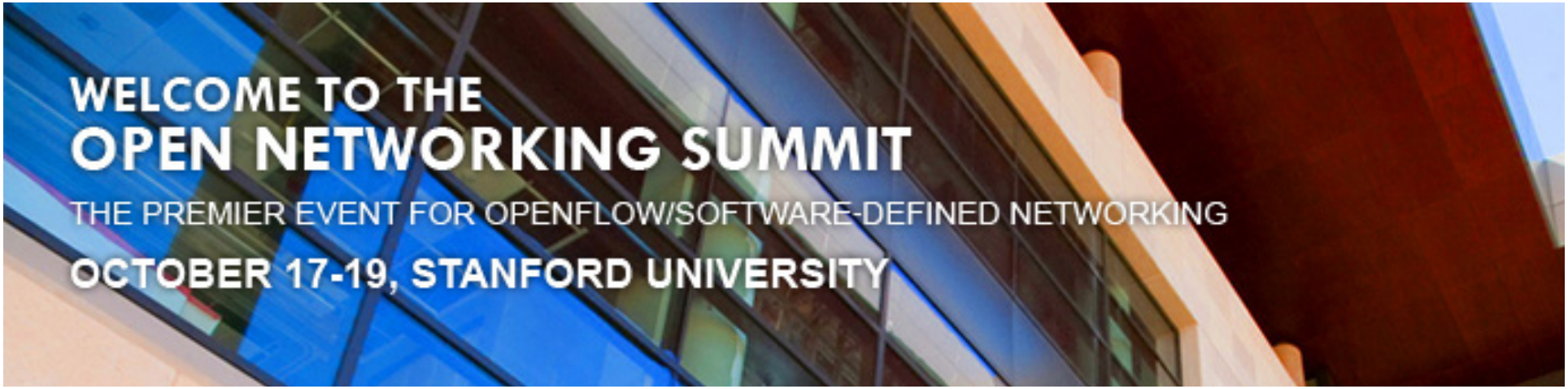

WELCOME TO THE
OPEN NETWORKING SUMMIT
THE PREMIER EVENT FOR OPENFLOW/SOFTWARE-DEFINED NETWORKING
OCTOBER 17-19, STANFORD UNIVERSITY

Time	Topic
9.30-10am	Speakers introduction and industry update panel
10.30-11am	Overview of OpenFlow/SDN concepts
11am - 12pm	OpenFlow/SDN in the service provider WAN
12.30-1pm	**Lunch Break**
1-2pm	OpenFlow/SDN in the campus LAN environment
2-2.30pm	OpenFlow/SDN for IaaS providers
2.30-3.30pm	OpenFlow/SDN in the enterprise data center
3.30-4pm	Wrap-up and closing comments

GOT QUESTIONS?

Please submit your questions via twitter using the hashtag #ONS2011T2

ABOUT YOU!

A quick set of stats about the audience

88 people registered

57 companies

67% vendors, 22% customers, 11% research

78% technical-ish (by title)

ABOUT US!

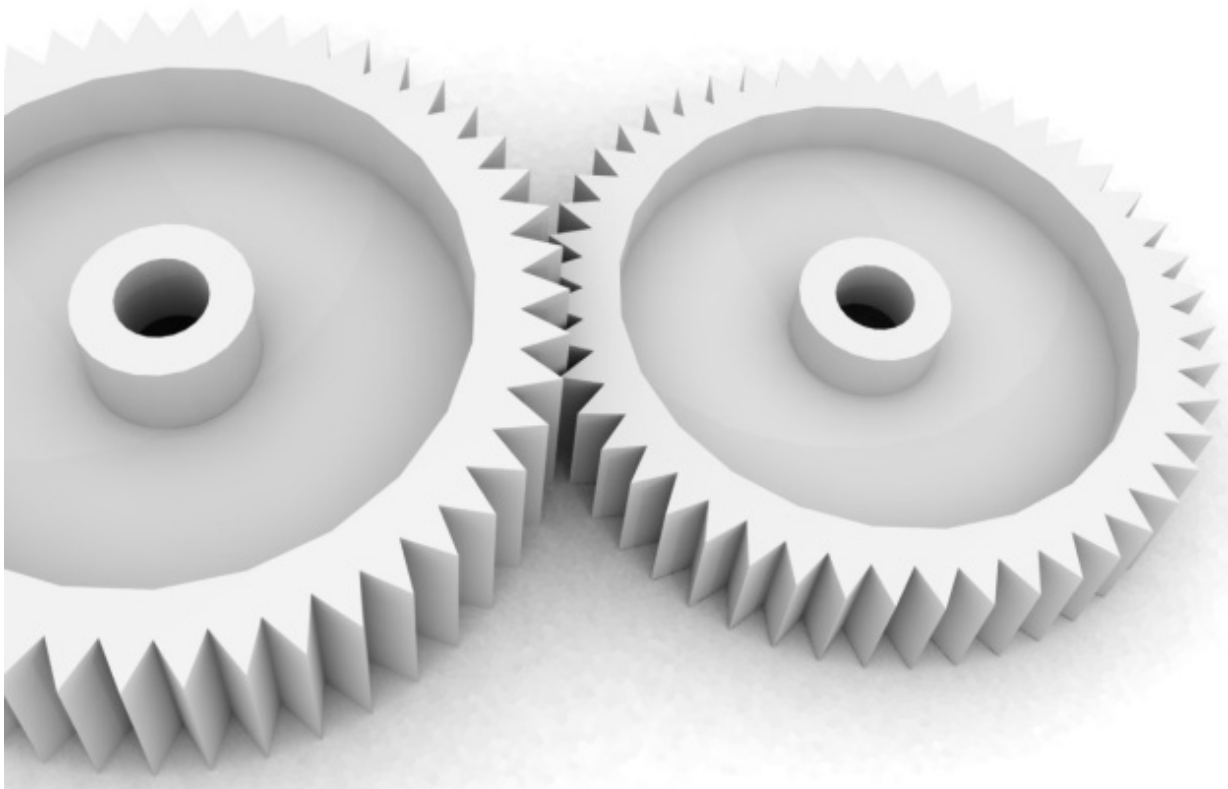
Speakers for today in (rough) order of appearance

