

---

# **GENI Network Stitching**

## **Overview and Status**

GEC13

March 14, 2012

Tom Lehman (USC/ISI)

Xi Yang (USC/ISI)

# Stitching Architecture Objectives

---

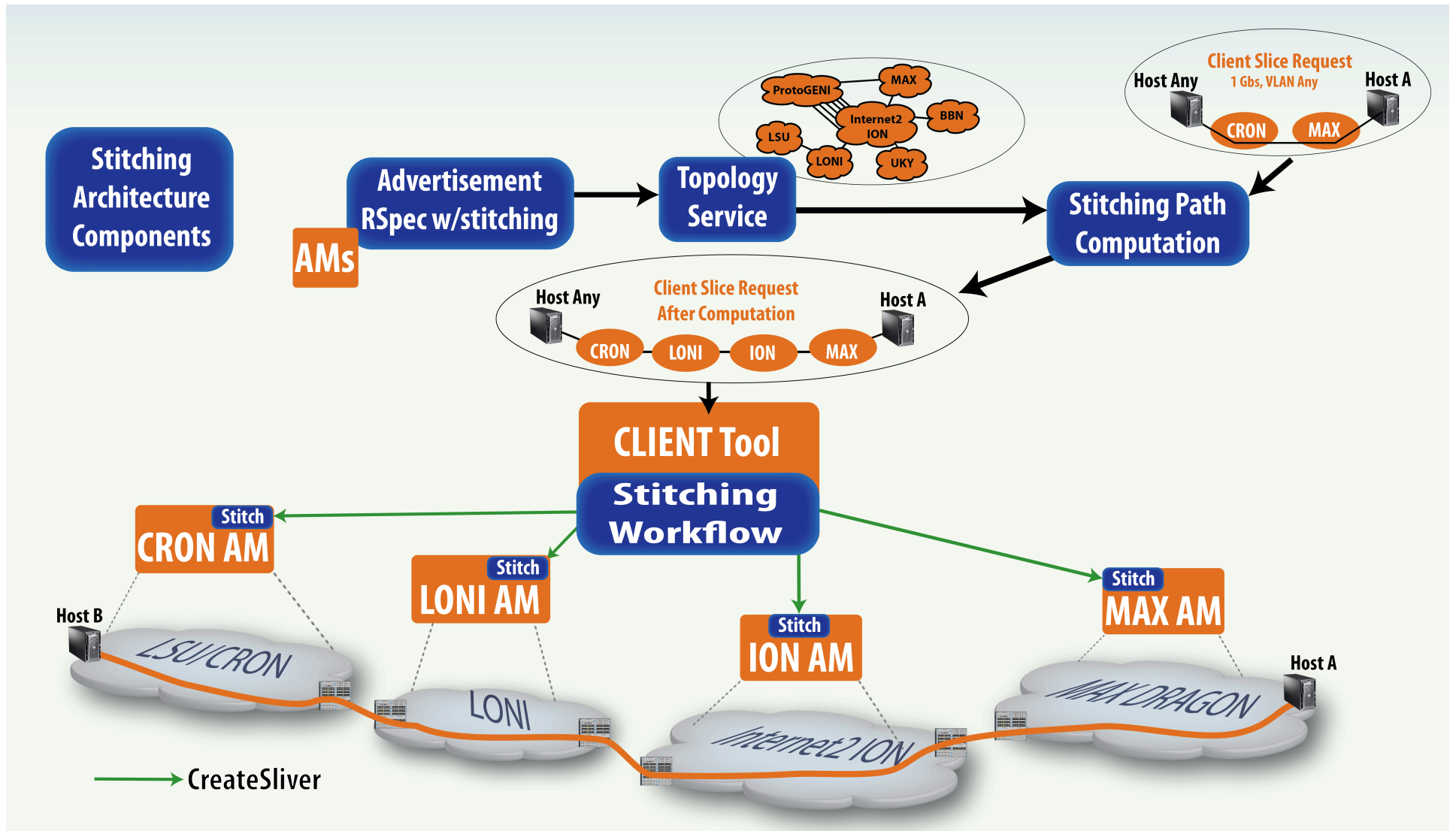
- Inter-Aggregate Network Connections
  - Enable the provision of inter-aggregate network connections as part of multi-aggregate Slice instantiations
  - Stitching based on Layer2 (Ethernet VLANs) to start
- Integrate stitching into standard GENI AM API processes/functions
- Leverage deployed dynamic network capabilities/infrastructure where it helps GENI

# Stitching Architecture Objectives

---

- Leverage currently deployed Dynamic Networking capabilities/infrastructure provides: Ethernet Virtual Private Line (EVPL) Service (multi-domain, schedulable)
- Consists of multiple interoperable networks:
  - Internet2 ION, Internet2 NDDI, ESnet SDN, GEANT AutoBAHN (Europe), RNP (Brazil), JGN2 (Japan), and approximately 15 other regional and campus networks
  - Approximately 30 other deployments underway at regional and campus networks as part of NSF DYNES project

# Stitching Architecture





# Architecture Components

---

- **Stitching Extension**

- mechanism for Advertisement RSpecs to define and describe inter-aggregate touch points
- Advertisement RSpecs contain mostly static data, e.g., 10G physical inter-connect with vlans 2050-2099 reserved for GENI use
- mechanism for Request RSpecs and Manifest RSpecs to define slice specific stitching details

- **Stitching Topology Service**

- collects Stitching RSPECs from all GENI AMs
- Allows the building of larger views (global if desired) of inter-connected GENI AM space

# Architecture Components

---

- **Stitching Computation Service**
  - use Topology Service information to perform slice instantiation specific computations
- **Stitching Workflow Function**
  - uses output from computation service and executes the workflow steps to accomplish network stitching.
  - integrated into user tools which will do multi-aggregate slice instantiations
- **GENI AM API Stitching Extensions**
  - none required initially
  - extensions to support resource negotiation desired for future

