Robert Ricci and Gary Wong
GEC 8
July 20, 2010

ProtoGENI Tutorial

THE UNIVERSITY OF UTAH

Wireless Network

- **SSID**: hhonors
- Open your web browser, to get auth page
- **IMPORTANT:**
  - Check the ‘Advanced User’ box on this page
- Promotional code: `GENI0710`

This Tutorial

- Overview, signing up for accounts
- Creating a slice using command-line tools
- Using Kentucky Instrumentation Tools
- If time: Emulab Frontend

ProtoGENI Basics

The Basics

THE UNIVERSITY OF UTAH

You describe a network

ProtoGENI builds it for you

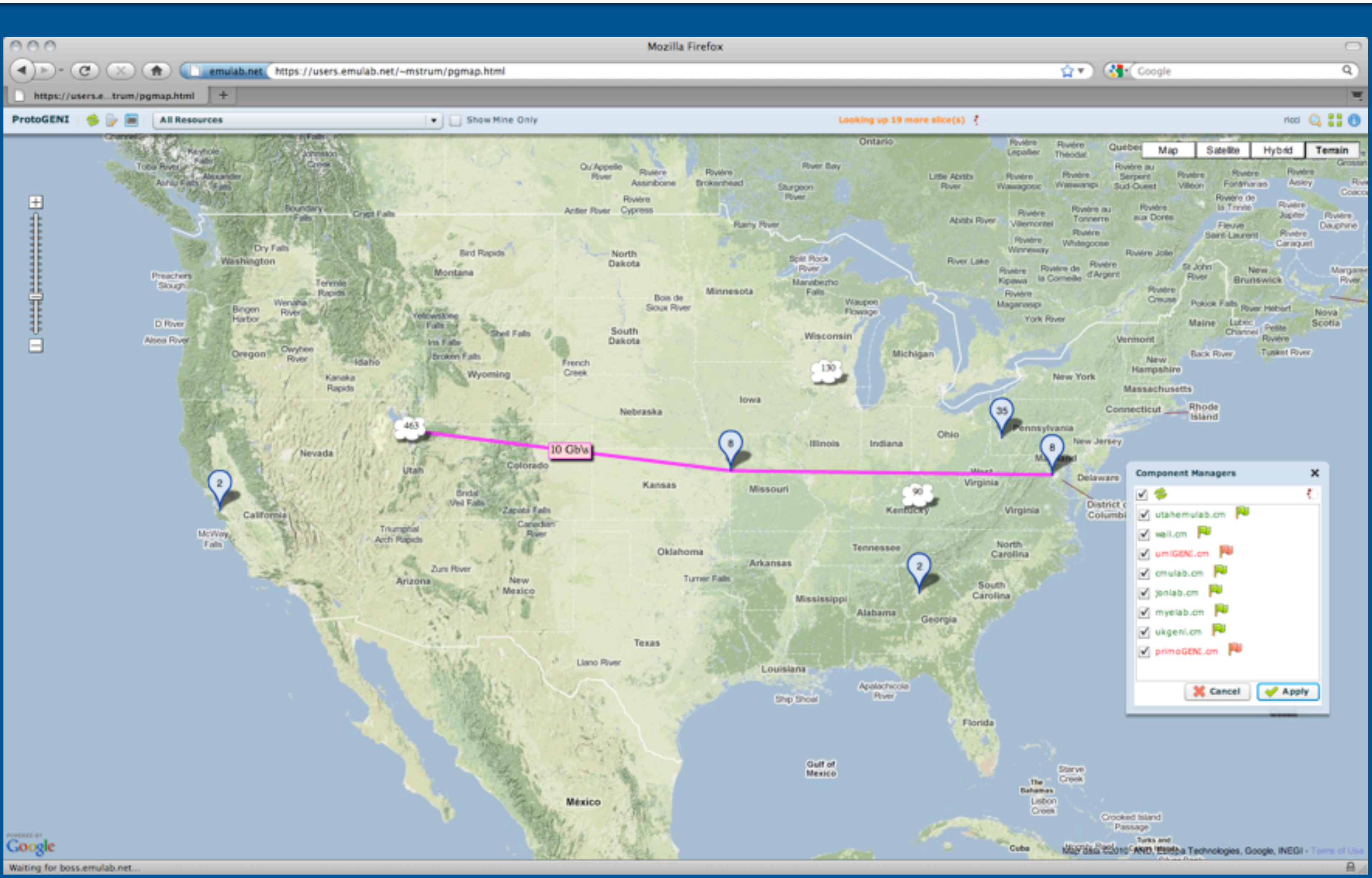
Your Network

- A “Slice”
  - Isolated from everyone else’s slice
- Described in terms of nodes (hosts, routers, etc.) and links
  - Description is done with an RSpec

What We Build It From

- PCs, switches, and tunnels (“components”)
  - At several sites around the country
  - Nationwide backbone with Internet2
- Guaranteed resources
  - Raw PCs
  - VLANs
- “Best effort” resources
  - Shared PCs (OpenVZ)
  - Tunnels across the Internet
- Other hardware as well (NetFPGAs, wireless devices, OpenFlow switches)

Current ProtoGENI Federation



Mozilla Firefox

https://users.emulab.net/~mstrum/pgmap.html

ProtoGENI | All Resources | □ Show Mine Only | Looking up 19 more slice(s)

Component Managers

☑

☑ utahemulab.cm

☑ wail.cm

☑ umlGENI.cm

☑ cmulab.cm

☑ jonlab.cm

☑ myelab.cm

☑ ukgeni.cm

☑ primoGENI.cm

Cancel | Apply

Waiting for boss.emulab.net...

What You Can Do With Your Network

- Log in to the PCs (ssh)
- Run your own code
  - Exclusive PCs – anything at all, including replacing the OS, any protocol you want
  - On shared PCs – ‘mostly root’, any protocol you want as long as it’s IP
- Send traffic
  - Within the links you defined (“experimental net”)
  - To the Internet (“control net”)

So How Do I Make a Slice?

- Raw APIs
- **Command-line tools (“test scripts”)**
- **Emulab Frontend**
- Coming soon: Flash GUI

Hands-On

Getting Ready for Hands-On

- Get in teams of 2 or 3
  - 20 teams max
- One laptop per team
  - WiFi
  - Emulab.net account (will cover this next)
  - SSH client (on Windows, putty works well)
    - www.putty.org
  - Browser with Java (Firefox, Chrome tested)
    - Only if we get time

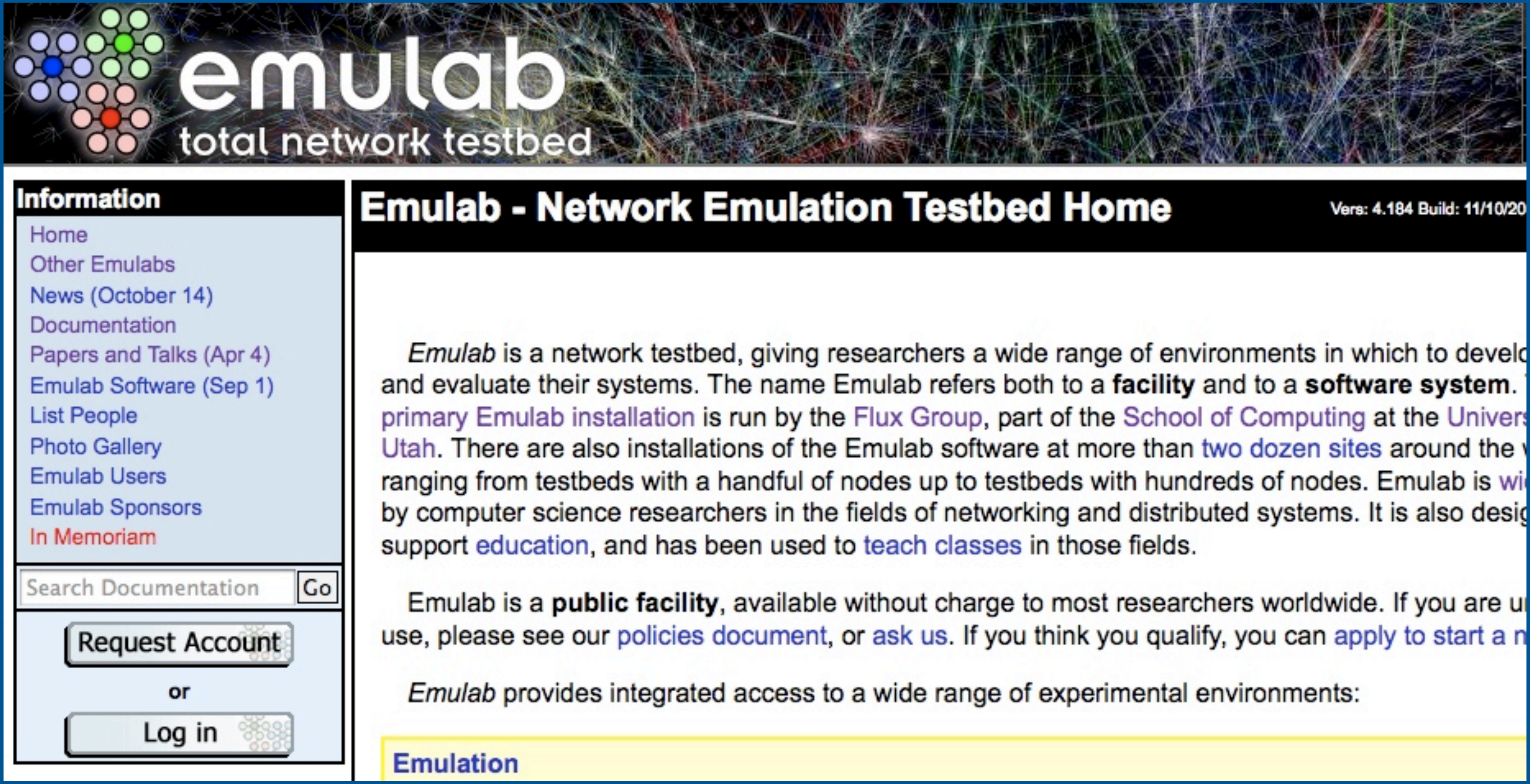
Getting An Account

- Apply at www.emulab.net
  - Operators are standing by
- Why emulab.net?
  - ProtoGENI is built on Emulab software
  - emulab.net is the site at Utah
  - Having an account there will get you into to any federated ProtoGENI site

