OpenSourceRouting	IP Routing Stack	https://www.opensourcerouting.org/	http://git.savannah.gnu.org/cgit/quagga.git	GNU GPL	quagga-users@lists.quagga.net	https://lists.quagga.net/mailman/list	https://lists.quagga.net/pipermail/q	Active			vRouter	BGP	WAN	DCP
Calico	Tigera	Highly efficient vRouter in each compute node that leverages the existing Linux kernel forwarding engine without the need for vSwitches	https://www.projectcalico.org/	https://github.com/projectcalico/calico	Apache 2.0	calico-announce@lists.projectcal	http://lists.projectcalico.org/mailma	http://lists.projectcalico.org/piper	Active			vRouter	BGP	WAN	
XORP	XORP	IP Routing Stack	http://www.xorp.org/	https://github.com/greearb/xorp.ct	GNU GPL	xorp-users@xorp.org	http://mailman.icsi.berkeley.edu/m	http://mailman.icsi.berkeley.edu/	Active			vRouter	BGP	WAN	
Astara	Akanda	Orchestration platform based on Layer 3 agnostic and interfaces with the OpenStack Neutron REST APIs featuring sophisticated lifecycle management to monitor, configure, and manage 3rd party virtualized routers, load balancers and firewalls.	http://astara.io/	https://github.com/openstack/astara	Apache 2.0	#openstack-astara channel on freenode.net						nFVD	vRouter	OpenStack	VIM

https://docs.google.com/spreadsheets/d/1NHI4MZZWVDpxF_Rs7OOSTUa_aHL2ACUVA_Ov-YQs1DA/edit#gid=0

Concluding Remarks

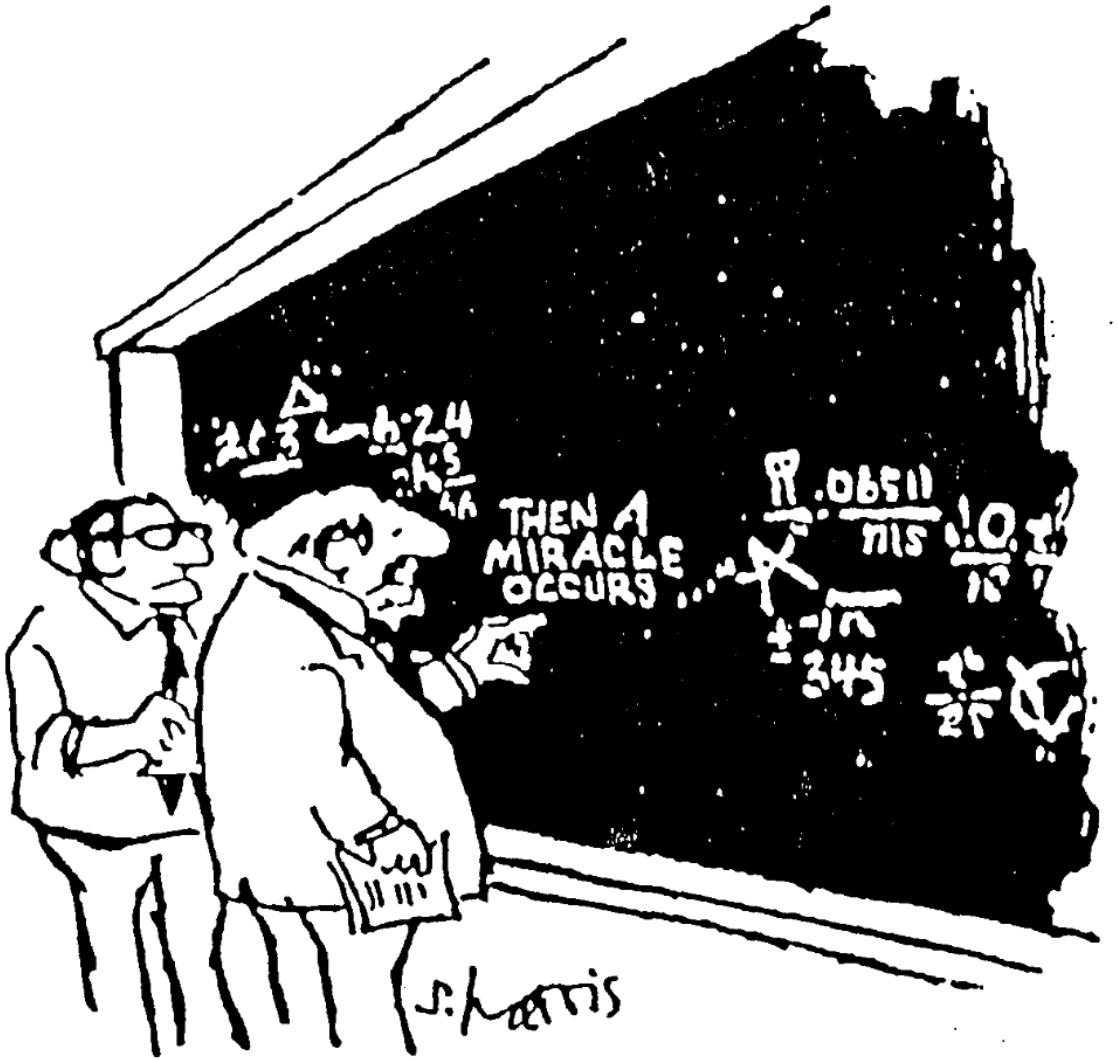
Think and be “open” in your research

- Carry effective open source strategy (tool selection + upstream contributions)
- Reproducible research (open source + open data)

The future is all about Software Ecosystems

- Open Interfaces: Protocols, APIs, Code, Tool Chains
- Open Control Platforms at every level (layer?)
- “Best of Breed” markets

Thank you!
Questions?



INTRIG
INFORMATION & NETWORKING TECHNOLOGIES RESEARCH & INNOVATION GROUP

UNICAMP
ANO 50

FEEC
Faculdade de Engenharia Elétrica e de Computação
UNICAMP

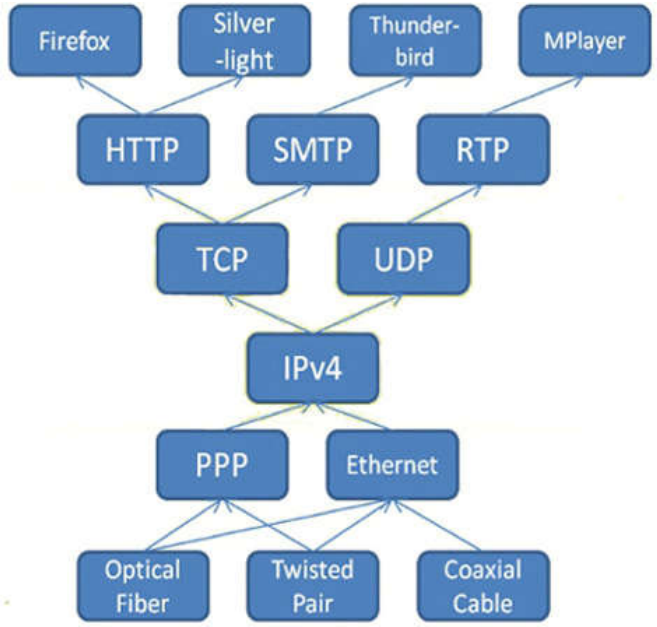
BACKUP

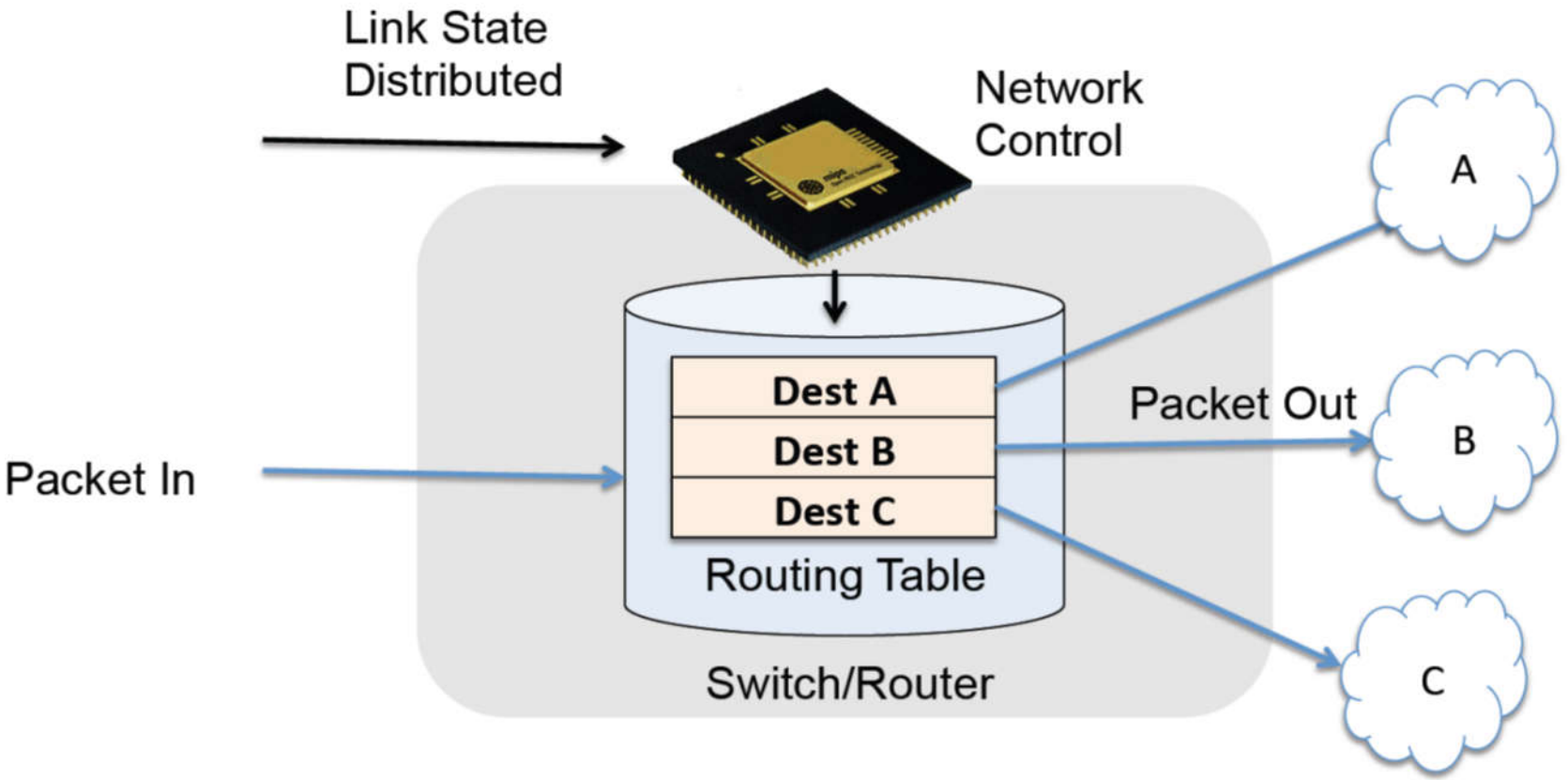
Phase 3: Open the Wiki to the public community

