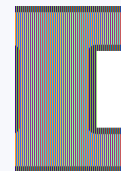




# FABRIC: An Everywhere Programmable Research Infrastructure for Network Experimentation

Kuang-Ching (KC) Wang, Clemson University  
Paul Ruth, RENCi

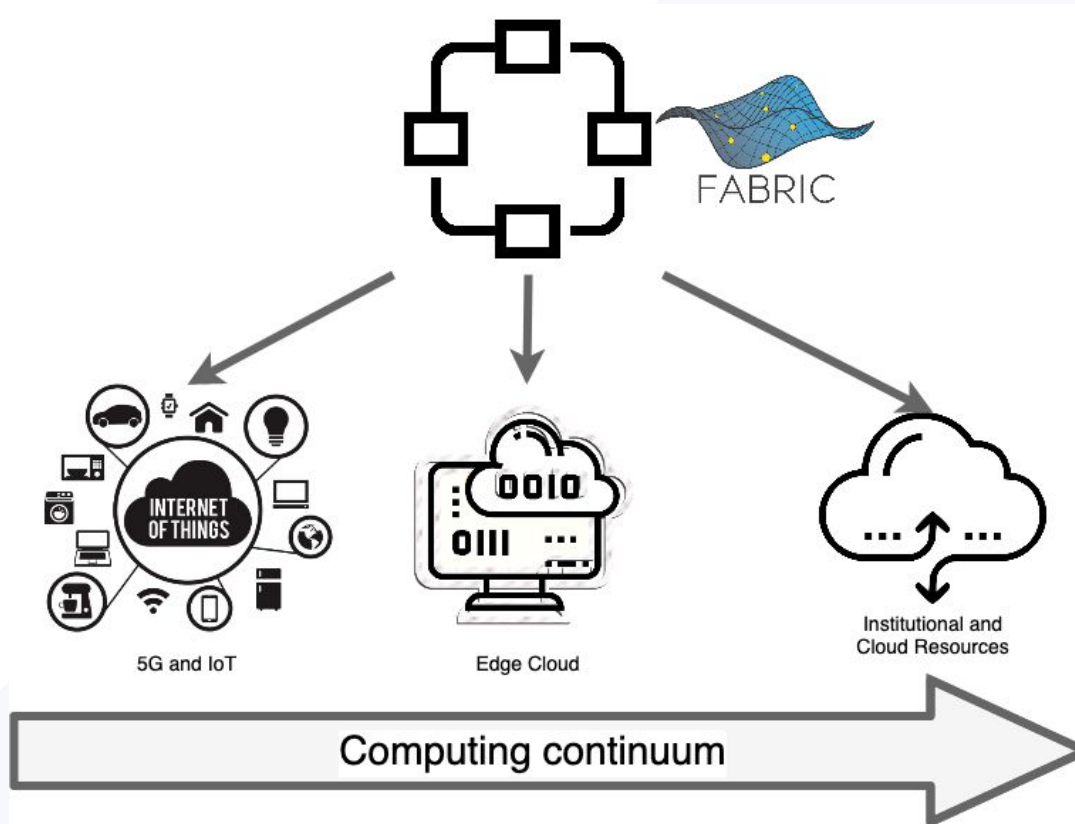
Programmable Switches Workshop, Feb. 16, 2022



# Why FABRIC?

- Change in economics of compute and storage allow for the possibility that future Internet is more stateful than we've come to believe
  - “If we had to build a router from scratch today it wouldn't look like the routers we build today”
  - Explosion of capabilities in augmented computing - GPUs, FPGAs
  - Opportunity to reimagine network architecture as more stateful
- ML/AI revolution
  - Network as a 'big-data' instrument: real-time measurements + inferencing control loop
    - Network vendors have caught on to it:
      - “Self-driving network” - Juniper CTO Kireeti Kompella
  - Provisioning, cyber-security, other applications
- IoT + 5G - the new high-speed intelligent network edge
- New science applications
  - New distributed applications - data distribution, computing, storage
- A continuum of computing capabilities
  - Not just fixed points - “edge” or “public cloud”
  - Network as part of the computing substrate - computing, fusing, processing data on the fly

# Network as part of computing continuum



# FABRIC for everyone



## **FABRIC Enables New Internet and Science Applications**

- Stateful network architectures, distributed applications that directly program the network



## **FABRIC Advances Cybersecurity**

- At-scale realistic research facilitated by peering with production networks



## **FABRIC Integrates HPC, Wireless, and IoT**

- A diverse environment connecting PAWR testbeds, NSF Clouds, HPC centers and instruments



## **FABRIC Integrates Machine Learning & Artificial Intelligence**

- Support for in-network GPU-accelerated data analysis and control



## **FABRIC helps train the next generation of computer science researchers**

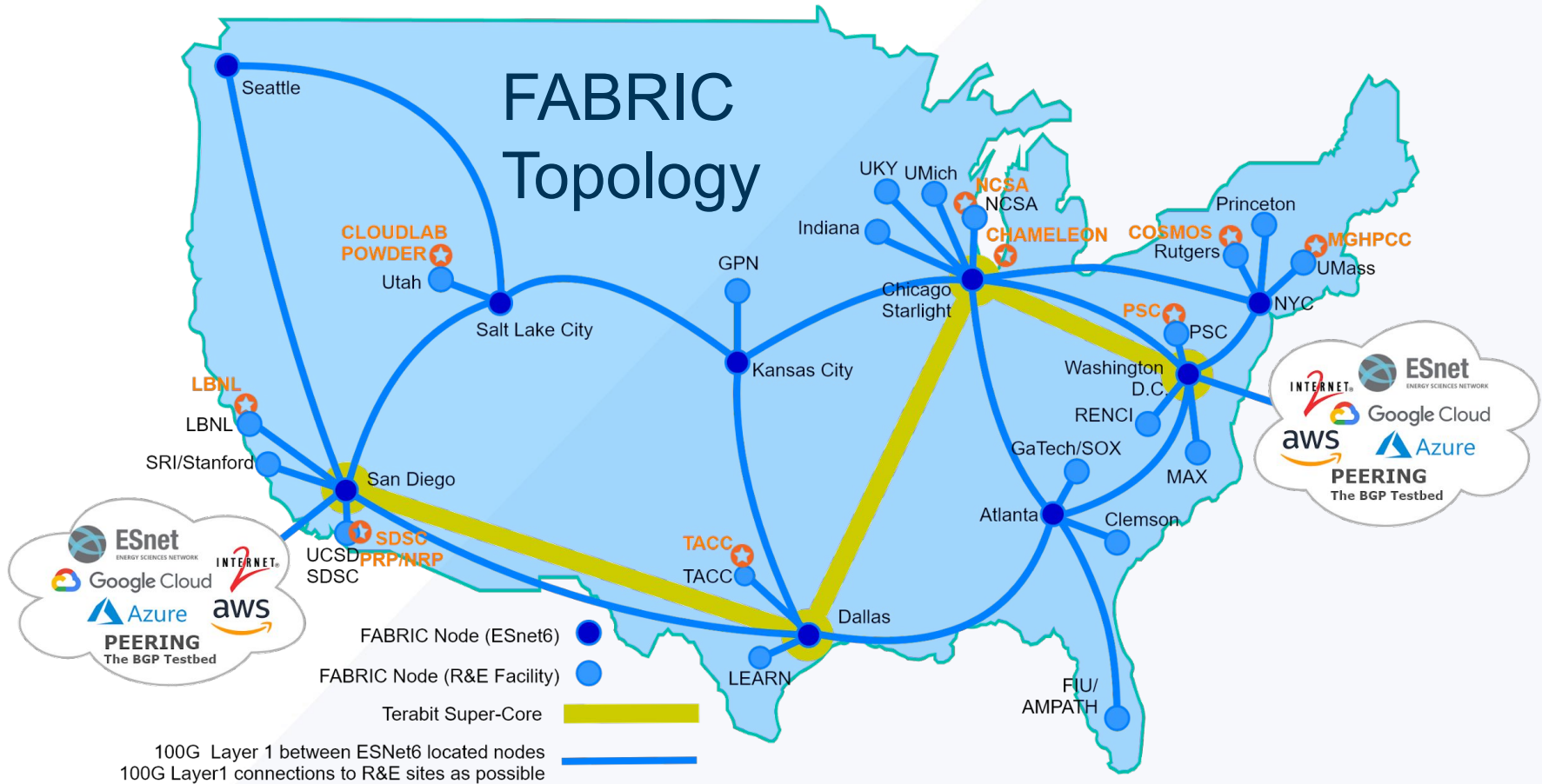
# What is FABRIC?

**FABRIC enables a completely *new paradigm for distributed applications and Internet protocols and services:***

- A **nation-wide programmable network** testbed with **significant compute and storage at each node**, allowing users to run computationally intensive programs and applications and protocols to maintain a lot of information **in the network**.
- Provides **GPUs, FPGAs, and network processors (NICs)** inside the network.
- Supports **quality of service (QoS)** using dedicated optical 100G links or dedicated capacity
- **Interconnects national facilities:** HPC centers, cloud & wireless testbeds, commercial clouds, the Internet, and edge nodes at universities and labs.
- Allows you to design and test **applications, protocols and services that run at any node in the network**, not just the edge or cloud.



