

# OpenFlow Tutorial

## OFELIA Summer School

### Nov 8, 2011

Srini Seetharaman, Paul Weissmann  
Deutsche Telekom Innovation Laboratories

# Welcome! Getting Started

- Copy to your hard disk from Flash Drive/:
  - Corresponding files (terminal, X server, Vmware) for your platform (Win/Mac/Linux)
  - OpenFlowTutorial-20110131.vmwarevm.zip
- unzip OpenFlowTutorial-20110131.vmwarevm.zip
  - Instructions:  
<http://www.openflow.org/wk/index.php/OpenFlowTutorialOFELIA2011>
- You may also copy the routeflow.tar.gz file for using in tomorrow's tutorial

# Introductions

Your info:

- Name
- Affiliation
- Interest in OF
- Tell about your work

# Agenda

Time	Description
9:00-9:45	Introduction How OpenFlow works, Potential, Limitations, SDN, ONF
9:45-10:30	Begin <b>Hands-on Portion</b> (learn tools, self-exploration, mininet, NOX-based switch)
10:30-11:00	Intro to Slicing, FlowVisor, GENI OpenFlow stack
11:00-11:20	Continue <b>Hands-on Portion</b> (Slice using FlowVisor)
11:20-11:30	Wrap-up, Q&A, Discussions

# Goals of this Tutorial

- By the end, everyone should know:
  - **what** OpenFlow is
  - **how** it's used and how *you* can use it
  - **where** it's going
  - how OpenFlow compares to other platforms
  - how OpenFlow fits in the Software-Defined Networking (SDN) spectrum
- Present a useful mix of hands-on and lecture-based content
- **Have fun**



# Why OpenFlow?



# The Ossified Network

Routing, management, mobility management, access control, VPNs, ...



Million of lines  
of source code

5400 RFCs

Barrier to entry

Billions of gates

Bloated

Power Hungry

Many complex functions baked into the infrastructure

*OSPF, BGP, multicast, differentiated services,  
Traffic Engineering, NAT, firewalls, MPLS, redundant layers, ...*

An industry with a “mainframe-mentality”, reluctant to change

# Research Stagnation

- Lots of *deployed* innovation in other areas
  - OS: filesystems, schedulers, virtualization
  - DS: DHTs, CDNs, MapReduce
  - Compilers: JITs, vectorization
- Networks are largely the same as years ago
  - Ethernet, IP, WiFi
- Rate of change of the network seems slower in comparison
  - Need better tools and abstractions to demonstrate and deploy



# Closed Systems (Vendor Hardware)

- Stuck with interfaces (CLI, SNMP, etc)
- Hard to meaningfully collaborate
- Vendors starting to open up, but not usefully
- Need a fully open system – a Linux equivalent

# Open Systems

	Performance Fidelity	Scale	Real User Traffic?	Complexity	Open
Simulation	medium	medium	<b>no</b>	medium	yes
Emulation	medium	<b>low</b>	<b>no</b>	medium	yes
Software Switches	<b>poor</b>	<b>low</b>	yes	medium	yes
NetFPGA	high	<b>low</b>	yes	<b>high</b>	yes
Network Processors	high	medium	yes	<b>high</b>	yes
Vendor Switches	high	high	yes	low	<b>no</b>

gap in the tool space  
**none** have all the desired attributes!

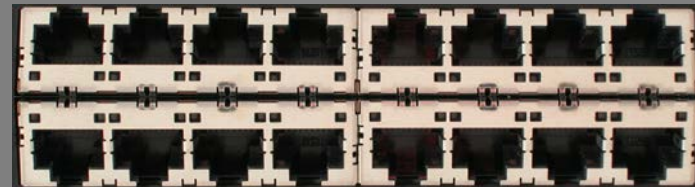
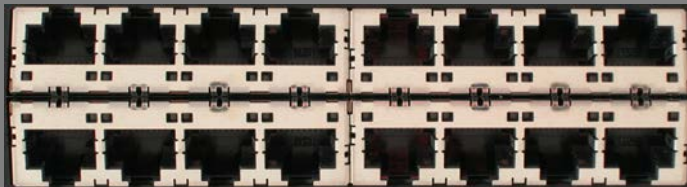
# OpenFlow: a pragmatic compromise

- + Speed, scale, fidelity of vendor hardware
- + Flexibility and control of software and simulation
- Vendors don't need to expose implementation
- Leverages hardware inside most switches today (ACL tables)



# How does OpenFlow work?

# Ethernet Switch



**Control Path (Software)**

---

**Data Path (Hardware)**

# OpenFlow Controller

OpenFlow Protocol (SSL/TCP)



Control Path

OpenFlow

---

Data Path (Hardware)

# OpenFlow Example

Controller

Software  
Layer

OpenFlow Client

Flow Table

MAC src	MAC dst	IP Src	IP Dst	TCP sport	TCP dport	Action
*	*	*	5.6.7.8	*	*	port 1

Hardware  
Layer

port 1

port 2

port 3

port 4



5.6.7.8



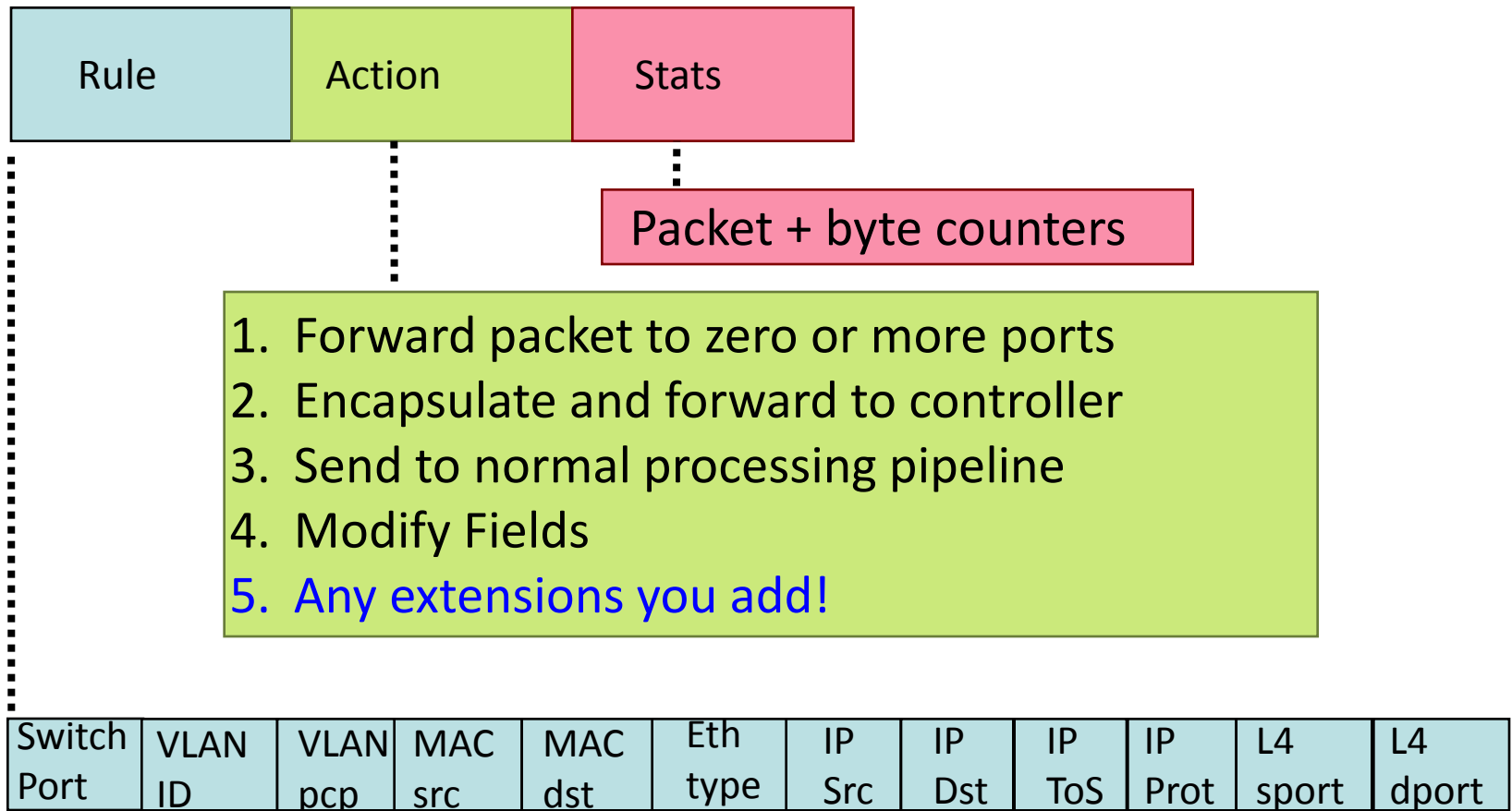
1.2.3.4





# OpenFlow Basics

## Flow Table Entries



+ mask what fields to match

# Examples

## Switching

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	00:1f:..	*	*	*	*	*	*	*	port6

## Flow Switching

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
port3	00:20..	00:1f..	0800	vlan1	1.2.3.4	5.6.7.8	4	17264	80	port6

## Firewall

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	*	*	*	22	drop

# Examples

## Routing

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	5.6.7.8	*	*	*	port6

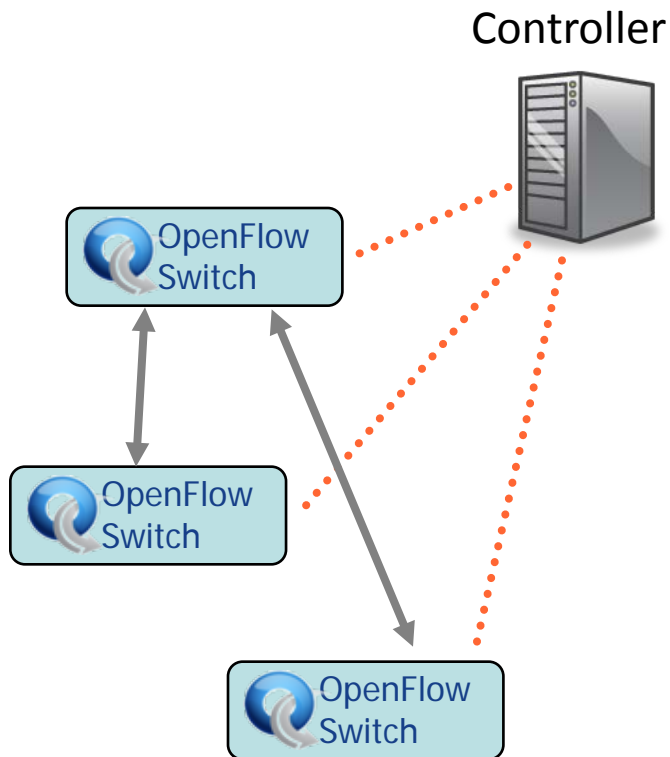
## VLAN Switching

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	00:1f..	*	vlan1	*	*	*	*	*	port6, port7, port9