Applying For An Account

- Go to ‘www.emulab.net’
- Click on ‘Request Account’
- Click on ‘Join Project’
- Fill out form
- Project name **gec8tutorial**
- Watch your email for confirmation step

Applying For An Account



Information

- Home
- Other Emulabs
- News (October 14)
- Documentation
- Papers and Talks (Apr 4)
- Emulab Software (Sep 1)
- List People
- Photo Gallery
- Emulab Users
- Emulab Sponsors
- In Memoriam

Emulab - Network Emulation Testbed Home

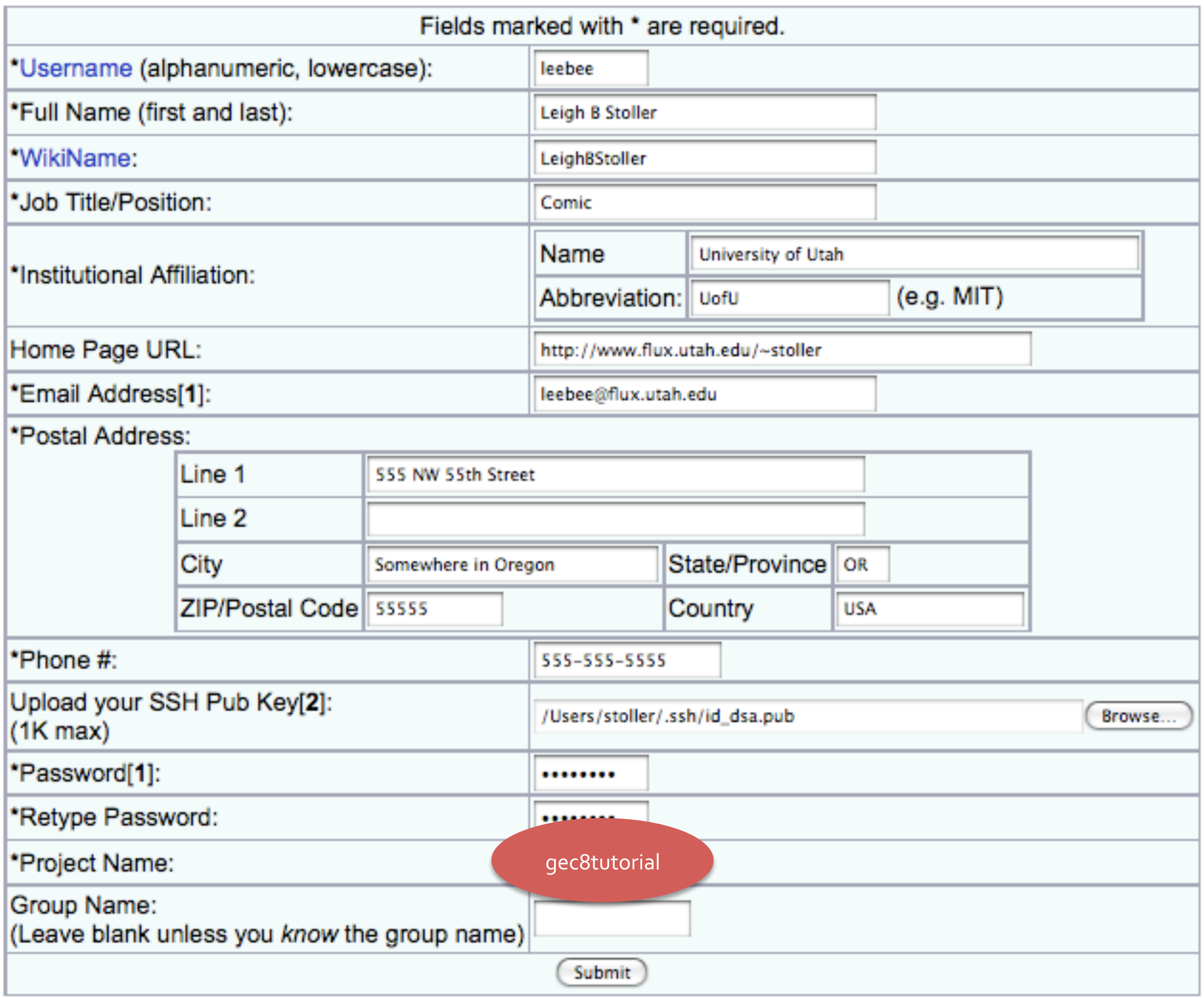
Vers: 4.184 Build: 11/10/20

Emulab is a network testbed, giving researchers a wide range of environments in which to develo and evaluate their systems. The name Emulab refers both to a **facility** and to a **software system**. primary Emulab installation is run by the Flux Group, part of the School of Computing at the Univers Utah. There are also installations of the Emulab software at more than two dozen sites around the ranging from testbeds with a handful of nodes up to testbeds with hundreds of nodes. Emulab is wi by computer science researchers in the fields of networking and distributed systems. It is also desig support education, and has been used to teach classes in those fields.

Emulab is a **public facility**, available without charge to most researchers worldwide. If you are u use, please see our policies document, or ask us. If you think you qualify, you can apply to start a n

Emulab provides integrated access to a wide range of experimental environments:

Emulation



Fields marked with * are required.

*Username (alphanumeric, lowercase):	leebee		
*Full Name (first and last):	Leigh B Stoller		
*WikiName:	LeighBStoller		
*Job Title/Position:	Comic		
*Institutional Affiliation:	Name	University of Utah	
	Abbreviation:	UofU	(e.g. MIT)
Home Page URL:	http://www.flux.utah.edu/~stoller		
*Email Address[1]:	leebee@flux.utah.edu		
*Postal Address:			
Line 1	555 NW 55th Street		
Line 2			
City	Somewhere in Oregon	State/Province	OR
ZIP/Postal Code	55555	Country	USA
*Phone #:	555-555-5555		
Upload your SSH Pub Key[2]: (1K max)	/Users/stoller/.ssh/id_dsa.pub	Browse...	
*Password[1]:	•••••••••		
*Retype Password:	•••••••••		
*Project Name:	gec8tutorial		
Group Name: (Leave blank unless you *know* the group name)			

Submit

After Today: Start a New Project

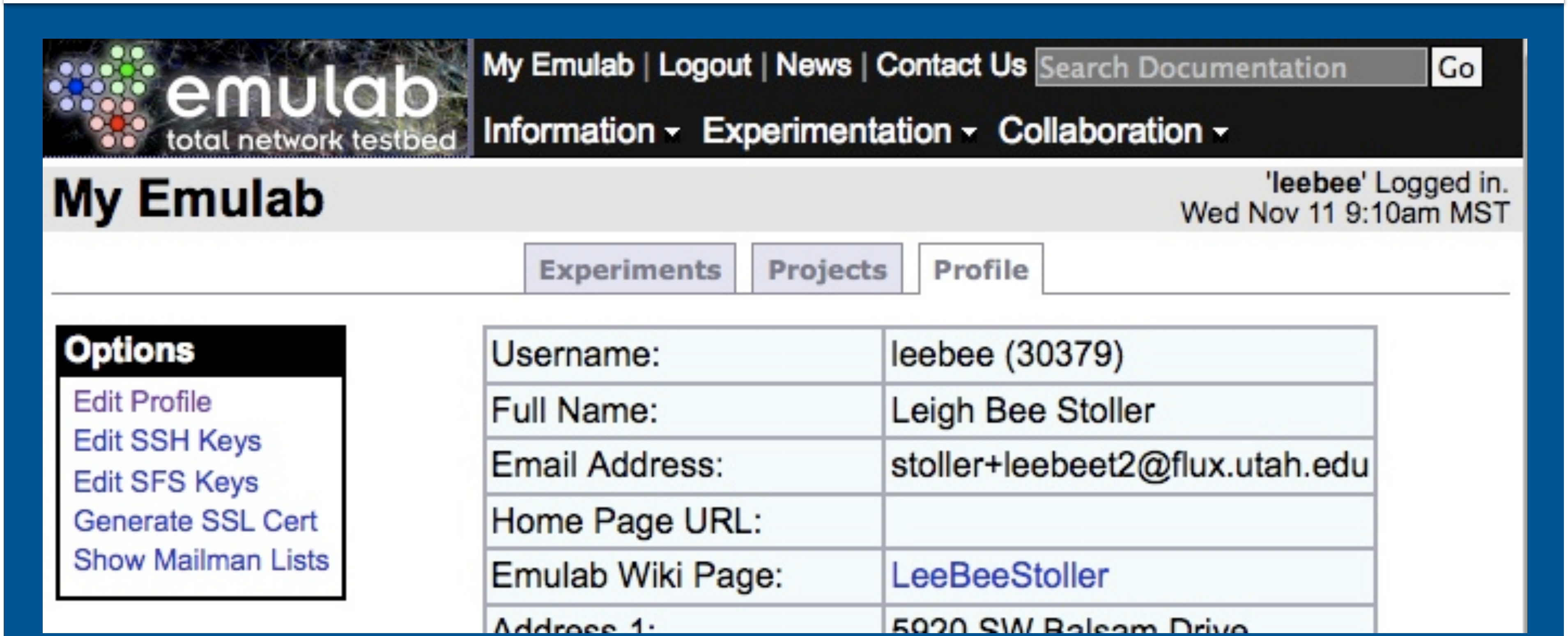
THE UNIVERSITY OF UTAH

- Projects are what they sound like
  - Collection of people collaborating on something
- Click on “Start a New Project”
  - Most be done by faculty member
- No sharing accounts
  - Project leader can approve users into the project
- Can use the account you’ve already created
  - Just log in first

What Your Account Gives You

- Accepted by every member of the ProtoGENI federation
- You are identified by a URN
  - eg. urn:publicid:IDN+emulab.net+user+ricci
- Certificate for authentication
  - Standard x.509
  - Can be used with OpenSSL or in your browser