- Managed by the community
  - Open to any registered user
  - Scales nicely / cost-effective (effort/per-person)
  - Content updates, curation, etc.
- Area caretakers to validate updates

Image Source: Photographed by Lane Hartwell (http://fetching.net/) on behalf of the Wikimedia Foundation.
https://commons.wikimedia.org/wiki/File:Wikipedia_mini_globe_handheld.jpg

Networking as Learned in School (text books)

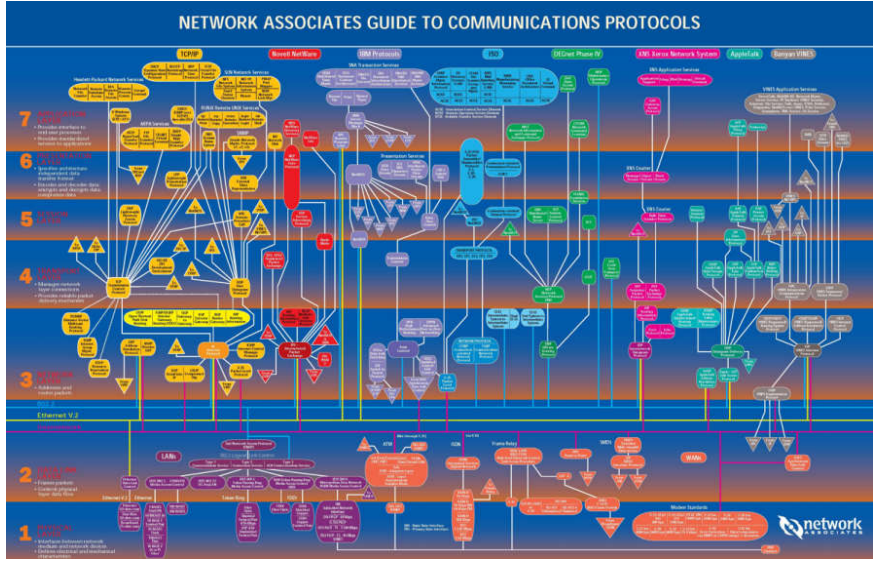


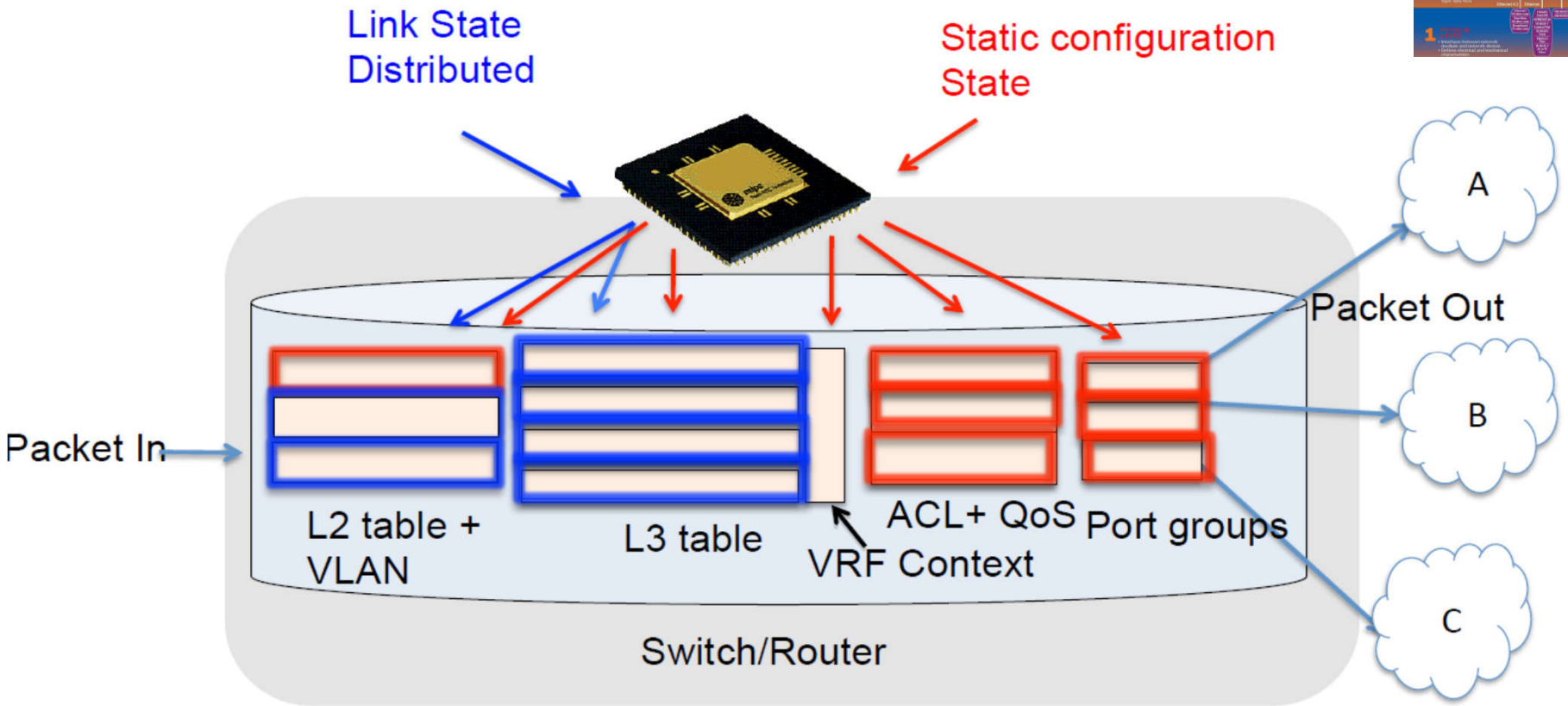


Source: Martin Casado CS244 Spring 2013, Lecture 6, SDN

Networking in Practice

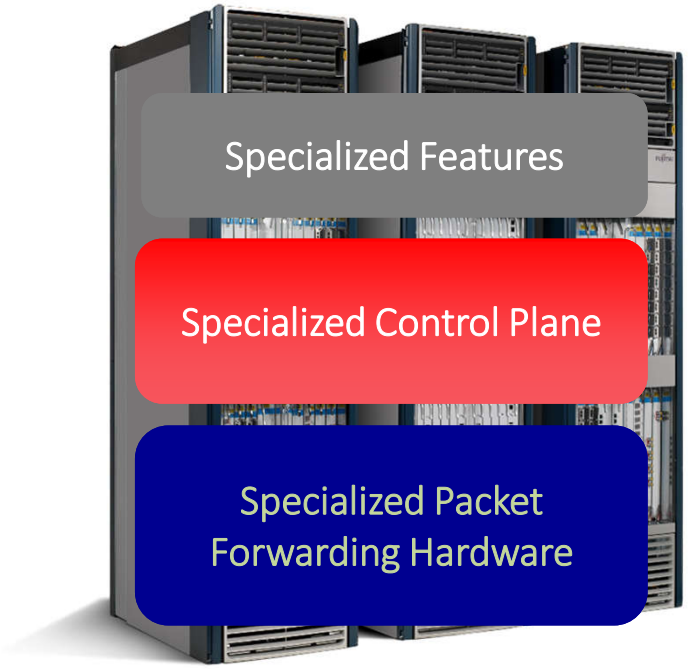
"in theory, theory and practice are the same; in practice they are not..."