# FABRIC Topology



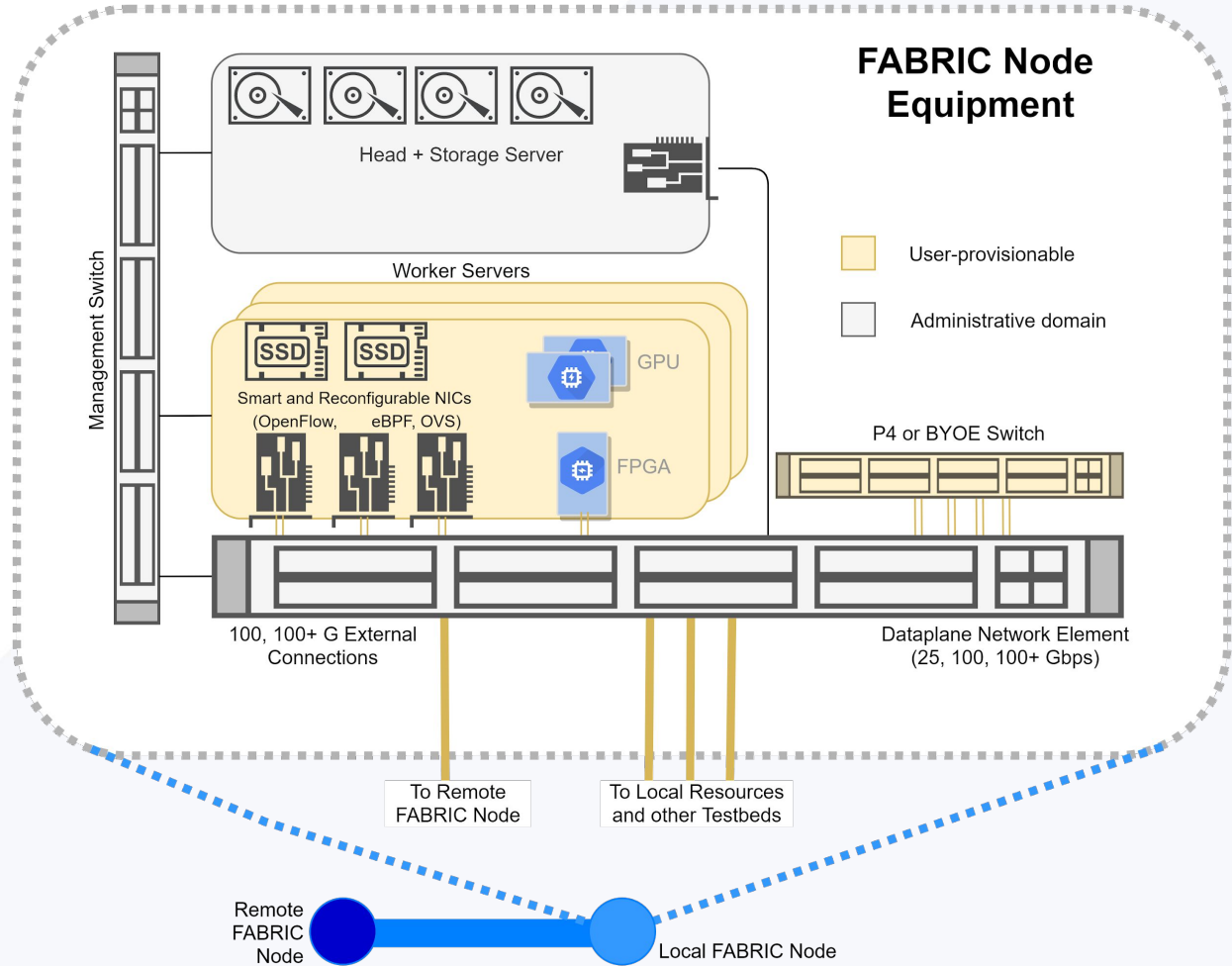
100G Layer 1 between ESNet6 located nodes  
100G Layer1 connections to R&E sites as possible

# FAB

- FAB (FABRIC Across Borders) is the international expansion of FABRIC to Asia and Europe
  - Funded by NSF IRNC (International Research Network Connections) program
- Led by Anita Nikolich
- Includes sites in Japan (University of Tokyo), UK (University of Bristol), EU (University of Amsterdam and CERN)
  - To be deployed in 2021-2023 timeframe similar to the rest of FABRIC
  - Linked by available capacity on IRNC trans-oceanic links
- Brings new use-cases
  - Astronomy/Cosmology, High-Energy Physics, Urban Sensing/IoT
  - Computer Science: 5G across borders, P4/SDN, Cyber-security/Censorship evasion

# Conceptual FABRIC Node 'Hank' Overview

a.k.a. 'A  
disaggregated  
router'





# FABRIC Nodes

- Interpose compute and storage into the path of fast packet flows
- Rack of high-performance servers (Dell 7525) with:
  - 2x32-core AMD 7532 with 512G RAM
  - GPUs (RTX 6000 and T4), FPGA network/compute accelerators
  - Storage - experimenter provisionable 1TB NVMe drives in servers and a pool of ~250TB rotating storage at each site.
  - Network ports connect to a 100G+ switch, programmable through control software
- Reconfigurable Network Interface Cards
  - FPGAs (with P4 support)
  - Mellanox ConnectX-5 and ConnectX-6 with hardware off-load
  - Multiple interface speeds (25G, 100G, 200G+(future))
- Kernel Bypass/Hardware Offload
  - VM/Containers sized to support full-rate DPDK for access to Programmable NICs, FPGA, and GPU resources via PCI pass-through

# FABRIC Node Design: Measurement Hardware

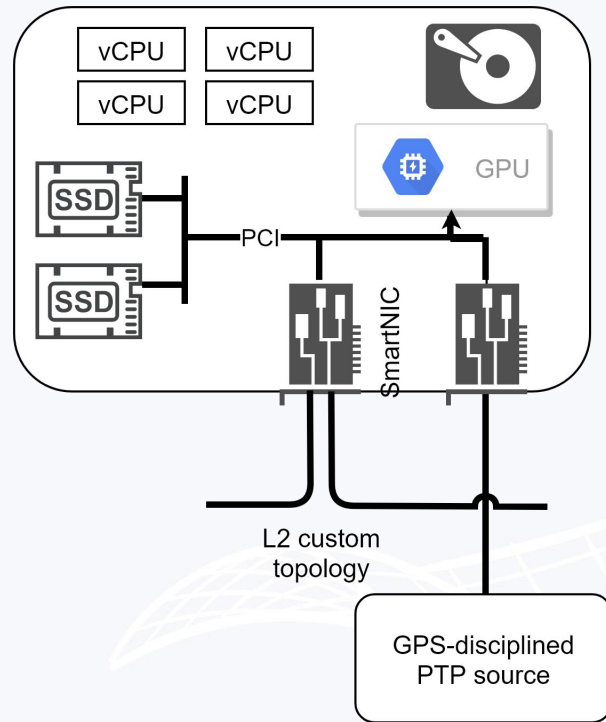
- GPS-disciplined clock source at most sites using PTP
  - Subject to constraints of the hosting site
- NICs capable of accurate packet sampling/timestamping
  - High touch/ sampling story
- Programmable port mirroring
- Smart PDUs to measure power
- Optical layer measurements (where available)
- CPU, memory, disk, port/interface utilization and other time-series (software)

# FABRIC Experiment building blocks

- Each experiment is encapsulated in a slice - a topology
- Slices consist of slivers
  - Individually programmable or configurable resources
- Slices can change over time
  - Grow or shrink, adding or shedding resources under programmatic control
- Slice topologies can be
  - Custom L2 using underlying MPLS-SR
  - Rely on persistent routable IPv6 layer in FABRIC
- Basic sliver classes
  - Nodes - can include a selection of PCI-passthrough devices
  - Links - L2 or L3 with QoS and without
  - Measurement points - inside and outside the slice

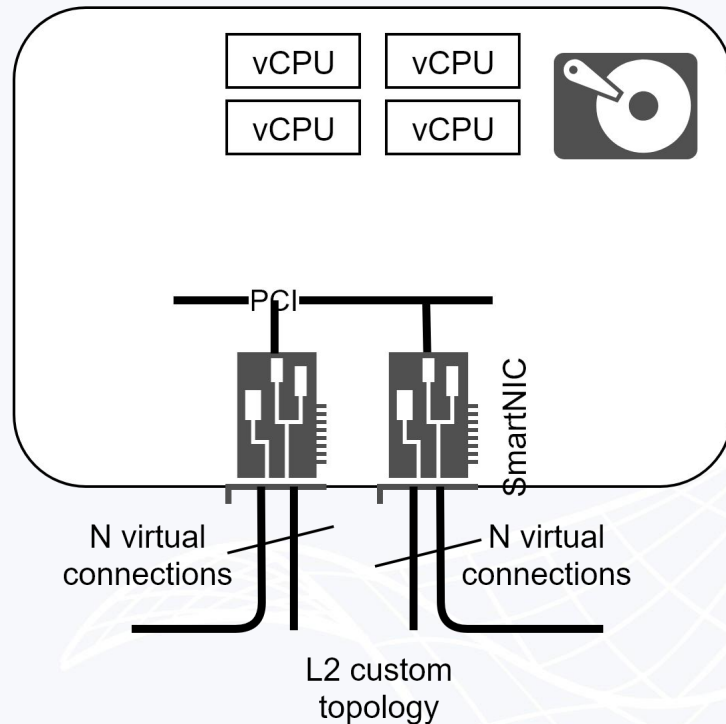
# Bump-in-wire sliver

- Useful for collecting and analysing high-volume packet traces
  - Rely on NVMe drive for high-throughput local storage
  - Use GPU to assist in analysis
- Can optionally use a local GPS-disciplined PTP source to achieve millisecond-level accuracy for measurements
  - Multiple 'bumps-in-wire' in a slice can help create a snapshot of traffic across the network in a given instant in time



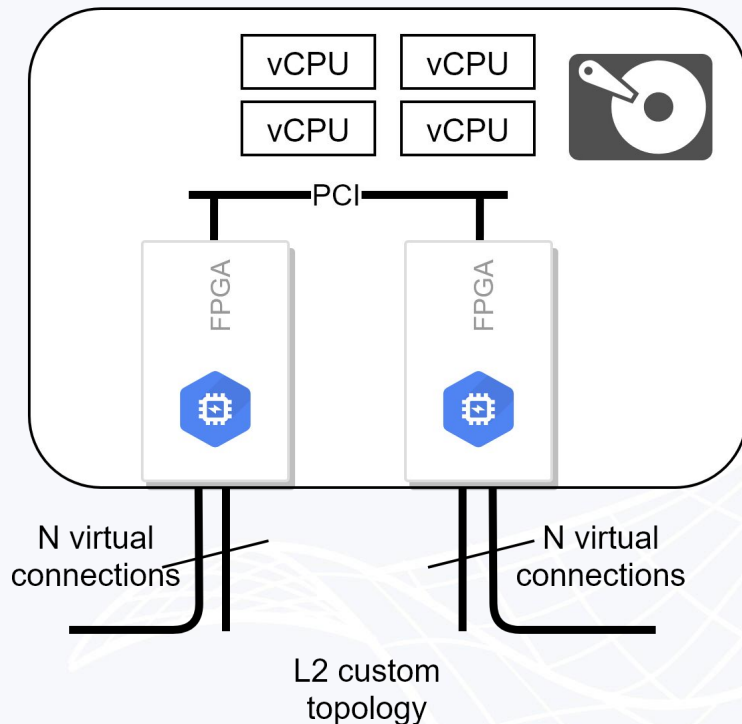
# SmartNIC router sliver

- Can create a small-port-count OpenFlow router with hardware acceleration via Mellanox ConnectX-[5,6] cards
  - Direct access to PCI allows to bypass CPU in many cases.