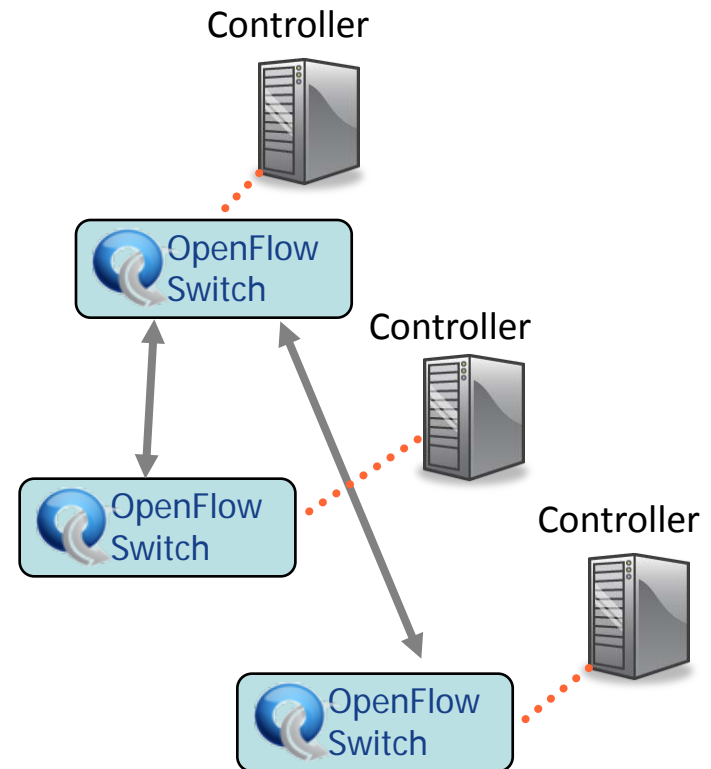
# Centralized vs Distributed Control

Both models are possible with OpenFlow

## Centralized Control



## Distributed Control



# Flow Routing vs. Aggregation

Both models are possible with OpenFlow

## Flow-Based

- Every flow is individually set up by controller
- Exact-match flow entries
- Flow table contains one entry per flow
- Good for fine grain control, e.g. campus networks

## Aggregated

- One flow entry covers large groups of flows
- Wildcard flow entries
- Flow table contains one entry per category of flows
- Good for large number of flows, e.g. backbone

# Reactive vs. Proactive (pre-populated)

Both models are possible with OpenFlow

## Reactive

- First packet of flow triggers controller to insert flow entries
- Efficient use of flow table
- Every flow incurs small additional flow setup time
- If control connection lost, switch has limited utility

## Proactive

- Controller pre-populates flow table in switch
- Zero additional flow setup time
- Loss of control connection does not disrupt traffic
- Essentially requires aggregated (wildcard) rules

# Usage examples

- Alice's code:
    - Simple learning switch
    - Per Flow switching
    - Network access control/firewall
    - Static “VLANs”
    - Her own new routing protocol: unicast, multicast, multipath
    - Home network manager
    - Packet processor (in controller)
    - IPvAlice
- VM migration
  - Server Load balancing
  - Mobility manager
  - Power management
  - Network monitoring and visualization
  - Network debugging
  - Network slicing

... and much more you can create!

# Quiz Time

- How do I provide control connectivity? Is it really clean slate?
- Why aren't users complaining about time to setup flows over OpenFlow? (Hint: What is the predominant traffic today?)
- How scalable is OpenFlow? How does one scale deployments?
- Considering switch CPU is the major limit, how can one take down an OpenFlow network?



# What can you not do with OpenFlow ver1.1

- Non-flow-based (per-packet) networking
  - ex. Per-packet next-hop selection (in wireless mesh)
  - yes, this is a fundamental limitation
  - BUT OpenFlow can provide the plumbing to connect these systems
- Use all tables on switch chips
  - yes, a major limitation (cross-product issue)
  - BUT an upcoming OF version will expose these

# What can you not do with OpenFlow ver1.1

- New forwarding primitives
  - BUT provides a nice way to integrate them through extensions
- New packet formats/field definitions
  - BUT a generalized OpenFlow (2.0) is on the horizon
- Optical Circuits
  - BUT efforts underway to apply OpenFlow model to circuits
- Low-setup-time individual flows
  - BUT can push down flows proactively to avoid delays

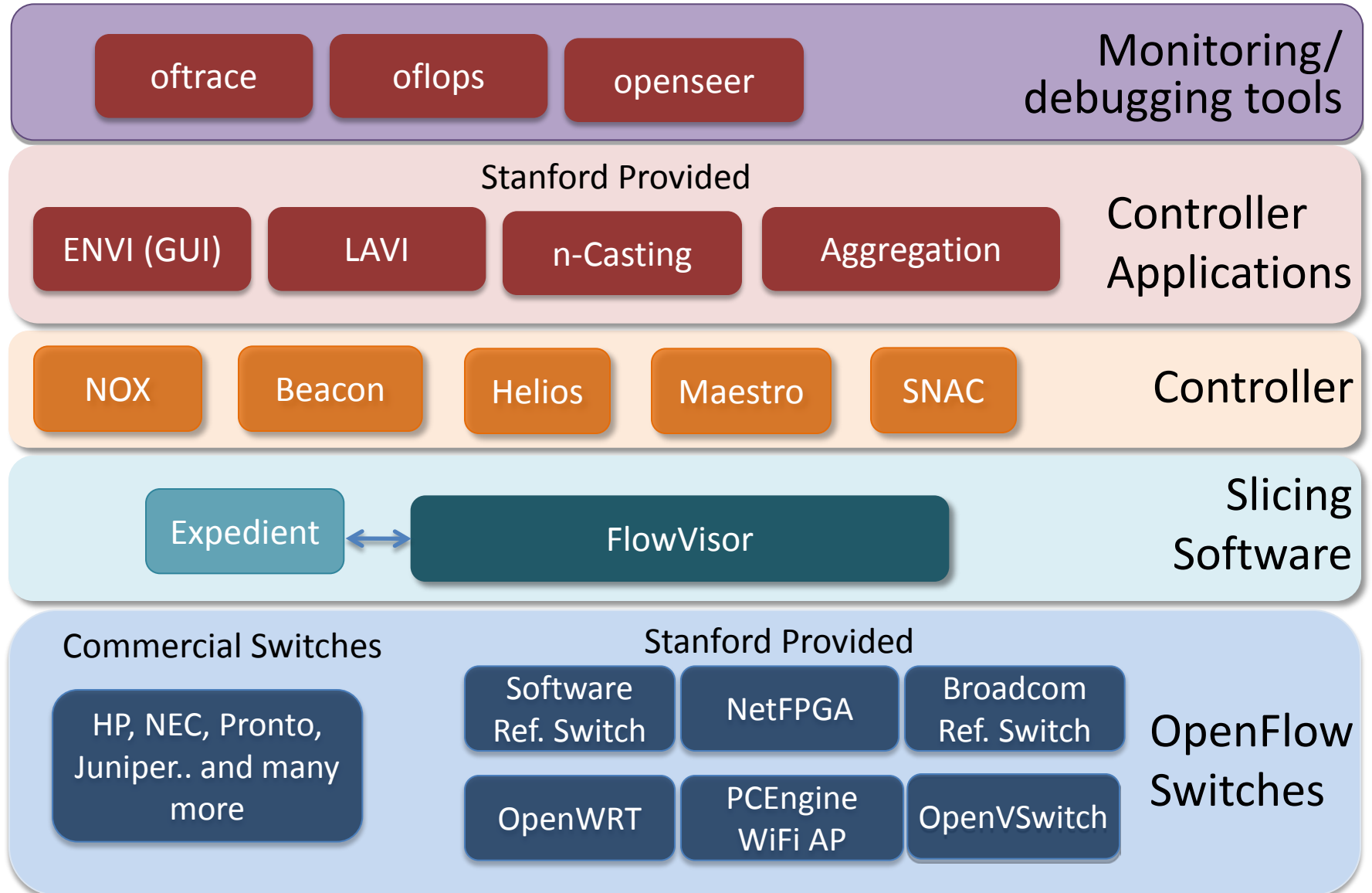
# Where it's going

- OF v1.1: Extensions for WAN, spring 2011
  - multiple tables: leverage additional tables
  - tags and tunnels
  - multipath forwarding
- OF v2+
  - generalized matching and actions: an “instruction set” for networking






# OpenFlow Implementations (Switch and Controller)

# OpenFlow building blocks




# Commercial Switch Vendors

Model	Virtualize	Notes	
HP Procurve 5400zl or 6600	1 OF instance per VLAN	<ul style="list-style-type: none"> <li>-LACP, VLAN and STP processing before OpenFlow</li> <li>-Wildcard rules or non-IP pkts processed in s/w</li> <li>-Header rewriting in s/w</li> <li>-CPU protects mgmt during loop</li> </ul>	
NEC IP8800	1 OF instance per VLAN	<ul style="list-style-type: none"> <li>-OpenFlow takes precedence</li> <li>-Most actions processed in hardware</li> <li>-MAC header rewriting in h/w</li> </ul>	
Pronto 3290 or 3780 with Pica8 or Indigo firmware	1 OF instance per switch	<ul style="list-style-type: none"> <li>-No legacy protocols (like VLAN and STP)</li> <li>-Most actions processed in hardware</li> <li>-MAC header rewriting in h/w</li> </ul>	

# Open-source controllers

Vendor	Notes
Nicira's NOX	<ul style="list-style-type: none"><li>•GPL</li><li>•C++ and Python</li></ul>
SNAC	<ul style="list-style-type: none"><li>•GPL</li><li>•Code based on NOX0.4</li><li>•Enterprise network</li><li>•C++, Python and Javascript</li><li>•Currently used by campuses</li></ul>

Vendor	Notes
Stanford's Beacon	<ul style="list-style-type: none"><li>•BSD-like license</li><li>•Java-based</li></ul>
Maestro (from Rice Univ)	<ul style="list-style-type: none"><li>•GPL</li><li>•Based on Java</li></ul>
NEC's Trema	<ul style="list-style-type: none"><li>•Open-source</li><li>•Written in C and Ruby</li><li>•Included test harness</li></ul>