Chris Liljenstolpe, Big Switch Networks, formerly Dir Cross-Domain Architecture at Telstra, Chief Arch at Woven, CTO of IP & Data at Alcatel, Chief Arch at Cable & Wireless, Co-Chair Operations WG at IETF

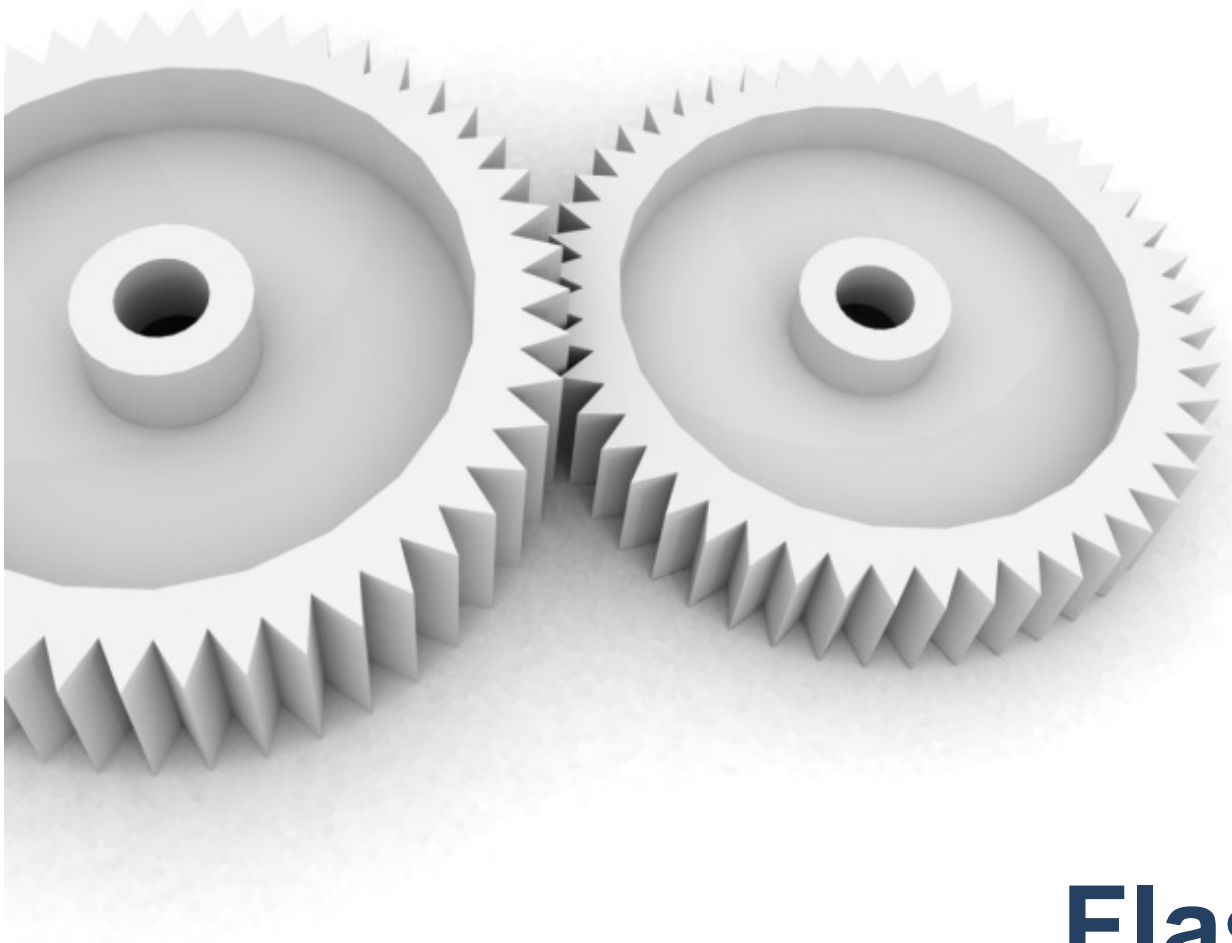
Matt Davy, Chief Architect Indiana University, Director InCNTRE, OpenFlow Evangelist at Internet2 / NDDI

Paul Lappas, Formerly VP of Engineering and Co-Founder at GoGrid, Eng and Ops Lead at ServePath, Sr Engineer at Epicentric, Application Engineer at Motorola (Iridium)

Peter Krey, Founder & President at Krey Associates, MD and Chief Architect / CTO at Deutsche Bank Asset Mgmt, VP BD at Storage Apps, (various) at Morgan Stanley and JP Morgan



Industry Update Panel



Flash Break

OpenFlow/SDN Concepts

Today's talk:

OpenFlow fundamentals

A few tech notes

Implications

THE BASICS OF OPENFLOW

Overview

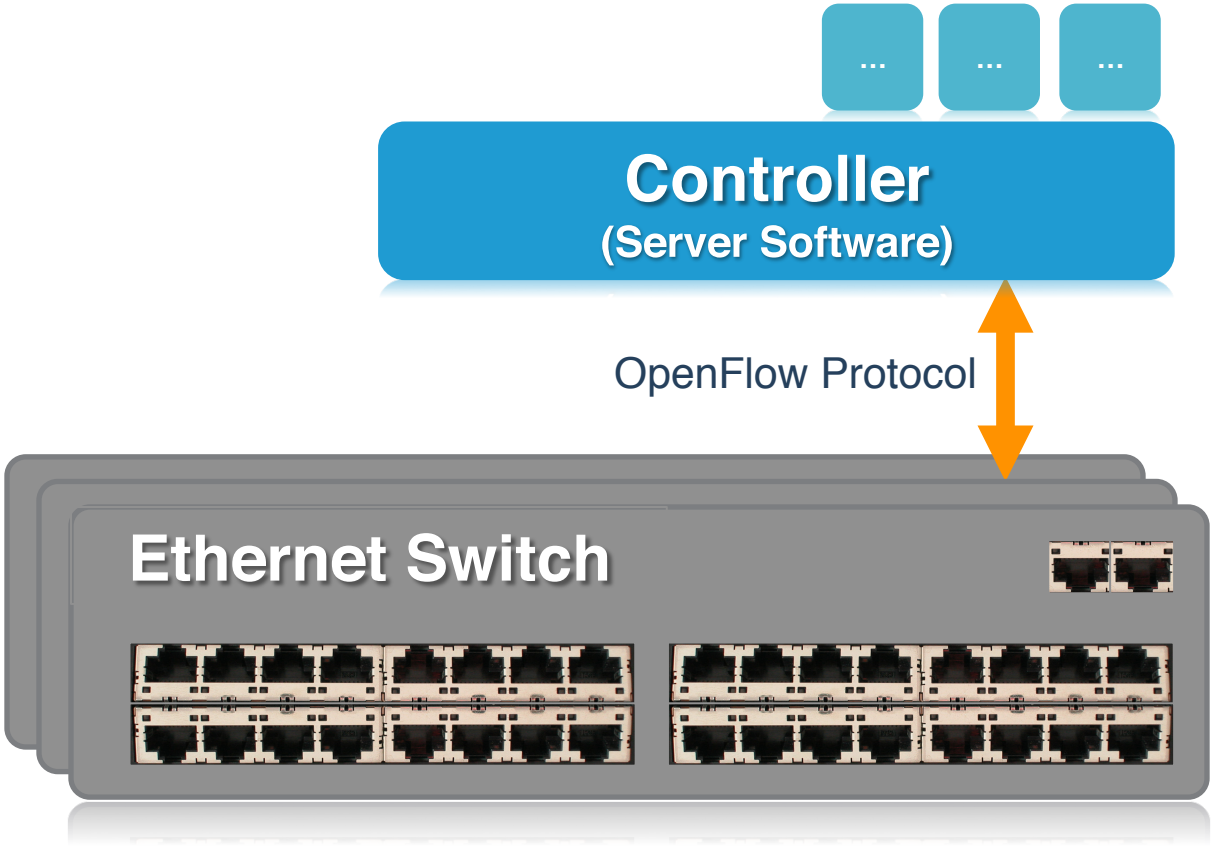
Protocol that allows an external server to control the data path of a switch

- Vendor Independent, works with today's networking chipsets
- Protocol is open source, core technologies are IP-clear
- Allows wide range of operations

OpenFlow

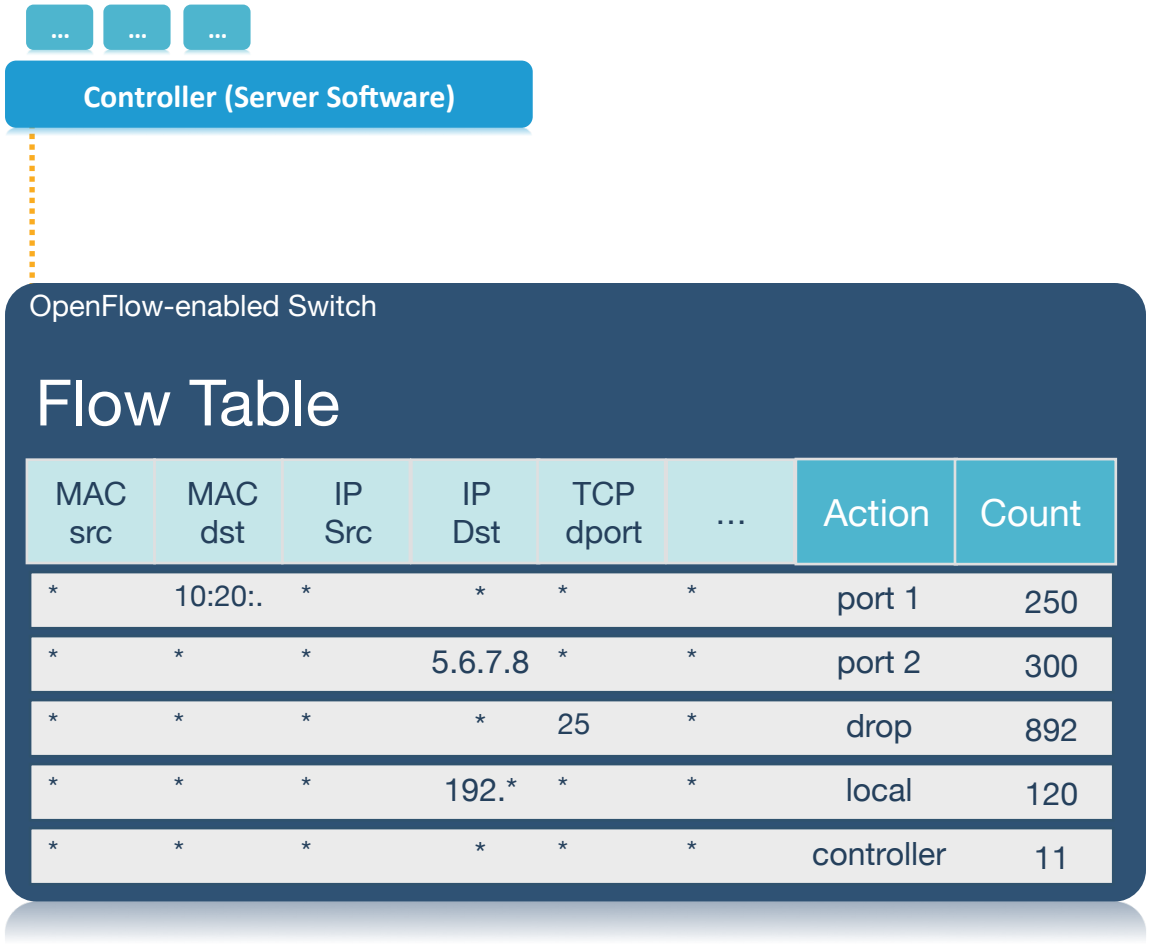
Controller is (usually) a data center-class Linux Server running a controller application. Open Source implementations exist.