# Stitching Status as of GEC 13

---

- A Stitching Architecture Defined
- Stitching Schema Available
  - [hpn.east.isi.edu/rspec/ext/stitch/0.1/stitch-schema.xsd](http://hpn.east.isi.edu/rspec/ext/stitch/0.1/stitch-schema.xsd)
- ProtoGENI AM Advertisement RSpecs include Stitching Extension
- MAX AM instance used as a prototype Internet2 ION AM
  - supports stitching (uses native MAX RSpecs currently, will convert to GENI RSpec v3 format)
- Tree Mode Workflow available OMNI1.5 & 1.6 (separate library)

# Stitching Architecture Information

---

- **Overview:** [geni.dragon.maxgigapop.net](http://geni.dragon.maxgigapop.net) → Network Stitching
- **Architecture:** [geni.maxgigapop.net/twiki/bin/view/GENI/NetworkStitching](http://geni.maxgigapop.net/twiki/bin/view/GENI/NetworkStitching)
- **Stitching Schema Examples:**  
[geni.maxgigapop.net/twiki/bin/view/GENI/NetworkStitchingRpecsandWorkFlow](http://geni.maxgigapop.net/twiki/bin/view/GENI/NetworkStitchingRpecsandWorkFlow)
- **WorkFlow Example:** [geni.maxgigapop.net/twiki/bin/view/GENI/NetworkStitchingWorkFlowExamples](http://geni.maxgigapop.net/twiki/bin/view/GENI/NetworkStitchingWorkFlowExamples)
- **GENI Wiki:** <http://groups.geni.net/geni/wiki/GeniNetworkStitching>

# Stitching Next Steps

---

- Work with ORCA, OpenFlow, PlanetLab AMs to include stitching extension in Advertisement RSpecs (and subsequent Request and Manifest RSpecs)
- Develop a stitching workflow library and/or service
  - clients can integrate library, or
  - use stand alone Stitching Service
- Enhance OMNI client stitching library
  - use above library or service

# Stitching Next Steps

---

- Work with GEMINI project and PerfSONAR communities to design a GENI Topology and Computation Service
- Work with community on GENI AM API extensions which can be used for resource negotiation (based on tickets or something conceptually similar)

