Cyberinfrastructure Futures
IaaS, SDN, S-DMZ, and All That
Linking Virtual Infrastructure to Science

Jeff Chase and Paul Ruth

Duke University / RENCI

renci

NSF

geni
Exploring Networks
of the Future

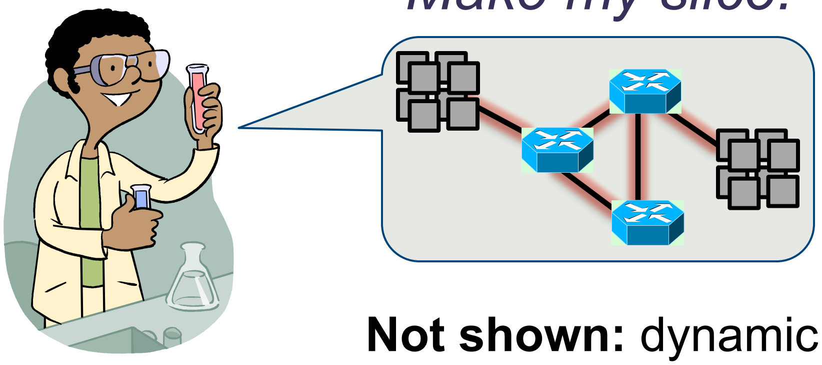
usignite

Federated substrate with end-to-end virtualized “slices”

- **GENI**: NSF initiative for network testbed integration
- It has grown into an architectural blueprint for:
  - **Federated IaaS** (Infrastructure-as-a-Service) clouds
  - Delivery platform for **next-generation network apps**
  - **Clouds integrated with advanced networks**
- Campus deployment with “GENI Racks”
- RENCI ExoGENI testbed

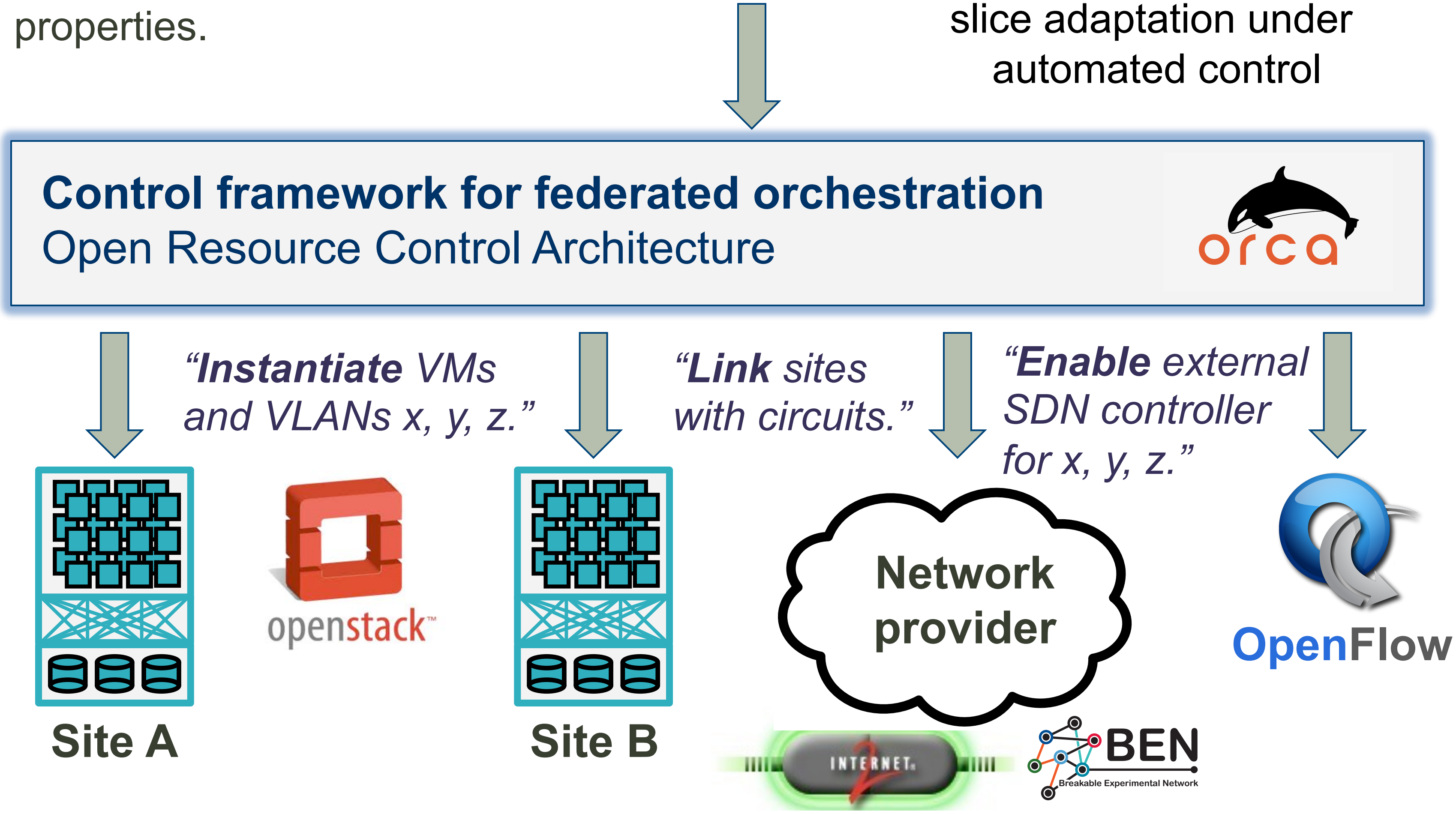
The Big Picture: Cartoon Version

Slice: an end-to-end virtual infrastructure with configurable properties.



"Make my slice."

Not shown: dynamic slice adaptation under automated control



Control framework for federated orchestration

Open Resource Control Architecture

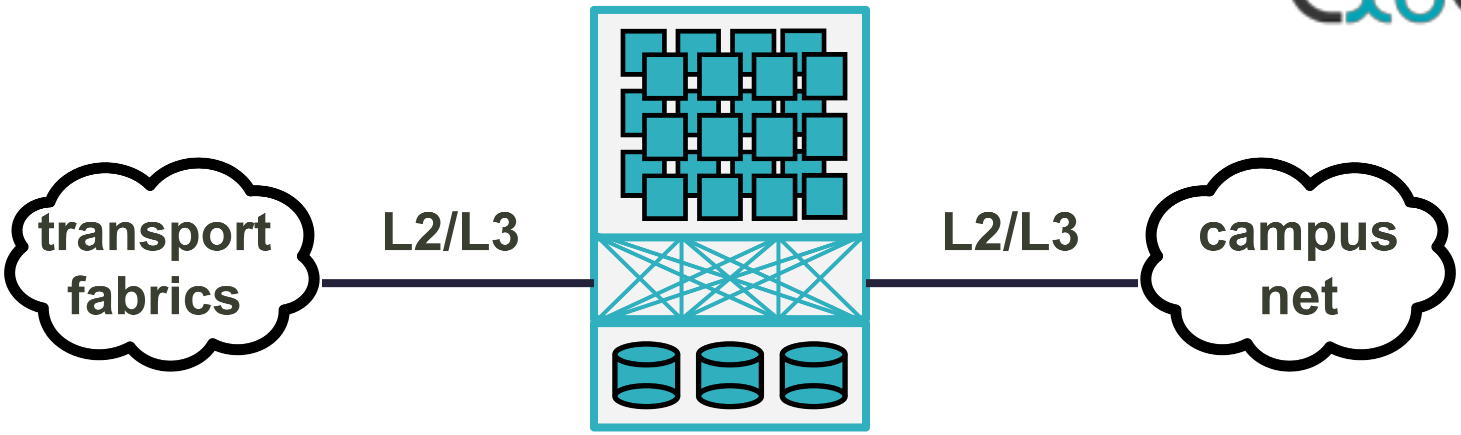
*"**Instantiate** VMs and VLANs x, y, z."*

*"**Link** sites with circuits."*

*"**Enable** external SDN controller for x, y, z."*

ExoGENI.net

ExoGENI



ExoGENI Rack

A packaged small-scale cloud site for a campus, lab, or PoP.

