# Software Defined Networking for big-data science

Eric Pouyoul

Chin Guok

**Inder Monga (presenting)**

SRS presentation

November 15th, Supercomputing 2012

U.S. DEPARTMENT OF **ENERGY**
Office of Science

**BERKELEY LAB**

# Acknowledgements

Many folks at ESnet who helped with the deployment and planning

- Sanjay Parab (CMU), Brian Tierney, John Christman, Mark Redman, Patrick Dorn among other ESnet NESG/OCS folks
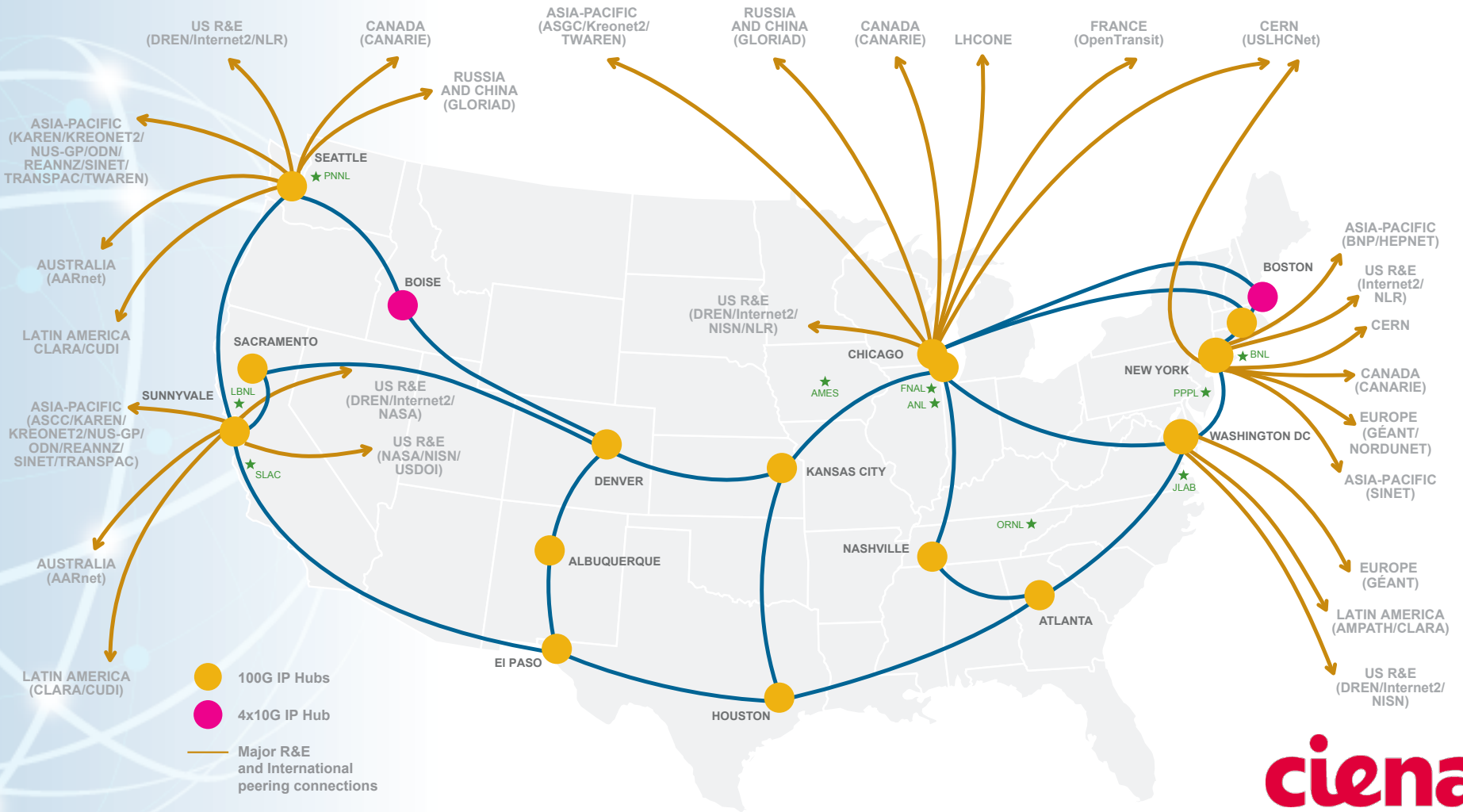
Ciena Collaborators:

