

# *Network RSpec Workshop*

*Chicago O'Hare Hilton*

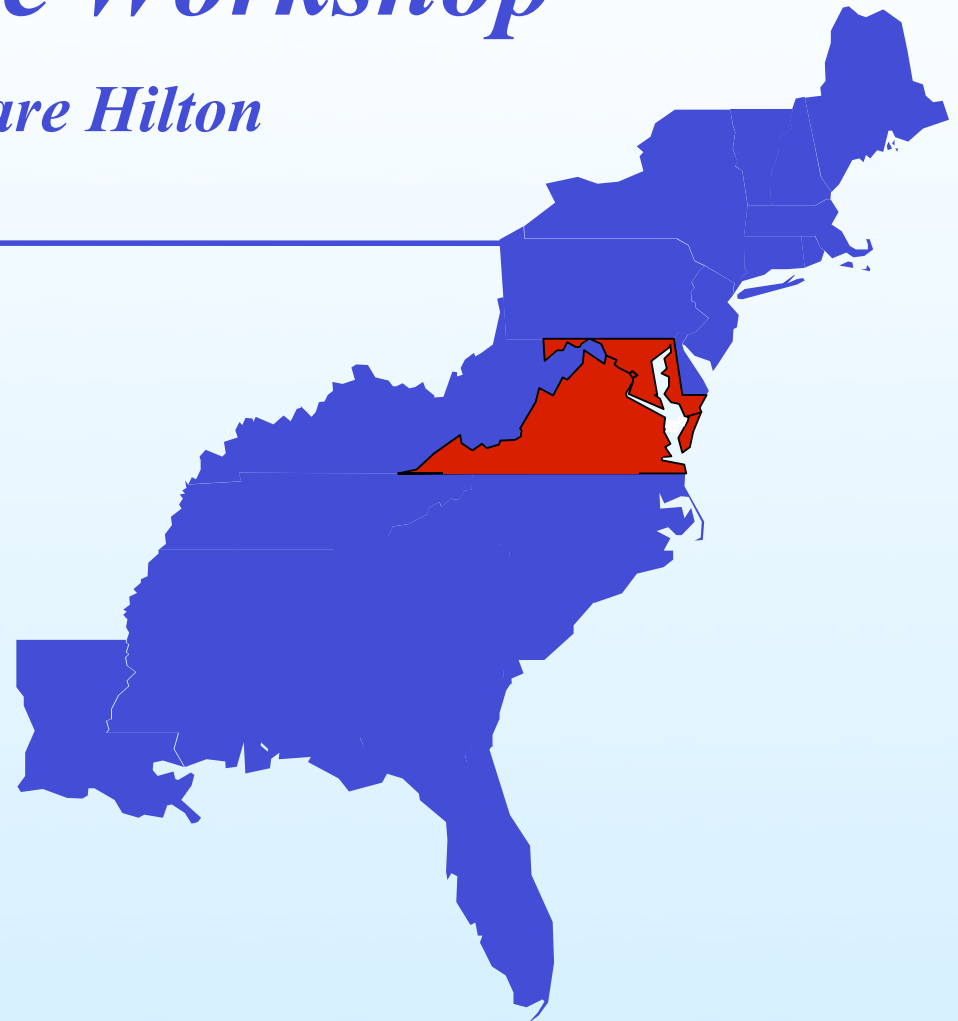
**Chris Tracy**

**Jarda Flidr**

**Peter O'Neil**

Cluster B Participant

June 25th, 2009

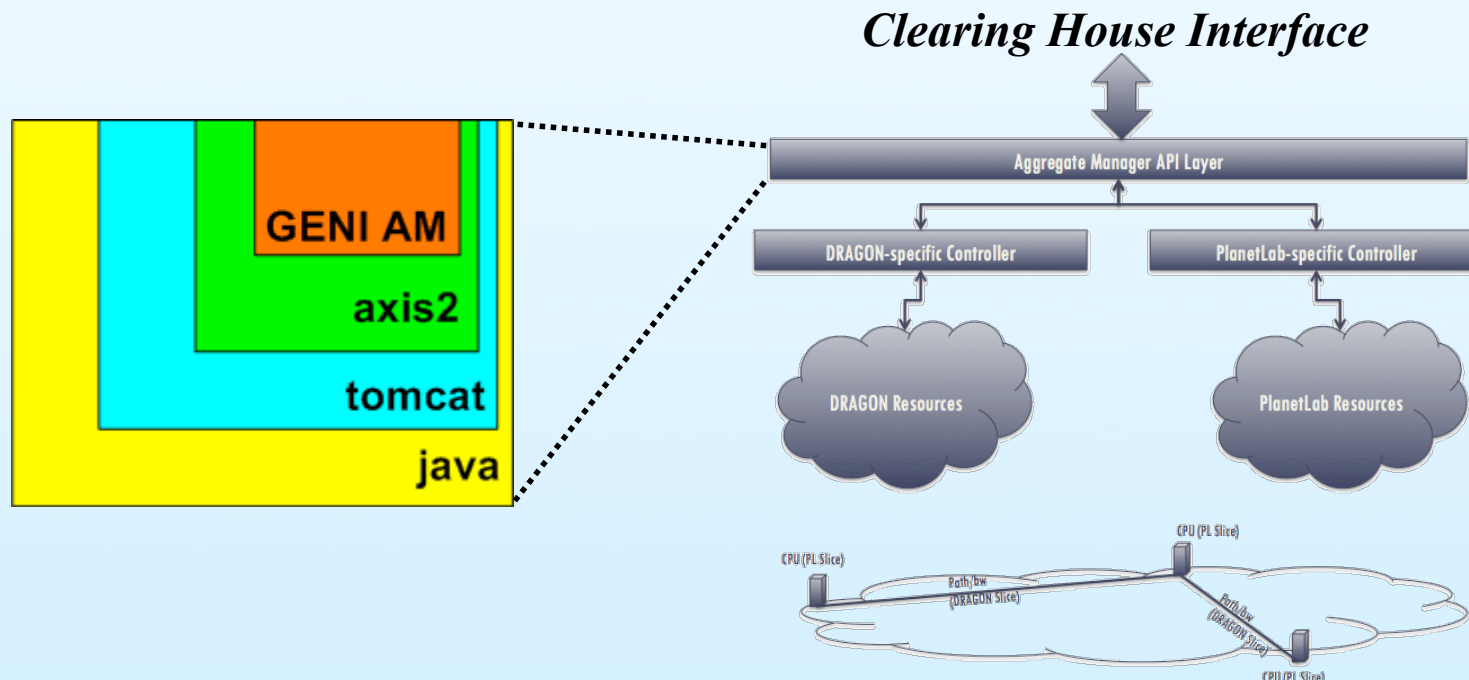


## *Outline*

- SOAP-based GENI Aggregate Manager
  - Overview and Architecture
  - Intended Usage
  - Implementation Details
  - Usage / Interaction Examples / Code
  - Current Issues / Future Plans
- End-to-End Slices Across Network Aggregates
  - Ontology used by DCN Software Suite (DRAGON + OSCARS)
  - Inter-domain Path Computation Example
  - Inter-domain Circuit Provisioning Example
  - Use of Static Tunnels
- Discussion: Exploring Cross-Aggregate Network Slices