---

**The End**  
**Thank-you!**

---

The Following slides are extras



---

# Stitching Architecture and Schema Review

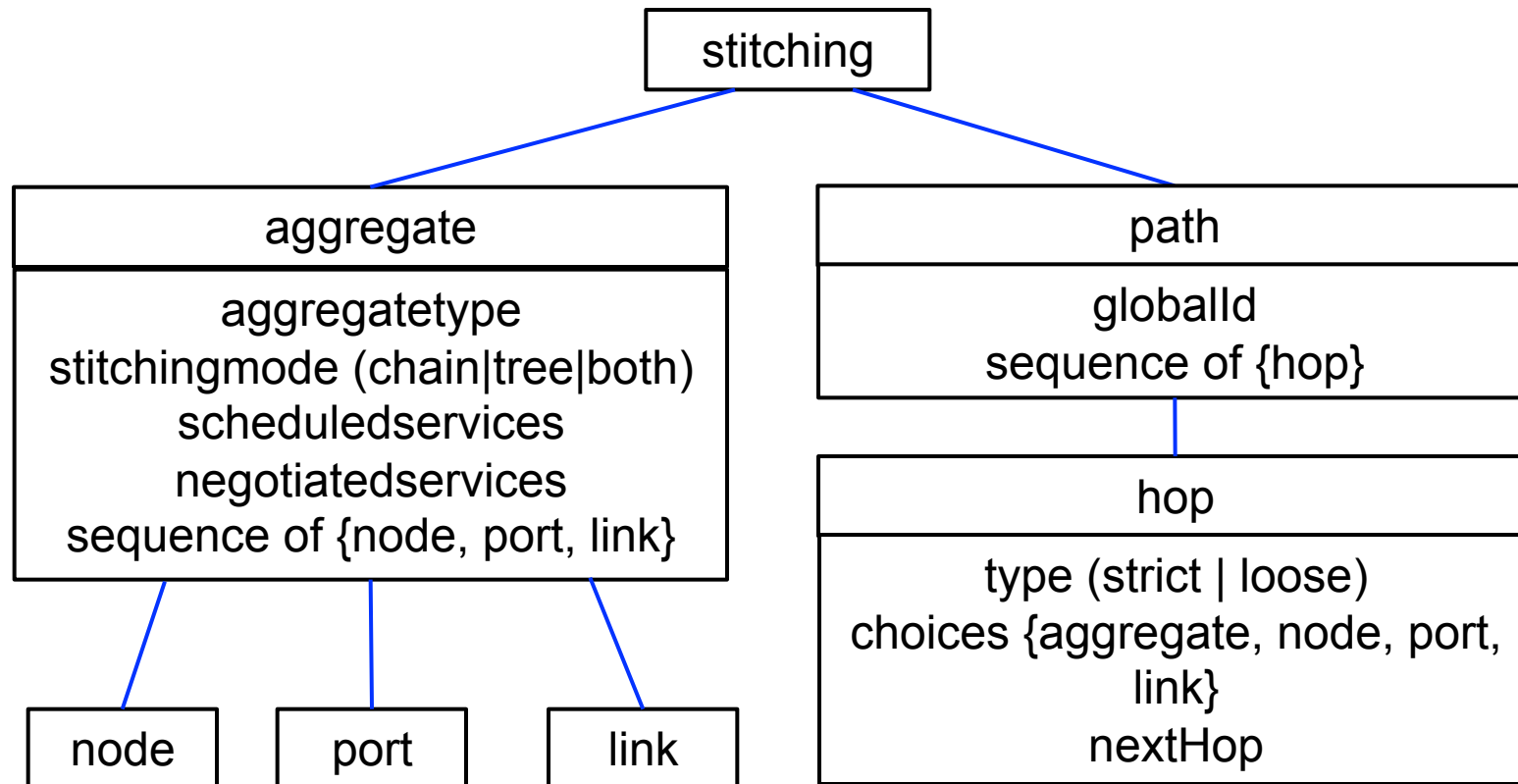
# Stitching Schema

---

- Stitching schema based on Dynamic Circuit Network (ION, IDC, OSCARS) schemas and perfSONAR Topology Service schemas
- Some adjustments for GENI Aggregate specific considerations
- Two goals are
  - maximize interoperation with existing and future dynamic networking deployments
  - provide opportunities to leverage existing perfSONAR systems and software

# Stitching Schema

---



*technology specific details here*

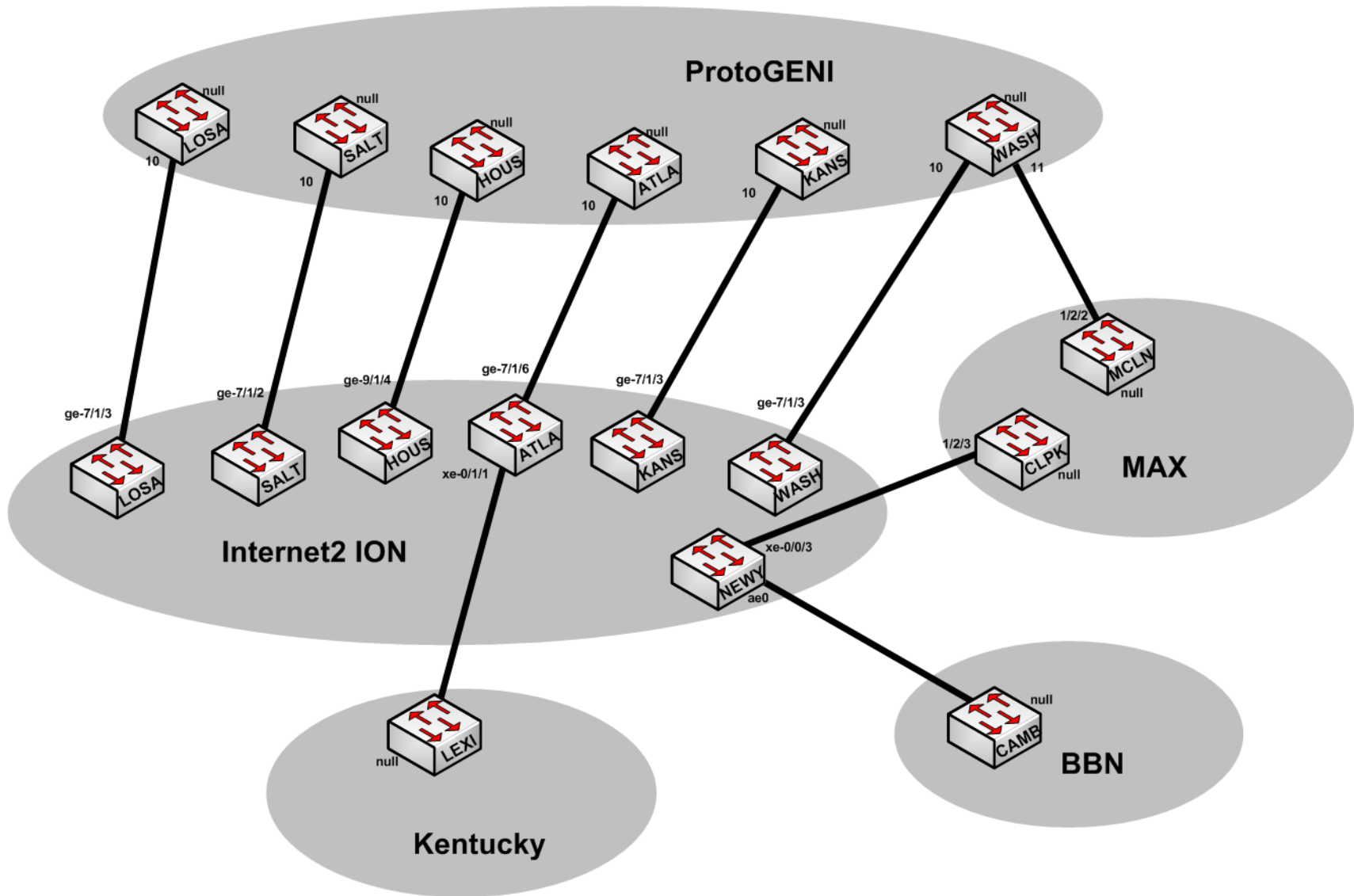
<http://hpn.east.isi.edu/rspec/ext/stitch/0.1/stitch-schema.xsd>