- Rodney Wilson, Marc Lyonnais, Joshua Foster, Bill Webb
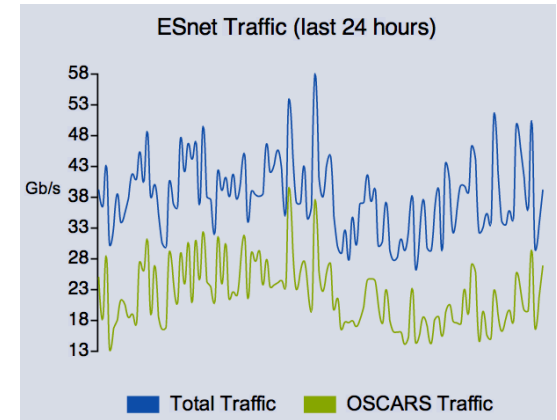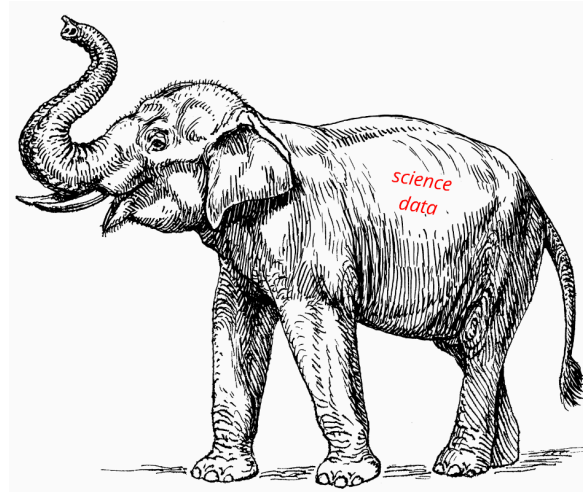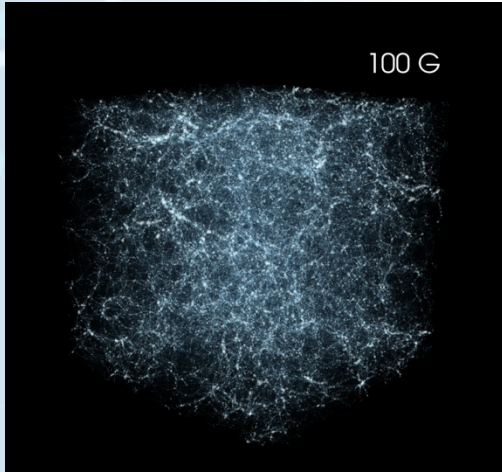
SRS Team

