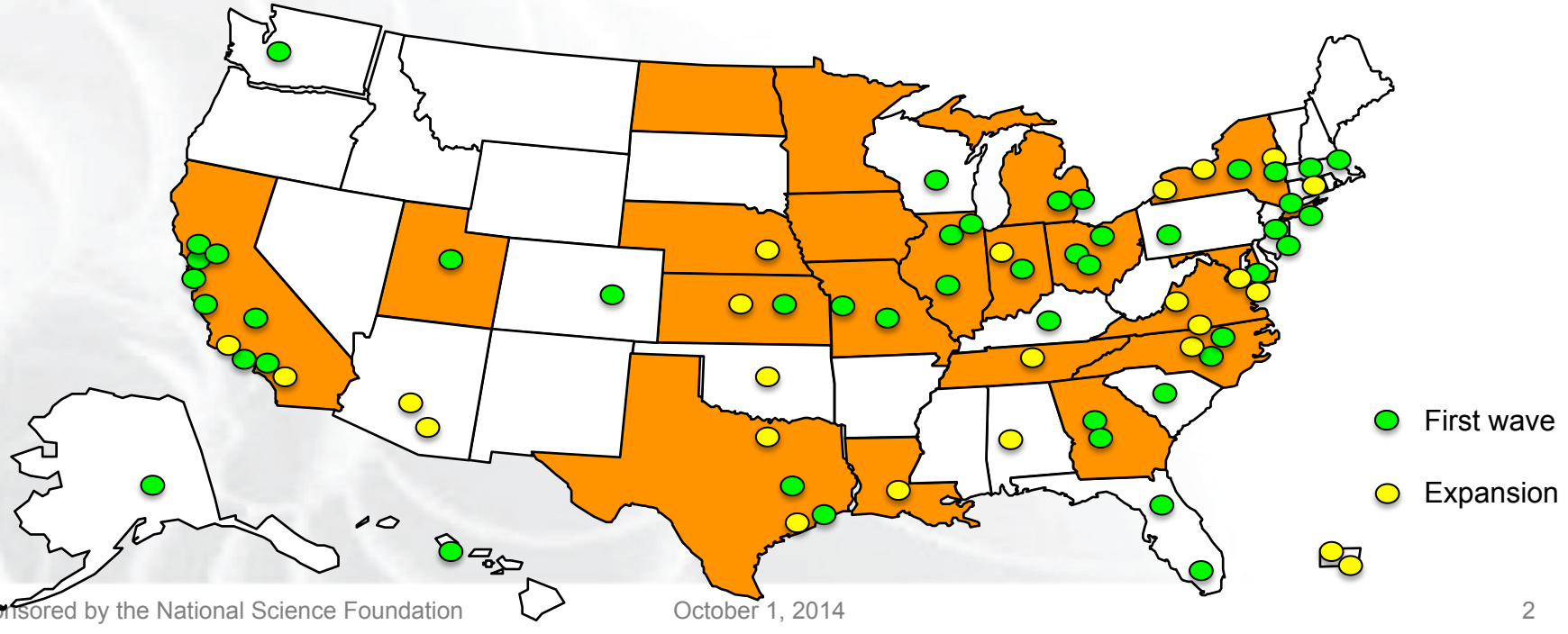
Software Defined Networks: Engineering GENI

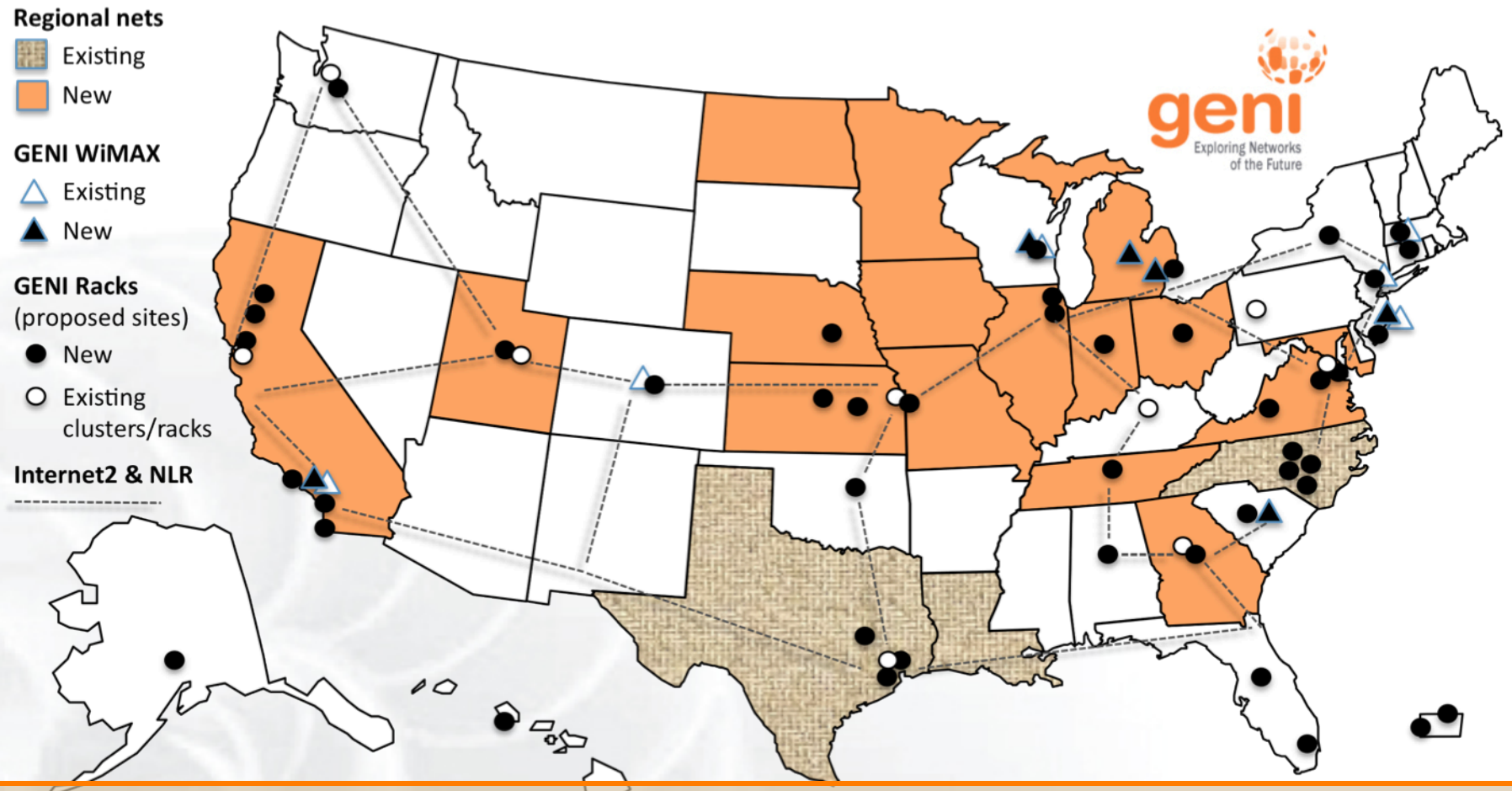
Heidi Picher Dempsey
and Tim Upthegrove
BBN Technologies
NANOG62 Meeting
October 6-8, 2014
www.geni.net