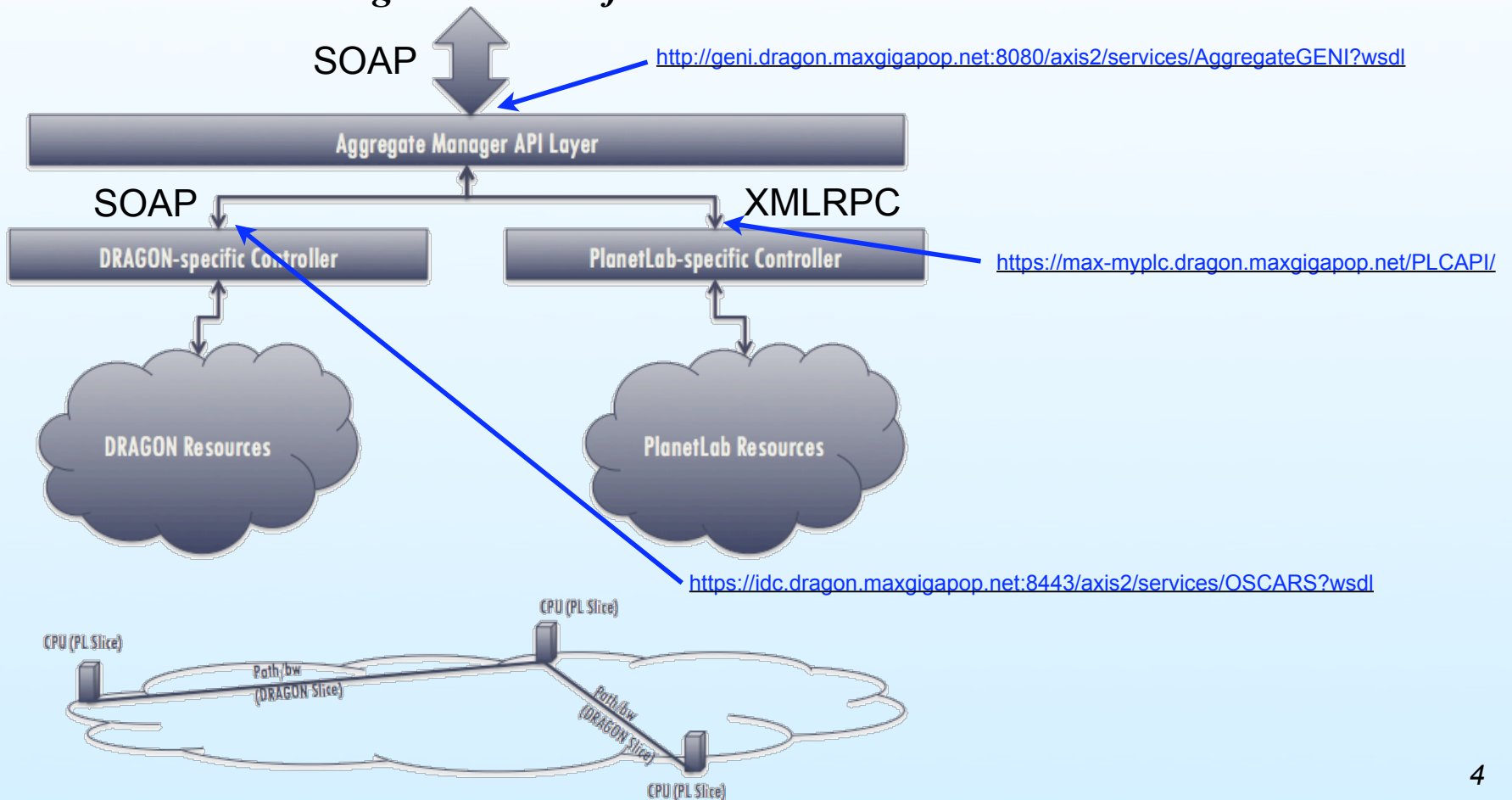
## *SOAP-based GENI Aggregate Manager: Overview and Architecture*

- Java-based reference implementation
- Provides Web Services API (WSDL) to clients
- Deployed in Apache Tomcat as an Axis2 service



# SOAP-based GENI Aggregate Manager: Overview and Architecture

## Clearing House Interface



## ***SOAP-based GENI Aggregate Manager: Intended Usage***

- Intended to control any component located at MAX:
  - PlanetLab virtualization nodes
  - DRAGON network substrate (dynamic E2E VLANs)
  - Eucalyptus virtualization nodes
  - PASTA wireless sensor nodes
  - NetFPGA-based OpenFlow switches
- Architecture is public
- Code base is public
- Instances will be site-specific
  - anybody can use the code, will be limited to their particular aggregate