Get an SSL Certificate



My Emulab

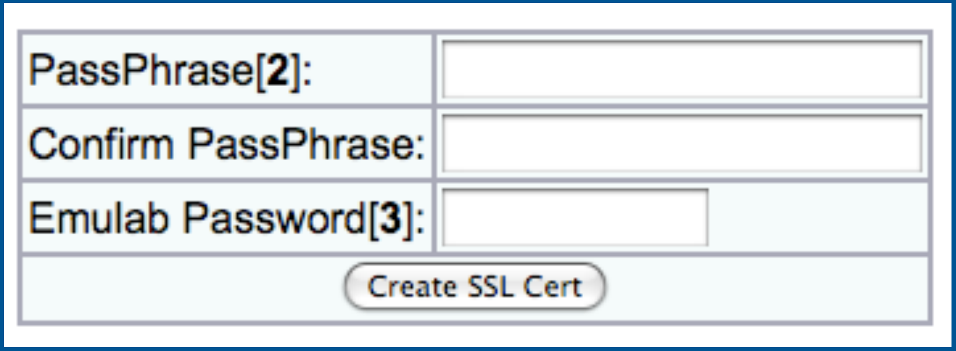
'leebee' Logged in.
Wed Nov 11 9:10am MST

Username:	leebee (30379)
Full Name:	Leigh Bee Stoller
Email Address:	stoller+leebeet2@flux.utah.edu
Home Page URL:	
Emulab Wiki Page:	LeeBeeStoller
Address 1:	5920 SW Balsam Drive

- Click on “My Emulab” if you do not see this page

Create an SSL Cert

- Click on “Generate SSL Certificate”



PassPhrase[**2**]:	
Confirm PassPhrase:	
Emulab Password[**3**]:	

- Click on Generate SSL Certificate
- Pick a good PassPhrase!
    - But note: will be stored in plaintext

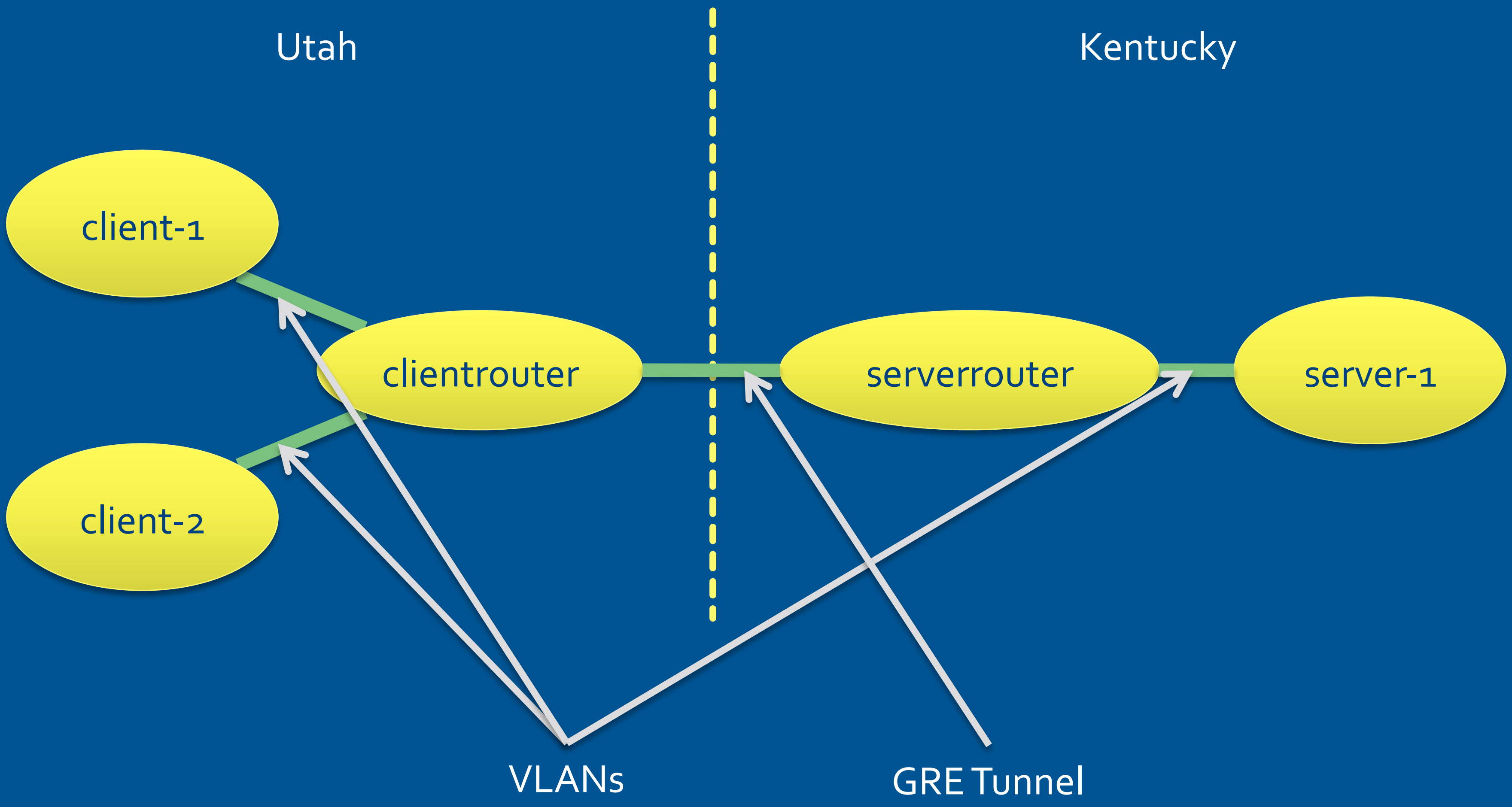
Hands-On: Command-line Slice Creation

Hands-On: Instrumentation Tools

http://www.netlab.uky.edu/p/instools/

Hands-On: Emulab Frontend

https://www.protogeni.net/trac/protogeni/wiki/GEC8Tutorial


Utah
Kentucky
client-1
client-2
clientrouter
serverrouter
server-1
VLANs
GRE Tunnel

Where To Get More Information

http://www.protogeni.net
http://www.emulab.net

http://www.protogeni.net/trac/protogeni/wiki/Tutorial
http://www.protogeni.net/trac/protogeni/wiki/API
http://www.protogeni.net/trac/protogeni/wiki/RSpec

geni-users@emulab.net

Extra Slides

Resources

What are Components?

- Nodes
  - PCs
  - NetFPGAs
  - Routers
  - Wireless Nodes
- Links -- Connectivity between nodes
  - VLANs
  - Tunnels

Nodes

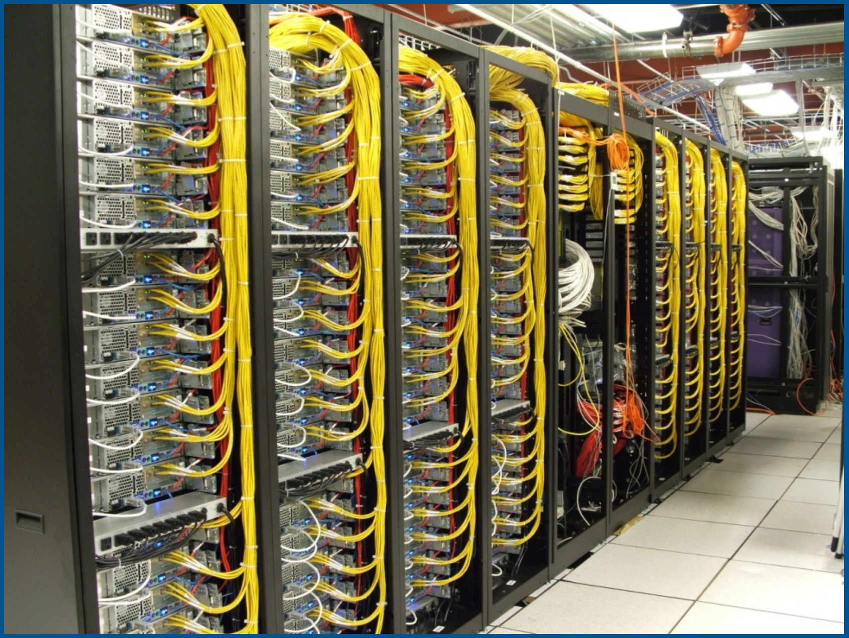
What kinds of nodes can I get?

Node Sharing

- Shared Nodes
  - Multiple Experimenters
  - Only OpenVZ Containers
  - Experimenters all have root in separate containers
- Exclusive Nodes
  - One experiment at a time
  - May have multiple containers at once

Raw PCs

- Physical PC
- Exclusive Access
- No Multiplexing
- Running FreeBSD or Linux
- You have the power
  - Root
  - Modify the kernel
  - Entirely new OS



OpenVZ Containers

- Container on a Physical PC
- Exclusive or shared access
- Multiple containers per PC
- OpenVZ Linux Image
- Root Authority
  - Within Container