GENI provides a virtual lab for networking and distributed systems research and education

- GENI started with exploratory, rapid prototyping 5 years ago
- GENI design assumes federation of *autonomously owned and operated* systems
- Yearly prototyping cycle for an idea: develop, integrate and *operate*
- Experimenters use the testbed *while we are building it out*
- Even prototypes have “activist” users, and must evolve to satisfy those users or fade away. Two of five original design frameworks predominate now.
- “Horizontal” dataplane slicing as a service (or sometimes just engineered)
- “Vertical” control plane APIs to negotiate and allocate resources

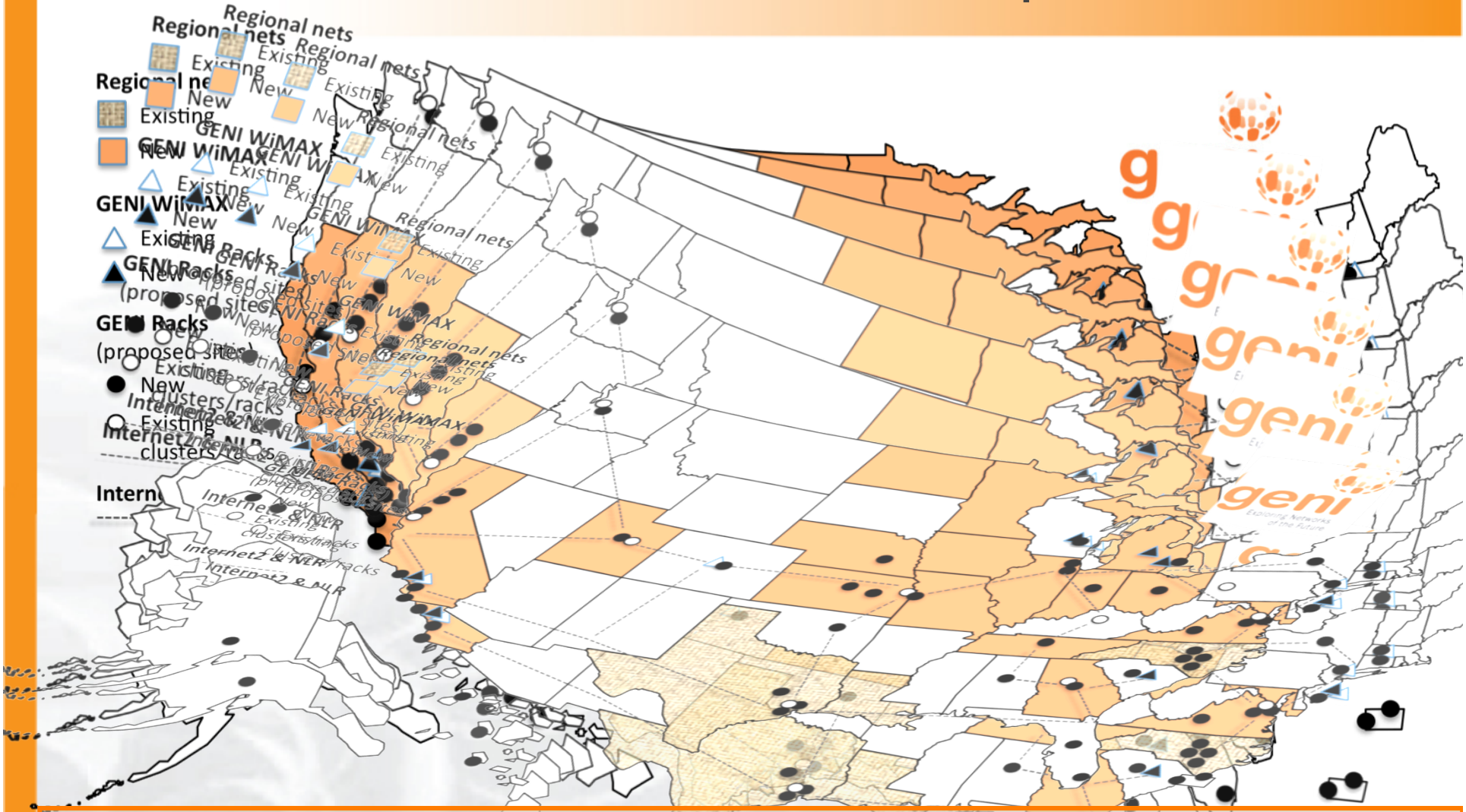


GENI: Infrastructure for Experimentation



GENI provides compute, network, and wireless resources that can be connected in experimenter-specified Layer 2 topologies.