# FPGA or P4 router sliver

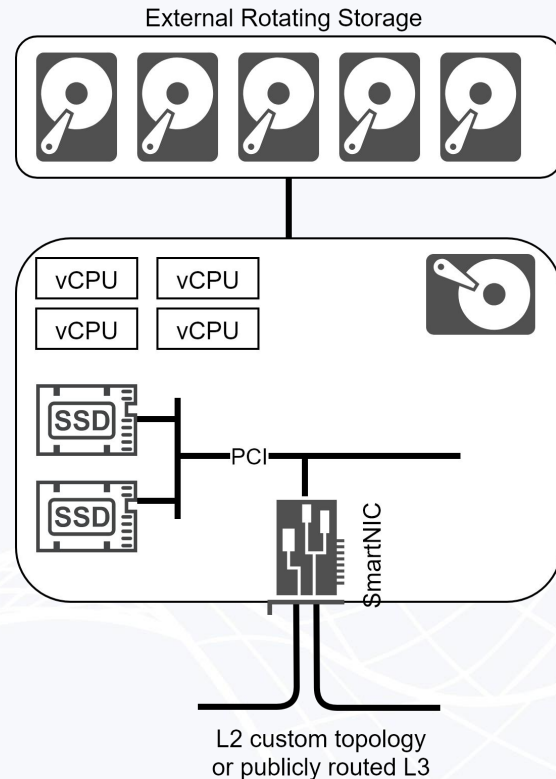
- Uses Xilinx FPGAs in a node
- Can build a small port-count FPGA router
- With additional tools support can also serve as a P4 router built on top of the FPGA
- Can route between multiple virtual connections based on e.g. VLAN tags or other header information
- 



# Caching/processing with tiered storage

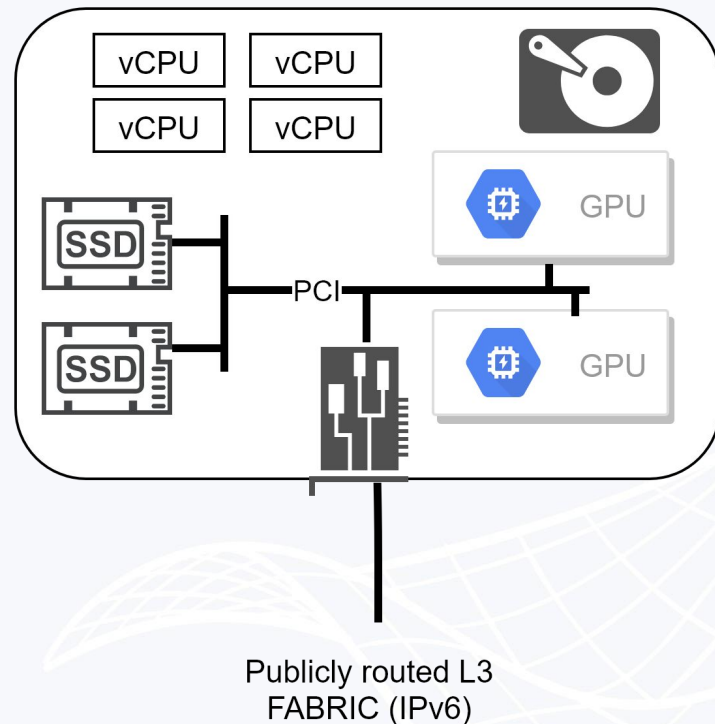
- Collect in-network measurement data and store using different storage tiers:

- Speed ↑
- RAM
  - Attached NVMe drive
  - Local rotating storage
  - External (local to the site) large volume rotating storage
- ↓ Available Size



# In-network AI/ML

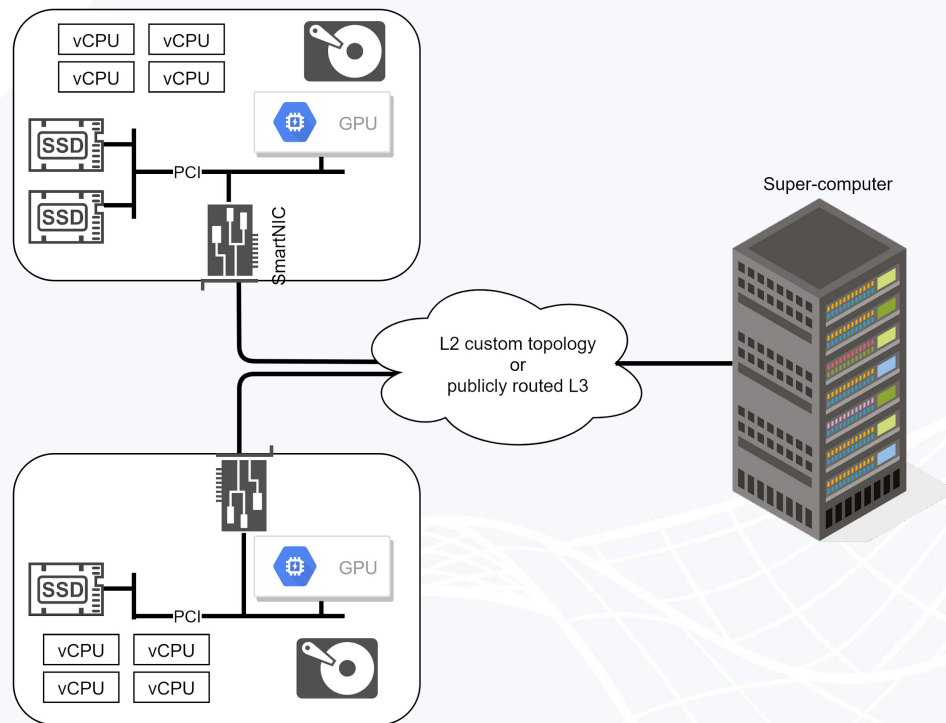
- Investigating autonomous network behavior using in-network GPU support
  - Using RTX6000 for learning and inference using streaming data
- Perform intelligent data fusion/processing in the network
- Implement in-network analytics/security functions





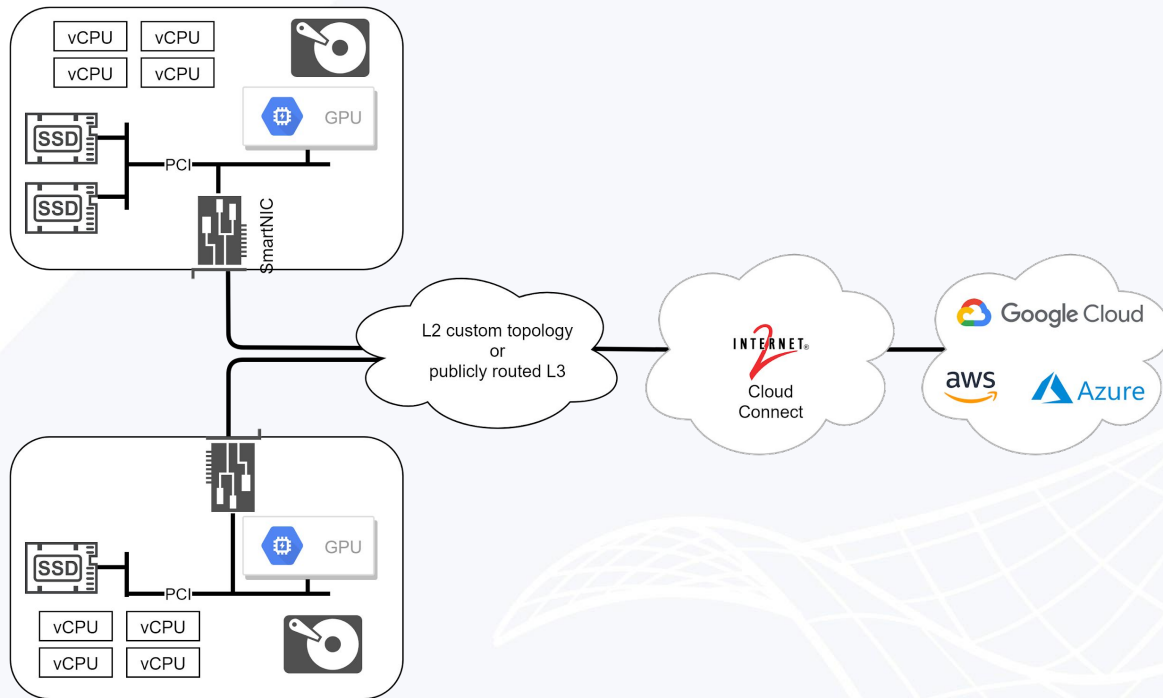
# Attaching external facilities

- The US NSF has made significant investments in scientific CI
- Future networks must better support domain science needs
- FABRIC connects to a number of facilities and testbeds to enrich the set of resources that can be used in experiments
  - Supercomputing centers (PSC, NCSA, SDSC, TACC, MGHPCC)
  - Cloud testbeds - CloudLab, Chameleon, Open Cloud Testbed
  - 5G testbeds - COSMOS, Powder
- Through FAB we will also reach
  - University of Bristol, University of Amsterdam, University of Tokyo, CERN



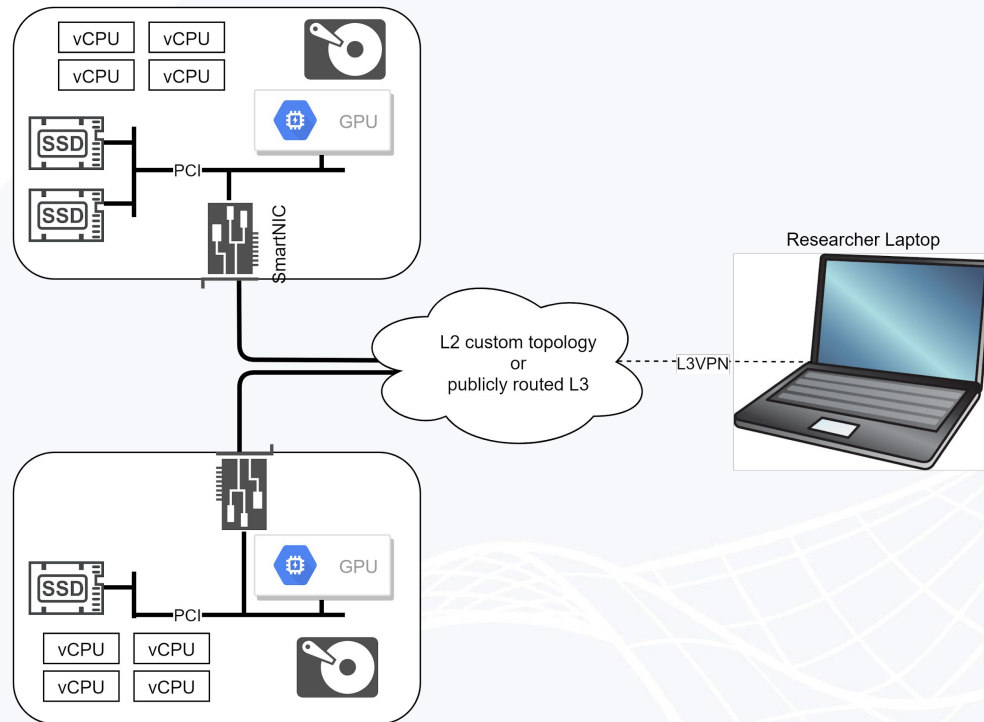
# Using public clouds in experiments

- Future networks will connect clouds and their customers
- 5G+Cloud experiments
- Through partnership with Internet 2 FABRIC will provide connectivity to commercial clouds
  - Utilize I2 CloudConnect system