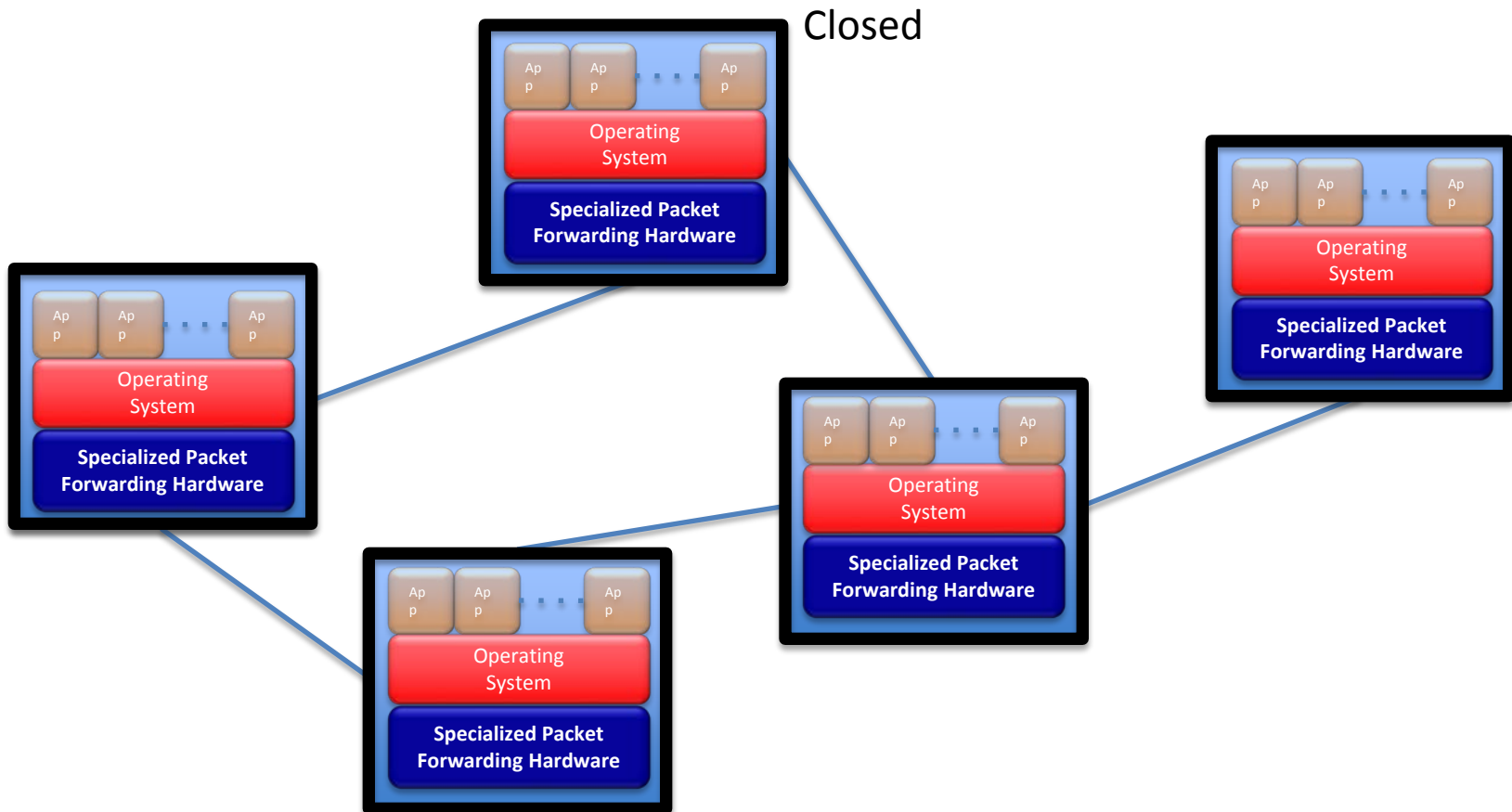
The logo consists of a blue circular arrow pointing clockwise, with a grey arrow pointing counter-clockwise, creating a circular flow effect.

# Software-Defined Networking (SDN)

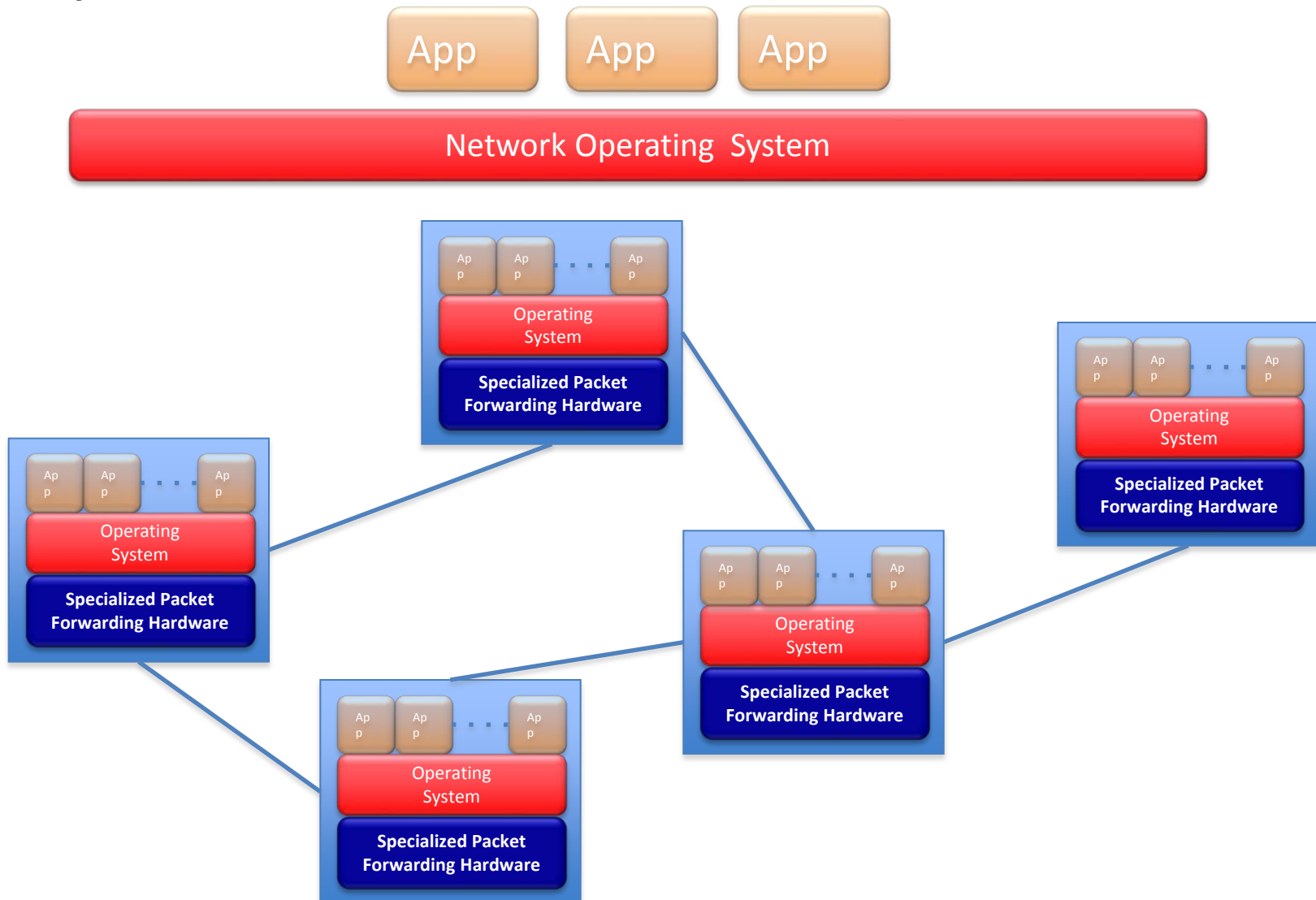


# Current Internet

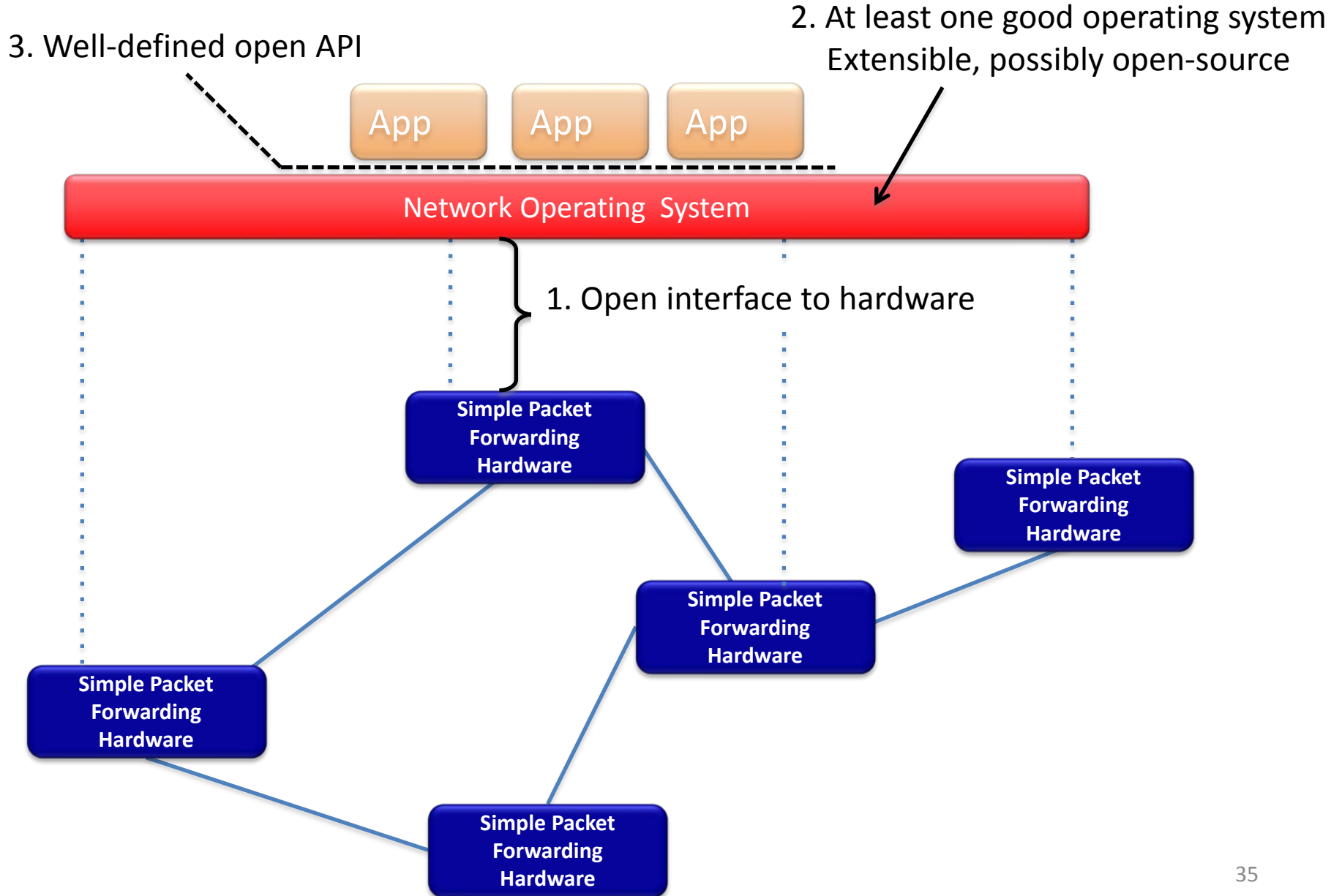
## Closed to Innovations in the Infrastructure