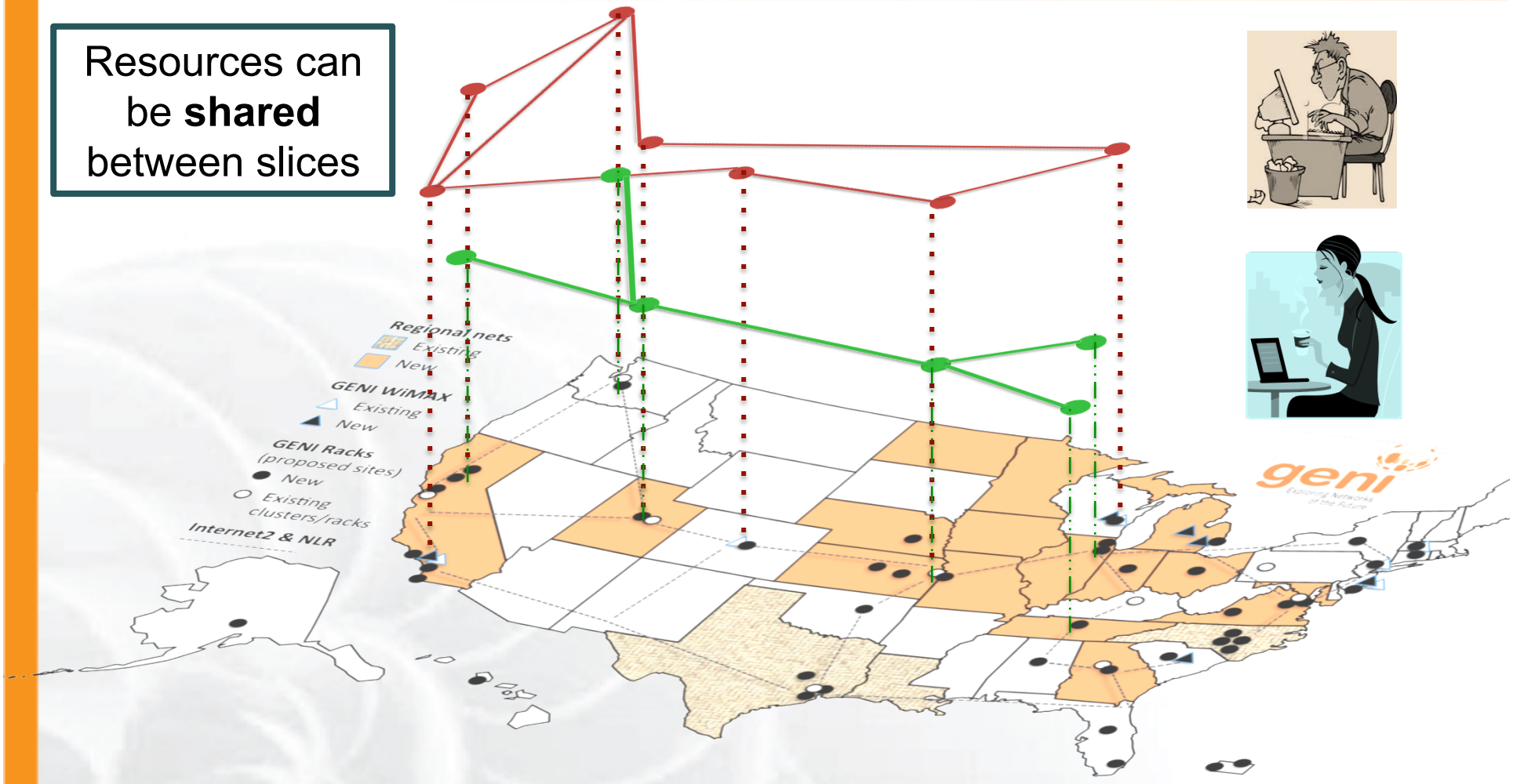
GENI: Infrastructure for Experimentation



GENI provides compute resources that can be connected in experimenter specified Layer 2 topologies.

Multiple GENI Experiments run Concurrently

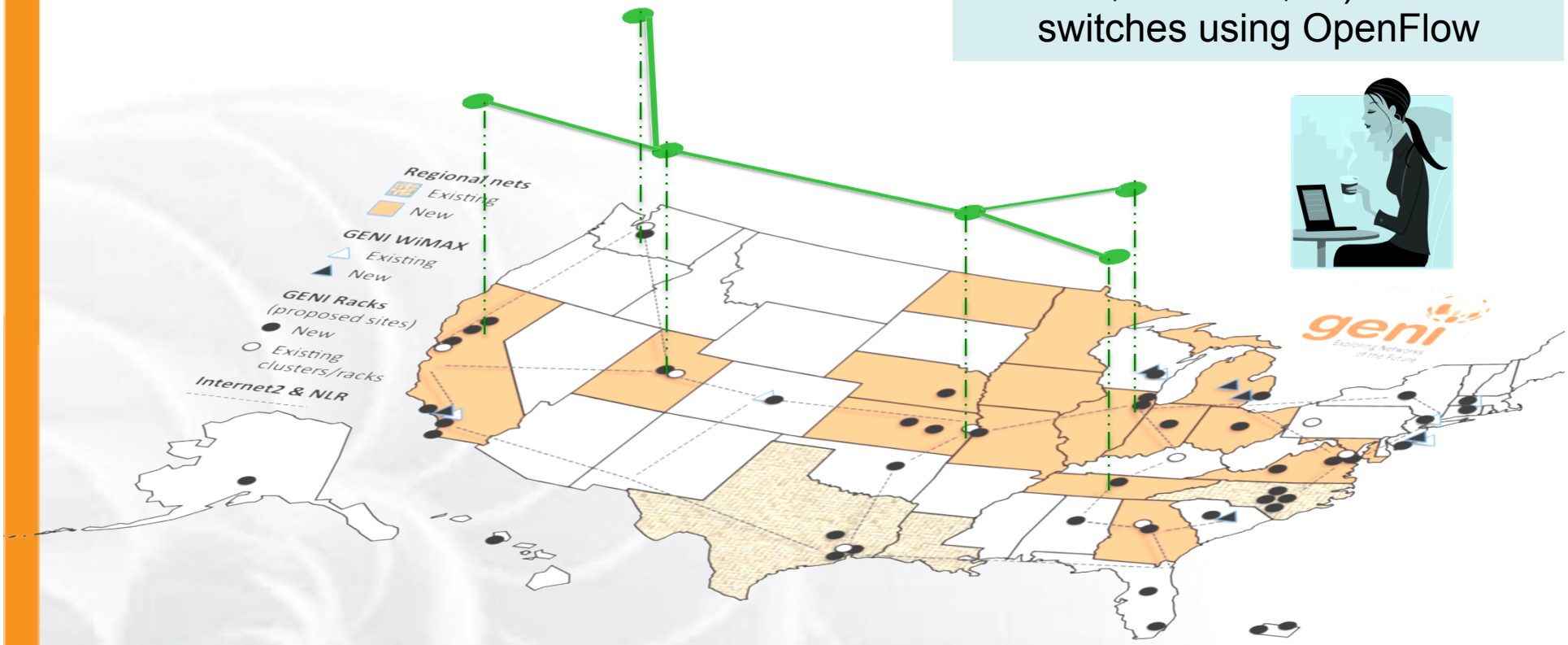
Resources can be **shared** between slices



Experiments live in somewhat isolated “slices”

GENI is “Deeply Programmable”

I install software I want throughout my network slice (into routers, switches, …) or control switches using OpenFlow