# Adding experimenter-owned resources

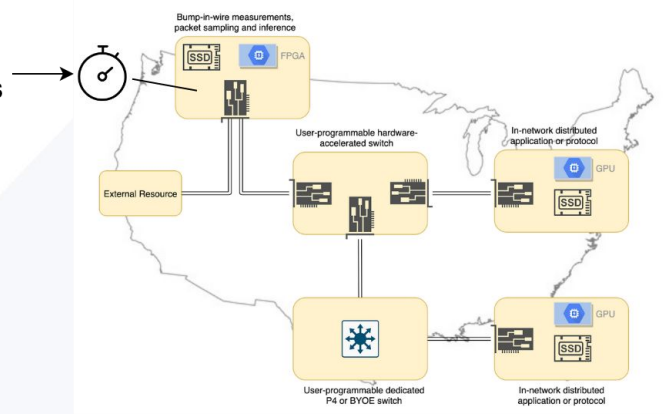
- Many experimenters may be interested in connecting their own resources to their slice topologies
  - FABRIC may not be able to reach every campus with a dedicated connection
- VPN/VPW options will be available to support these cases.
  - Allow experimenters to offer services to others from their slices



# FABRIC Measurement Capabilities

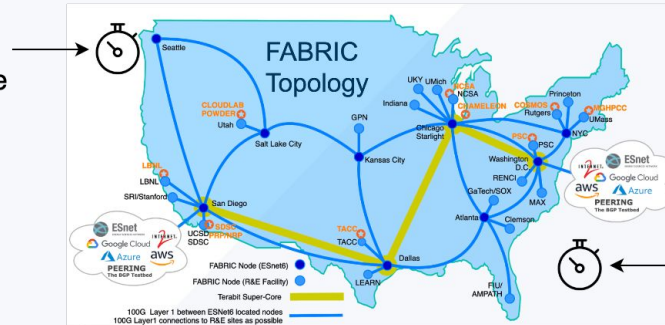
- Key to FABRIC being a scientific instrument
- Provides measurements
  - Inside the slice
  - Outside the slice

In-slice measurements



Slice

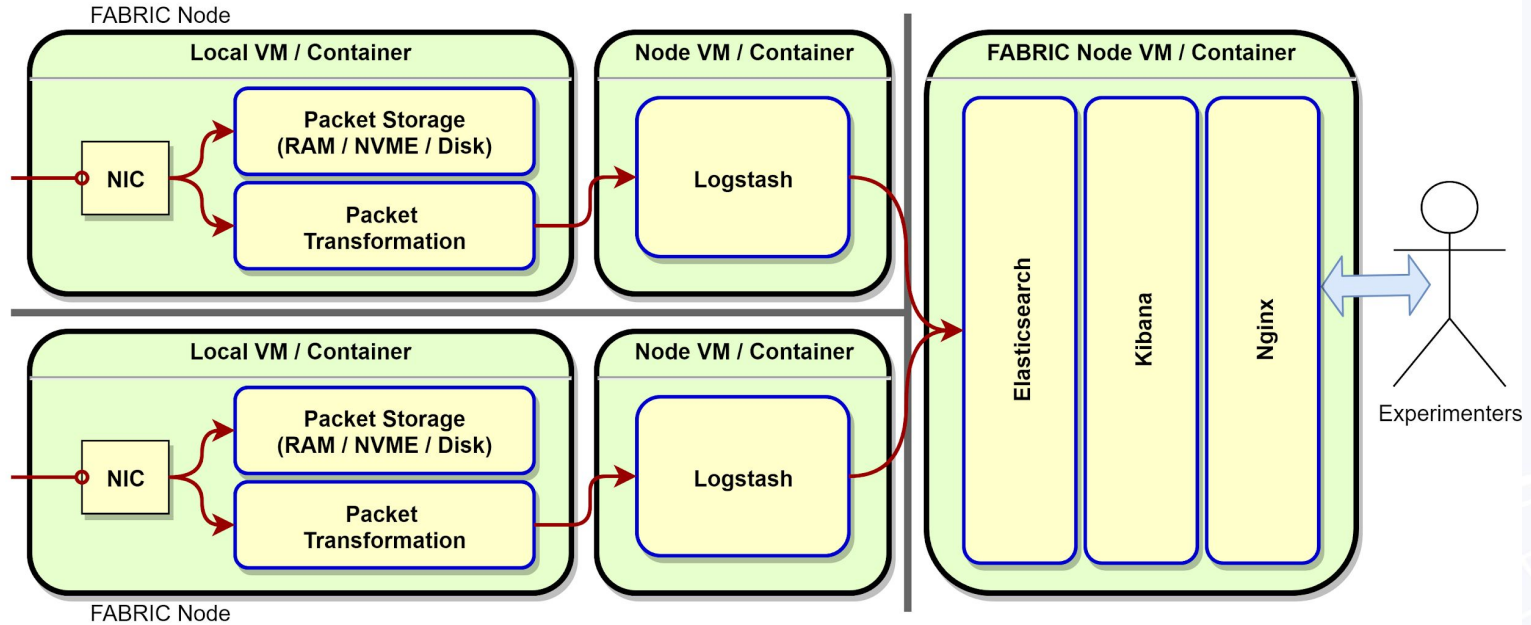
Measurements of infrastructure supporting the slice



FABRIC Substrate

Infrastructure health measurements

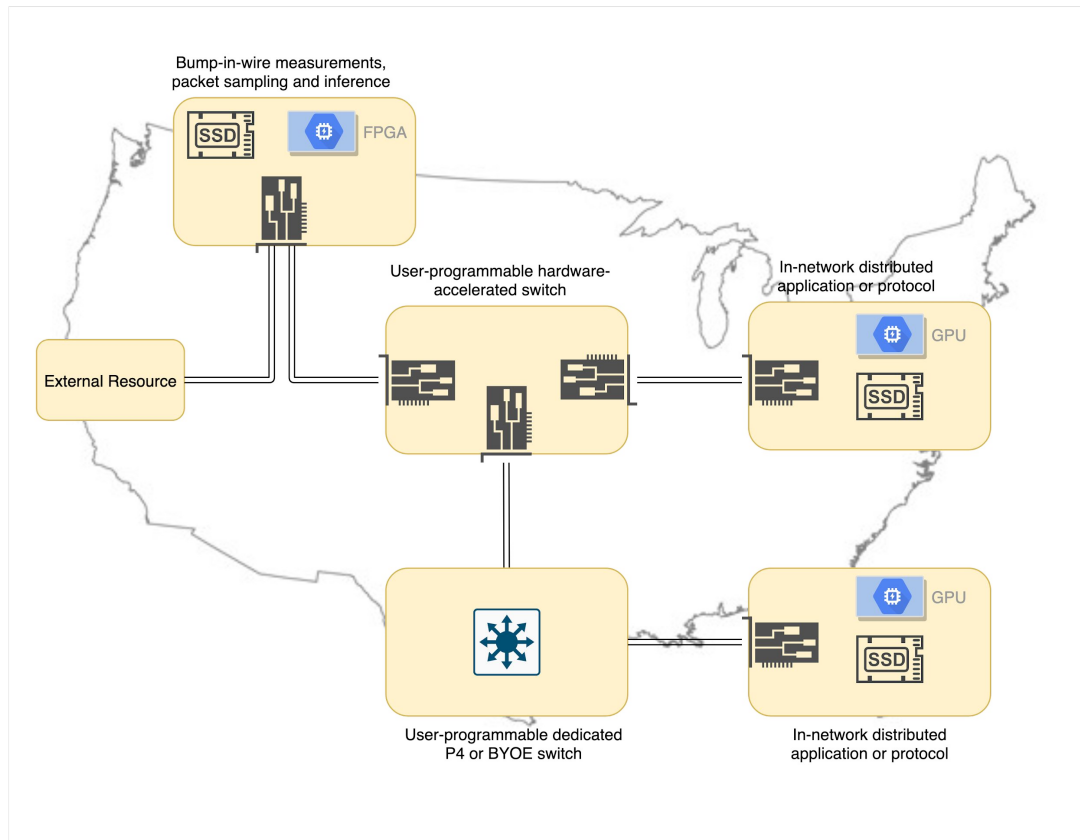
# FABRIC Measurement Data Processing/Analysis



# Enabling P4 on FABRIC

Examples of potential uses:

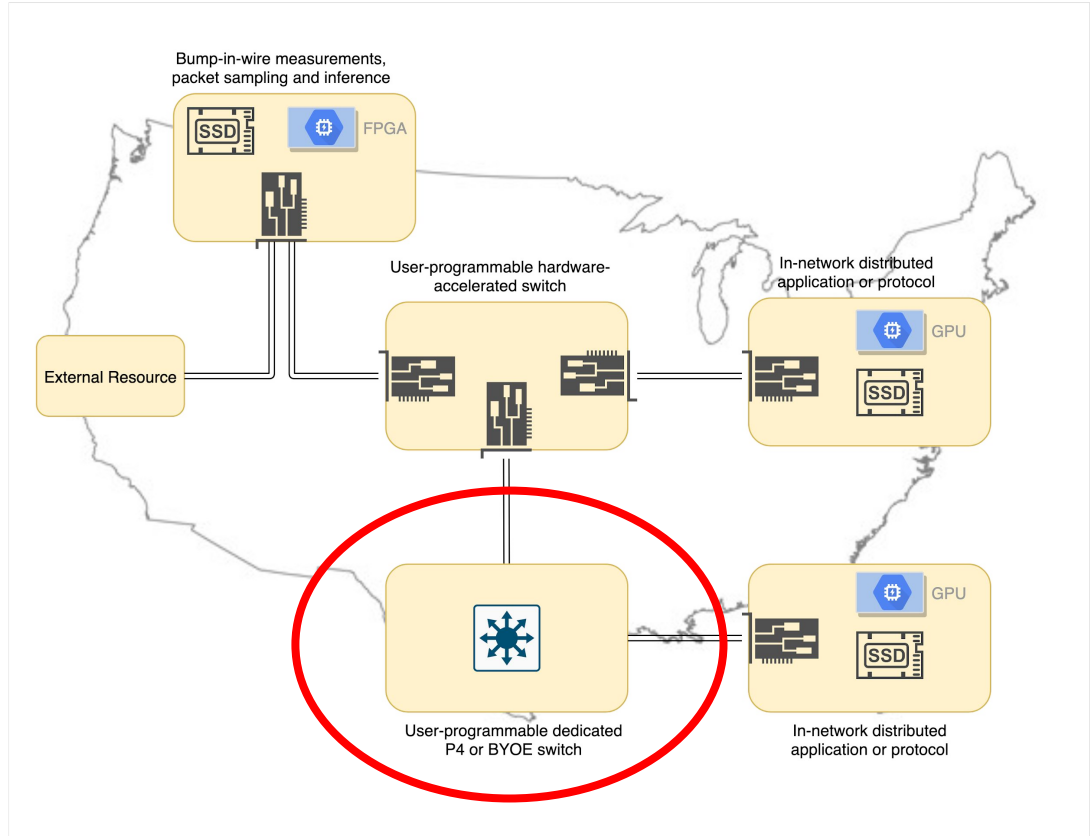
- **Bump-in-wire** measurements and packet sampling at high bit rates (25, 40, 100, 100+ Gbps)
- **Hardware-accelerated switching** using Smart NICs, FPGA NICs or P4 switches in individual nodes
- **Hosting in-network applications** and stateful architectures using a combination of storage and compute resources in individual nodes
- **In-network inference**, other types of accelerated computing via FPGAs and GPUs
- **Connect experiments to external facilities** like IoT, 5G, cloud testbeds, public clouds and HPC resources.
- **Deploy non-IP protocols** on top of wide-area L2 topologies, that may include in-network processing and storage



# Enabling P4 on FABRIC

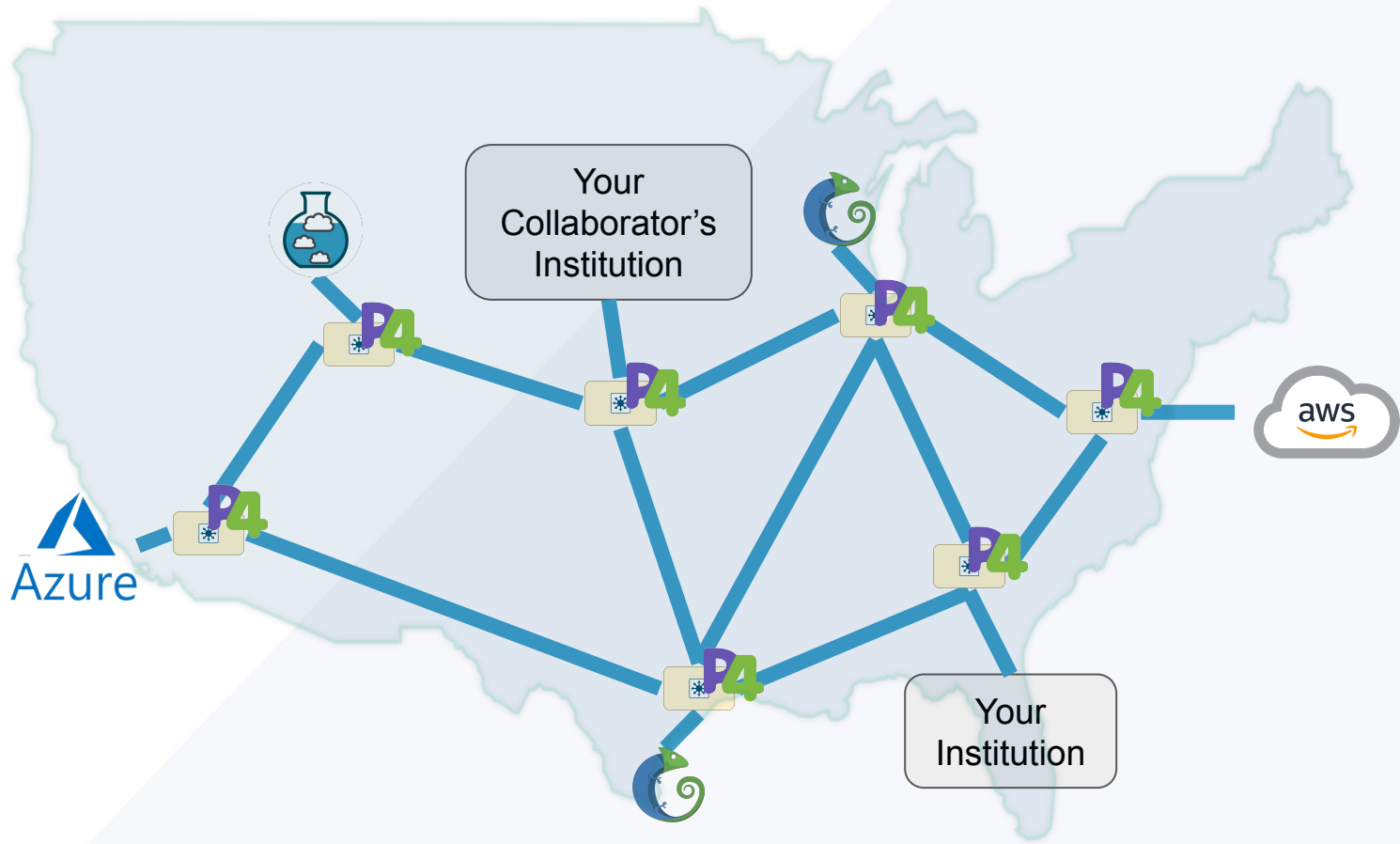
Examples of potential uses:

- **Bump-in-wire** measurements and packet sampling at high bit rates (25, 40, 100, 100+ Gbps)
- **Hardware-accelerated switching** using Smart NICs, FPGA NICs or P4 switches in individual nodes
- **Hosting in-network applications** and stateful architectures using a combination of storage and compute resources in individual nodes
- **In-network inference**, other types of accelerated computing via FPGAs and GPUs
- **Connect experiments to external facilities** like IoT, 5G, cloud testbeds, public clouds and HPC resources.
- **Deploy non-IP protocols** on top of wide-area L2 topologies, that may include in-network processing and storage



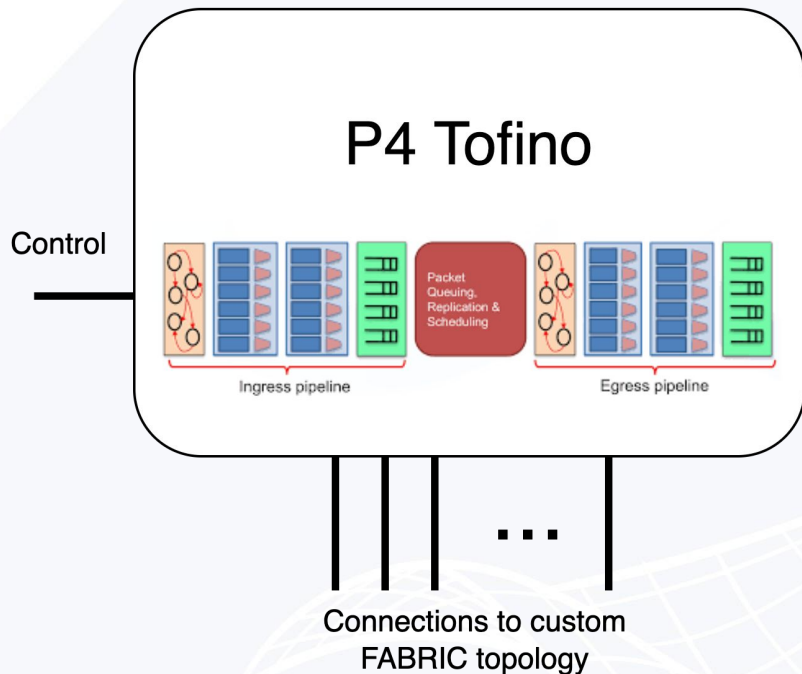






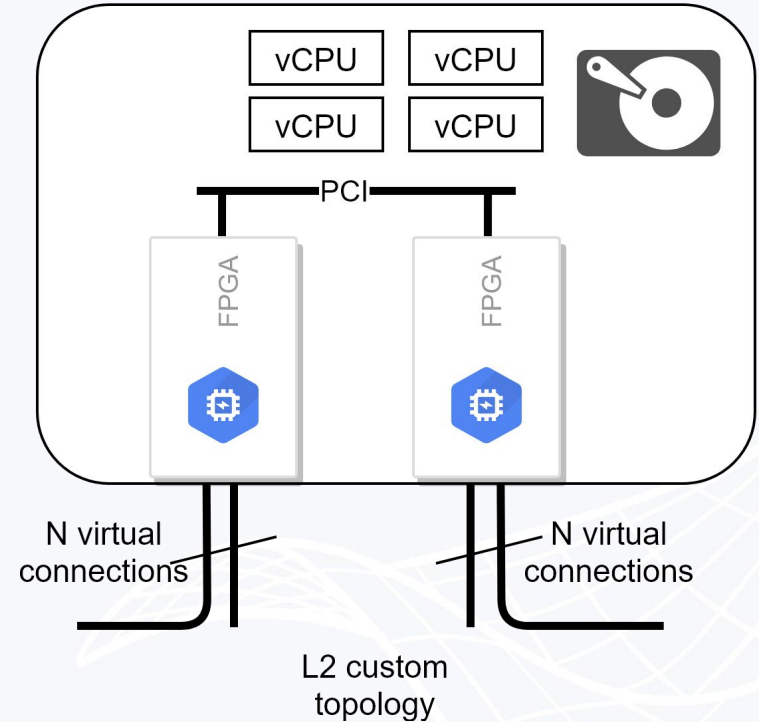
# P4 Resources: Tofino (Not yet available)

- Dedicated Tofino switches controlled by the user
- Initial hardware is being acquired
- Licensing and NDAs complicate sharing with arbitrary users.
  - User workflow is being designed