Source: Martin Casado CS244 Spring 2013, Lecture 6, SDN

Problem with Internet Infrastructure



Hundreds of protocols
6,500 RFCs

Tens of Millions of lines of code
Closed, proprietary, outdated

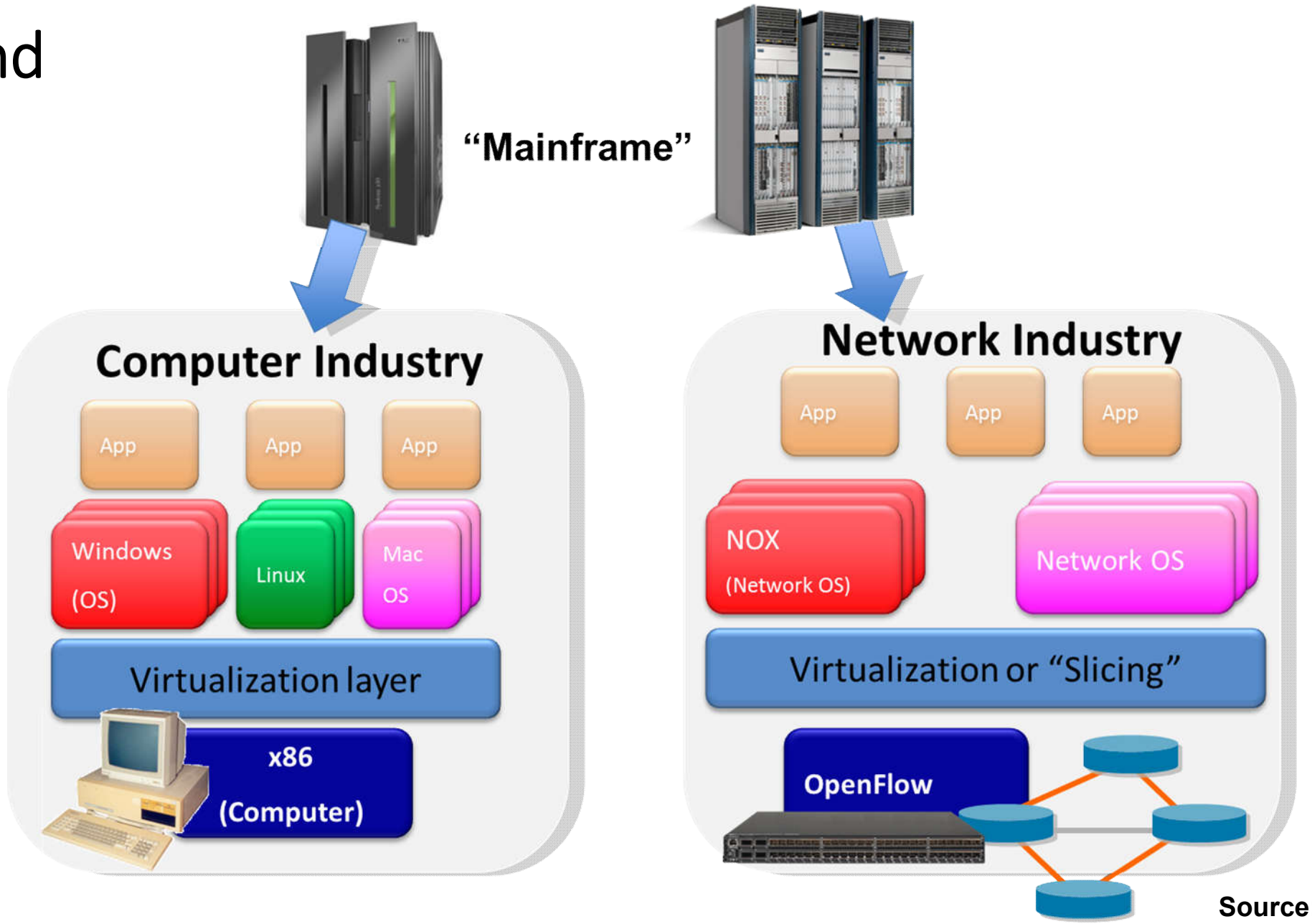
Billions of gates
Power hungry and bloated

Vertically integrated, complex, closed, proprietary

Not good for network owners and users

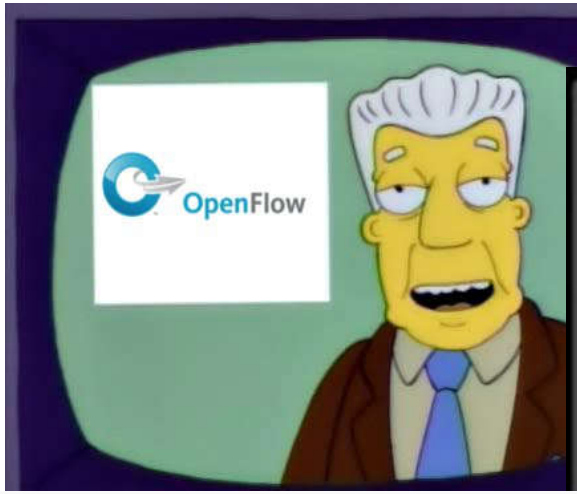
Source: ON.LAB

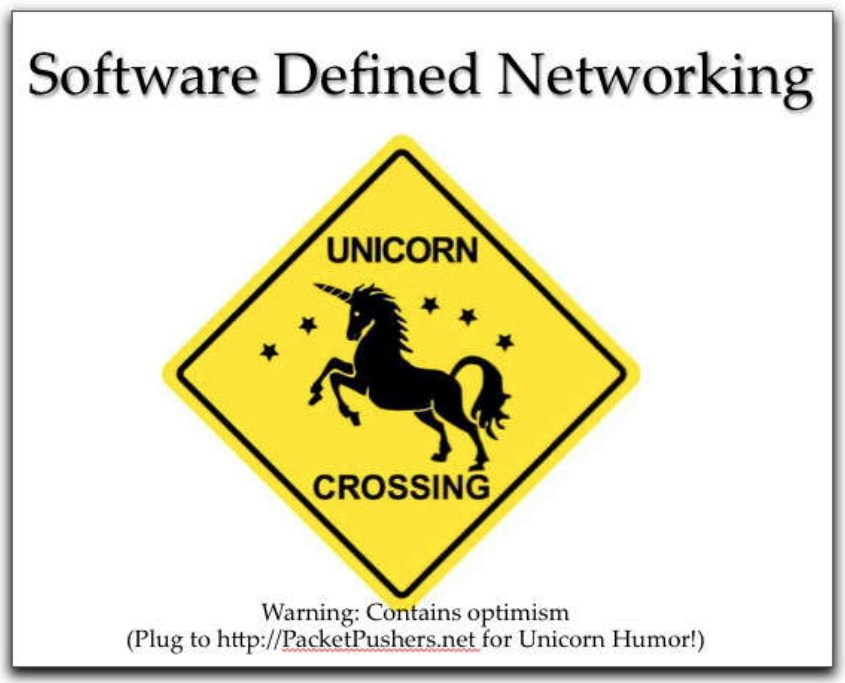
Trend



Source: ON.LAB

SDN to the rescue!





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 SIGCOMM?) – Researcher Z

What is SDN?

In the SDN architecture, the control and data planes are decoupled, network intelligence and state are logically centralized, and the underlying network infrastructure is abstracted from the applications.

– Open Networking Foundation white paper

Software Defined Networking (SDN) refactors the relationship between network devices and the software that controls them. Open interfaces to network switches enable more flexible and predictable network control, and they make it easier to extend network function.

– HotSDN CFP

SDN definitions

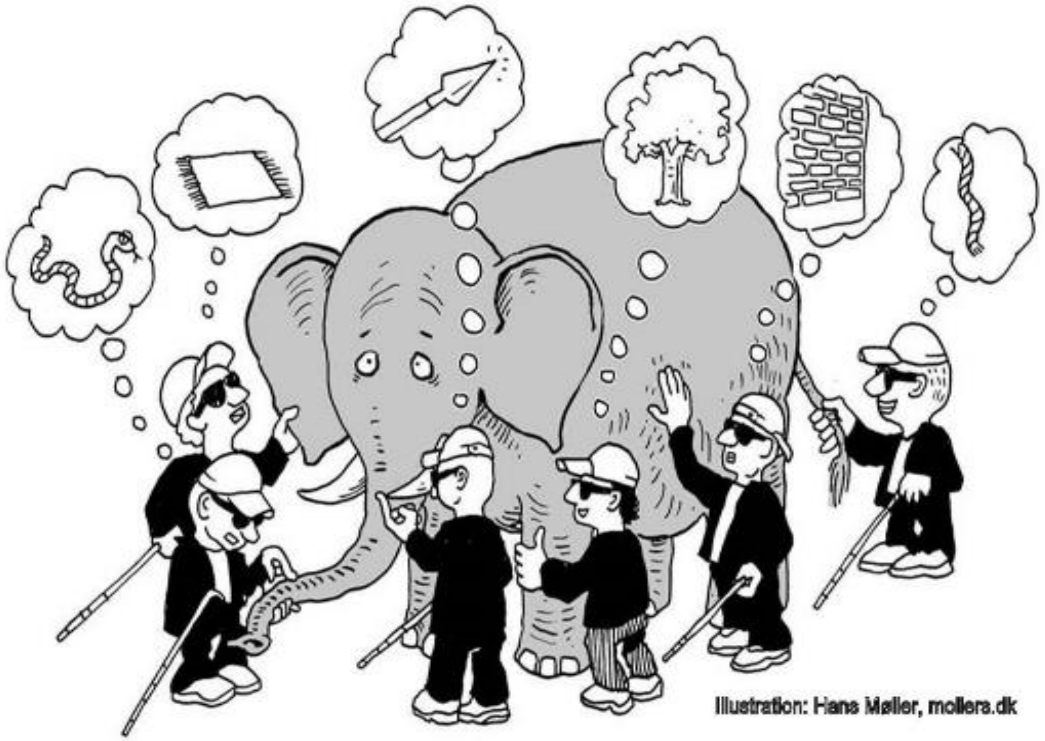
- With the original (OpenFlow) definition, SDN represented a network architecture where the forwarding state is solely managed by a control plane and is decoupled from the data plane.
- The industry, however, has moved on from the *original academic purist view* of SDN to referring *to anything disruptive or fundamentally new* as part of SDN.