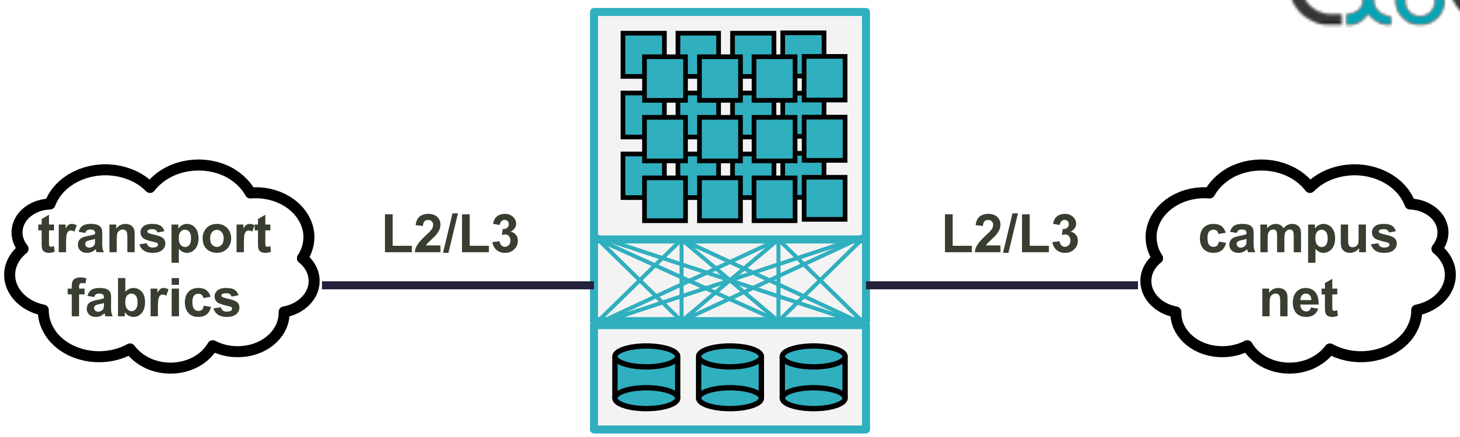
control: local or remote
policy: local or remote

openstack™

OpenFlow

ExoGENI.net

ExoGENI



ExoGENI Rack

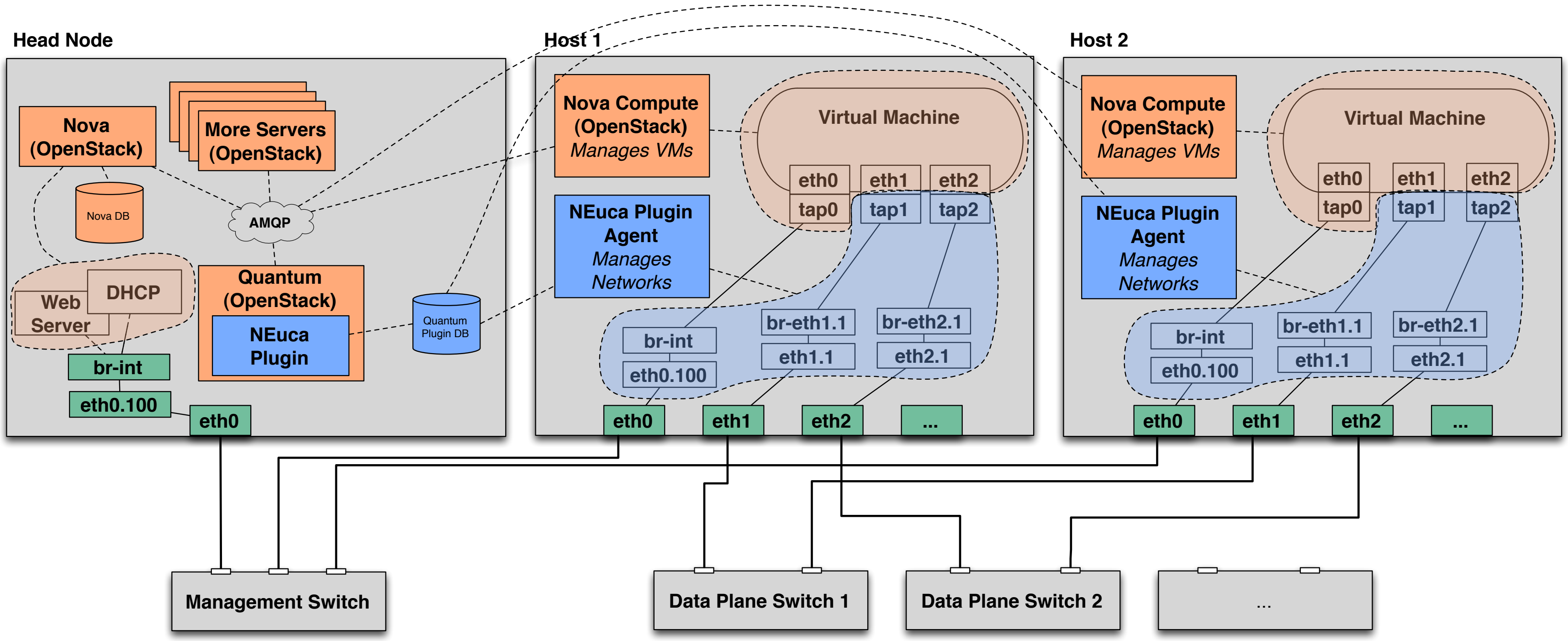
openstack

Dynamically sliverable
Software-defined network (SDN)
Per-slice network control
Bare-metal provisioning

OpenFlow

XCAT

OpenStack Quantum extensions for dynamic virtual networks

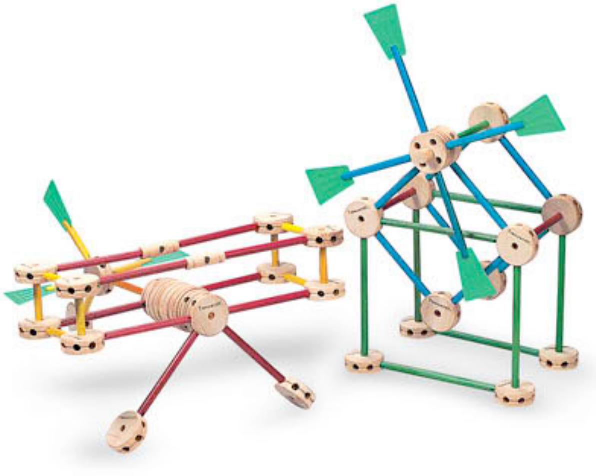


openstack™

ExoGENI uses OpenStack as a **component**, but it needs **hooks** for dynamic multi-homed network topologies.

What can we build?

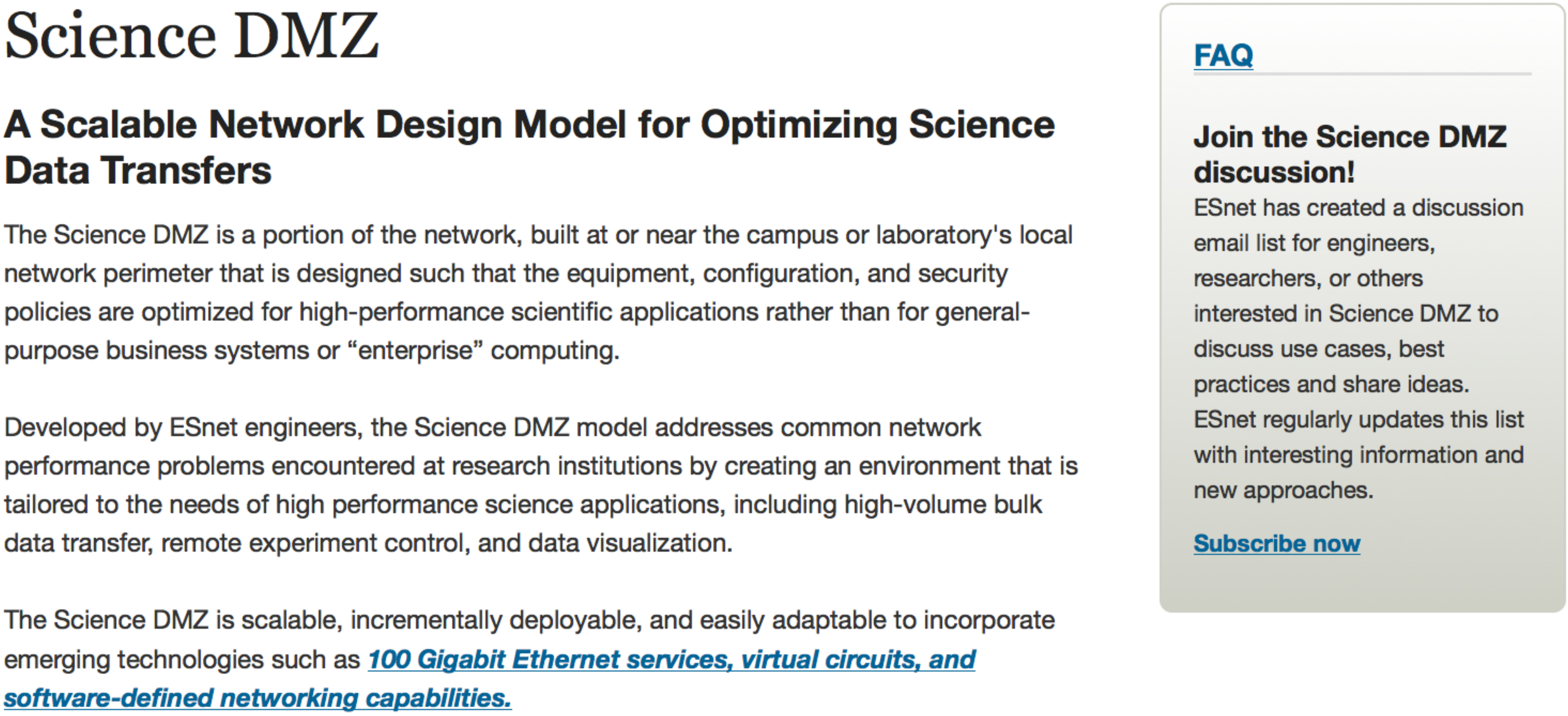
- Slices built to order from dynamic virtual infrastructure
- Link resources with advanced networks across multiple campuses



