

Are you ready for the tutorial?

Grab a Worksheet and Instructions

Did you do the pre-work?

- 1. Have an account at the GENI Portal.
- 2. Be a member of the GEC20 Project.
- 3. Installed the Omni tools version 2.6.
- 4. Configured the Omni tools.
- 5. Able to SSH from your PC.

Pre-work at:

<http://tinyurl.com/gec20iaprework>

Tutorial: Inter-Aggregate Experiments

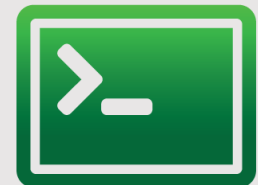
Aaron Helsing,
Luisa Nevers,
Niky Riga
GENI Project Office

{ahelsing, lnevers, nriga}@bbn.com

GEC20: June 22, 2014



Design/Setup




Execute



Finish

GENI: Infrastructure for Experimentation



Regional nets

-  Existing
-  New

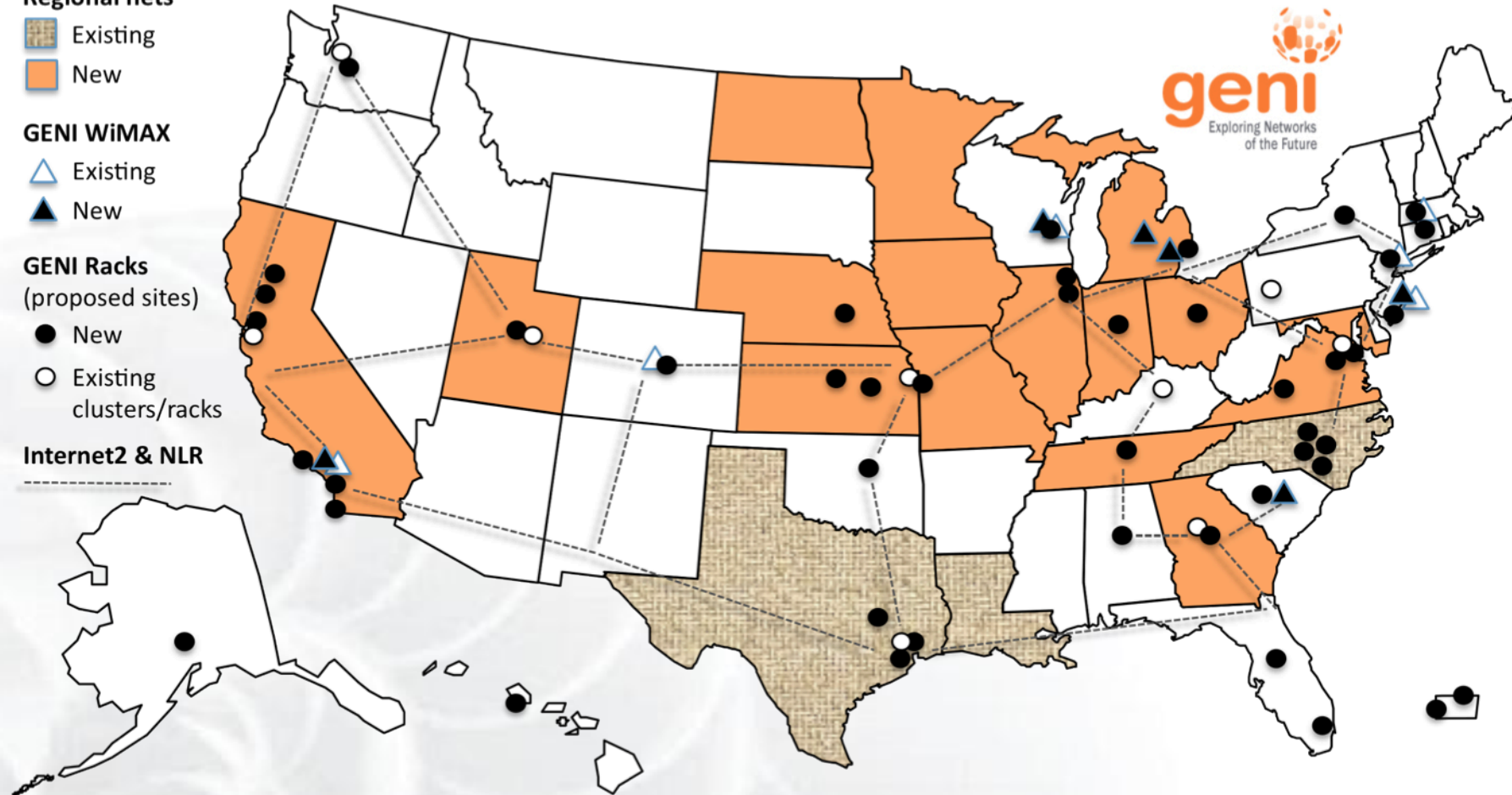
GENI WiMAX

-  Existing
-  New

GENI Racks (proposed sites)

-  New
-  Existing clusters/racks

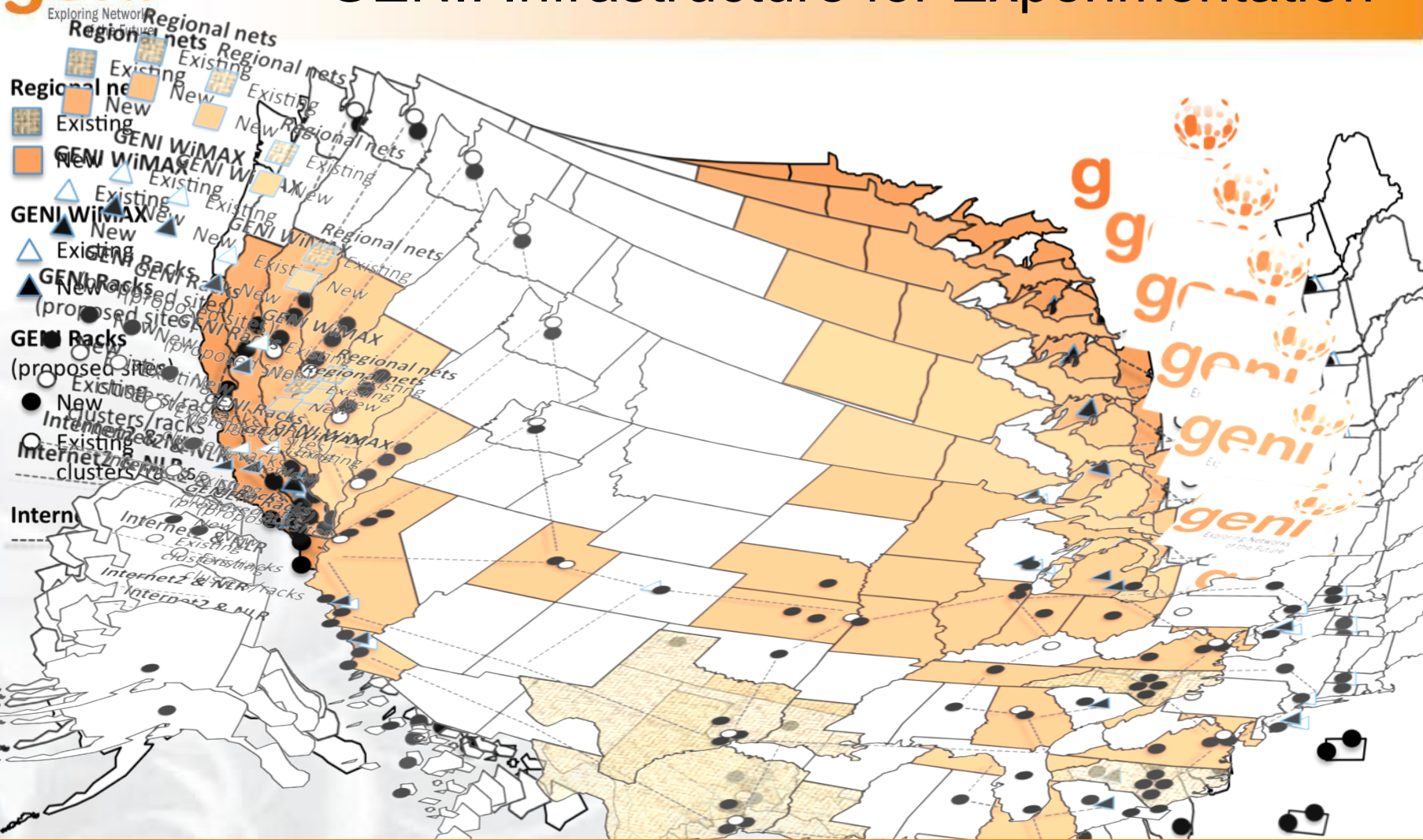
Internet2 & NLR



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



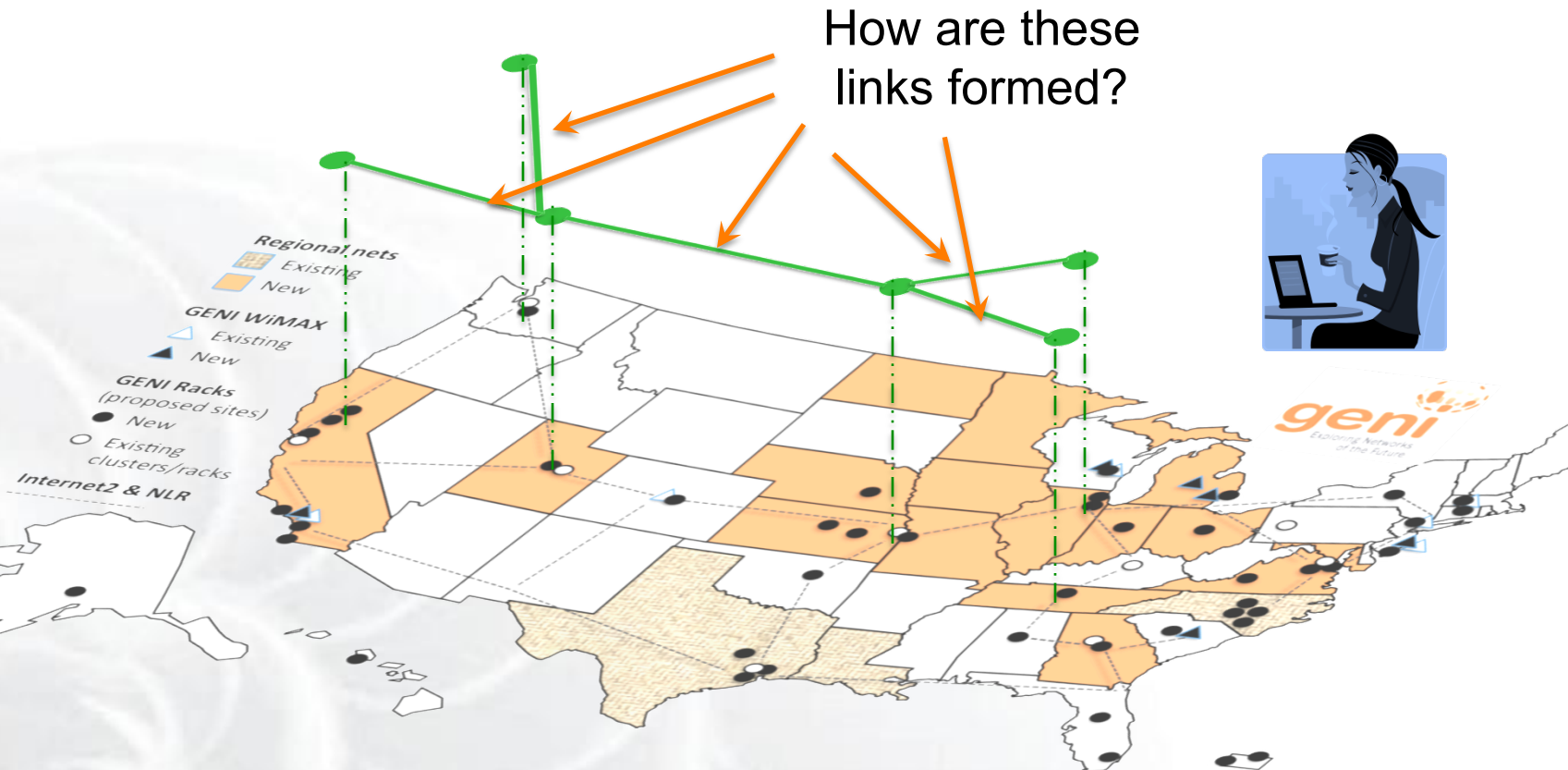
GENI: Infrastructure for Experimentation