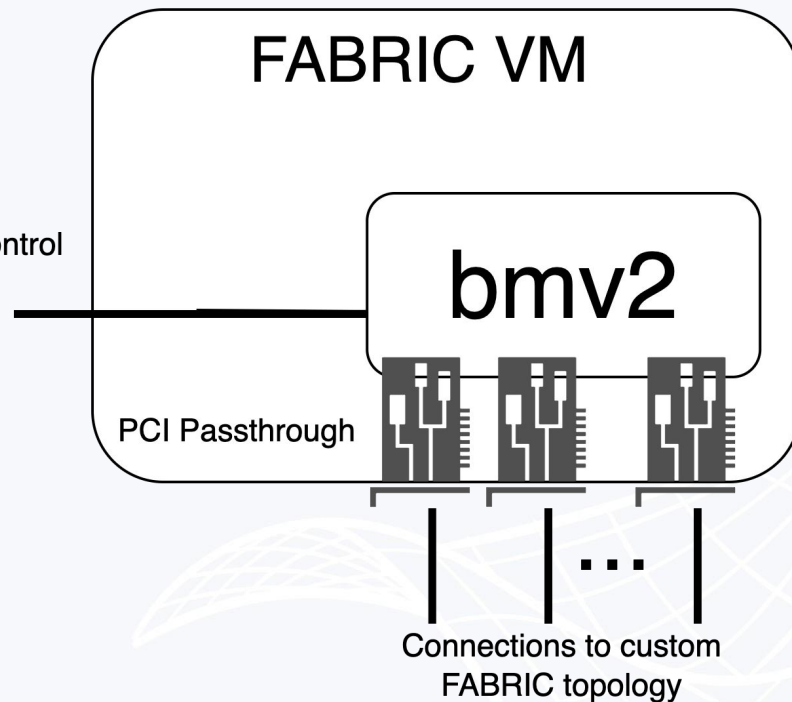
# P4 Resources: Xilinx FPGA (Not yet available)

- Uses Xilinx FPGAs in a node
- Can build a small port-count FPGA router
- With additional tools support can also serve as a P4 router built on top of the FPGA
- Xilinx P4-SDNet extensions
- FABRIC P4 bit code currently under development at Northeastern (Miriam Leeser)



# P4 Resources: Software P4 (available)

- P4 Behavioral Model (BMV2)
  - Opensource
  - Jupyter notebook available
- Tofino Native Architecture (possible)
  - User must provide Tofino Model and Intel P4 Studio
  - Intel NDA Required
- Primarily for education and development

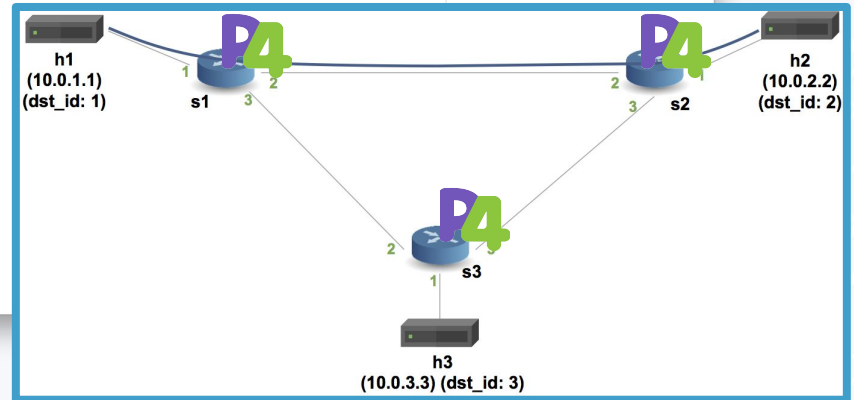


# Example: P4Lang Tutorials

- P4Lang Tutorials
  - Great place to start with P4
  - Hands-on exercises
  - Github repository
- Resources and Topology
  - One virtual machine
  - Mininet



The screenshot shows the GitHub repository for P4Lang tutorials. The main content is the README.md file, which includes a table of contents with five sections: 1. Introduction and Language Basics, 2. P4Runtime and the Control Plane, 3. Monitoring and Debugging, 4. Advanced Behavior, and 5. Stateful Packet Processing. Each section lists specific topics like 'Basic Forwarding', 'Source Routing', and 'Firewall'. On the right side, there are statistics for contributors (38 total) and languages used (P4 at 42.7%, Python at 41.3%, etc.).



# Example: P4Lang Tutorials



- P4Lang Tutorials
  - Great place to start with P4
  - Hands-on exercises
  - Github repository
- Resources and Topology
  - One virtual machine
  - Mininet

The screenshot shows the GitHub repository for P4Lang Tutorials. The main content is the README.md file, which includes a table of contents and a list of exercises. The network diagram illustrates a topology for an exercise titled "Exercise: Basic custom tunnel".

**Exercise: Basic custom tunnel**

The network diagram shows three hosts (h1, h2, h3) and three switches (s1, s2, s3). Host h1 (10.0.1.1) is connected to switch s1. Host h2 (10.0.2.2) is connected to switch s2. Host h3 (10.0.3.3) is connected to switch s3. Switches s1 and s2 are connected to each other. Switches s1 and s3 are connected to each other. Switches s2 and s3 are connected to each other. The diagram also shows a direct connection between h1 and h2. The exercise involves creating a custom tunnel between h1 and h2.

Language	Percentage
P4	42.7%
Python	41.3%
Shell	6.9%
Emacs Lisp	4.2%
Vim Script	2.5%
TeX	1.8%
Makefile	0.6%

# Example: P4Lang Tutorials



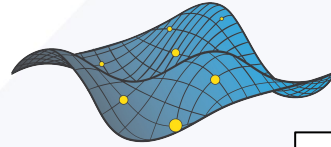
Great videos on YouTube featuring your next speaker!  
(Vladimir Gurevich)

A screenshot of a YouTube video player. The video is titled "Introduction to P4\_16, Part 1" and is from the "2017 P4 Developer Day (P4 D2) May 16 2017" series. The video content shows a slide titled "V1 Architecture" with the following bullet points:

- Compatible with P4\_14 architecture
- Is implemented on top of bmv2-simple\_switch target
- Will be gradually introduced in the due course

The slide also contains a diagram of the V1 architecture. The diagram shows a flow from left to right: a "Programmable Parser" (represented by a box with circular arrows), followed by two blue boxes representing switch fabric stages, then a red box labeled "Packet Queuing, Replication & Scheduling", followed by two more blue boxes representing switch fabric stages, and finally a "Programmable Deparser" (represented by a box with horizontal lines). The diagram is annotated with red brackets and arrows indicating the flow and components. The video player interface includes a progress bar at 33:23 / 1:14:39, a video thumbnail of the speaker, and engagement metrics like 112 likes and 10,576 views.

# Example: P4Lang on FABRIC



FABRIC



- P4Lang Tutorials
  - Same tutorials
  - JupyterHub notebook using FABlib library
- Resources and Topology
  - Individual BMV2 switches
  - Uses any FABRIC site
  - Dedicated WAN L2 links

**P4Lang Tutorials on FABRIC**

This notebook walks the user through setting up a FABRIC experiment that is suitable for completing the P4 tutorials created by P4Lang. The tutorials were originally designed to use a mininet topology. This example replaces the mininet topology with a FABRIC experiment topology that may span multiple sites across the FABRIC testbed.

Additional resources:

- FABRIC Knowledge Base
- FABRIC Forums
- P4Lang Tutorials
- P4Lang YouTube Presentations

```
[1]: import os

# If you are using the
# were automatically P
os.environ["FABRIC_CA
os.environ["FABRIC_OR
os.environ["FABRIC_TR

# Bastion IPs
os.environ["FABRIC_BAS

# Set your Bastion use
os.environ["FABRIC_BAS
os.environ["FABRIC_BAS

# Set the keypair FABR
os.environ["FABRIC_SL1
os.environ["FABRIC_SL1

# If your private key
# from getpass import g
#print("Please input P
os.environ["FABRIC_SL

Import the FABLIB

[2]: import json
import traceback
from fabrictestbed_ext
```

**Network Diagram:** A topology diagram showing four Hosts (yellow boxes) and three P4 Switches (blue boxes). The Hosts are arranged in a line: Host - P4 Switch - P4 Switch - Host. A third P4 Switch is connected to the two middle P4 Switches and a Host below it. The Hosts are also connected to the P4 Switches they are adjacent to.