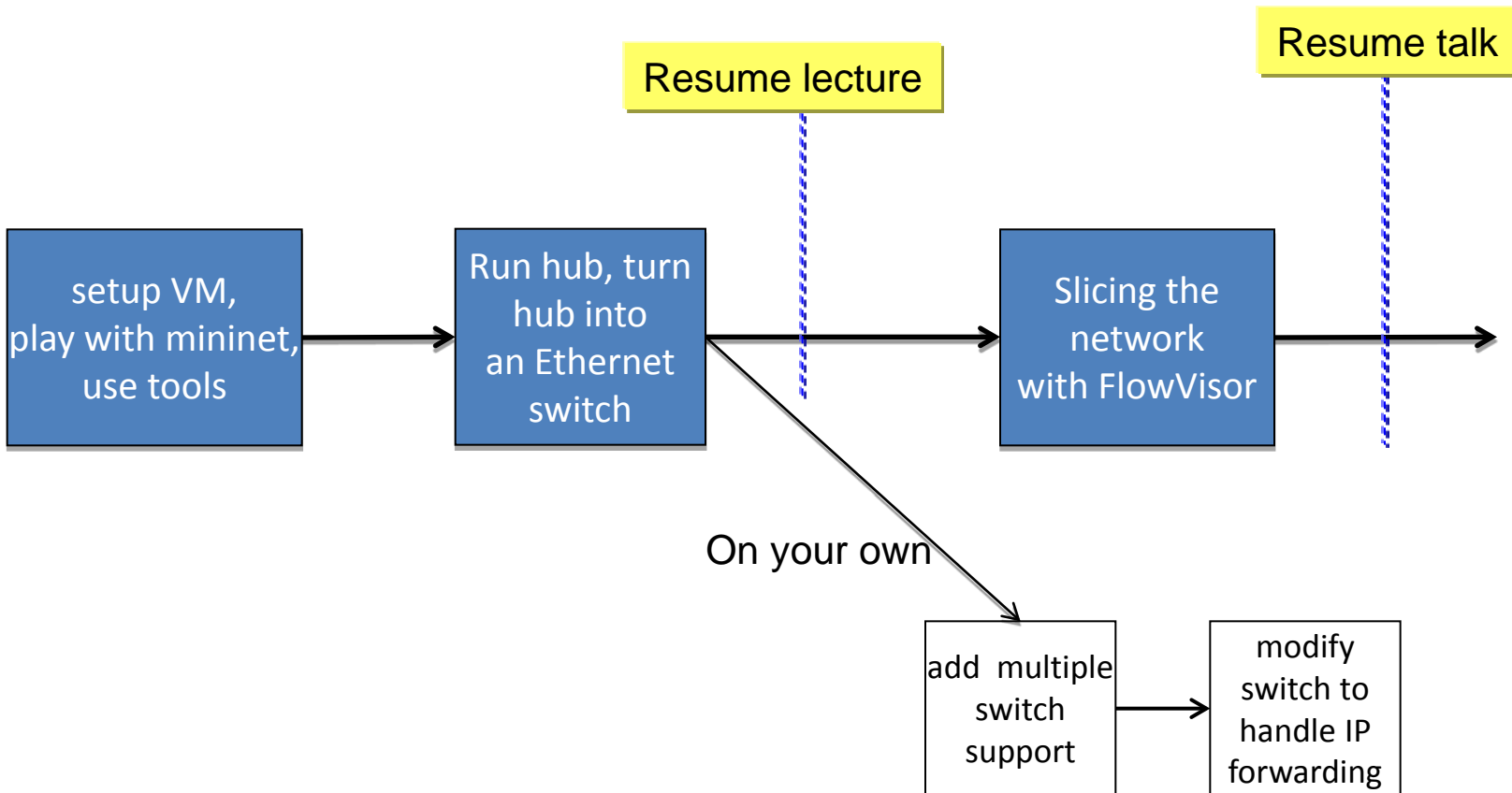
# “Software Defined Networking” approach to open it



# The “Software-defined Network”



# Hands-on Session Flow

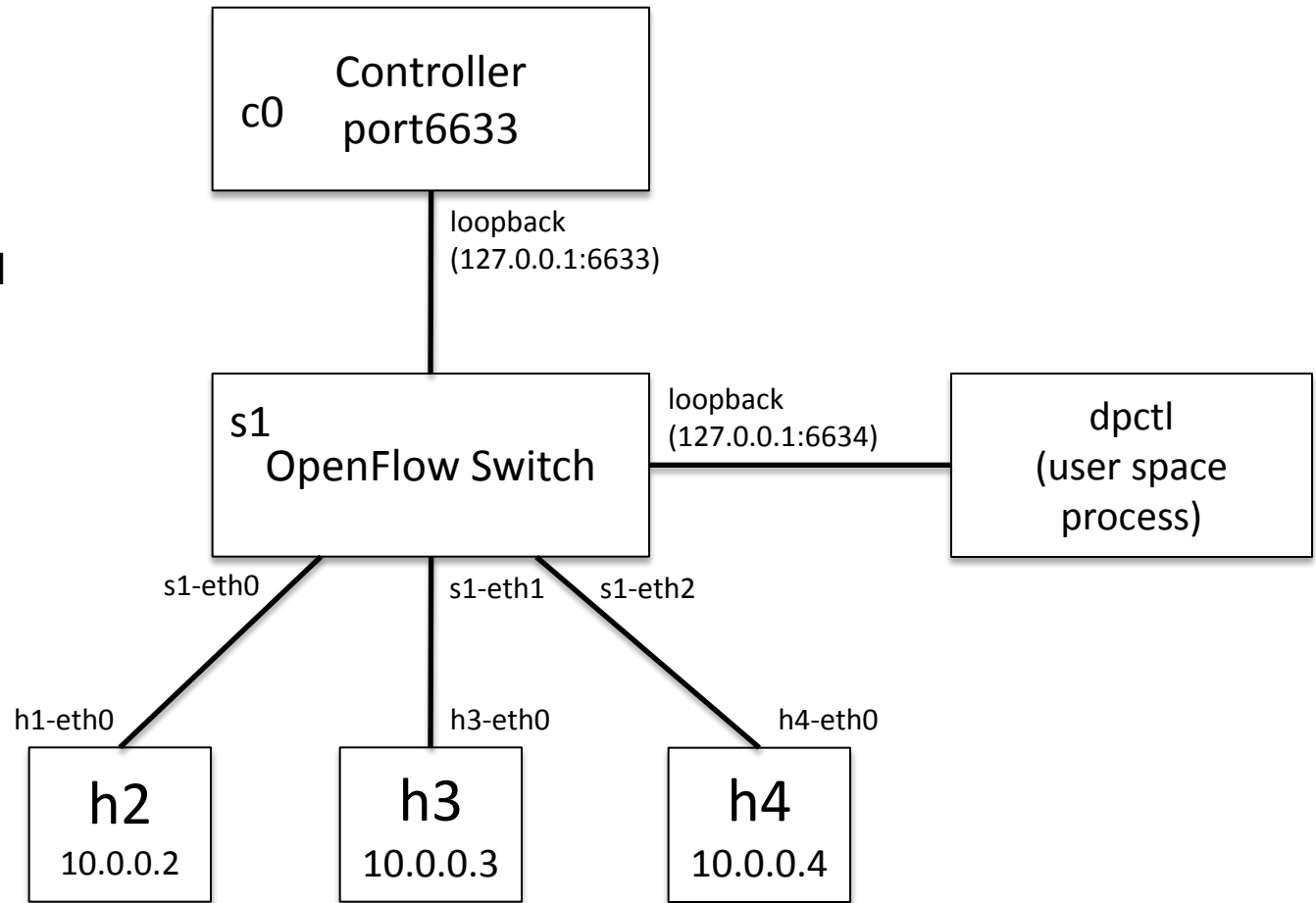


# Stuff you'll use

- NOX
- Reference Controller/Switch
- OpenvSwitch
- Mininet
- cbench
- flowvisor
  
- iperf
- tcpdump
- Wireshark

# Tutorial Setup

OpenFlow Tutorial  
3hosts-1switch  
topology



virtual hosts



# Hands-on Tutorial

Next presentation starts at **11:00**

[www.openflow.org/wk/index.php/OpenFlowTutorialOFELIA2011](http://www.openflow.org/wk/index.php/OpenFlowTutorialOFELIA2011)