## ***SOAP-based GENI Aggregate Manager: Intended Usage***

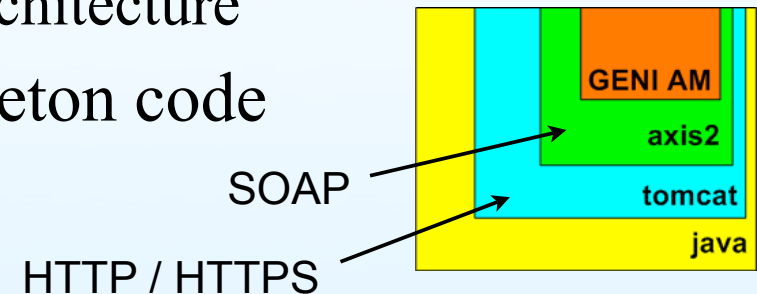
- Could be used on any aggregate component
  - dynamic networks
  - virtualized compute resources
  - cloud resources
  - etc.
- Framework is relatively generic
  - could be used by anybody who is interested in experimenting with a Java-based Aggregate Manager using Web

## ***SOAP-based GENI Aggregate Manager: Intended Usage***

- One instance might be deployed on a network aggregate running the DCN Software Suite (DCNSS)
- DCNSS := OSCARS + DRAGON
  - » <https://wiki.internet2.edu/confluence/display/DCNSS>
  - OSCARS: SOAP-based Web Services API, provides advance reservation (book-ahead scheduling) and strong AAA
    - » interfaces to DRAGON or other low-level control plane software
  - DRAGON: GMPLS control plane, provides on-demand circuit provisioning
    - » interfaces to switch hardware
  - “GENI-ize” a network aggregate running the DCNSS

## *SOAP-based GENI Aggregate Manager: Implementation Details*

- AggregateGENI.wsdl file written by hand
  - Based on Slice-based Facility Architecture
- wsdl2java to generate Java skeleton code
- Dependencies:
  - Java Development Kit
  - Apache Tomcat (application container, HTTP/HTTPS transport)
  - Apache Axis2 for SOAP messaging
  - Apache Ant (build & deploy)
  - SQL database





## ***SOAP-based GENI Aggregate Manager: Usage / Interaction Examples***

- Using a Web Services / SOAP client, we can interact with the service endpoint.
  - Show Apache Axis2 Available Services page:
    - <http://geni.dragon.maxgigapop.net:8080/axis2/services/listServices>
    - Click on AggregateGENI to view WSDL file
- Load up SOAP Client.app
  - <http://www.ditchnet.org/soapclient/>
  - point it to WSDL file / service endpoint:
    - <http://geni.dragon.maxgigapop.net:8080/axis2/services/AggregateGENI?wsdl>
    - set Method == ListCapabilities, click Execute
    - next, ListNodes with capability=dragon
      - » urn:ogf:geni:domain=dragon.maxgigapop.net:capability=dragon

## ***SOAP-based GENI Aggregate Manager: Usage / Interaction Examples***

- Using Web Services perl client
  - run ws\_client.pl
    - » example to demonstrate the interface use
    - » just a test tool intended for development
  - examine output in TextMate.app (easier to read)
- Demonstrate ListNodes with capabilities filter:
  - ask for nodes with capabilities:
    - » dragon & planetlab
    - » pasta
    - » pasta & eucalyptus
    - » openflow & dragon
  - controllerURL returned in response

## ***SOAP-based GENI Aggregate Manager: Code***

- Available in Subversion:
  - `svn co svn://svn.maxgigapop.net/geni-aggregate`
- Documentation/additional information will be posted here:
  - <https://geni.maxgigapop.net/twiki/bin/view/GENI/Software>

## ***SOAP-based GENI Aggregate Manager: Current Issues / Future Plans***

- This is only a prototype, so:
  - Ticketing and authentication is still a work in progress
    - » HTTPS could be used for encryption
    - » Signed SOAP messages for authentication
  - Back-end hooks to provisioning systems is still under development
    - » will interface to OSCARS/DRAGON via Web Services API
    - » will interface to PlanetLab via XMLRPC directly to PLCAPI, or XMLRPC (or SOAP) to PlanetLab GENIWrapper AM
    - » will be extended to find OpenFlow controllers or other technologies