- **Integrate dynamic virtual infrastructure and transport network connectivity into campus computing environments.**

Science Network: the es.net view

http://fasterdata.es.net/science-dmz/



Science DMZ

A Scalable Network Design Model for Optimizing Science Data Transfers

The Science DMZ is a portion of the network, built at or near the campus or laboratory's local network perimeter that is designed such that the equipment, configuration, and security policies are optimized for high-performance scientific applications rather than for general-purpose business systems or "enterprise" computing.

Developed by ESnet engineers, the Science DMZ model addresses common network performance problems encountered at research institutions by creating an environment that is tailored to the needs of high performance science applications, including high-volume bulk data transfer, remote experiment control, and data visualization.

The Science DMZ is scalable, incrementally deployable, and easily adaptable to incorporate emerging technologies such as ***100 Gigabit Ethernet services, virtual circuits, and software-defined networking capabilities.***

FAQ

Join the Science DMZ discussion!

ESnet has created a discussion email list for engineers, researchers, or others interested in Science DMZ to discuss use cases, best practices and share ideas. ESnet regularly updates this list with interesting information and new approaches.

Subscribe now

NSF funds campus network upgrades under CC-NIE program, with inducements to adopt "Science DMZ".

Science DMZ: what does it mean?

- S-DMZ is “optimized for high-performance scientific apps”.
  - Once connectivity is established, the science traffic bypasses inline security checks, e.g., firewalls, IDS/IPS.
  - Science traffic is directed on isolated paths that are provisioned to carry that traffic. Ideally this provisioning is **dynamic**.
- S-DMZ is “at or near the campus local network perimeter”.
  - Is it inside or outside the perimeter? How to cross? (???)
- **A key question**: how to connect external dynamic transport circuits into campus resources?
  - Any solution might be called “Science DMZ”.
  - But what is the **right** solution?

[A personal take]

Science networks are virtual

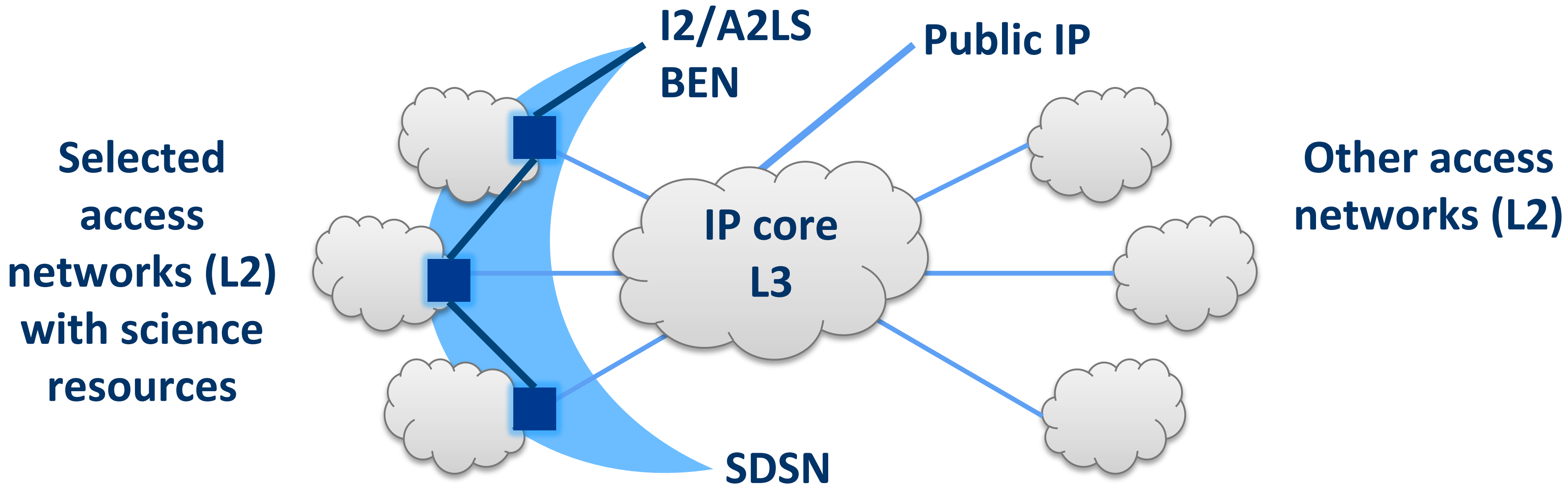
- We want to enable construction of purpose-built science networks as we need them. Let's call them **scinets**.
- A scinet is a **virtual** network for science traffic, with some useful degree of isolation and control.
  - Dynamic circuits → dynamic virtual topology
  - Path selection and bandwidth allocation (performance isolation)
  - Logical isolation, e.g., by VLANs, VPNs, or other SDN app
  - Flexible connectivity to other internal and external networks
- **Note**: the choice to dedicate capacity for a scinet is **dynamic**.
  - S-DMZ should not imply an inherently "segregated" network.
  - We believe in integrated campus networks!