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



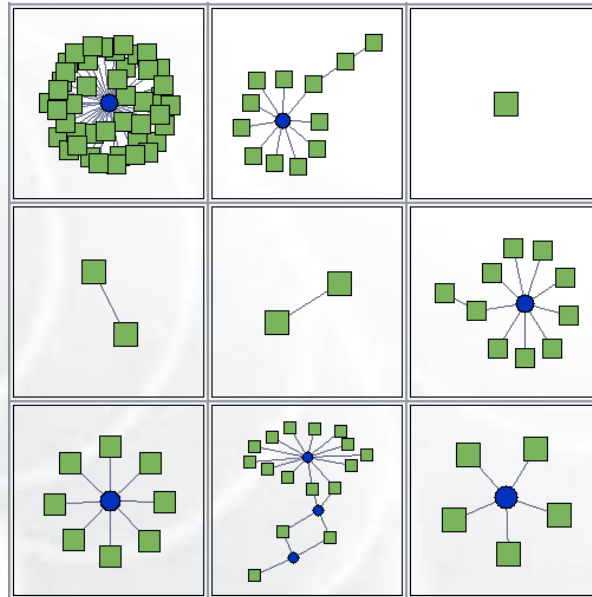
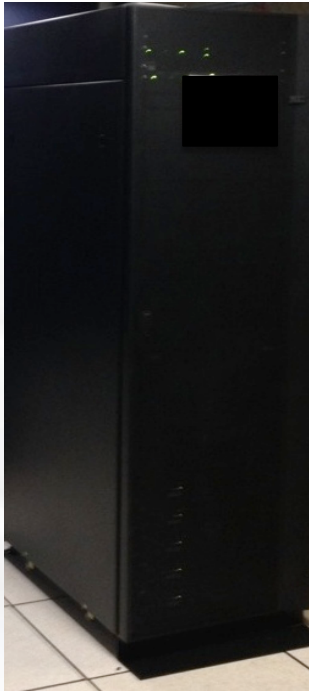
Inter aggregate connectivity



Experiments live in **isolated “slices”**

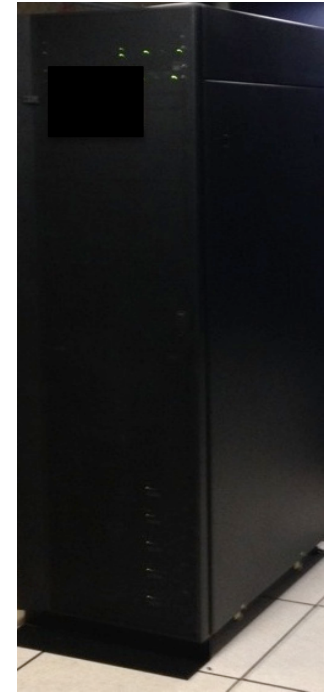
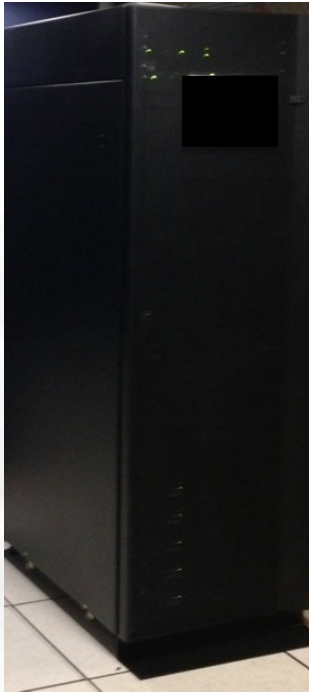
Separate Aggregate Manager

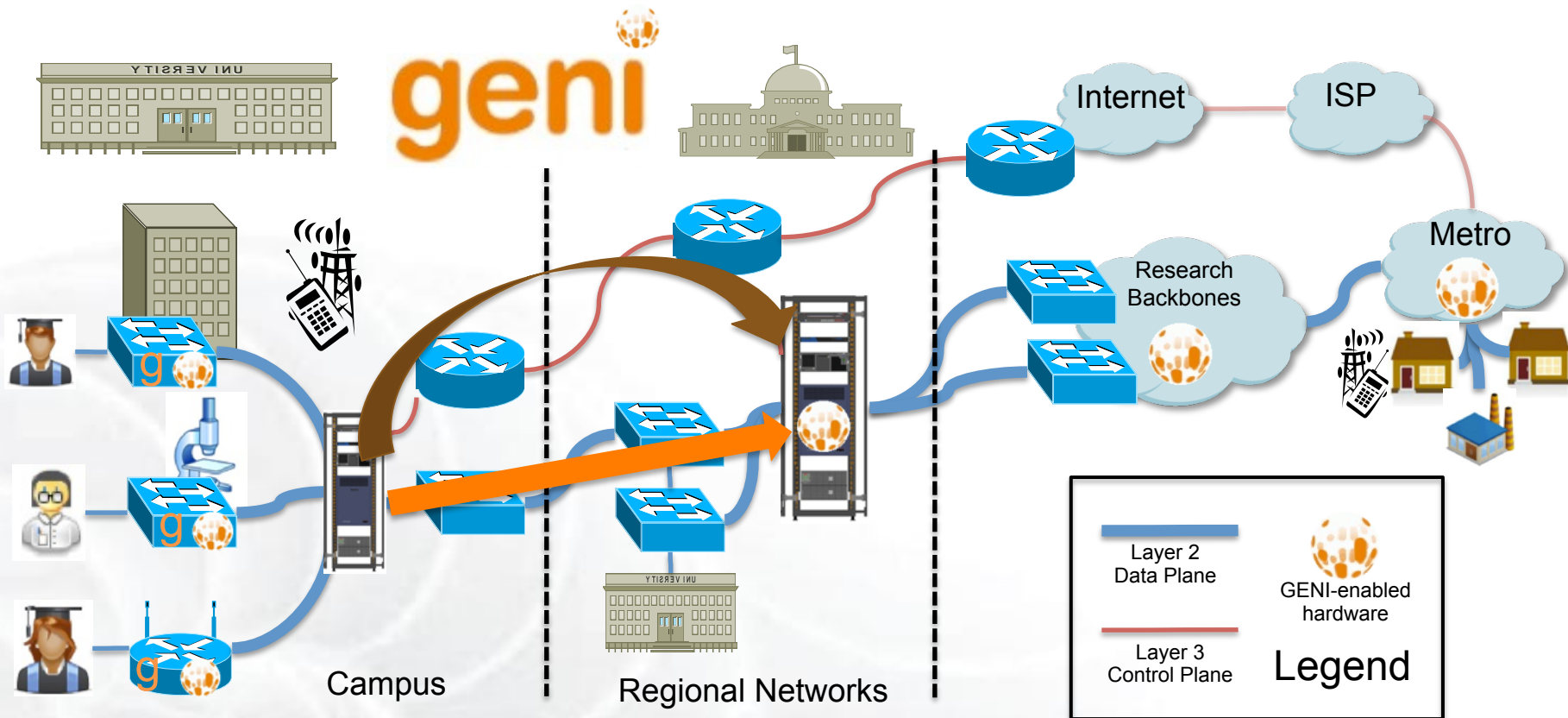
- 100s of VMs
- Bare metal hosts
- OpenFlow switch



Many topologies embedded in
only one AM

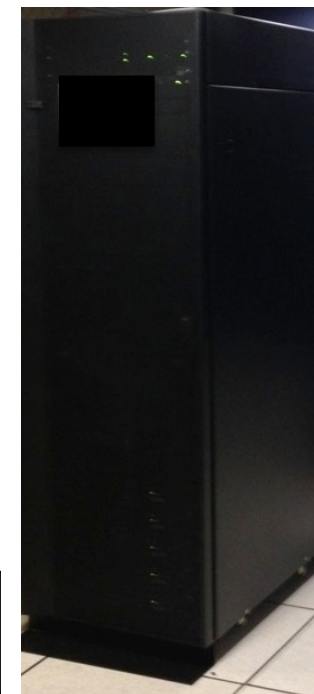
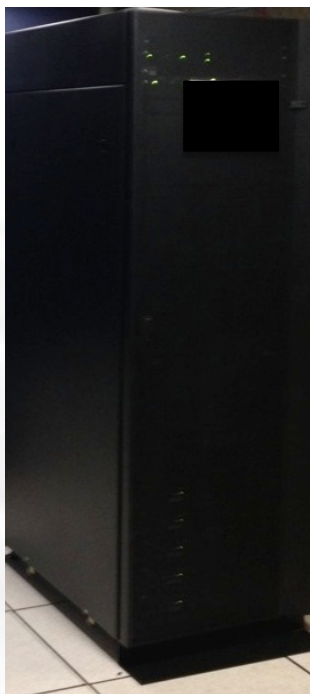
Inter AM Connectivity Options





Control Plane : Over the Internet
Data Plane : Over the GENI backbone

Inter AM Connectivity Options



	Control Plane	Data Plane
Static	Control Interface	Preconfigured VLANs
Dynamic	GRE Tunnels	GENI Stitching EG Stitching

- Already setup, tested, stable
 - Universally supported
-
- Limited bandwidth
 - Shared 100 Mbps with all experimenters
 - Shared bandwidth with the rest of the Internet
 - Only Layer 3
 - Firewall

More bandwidth

Shared 1Gbps for most sites

Layer 2 connection

No Firewalls

Use of HW OpenFlow switches

Limited availability

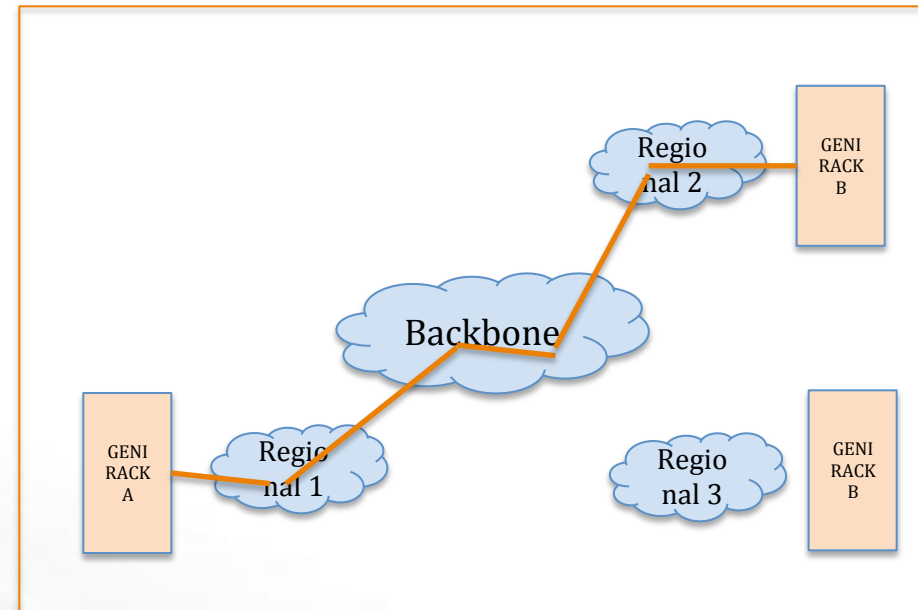
Limited topologies

Limited sites

Harder to setup/use

Today's tutorial will use the Data Plane

- Setup point-to-point VLANs
 - Between hosts on different AMs
 - One host/per AM/per stitch
 - Not a broadcast domain
- Dynamic, real-time setup
 - Need to coordinate multiple AMs
 - Takes time
 - Can fail
- Provides traffic isolation and bandwidth constraints



A common concept used in other networks, applied to GENI, e.g. OSCARS, GLIF

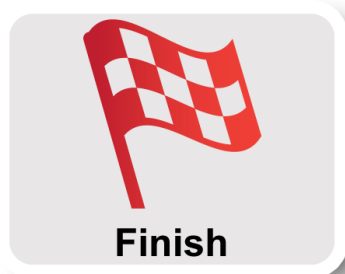
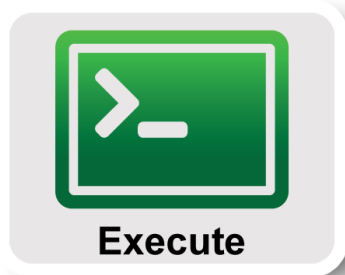
Command line tool

- Distributed with Omni/gcf
- Script built on top of Omni
- Has same Omni commands