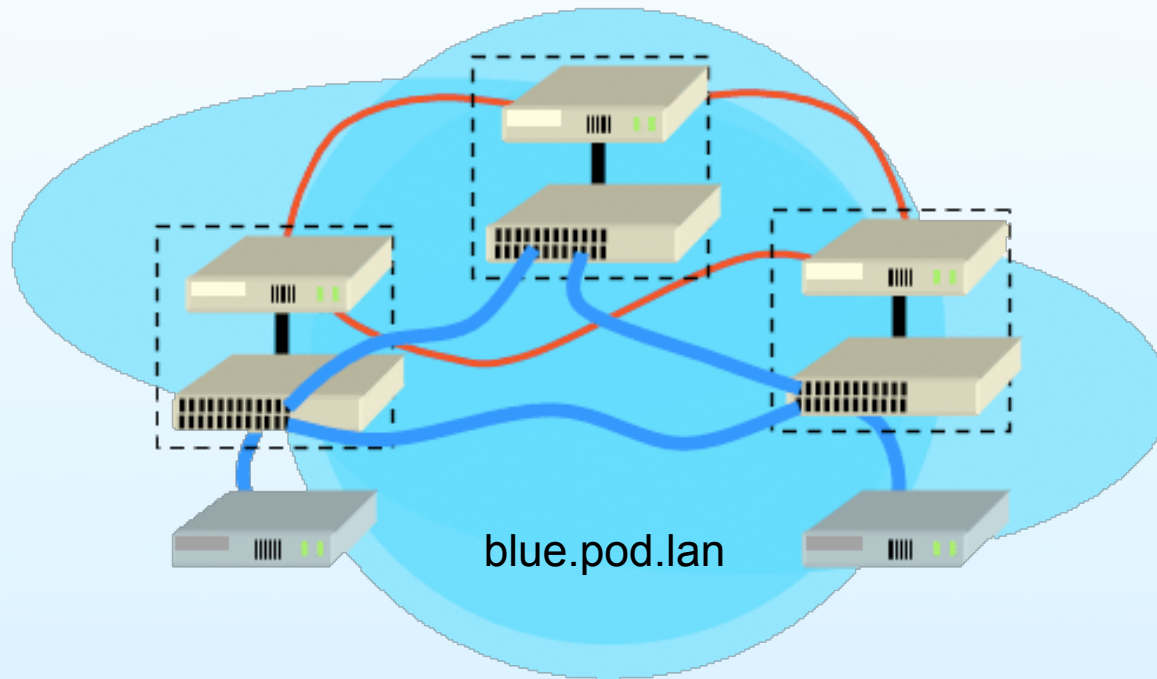
## ***SOAP-based GENI Aggregate Manager: Current Issues / Future Plans***

- End-to-end slices across AM's will require something very similar to the inter-domain interaction used to create inter-domain dynamic circuits in networks like DRAGON, Internet2 DCN, ESnet, etc.
- For example:
  - calculate the end-to-end slice (multi-AM slice) first, see if it is achievable
  - then go from AM to AM and try to provision all of the resources
- We believe this will look something like our Path Computation Element (PCE) now
  - but will be more like a Resource Computation Engine (RCE) where Path will be just one of the constraints
  - ...which leads to the discussion about how our PCE is designed

## *End-to-End Slices Across Network Aggregates*

- Ontology used by DCN Software Suite
  - Domains
  - Nodes
  - Ports
  - Links
- Based on OGF Network Measurements Working Group (NM-WG) control plane schema
  - <https://forge.gridforum.org/sf/projects/nm-wg>