# Stitching Schema

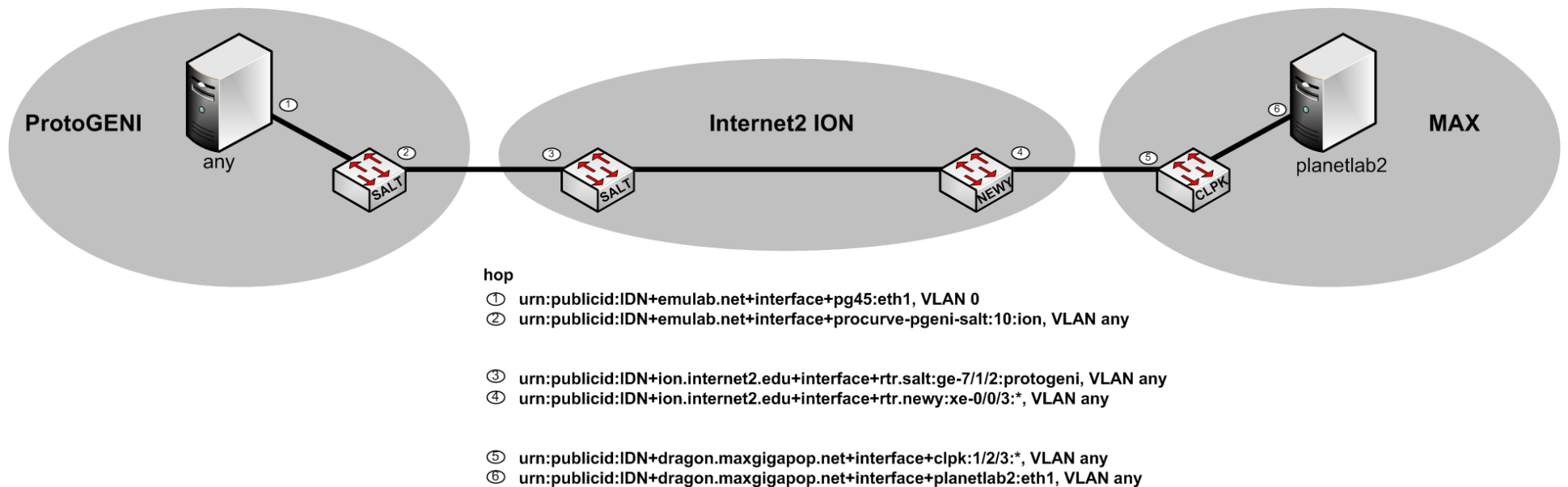
---

- aggregate element
  - used by Aggregate Managers in Advertisement RSpec to describe inter-aggregate touch points
- path element
  - used in Request to indicate stitching preferences and/or requirements
    - may be fed by path computation or simply client preferences
  - used in Manifest RSpec to describe stitching of instantiated slice
- Stitching schema available as an extension to ProtoGENIv2 RSpec Schemas

# Advertisement (aggregate element) – Graph View



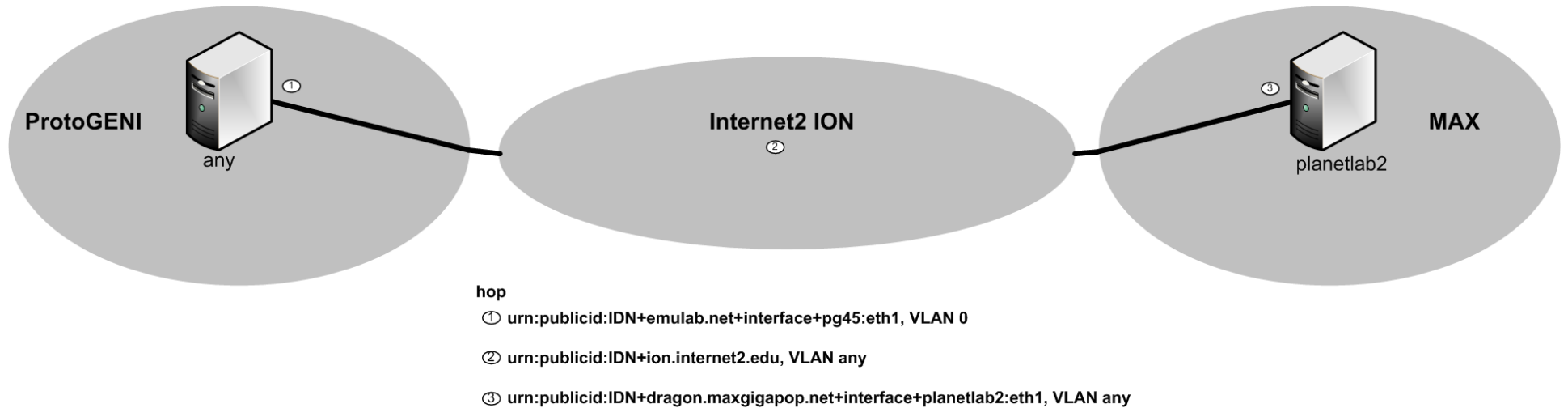
# Request (path element) – Graph View



- Hops can be loose or strict
- Hops can be specified at the aggregate, node, port, or link level
- Manifests will look similar to Requests (annotated with instantiated info)

# Request (path element) – Graph View

---



- Hops can be loose or strict
- Hops can be specified at the aggregate, node, port, or link level
- Manifests will look similar to Requests (annotated with instantiated info)

# Request RSpec Example

<http://geni.maxgigapop.net/twiki/pub/GENI/NetworkStitchingRspecsandWorkFlow/request-from-client-after-computation-v4.xml>