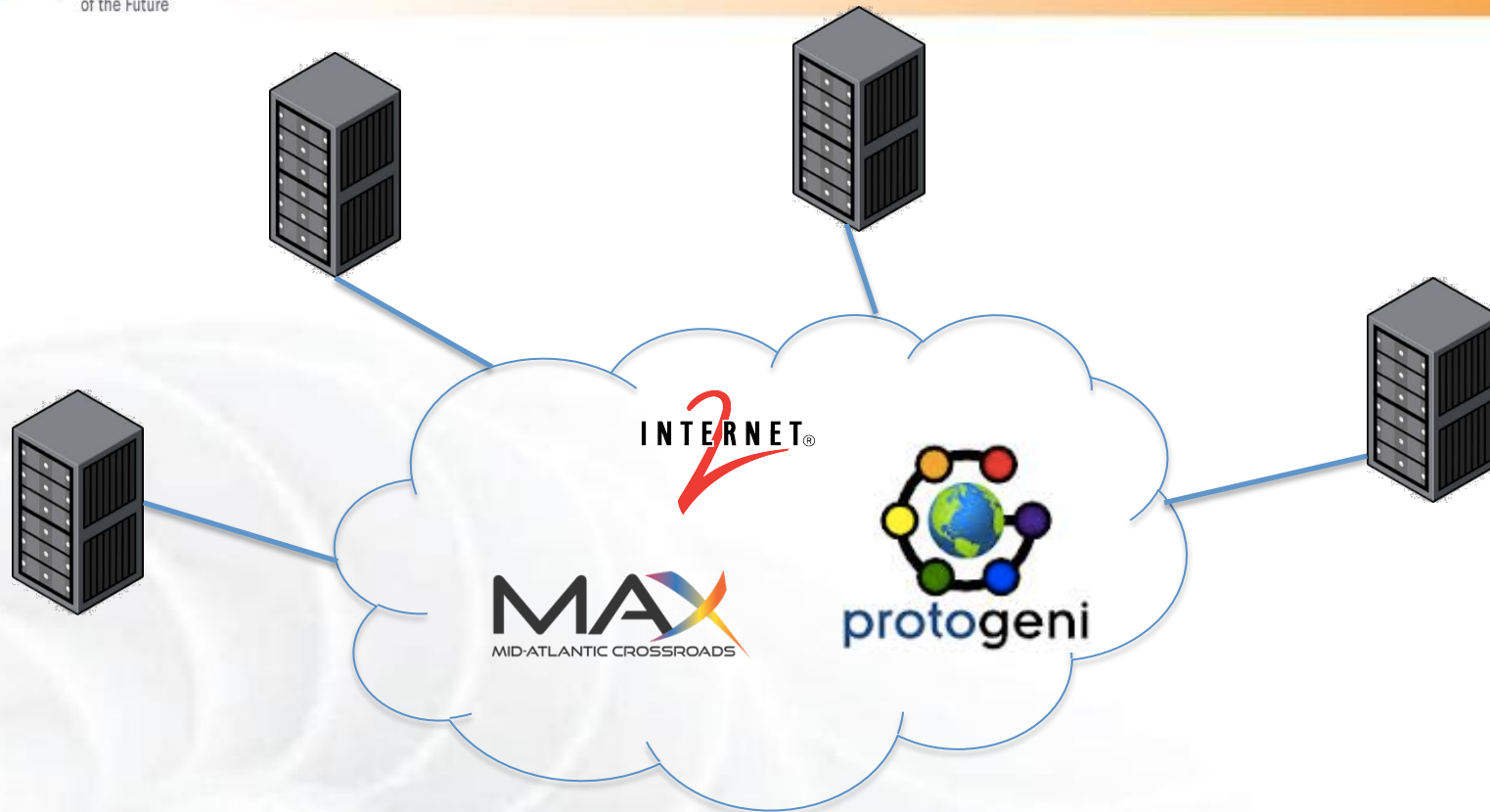
It takes care of:

- Path computation
- AM coordination
- Error handling and recovery

```
$ stitcher createsliver myslice myRspec.xml
INFO    stitcher: Loading agg_nick_cache fi
INFO    stitcher: Loading config file /User
INFO    stitcher: Using control framework p
INFO    stitcher: Slice urn:publicid:IDN+ch
        expires on 2014-03-21 03:2
INFO    stitcher: Stitched reservation will
INFO    stitcher:   <Aggregate nysernet-ig>
INFO    stitcher:   <Aggregate missouri-ig>
INFO    stitcher:   <Aggregate utahddc-ig>
INFO    stitcher:   <Aggregate ion>
INFO    stitcher:   <Aggregate stanford-ig>
INFO    stitcher:   <Aggregate wisconsin-ig>
INFO    stitcher:   <Aggregate utah-pg>
INFO    stitch.Aggregate:
        Stitcher doing createsliver at
        https://www.instageni.nysernet.org:12369/p
```



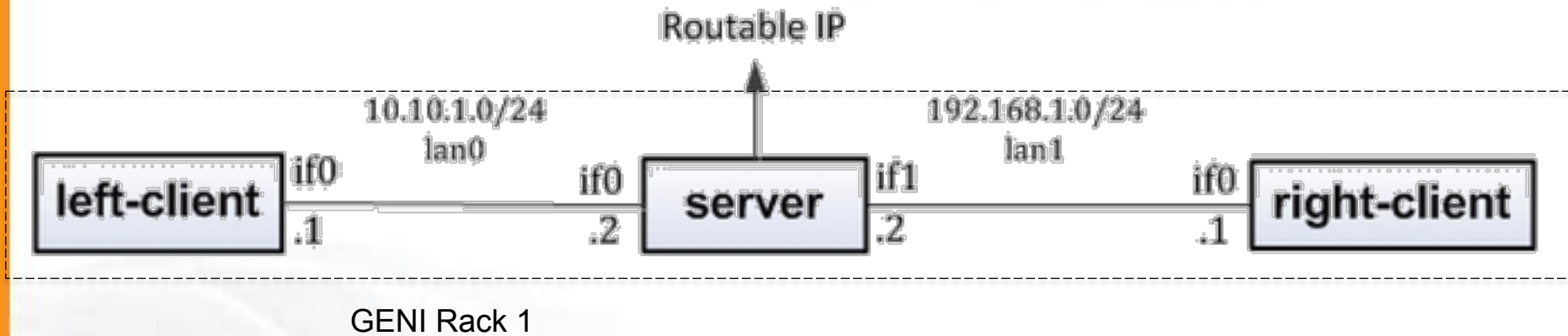
- **Part I: Design/Setup**
 - Design / Tutorial Logistics
 - Craft your RSpecs
 - Obtain Resources
- **Part II: Execute**
 - Execute Experiment
 - Collect Measurements
- **Part III: Finish**
 - Teardown Experiment



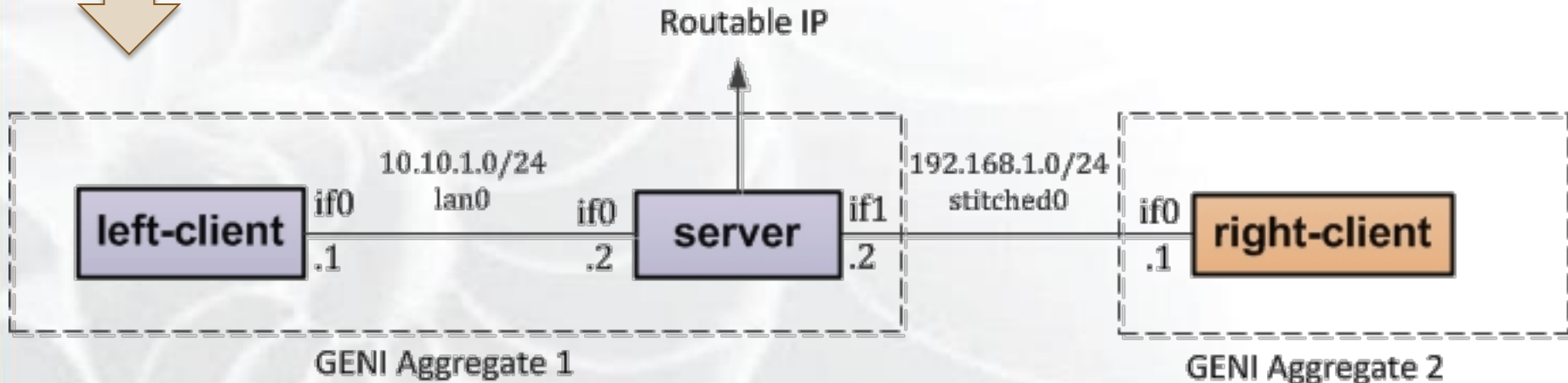
Use InstaGENI Racks across the US and stitch across one or more of the above three networks

- **Use** the information on **your worksheet**
- We will reserve resources in **5 groups**
 - To avoid overloading the Internet2 (ION) aggregate
 - Your group # is on your worksheet
- Seeing errors in the logs is normal, most of them are not fatal

Wait for the blinking ball before running 'stitcher' and reserving your resources



Go from single AM experiment to multi-AM



1. Log in to the Portal and create a slice in the GEC20 project
2. Launch Flack and load the RSpec
 - URL in online instructions:
 - <http://www.gpolab.bbn.com/exp/StitchingTutorial/3nodes-1rack.rspec>
1. Adjust the topology
 - **Use *your* AMs,**
 - Make the inter-AM link of type 'stitched'
2. Download the edited RSpec



Wait!

Run:

```
$ stitcher.py createsliver <slice> <rspec>
```

You should see something like:

```
14:37:48 INFO      : Configured logging from file ...stitcher_logging.conf
14:37:48 INFO      : Reading slice ... credential...
14:38:04 INFO      : Slice ..+slice+..expires on 2014-.. UTC
14:38:04 INFO      : Calling SCS...
14:38:35 INFO      : Multi-AM reservation will include resources from these
aggregates:
14:38:35 INFO      :           <Aggregate ...>
14:38:35 INFO      :           <Aggregate ...>
14:38:35 INFO      :           <Aggregate ...>
14:38:35 INFO      :           <Aggregate ion>
14:38:35 INFO      : Stitcher doing createsliver at <Aggregate ...>...
```

When you've gotten that, raise your hand



- Creating dynamic circuits takes time
 - Particularly across Internet2 ION
 - Negotiate a VLAN tag that is free at each aggregate
- You will see output messages that look like errors
 - For now, let them go
- While stitcher works, we'll explain what is going on



How does GENI Stitching Work?

1. Rack Configuration (network admins)

- Long process (~weeks, months)
- Done once in advance
- Manual



2. Inter-aggregate link reservations (experimenters)

- Automated (tools can make them)
- Quick_{ish} (~minutes, hours)
- Live, Easy
- Repeatable

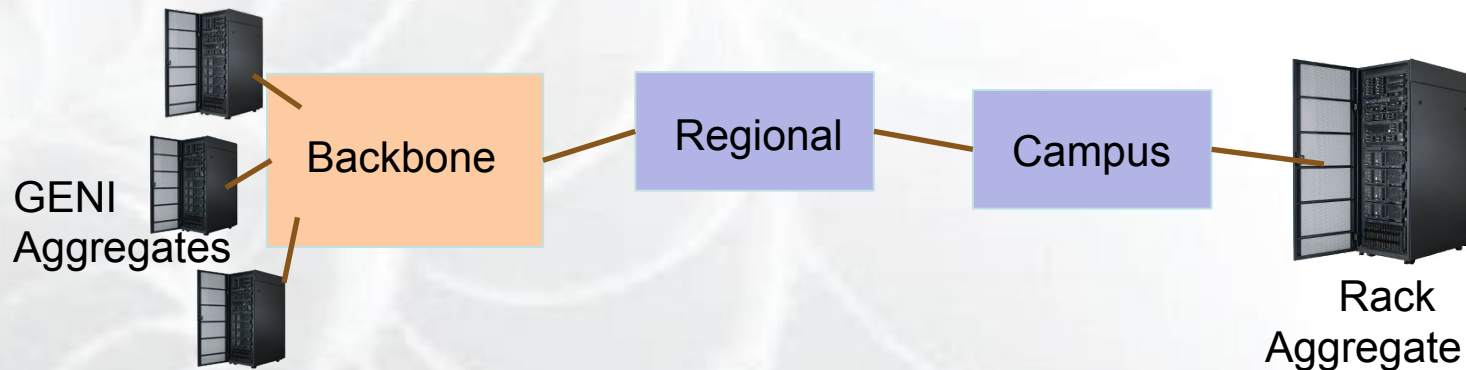


- Rack are pre-configured for GENI Stitching by network engineers
 - Not all racks are configured!
 - *See portal for stitchable AMs*
- We use VLANs – they are limited
- Most circuits will go across the ION aggregate
 - Circuits there are complex
 - This takes time, and can fail

More Details Later....