Goal: Replace Mininet with FABRIC resources





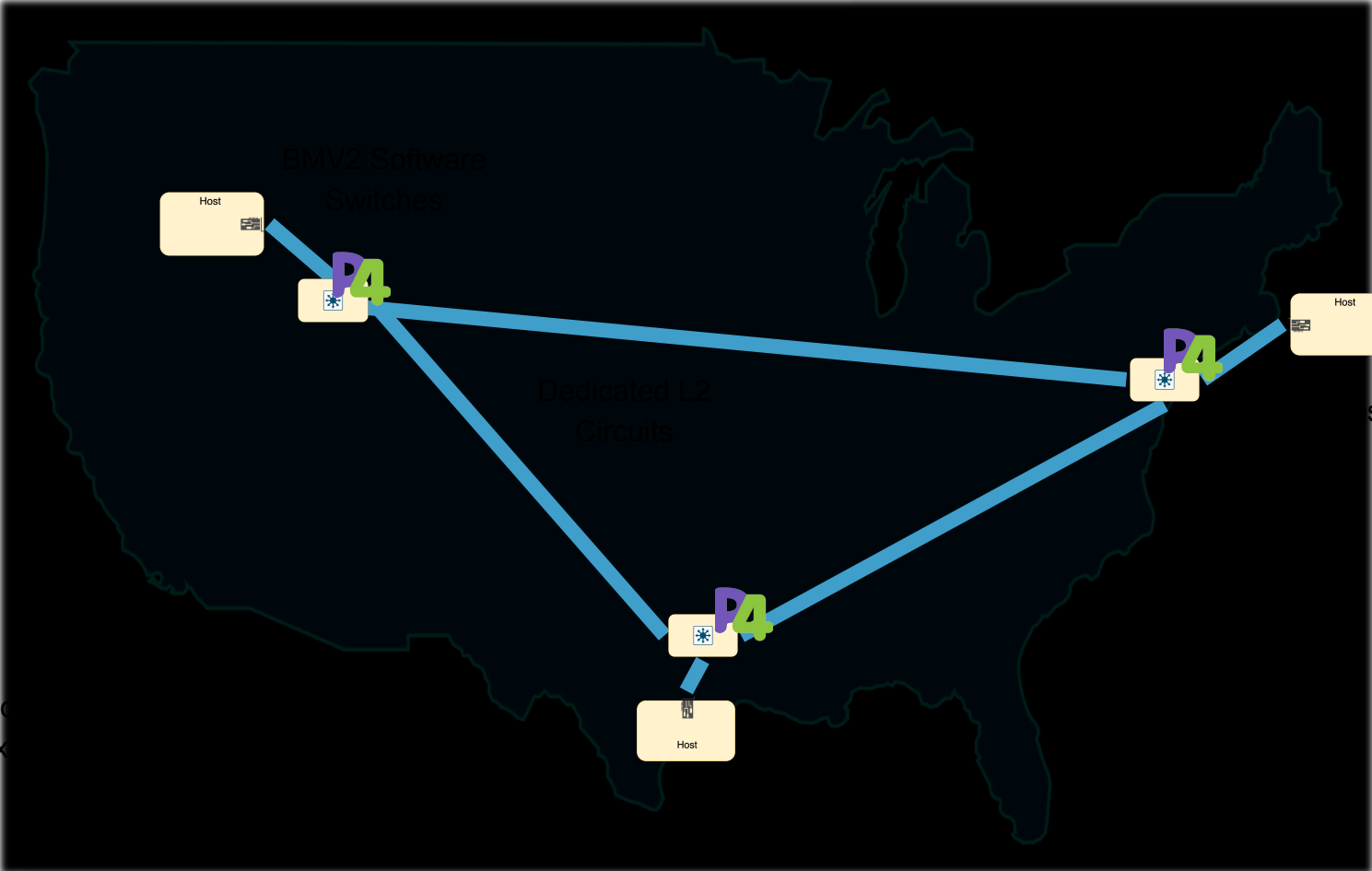
# FABRIC: P4Lang Tutorials



- Logical View from FABRIC Portal
- Sites
  - TACC: Texas Advanced Computing Center (Austin)
  - MAX: Mid-Atlantic Crossroads (U. Maryland)
  - UTAH: (U. Utah)

A screenshot of a web browser displaying the FABRIC Portal. The browser's address bar shows the URL "portal.fabric-testbed.net/slices/eea0205f-3435-4f3f-a2d0-298808508db3". The page header includes navigation links for Home, Resources, Projects, Experiments, Links, and User Profile, along with a "Log out" button. A prominent orange banner contains a "Disruptive Maintenance Notice" for the period of 2/14/2022 00:00:00 EST until 2/28/2022 00:00:00 EST. The main content area is titled "Slice Viewer" and features a network diagram with three main clusters labeled MAX, TACC, and UTAH. Each cluster contains VMs, Network Services, and L2TP components. The diagram is connected by lines representing network links. To the right of the diagram is a "Details" sidebar with fields for Name (s2-s2\_switch\_nic), Model (ConnectX-5), and Detail (Mellanox ConnectX-5 Dual Port 10/25GbE). Buttons for "Download in PNG" and "Download in JSON" are located above the diagram. A "Back to Slice List" button is in the top right corner of the slice viewer area.

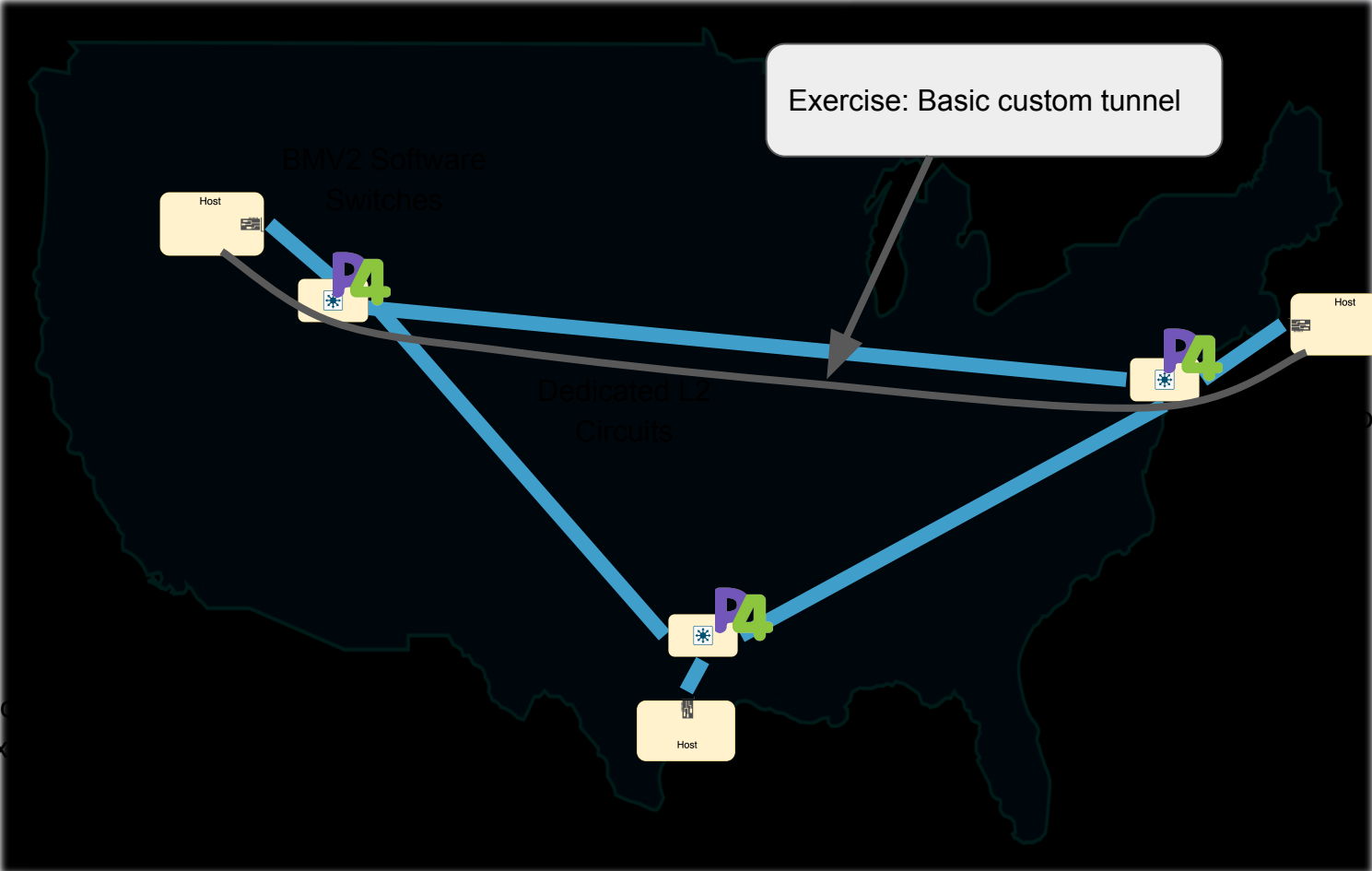




- Tofino
- Xilinx



Exercise: Basic custom tunnel



- Tofino
- Xilinx



# Summary: P4 Experiments on FABRIC

- User controlled P4 switches in the network core
- Dedicated L2 circuits
- Hardware P4 (Tofino and Xilinx) coming soon
- Software P4 (BMV2 and Tofino) possible now
- Jupyter notebooks to streamline deployment

# Thank You!

Questions?

Visit <https://fabric-testbed.net>

Learn more, and Join the Forum at <https://learn.fabric-testbed.net>

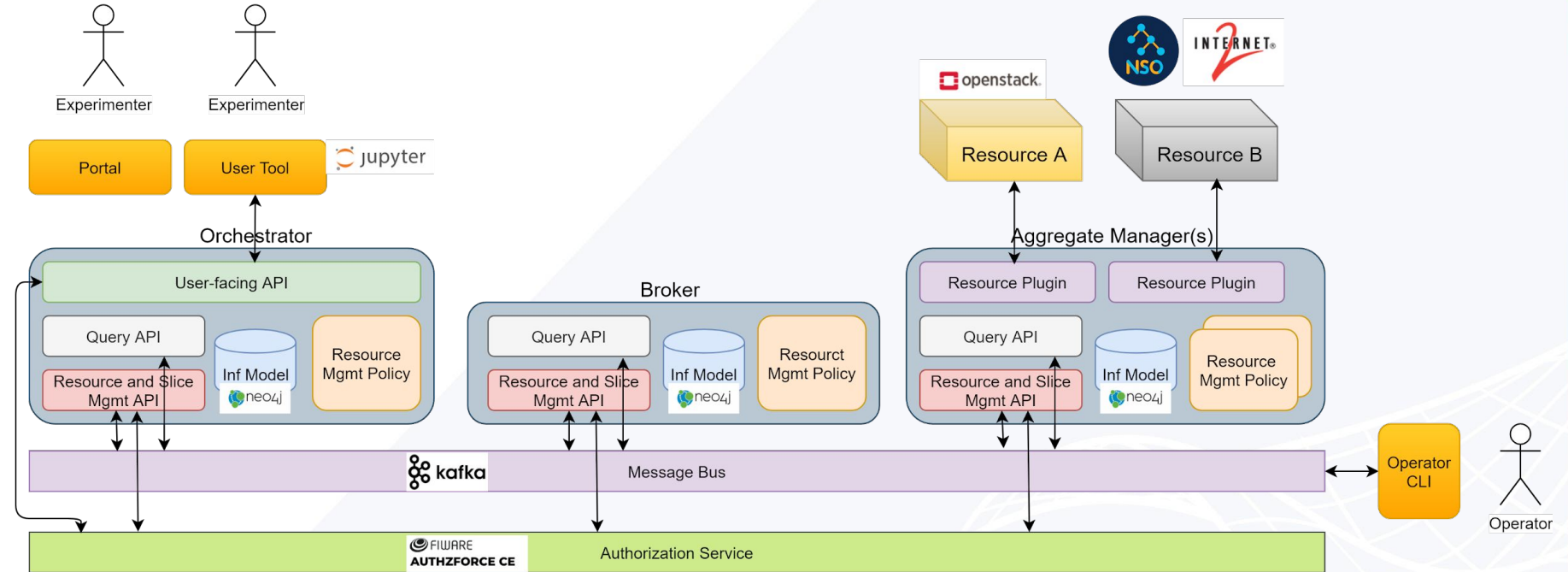
Ask [info@fabric-testbed.net](mailto:info@fabric-testbed.net)