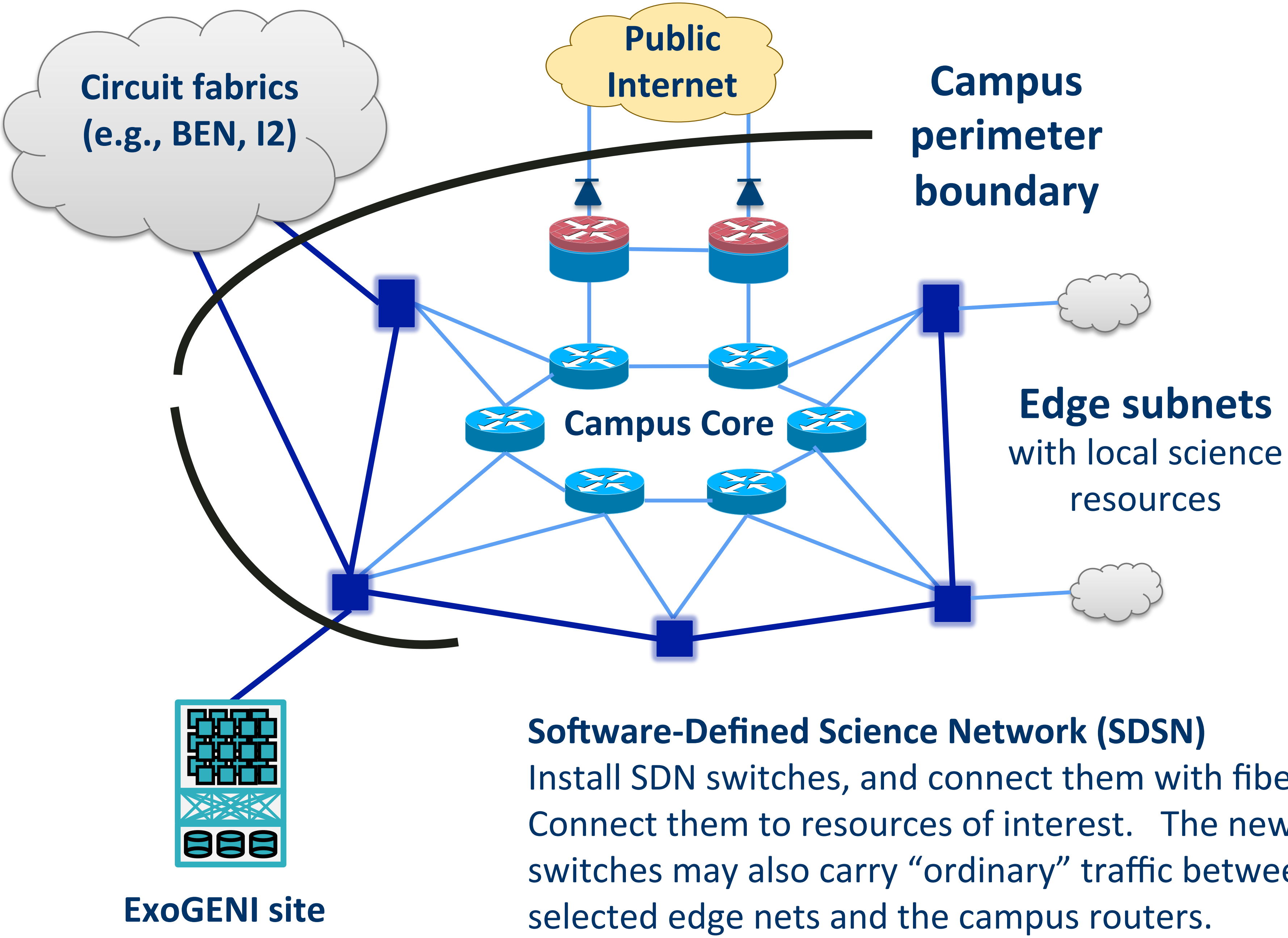
A GENI perspective

- A scinet is similar to a [GENI] **slice**.
  - A **slice** is a dynamic virtual topology created on demand, with varying (but specified) degrees of isolation and assurance.
  - Allocate network in tandem with edge resources: a slice is virtual compute nodes linked by a virtual network.
- But to build real scinets, we need to extend slices to more infrastructure that is “outside of GENI”.
  - campus networks and clusters
  - cyberinfrastructure resources off campus
  - batch computing systems, data repositories
- **Or whatever**. We just want more dynamic control of connectivity and traffic flow…and edge resources too.

Duke approach: summary

- Duke’s **software-defined science network** (SDSN) is an upgrade integrated with our existing campus network.
- The SDSN is a group of linked SDN switches in selected campus access networks: the MPLS-VPN IP core is unchanged.
- The SDSN is a switched Ethernet network with dynamic VLANs. A new SDN **ramps** service enables controlled dynamic interconnections among VLANs, subject to various L3 constraints.


I2/A2LS
BEN
Public IP
Selected access networks (L2) with science resources
IP core
L3
Other access networks (L2)
SDSN



Software-Defined Science Network (SDSN)
Install SDN switches, and connect them with fiber. Connect them to resources of interest. The new switches may also carry “ordinary” traffic between selected edge nets and the campus routers.

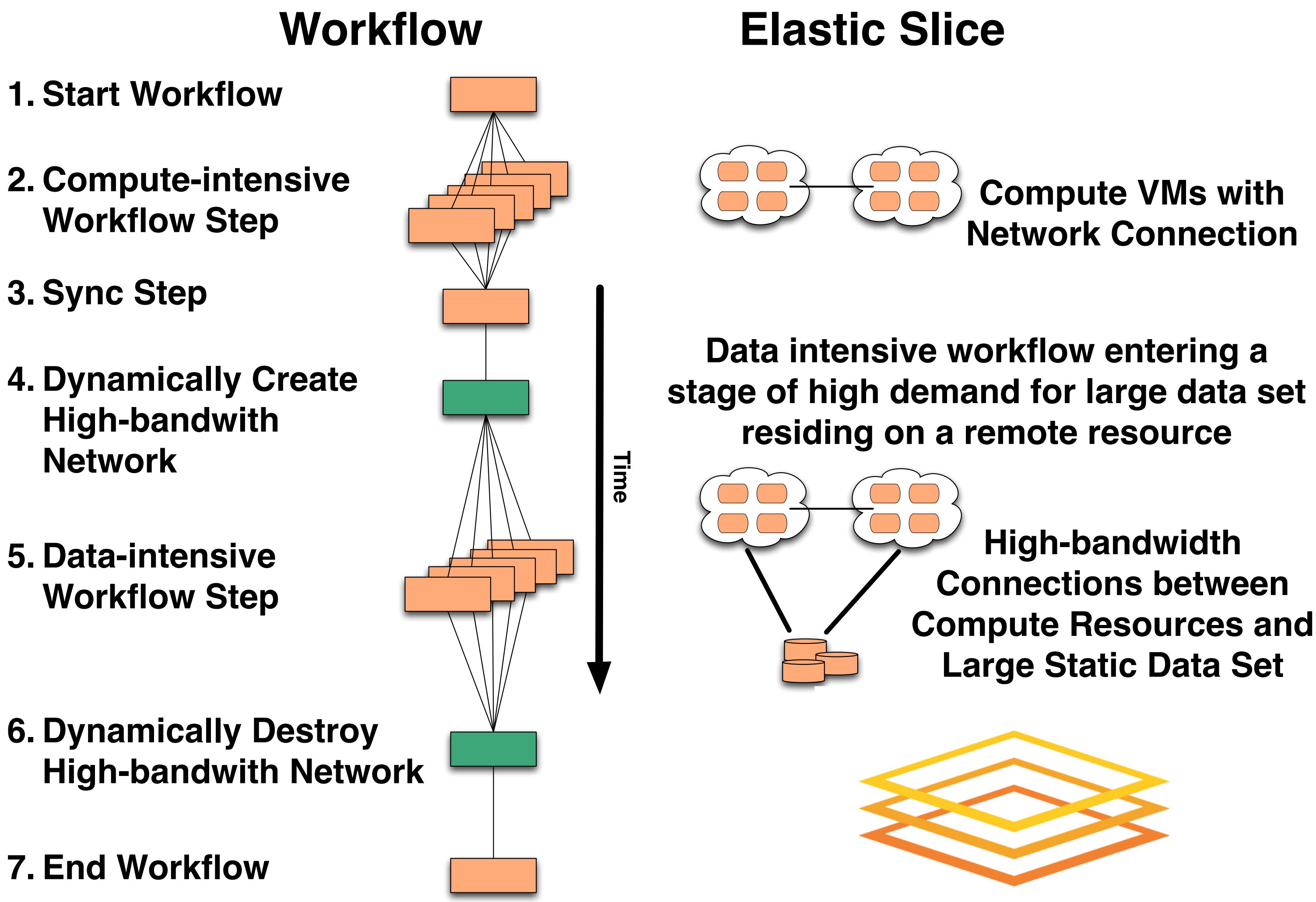
So…

- There's a lot of potential here, but…
  - "User-friendly" is still somewhere over the rainbow.
- Can we integrate dynamic virtual infrastructure seamlessly with mature CI environments like OSG?
- How to orchestrate next-gen science apps?
  - Allocate infrastructure on demand
  - Link sites and storage with dynamic pipes
  - Instantiate middleware: HT-Condor etc.
  - Link to campus resources and deployed systems, e.g., OSG

Scientific Workflows

- Workflow Management Systems
  - Pegasus, Custom scripts, etc.
- Lack of tools to integrate with dynamic infrastructures
  - Orchestrate the infrastructure in response to application
  - Integrate data movement with workflows for optimized performance
  - Manage application in response to infrastructure
- Scenarios
  - Computational with varying demands
  - Data-driven with large static data-set(s)
  - Data-driven with large amount of input/output data

Multi-domain networked cloud apps



Workflow

1. Start Workflow
2. Compute-intensive Workflow Step
3. Sync Step
4. Dynamically Create High-bandwith Network
5. Data-intensive Workflow Step
6. Dynamically Destroy High-bandwith Network
7. End Workflow

Elastic Slice

Compute VMs with Network Connection

Data intensive workflow entering a stage of high demand for large data set residing on a remote resource