At least two definitions for SDN:

1.academic
(purist view : strict decoupling of the data and control plane)

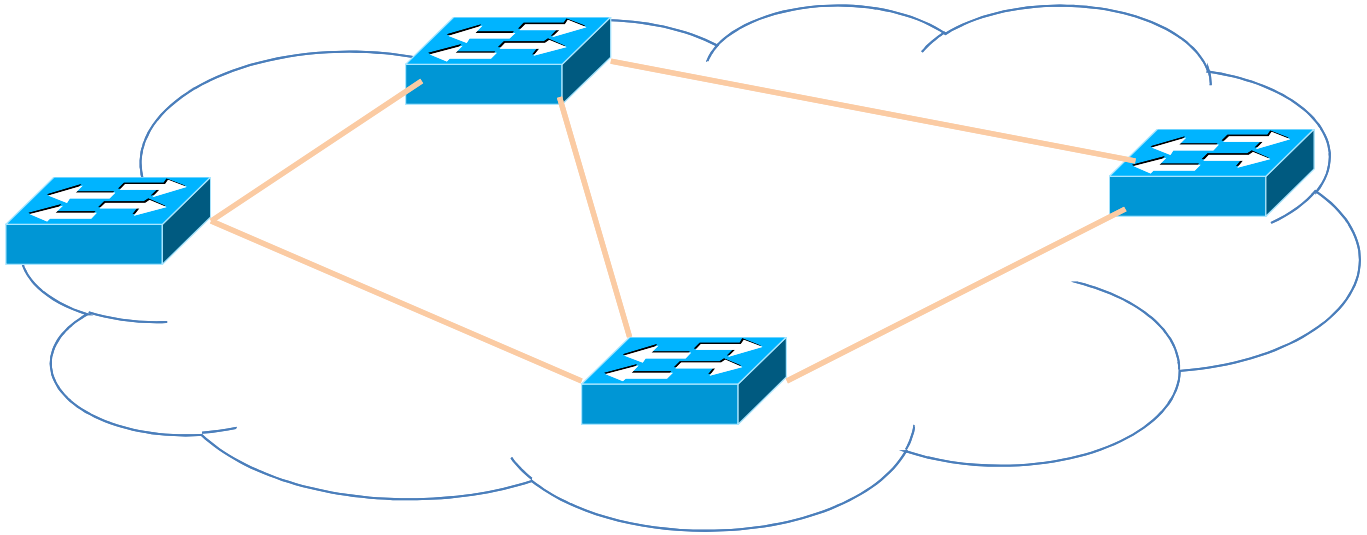
2.industry
(many-fold business-driven views)