- Andrew Lee, Srini Seetharaman

DOE ASCR research funding that has made this work possible

# ESnet:
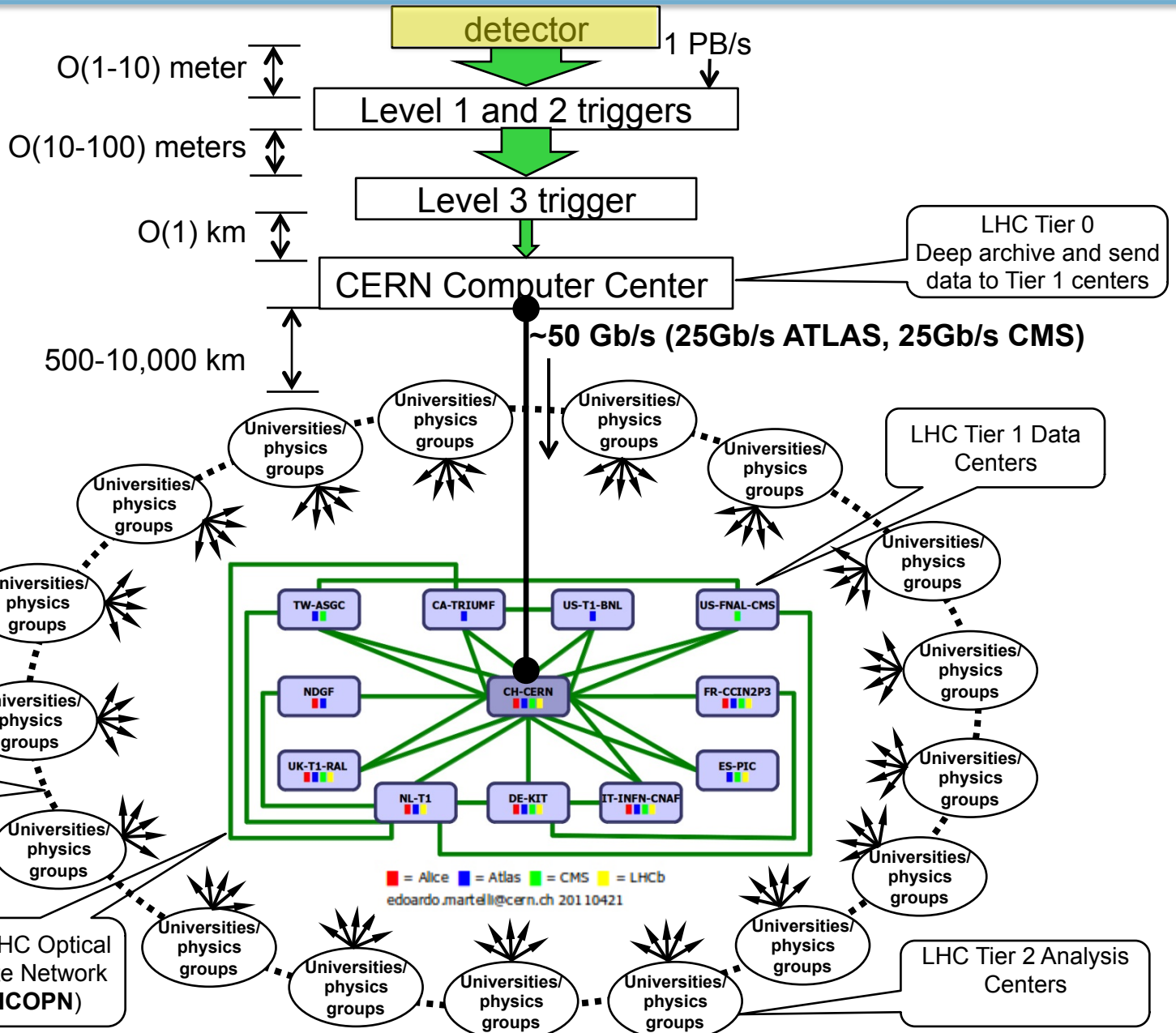# World's Leading Science Network

# Opportunities for innovation (1)



Elephant Flows: 'big-data' movement for Science, end-to-end

# Opportunity (2): Global Multi-Domain Collaborations like LHC

| CERN →T1 | miles | kms |
|---|---|---|
| France | 350 | 565 |
| Italy | 570 | 920 |
| UK | 625 | 1000 |
| Netherlands | 625 | 1000 |
| Germany | 700 | 1185 |
| Spain | 850 | 1400 |
| Nordic | 1300 | 2100 |
| USA – New York | 3900 | 6300 |
| USA - Chicago | 4400 | 7100 |
| Canada – BC | 5200 | 8400 |
| Taiwan | 6100 | 9850 |

detector

1 PB/s

O(1-10) meter

Level 1 and 2 triggers

O(10-100) meters

Level 3 trigger

O(1) km

CERN Computer Center

LHC Tier 0
Deep archive and send data to Tier 1 centers

**~50 Gb/s (25Gb/s ATLAS, 25Gb/s CMS)**

500-10,000 km

Universities/physics groups

LHC Tier 1 Data Centers

TW-ASGC   CA-TRIUMF   US-T1-BNL   US-FNAL-CMS

NDGF   CH-CERN   FR-CCIN2P3

UK-T1-RAL   ES-PIC

NL-T1   DE-KIT   IT-INFN-CNAF

■ = Alice  ■ = Atlas  ■ = CMS  ■ = LHCb
edoardo.martelli@cern.ch 2011 0421

The LHC Open Network Environment (**LHCONE**)

The LHC Optical Private Network (**LHCOPN**)

LHC Tier 2 Analysis Centers

Source: Bill Johnston

# Software-Defined Networking
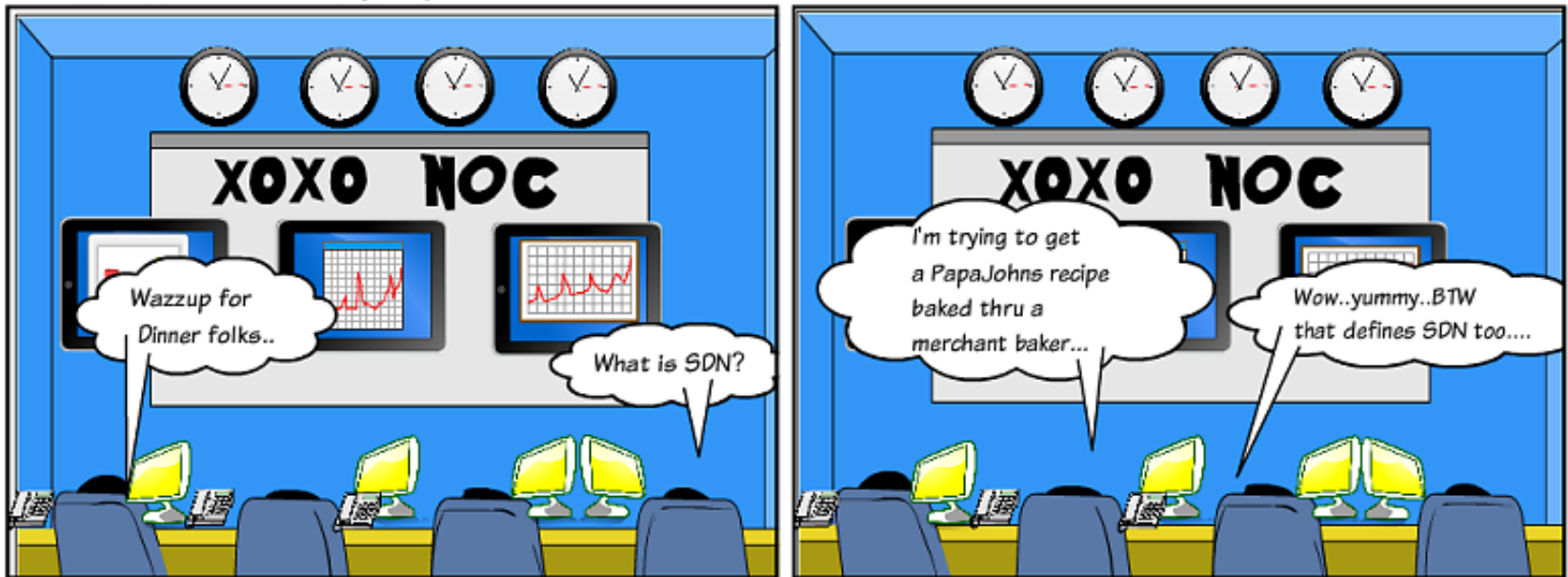
# What is Software-Defined Networking?
(as defined by Scott Shenker, October 2011)

"The ability to master complexity is not the same as the ability to extract simplicity"

"Abstractions key to extracting simplicity"

"SDN is defined precisely by these **three abstractions**
  • Distribution, forwarding, configuration "