OpenFlow client is added to switch software. Existing switches can be OpenFlow enabled via software upgrade.



AN OPENFLOW SWITCH SIMPLIFIED

Core of the OpenFlow switch: the flow table abstraction



Flow Table

Generic primitive that sits on top of switch TCAM, designed to match well with common switch ASICs.

Example actions:

1. Switching and routing (port),
2. Firewalling (drop),
3. Using to switch's non-OpenFlow logic (local),
4. Send to controller for processing (controller)

Foundation network functions are split between per-packet rules on the switch and high-level decisions at the server

OPENFLOW PROTOCOL, ARCHITECTURES AND SDN

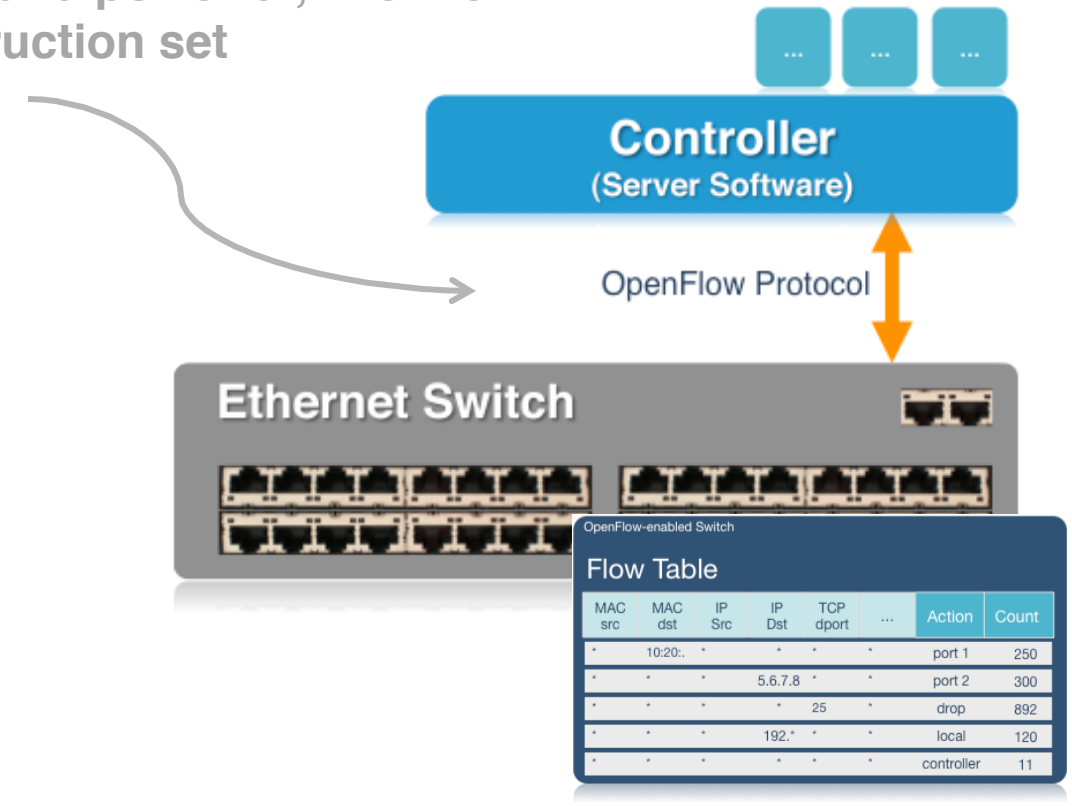
An architecture of switches, controllers and software applications