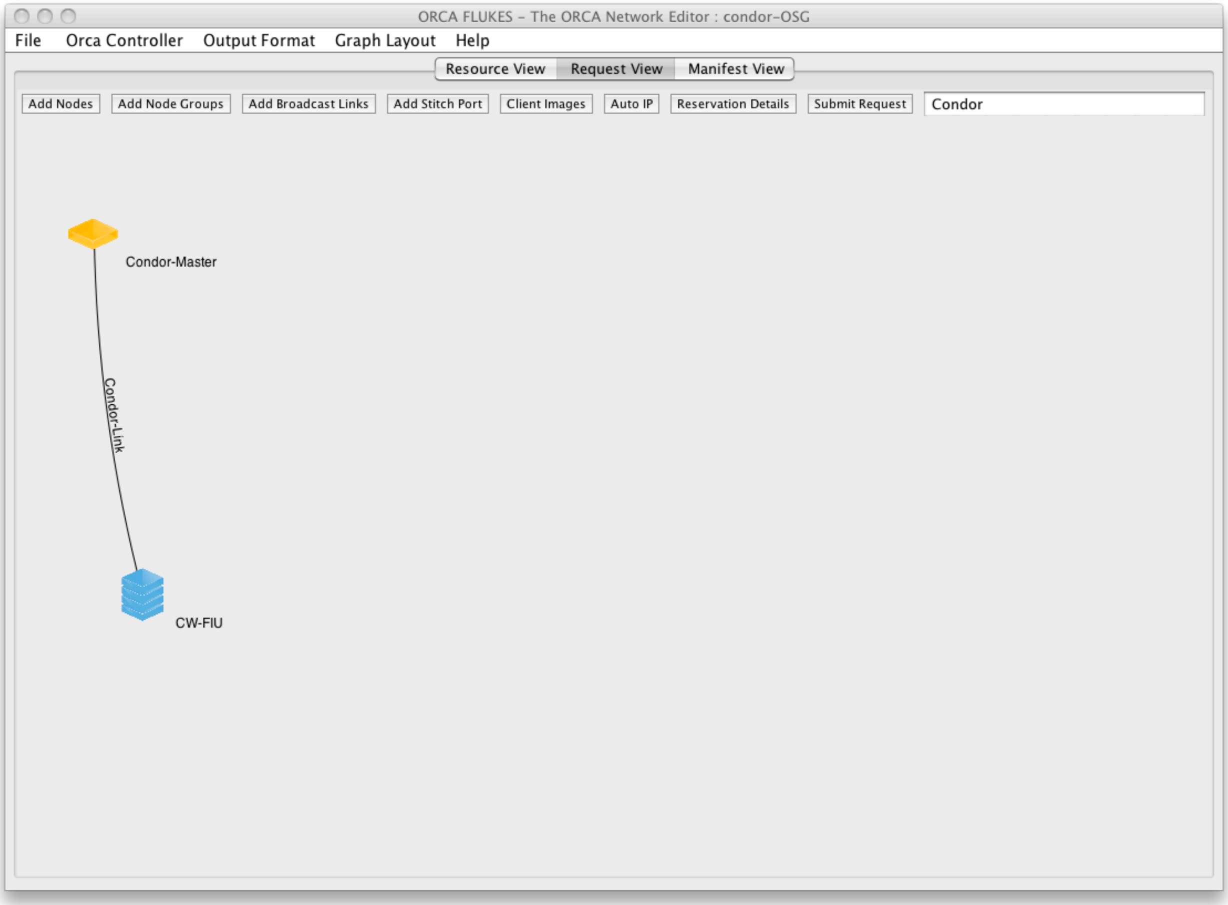
High-bandwidth Connections between Compute Resources and Large Static Data Set

Open Science Grid

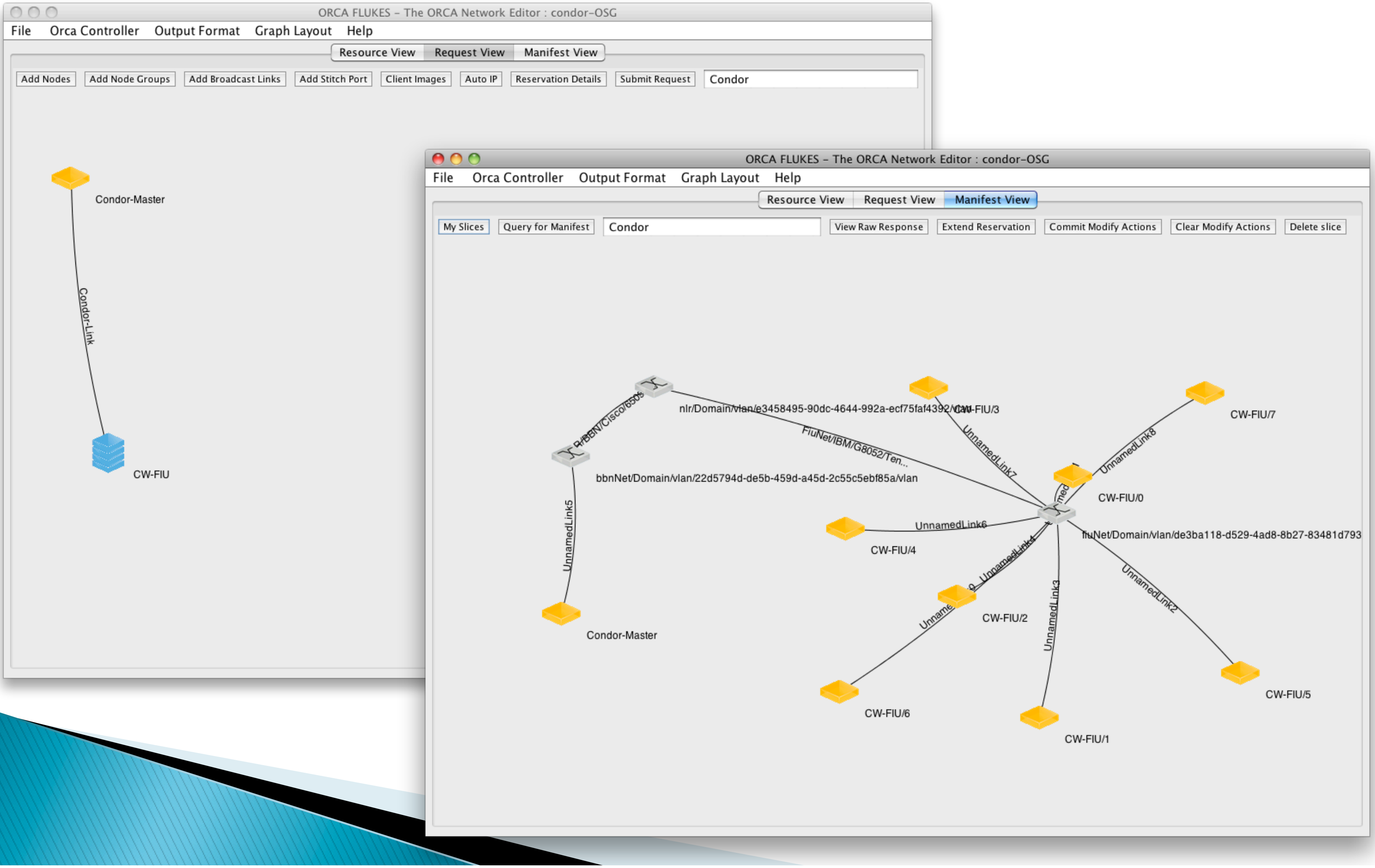
What does this have to do with OSG?

- OSG makes it easy to submit jobs/workflows to remote shared compute resources.
- ExoGENI makes it easy to create custom private virtual networked infrastructure.
- OSG with ExoGENI allows for:
  - Cloud bursting OSG jobs into ExoGENI when custom infrastructure is required.
  - Cloud bursting ExoGENI applications into OSG when massive compute power is required.
  - Linking multiple OSG sites with temporary network topologies.
  - Linking OSG sites with large data repositories.