## *Domain*



URN identifying a particular domain  
urn:ogf:network:domain=blue.pod.lan

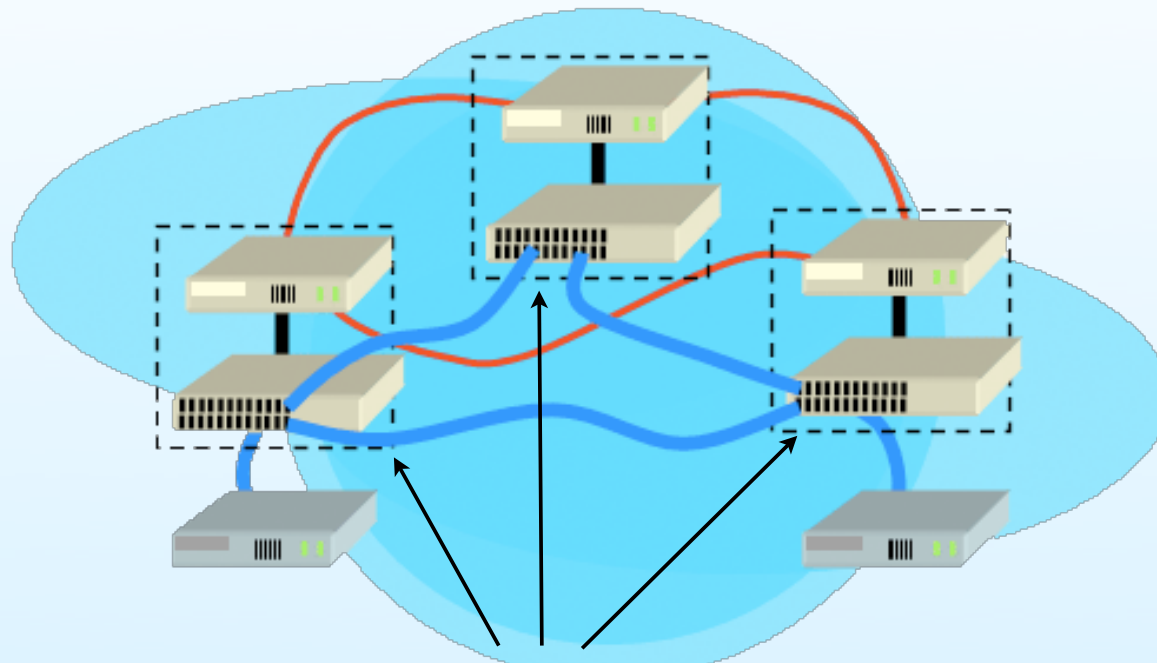
## *Domain*

```
<?xml version="1.0" encoding="UTF-8"?>
<topology
  xmlns="http://ogf.ogf/schema/network/topology/ctrlPlane/20071023/"
  id="blue-topology">
  <idcId>https://idc.blue.pod.lan:8443/axis2/services/OSCARS</idcId>
  <domain id="urn:ogf:network:domain=blue.pod.lan">
</topology>
```

↑  
URN identifying a particular domain



## *Nodes*



3 URNs identifying switching nodes

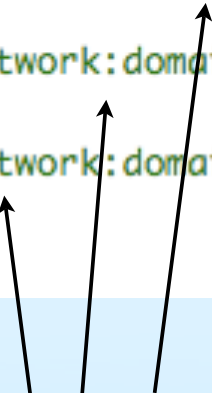
urn:ogf:network:domain=blue.pod.lan:node=vlsr1