THE ARCHITECTURE

Rich variety of uses and applications, like x86 + Linux + applications + perl

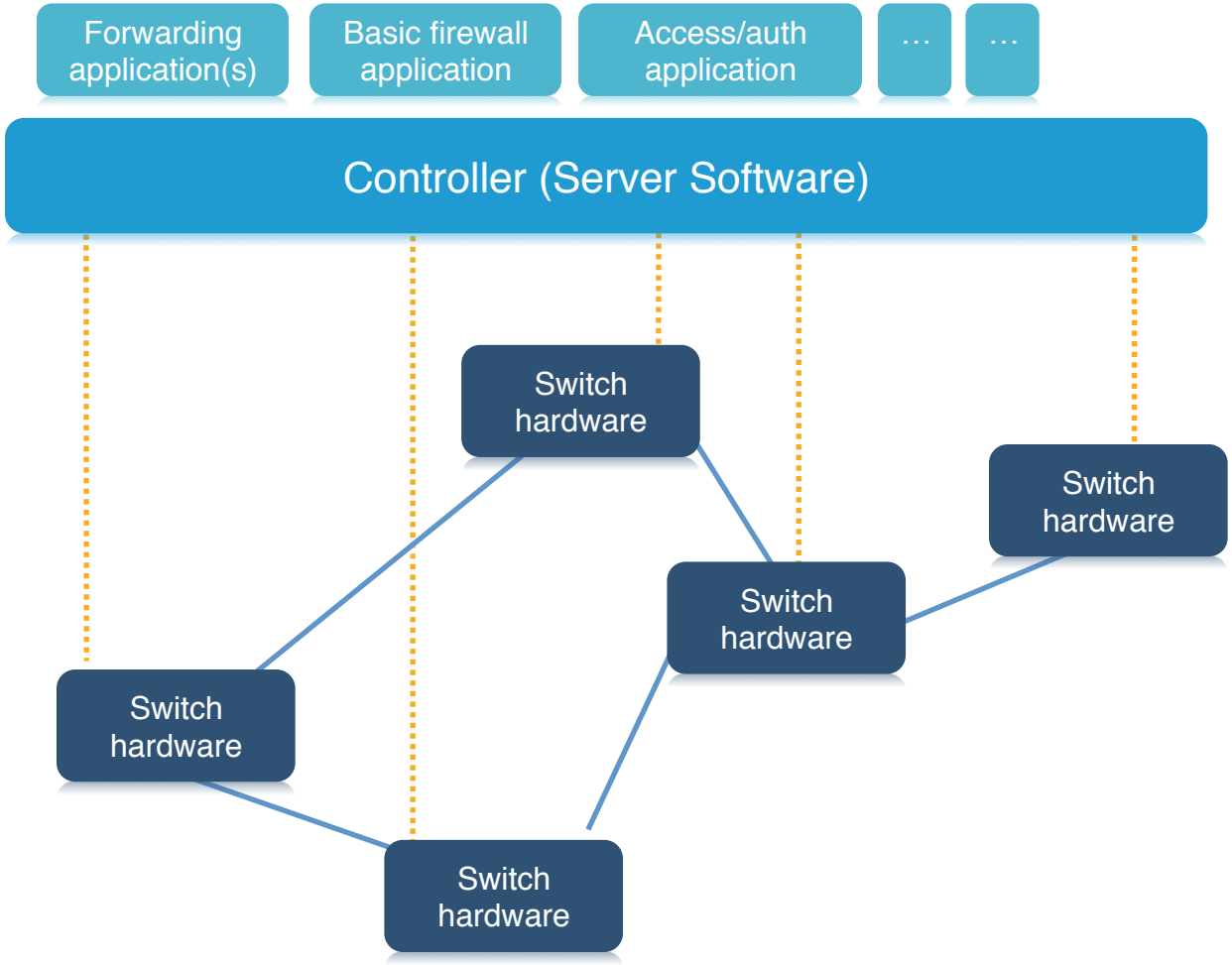
THE PROTOCOL

Minimal and powerful, like the x86 instruction set



THREE TIER MODEL OF OPENFLOW

A switch hardware layer, a controller layer and an application layer



3 Tier Model

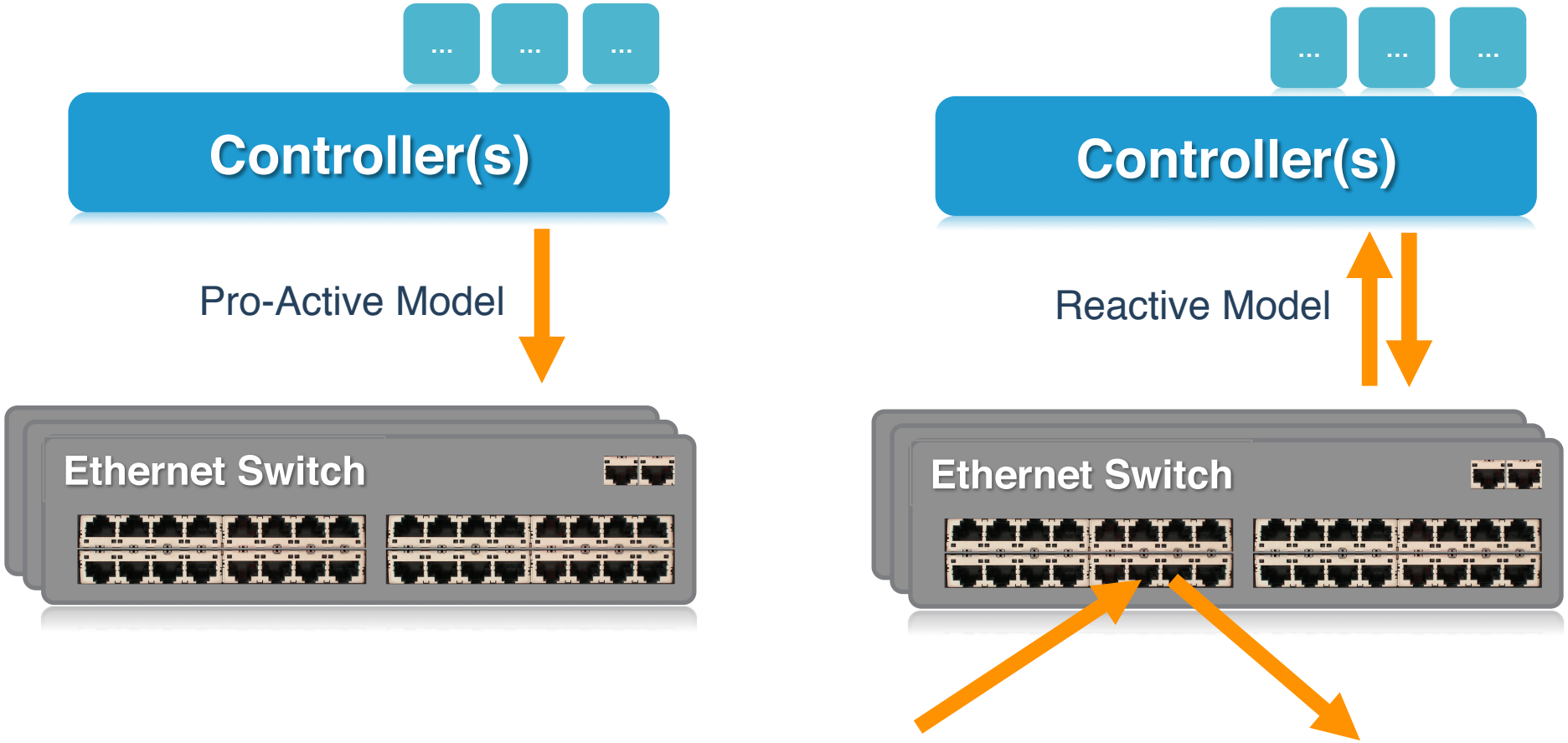
Many of the new networking functions can be built as server-side applications on this framework with line-rate performance.

Network functions are decoupled from underlying hardware (and location).

Integration of applications with external services (LDAP, Virtual Center) is vastly simplified.