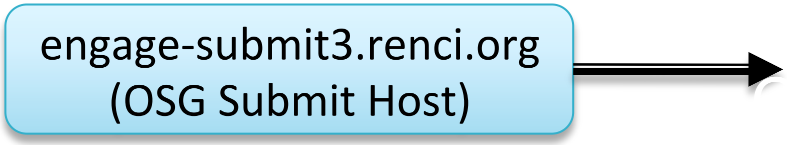
Example: HTCondor



Example: HTCondor

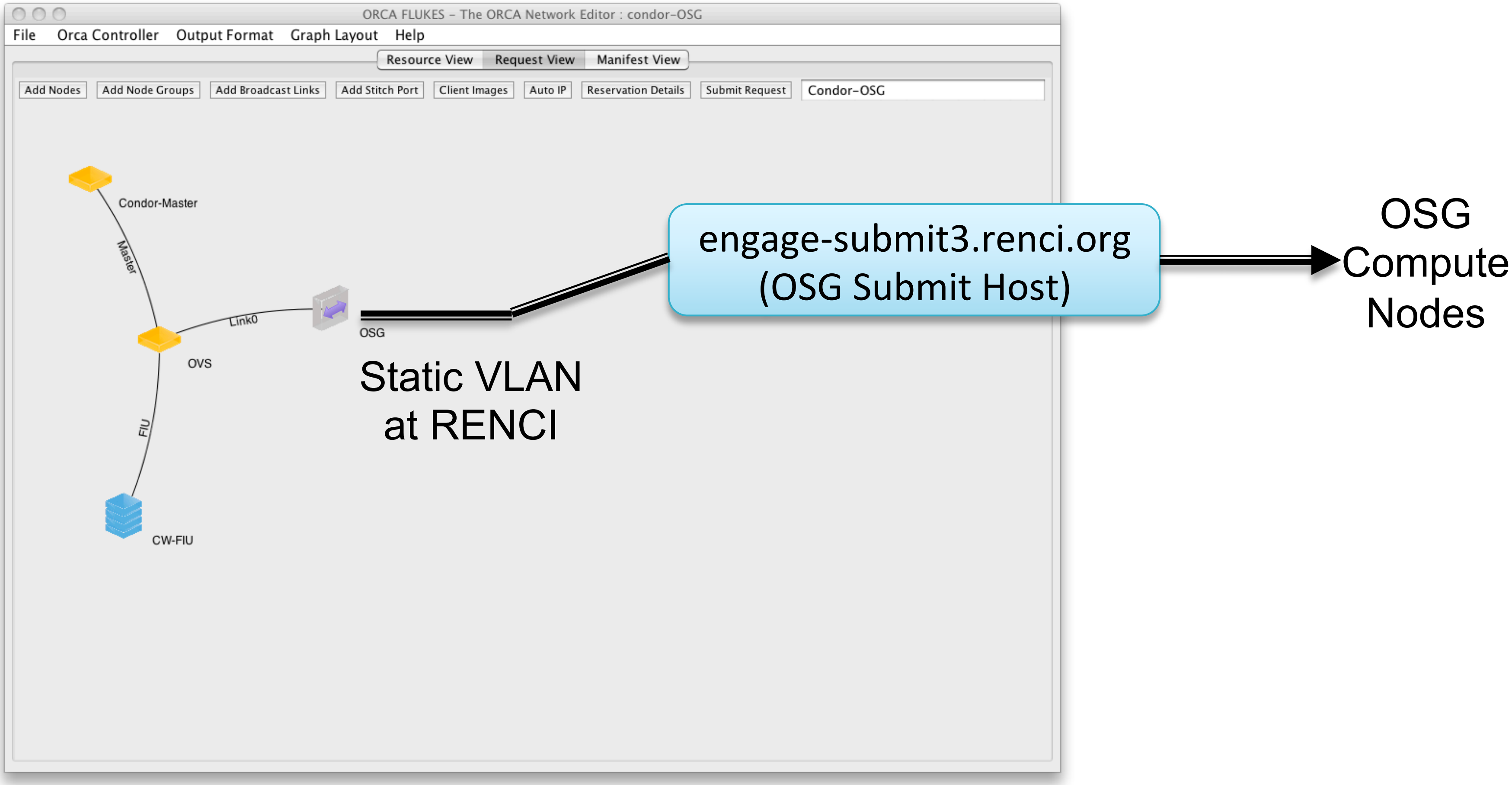


Example: OSG



engage-submit3.renci.org
(OSG Submit Host)

Example: OSG



ORCA FLUKES – The ORCA Network Editor : condor-OSG

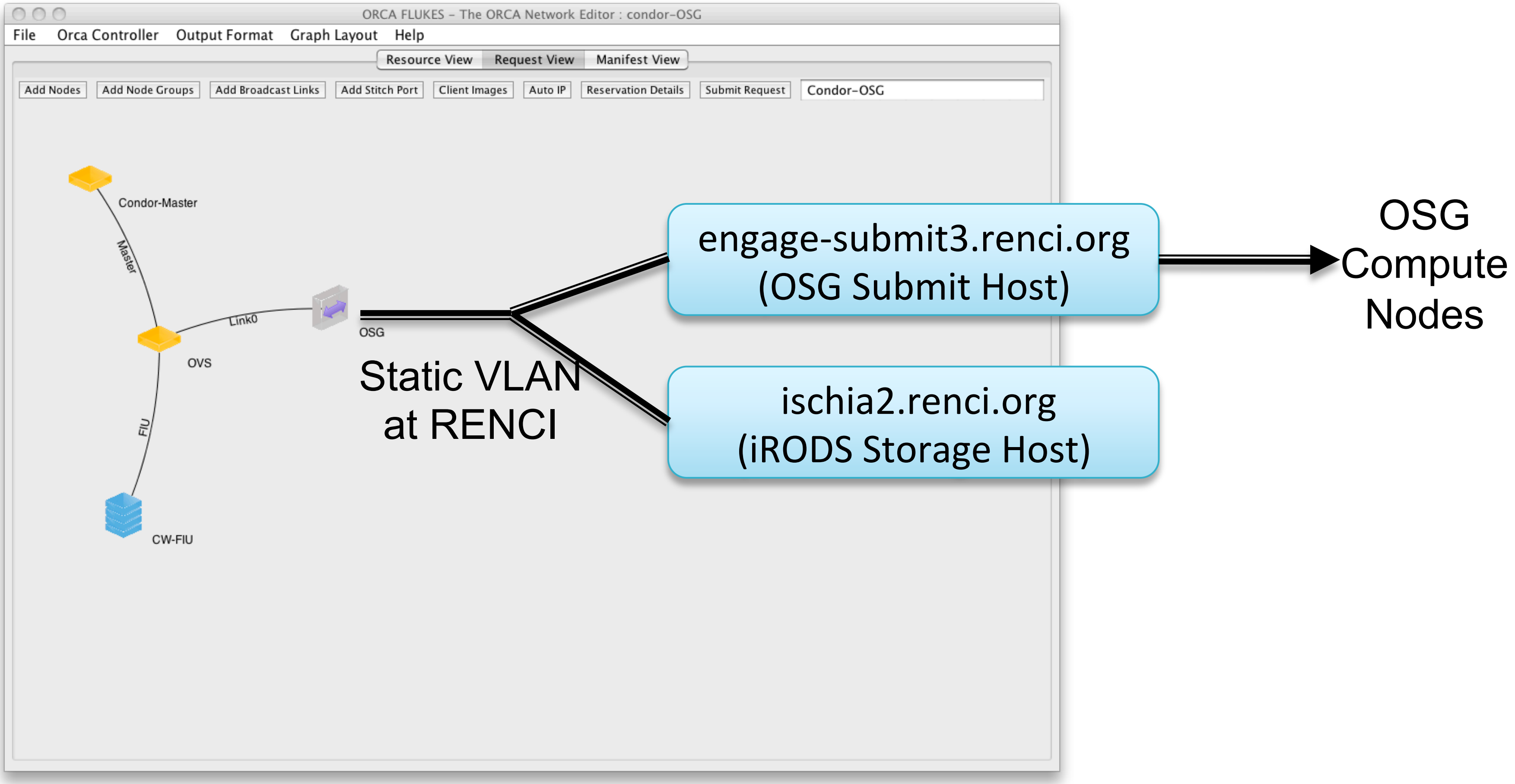
Condor-OSG

Static VLAN at RENCI

engage-submit3.renci.org (OSG Submit Host)