# Fundamental Network Abstraction:
# a end-to-end circuit

Wavelength, PPP, MPLS, L2TP, GRE, NSI-CS…

A ──●────●────●────●── Z

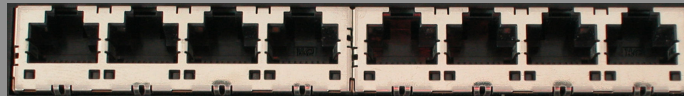Switching points, store and forward, transformation …

**Simple, Point-to-point, Provisonable**

# New Network Abstraction: "WAN Virtual Switch"


WAN Virtual Switch

## Simple, Multipoint, Programmable

**Configuration** abstraction:
- Expresses desired behavior
- Hides implementation on physical infrastructure

It is not only about the concept, but implementation
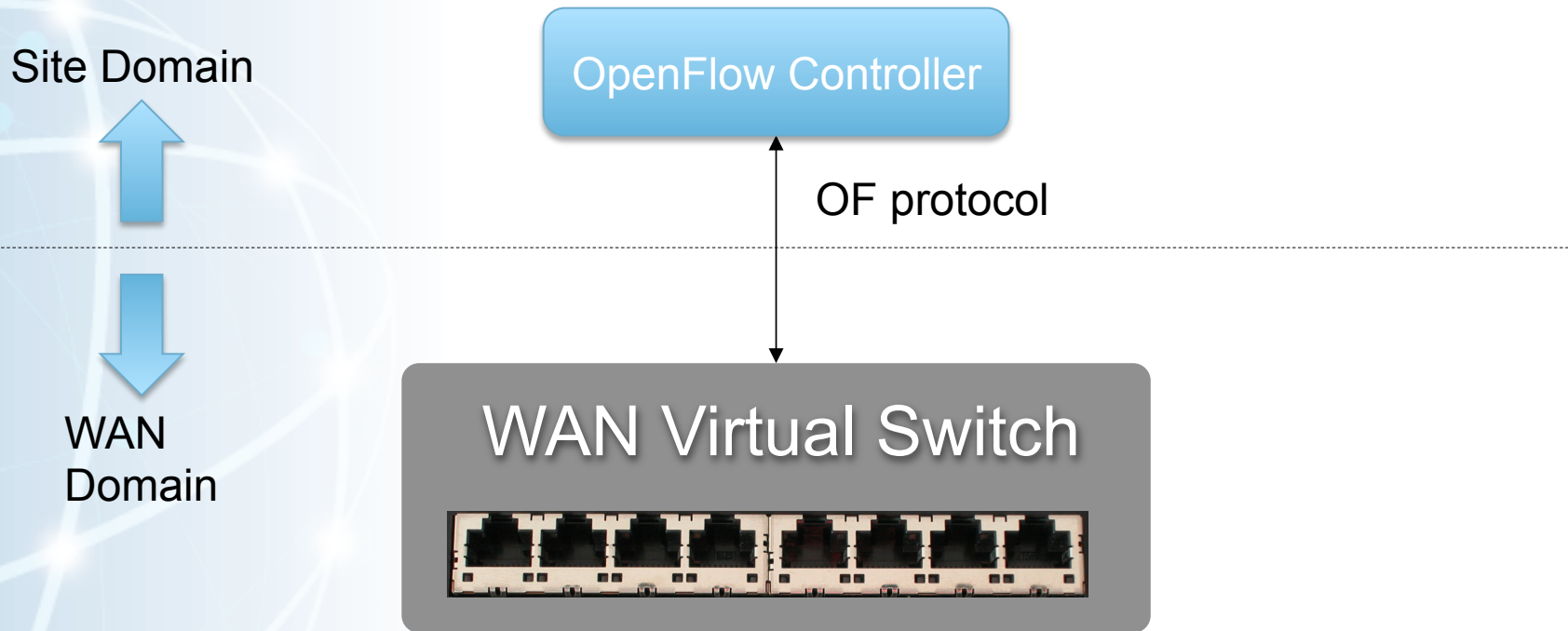
# Simple Example: One Virtual Switch per Collaboration

# Programmability



Site Domain

WAN Domain

OpenFlow Controller

OF protocol

WAN Virtual Switch

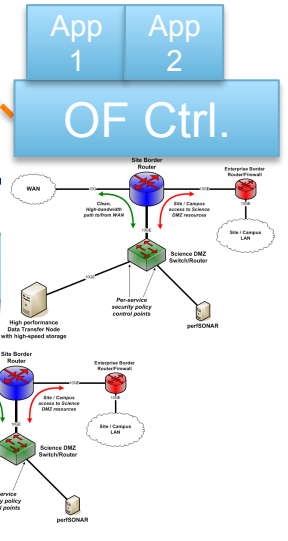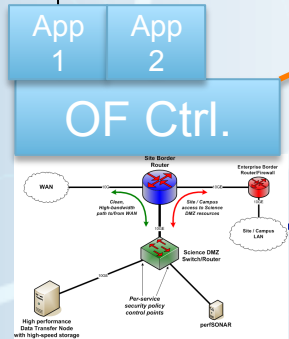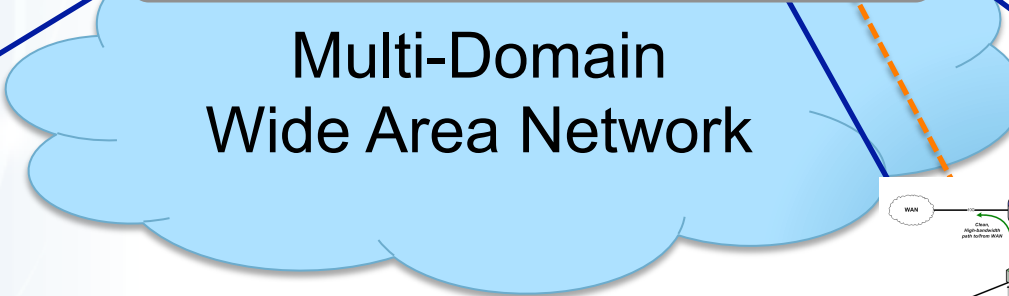Expose 'flow' programming interface leveraging standard OF protocol