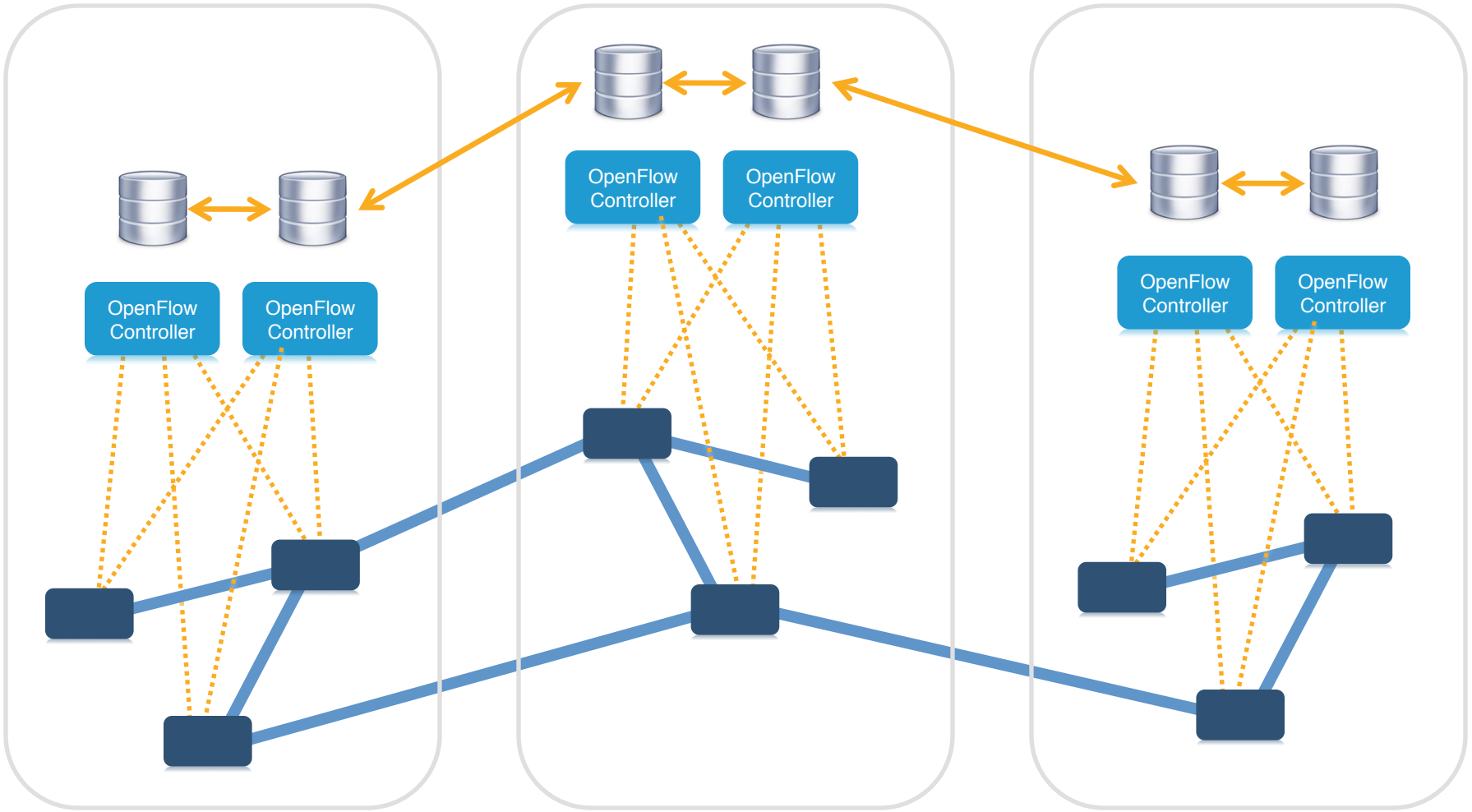
PRO-ACTIVE VERSUS REACTIVE MODELS

Performance versus a priori knowledge in app software + network design



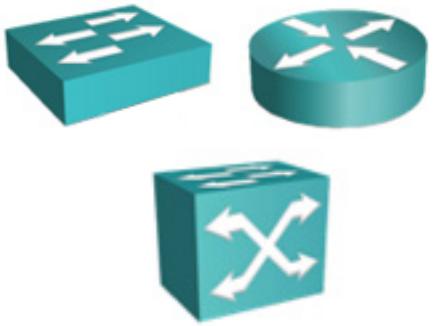