SDN :: Evolving Definition



Rethinking the "Division of Labor"
Traditional Computer Networks

Data plane:
Packet streaming

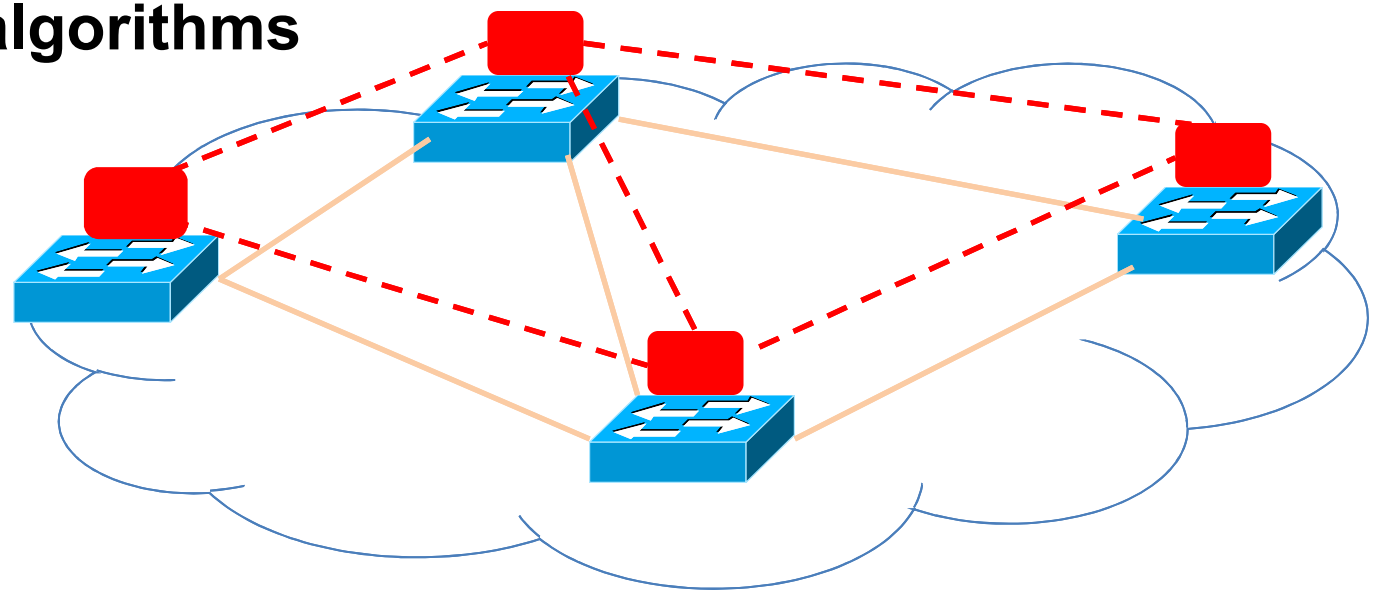


Forward, filter, buffer, mark, rate-limit, and measure packets

Source: Adapted from J. Rexford

Rethinking the "Division of Labor"
Traditional Computer Networks

Control plane:
Distributed algorithms

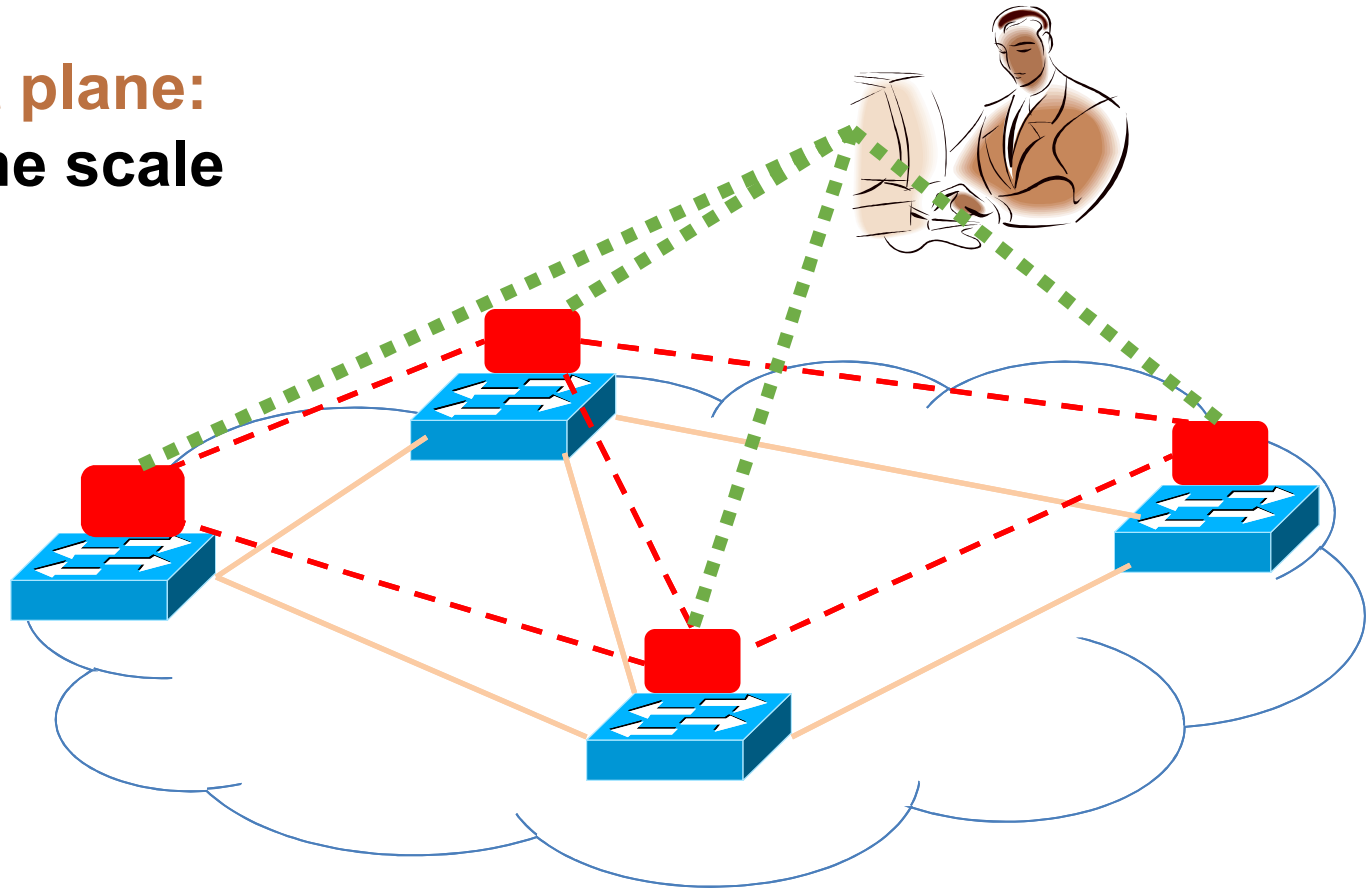


Track topology changes, compute routes, install forwarding rules

Source: Adapted from J. Rexford

Rethinking the "Division of Labor"
Traditional Computer Networks

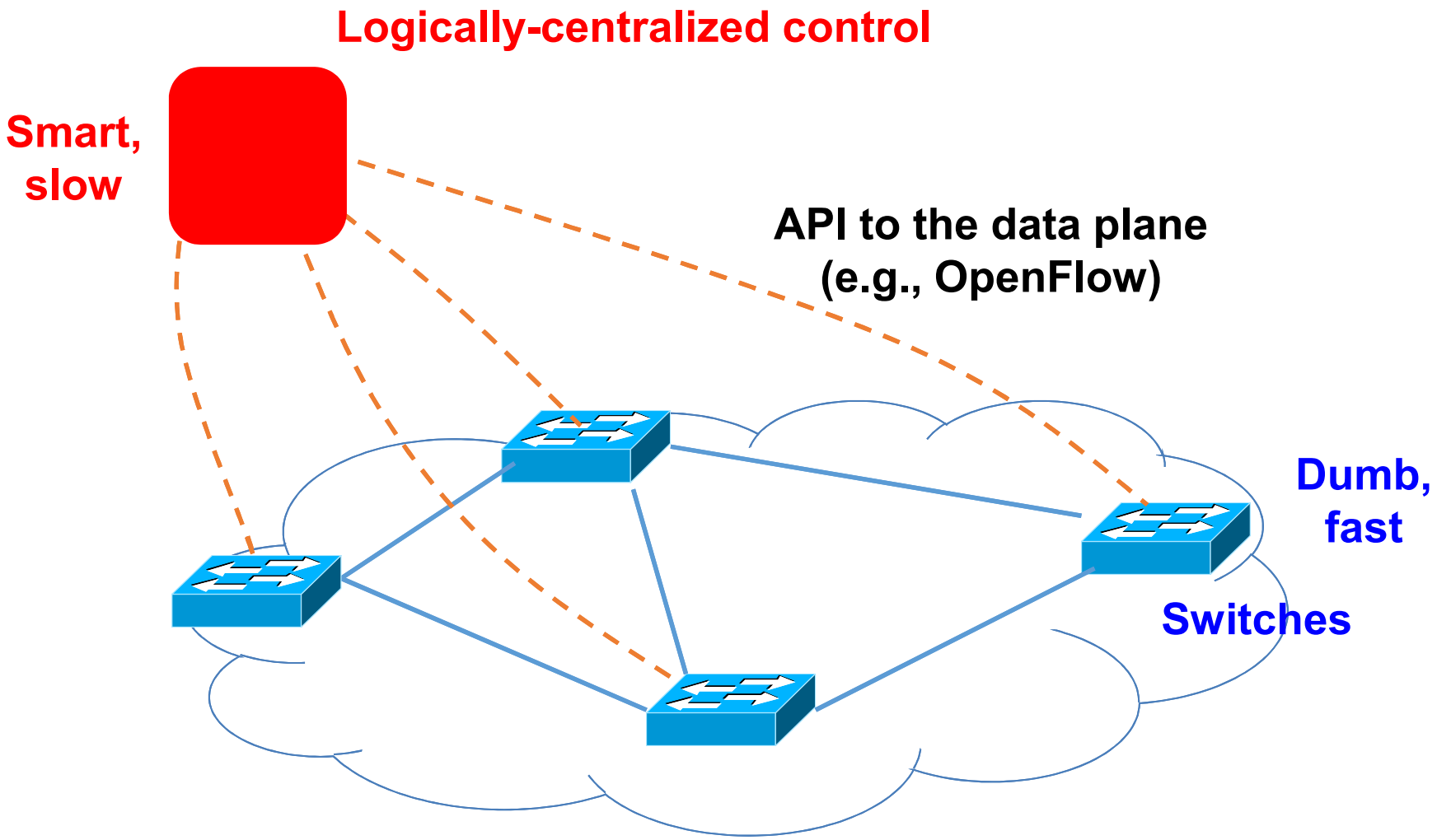
Management plane:
Human time scale



Collect measurements and configure the equipment

Source: Adapted from J. Rexford

Software Defined Networking (SDN)

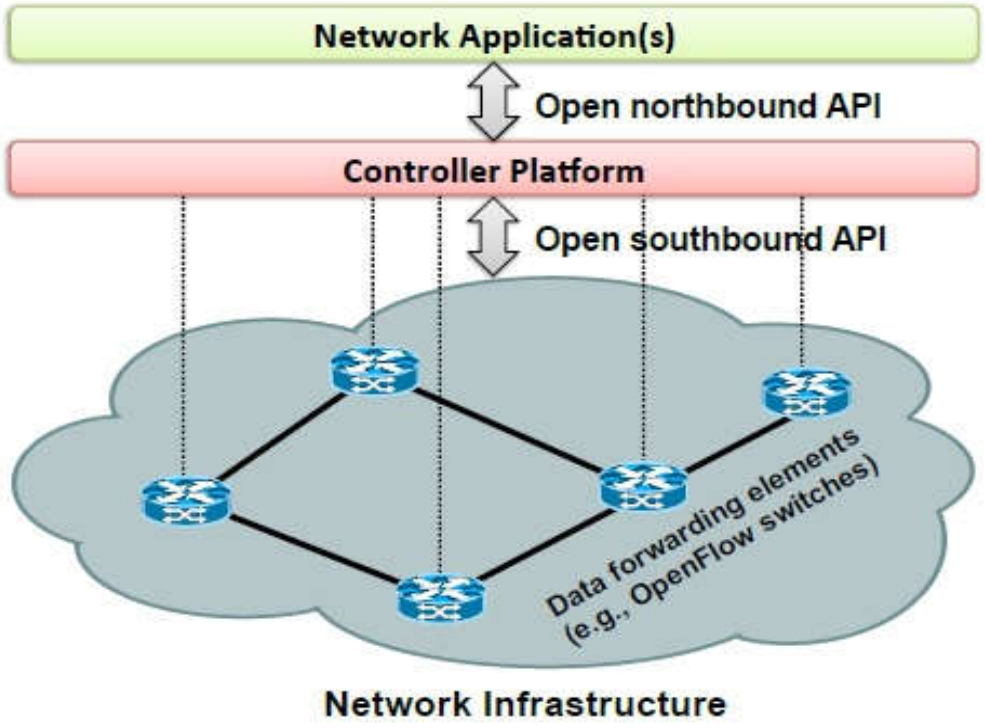


Source: Adapted from J. Rexford

SDN: Definitions, Concepts, and Terminology

SDN refers to software-defined networking architectures where:

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

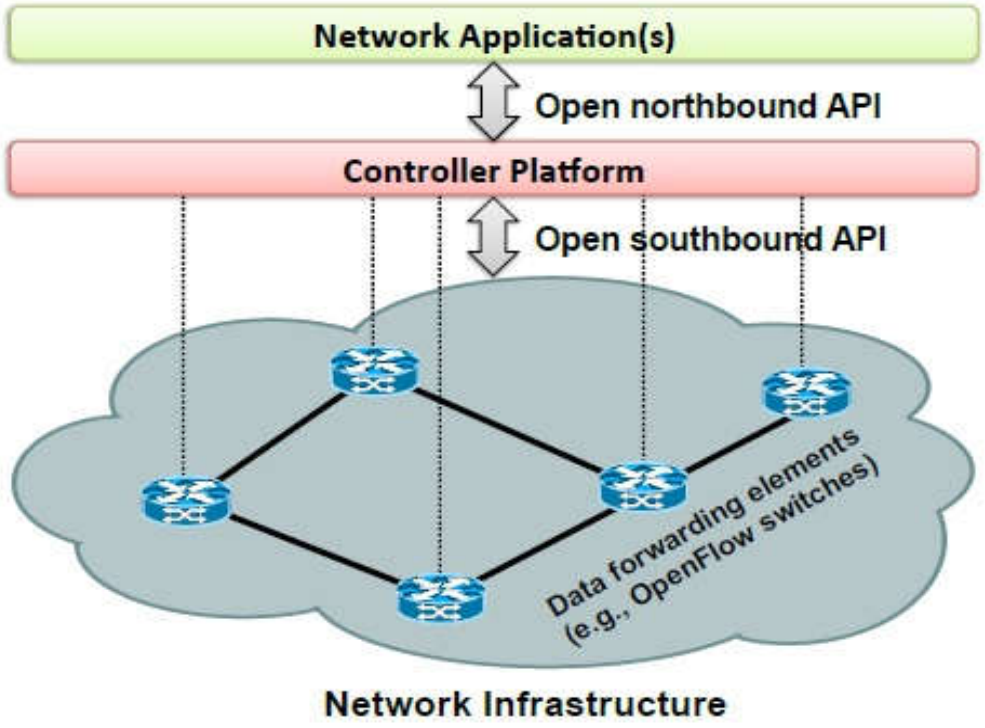


Source:
“Software-Defined Networking: A Comprehensive Survey”,
Kreutz et al., In Proceedings of the IEEE, Vol. 103, Issue 1, Jan. 2015..