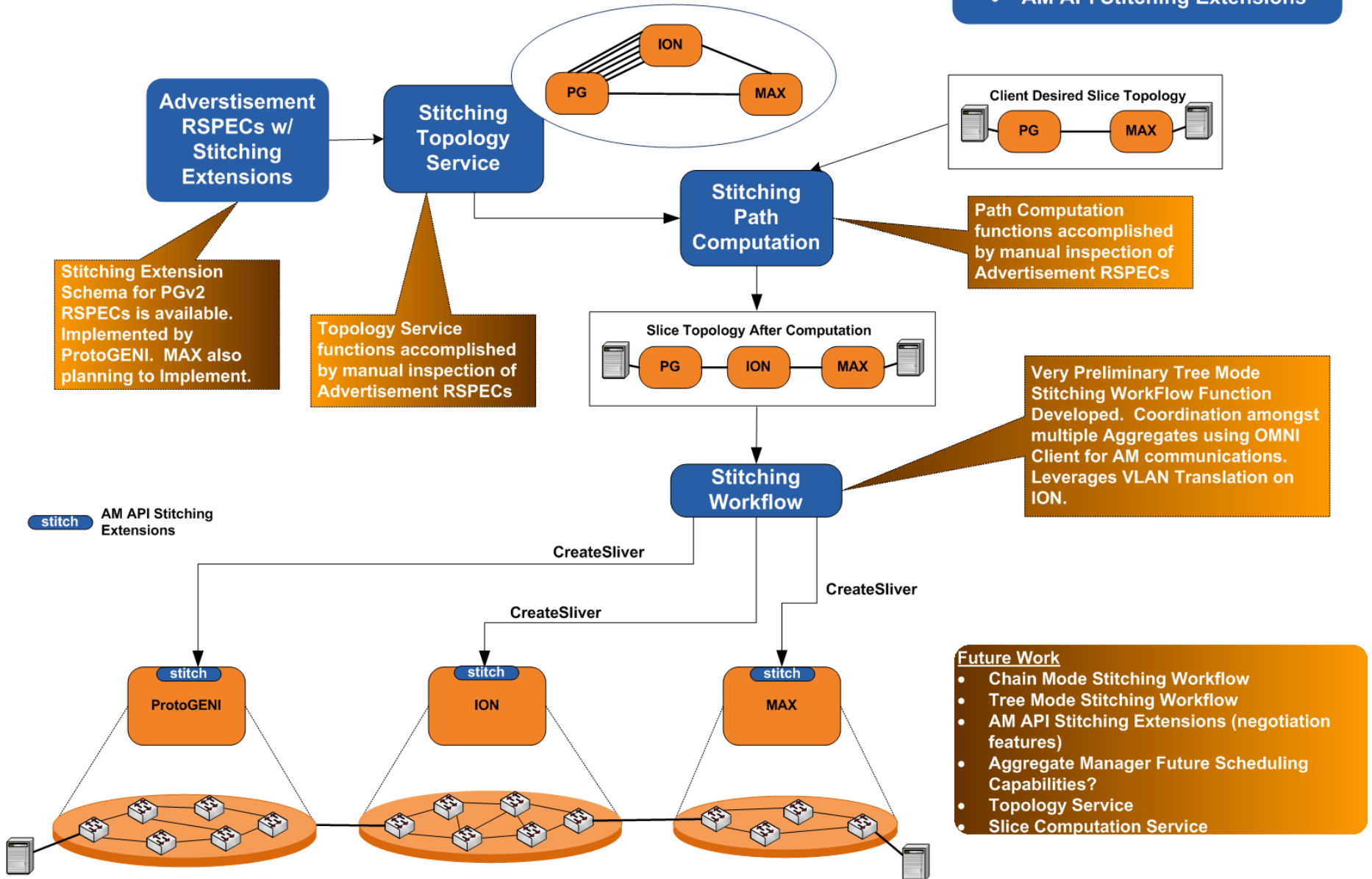
```
- <rspec xsi:schemaLocation="http://www.protogeni.net/resources/rspec/2 http://www.protogeni.net/resources/rspec/2/request.xsd" type="request">
+ <node client_id="left" component_manager_id="urn:publicid:IDN+emulab.net+authority+cm" exclusive="true"></node>
+ <node client_id="right" component_id="urn:publicid:IDN+dragon.maxgigapop.net+node+planetlab2"
  component_manager_id="urn:publicid:IDN+dragon.maxgigapop.net+authority+cm" exclusive="true"></node>
+ <link client_id="mylink"></link>
- <stitching lastUpdateTime="20110220:09:30:21">
  - <path id="mylink">
    - <hop id="1" type="strict">
      + <link id="urn:publicid:IDN+emulab.net+interface+*:*"></link>
      <nextHop>2</nextHop>
    </hop>
    - <hop id="2" type="strict">
      + <link id="urn:publicid:IDN+emulab.net+interface+procurve-pgeni-salt:eth0:ion"></link>
      <nextHop>3</nextHop>
    </hop>
    - <hop id="3" type="strict">
      + <link id="urn:publicid:IDN+ion.internet2.edu+interface+rtr.salt:ge-7/1/2:protogeni"></link>
      <nextHop>4</nextHop>
    </hop>
    - <hop id="4" type="strict">
      + <link id="urn:publicid:IDN+ion.internet2.edu+interface+rtr.newy:xe-0/0/3:*"></link>
      <nextHop>5</nextHop>
    </hop>
    - <hop id="5" type="strict">
      + <link id="urn:publicid:IDN+dragon.maxgigapop.net+interface+clpk:1/2/3:*"></link>
      <nextHop>6</nextHop>
    </hop>
    - <hop id="6" type="strict">
      + <link id="urn:publicid:IDN+dragon.maxgigapop.net+interface+planetlab2:eth1"></link>
      <nextHop>null</nextHop>
    </hop>
  </path>
</stitching>
</rspec>
```



# How Does Current Implementation and Demonstration Capability Relate to Stitching Architecture?

## Stitching Architecture Components

- RSPEC Stitching Extensions
- Stitching Topology Service
- Stitching Path Computation
- Stitching Workflow
- AM API Stching Extensions