# "Programmable" by end-sites
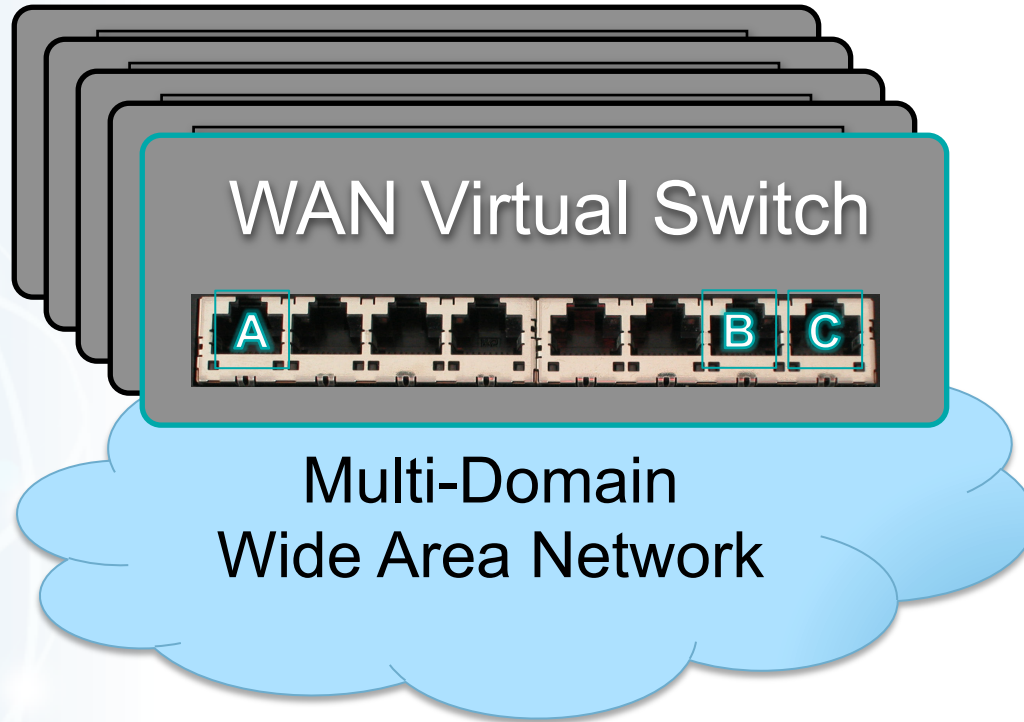
Program flows:
Science Flow1: A → B, QoS, Label
Science Flow2: A → C, VLAN
Science Flow3: A → B,C



WAN Virtual Switch

A    B  C

Multi-Domain
Wide Area Network

App 1  App 2
OF Ctrl.

App 1  App 2
OF Ctrl.