—Next session will start with a quick review of slicing

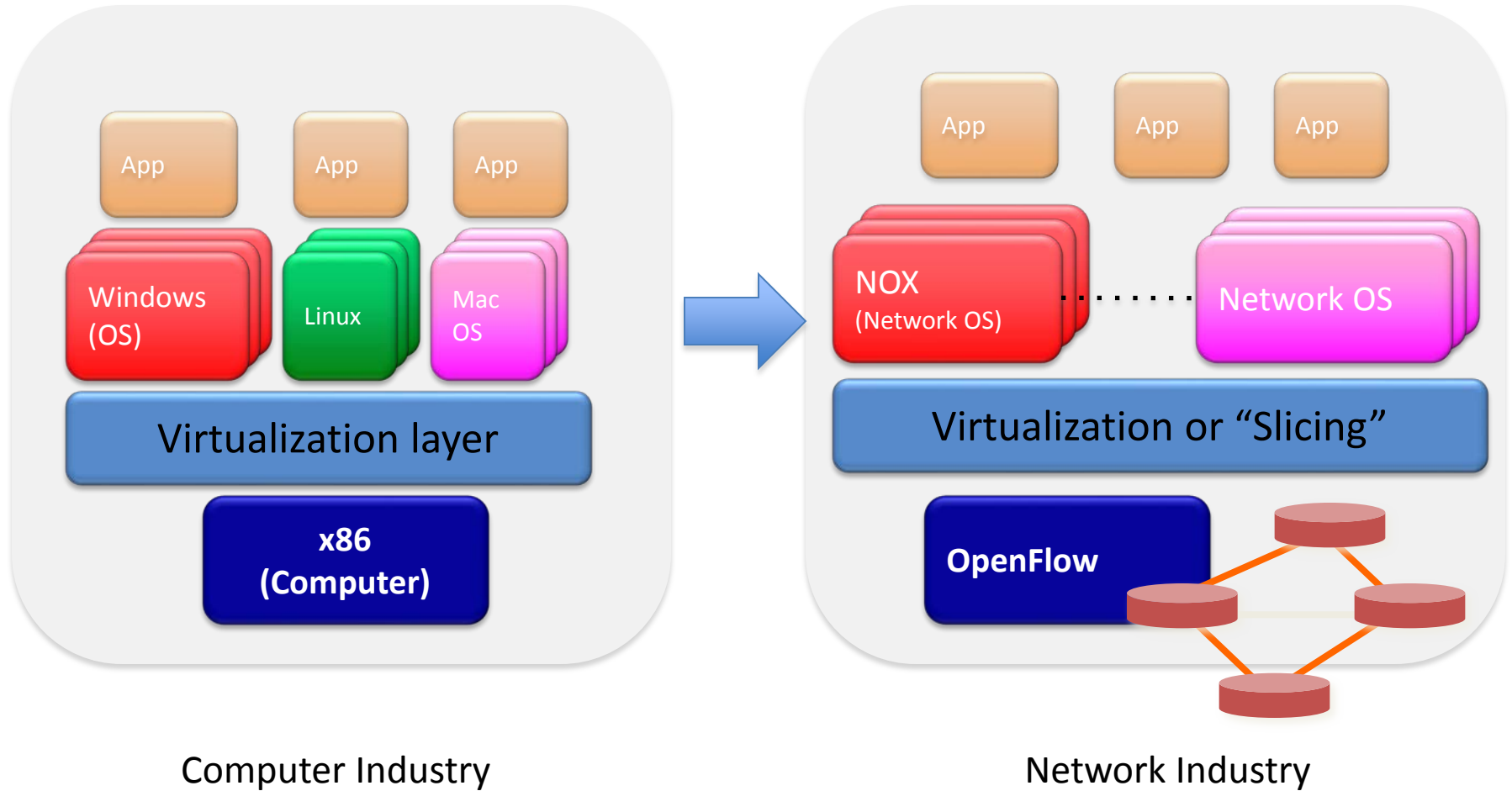
- VMWare Key (if needed):
  - 4J09J-3U2E5-58C32-08AK0-8DH42