---

# Stitching Architecture Next Steps

# AM API Negotiation

---

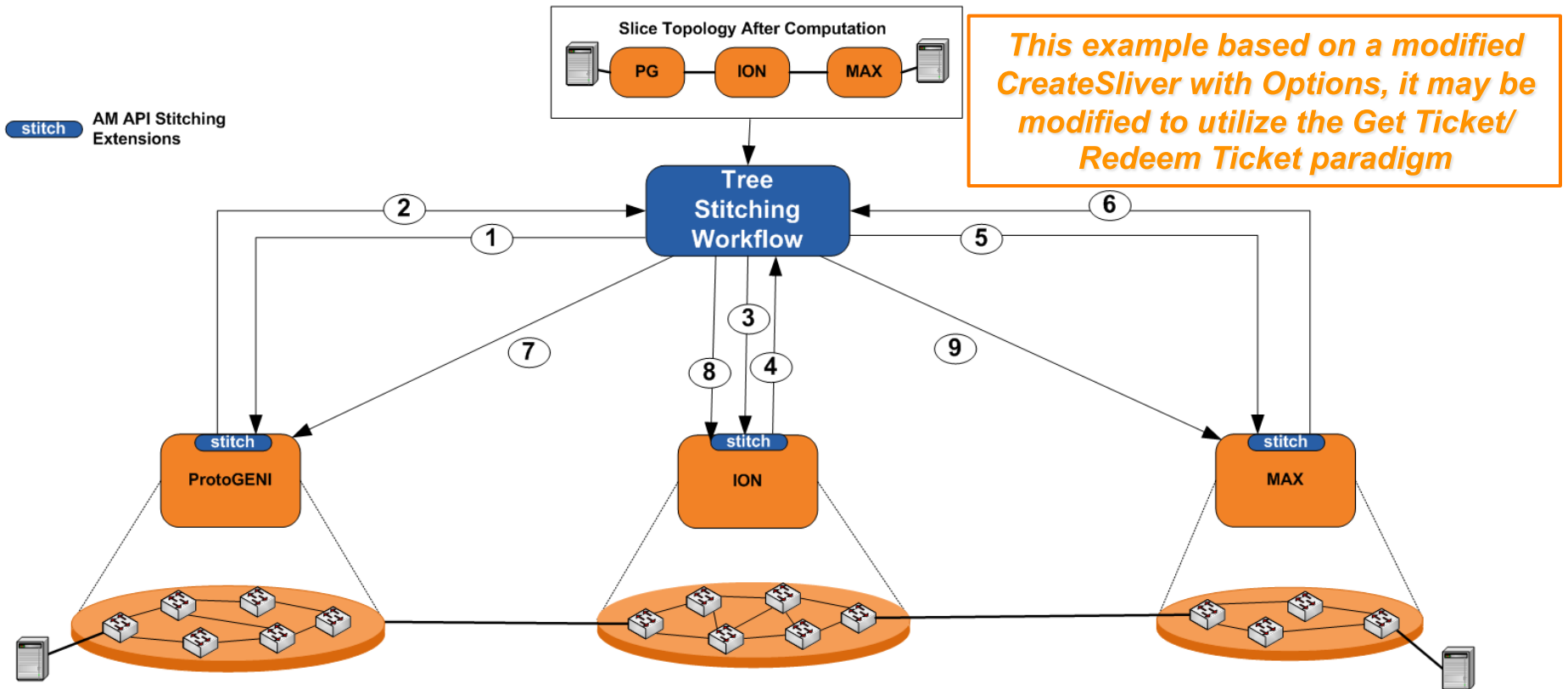
- AM API Extension for Negotiation
  - It is anticipated that the additions (based on tickets or something similar) to AM API will be utilized as the basis for negotiation. This may require some options to indicate when service negotiation should be invoked.
  - Negotiated services will include request/responses which are similar to non-negotiated services, except the return may include items like a "range of vlans that are available for use" and one "suggested vlan" which will be held for a specific amount of time.

# AM API Negotiation

---

- AM API Extension for Negotiation (cont'd)
  - Tree Mode Workflows will use this to tailor requests to subsequent AMs
  - Chain Mode Workflows will use this as part of AM to AM messaging which will work their way down and back up the chain as "vlan range" and "suggested vlan" fields are modified along the way
  - This is a vlan based description, can also be used for negotiation of any other resources in the schema as well

# Tree Mode Stitching Workflow with Negotiation



1. CreateSliver Request, negotiatedservices=on, stitchingmode=tree, vlan=any

2. CreateSliver Reponse, vlanRangeAvailability=3003-3015, suggestedVLANRange=3005

3. CreateSliver Request, negotiatedservices=on, stitchingmode=tree, vlanRangeAvailability=3003-3015, suggestedVLANRange=3005

4. CreateSliver Response, vlanRangeAvailability=3004-3010, suggestedVLANRange=3005

5. CreateSliver Response, negotiatedservices=on, stitchingmode=chain, vlanRangeAvailability=3004-3010, suggestedVLANRange=3010