# Many collaborations, Many Virtual Switches



WAN Virtual Switch

A    B    C

Multi-Domain
Wide Area Network

# SRS Demonstration Physical Topology



Ciena 5410
@Ciena booth

DTNs

SRS
Brocade
@SCinet

@ANL

@BNL

OSCARS virtual circuits

NEC IP8800
@ LBL
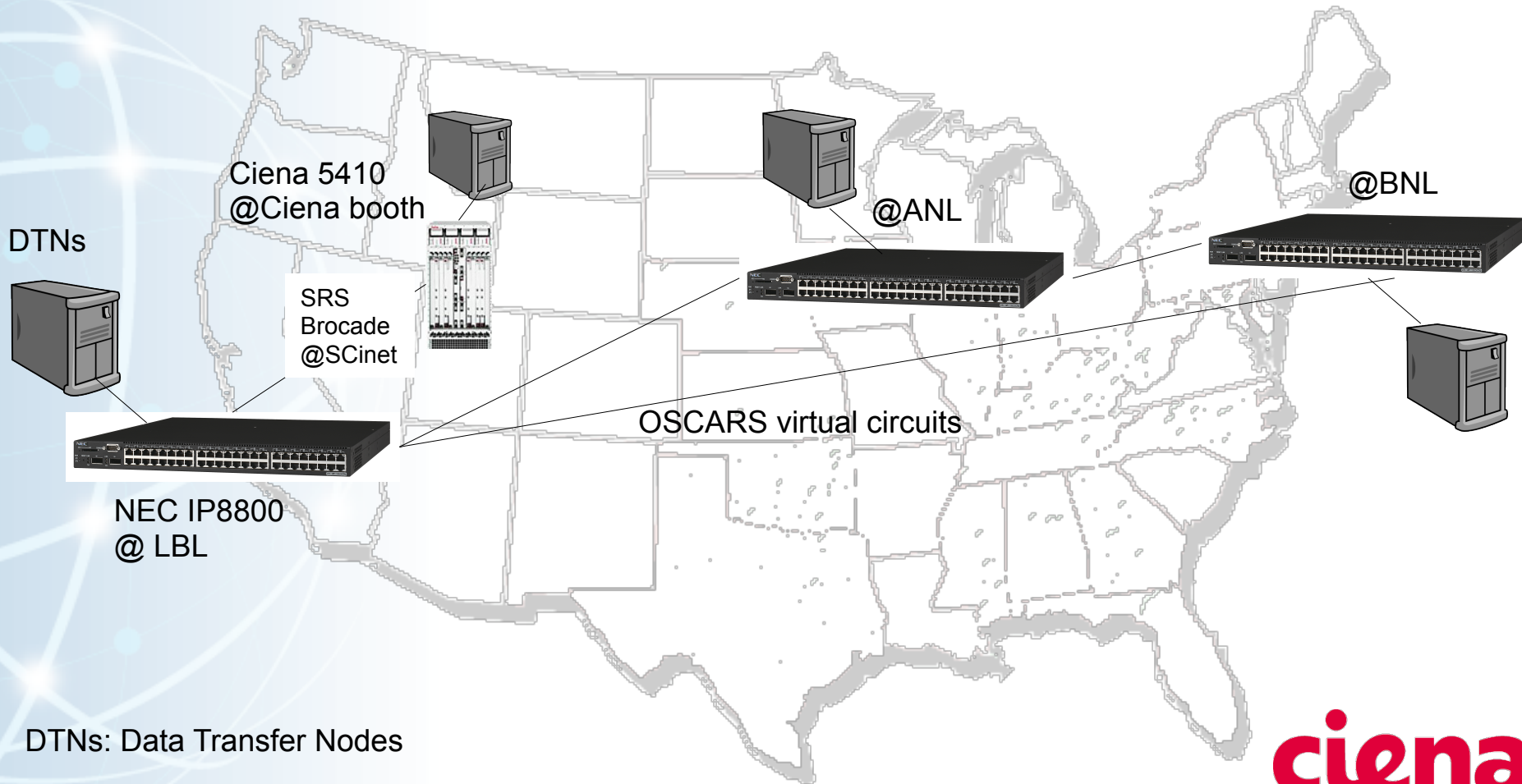
DTNs: Data Transfer Nodes

# Virtual Switch Implementation:
# Mapping abstract model to the physical



Create Virtual switch:
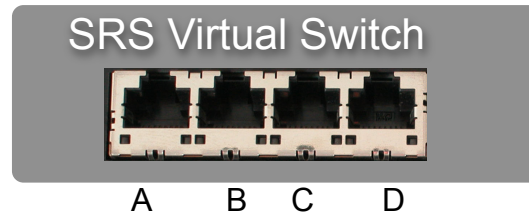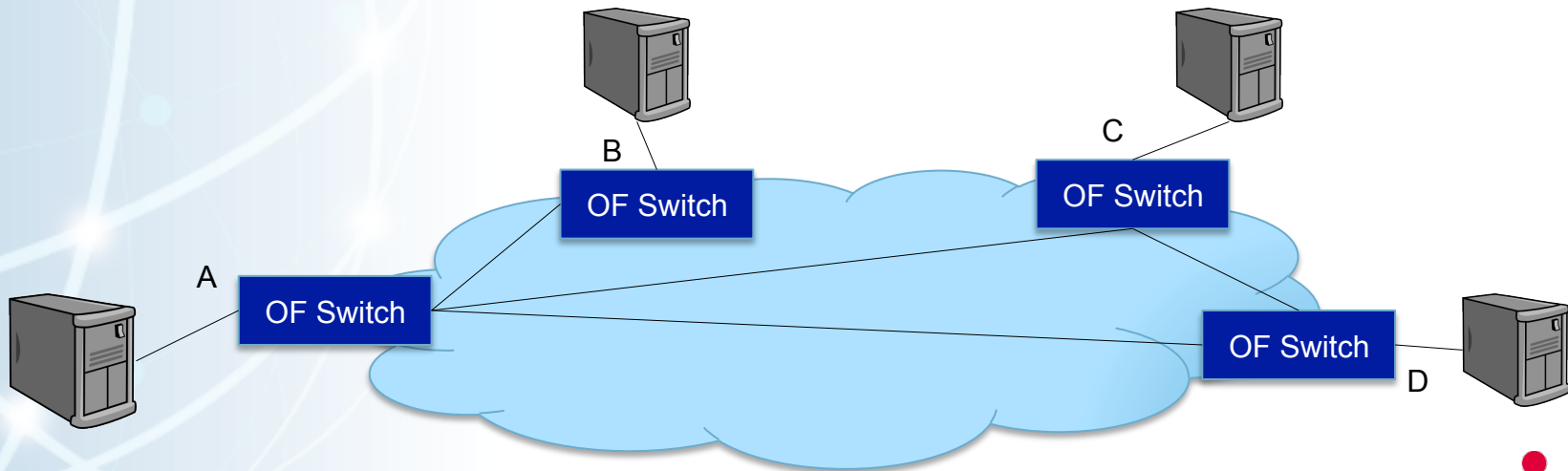- Specify edge OF ports
- Specify backplane topology and bandwidth
- Policy constraints like flowspace
- Store the switch into a topology service