SDN: Definitions, Concepts, and Terminology

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

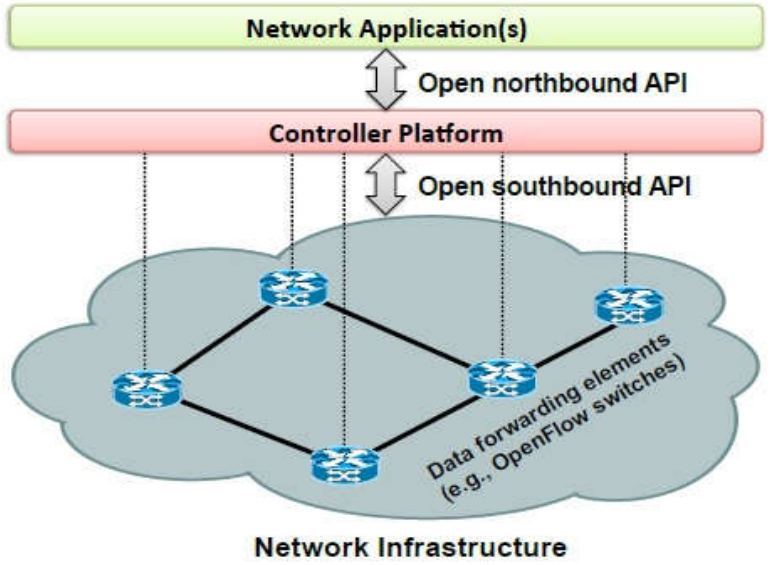
- **Southbound interface:**
  (instruction set to program the data plane
  +
  (protocol btw control- and data planes).
  E.g., OpenFlow, POF, Forces, Netconf



Source:
"Software-Defined Networking: A Comprehensive Survey",
Kreutz et al., In Proceedings of the IEEE, Vol. 103, Issue 1, Jan. 2015..

SDN: Definitions, Concepts, and Terminology

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



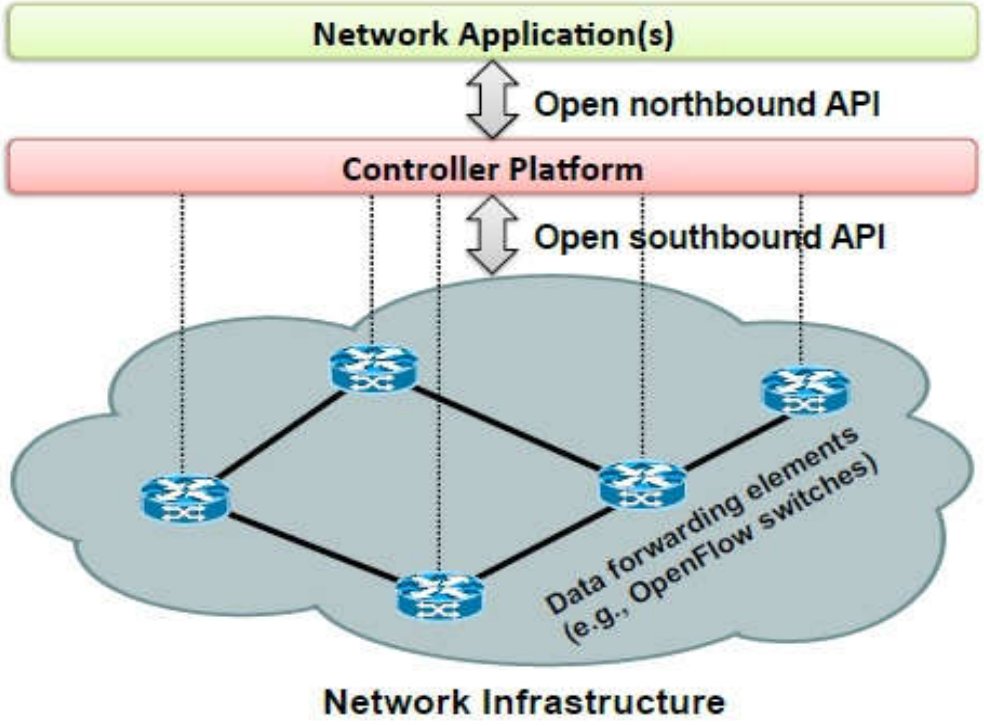
	Switch port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Prot	TCP sport	TCP dport	Action
Switching	*	*	00:1f:...	*	*	*	*	*	*	Port6
Flow switching	Port3	00:20..	00:1f..	0800	Vlan1	1.2.3.4	5.6.7.8	4	17264	Port6
Firewall	*	*	*	*	*	*	*	*	22	Drop
Routing	*	*	*	*	*	*	5.6.7.8	*	*	Port6
VLAN switching	*	*	00:1f..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

Source:
"Software-Defined Networking: A Comprehensive Survey",
Kreutz et al., In Proceedings of the IEEE, Vol. 103, Issue 1, Jan. 2015..

SDN: Definitions, Concepts, and Terminology

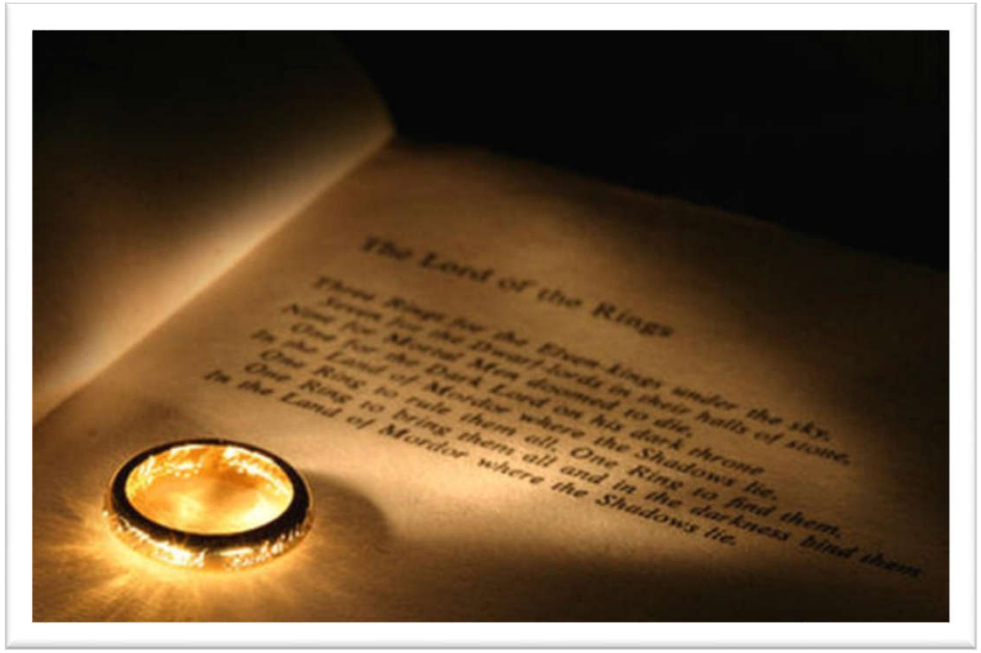
- **Northbound interface:** API offered by control plane to develop network control- and management applications.

- **Application Layer / Business Applications (Management plane):** functions, e.g., routing, traffic engineering, that use Controller functions / APIs to manage and control network infrastructure.



Source:
"Software-Defined Networking: A Comprehensive Survey",
Kreutz et al., In Proceedings of the IEEE, Vol. 103, Issue 1, Jan. 2015..

One SDN controller to rule them all, with
a discovery app to find them,
One SDN controller to tell them all, on
which switchport to bind them.
In the Data Center, where the packets fly.



One SDN to rule them all

Actually not, different reasonable models and approaches to SDN are being pursued

Source Poem: http://dovernetworks.com/?p=83
Further reading: http://theborgqueen.wordpress.com/2014/03/31/the-legend-of-sdn-one-controller-to-rule-them-all/