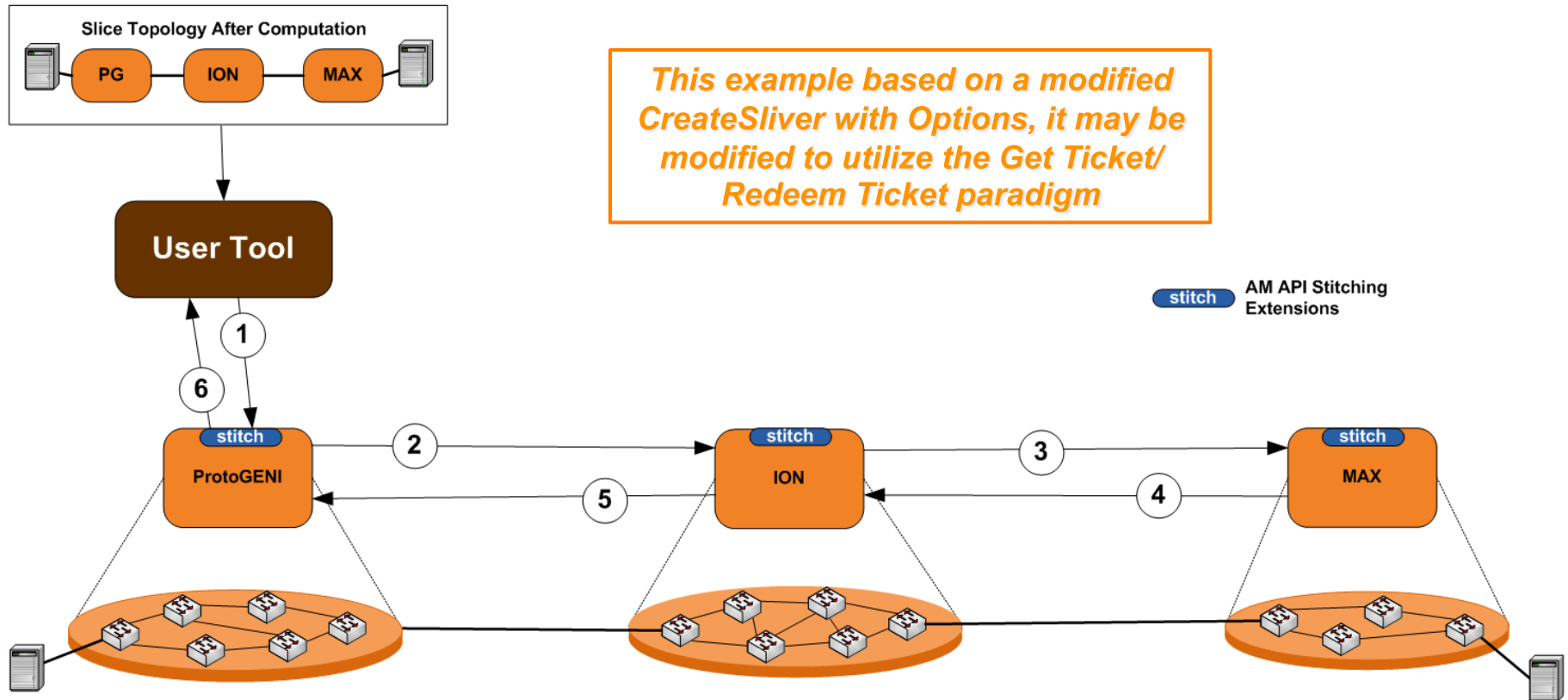
6. CreateSliver Response, vlanRangeAvailability=3007-3009, suggestedVLANRange=3009

7. CreateSliver Request, negotiatedservices=off, stitchingmode=tree, vlanRangeAvailability=3009, suggestedVLANRange=3009

8. CreateSliver Request, negotiatedservices=off, stitchingmode=tree, vlanRangeAvailability=3009, suggestedVLANRange=3009

9. CreateSliver Request, negotiatedservices=off, stitchingmode=tree, vlanRangeAvailability=3009, suggestedVLANRange=3009

# Chain Mode Stitching Workflow with Negotiation



1. CreateSliver Request, negotiatedservices=off, stitchingmode=chain, vlan=any

2. CreateSliver Request, negotiatedservices=on, stitchingmode=chain, vlanRangeAvailability=3003-3015, suggestedVLANRange=3005

3. CreateSliver Request, negotiatedservices=on, stitchingmode=chain, vlanRangeAvailability=3004-3010, suggestedVLANRange=3005

4. CreateSliver Response, negotiatedservices=off, stitchingmode=chain, vlanRangeAvailability=3009, suggestedVLANRange=3009

5. CreateSliver Response, negotiatedservices=off, stitchingmode=chain, vlanRangeAvailability=3009, suggestedVLANRange=3009

6. CreateSliver Response, negotiatedservices=off, stitchingmode=chain, vlanRangeAvailability=3009, suggestedVLANRange=3009

# Topology Service - Next Steps

---

- Topology Service
  - Deploying a "GENI Topology Service" as persistent infrastructure item would be convenient for user to explore the interconnected global Aggregate Space
  - could be based on modified perfSONAR topology service
    - populated by AM Advertisement RSpecs
    - provides basis for visualization of global GENI space
  - may not be necessary for large aggregates/control frameworks with their own advertisement aggregation and computation capabilities

# Computation Service - Next Steps

- Computation Service
  - Deploying a "GENI Computation Service" as persistent infrastructure item would be convenient for user to get specific options to seed their Sliver Requests
  - could be based on modified IDC computation service
    - uses information from the Topology service
  - may not be necessary for large aggregates/control frameworks with their own advertisement aggregation and computation capabilities



# Stitching Architecture

## Stitching Architecture Components

Workflow process may need some intelligence to assist with stitching:

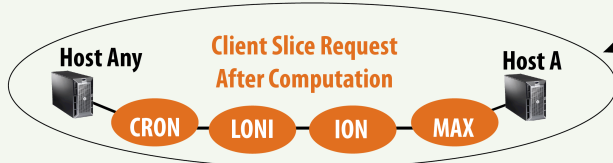
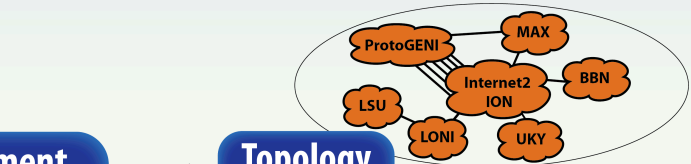
- examine advertisements to see which networks have vlan translation capabilities
- understand which network regions do not have GENI AM (leverage native system APIs)