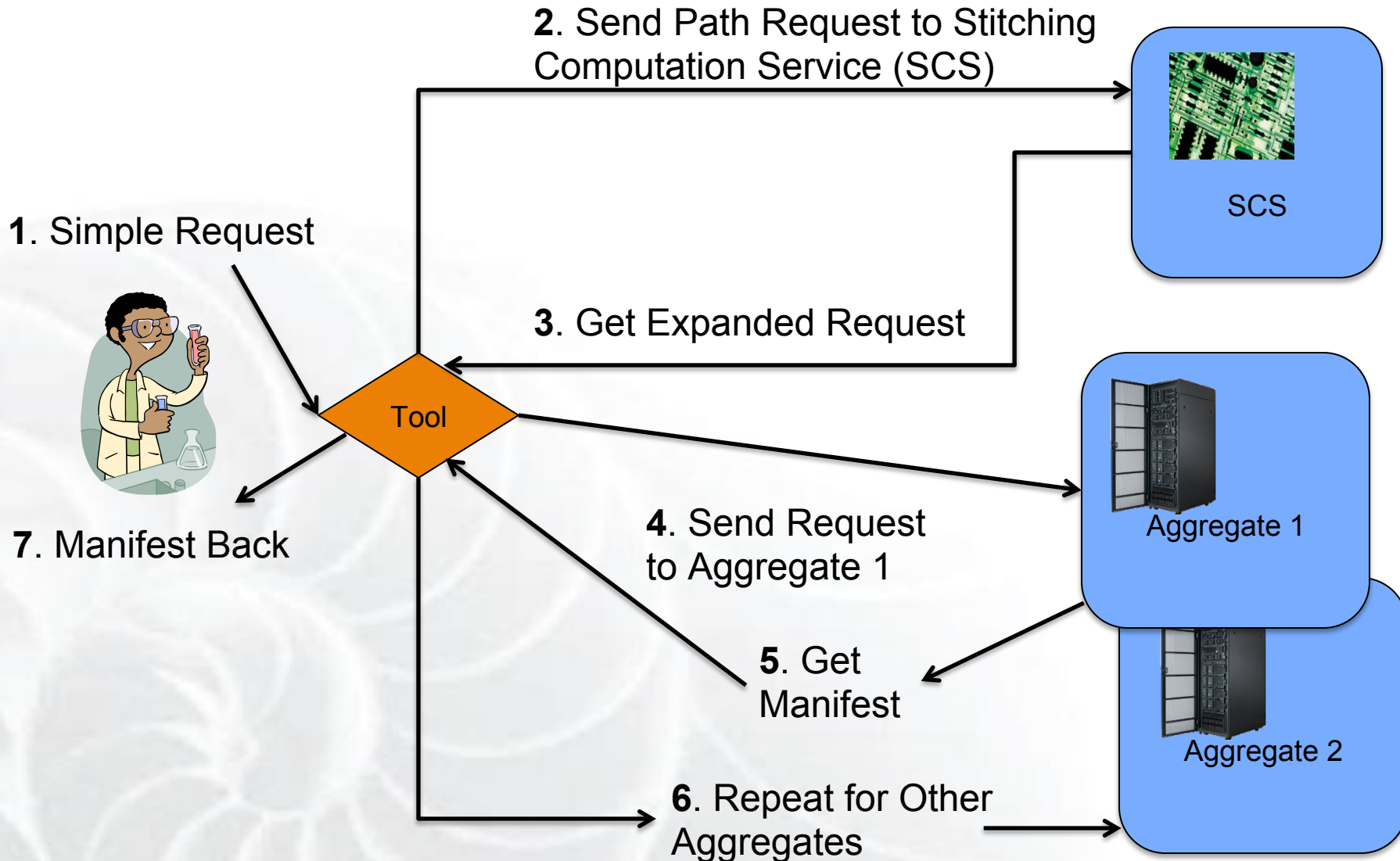
GENI Stitching Pre-Configuration

- Identify paths from a rack to GENI backbone
- Identify the network providers
 - Typically a campus, a regional, and the backbone
- Identify the **endpoints** and **VLAN tags** that can be used to connect to the rack





Experimenter: Creating a Circuit





Experimenter: Creating a Circuit

**Automated
by the tool**

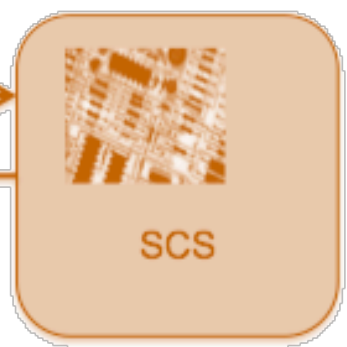
1. Simple Request



7. Manifest Back



2. Send Path Request to Stitching
Computation Service (SCS)



3. Get Expanded Request

4. Send Request
to Aggregate 1



5. Get
Manifest



6. Repeat for Other
Aggregates

1. Write one RSpec for all your resources

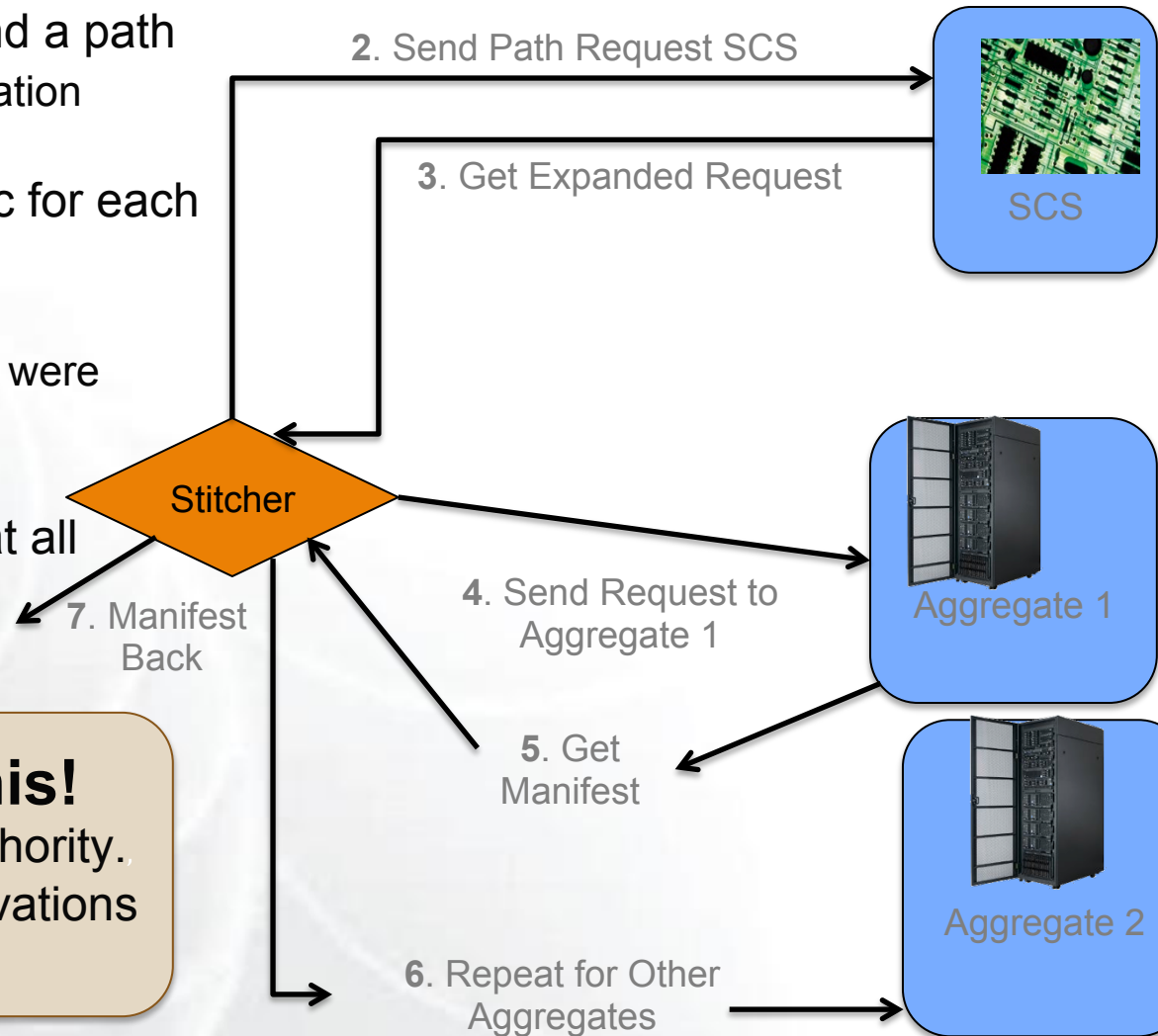
- ◆ Assign nodes to specific aggregates
- ◆ Add a link between hosts at 2 aggregates (or >1 links!)
 - ◆ 2 different `<component_manager>` tags
 - ◆ You can specify particular VLAN tags if you choose to – **not advised!**

2. Reserve your resources

- ◆ Links and compute nodes reserved together
- ◆ Remember: links are node-to-node
 - ◆ Multi-point VLANs are not supported

Tool: Creating a Circuit

- 2-3. Expand your request to find a path
 - Using the Stitching Computation Service (SCS)
- 4-5. Generate a request RSpec for each aggregate and make the reservations
 - Check that dynamic circuits were successfully created
7. Report back a combined summary of what you have at all the aggregates



Any tool can do this!

No central GENI stitching authority.
Stitching is just a set of reservations
at multiple aggregates.

```
14:43:39 INFO      : All aggregates are complete.
14:43:39 INFO      : Your resources expire at 3
different times at different AMs. The first expiration
is 2014-...
14:43:39 INFO      : Saved combined reservation RSpec at
4 AMs to file '...-manifest-rspec-stitching-
combined.xml'
Success: Reserved resources in slice ahtest at 4
Aggregates (including 2 intermediate aggregate(s) not
in the original request), creating 1 stitched link(s).
```

**If you've gotten that,
raise your hand**



Run:

```
$ stitcher.py createsliver <slice> <rspec>
```

You should see something like:

```
14:37:48 INFO      : Configured logging from file ...stitcher_logging.conf
14:37:48 INFO      : Reading slice ... credential...
14:38:04 INFO      : Slice ..+slice+..expires on 2014-.. UTC
14:38:04 INFO      : Calling SCS...
14:38:35 INFO      : Multi-AM reservation will include resources from these
aggregates:
14:38:35 INFO      :           <Aggregate ...>
14:38:35 INFO      :           <Aggregate ...>
14:38:35 INFO      :           <Aggregate ...>
14:38:35 INFO      :           <Aggregate ion>
14:38:35 INFO      : Stitcher doing createsliver at <Aggregate ...>...
```

When you've gotten that, raise your hand

Stitcher: What is Happening?

If you have used stitcher, you know it prints a lot of log messages. What is it doing?

```
...
14:37:48 INFO      : Configured logging from file ../stitcher_logging.conf
14:37:48 INFO      : Reading slice ../credential...
14:38:04 INFO      : Slice ../slice+../expires on 2014-.. UTC
14:38:04 INFO      : Calling SCS...
14:38:35 INFO      : Multi-AM reservation will include resources from these
aggregates:
14:38:35 INFO      :           <Aggregate ...>
14:38:35 INFO      :           <Aggregate ...>
14:38:35 INFO      :           <Aggregate ...>
14:38:35 INFO      :           <Aggregate ion>
14:38:35 INFO      : Stitcher doing createsliver at <Aggregate ...>...
14:39:18 INFO      : ... Allocation at <Aggregate ...> complete.
...
14:41:11 INFO      : Stitcher doing createsliver at <Aggregate ion>...
14:42:25 INFO      : DCN AM <Aggregate ion>: must wait for status ready....
14:42:25 INFO      : Pausing 30 seconds to let circuit become ready...
```

```
14:37:48 INFO      : Reading slice .. credential...  
14:38:04 INFO      : Slice ..+slice+..expires on 2014-..  
UTC
```

Stitcher does these early steps:

1. Check the RSpec
 - Does it require stitching? Malformed?
2. Get your slice credential
 - Do it once to save time

Stitcher Expands your Request

- Your request RSpec doesn't need to say how to get from AM1 to AM2
- Stitcher expands your request with stitching information
 - Finds a Path
 - Gets extra aggregate information

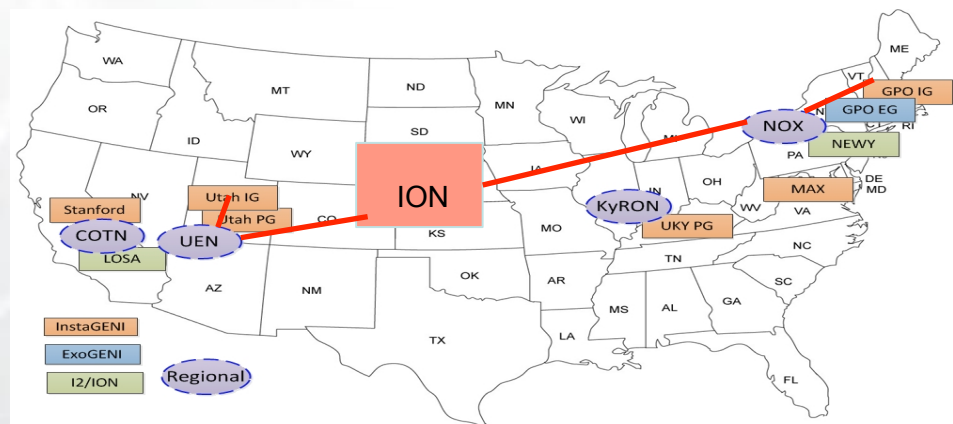
You will see:

```
14:38:04 INFO      : Calling SCS...
14:38:35 INFO      : Multi-AM reservation will include
resources from these aggregates:
14:38:35 INFO      :          <Aggregate ...>
14:38:35 INFO      :          <Aggregate ...>
14:38:35 INFO      :          <Aggregate ...>
14:38:35 INFO      :          <Aggregate ion>
```


Stitching Computation Service

Finding a workable path, and the right reservation order can be hard.