Backbone Nodes

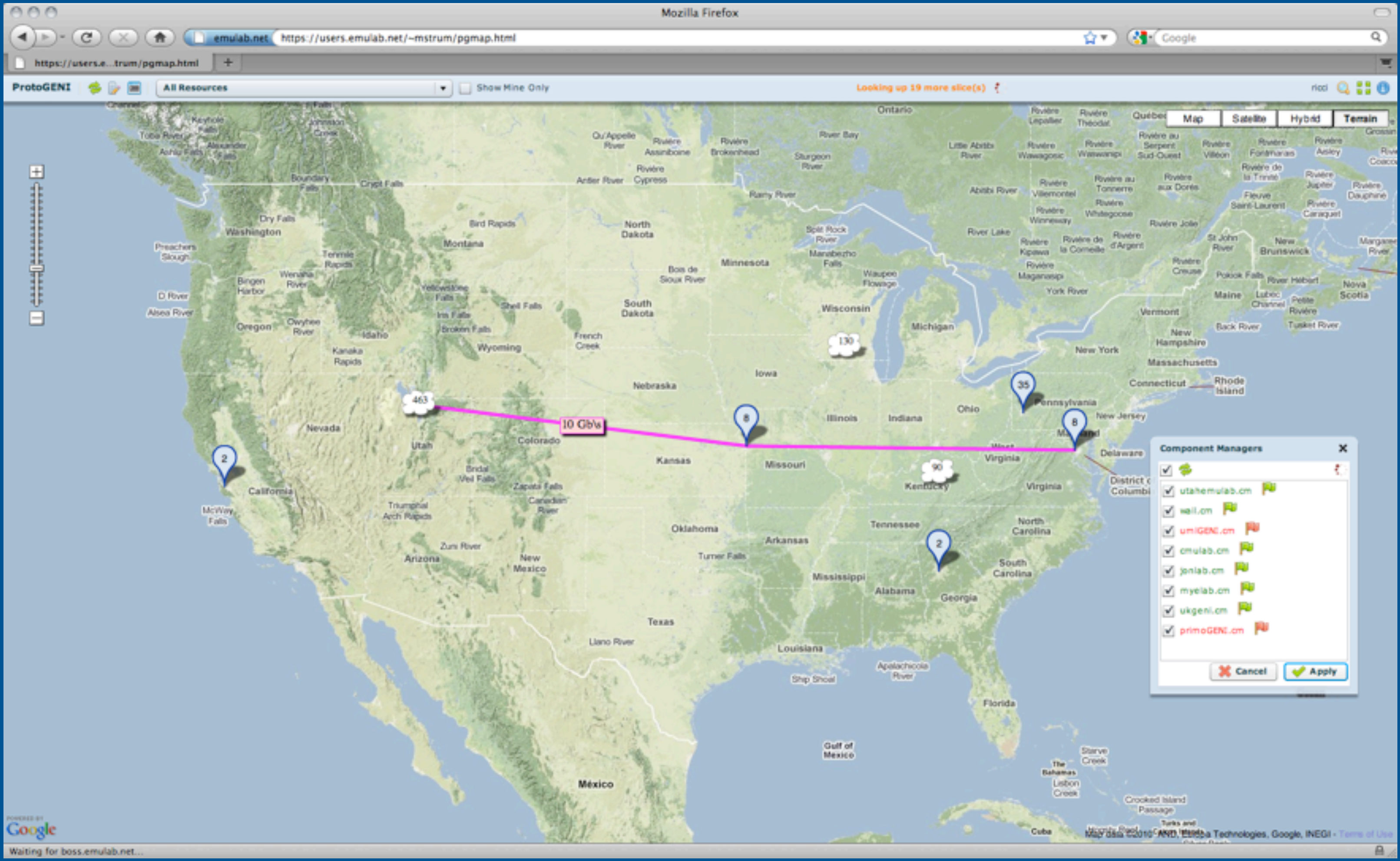
- Located in Internet 2 PoPs
- 2 PCs in each PoP
- Contain NetFPGAs
- Connectivity
  - 10 Gbps Dedicated (now)
  - 1 Gbps MPLS Tunnels (soon)
  - Ethernet Layer

CMs other thanUtah

THE UNIVERSITY OF UTAH

- Emulabs
  - Kentucky (PCs)
  - Wisconsin (PCs)
  - Umass- Lowell (PEN)
  - CMU (PCs, wireless)
  - FIU (PCs, simulator)
  - “Inner Elabs” at Utah
- Reference CM
  - ETRI (programmable router – in progress)
  - Sparta (ABAC testing)

Current ProtoGENI Federation


Mozilla Firefox
emulab.net https://users.emulab.net/~mstrum/pgmap.html
https://users.e...trum/pgmap.html
ProtoGENI
All Resources
Show Mine Only
Looking up 19 more slice(s)
ricci
Map
Satellite
Hybrid
Terrain
463
10 Gb/s
130
8
35
8
90
2
2
Waiting for boss.emulab.net...

Component Managers

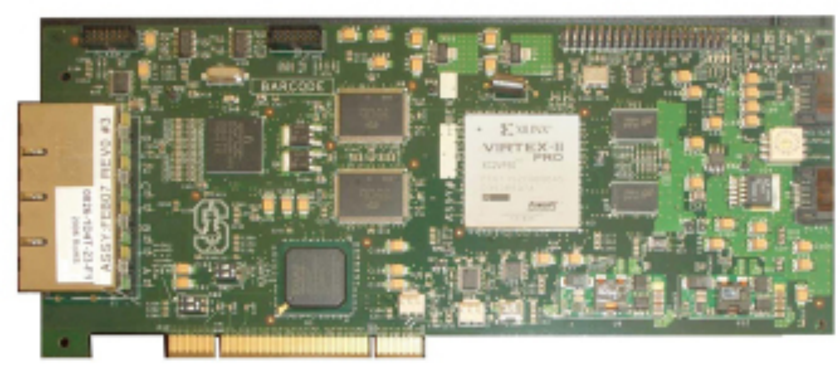
☑ utahemulab.cm
☑ wail.cm
☑ umIGENI.cm
☑ cmulab.cm
☑ jonlab.cm
☑ myelab.cm
☑ ukgeni.cm
☑ primoGENI.cm

Others

- NetFPGAs
- Wireless nodes
- Widearea nodes
- Non-Emulab sites

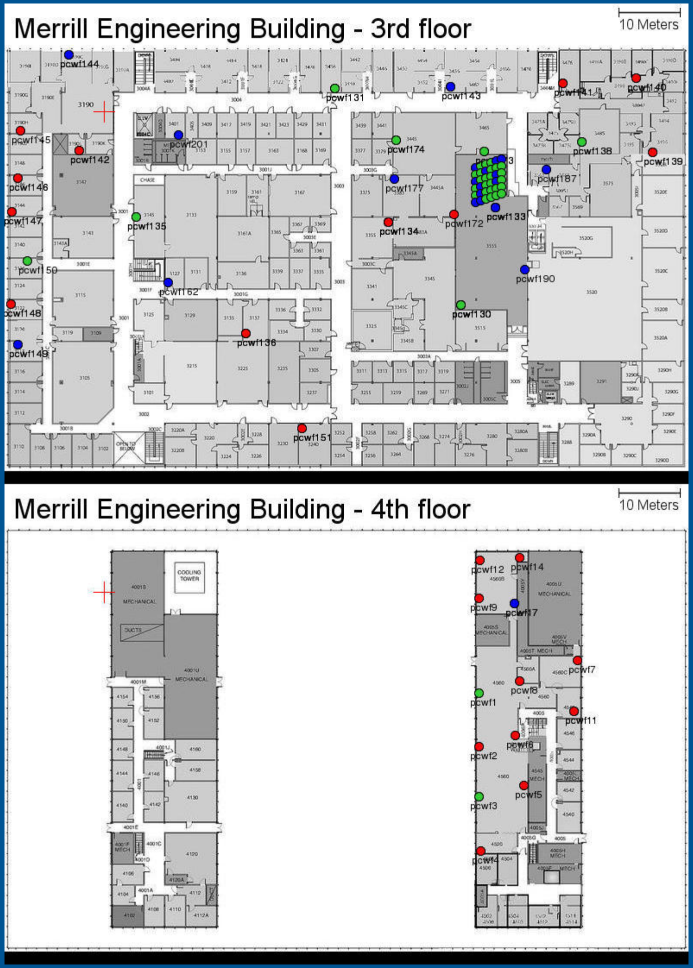
NetFPGA (Stanford)

- Network interfaces
- With FPGA for hardware design
- Attached to a PC
- Exclusive Access
- Allocated along with that PC



Wireless Nodes

- Where
  - Building-scale Wireless Testbed (Utah)
  - Wireless Emulator (CMU)
- Select and login like a normal node
- Wireless Interfaces
- No isolation
- Map of wireless locations available



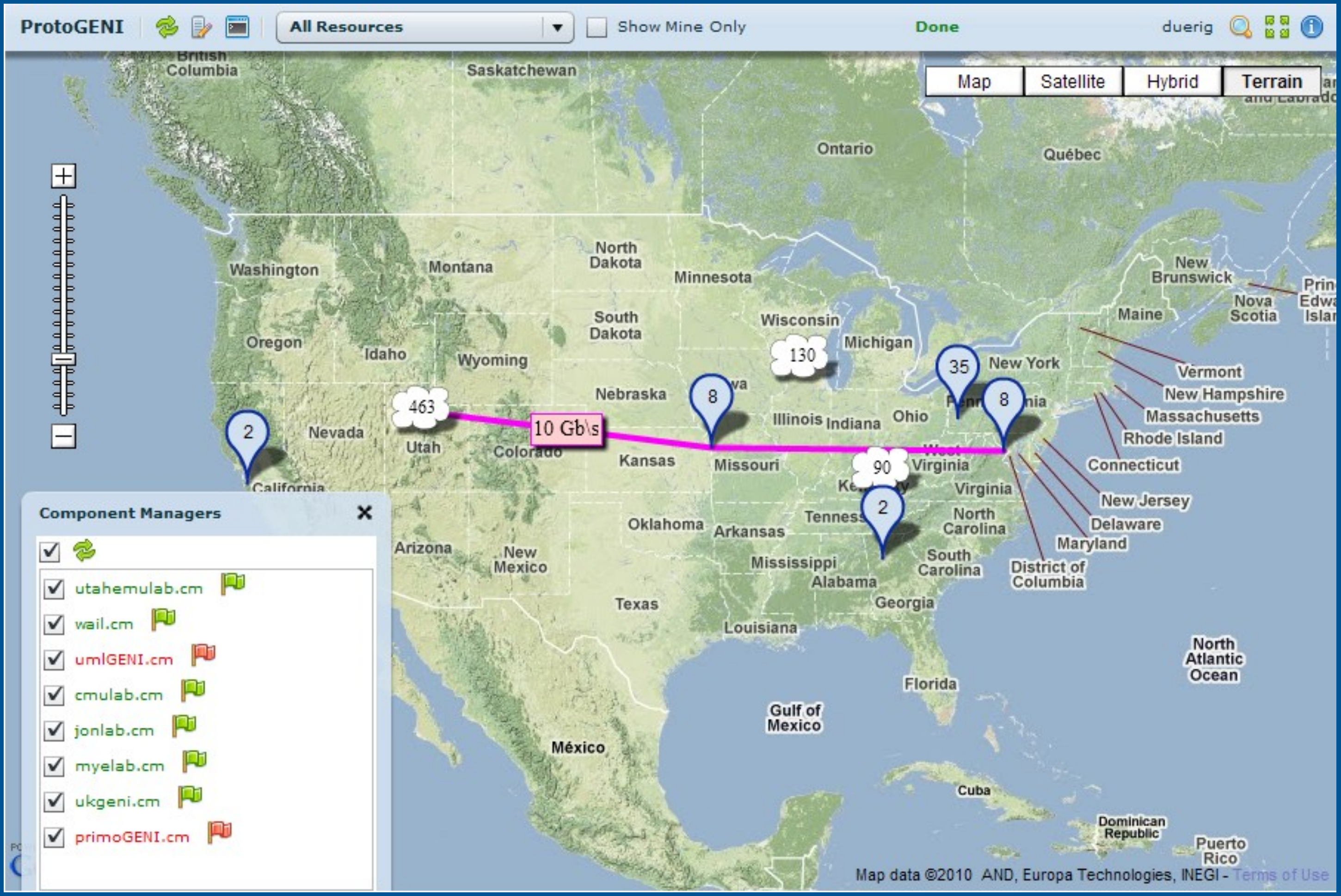
Merrill Engineering Building - 3rd floor