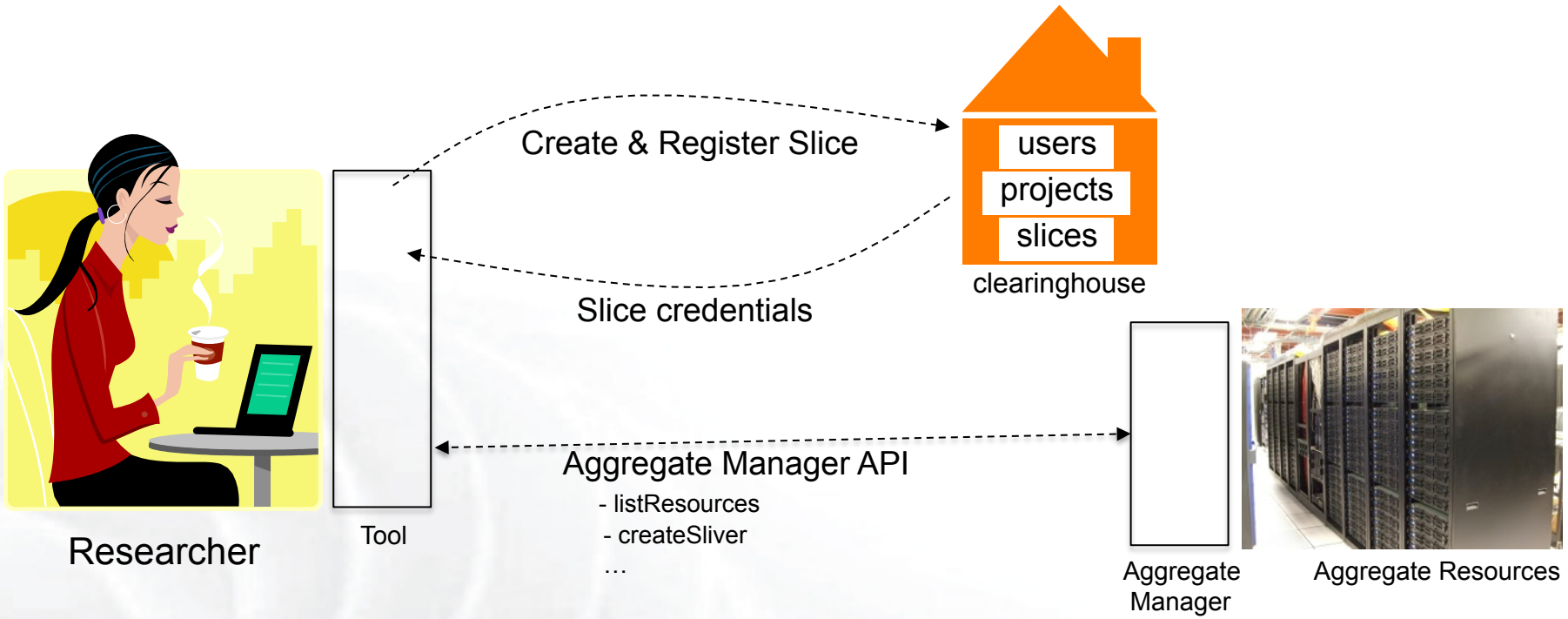


Experimenters can set up custom topologies, protocols and switching of flows

Access to GENI

- Over 2500 users (September, 2014)
- Experimental resources at 52 campuses, 11 regional networks, 10 WiMAX/LTE wireless sites
- GENI credentials and management based on Shibboleth single sign on and InCommon
- GENI experiments run continuously
- Operations support from six groups in different US locations

Software: Clearinghouse and Aggregates



- Clearinghouse: manages users, projects and slices
  - Standard credentials shared via custom API or new Common CH API
  - GENI supported accounts: GENI Portal/CH, PlanetLab CH, ProtoGENI CH
- Aggregate: provides resources to GENI experimenters
  - Typically owned and managed by an organization
  - Speaks the GENI Aggregate Manager API (AM API)
  - http://groups.geni.net/geni/wiki/GAPI_AM_API_V3 most recent version
  - http://trac.gpolab.bbn.com/gcf download reference implementation (gcf), OMNI command line client
  - Examples: PlanetLab, Emulab, GENI racks on various campuses

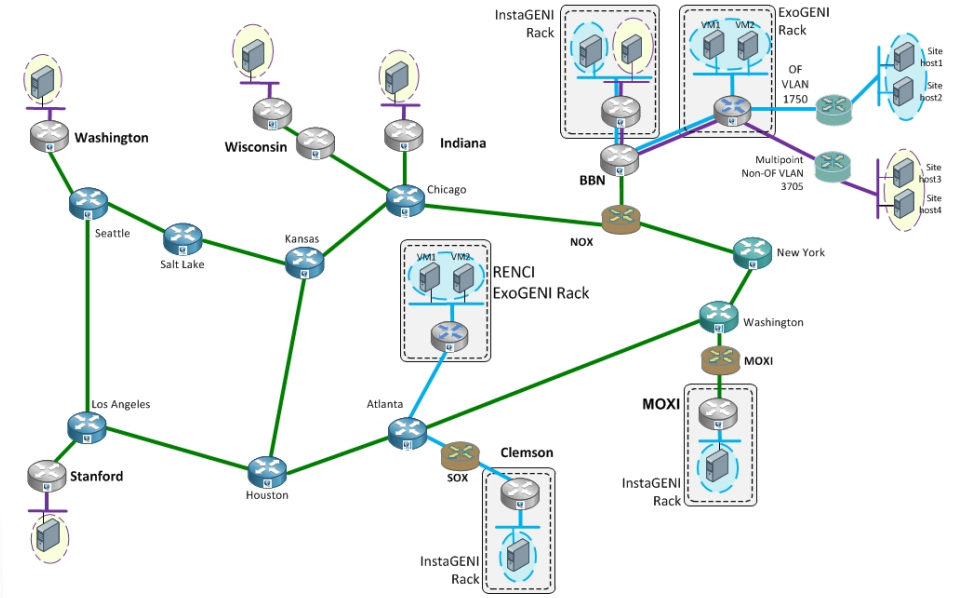
Engineering for Layer2 SDN

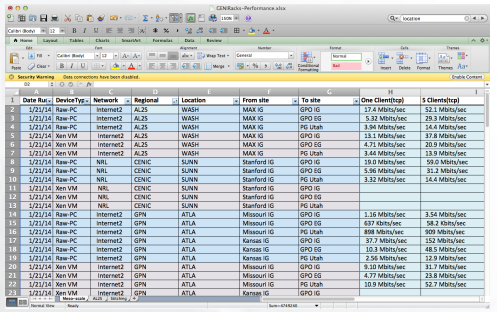
- Experimenters run their own SDN controllers
- Each network aggregate may run their own controller (many don't)
- SDN switches and endpoints are configured with VLAN ranges that can be used for SDN experiments. Supported configurations:
  - One VLAN per experiment with/without OpenFlow controller
  - One shared VLAN with multiple OpenFlow controllers (per-experiment addressing and controllers mediated by GPO and GENI software)
  - One multipoint VLAN with one service (e.g. wireless network experiments)
- Experimenters can choose software or hardware switches—this talk is about hardware switches
- GENI Aggregate Manager (AM) software negotiates and coordinates resource access
  - AM API includes VLAN "stitching" http://groups.geni.net/geni/wiki/GAPI_AM_API_V3
  - OpenFlow site/network access AM http://groups.geni.net/geni/wiki/OpenFlow/FOAM