- **Stitching Computation Service (SCS) for path and workflow computation**
 - Tom Lehman and Xi Yang wrote this optional service
 - Includes many heuristics to optimize path, chance of success
 - Allows excluding particular connection points, VLANs
- **Other tools may use different heuristics**
 - Stitcher uses the SCS



- **Advertisement RSpecs**

- This local switch/port connects to
- That remote aggregate switch/port
- Using this VLAN range

- **Request RSpecs**

- I want a circuit from A to B
- Get from A to B using this series of
- Hops: switch/port/suggested VLAN tag

- **Manifest RSpecs**

- VLAN tags assigned for each hop

```
<stitching>
  <path id="mylink">
    <hop id="1">
      <link
id=switch1:port1">
        ...
      <vlan...>3747</vlan...>
    <hop id="2">...
```

Describes connections, paths, and requested or allocated VLANs

* <http://www.geni.net/resources/rspec/ext/stitch/>

```

<link client_id="link-pg-utah1-ig-gpo1">
  <component_manager name="urn:publicid:IDN+utah.geniracks.net+authority+cm"/>
  <interface_ref client_id="pg-utah1:if0"/>
  <component_manager name="urn:publicid:IDN+emulab.net+authority+cm"/>
  <component_manager name="urn:publicid:IDN+ion.internet2.edu+authority+cm"/>
  <component_manager name="urn:publicid:IDN+instageni.emulab.bbn.com+authority+cm"/>
  <interface_ref client_id="ig-gpo1:if0"/>
  <property capacity="20000" dest_id="ig-gpo1:if0" latency="0" packet_loss="0" source_id="pg-utah1:if0"/>
  <property capacity="20000" dest_id="pg-utah1:if0" latency="0" packet_loss="0" source_id="ig-gpo1:if0"/>
</link>
<stitching lastUpdateTime="20130128:17:20:44" xmlns="http://hpn.east.isi.edu/rspec/ext/stitch/0.1/">
  <path id="link-pg-utah1-ig-gpo1">
    <hop id="1">
      <link id="urn:publicid:IDN+emulab.net+interface+procurve-pgeni-atla:3.21">
        <trafficEngineeringMetric>10</trafficEngineeringMetric>
        <capacity>1000000000</capacity>
        <switchingCapabilityDescriptor>
          <switchingcapType>l2sc</switchingcapType>
          <encodingType>ethernet</encodingType>
          <switchingCapabilitySpecificInfo>
            <switchingCapabilitySpecificInfo_L2sc>
              <interfaceMTU>9000</interfaceMTU>
              <vlanRangeAvailability>750-1000</vlanRangeAvailability>
              <suggestedVLANRange>990</suggestedVLANRange>
              <vlanTranslation>>false</vlanTranslation>
            </switchingCapabilitySpecificInfo_L2sc>
          </switchingCapabilitySpecificInfo>
        </switchingCapabilityDescriptor>
      </link>
      <nextHop>2</nextHop>
    </hop>
    <hop id="2">

```

- ◆ Added intermediate AMs on links
- ◆ Capacity specified
- ◆ Stitching extension showing suggested VLANs at the bottom, and the range to pick from

- Your slice may require resources at additional aggregates
 - Regionals and a backbone that connect your aggregates
- SCS figures this out and expands the request
- Stitcher keeps track of all used aggregates
 - Used later to renew and delete at all your aggregates

Intermediate Aggregates: Examples

- **Internet 2:** ION and Internet2 service for dynamic circuits
 - OSCARS to do dynamic circuits and VLAN translation
 - SFA based AM translates GENI calls to OSCARS
 - Internet2 operates this aggregate



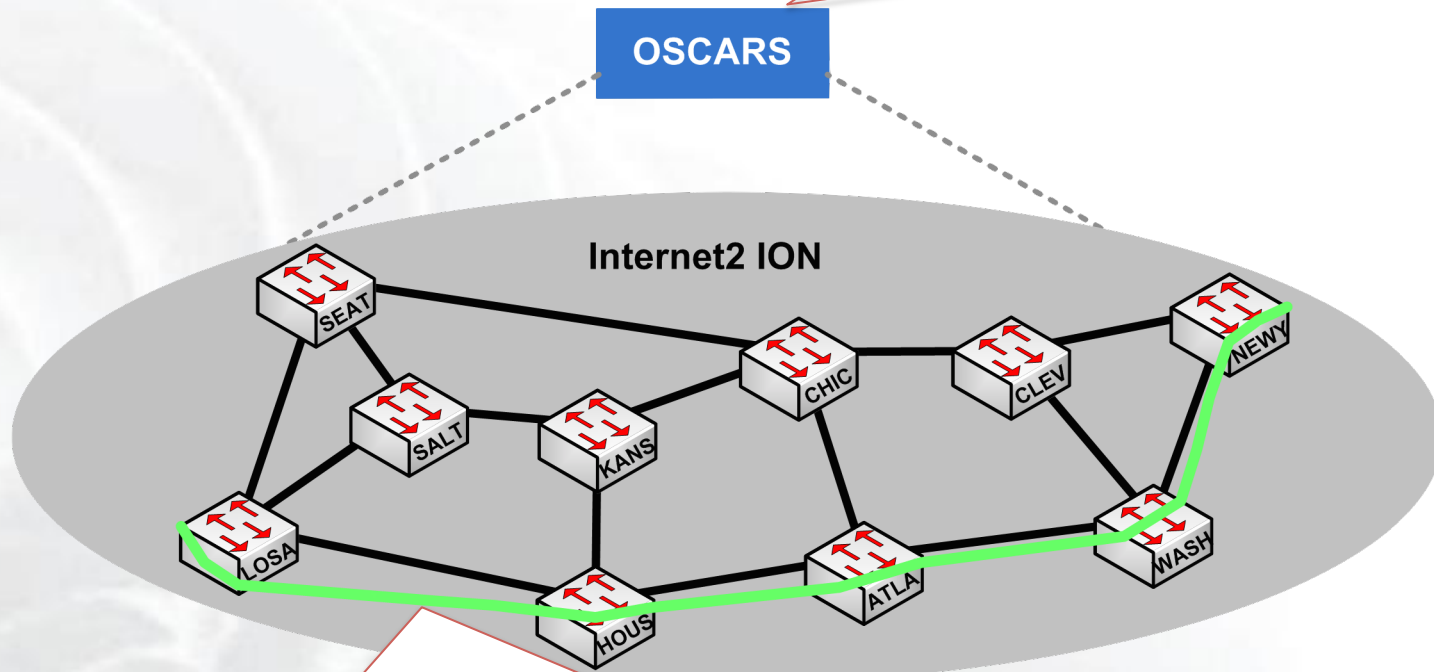
- **MAX Regional**
 - Uses OSCARS like ION
 - Operates a GENI AM



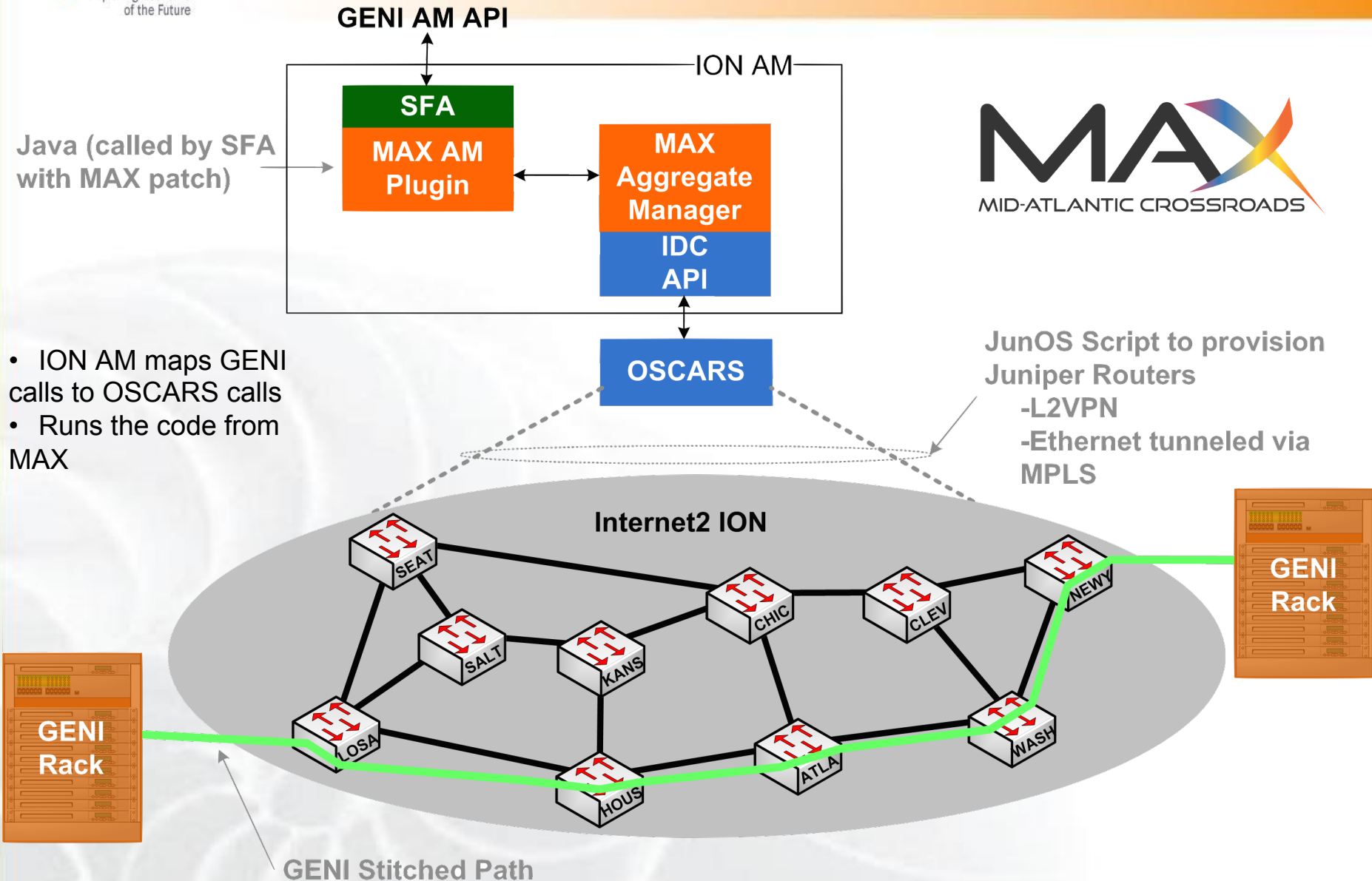
This is powerful!
Connect arbitrary GENI ION endpoints
Enables stitching to non-GENI resources

OSCARS (On-Demand Secure Circuits and Advance Reservation)

- Dynamically provisions circuits on the Internet2 production Juniper routers
- Uses JunOS scripts



- **L2VPN**, i.e. point-to-point Ethernet circuit, i.e. Ethernet tunneled **via MPLS**
- Bandwidth guaranteed circuits



- ION AM maps GENI calls to OSCARS calls
- Runs the code from MAX

- Upcoming replacement for ION
 - Different dataplane technology: OpenFlow VLANs vs MPLS L2VPN
- Supports multipoint circuits
 - We hope to support that in GENI at some point
- AL2S Aggregate Manager
 - Will control OESS as the ION AM controls ION
 - Developed by Internet2 based on FOAM and OESS
- Can use GENI AM API to stitch between ION and AL2S

```
14:43:39 INFO      : All aggregates are complete.
14:43:39 INFO      : Your resources expire at 3
different times at different AMs. The first expiration
is 2014-...
14:43:39 INFO      : Saved combined reservation RSpec at
4 AMs to file '...-manifest-rspec-stitching-
combined.xml'
Success: Reserved resources in slice ahtest at 4
Aggregates (including 2 intermediate aggregate(s) not
in the original request), creating 1 stitched link(s).
```

**If you've gotten that,
raise your hand**



Run:

```
$ stitcher.py createsliver <slice> <rspec>
```

You should see something like:

```
14:37:48 INFO      : Configured logging from file ...stitcher_logging.conf
14:37:48 INFO      : Reading slice ... credential...
14:38:04 INFO      : Slice ..+slice+..expires on 2014-.. UTC
14:38:04 INFO      : Calling SCS...
14:38:35 INFO      : Multi-AM reservation will include resources from these
aggregates:
14:38:35 INFO      :           <Aggregate ...>
14:38:35 INFO      :           <Aggregate ...>
14:38:35 INFO      :           <Aggregate ...>
14:38:35 INFO      :           <Aggregate ion>
14:38:35 INFO      : Stitcher doing createsliver at <Aggregate ...>...
```