Merrill Engineering Building - 4th floor

Wide Area Nodes

- Managed by Utah CM
- Connected to different campuses
- Use tunnels to connect to other nodes

Map Interface



Component Managers

☑

☑ utahemulab.cm

☑ wail.cm

☑ umlGENI.cm

☑ cmulab.cm

☑ jonlab.cm

☑ myelab.cm

☑ ukgeni.cm

☑ primoGENI.cm

Links

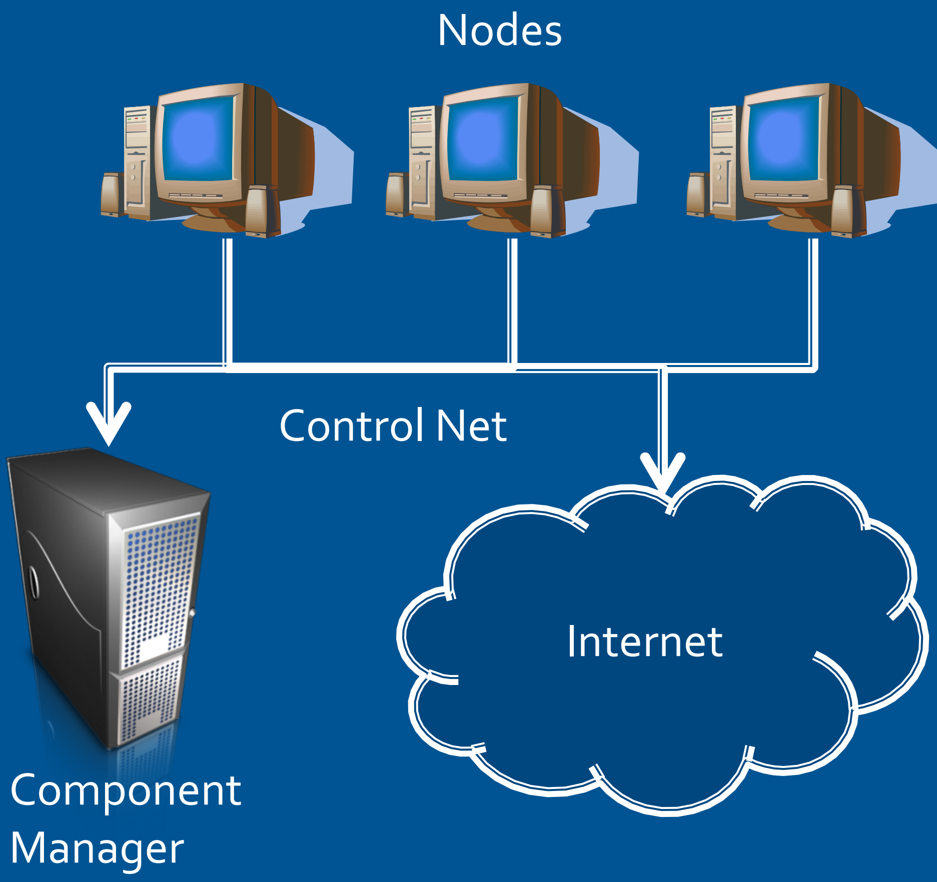
Connecting nodes together

Three Planes of Communication

- Control Plane
  - Configuration of components
- Access Plane
  - User access to resources
  - Connection to commodity Internet or I2
- Data Plane
  - Connectivity between slivers
  - Experimental data

Control Net

- Control Plane
  - Disk Image Transfer
  - Node Setup
  - Node Health
- Access Plane
  - Not All Sites
  - Tunnels
  - Non-GENI World
  - Internet



Experimental Net

- Data Plane
- Isolated from control & access planes
- Intra-experiment Traffic Only
- Current CMs Support Some of:
  - Connection Between Cluster Nodes
  - Dedicated Bandwidth

Experimental Net

- Can Be Shaped (Emulab Only)
  - Bandwidth
  - Latency
  - Loss
  - Bring Links Down/Up
- Emulate Wide-area Links
  - Any Network Conditions

VLANs

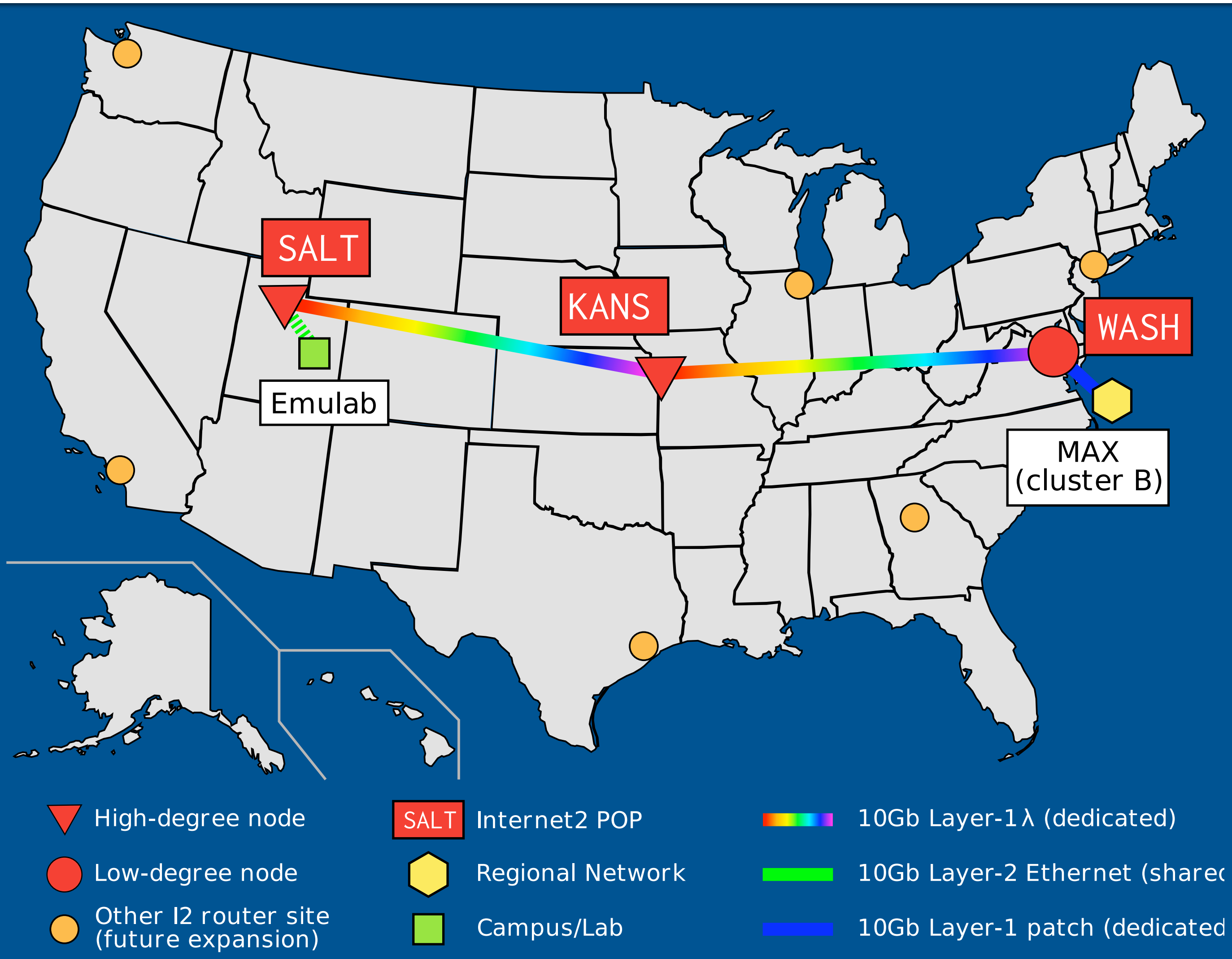
- Within Cluster
  - Wherever Emulab controls switches
  - Over experimental network
  - Dedicated
- Backbone Nodes
  - Using MPLS over Internet 2 (soon)
  - Not Dedicated
- Ethernet Layer
  - IPv4, IPv6, Your IP Replacement