OPENFLOW IN PRACTICE

Still a distributed system, but less distributed (and less problematic)



(server)

(switches)

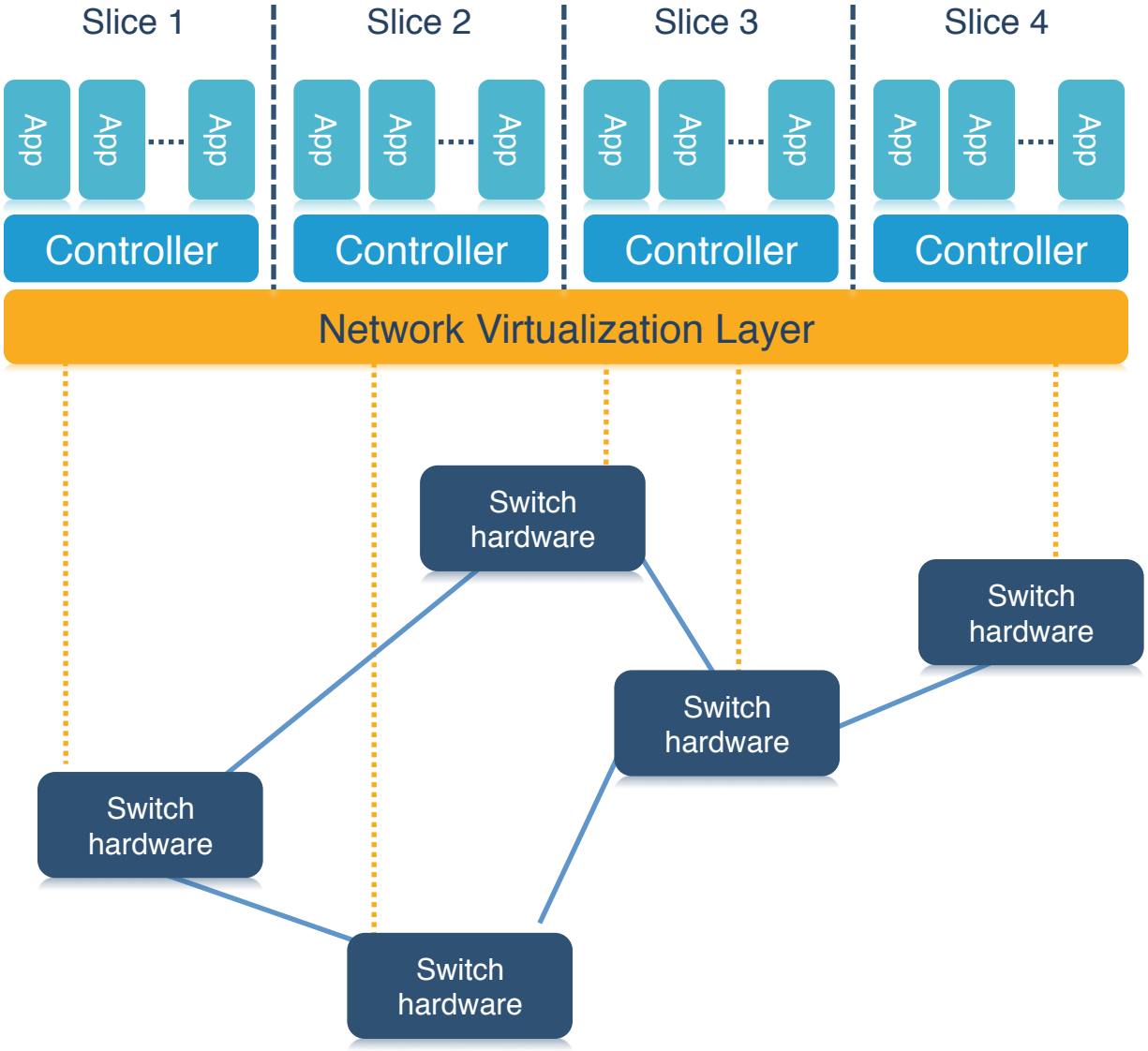


...