Network Engineering Requirements for Shared Services

- L2 dataplane engineering
  - campuses, regional, core and international networks
  - many vendors and technologies
  - 1-100GbE interfaces (GENI shares with other R&E projects)
  - Shared or exclusive experimenter VLANs on interfaces, depending on experiment (mostly exclusive)
- SDN (OpenFlow 1.0) switches with experimenter's and sometimes R&E network's controllers (many vendors, varying implementation of standards)
- Standard Internet control plane
- Internet2 AL2S cross-connects and ION

  http://groups.geni.net/geni/wiki/GENIOESSTopologiesPerformance - IONtoAL2SPerformance





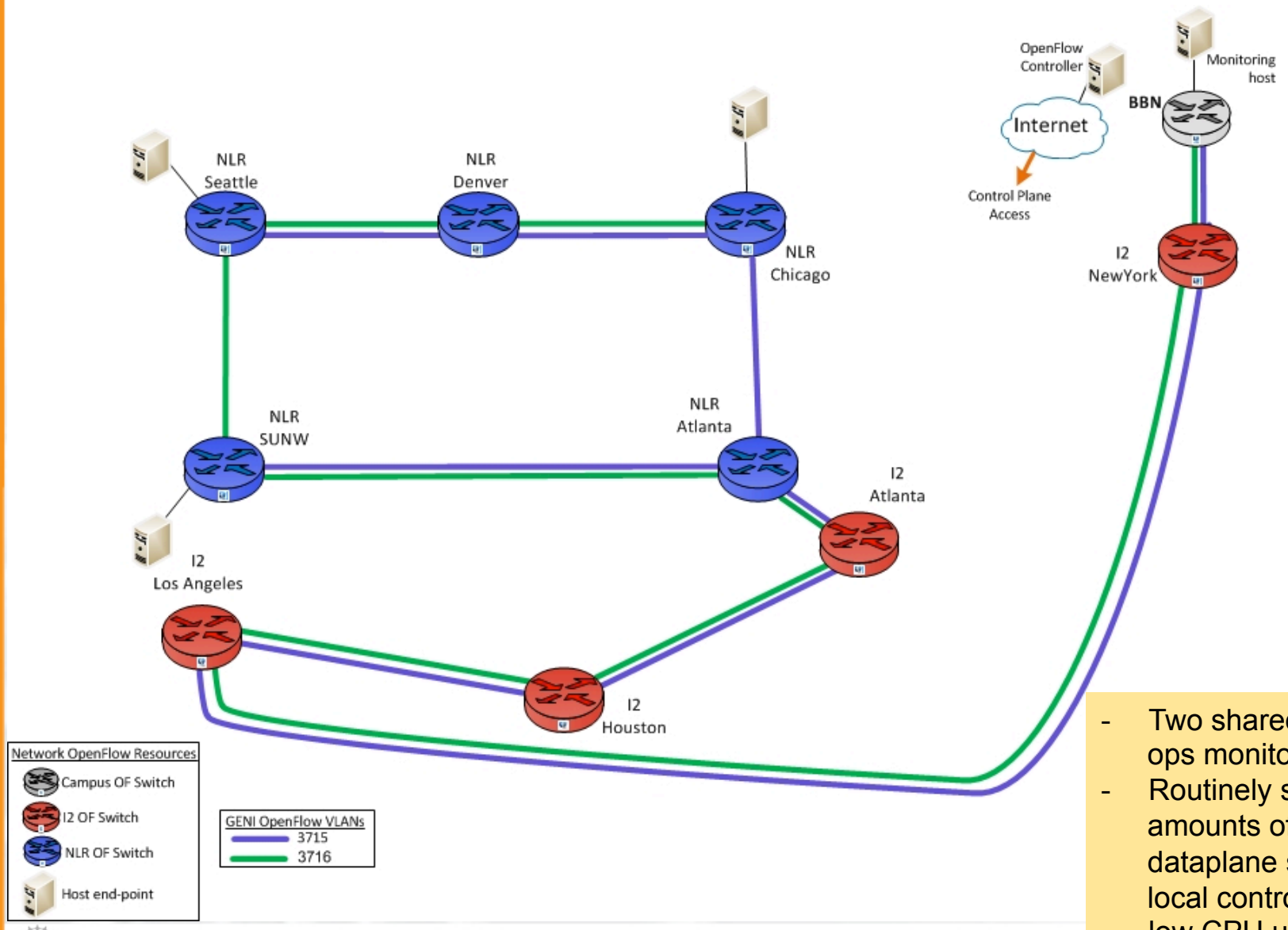
GENI Interoperable SDN

- Network aggregates operate various switches
  - Brocade
  - IBM
  - HP
  - Pica8
  - Cisco
  - NEC
  - Juniper
  - Dell
  - Open vSwitch (software-only switch)
- Experimenters and network engineers develop various controllers, based on open source projects
  - Floodlight
  - POX (replaced NOX)
  - OpenDaylight
- Operators develop additional open source tools to support resource sharing and monitoring (several—see www.geni.net)