Backbone

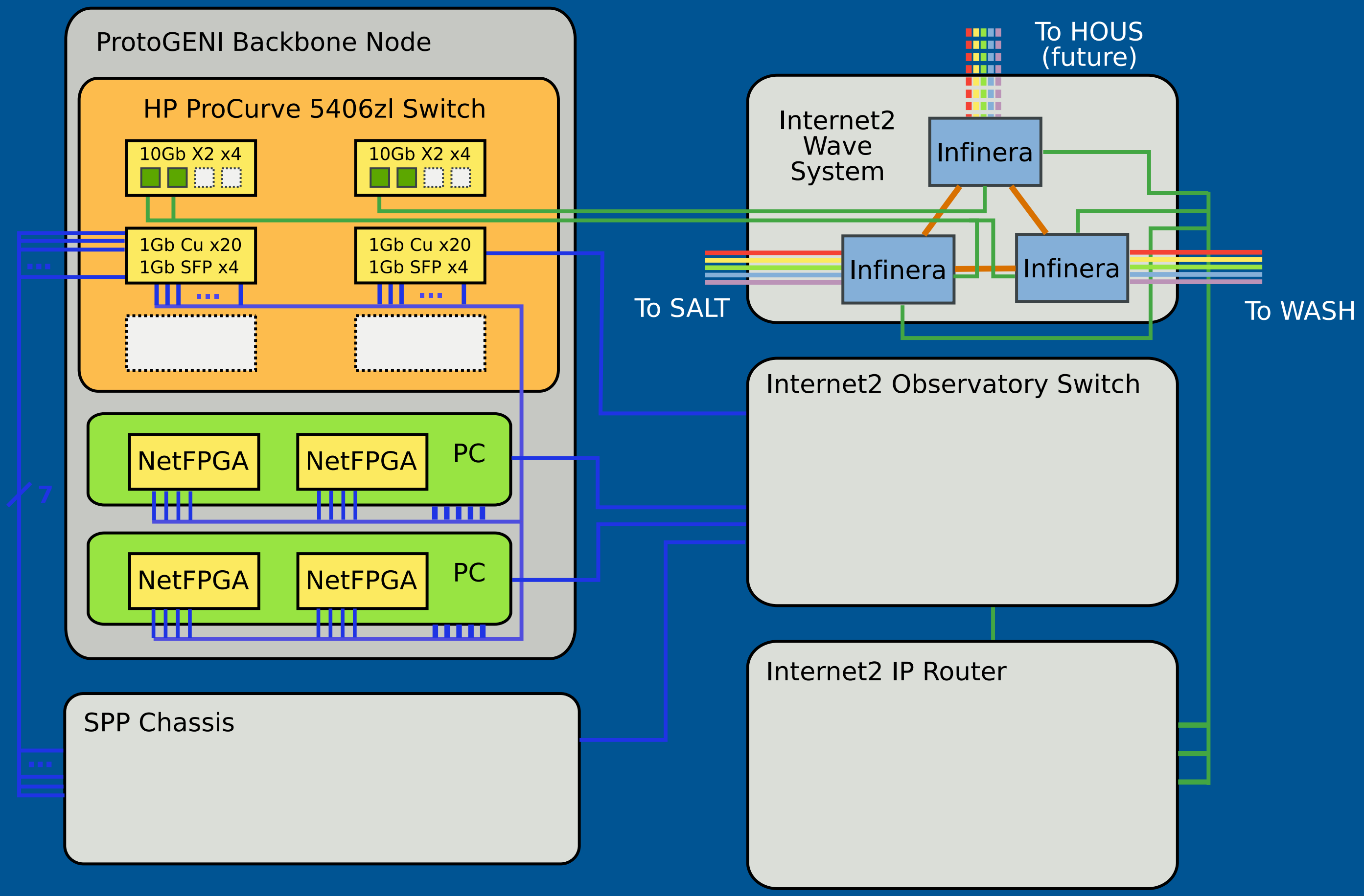
- Nodes in Internet2 POPs and links between them
  - Nature of links changing, but will remain layer 2
  - Allocate just like resources in an Emulab cluster
- Controlled by the Utah component manger
- Connections out to other projects
  - MAX, GpENI, SPP

Backbone Footprint

THE UNIVERSITY OF UTAH



POP Contents (KANS)



Backbone Partners

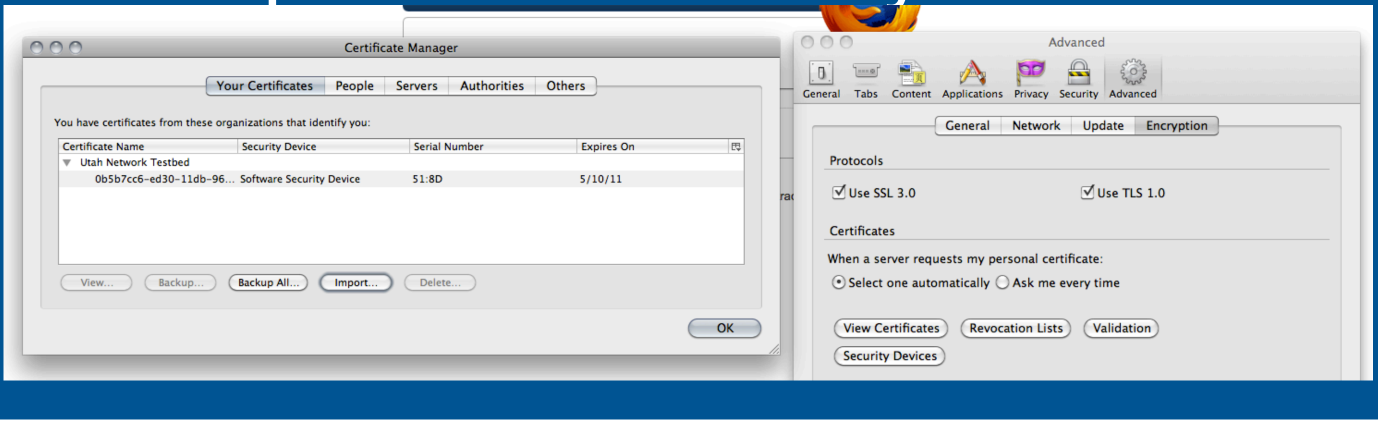
- Internet 2 Ion Service
- SPP Nodes from WUSTL
- Mid-atlantic crossroads
- GPENI

Tunnels

- Over control network
- Between CMs
- Best Effort

Not for Today: Download Certificate

- Click on the download link
- Put the file in your ~/.ssl directory
- Set permissions appropriately
- Optionally, create a file to hold your passphrase to avoid entering repeatedly
- Load the .p12 version into your browser



Certificate Manager

You have certificates from these organizations that identify you:

Certificate Name	Security Device	Serial Number	Expires On
Utah Network Testbed			
0b5b7cc6-ed30-11db-96…	Software Security Device	51:8D	5/10/11

Advanced

Protocols

☑ Use SSL 3.0 ☑ Use TLS 1.0

Certificates

When a server requests my personal certificate:

● Select one automatically ○ Ask me every time

Creating a Slice: Main Objects

Important Objects

- Components
- Aggregates
  - Collections of components
- Slivers
  - Allocated fraction (possibly 100%) of a component
- Slices
  - Container for slivers

Principals

- Users
  - Permissions are defined by the set of credentials they hold
- Authorities
  - Slice Authorities
    - Vouch for (take responsibility for) users and slices
  - Management Authorities
    - Same for components

Services

- Slice Authority
  - Where user has an account, creates slices (names)
- Component Manager (Aggregate Manager)
  - List component resources, create and manage slivers
- Registry/Clearinghouse
  - Place to find lists of SAs and CMs
- Other Services
  - (eg. Slice Embedding Service)

Other Important Things

- URNs
  - Identify objects
- Certificates
  - Authenticate principals (a type of object)
- Credentials
  - Used in authorization decisions
- RSpecs
  - Represent resources

Creating A Slice

The Process

Overview of Slice Creation

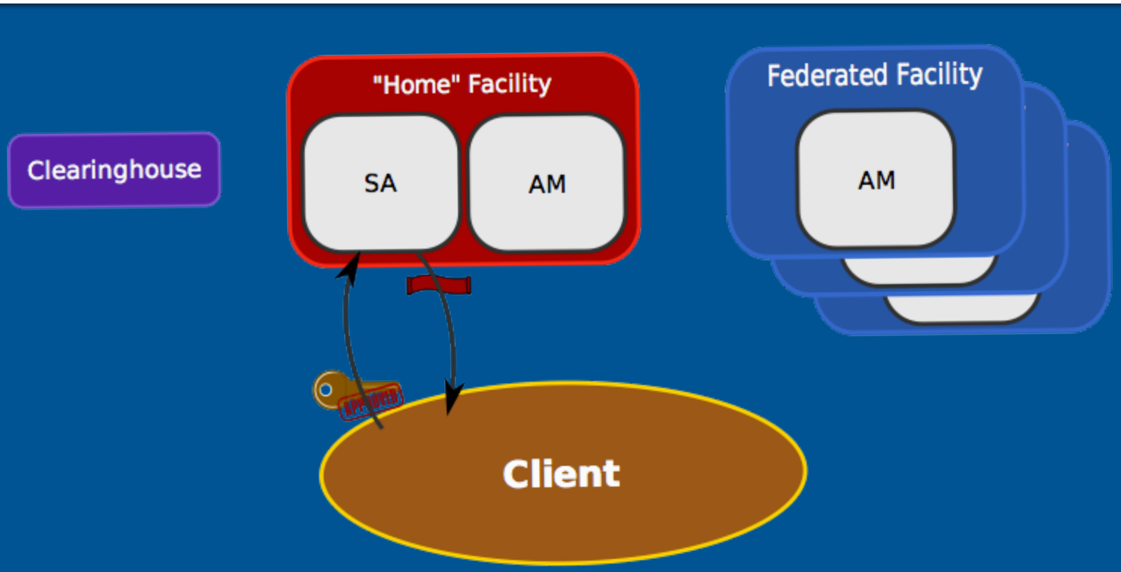
THE UNIVERSITY OF UTAH

- Get an account
- Register your new slice at your Slice Authority
- List component/aggregate managers
- Discover resources
- Allocate resources
- Renew resources