When you've gotten that, raise your hand

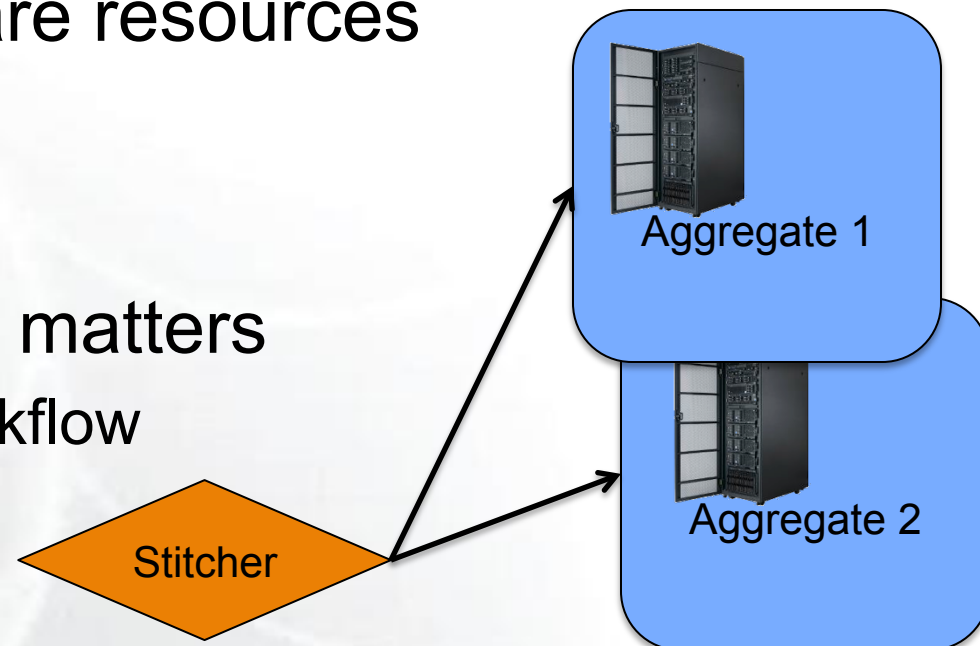
Stitcher Output: createsliver

```
14:37:48 INFO      : Configured logging from file ...
14:37:48 INFO      : Reading slice ... credential...
14:38:04 INFO      : Slice ..+slice+..expires on 2014-.. UTC
14:38:04 INFO      : Calling SCS...
14:38:35 INFO      : Multi-AM reservation will include
resources from these aggregates:
14:38:35 INFO      : <Aggregate ...>
14:38:35 INFO      : <Aggregate ...>
14:38:35 INFO      : <Aggregate ...>
14:38:35 INFO      : <Aggregate ion>
14:38:35 INFO      : Stitcher doing createsliver at <Aggregate
...>...
14:39:18 INFO      : ... Allocation at <Aggregate ...> complete.
...
14:41:11 INFO      : Stitcher doing createsliver at <Aggregate
ion>...
14:42:25 INFO      : DCN AM <Aggregate ion>: must wait for
status ready....14:42:25 INFO      : Pausing 30 seconds to let
circuit become ready...
```

Continuing to look behind the scenes...

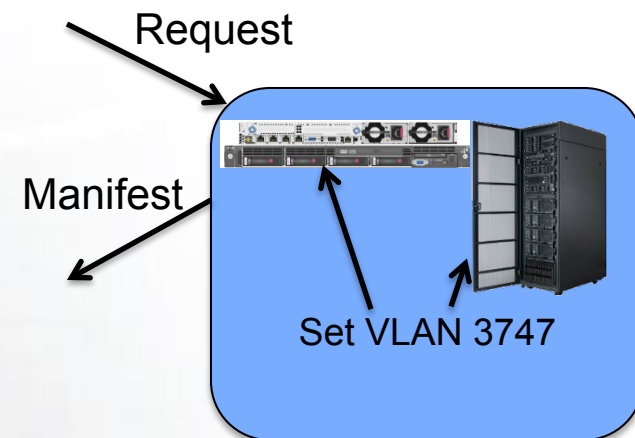
Stitcher Reserves the Resources

- Resources are reserved through the GENI AM API 'CreateSliver' calls
 - Stitcher uses Omni to make the AM API calls
- Circuits and VLANs are resources like any other
- Order of reservations matters
 - Use the computed workflow



Aggregate Handles Stitch Requests

- Checks resource availability
 - VLANs, nodes, bandwidth?
- Allocates resources
 - Configure nodes
 - Configure the switch to connect the VLAN to the node
- Returns manifest
 - VLAN allocated is reported in stitching extension



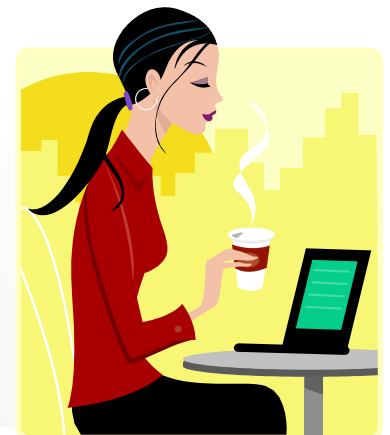
- **Stitcher handles errors**
 - VLAN is in use?
 - Try another VLAN in the suggested range
 - Wait 30 sec - 10 min to let the aggregate free up resources
 - No VLANs available here?
 - Try a different path
 - Something else, like no node available?
 - Tell the user
- **Stitcher reads VLAN out of manifest**
 - VLAN is inserted into request at next aggregate
 - e.g. GPO IG picks tag 3747, so request to ION uses 3747

- Dynamic circuits may fail later when configuring the switches
 - OSCARS, ION

- **Common ION errors:**

```
Sliver status for circuit ... was (still): failed  
sliverstatus: X is (still) ... at <Aggregate ...>. Had  
error message: ...  
sliverstatus: X is (still) ... at <Aggregate ...>.  
Delete and retry.
```

- **Stitcher will retry**
 - Many ION errors are transient



- You will see error messages. *Don't Panic!*
- Stitcher handles many errors itself
- For an explanation of common error messages, see the appendix or wiki

<http://trac.gpolab.bbn.com/gcf/wiki/Stitcher>

```
16:27:47 ERROR      : {'output': "vlan tag 1712 for 'stitched4' not available", 'code': {'protogeni_error_log'...
```

```
16:27:47 INFO       : A requested VLAN was unavailable doing createsliver XX at <Aggregate xxx-ig>: AMAPIError: Error from Aggregate: code 1. protogeni AM code: 1: vlan tag 1712 for 'stitched4' not available.
```

```
16:27:47 INFO       : Will put <Aggregate xxx-ig> back in the pool to allocate. Got VLAN was unavailable. Retry <Aggregate xxx-ig> 2nd time with <Hop u'urn:publicid:IDN+instageni...' on path u'stitched4'> new suggested 1713 (not 1712)
```

```
16:27:47 INFO       : Pausing for 30 seconds for Aggregates to free up resources...
```

You have your circuit connecting your nodes!



Stitcher

```
<link client_id="mylink">
  <component_manager...
  <component_manager...
  ...
  <stitching>
    <path id="mylink">
      <hop id="1">
        <link id="switch1:port1"
        ...
        <vlan...>3747</vlan...>
```

7. Manifest

- What do you have?
 - Details aren't usually interesting
 - Which VLAN tag, etc
- Stitcher combines the manifests from all your aggregates
 - You may have reservations at places you didn't expect, like ION

```
<!--  
Aggregate Details  
{  
  "url": "https://www.emulab.net:12369/protogeni/xmlrpc/am",  
  "hops_info": [  
    {  
      "path_id": "link-pg-utah-ig-utah",  
      "path_global_id": null,  
      "urn": "urn:publicid:IDN+emulab.net+interface+procurveA:3.19",  
      "vlan_tag": "992",  
      "id": "1"  
    }  
  ],  
  "urn": "urn:publicid:IDN+emulab.net+authority+cm",  
  "user_requested": true,  
  "api_version": 2  
}  
...  
<stitching lastUpdateTime="20140310:17:37:40" xmlns="http://hp1.east.isi.edu/rspec/ext/stitch/0.1/">  
  <path id="link-pg-utah-ig-utah">  
    <hop id="1">  
      <link id="urn:publicid:IDN+emulab.net+interface+procurveA:3.19">  
        ...  
        <vlanRangeAvailability>992</vlanRangeAvailability>  
        <suggestedVLANRange>992</suggestedVLANRange>
```

- ◆ Summary at the top, including circuits and VLANs
- ◆ Stitching extension showing specific VLANs at the bottom
- ◆ Plus all the usual bits.


```
14:43:39 INFO      : All aggregates are complete.
14:43:39 INFO      : Your resources expire at 3
different times at different AMs. The first expiration
is 2014-...
14:43:39 INFO      : Saved combined reservation RSpec at
4 AMs to file '...-manifest-rspec-stitching-
combined.xml'
Success: Reserved resources in slice ahtest at 4
Aggregates (including 2 intermediate aggregate(s) not
in the original request), creating 1 stitched link(s).
```

**If you've gotten that,
raise your hand**



Run:

```
$ stitcher.py createsliver <slice> <rspec>
```

You should see something like:

```
14:37:48 INFO      : Configured logging from file ...stitcher_logging.conf
14:37:48 INFO      : Reading slice ... credential...
14:38:04 INFO      : Slice ..+slice+..expires on 2014-.. UTC
14:38:04 INFO      : Calling SCS...
14:38:35 INFO      : Multi-AM reservation will include resources from these
aggregates:
14:38:35 INFO      :           <Aggregate ...>
14:38:35 INFO      :           <Aggregate ...>
14:38:35 INFO      :           <Aggregate ...>
14:38:35 INFO      :           <Aggregate ion>
14:38:35 INFO      : Stitcher doing createsliver at <Aggregate ...>...
```

When you've gotten that, raise your hand

- ◆ Try your experiment within a single aggregate first
 - ◆ Debug and make sure your experiment works
- ◆ Ask questions on geni-users@googlegroups.com
 - ◆ Tell us when you can't run your experiment, and what went wrong if you know
 - ◆ **Include `stitcher.log`** (Includes full stitcher output)
 - ◆ We need to know what will help you most as we grow GENI
- ◆ Use Flack to create an RSpec and use stitcher to reserve resources
- ◆ Your results are saved in your current directory
 - ◆ Change this with `-fileDir`

- ◆ You can specify link capacity on each circuit.
 - ◆ In kbps in InstaGENI and ION, but in bps in ExoGENI
- ◆ Use the `-defaultCapacity` option to change the default requested capacity of 20,000 kbps
- ◆ Most links are 1Gbps. So if you think you need more than 500,000 kbps, then you should:
 - a) Expect failure due to insufficient capacity often, and
 - b) Work with help@geni.net to file a ticket with GMOC for a disruptive experiment.

- Stitcher remembers what aggregates it used for each slice in a local file
 - Named like `ch-geni-net-GEC20-XX-amlist.txt`
- Renew your reservations!
 - ION in particular defaults to only 24 hours
- Use stitcher to renew or delete your reservation
 - stitcher remembers **all aggregates** including those **intermediate aggregates**:

```
$ stitcher renewsliver XXX 20140401
```

Stitching is in early stages of deployment

- Supported at limited sites
- Limited number of available VLANs
- Success is correlated with the complexity of the topology
- We are working hard on making this robust and widely available. In the meantime ...