OSG Compute Nodes

Example: OSG



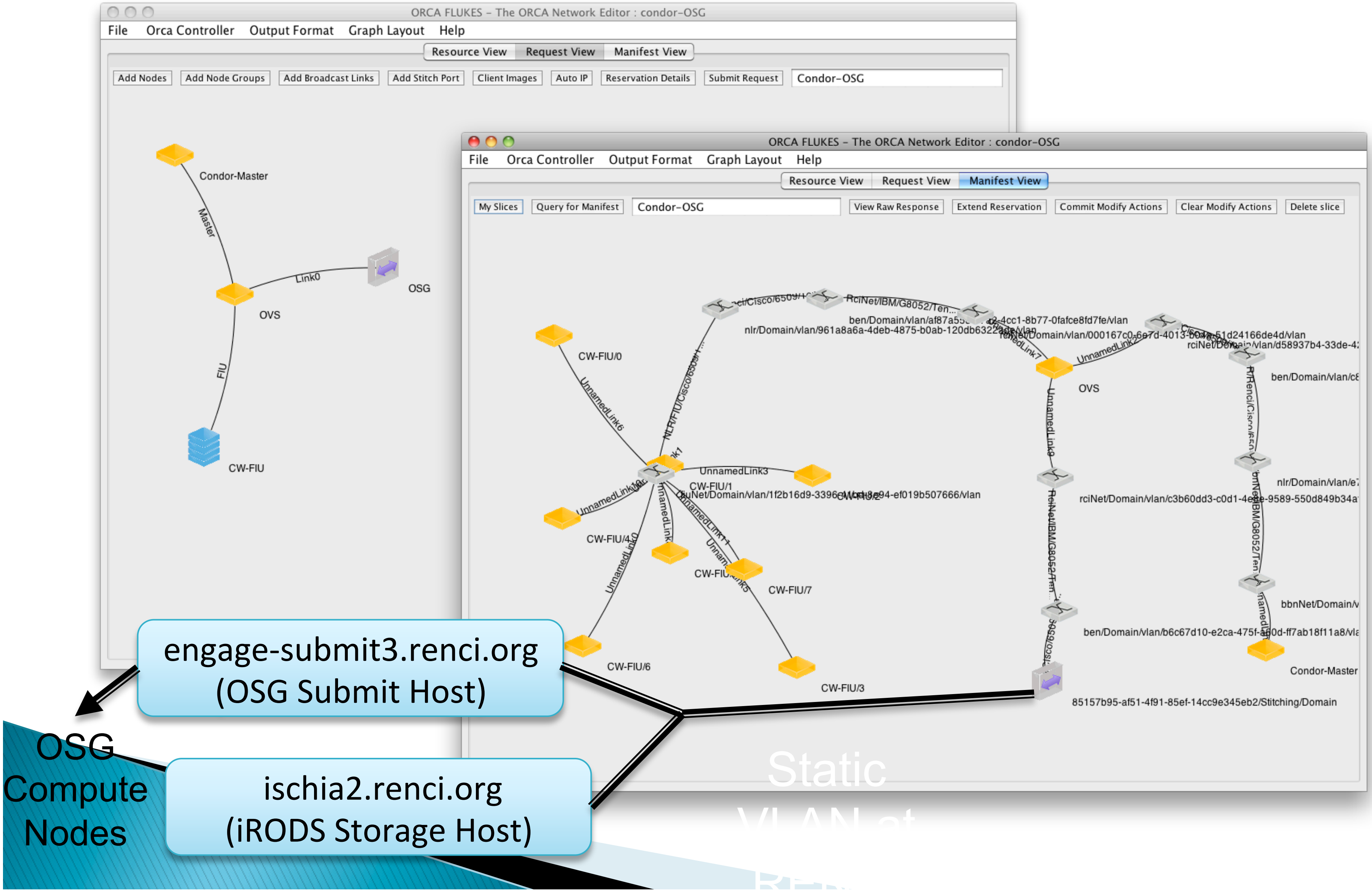
Static VLAN at RENCI

engage-submit3.renci.org
(OSG Submit Host)

ischia2.renci.org
(iRODS Storage Host)

OSG Compute Nodes

Example: OSG

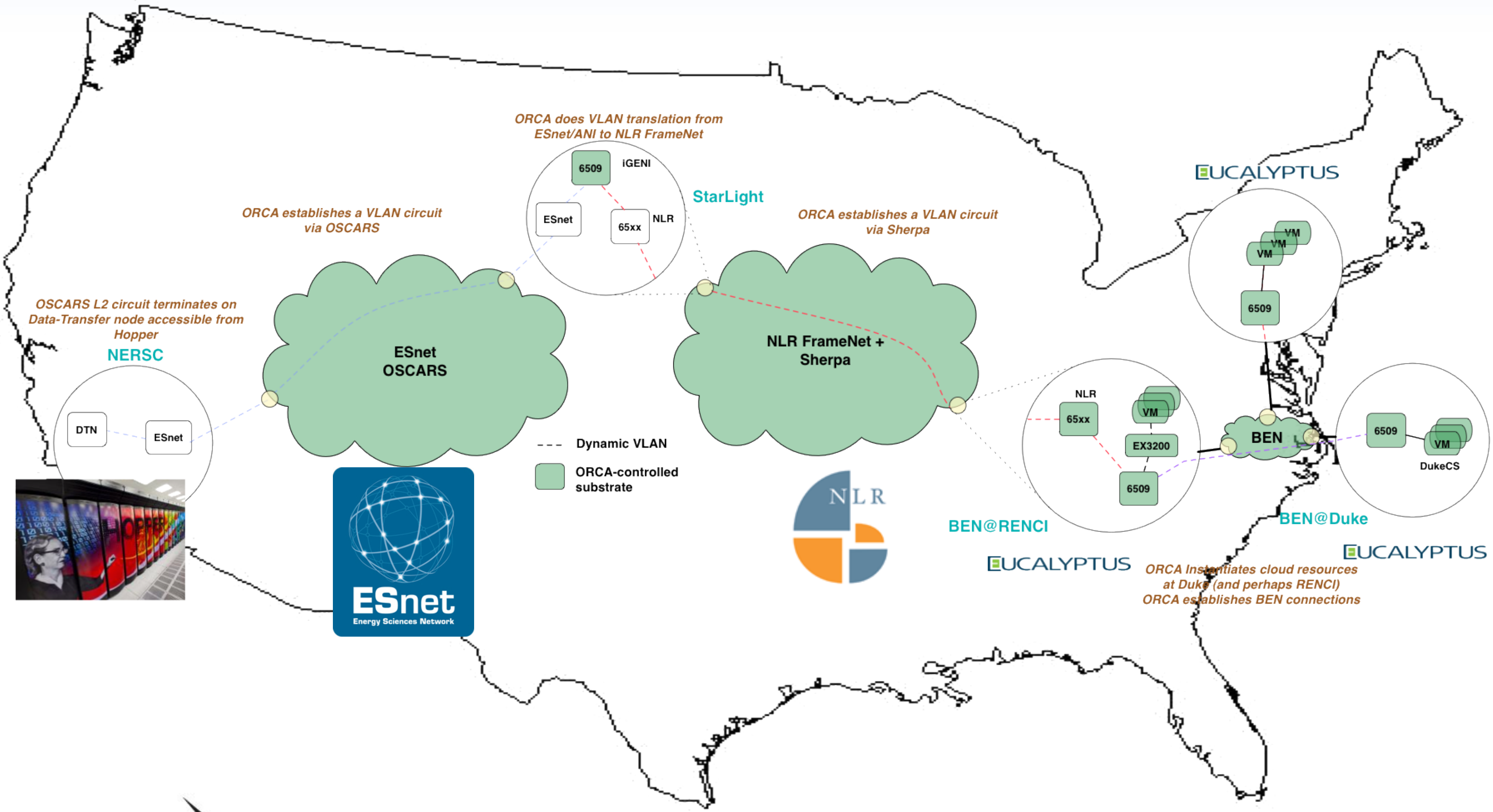


engage-submit3.renci.org
(OSG Submit Host)

ischia2.renci.org
(iRODS Storage Host)

OSG
Compute
Nodes

SC11 Demo: Solar Fuels Workflow



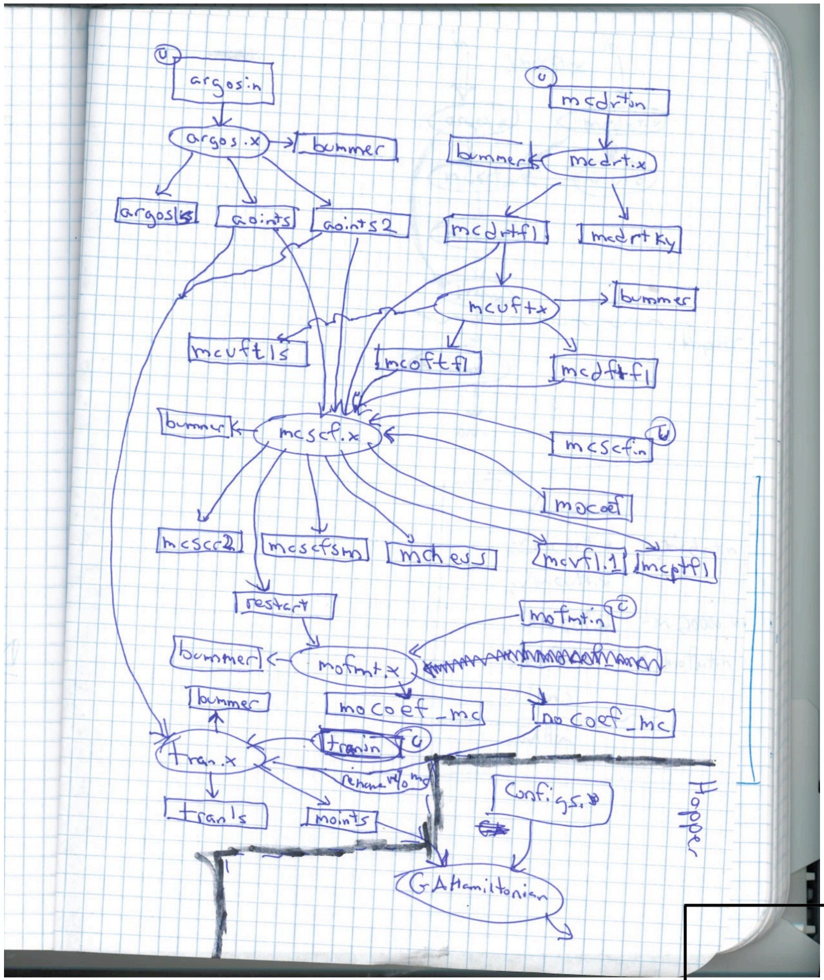
OSCARS L2 circuit terminates on Data-Transfer node accessible from Hopper