Getting An Account

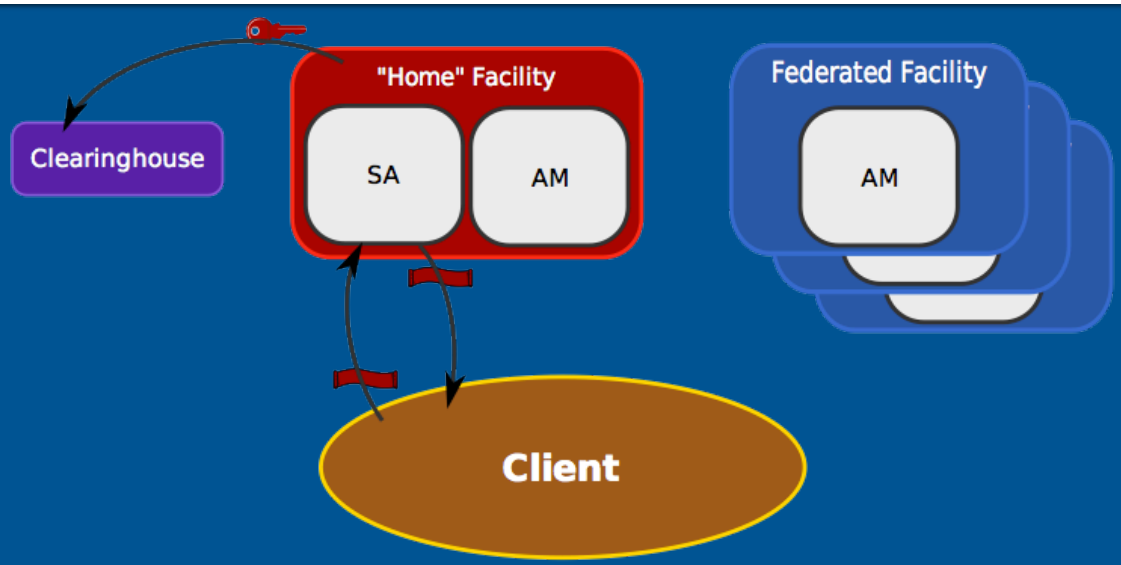
- A URN and certificate will be assigned to you by an authority
  - eg. urn:publicid:IDN+emulab.net+user+ricci
- You'll use these to authenticate yourself to other objects, such as CMs/AMs
  - "Offline" authentication
- You got one automatically by signing up for an Emulab account

Obtaining a Credential

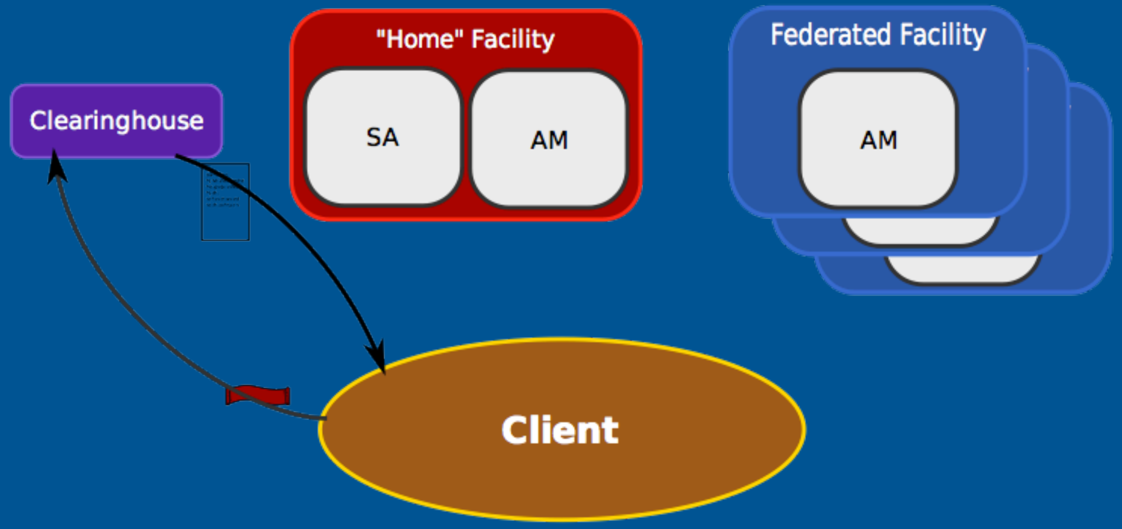


- The client invokes the GetCredential() operation on its home Slice Authority. The client must authenticate with a certified key (thus confirming its identity).
- The Slice Authority verifies the user's information, and supplies a credential conferring privileges to perform subsequent operations at the SA.

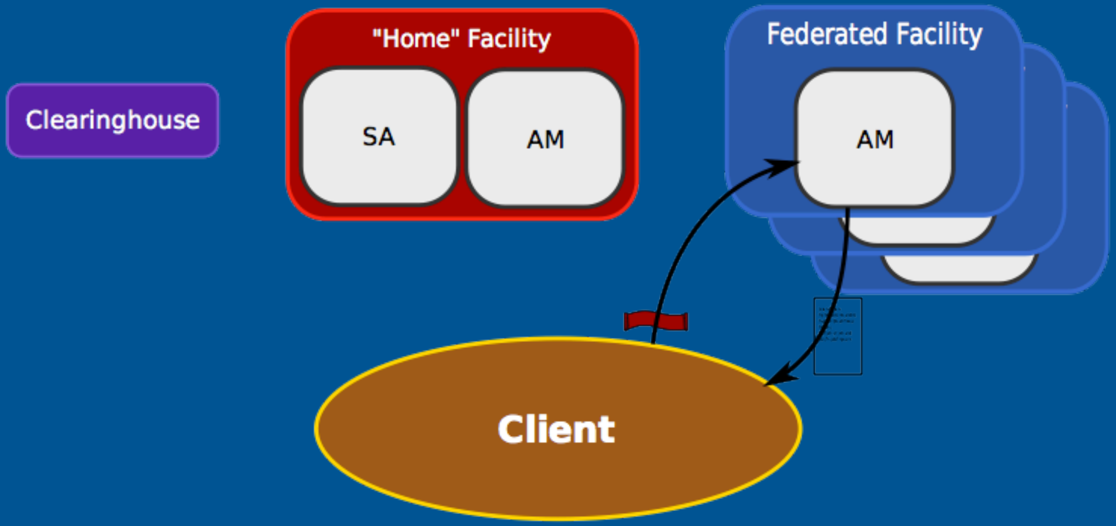
Registering A New Slice



- The client supplies the previous credential and invokes a CreateSlice() operation.
- The SA responds by registering the user and slice name at the Clearinghouse or Registry, and issuing a credential for the newly created slice back to the client.

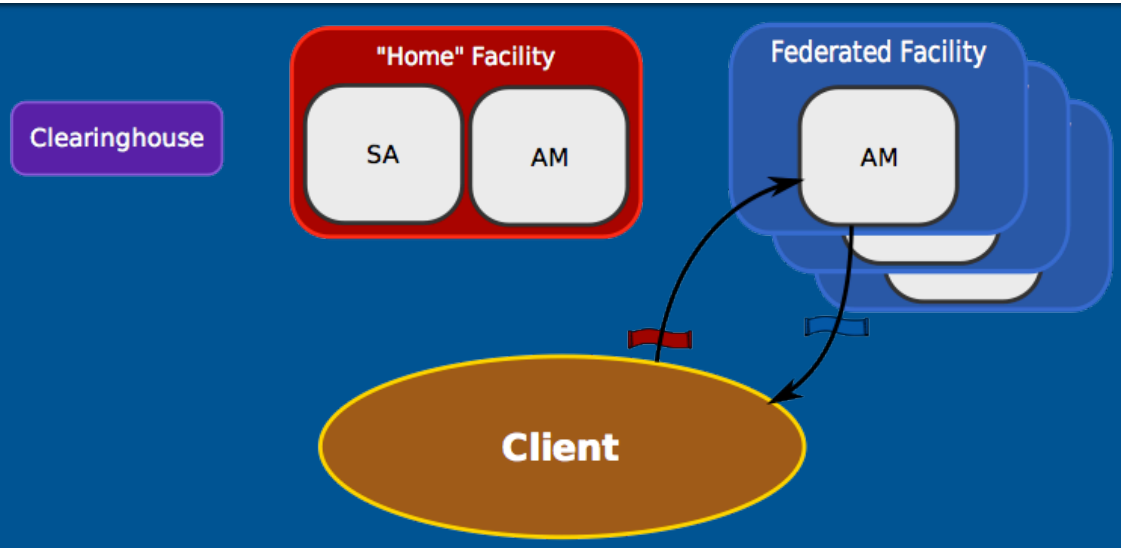


- A user can supply any valid credential from the federation to the Clearinghouse or Registry, and ask for a list of aggregate/ component managers with the ListComponents() call.
- The Clearinghouse responds with a set of name/location tuples, one for each AM/CM in the federation.



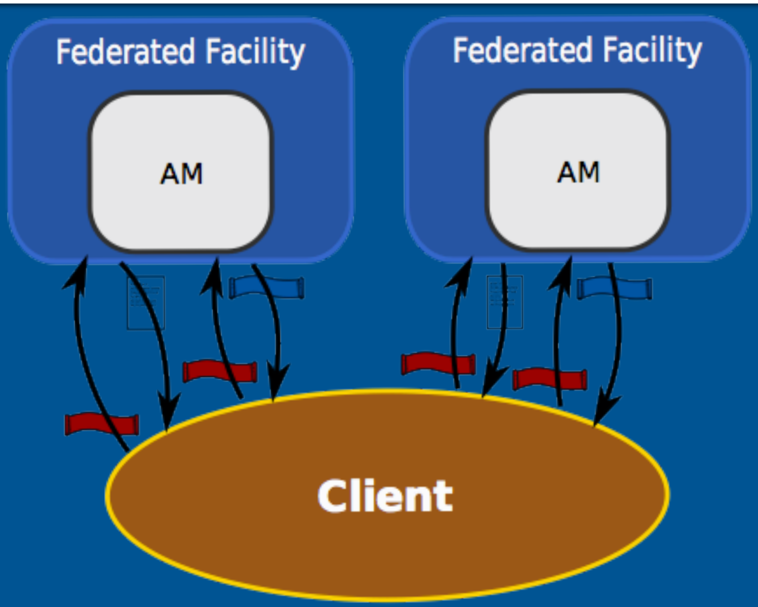
- For detailed information about available resources, the client can contact any aggregate manager with a slice credential and a DiscoverResources() request.
- The aggregate manager responds with an advertisement RSpec describing its available components.
- This operation can be performed multiple times (once per aggregate manager of interest).