Deployment engineering for SDN

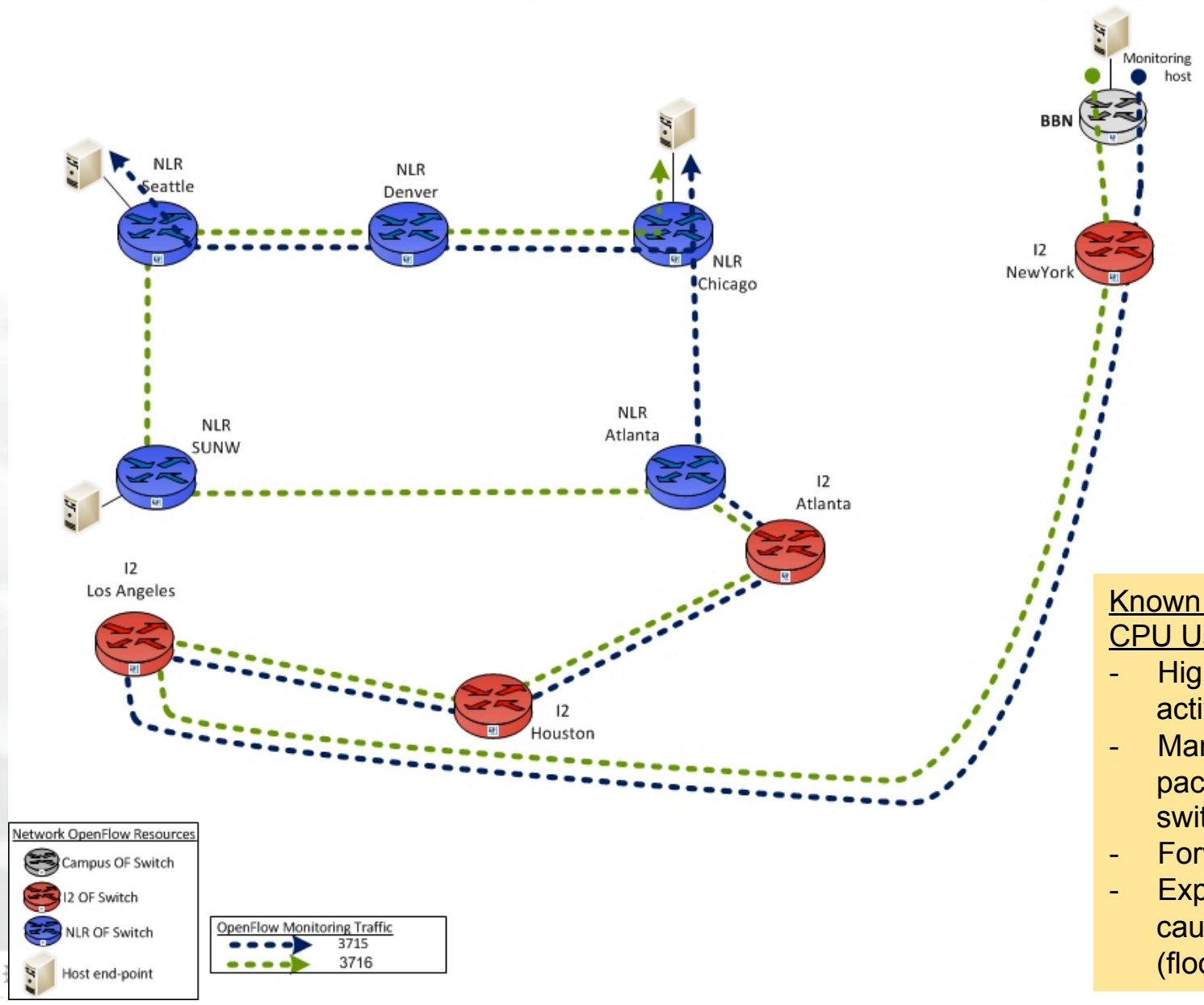
- Disable Spanning Tree Protocol (not just on SDN switches)
- Disable MAC learning
- Coordinate IP address ranges to avoid duplication, especially with shared VLANs
- Monitor for loops and load, use external limits if needed
- Compare firewall rules to SDN traffic profiles
- Separate control plane from SDN data plane
- Beware partial OpenFlow specification implementations

SDN Operations Example



- Two shared VLANs for ops monitoring
- Routinely send small amounts of OF data on dataplane slices using local controller
- low CPU use

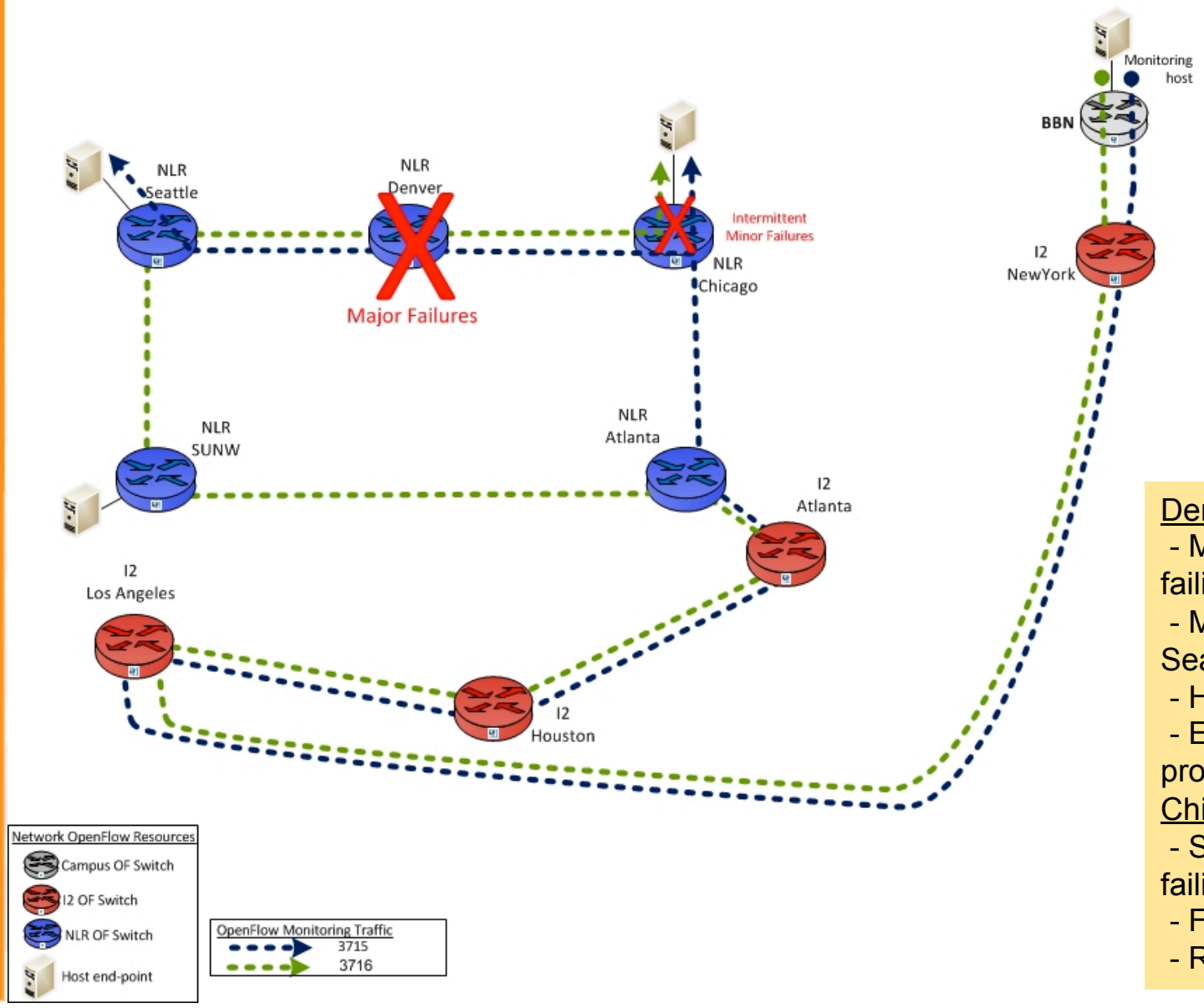
Operations Traffic and CPU Monitoring



Known Causes of High CPU Usage

- High control plane activity
- Many unmatched packets arriving on switch
- Forwarding loops etc.
- Experimenters may cause intentionally (flood, rewrite)