	Control Plane	Data Plane
Static	Control Interface	Preconfigured VLANs
Dynamic	GRE Tunnels	GENI Stitching EG Stitching

Control Plane: Use Control Interface

Use the Internet connectivity of the control interface to exchange data between your nodes

- Limited shared BW up to 100Mbps
- Competes with Internet traffic
- Competes with your control traffic
- Might run into firewall issues
- Non-repeatable experiment
- In IG need to ask for this
- **Use the PlanetLab AM**
 - More geographic diversity



PlanetLab nodes

Use this option if you have light traffic and you want to run an “in-the-wild” experiment

Tunnel experiment traffic over the control interface

- **Encapsulates** network layer protocols inside **virtual point-to-point links** over an IP
- Gives the impression of one-hop
- Same issues as with the control interface (bandwidth, firewalls)
- Uses private IP addresses
- Separate interface that you can control separately from the control interface
- Useful when sites don't have real layer 2, and you don't need specific performance

Use this option if you have light traffic and want control over a separate data interface

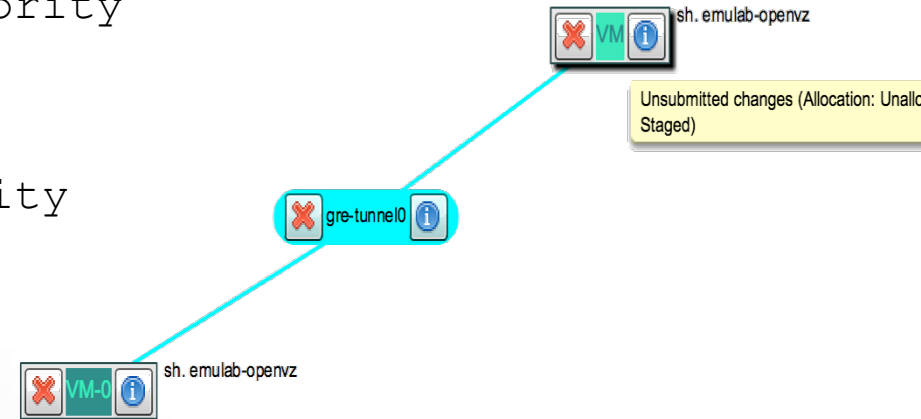
- Automated for experiments in PG/IG
- Two types:
 - GRE: IP inside
 - Supported on OpenVZ containers, bare metal
 - EGRE: Ethernet inside
 - Supported on Xen, transparent to guest OS
 - The two cannot be connected
 - Can't connect OpenVZ with Xen!
- Separate dataplane link type in RSpecs

Making GRE Tunnels

```

<link client_id="link0">
  <component_manager
name="urn:publicid:IDN
+instageni.northwestern.edu+authority
+cm"/>
  <component_manager
name="urn:publicid:IDN
+instageni.gpolab.bbn.com+authority
+cm"/>
  <interface_ref
client_id="VM-0:if0"/>
  <interface_ref
client_id="VM:if0"/>
  <property source_id="VM-0:if0"
dest_id="VM:if0"/>
  <property source_id="VM:if0"
dest_id="VM-0:if0"/>
  <link_type name="gre-tunnel"/>
</link>

```



Use Flack or Omni to bring up topologies using GRE tunnels

```
14:43:39 INFO      : All aggregates are complete.
14:43:39 INFO      : Your resources expire at 3
different times at different AMs. The first expiration
is 2014-...
14:43:39 INFO      : Saved combined reservation RSpec at
4 AMs to file '...-manifest-rspec-stitching-
combined.xml'
Success: Reserved resources in slice ahtest at 4
Aggregates (including 2 intermediate aggregate(s) not
in the original request), creating 1 stitched link(s).
```

**If you've gotten that,
raise your hand**



Run:

```
$ stitcher.py createsliver <slice> <rspec>
```

You should see something like:

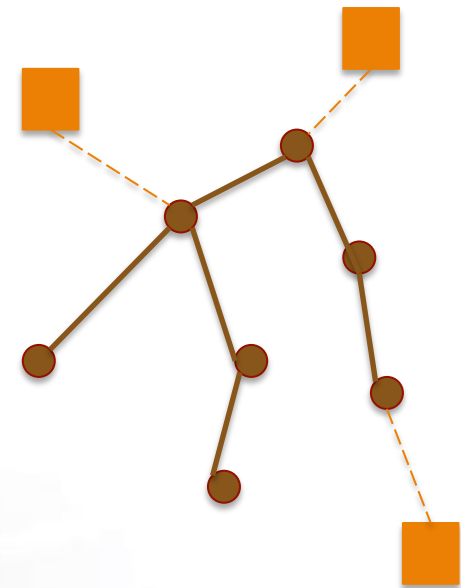
```
14:37:48 INFO      : Configured logging from file ...stitcher_logging.conf
14:37:48 INFO      : Reading slice ... credential...
14:38:04 INFO      : Slice ..+slice+..expires on 2014-.. UTC
14:38:04 INFO      : Calling SCS...
14:38:35 INFO      : Multi-AM reservation will include resources from these
aggregates:
14:38:35 INFO      :           <Aggregate ...>
14:38:35 INFO      :           <Aggregate ...>
14:38:35 INFO      :           <Aggregate ...>
14:38:35 INFO      :           <Aggregate ion>
14:38:35 INFO      : Stitcher doing createsliver at <Aggregate ...>...
```

When you've gotten that, raise your hand

Data plane: Pre-Configured VLANs

Preconfigured Layer 2 topologies that you can attach your experiment to

- Static topologies
- Shared bandwidth
- Layer 2
- OpenFlow VLANs (mesoscale)
- Non-OpenFlow VLANs (FIA VLANs)
- Higher repeatability
- Coordinate with help@geni.net



Use this option for higher bandwidth experiments using the dataplane

Using Pre-Configured (shared) VLANs

```

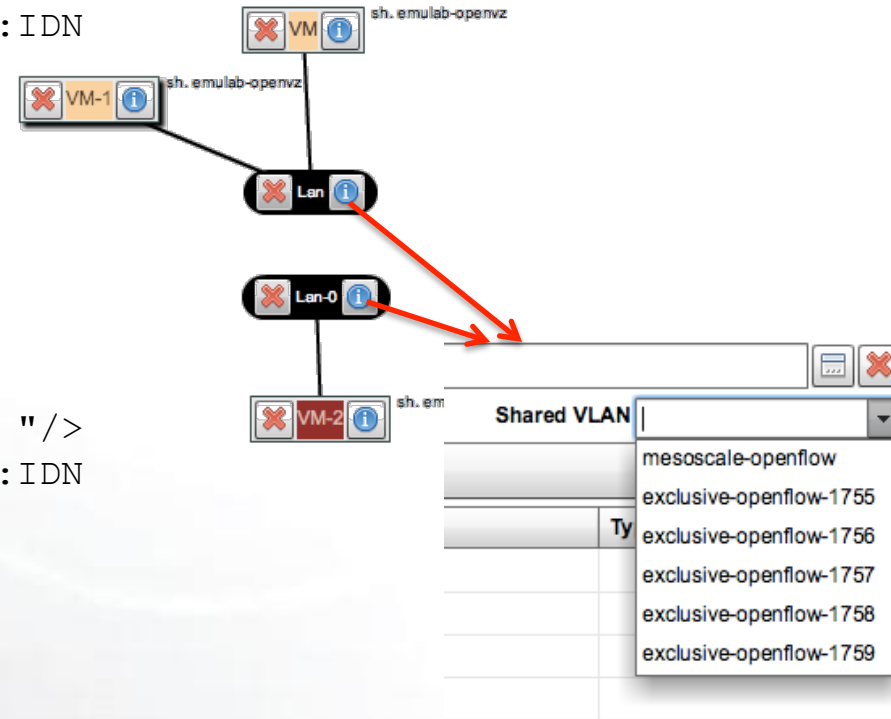
<link client_id="link0">
  <link_type name="lan"/>
  <vlan:link_shared_vlan name="fia-xia" "/>
  <component_manager name="urn:publicid:IDN
+instageni.gpolab.bbn.com+authority+cm"/>
  <interface_ref client_id="VM:if0"/>
  <interface_ref client_id="VM-1:if0"/>
  <interface_ref client_id="VM-2:if0"/>
</link>

```

```

<link client_id="link0">
  <link_type name="lan"/>
  <vlan:link_shared_vlan name="fia-xia" "/>
  <component_manager name="urn:publicid:IDN
+utah.instageni.net+authority+cm"/>
  <interface_ref client_id="VM:if0"/>
  <interface_ref client_id="VM-1:if0"/>
  <interface_ref client_id="VM-2:if0"/>
</link>

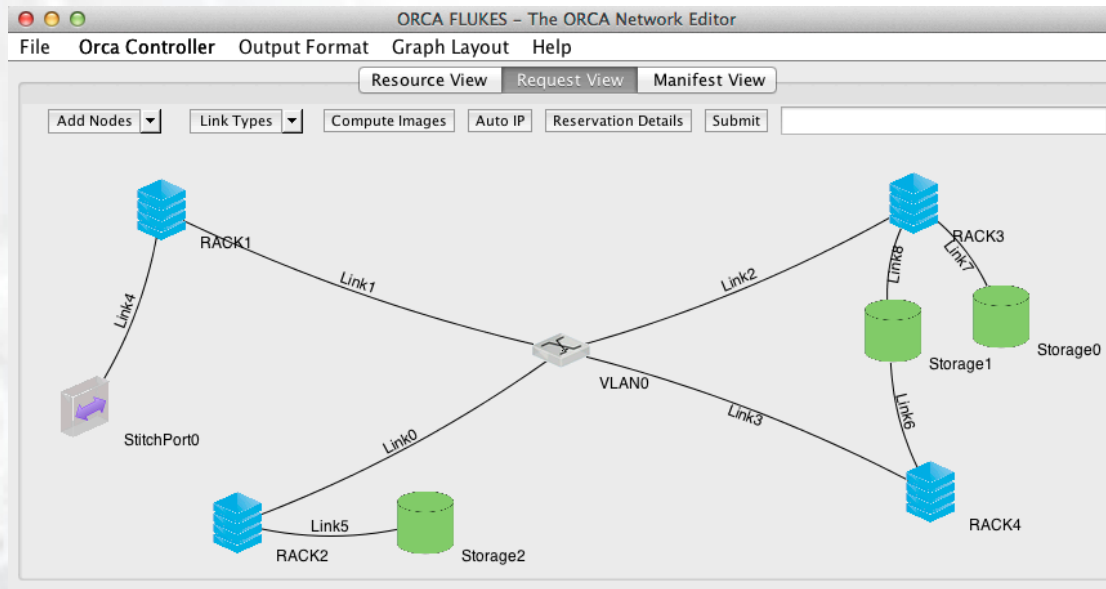
```



Use Flack, Omni or the Portal

Data Plane: ExoGENI stitching

- Works across multiple network aggregates:
 - AL2S, ION, ESnet
- Integrated with ExoGENI operations as part of topology embedding
 - Multipoint VLAN support – create large Layer 2 domains
 - Storage stitching support – create large slivers of iSCSI storage
 - Stitch ports – connect slices to the outside world



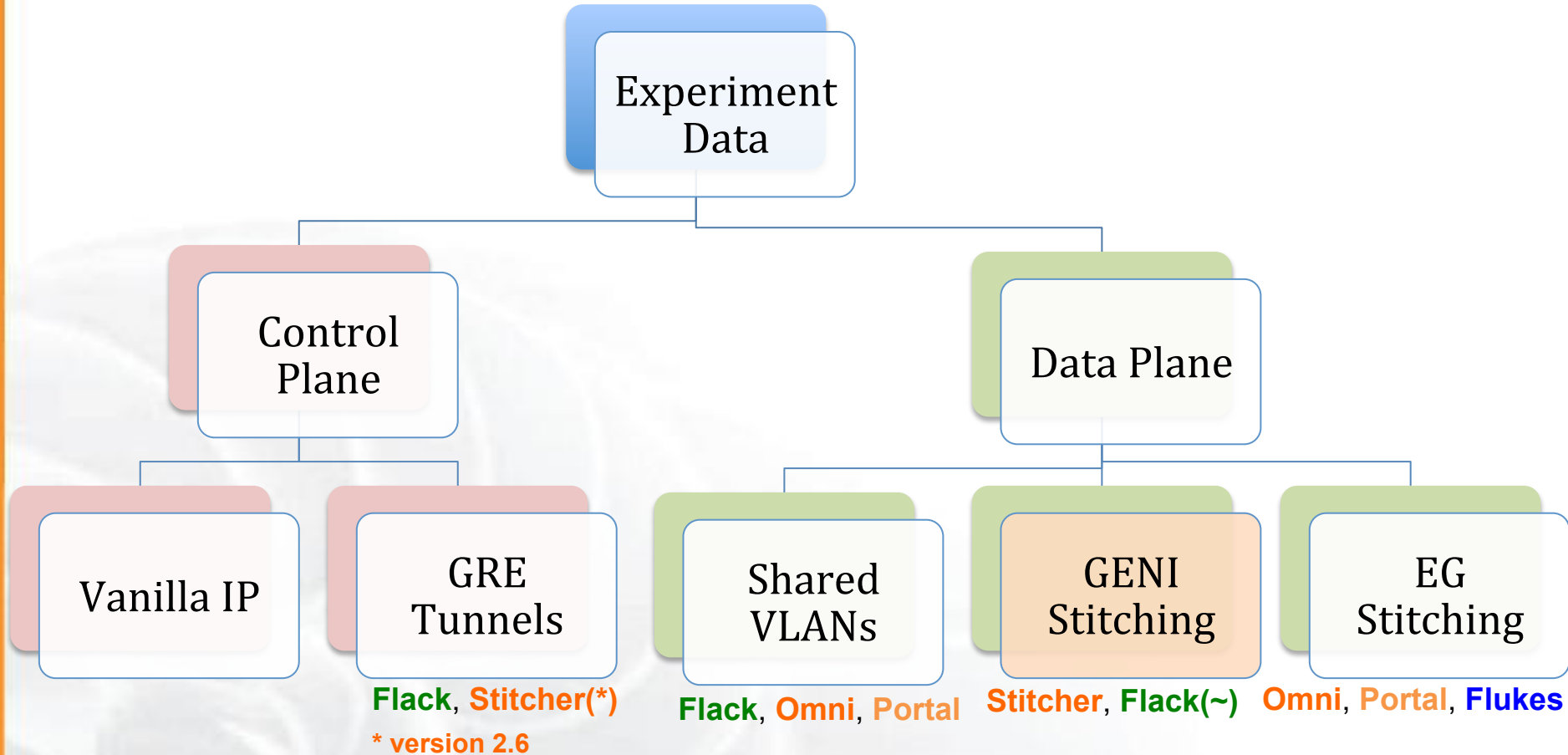
ExoGENI Stitching: Ways to use it

- Create a topology with two sites and submit it to ExoSM
- Works only between EG racks
- Works with:
 - GENI Tools: **omni**, **GENI Portal**
 - Limited capabilities
 - ExoGENI Tools: **Flukes**
 - + Expose full range of capabilities
 - + Use regular GENI credentials
- More information:
 - + <http://tinyurl.com/l5qsbwy>
 - + <http://www.exogeni.net>

```
<link client_id="link0">  
  <link_type name="lan"/>  
  <component_manager name="urn:publicid:IDN  
+exogeni.net:rencivmsite+authority+am"/>  
  <component_manager name="urn:publicid:IDN  
+exogeni.net:bbnvmsite+authority+am"/>  
  <interface_ref client_id="VM-0:if0"/>  
  <interface_ref client_id="VM:if0"/>  
  <property source_id="VM-0:if0" dest_id="VM:if0"/>  
  <property source_id="VM:if0" dest_id="VM-0:if0"/>  
</link>
```

Submit once to ExoSM

Use this option if you're using only ExoGENI nodes



Different experiments have different needs; choose based on your experiment!