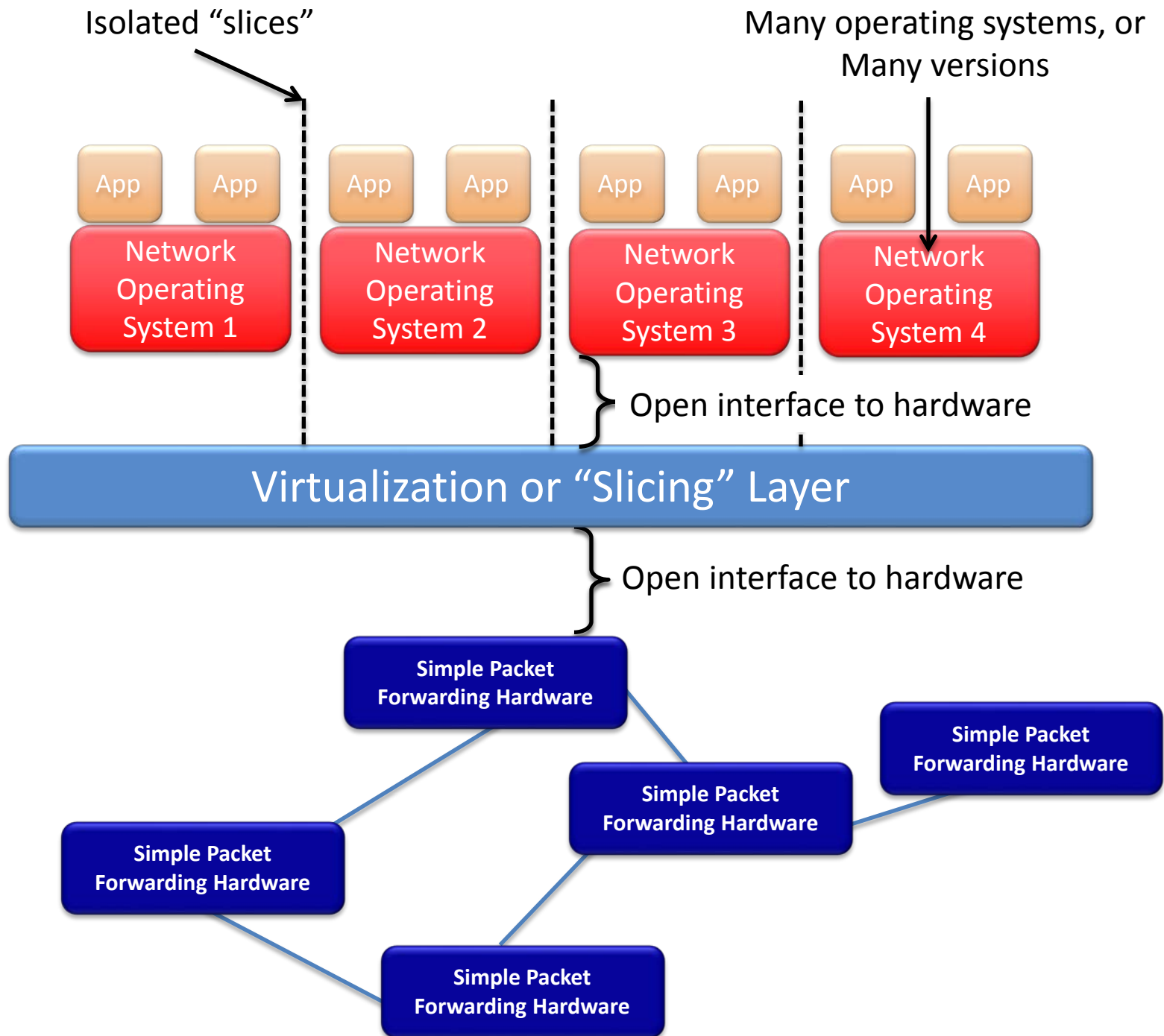


# Virtualizing OpenFlow



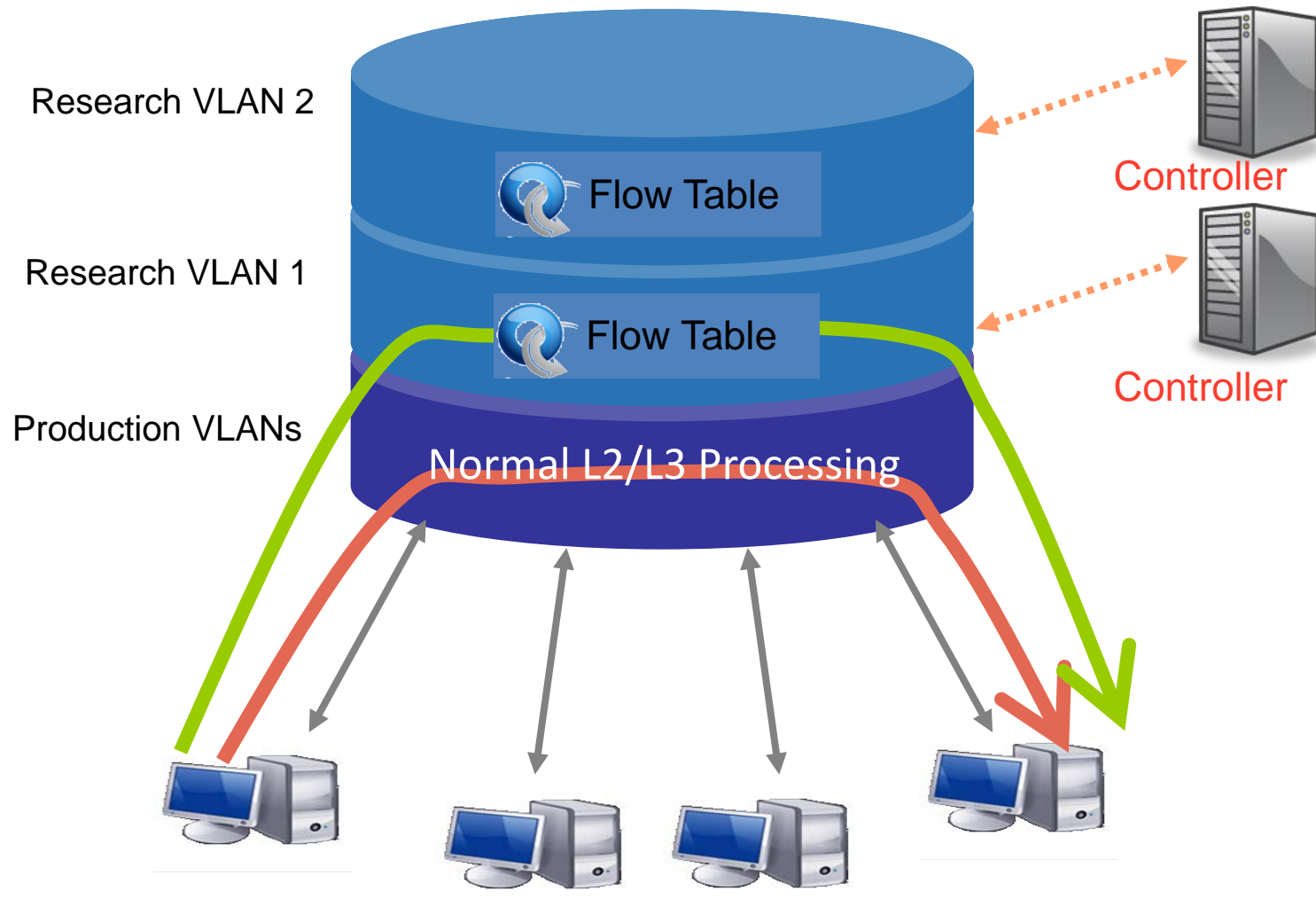
# Trend





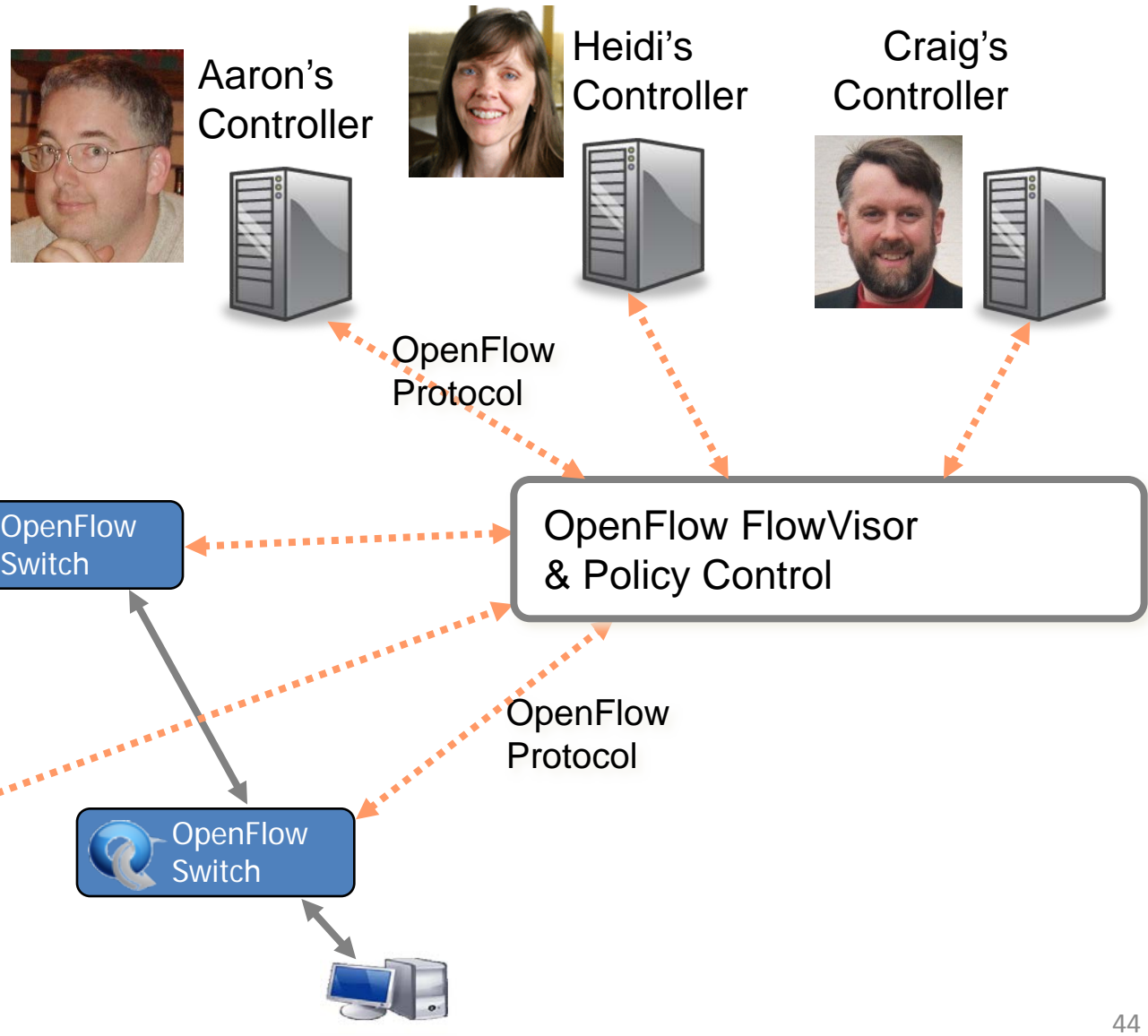
# Switch Based Virtualization

Exists for NEC, HP switches but not flexible enough



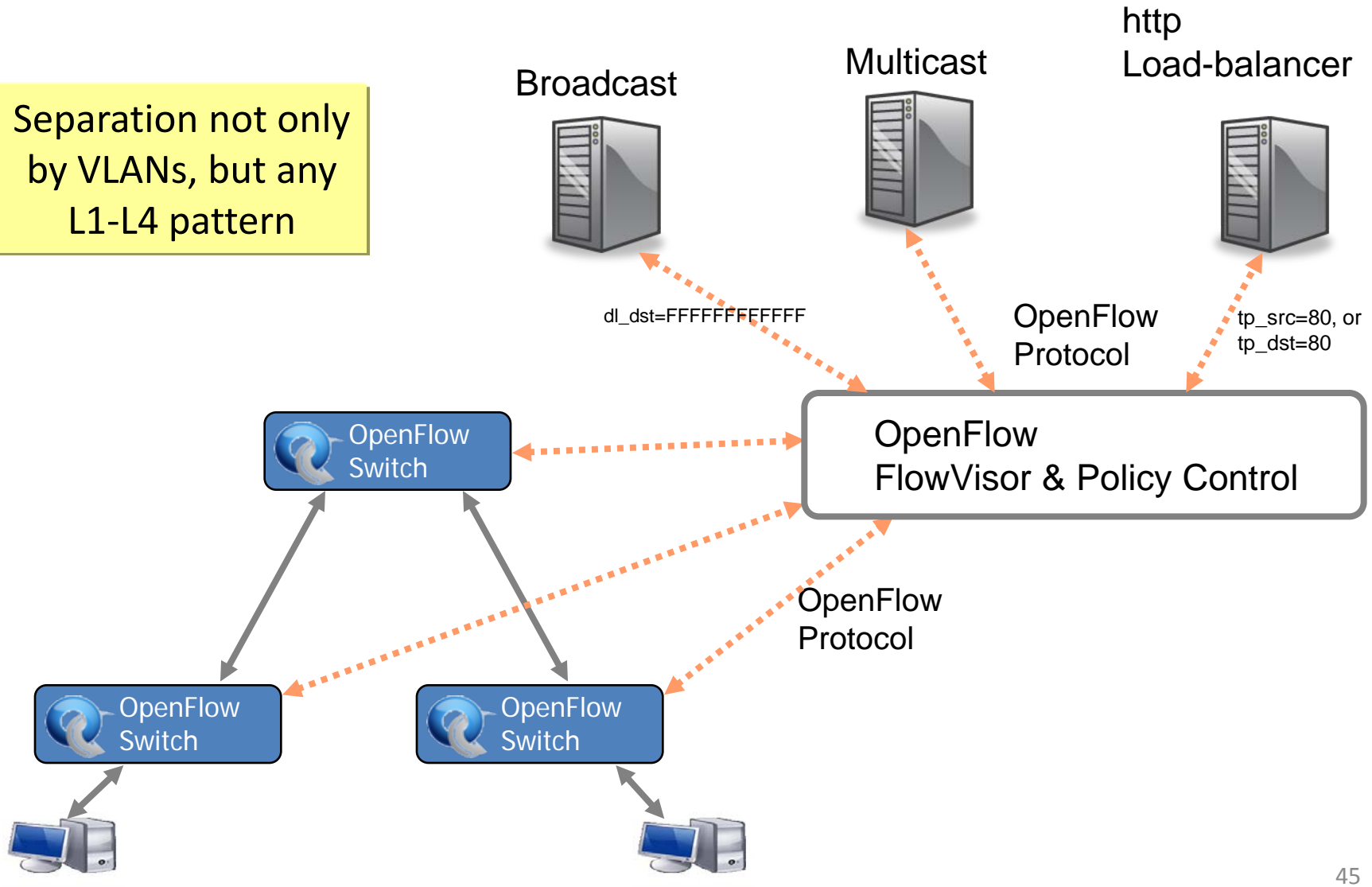
# FlowVisor-based Virtualization

Topology  
discovery is  
per slice

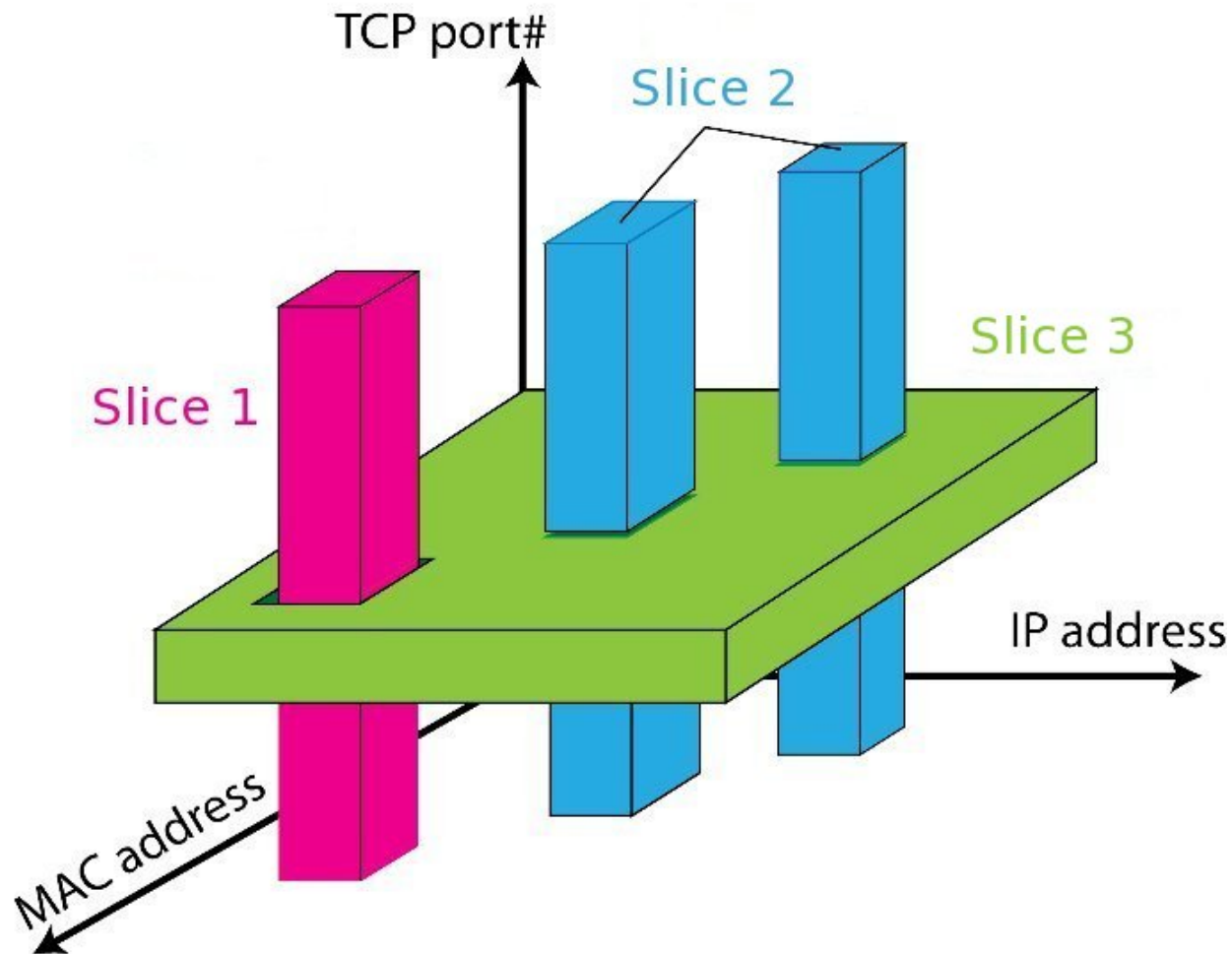


# FlowVisor-based Virtualization

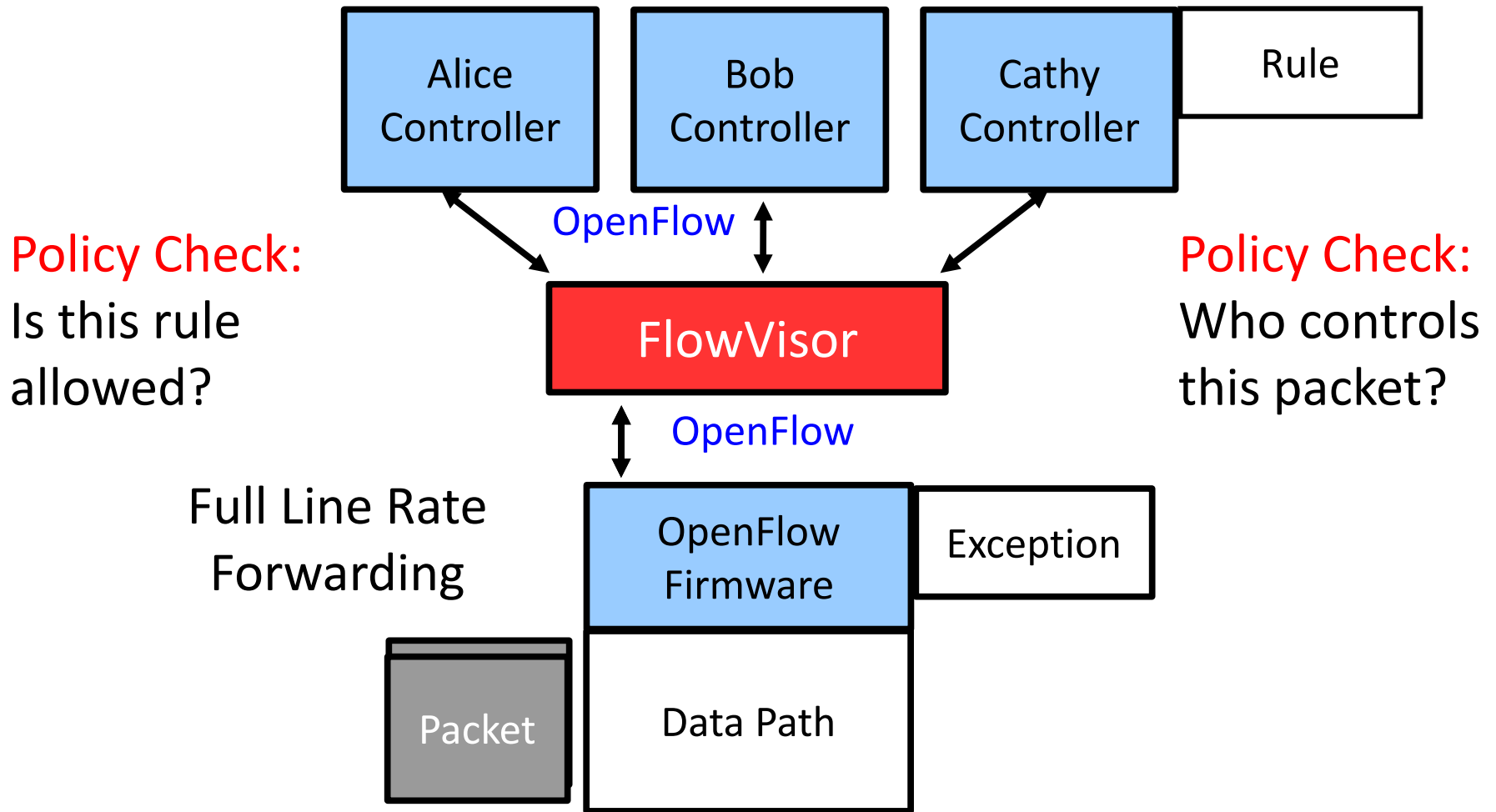
Separation not only  
by VLANs, but any  
L1-L4 pattern



# FlowSpace: Maps Packets to Slices



# FlowVisor Message Handling



# Use Case: New CDN - Turbo Coral ++

Basic Idea: Build a CDN where you control the entire network

- All traffic to or from Coral IP space controlled by Experimenter
- All other traffic controlled by default routing
- Topology is entire network
- End hosts are automatically added (no opt-in)

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport
-------------	---------	---------	----------	---------	--------	--------	---------	-----------	-----------



*	*	*	*	*	84.65.*	*	*	*	*
*	*	*	*	*	*	84.65.*	*	*	*



*	*	*	*	*	*	*	*	*	*
---	---	---	---	---	---	---	---	---	---



# Use Case: Aaron's IP

- A new layer 3 protocol
- Replaces IP
- Defined by a new Ether Type

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport
-------------	---------	---------	----------	---------	--------	--------	---------	-----------	-----------



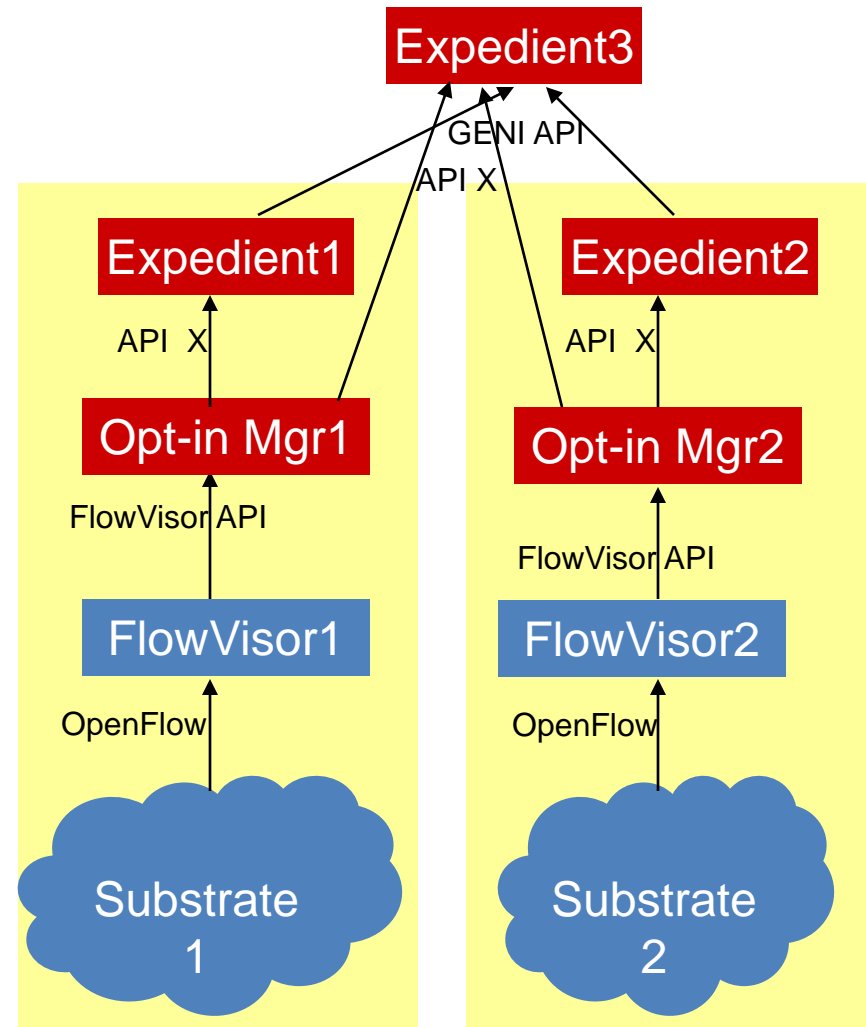
*	*	*	AaIP	*	*	*	*	*	*
---	---	---	------	---	---	---	---	---	---



*	*	*	!AaIP	*	*	*	*	*	*
---	---	---	-------	---	---	---	---	---	---

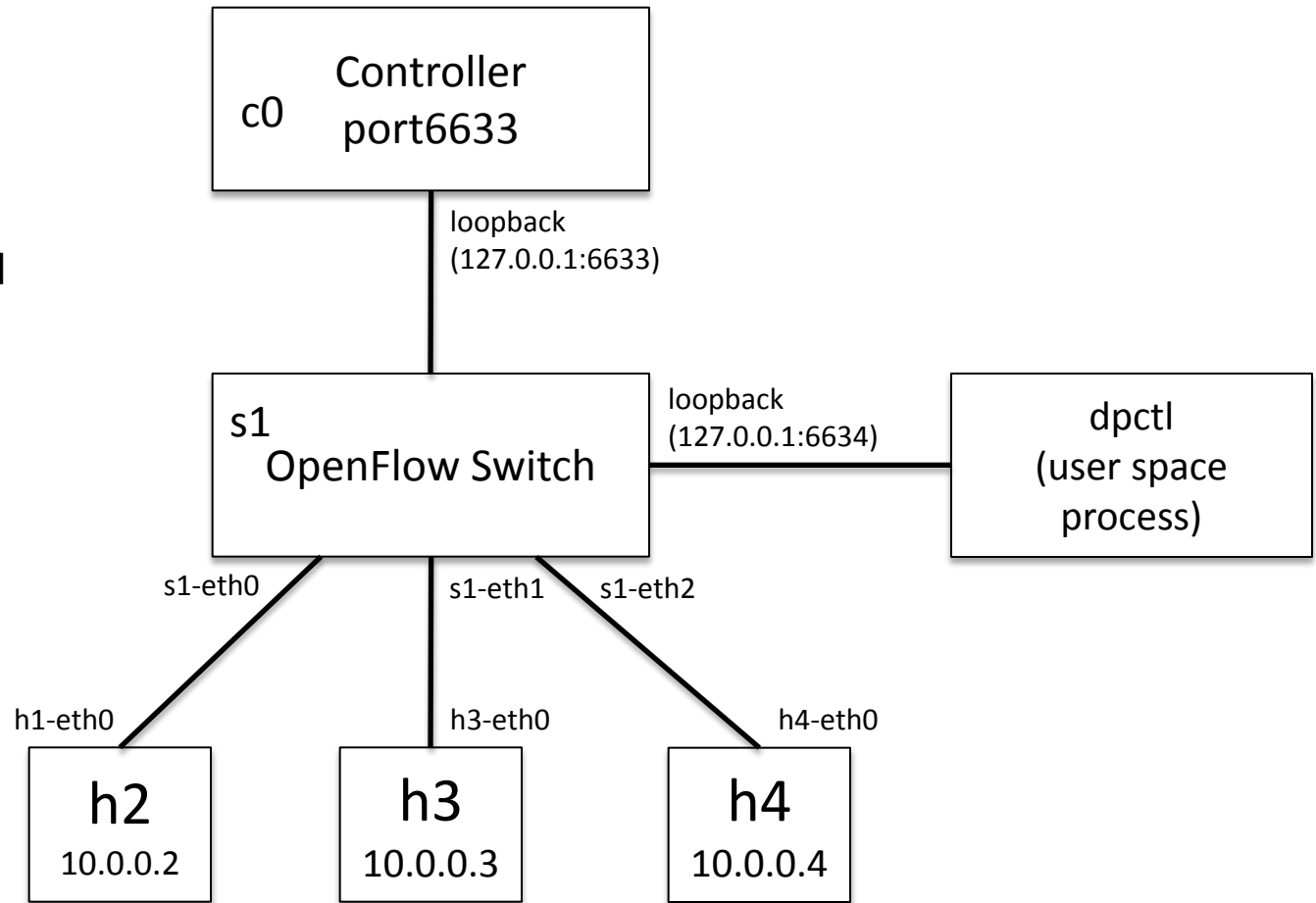
# GENI Integration

- FlowVisor
  - Slicing control
- Expedient
  - Experimenter's portal for slice management
- Opt-in Manager
  - Network admins' portal to approve/ deny expt requests for traffic