SRS Virtual Switch

A   B   C   D

Virtual

Physical

OF Switch

OF Switch

OF Switch

OF Switch

A

B

C

D

# WAN Virtual Switch:
# Deploying it as a service



**Policy/Isolation of customer OF control**

App 1 | App 2

Customer Flowvisor — OF — End-site OF controller

OF
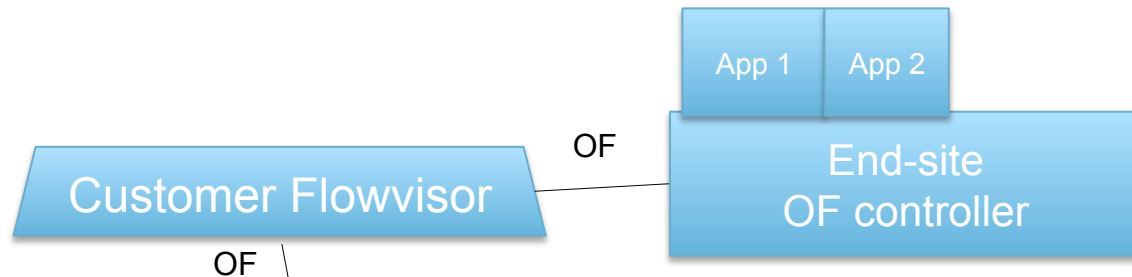
**Virtual Switch Software stack**

OpenFlow API

Virtual Switch Application

Virtual SW Controller | OSCARS Client
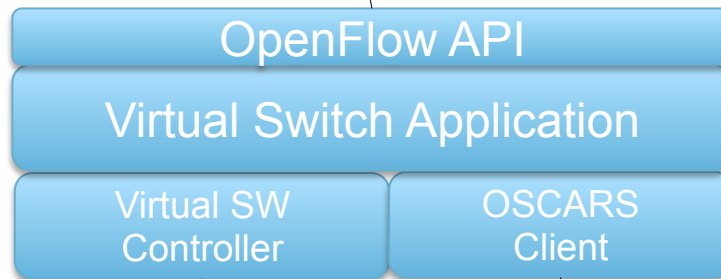
**Infrastructure Software, Slicing and provisioning**

OF | OSCARS API

Network Flowvisor | OSCARS

OF

Wide Area Network

OF Switch | OF Switch | OF Switch | OF Switch

# Example of ping across WAN virtual switch



3. Need to ARP

6. virtual-mac-addresses learned and mapped to real flow

SRS Virtual Switch

4. ARP

2. Packet_in

4. ARP

5. ARP response   H2

1. Ping H2

7. flow_mod

OF Switch

8. Ping H2

H1

OF Switch

8. Ping H2

OF Switch
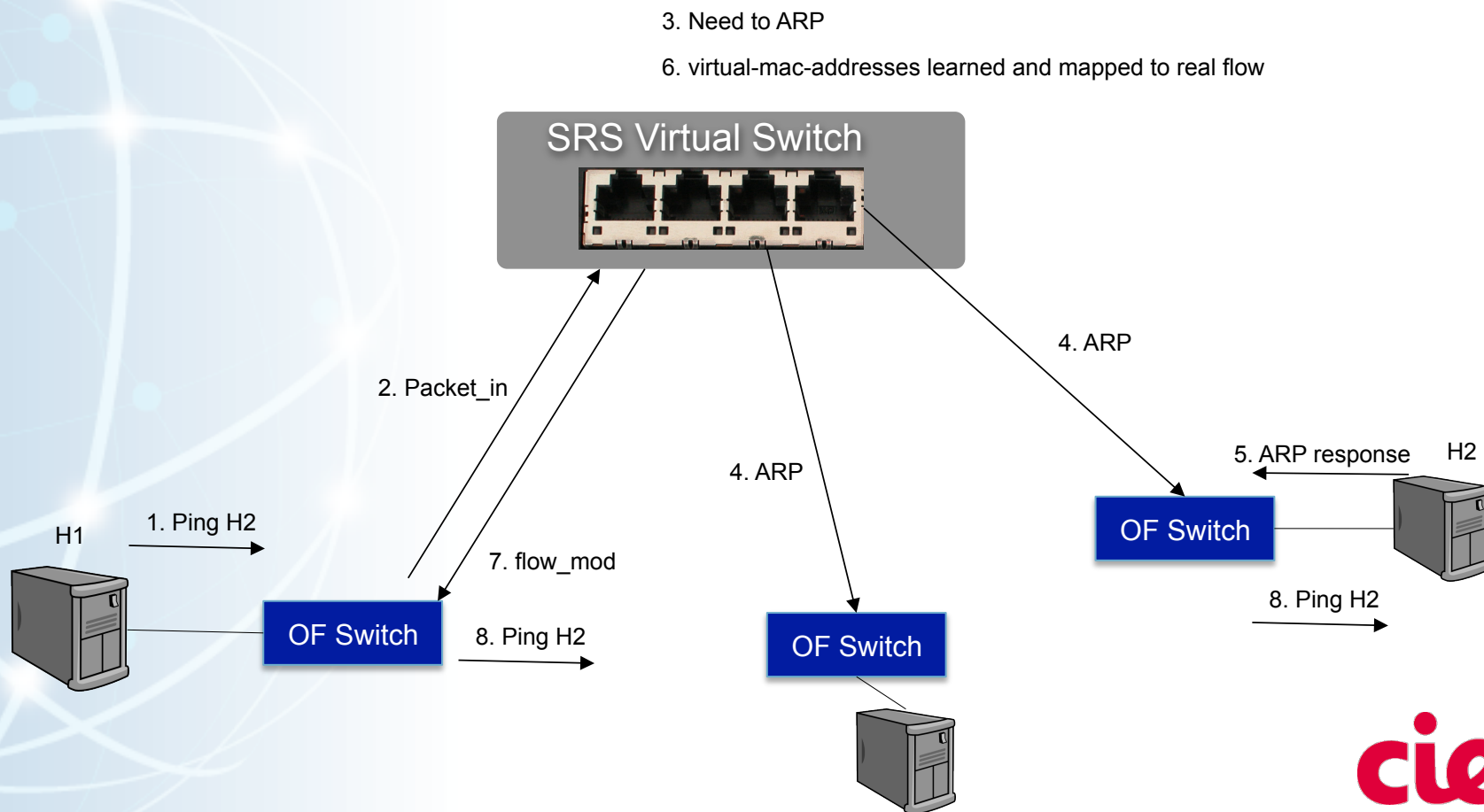
# What does this mean for networking?