(switching, routing, firewalling, load balancing, spanning, and more…)

NETWORK SLICING

Virtualization of both data and control in the network



OpenFlow Protocol

A centralized virtualization layer allows to divide the network in “slices”.

Traffic and control of slices are isolated. Administrators for each slice can configure applications for the slice.

Slices can be very different, for example one slice can use a switch as an L2 switch, while another uses the same hardware concurrently as an L3 router.

NETWORK AUTOMATION

There is rarely a business case to justify solving these (hard) problems

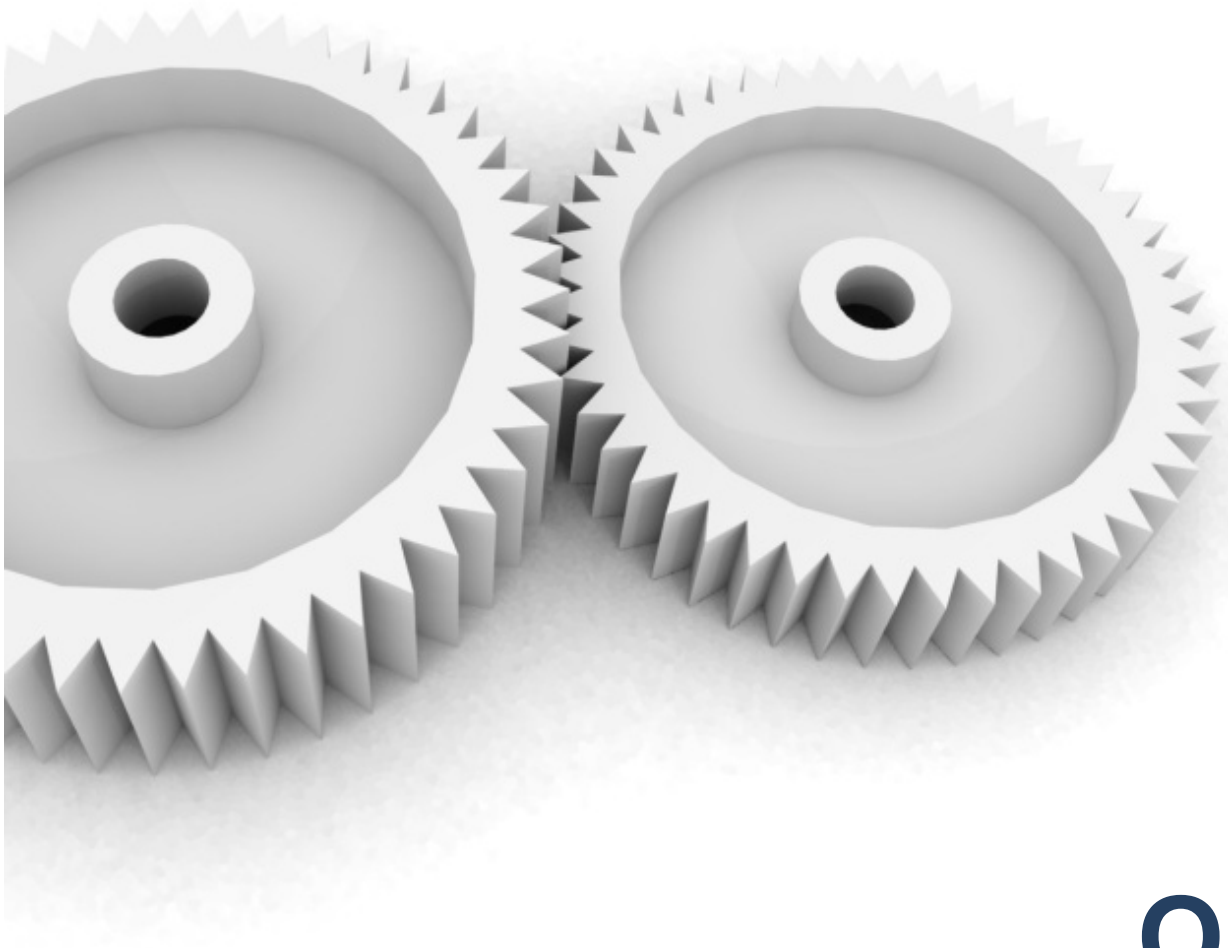
```
try {

  //update state of a network slice
  //double-check it is working as intended
  //now start to allow traffic

} catch(Exception e) {

  //roll back to previous known-good state

}
```



Questions

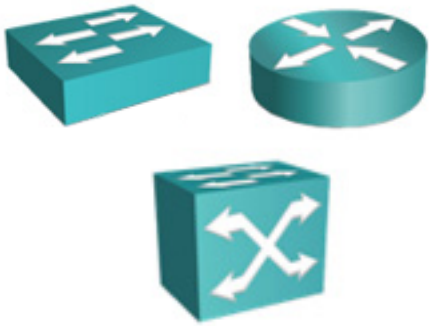
EVERY SEGMENT SEES SOMETHING DIFFERENT

OpenFlow has many potential propositions

	Service Provider WAN	Campus LAN	IaaS Provider	Enterprise Datacenter
Virtualization	Multi-tenancy	Multi-team	Delegated admin	The “new stackable”
Advanced Forwarding	Fat Trees for Big Data apps	Larger L2 domains, VM mobility	No more spanning tree failures	No more spanning tree failures
Programmability	Integration with proprietary systems	Vendor choice		

(server)

(switches)



...

(switching, routing, firewalling, load balancing, spanning, and more…)