SDN asks (at least) three major questions

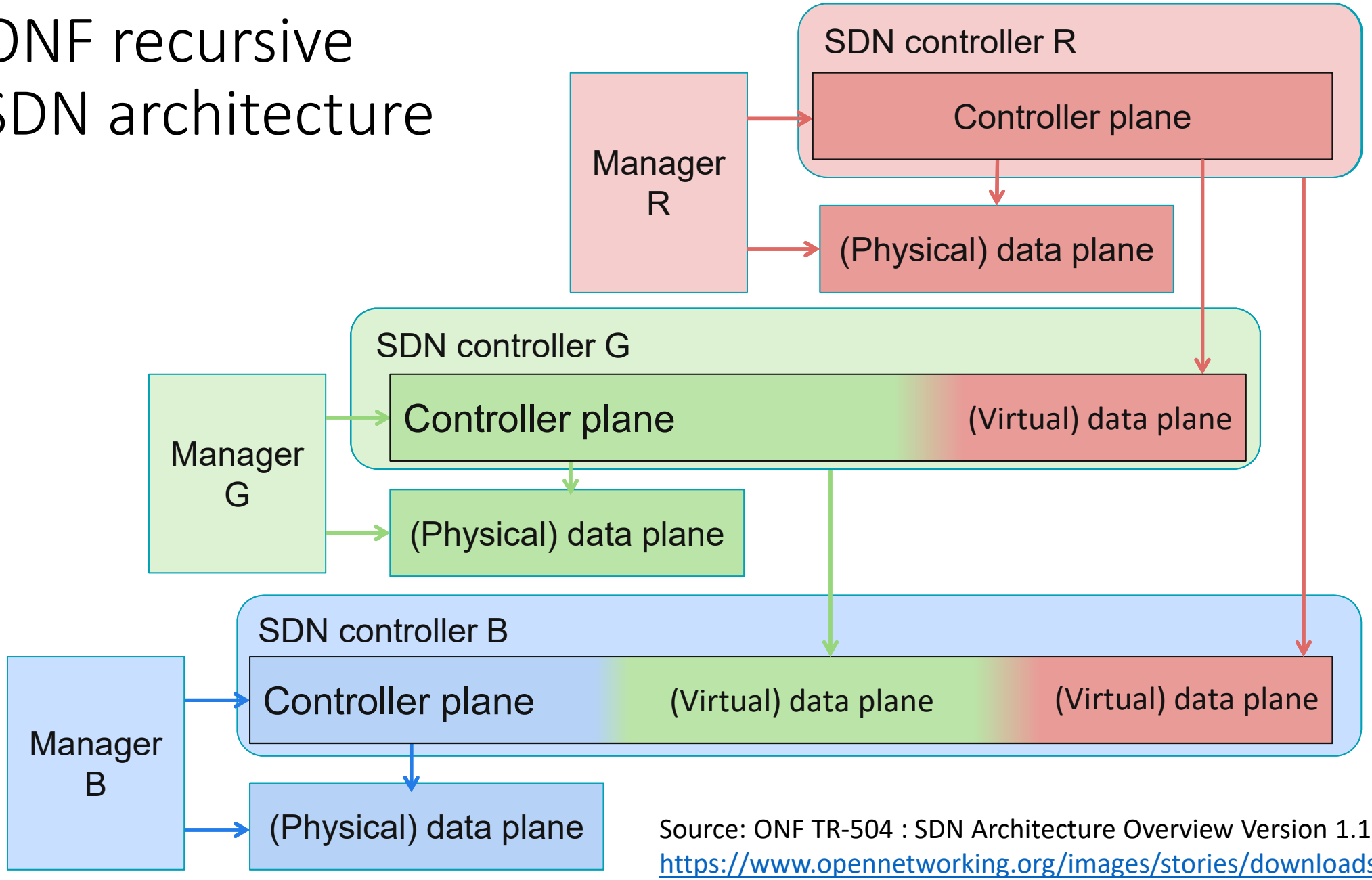
Where the control plane resides
"Distributed vs Centralized" ?

How does the Control Plane talk
to the Data Plane ?

How are Control and
Data Planes programmed ?

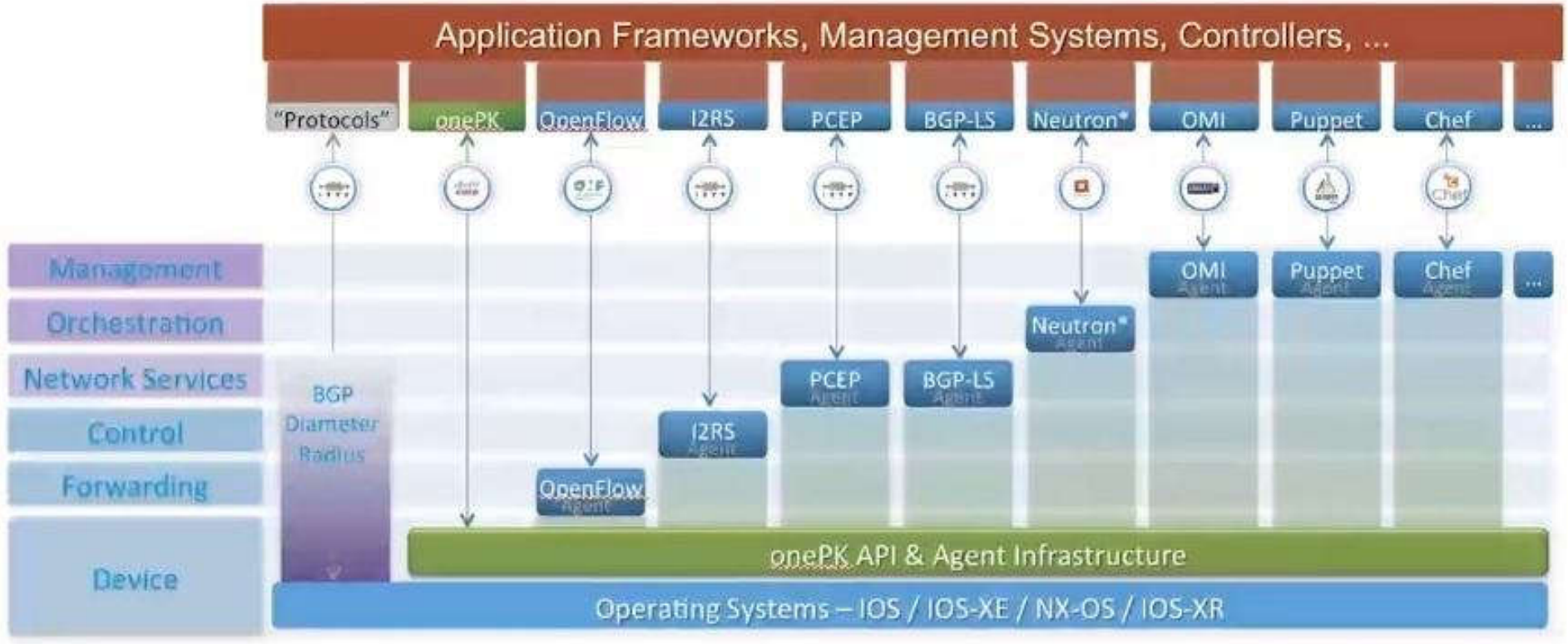
Source: Adapted from T. Nadeu, slides-85-sdnrg-5.pptx

ONF recursive SDN architecture



Source: ONF TR-504 : SDN Architecture Overview Version 1.1, https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR_SDN-ARCH-Overview-1.1-11112014.02.pdf

Network Programmability Layers



Source: Introducing Network Programmability Fundamentals
Part#: CTOD-SDN-1.0-017141 https://learningnetworkstore.cisco.com/skillsoft/introducing-network-programmability-fundamentals-ctod-sdn-1-0-017141

SDN asks (at least) three major questions

Where the control plane resides
"Distributed vs Centralized" ?

1

- **What state belongs in distributed protocols?**
- **What state must stay local to switches?**
- **What state should be centralized?**

- What are the effects of each on:
  - state synchronization overhead
  - total control plane overhead
  - system stability and resiliency
  - efficiency in resource use
  - control loop tightness

Source: E. Crabbe, slides-85-sdnrg-7.pdf

SDN asks (at least) three major questions

How does the Control Plane talk to the Data Plane ?

2

- **Prop. IPC**
- **OpenFlow (with or w/extensions)**
- **Open Source south-bound protocols**
- **Via SDN controller broker and south-bound plug-ins**
- **Other standardized protocols**

- What are the effects of each on:
  - Interoperability, Evolvability, Performance
  - Vendor Lock-in

SDN asks (at least) three major questions

How are Control and
Data Planes programmed ?

3

- **Levels of Abstraction**
- **Open APIs**
- **Standardized Protocols**

- What are the effects of each on:
  - Data plane flexibility
  - Integration with legacy
  - Interoperability (CP / DP)
  - Vendor lock-in