AMs

Advertisement RSpec w/stitching

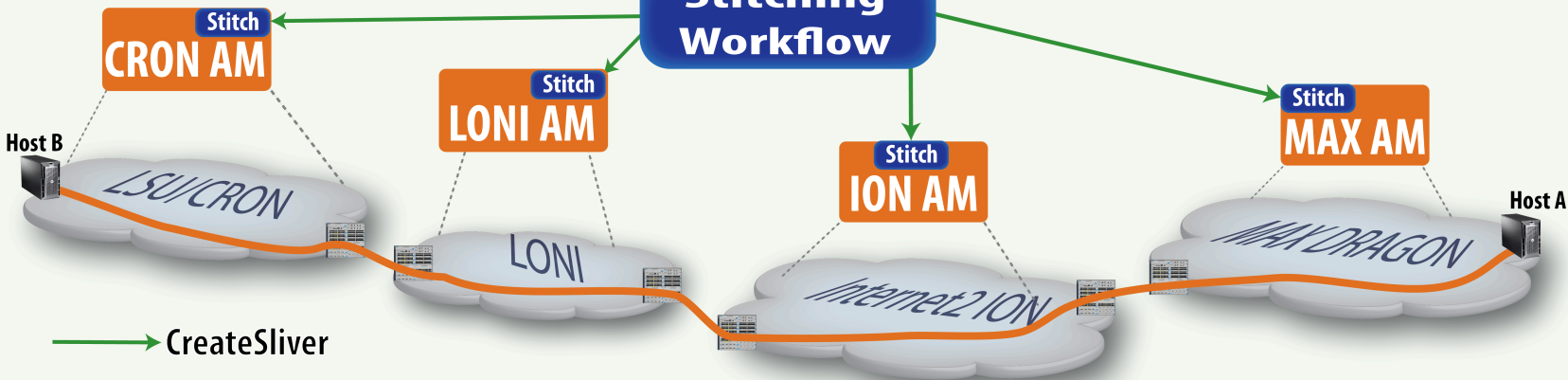
Topology Service

Stitching Path Computation

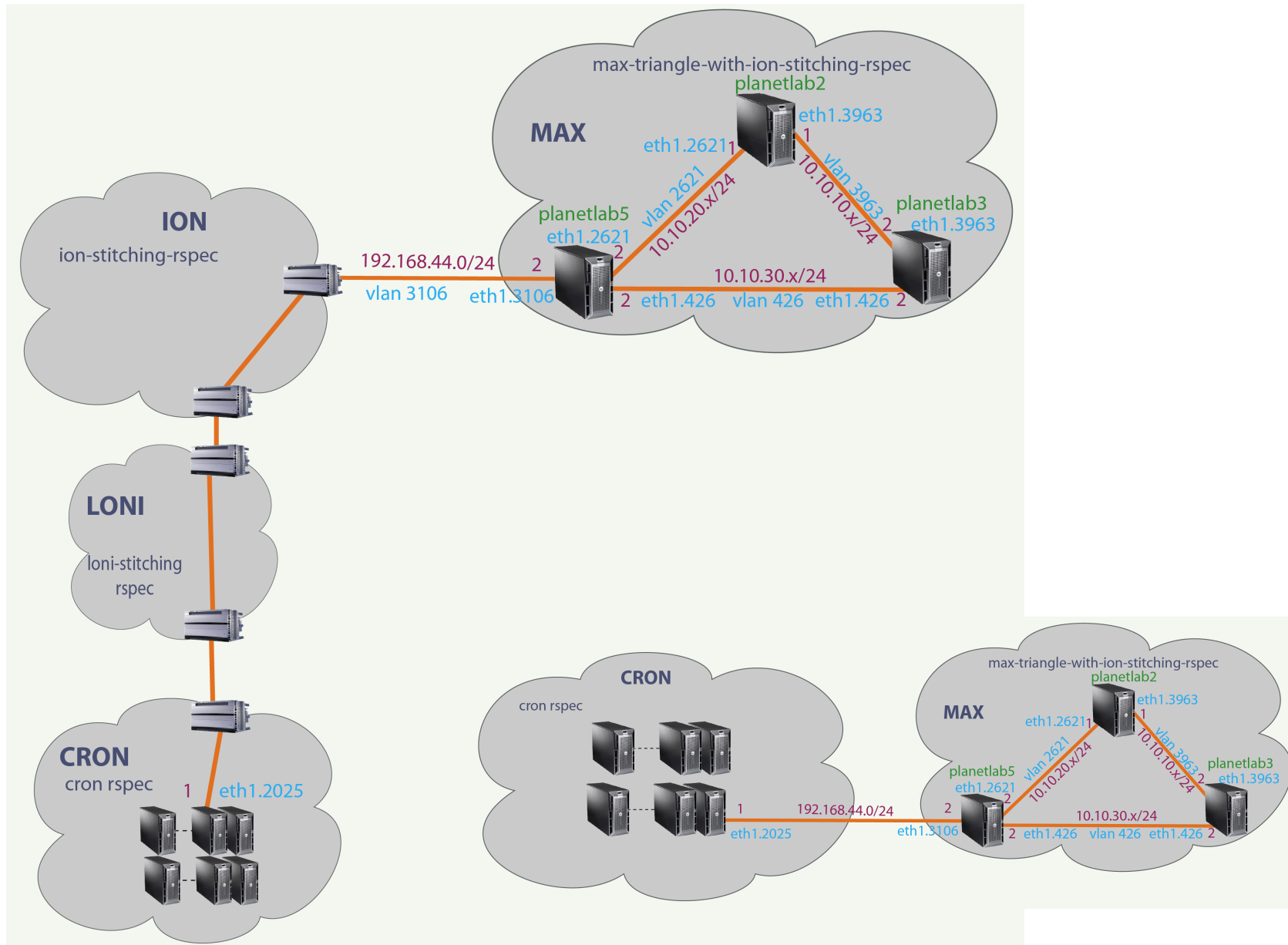


- Stitching Extensions in Advertisements identify Inter-Aggregate TouchPoints
- Topology Service allows construction of the global interconnected Aggregate space
- Computation results feed workflow process
- Request RSpecs include stitching information

## CLIENT Tool Stitching Workflow



# Stitching Topology Examples



# Stitching Topology Examples