User SDN Controller

*OpenFlow*

Customer/User Control Plane

*Policy and Isolation*

Network Service Interface

**WAN Virtual Switch**

**Programmable** service provisioning plane

Network Service Interface

*Multi-domain*

*Legacy and OpenFlow control plane*

**End-to-End** Dataplane
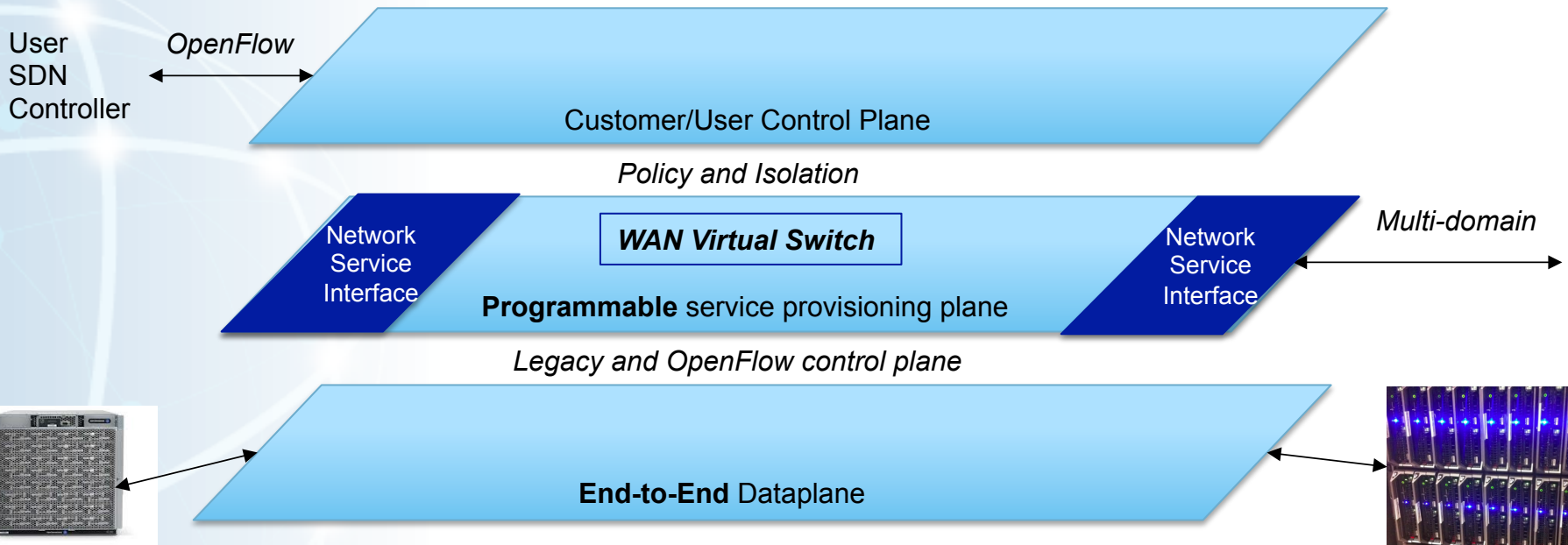
- Creation of a programmable network provisioning layer
- Sits on top of the "network OS"

# Summary

- Powerful network abstraction

  - Files / Storage

- Benefits

  - **Simplicity** for the end-site
    - Works with off-the-shelf, open-source controller
    - Topology simplification

  - **Generic** code for the network provider
    - Virtual switch can be layered over optical, routed or switched network elements
    - OpenFlow support needed on edge devices only, core stays same

  - **Programmability** for applications
    - Allows end-sites to innovate and use the WAN effectively

# Future Work

Harden the architecture and software implementation

- Move from experiment to test service

Verify scaling of the model

- Using virtual machines, other emulation environments

Automation and Intelligent provisioning

- Work over multi-domain

- Wizards for provisioning

- Dynamic switch backplane

Create recurring abstractions

- Virtual switch in campus

- How do we deal with a "network" of virtual switches

Questions – please contact imonga at es.net

www.es.net

# Thank you!

# Computer virtualization