Source: E. Crabbe, slides-85-sdnrg-7.pdf

NFV Concepts

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

Source: Adapted from Raj Jain

NFV Concepts

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

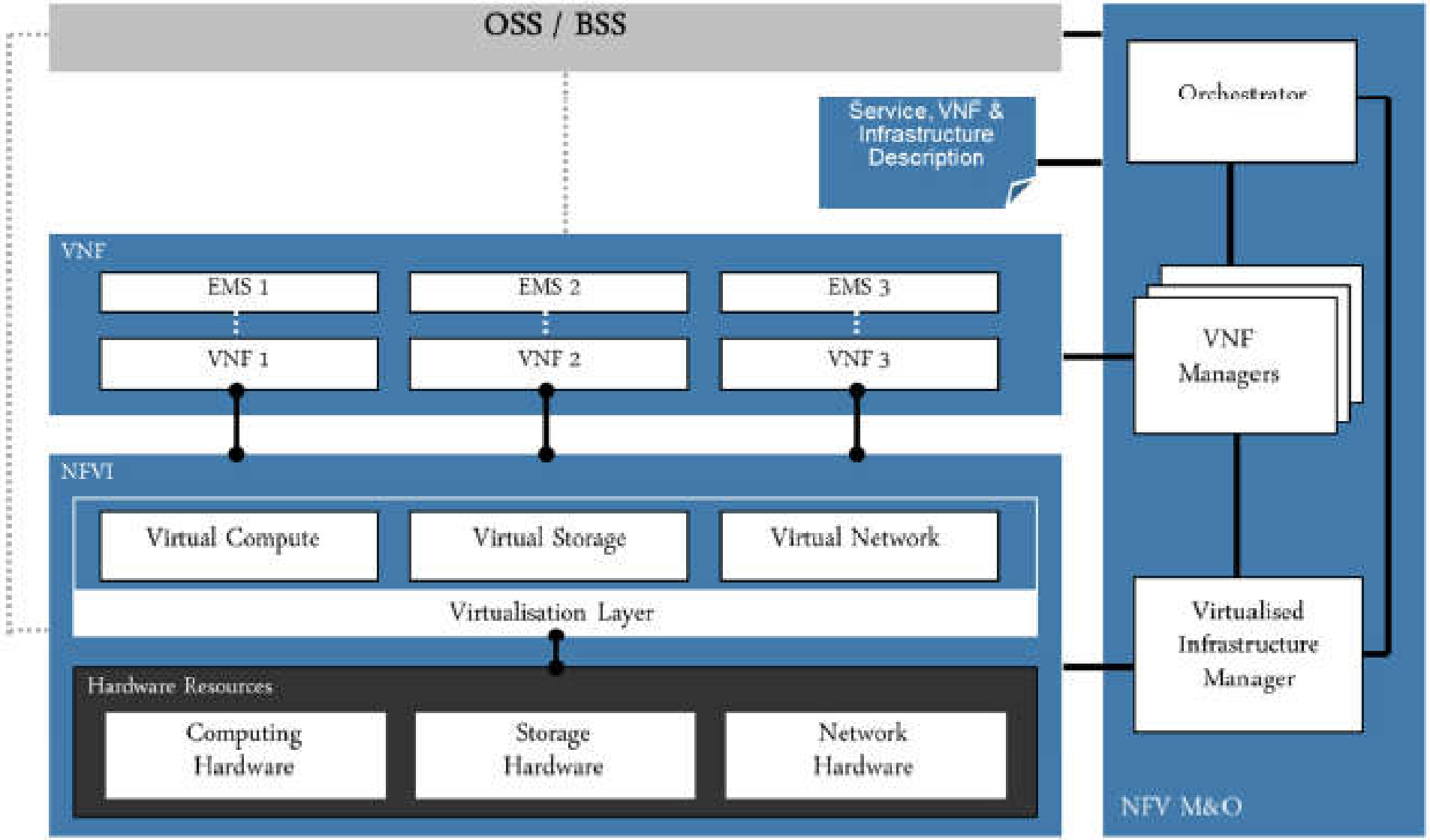
Source: Adapted from Raj Jain

NFV Concepts

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

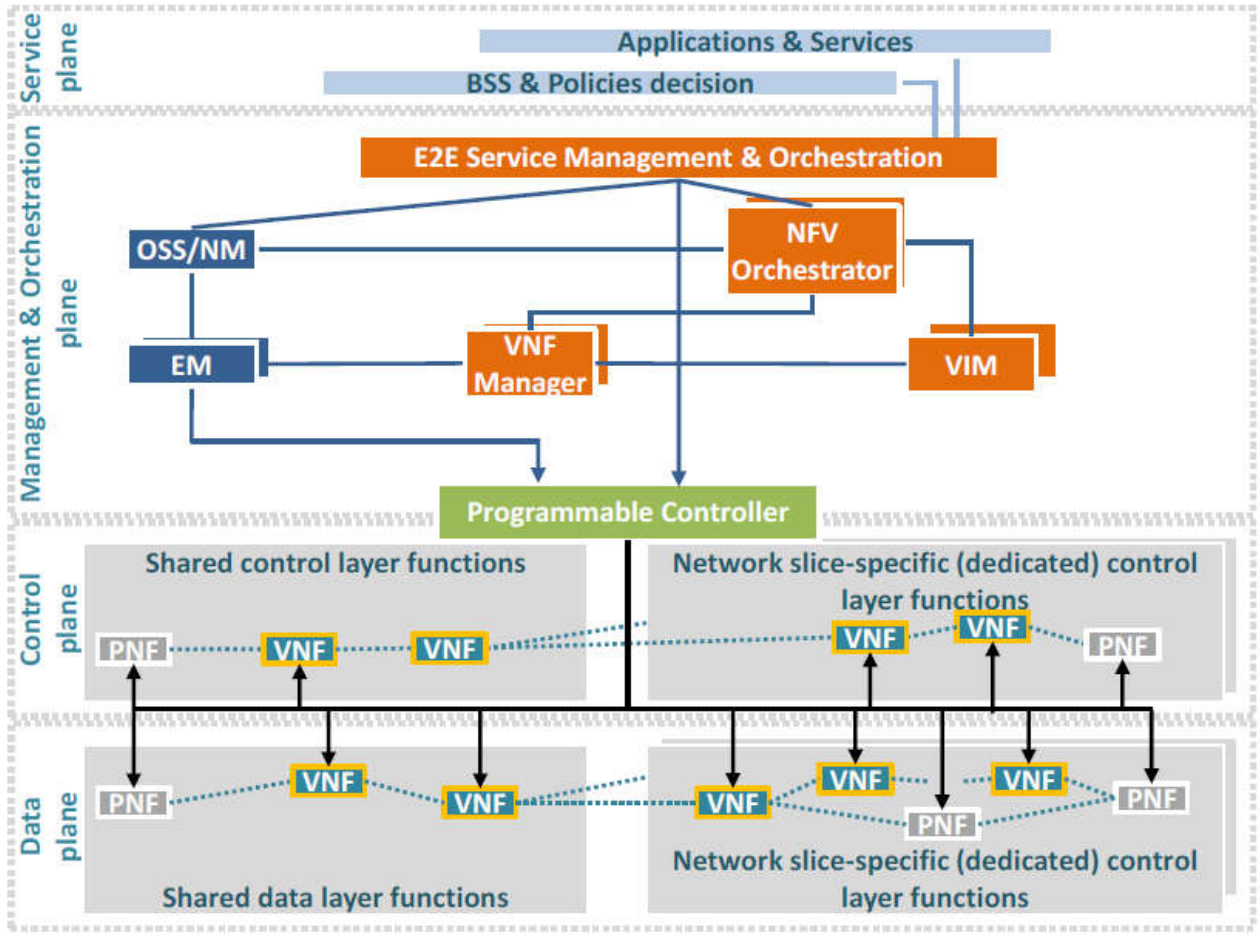
Source: Adapted from Raj Jain

Architectural Framework [ETSI NFV]



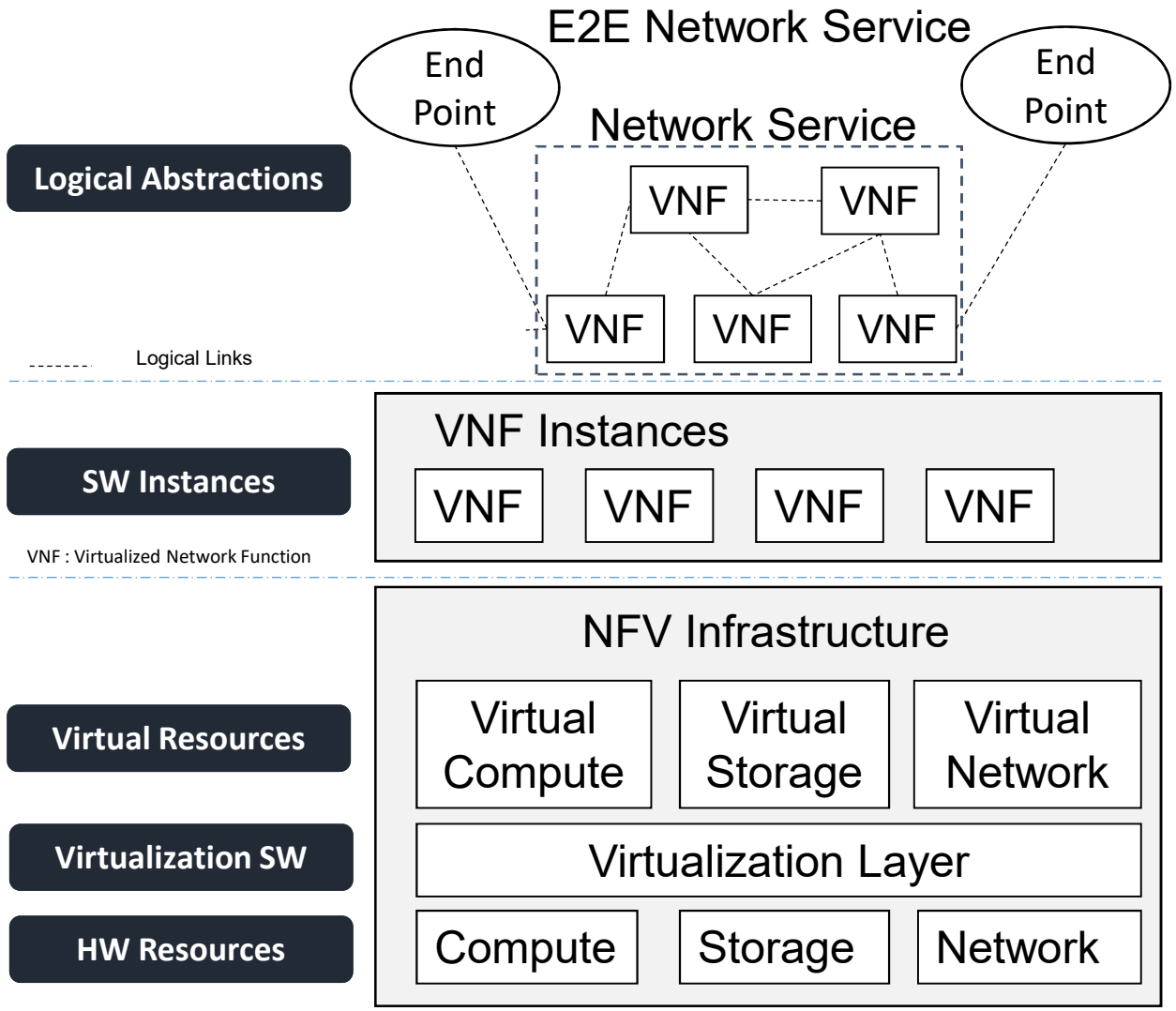
Souce: ETSI NFV White Paper 2

NFV



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

NFV Layers



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

Alternative options to virtualize NFV apps