urn:ogf:network:domain=blue.pod.lan:node=vlsr2

urn:ogf:network:domain=blue.pod.lan:node=vlsr3

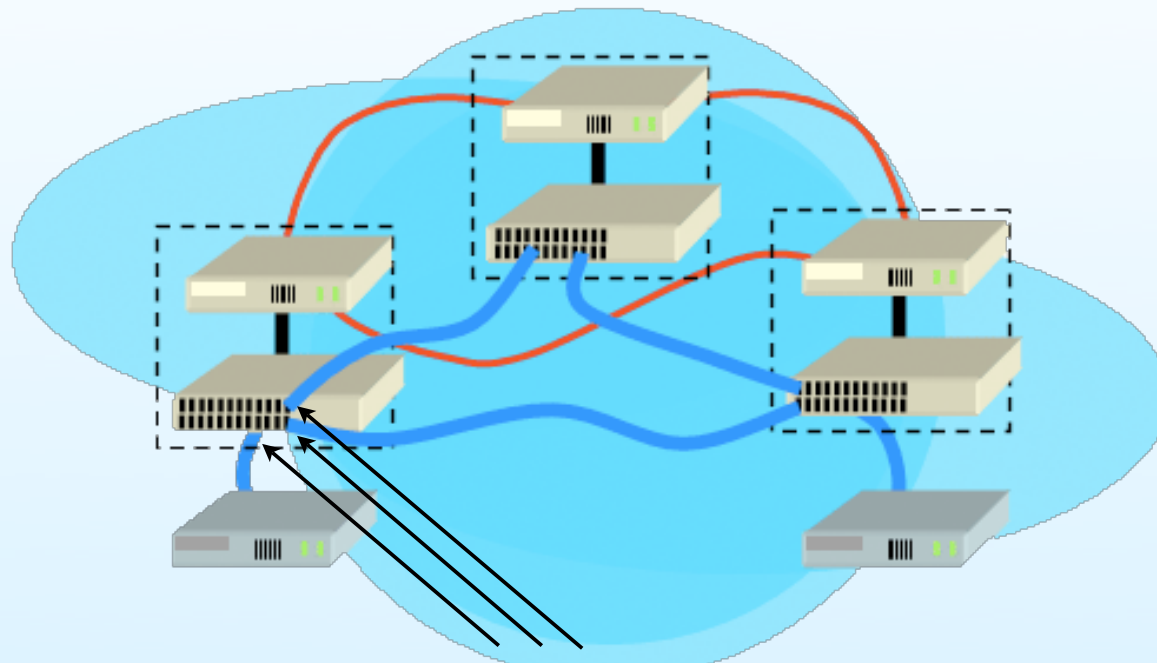
## Nodes

```
<?xml version="1.0" encoding="UTF-8"?>
<topology
  xmlns="http://ogf.ogf/schema/network/topology/ctrlPlane/20071023/"
  id="blue-topology">
  <idcId>https://idc.blue.pod.lan:8443/axis2/services/OSCARS</idcId>
  <domain id="urn:ogf:network:domain=blue.pod.lan">
    <!-- BLUE VLSR1 -->
    <node id="urn:ogf:network:domain=blue.pod.lan:node=vlsr1">
    <!-- BLUE VLSR2 -->
    <node id="urn:ogf:network:domain=blue.pod.lan:node=vlsr2">
    <!-- BLUE VLSR3 -->
    <node id="urn:ogf:network:domain=blue.pod.lan:node=vlsr3">
  </domain>
</topology>
```



3 URNs identifying switching nodes

## *Ports*



3 URNs identifying physical switch ports on 1 node  
urn:ogf:network:domain=blue.pod.lan:node=vlsr1:port=3  
urn:ogf:network:domain=blue.pod.lan:node=vlsr1:port=4  
urn:ogf:network:domain=blue.pod.lan:node=vlsr1:port=5