Reserving Resources



- Once the client decides on the set of resources to allocate, it constructs a request RSpec and issues it as a CreateSliver() request.
- The aggregate manager makes the components available for the client's use, and responds with a sliver credential.
- If the slice requires resources from multiple sites, this step must be repeated at each applicable aggregate manager.

Cross-Aggregate Slices



- The previous two steps (inquiring about and requesting allocations for components) can be repeated at as many aggregate managers as necessary, all within the same slice.
- Aggregate managers may be capable of provisioning dedicated or tunneled links to components at foreign facilities.

Renewing Resources

THE UNIVERSITY OF UTAH

- All slices and slivers have an expiration date
- Must periodically refresh if you want to keep them longer than this
- Can request longer expiration when creating
- Sliver lifetime cannot exceed slice lifetime

Command-Line Scripts

```
ops> registerslice.py -n myslice
Got my SA credential
No such slice registered here:
   Creating new slice called myslice
New slice created:
   urn:publicid:IDN+emulab.net+slice+myslice
```

Create a Sliver

THE UNIVERSITY OF UTAH

- A sliver is a container for resources at a single Component Manager
- Create an RSpec to describe the resources you want
- Submit your rspec and get back:
- Credential to control your sliver,
- Manifest describing your resources
- Example RSpec on next slide

Trivial RSpec

```
<rspec xmlns="http://protogeni.net/resources/rspec/0.1">
 <node virtual_id="geni1"
       virtualization_type="emulab-vnode"
       exclusive="1">
   <interface virtual_id="virt0"/>
 </node></rspec>
```

- Request a single “wildcard” node.
- Node is exclusive use, not a vnode on a shared node.
- Create your sliver with createsliver.py

createsliver.py

```
ops> createsliver.py —n myslice rspec.xml
Got my SA credential
Asking for slice credential for myslice
Got the slice credential
Creating the Sliver ...
Created the Sliver
<manifest> ... </manifest>
```

Manifest (fragment)

THE
UNIVERSITY
OF UTAH

```
<node virtual_id="geni1"
      component_urn="urn:publicid:IDN+emulab.net+node+pc72"
      sliver_urn="urn:publicid:IDN+emulab.net+sliver+11291”>
    <services>
       <login authentication="ssh-keys"
              hostname="pc72.emulab.net" port="22"/>
    </services></node>
```

- Physical node is “pc72”
- Sliver urn can be used to request another credential in case you lose it
- Hostname is “pc72.emulab.net”
- Login with ssh to port 22 on pc72

Sliver Status

- Createsliver "started" your sliver
- To get status, use sliverstatus.py
- Returns a "hash" array:

```
{'status' : 'ready',
 'state'   : 'started',
 'details': {'urn:publicid:IDN+emulab.net+sliver+11291':
             {'status': 'ready',
          'state'   : 'started',
          'error'   : '',
          'component_urn':
                   'urn:publicid:IDN+emulab.net+node+pc72'}}}
```

Another RSpec

```
<rspec xmlns="http://protogeni.net/resources/rspec/0.1">
 <node virtual_id="geni1"
       virtualization_type="emulab-vnode"
       exclusive="1">
   <interface virtual_id="virt0"/>
 </node>
 <node virtual_id="geni2"
       virtualization_type="emulab-vnode"
       exclusive="1">
   <interface virtual_id="virt0"/>
 </node>
 <link virtual_id="link0">
  <interface_ref virtual_interface_id="virt0"
                 virtual_node_id="geni1"/>
  <interface_ref virtual_interface_id="virt0"
                 virtual_node_id="geni2"/>
 </link>
</rspec>
```

RSpec Details

```
<link virtual_id="link0">
  <interface_ref virtual_interface_id="virt0"
                 virtual_node_id="geni1"/>
  <interface_ref virtual_interface_id="virt0"
                 virtual_node_id="geni2"/>
 </link>
```

- **Two wildcarded nodes**
- **Link between “geni1” and “geni2”**
- **Uses the interfaces named in the “interface” section of the node section**

Restarting your Sliver

- Createsliver.py started your slivers
- You can restart/stop/startyour slivers if they get into a bad state
- “restart” is the most common operation
- Use the sliveraction.py script

sliveraction.py

```
ops> sliveraction.py -n myslice restart
Got my SA credential.
Looking for slice ...
Found the slice,
    asking for a slice credential ...
Got the slice credential,
    asking for a sliver credential ...
Got the sliver credential,
    calling RestartSliver on the sliver
Sliver has been restart'ed.
```

Wait for “ready”

- After “start” or “restart”, must wait for nodes to come “ready”
- ie: nodes have to finish rebooting
- Use sliverstatus.py, described earlier
- Returns immediately, so need to poll (but not too fast)

Login

- Login into your nodes with ssh
- No passwords, public key only
- Use sudo to do “root” things (configure interfaces, install software, etc.)
- Write papers, get famous, send us money

More on ssh keys

- Your ssh keys are given to the CM in the RedeemTicket() method
- For convenience, you can request them in a bundle; see lookupuser.py
- createsliver.py does this for you
- Upload additional public keys via the Emulab web interface

Renew your Sliver

- Slivers time out after a short time
- Renew them so they are not expired
- Argument is number of minutes
- Use the renewsliver.py script
- But please, do not waste resources

Delete your Sliver

THE UNIVERSITY OF UTAH

- Slivers eventually timeout
- But remember to delete them anyway
- Use the deleteslice.py script:

```
ops> deleteslice.py -n myslice
Got my SA credential.
Looking for slice ...
Found the slice,
    asking for a credential ...
Got the slice credential,
    deleting the slice ...
Slice has been deleted.
```

Using other CMs

- **Use listcomponents.py to get list of CMs**

```
listcomponents.py
```

```
https://www.emulab.net/protogeni/xmlrpc/cm
https://www.uky.emulab.net/protogeni/xmlrpc/cm
https://www.schooner.wail.wisc.edu/protogeni/xmlrpc/cm
https://www.pgeni.gpolab.bbn.com/protogeni/xmlrpc/cm
https://boss.uml.emulab.net/protogeni/xmlrpc/cm
https://boss.cmcl.cs.cmu.edu/protogeni/xmlrpc/cm
https://myboss.jonlab.geni.emulab.net/protogeni/xmlrpc/cm
https://myboss.myelab.testbed.emulab.net/protogeni/xmlrpc/cm
```

Using other CMs, cont.

- **All scripts take an option to specify CM**

```
-m https://www.emulab.net/protogeni/xmlrpc/cm
```

- **Not all CMs have usable nodes**
- **Use discover.py to get advertisement**

Discovering Resources

THE UNIVERSITY OF UTAH

- What resources are available for RSpec?
- Use Discover (resources) to download advertisements from CMs
- Advertisements describe each node and link in detail

```
discover.py > advert.xml
```

ProtoGENI APIs

API Overview

- XMLRPC over HTTPS
- In "python speak," methods take a hashed array and return a hashed array
- Authentication done with X.509 certs

Arguments

- Most methods require at least one argument; a set of credentials
  - Each credential has a “target” and an “owner”
  - You generally get a credential to control an object as the return value for creating the object
  - Credentials are delegatable
- Objects to act on are named by a URN:
  - urn:publicid:IDN+emulab.net+slice+lbstestslice

Return Value

- All methods return a “GeniResponse”
  - An integer result code,
  - An optional string to aid debugging,
  - An arbitrary return value
  - Many include credentials
- The result code mimics unix “errno”

```
{“code”   : $code,
 “output” : $string,
 “value”  : $value}
```

Trivial Example

- Lets look at GetVersion:
  - response = GetVersion({})
- Where response is:

```
{'output' : '',
 'code'   : 0,
 'value'  : {'output_rspec' : 0.1,
             'api'          : 2,
             'input_rspec'  : [0.1, 2],
             'level'        : 1}}
```

“Self” Credential

- You need your “self” credential to get started:
  - response = GetCredential({})
- The response contains:
  - <signed-credential> … </signed-credential>
- Use this credential to say, register a slice name or get a copy of a slice credential

Register a Slice Name

- Register a new slice called “myslice”

```
Register({“type”      : “slice”,
        “urn”       : “urn:…+myslice”,
        “credential” : mycredential})
```

- Returns a slice credential
- Use this new credential to create a sliver

Documentation

THE UNIVERSITY OF UTAH

- Full documentation can be found at:
  - http://www.protogeni.net/trac/protogeni/wiki/API
- Example calls:
  - CreateSlice()
  - CreateSliver()
  - RenewSlice()/RenewSliver()
  - StopSliver()/RestartSliver()

More API Details

API Overview

- XMLRPC over HTTPS
- In “python speak,” methods take a hashed array and return a hashed array
- All of our example clients are written in Python
- But you can use any language that supports XMLRPC over SSL

Arguments

- Most methods require at least one argument; a credential
  - You generally get a credential to control an object as the return value for creating the object
- Most objects are named by a URN:
  - urn:publicid:IDN+emulab.net+slice+lbstestslice
- Always present as an array:
  - {"arg1" : arg1, "arg2" : arg2, …}

Return Value

- All methods return a “GeniResponse”
  - An integer result code,
  - An optional string to aid debugging,
  - An arbitrary return value
- The result code mimics unix “errno”

```
{“code”   : $code,
 “output” : $string,
 “value”  : $value}
```

Trivial Example

- Lets look at GetVersion:
  - response = GetVersion({})
- Where response is:

```
{'output' : '',
 'code'   : 0,
 'value'  : {'output_rspec' : 0.1,
             'api'          : 2,
             'input_rspec'  : [0.1, 2],
             'level'        : 1}}
```

“Self” Credential

- You need your “self” credential to get started:
  - response = GetCredential({})
- The response contains:
  - <signed-credential> … </signed-credential>
- Use this credential to say, register a slice name or get a copy of a slice credential

Register a Slice Name

- Register a new slice called “myslice”

```
Register({“type”      : “slice”,
       “urn”       : “urn:...+myslice”,
       “credential” : mycredential})
```

- Returns a slice credential
- Use this new credential to create a sliver

Documentation

- Full documentation can be found at:
  - http://www.protogeni.net/trac/protogeni/wiki/API