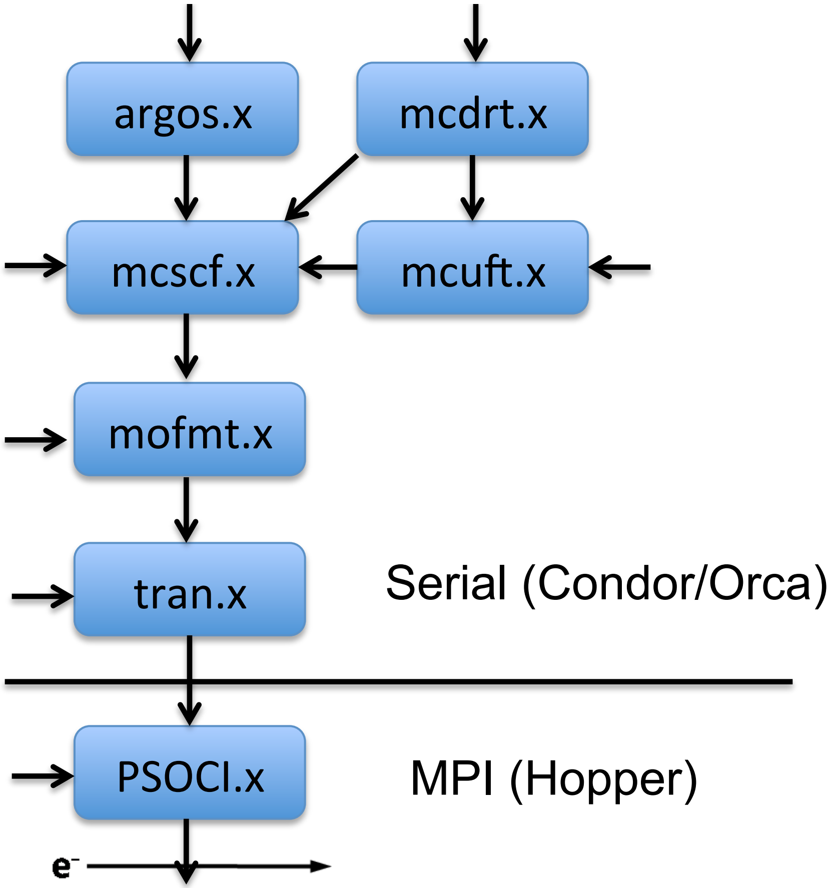
ORCA establishes a VLAN circuit via OSCARS

ORCA does VLAN translation from ESnet/ANI to NLR FrameNet

ORCA establishes a VLAN circuit via Sherpa

ORCA Instantiates cloud resources at Duke (and perhaps RENCI) ORCA establishes BEN connections





Renci
Renaissance Computing Institute
Catalyst for Innovation

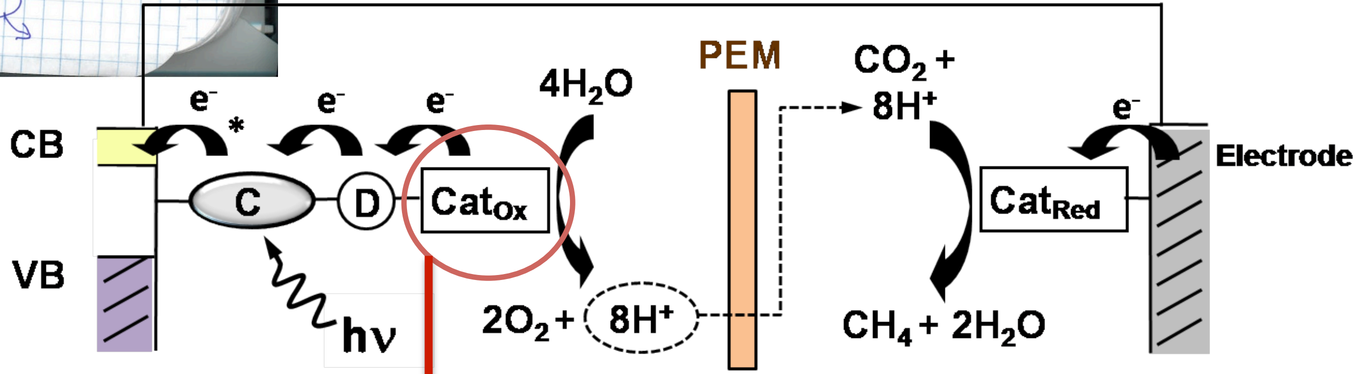
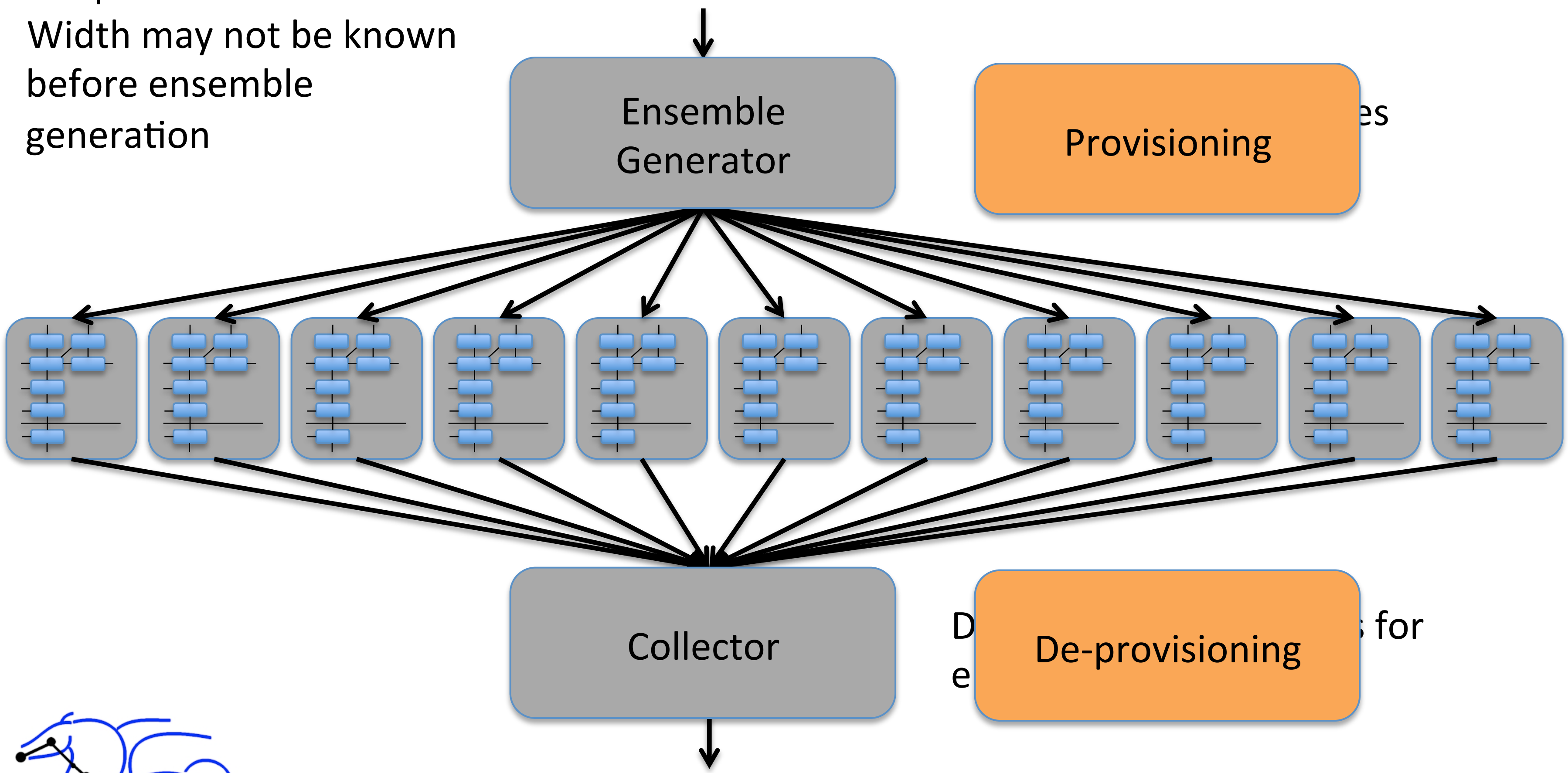


Image provided by UNC-CH EFRC: http://www.efrc.unc.edu/

Oxidation catalysts

Example Dynamic Workflow: Ensemble

- Wide step of Ensemble is temporal
- Width may not be known before ensemble generation



Ensemble Generator

Provisioning

es

Collector

D

e

De-provisioning

for

pegasus

Condor

High Throughput Computing

Thanks to our sponsors

- NSF SDCI/CC-NIE:
  - OCI-1032573, OCI-1032873, OCI-1245926
- NSF Trusted Computing
  - CNS 0910653
- DOE
  - ASCR DE-SC0005286
- GENI Project Office
- Duke/OIT has two NSF-funded SDN pilots:
  - “on-ramp” services (EAGER)
  - “Expressway” buildout for big-data science (CC-NIE)