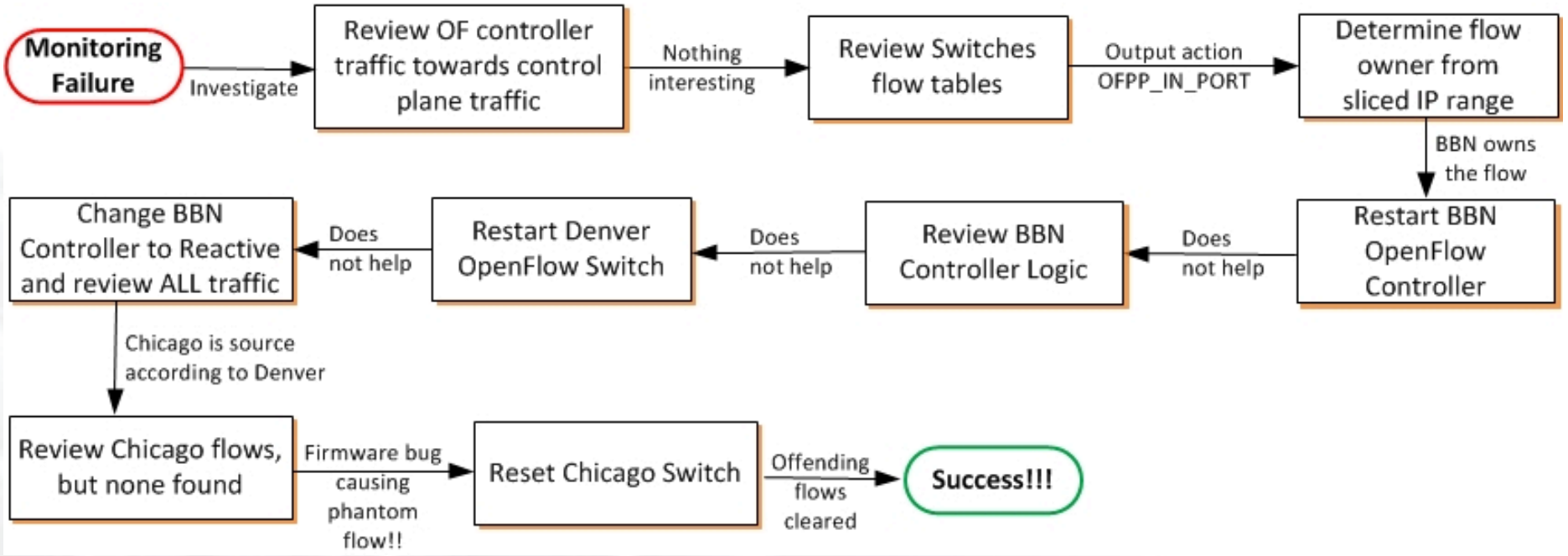
Reported Errors



Denver Switch Symptoms:
- Most monitoring traffic failing
- Many flows installed to Seattle
- High CPU use
- Experimenters report problems

Chicago Switch Symptoms:
- Some monitoring traffic failing
- Few flows
- Relatively low CPU use

Ops Debug Workflow



SDN Basic Tools

Floodlight Table Stats


```
"06:d6:00:26:f1:40:a8:00": [
    {
        "activeCount": 0,
        "length": 64,
        "lookupCount": 0,
        "matchedCount": 0,
        "maximumEntries": 1500,
        "name": "HW TCAM",
        "tableId": 0,
        "wildcards": 2629857
    },
    {
        "activeCount": 0,
        "length": 64,
        "lookupCount": 11284,
        "matchedCount": 0,
        "maximumEntries": 65536,
        "name": "hash",
        "tableId": 1,
        "wildcards": 0
    },
    {
        "activeCount": 0,
        "length": 64,
        "lookupCount": 0,
        "matchedCount": 0,
        "maximumEntries": 65536,
        "name": "classifier",
        "tableId": 2,
        "wildcards": 4194303
    }
],
```


Floodlight Flow Stats


```
"0e:84:00:26:f1:40:a8:00": [
    {
        "actions": [
            {
                "length": 8,
                "lengthU": 8,
                "maxLength": 0,
                "port": 50,
                "type": "OUTPUT"
            }
        ],
        "byteCount": 588,
        "cookie": 0,
        "durationNanoseconds": 755000000,
        "durationSeconds": 2844,
        "hardTimeout": 0,
        "idleTimeout": 5,
        "match": {
            "dataLayerDestination": "00:26:b9:7e:6c:c8",
            "dataLayerSource": "02:a0:02:7c:8f:2b",
            "dataLayerType": "0x0800",
            "dataLayerVirtualLan": -1,
            "dataLayerVirtualLanPriorityCodePoint": 0,
            "inputPort": 51,
            "networkDestination": "10.50.1.100",
            "networkDestinationMaskLen": 32,
            "networkProtocol": 1,
            "networkSource": "10.50.2.4",
            "networkSourceMaskLen": 32,
            "networkTypeOfService": 0,
            "transportDestination": 0,
            "transportSource": 8,
            "wildcards": 0
        },
        "packetCount": 2830,
        "priority": -1,
        "tableId": 0
    },
```


Wireshark OF dissector

OFP+ARP	144	Packet In (AM) (BufID=72627714) (78B) =>
OFP	90	Packet Out (CSM) (BufID=72627714) (24B)
TCP	66	51506 > 31750 [ACK] Seq=737 Ack=489 Win=
TCP	66	37157 > 31750 [ACK] Seq=311 Ack=73 Win=6
TCP	66	31750 > 38013 [ACK] Seq=73 Ack=448 Win=1
OFP+ICMP	182	Packet In (AM) (BufID=1151030723) (116B)
OFP+ICMP	182	Packet In (AM) (116B) => Echo (ping) req
OFP	146	Flow Mod (CSM) (80B)
OFP+ICMP	188	Packet Out (CSM) (122B) => Echo (ping) r
TCP	66	51395 > 31750 [ACK] Seq=659 Ack=489 Win=
OFP+LLDP	191	Packet In (AM) (BufID=127683122) (125B)
TCP	66	39449 > 31750 [ACK] Seq=195 Ack=73 Win=6
TCP	66	38616 > 31750 [ACK] Seq=389 Ack=233 Win=
OFP	90	Packet Out (CSM) (BufID=1151030723) (24B
OFP+LLDP	144	Packet In (AM) (BufID=43623008) (78B) =>
TCP	66	38616 > 31750 [ACK] Seq=389 Ack=257 Win=
TCP	66	31750 > 22539 [ACK] Seq=73 Ack=1328 Win=
TCP	66	31750 > 31549 [ACK] Seq=73 Ack=604 Win=1002 Len=0 TSval=781283131 TSecr=2993634626
TCP	66	35469 > 31750 [ACK] Seq=311 Ack=329 Win=501 Len=0 TSval=1735978286 TSecr=781283112
OFP+ICMP	182	Packet In (AM) (BufID=963453) (116B) => Echo (ping) request id=0xe118, seq=2106/14856, ttl=64
OFP+OFP+ICMP	298	Packet In (AM) (BufID=1151031476) (116B) => Packet In (AM) (BufID=1151032439) (116B) => Echo (ping) request id=0x230a
OFP+ICMP	182	Packet In (AM) (BufID=9122168) (116B) => Echo (ping) request id=0xe118, seq=2106/14856, ttl=64
OFP	90	Packet Out (CSM) (BufID=963453) (24B)
OFP	146	Flow Mod (CSM) (80B)
OFP	90	Packet Out (CSM) (BufID=9122168) (24B)
OFP+ICMP	182	Packet In (AM) (BufID=1151033359) (116B) => Echo (ping) request id=0x230a, seq=1/256, ttl=64
OFP	146	Flow Mod (CSM) (80B)
OFP+LLDP	191	Packet In (AM) (BufID=45458626) (125B) => Chassis Id = 00:07:43:13:ef:0f Port Id = 00:07:43:13:ef:0f TTL = 120
TCP	66	31750 > 38013 [ACK] Seq=73 Ack=573 Win=1002 Len=0 TSval=781283147 TSecr=1032155923
OFP+LLDP	191	Packet In (AM) (BufID=127666739) (125B) => Chassis Id = 00:07:43:14:82:7f Port Id = 00:07:43:14:82:7f TTL = 120
TCP	66	31750 > 22539 [ACK] Seq=73 Ack=1453 Win=6042 Len=0 TSval=781283149 TSecr=3631861413
TCP	66	48587 > 31750 [ACK] Seq=397 Ack=241 Win=501 Len=0 TSval=3166338685 TSecr=781283144
OFP	90	Packet Out (CSM) (BufID=1151033359) (24B)
TCP	66	38616 > 31750 [ACK] Seq=621 Ack=337 Win=17565 Len=0 TSval=3166338709 TSecr=781283140
OFP	194	Packet Out (CSM) (BufID=1151032439) (24B)