# Tutorial Setup so far

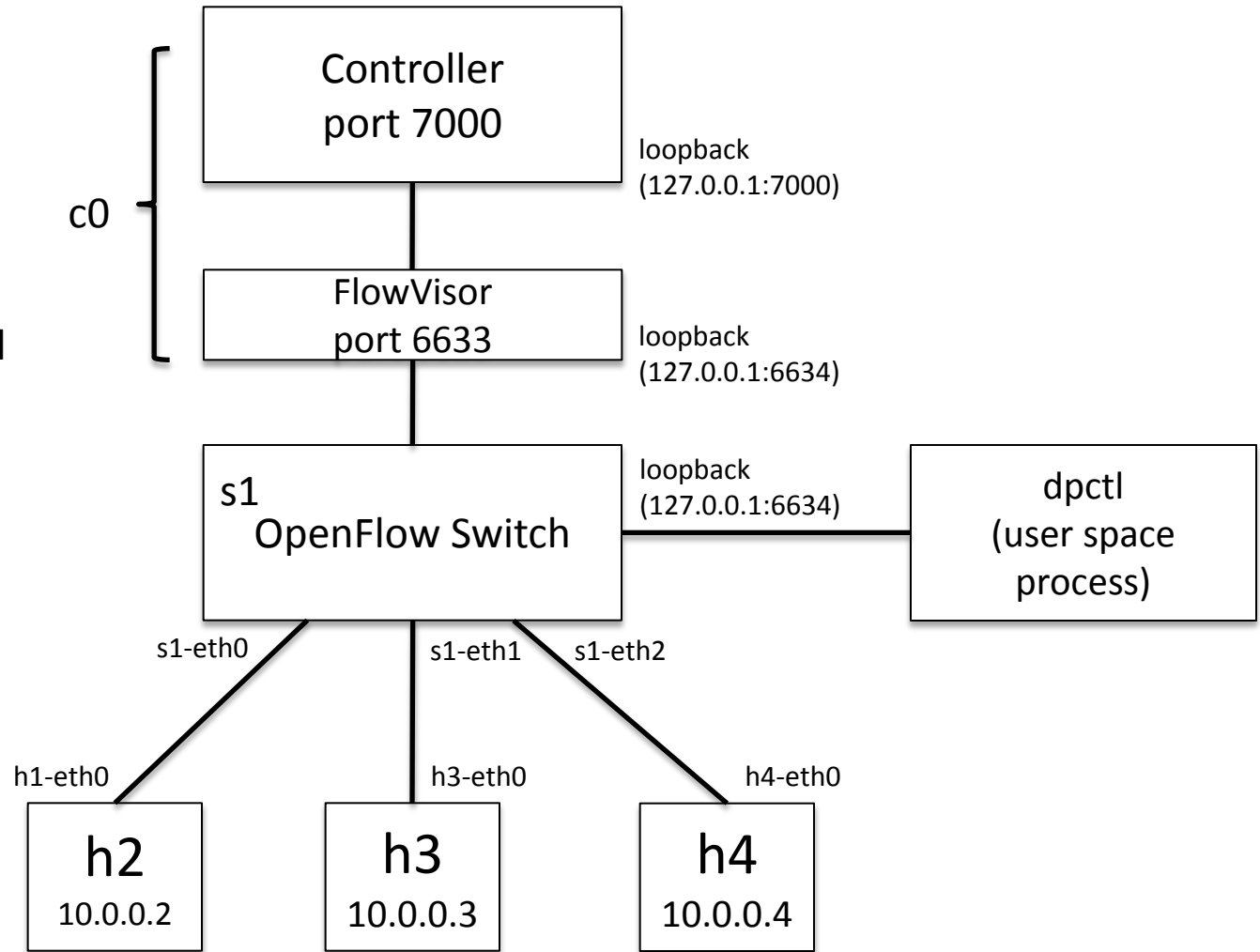
OpenFlow Tutorial  
3hosts-1switch  
topology



virtual hosts

# Virtualized Network Setup

OpenFlow Tutorial  
3hosts-1switch  
topology



virtual hosts

# FlowVisor Usage

- `man ./doc/flowvisor.8`
- `<flowvisor dir>/script/fvctl.sh`
- `listDevices`
  - list of all OpenFlow switches' datapath ID
- `getLinks`
  - list of all links (port # and datapath ID of both end switches)
- `createSlice`
  - creating slice (specifying controller's URL and slice name)
- `listSlices`
- `addFlowSpace`
  - add flow space to slices
- `listFlowSpace`
  - show current flow space

A blue circular icon with a grey arrow pointing downwards and to the right, partially overlapping the text.

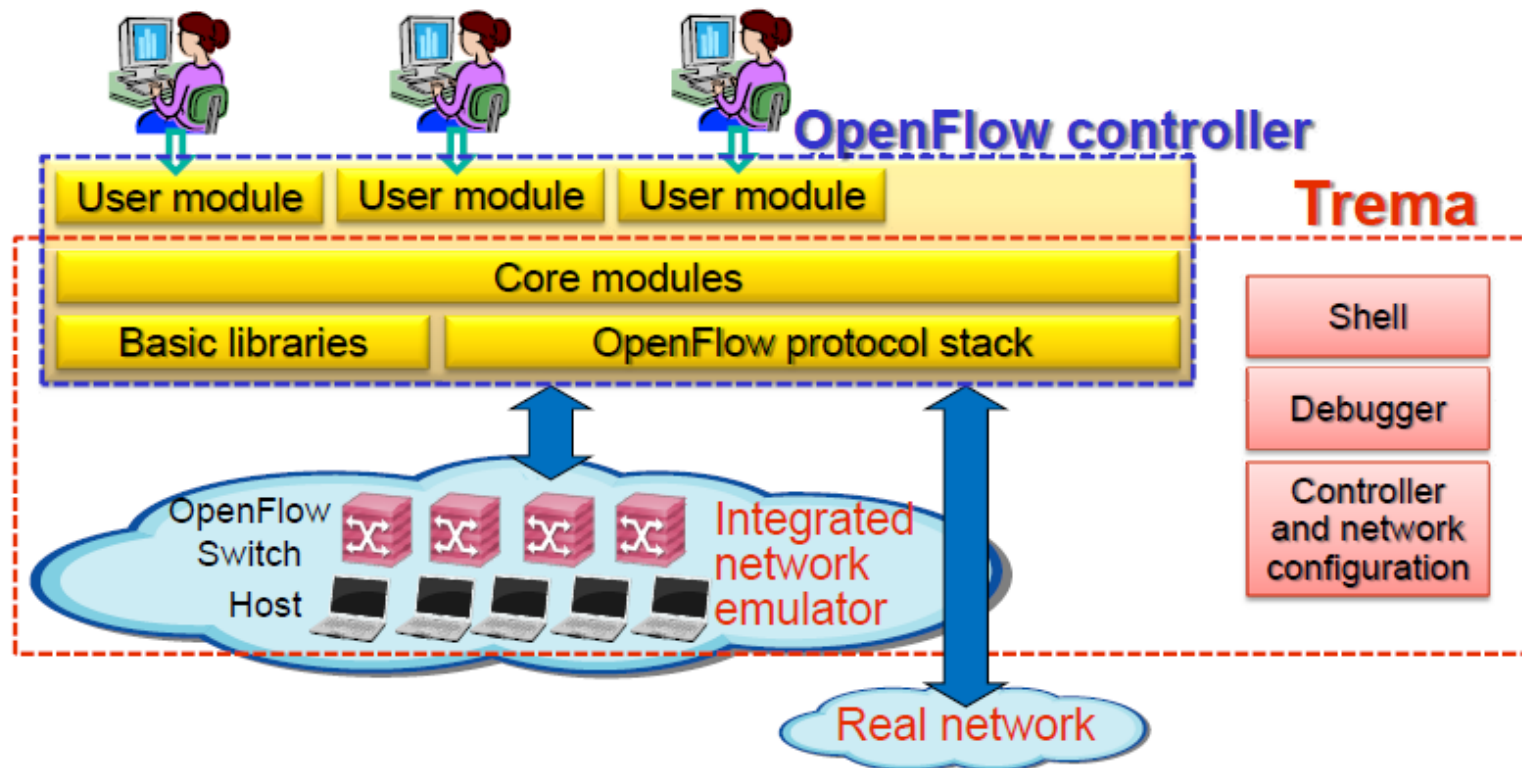
# Concluding Remarks

# Other controllers: Trema

## What is “Trema”

Empowered by Innovation **NEC**

- Trema @ <https://github.com/trema/trema>
- User modules @ <https://github.com/trema/apps>



# Other controllers: Beacon

- Check out:

<http://www.openflowhub.org/display/Beacon>

- Simple tutorial at:

[http://www.openflow.org/wk/index.php/OpenFlowTutorial#Controller Choice B: Beacon w.2F Java](http://www.openflow.org/wk/index.php/OpenFlowTutorial#Controller_Choice_B:_Beacon_w.2F_Java)



# Get involved!

- Ask and answer questions on mailing lists:
  - `openflow-discuss`
  - `openflow-spec`
- Sign up for the ONF lists, if you are part of a member institution
- Release open-source applications and controller platforms!

Are you innovating in your network?