FABRIC Software: <https://github.com/fabric-testbed>



This work is funded by  
NSF grant CNS-1935966

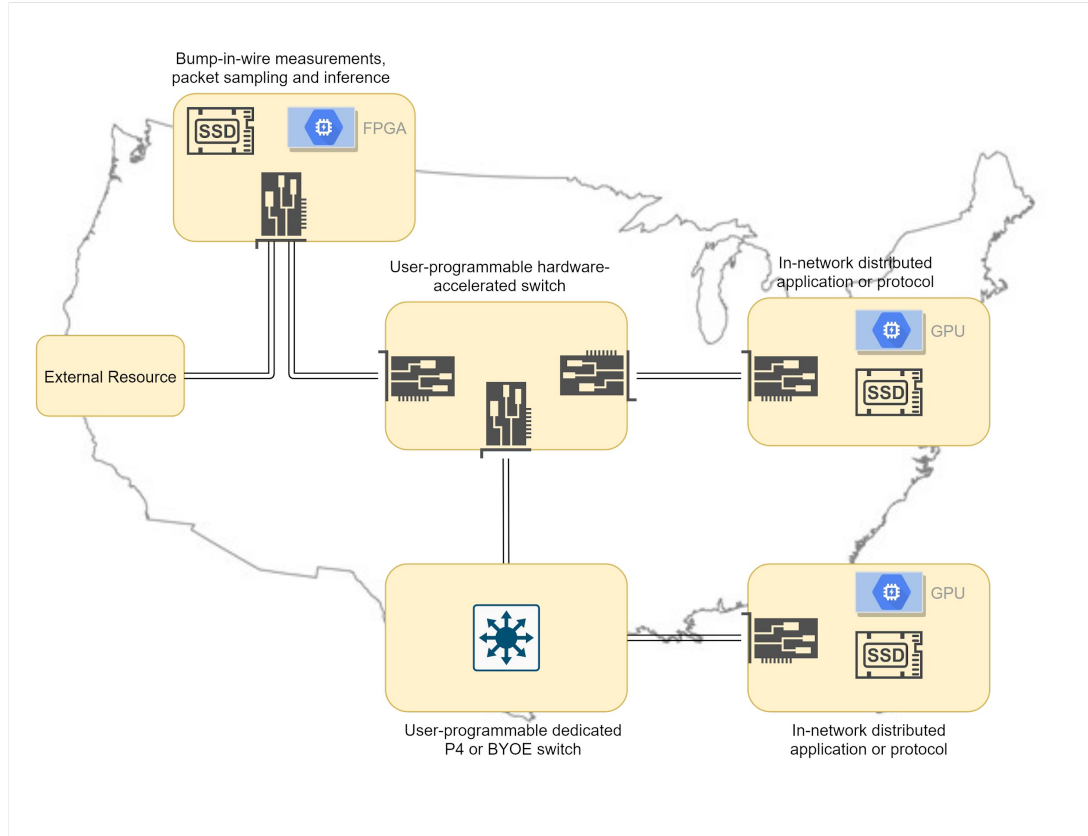
# Control Framework (CF) Components

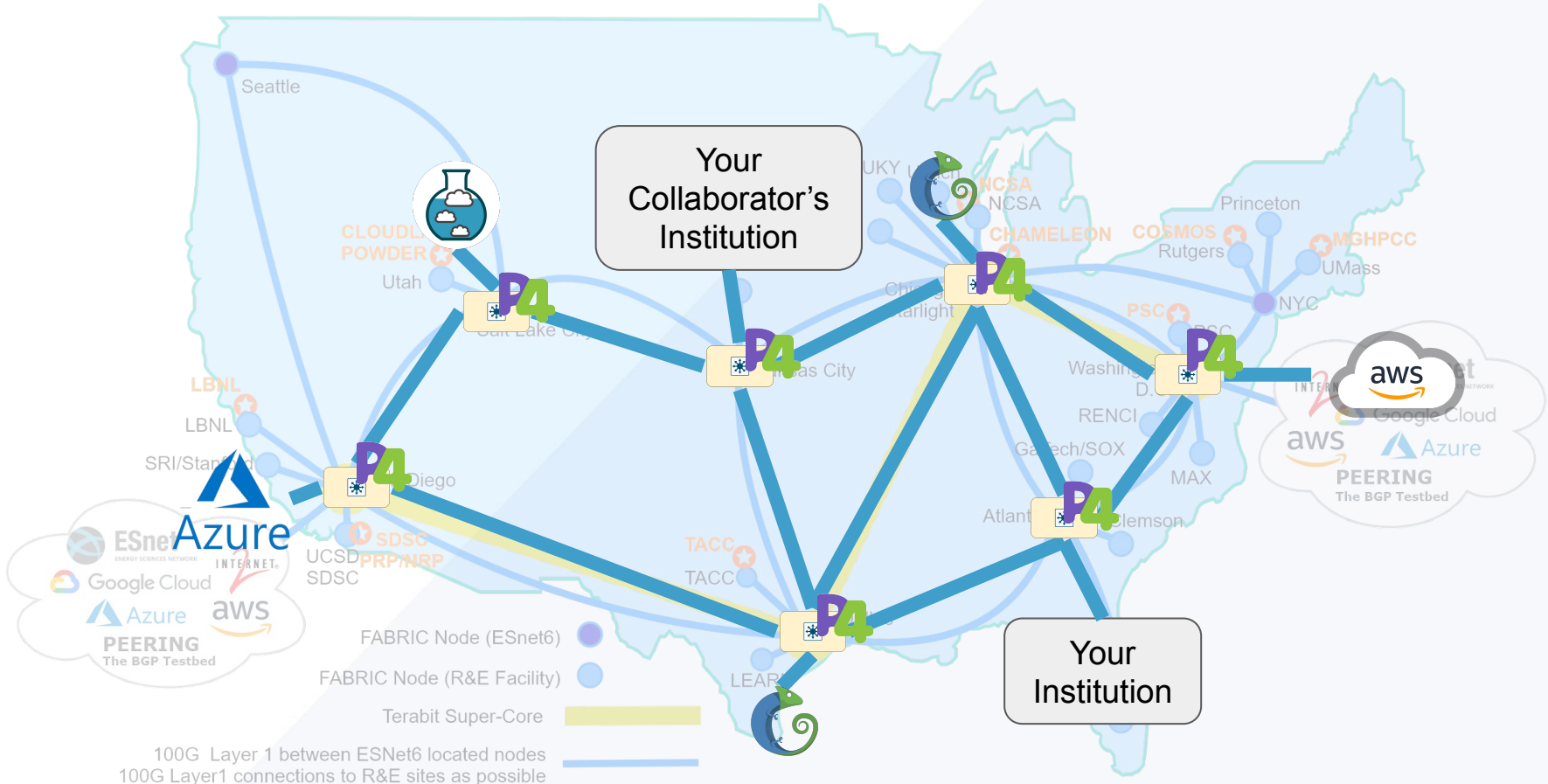


# Example FABRIC Use-case Scenarios

Examples of potential uses:

- **Bump-in-wire** measurements and packet sampling at high bit rates (25, 40, 100, 100+ Gbps)
- **Hardware-accelerated switching** using Smart NICs, FPGA NICs or P4 switches in individual nodes
- **Hosting in-network applications** and stateful architectures using a combination of storage and compute resources in individual nodes
- **In-network inference**, other types of accelerated computing via FPGAs and GPUs
- **Connect experiments to external facilities** like IoT, 5G, cloud testbeds, public clouds and HPC resources.
- **Deploy non-IP protocols** on top of wide-area L2 topologies, that may include in-network processing and storage



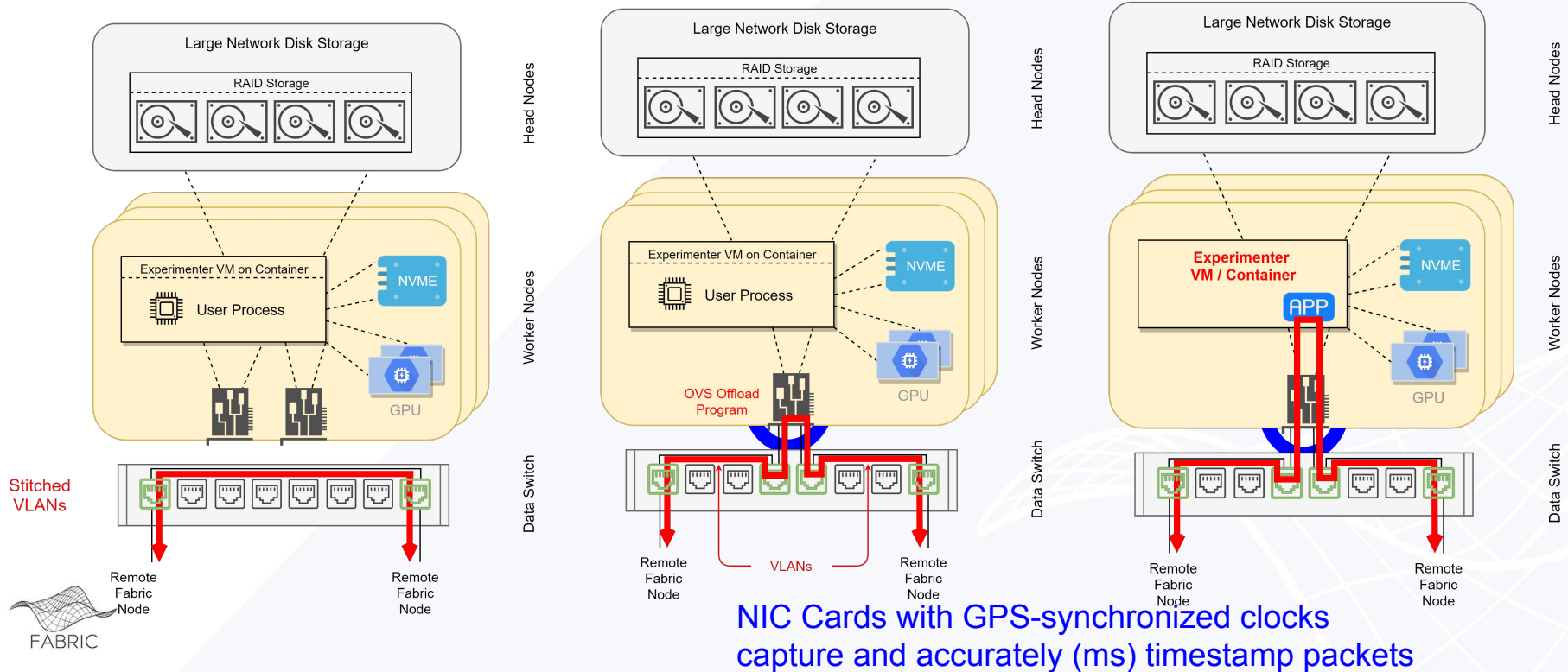


100G Layer 1 between ESNet6 located nodes  
 100G Layer1 connections to R&E sites as possible





# FABRIC Use Cases (Revisited)



NIC Cards with GPS-synchronized clocks capture and accurately (ms) timestamp packets