14:43:39 INFO : All aggregates are complete.

14:43:39 INFO : Your resources expire at 3 different times at different AMs. The first expiration is 2014-...

14:43:39 INFO : Saved combined reservation RSpec at 4 AMs to file '...-manifest-rspec-stitching-combined.xml'

Success: Reserved resources in slice ahtest at 4 Aggregates (including 2 intermediate aggregate(s) not in the original request), creating 1 stitched link(s).

**If you've gotten that,
raise your hand**

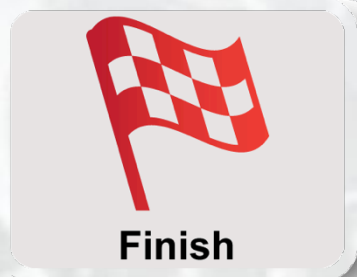




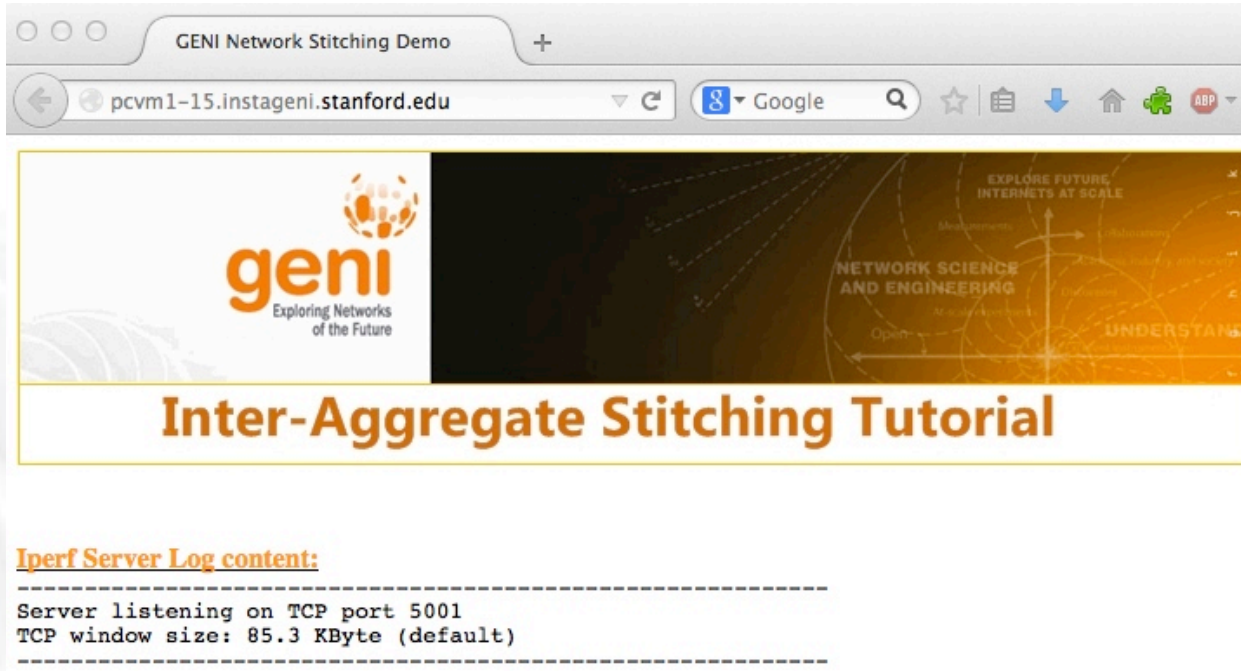
- **Part I: Design/Setup**
 - Design / Tutorial Logistics
 - Craft your RSpecs
 - Obtain Resources



- **Part II: Execute**
 - **Execute Experiment**
 - **Collect Measurements**



- **Part III: Finish**
 - Teardown Experiment



GENI Network Stitching Demo

pcvm1-15.instageni.stanford.edu

Google

geni
Exploring Networks
of the Future

Inter-Aggregate Stitching Tutorial

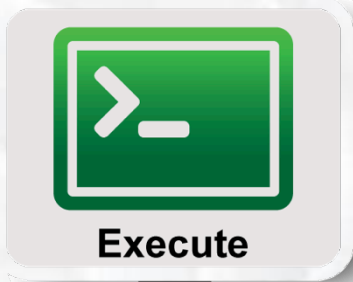
Iperf Server Log content:

Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)

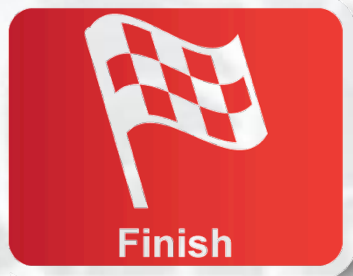
Check out the difference between
the local and the remote paths!



- **Part I: Design/Setup**
 - Design / Tutorial Logistics
 - Craft your RSpecs
 - Obtain Resources



- **Part II: Execute**
 - Execute Experiment
 - Collect Measurements



- **Part III: Finish**
 - Teardown Experiment

Delete your resources

project resource
aggregate experimenter



When your experiment is done, you should always release your resources:

```
$ stitcher deletesliver <slice>
```

More Complex Topologies

- Stitcher supports both stitched links and GRE tunnels
- Multiple circuits are possible: triangles, etc



- GENI provides multiple ways to connect your resources
- Stitcher lets you use GENI stitching to get private VLANs between your compute resources
- This is a powerful, complex capability
 - Understand what can go wrong
 - **Always test your experiment in a single rack!**
 - Report bugs and ask questions!

Stitching provides custom Wide Area Layer 2 topologies in GENI

- <http://trac.gpolab.bbn.com/gcf/wiki/Stitcher>
- **Stitchable Aggregates**
<http://groups.geni.net/geni/wiki/GeniNetworkStitchingSites>
- **Stitching Sample RSpecs**
<http://groups.geni.net/geni/browser/trunk/stitch-examples>
- geni-users@googlegroups.com

Stitcher Detailed References

- ◆ **Stitching and GRE tunnels:**
 - ◆ Stitcher can reserve your GRE tunnels too.
 - ◆ You can even combine GRE tunnels and stitched links
- ◆ **Stitching to fixed endpoints:**
 - ◆ A switch/port that happens to connect to other things but not an explicit node
 - ◆ Use the `--fixedEndpoint` option to be sure aggregates can handle this.
- ◆ **Stitching to ExoGENI aggregates**
 - ◆ Note that in ExoGENI, capacity is in **bps**
 - ◆ ExoGENI reservations can come from the specific rack, or from the ExoSM's allocation of resources at that rack. You can control in stitcher whether you use the local racks or the ExoSM, by using the `-useExoSM` or `-noExoSM` options.

- ◆ Usually there is only one path option for 2 aggregates
- ◆ Rarely (i.e. Utah-ProtoGENI) there are alternatives
- ◆ Stitcher provides 2 options you can use to control the path
- ◆ `--excludeHop <hop URN>`
 - ◆ **E.G.** `--excludeHop urn:publicid:IDN+ion.internet2.edu+interface+rtr.atla:ge-10/3/2:protogeni`
 - ◆ Exclude the given hop from any path in your request
 - ◆ Find the hop URN in the Ad RSpec for the aggregate, or in the manifest for a path you want to modify.
- ◆ **Alternative form** **E.G.** `--excludeHop urn:publicid:IDN+ion.internet2.edu+interface+rtr.atla:ge-10/3/2:protogeni=999`
 - ◆ Exclude the given VLAN tags from the set we pick from
- ◆ `--includeHop <hop URN>`
 - ◆ Force inclusion of this hop in any path in your request

- Fatal errors – something is wrong with your request

StitchingServiceFailedError: Error from Stitching Service: code 3: MxTCE
ComputeWorker return error message

```
'Action_ProcessRequestTopology_MP2P::Finish() Cannot find the set of paths  
for the RequestTopology. '
```

- Errors like this mean there is no GENI layer 2 path possible between your specified endpoints. Did you specify an `excludehop` or `includehop` you shouldn't have? Or include an aggregate that does not support stitching? Alternatively, it may mean that `stitcher` tried all available VLAN tags for one of your aggregates, and got a stitching failure on each - probably because all tags were not available.

Reservation request impossible at <Aggregate ...>

- Something about your request cannot be satisfied. The rest of the message may say more.

Node ... is unbound in request

- One of the nodes in your request did not specify an aggregate at which to reserve the resources. All nodes must be bound to a specific aggregates (include a `component_manager_id` attribute).

Inconsistent ifacemap

- Your request is impossible. Try the `-fixedEndpoint` option if that is relevant.

Not enough bandwidth to connect some nodes

- You requested a link with more bandwidth than is available. Edit the `capacity` attribute in your RSpec, or try specifying `--defaultCapacity` with a smaller number, or pick a different aggregate, or try again later.

Common Error Messages

Too many VMs requested on physical host

Not enough nodes with fast enough interfaces

- You have asked for more nodes than are available. Use fewer nodes or a different aggregate, or try again later.

*** ERROR: mapper

Could not verify topo

Could not map to resources

- You may have asked for more nodes or bandwidth than are available. Or your request may be malformed. The error message may say more, or you can ask on geni-users@googlegroups.com

Hostname > 63 char

- Try a shorter `client_id` (node name) or slice name

no edge hop

- Your request RSpec likely lists a `component_manager` naming an aggregate which has no interface on the given link. Perhaps a copy-and-paste error?

Duplicate link

- Do you have 2 links with the same `client_id`? Edit your request.

Must delete existing slice/sliver

CreateSliver: Existing record

Rspec error: VM with name ... already exists

- You already have a reservation in this slice at this aggregate. Delete it first.

Malformed keys

- Your SSH keys (from your omni_config usually) are malformed.

Edge domain does not exist

- Your ExoGENI request is malformed in some way

check_image_size error

Incorrect image URL in ImageProxy

- Check your ExoGENI disk image specification

Insufficient numCPUCores

- The ExoGENI AM has no room for your VM. Stitcher will try the ExoSM / the local rack to see if it has room.

Need node id for links

- You likely have a typo in an interface `client_id` in your link.

....: Edge iface mismatch when stitching

- You have listed 2 nodes at the same AM on the same stitched link. Each stitched link should be between 2 interfaces on 2 different nodes/AMs.

RSPEC requires AM ... which is not in workflow and URL is unknown!

- Check your RSPEC does not have a typo in the `component_manager`. You asked for resources at an unknown aggregate.

- Errors in the tool – you may need to report this as a bug

... has request tag XXX that is already in use by ...

- Stitcher made an error and picked a tag that is in use. Report this bug.

SCS gave error: ...

- The Stitching Computation Service had an error. You may need to report it.

- **Transient errors – stitcher can handle these**

Circuit reservation failed at ... (....). Try again from the SCS

- An aggregate reported an error. Stitcher will try a new path from the SCS to see if that solves your problem (it may not).

Could not reserve vlan tags

Error reserving vlan tag for ...

Vlan tag ... not available

Could not find a free vlan tag

Could not reserve a vlan tag for

Error in building the dependency tree, probably not available vlan path

- Some VLAN tag you requested is not available. Stitcher will try to find another and try again.

AddPersonToSite: Invalid argument: No such site

- This is the first time this aggregate has seen your project. Stitcher will retry and the error should go away.

- **After too many transient errors, stitcher gives up:**

Stitching reservation failed X times. Last error: ...

- Stitcher goes to the Stitching Service for a path a limited number of times. After that, it gives up with this error. Typically this means there are not enough VLANs or bandwidth to get to your aggregates.