Real Life Flow Matches—Only One Vendor

Flow match on v2 modules

Table 5 Flow match on v2 modules

Flow type	VLAN ID	VLAN Pty	In_Port	Ethernet Type	Source MAC	Destination MAC	Source IP	Destination IP	IP ToS	IP Prot.	Source Port	Destination Port	V2 module flow location
VLAN ID [a]													
VLAN PCP [a]	b	b	b	c	c	c	c	c	c	c	c	c	hardware
In_Port [a]													
Ethertype IP [d]	b	b	b	IP	c	c	b	b	b	b	b	b	hardware
Ethertype IP [e]	b	b	b	IP	b	b	b	b	b	b	b	b	software
Ethertype Non-IP [f]	b	b	b	Non-IP	b	b	c	c	c	c	c	c	hardware
Ethertype Non-IP [g]	b	b	b	Non-IP	b	b	b	b	b	b	b	b	software
No Ethertype [h]	b	b	b	c	b	b	b	b	b	b	b	b	software

[a] A flow that matches the VLAN-ID, VLAN-PCP and IN_PORT with all other fields being blank will be in hardware.

[b] **Wildcard** — It does not matter if this field is specified or not in the flow.

[c] **Blank** — This field MUST NOT be present in the flow or is not applicable.

[d] If the Ethertype is IP, the MAC address fields must not be specified for the flow to be in hardware.

[e] If the Ethertype is IP and any MAC address fields is specified, the flow will be in software.

[f] If the Ethertype is non-IP, the flow can match against MAC address fields also in hardware provided the IP address fields are not specified.

[g] If the Ethertype is non-IP and any of the IP fields are specified, the flow will be in software.

[h] If the Ethertype fields is blank and any of the MAC address fields or IP address fields are specified, the flow will be in software.

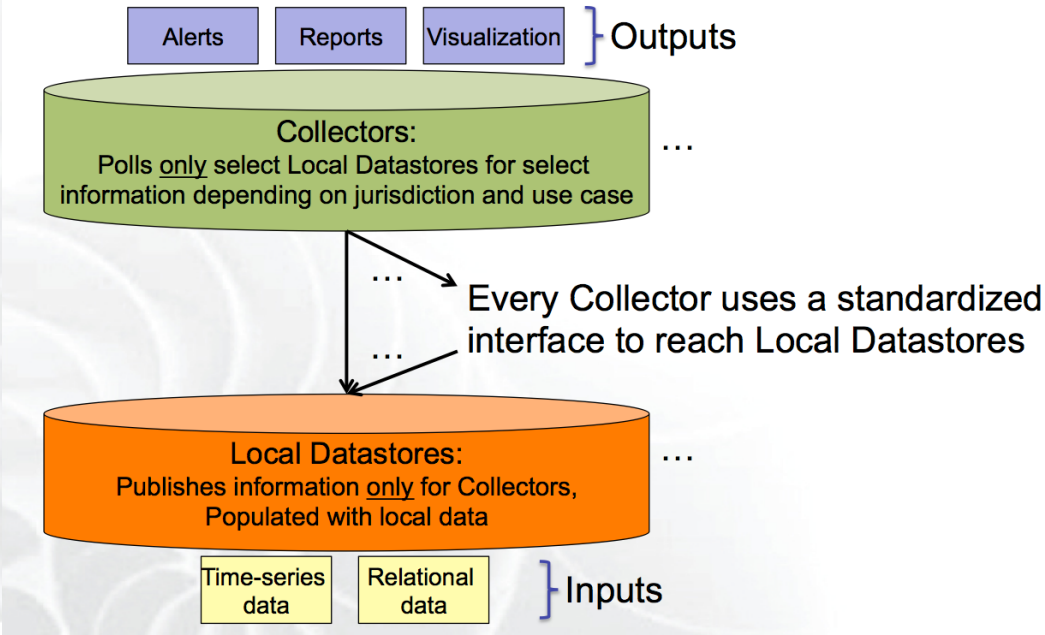
GENI SDN Evolution

- Switch support for hybrid networking (non-OF and OF on same switch)
- Separating network slicing from SDN control
- OpenFlow 1.0 to 1.3 migration
- OpenFlow policies on a distributed network
- Keeping networks interoperable
- SDN for broadband and home networks
- SDN Exchange points
- Cross-domain SDN monitoring

SDN Operations Requirements

- Site confirmation tests with logs and RSPECs
  http://groups.geni.net/geni/wiki/GENIRacksHome/InstageniRacks/ConfirmationTestStatus

  http://groups.geni.net/geni/wiki/GENIRacksHome/ExogeniRacks/ConfirmationTestStatus
- Emergency Stop and Legal, Law Enforcement and Regulatory Event Coordination (GMOC at Indiana University)
- Shared monitoring infrastructure and shared operations (6 major ops groups)



SDN Ops Deployment Requirements (cont)

- Standard installation processes
  http://groups.geni.net/geni/wiki/GENIRacksHome/RacksChecklistStatus
- System Acceptance Testing
  - Production: InstaGENI, ExoGENI
  - Provisional: OpenGENI (Dell), Cisco
- Shared site resource and access details
  http://groups.geni.net/geni/wiki/GeniAggregate