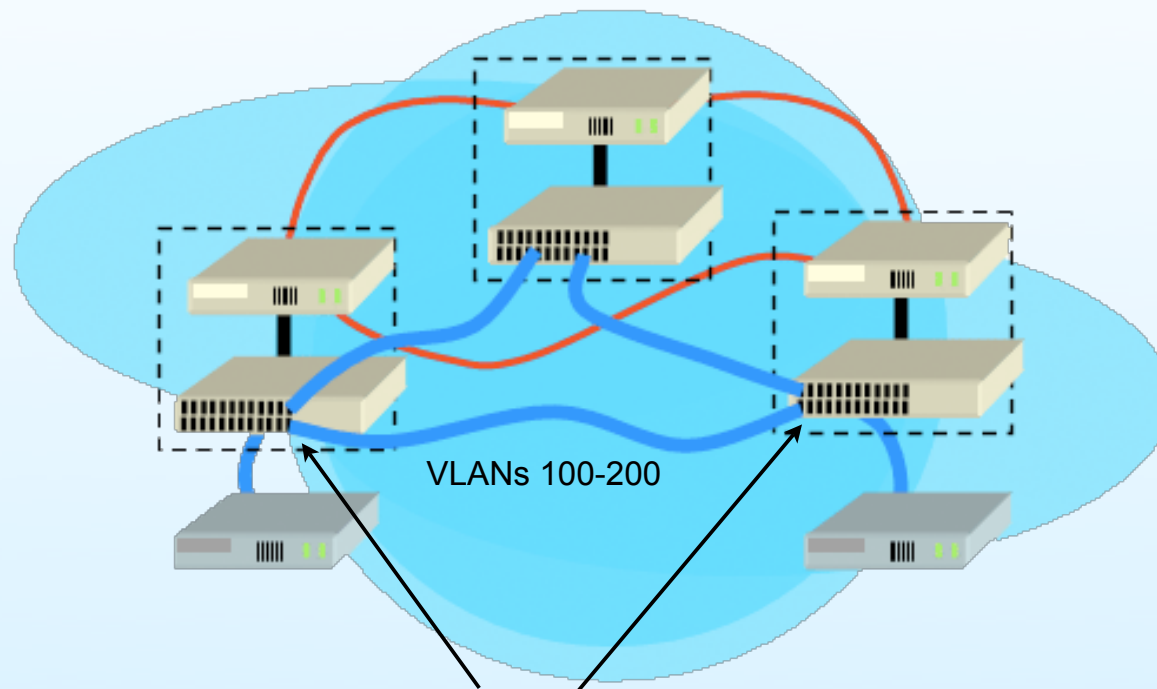
## Ports

```
<?xml version="1.0" encoding="UTF-8"?>
<topology
  xmlns="http://ogf.ogf/schema/network/topology/ctrlPlane/20071023/"
  id="blue-topology">
  <idcId>https://idc.blue.pod.lan:8443/axis2/services/OSCARS</idcId>
  <domain id="urn:ogf:network:domain=blue.pod.lan">
    <!-- BLUE VLSR1 -->
    <node id="urn:ogf:network:domain=blue.pod.lan:node=vlsr1">
      <address>192.168.2.4</address>
      <port id="urn:ogf:network:domain=blue.pod.lan:node=vlsr1:port=3">
      <port id="urn:ogf:network:domain=blue.pod.lan:node=vlsr1:port=4">
      <port id="urn:ogf:network:domain=blue.pod.lan:node=vlsr1:port=5">
    </node>
    <!-- BLUE VLSR2 -->
    <node id="urn:ogf:network:domain=blue.pod.lan:node=vlsr2">
    <!-- BLUE VLSR3 -->
    <node id="urn:ogf:network:domain=blue.pod.lan:node=vlsr3">
  </domain>
</topology>
```

3 URNs identifying physical switch ports on 1 node

## *Link*

Note: One physical port *may* contain multiple logical links



2 URNs identifying link endpoints for a single link

urn:ogf:network:domain=blue.pod.lan:node=vlsr1:port=5:link=11.2.3.1

urn:ogf:network:domain=blue.pod.lan:node=vlsr3:port=5:link=11.2.3.2

## Link

```
<node id="urn:ogf:network:domain=blue.pod.lan:node=vlsr1">
  <address>192.168.2.4</address>
  <port id="urn:ogf:network:domain=blue.pod.lan:node=vlsr1:port=3">
  <port id="urn:ogf:network:domain=blue.pod.lan:node=vlsr1:port=4">
  <port id="urn:ogf:network:domain=blue.pod.lan:node=vlsr1:port=5">
    <capacity>1000000000</capacity>
    [...]
    <link id="urn:ogf:network:domain=blue.pod.lan:node=vlsr1:port=5:link=11.2.3.1">
      <remoteLinkId>urn:ogf:network:domain=blue.pod.lan:node=vlsr3:port=5:link=11.2.3.2</remoteLinkId>
      <capacity>1000000000</capacity>
      [...]
      <SwitchingCapabilityDescriptors>
        <switchingcapType>l2sc</switchingcapType>
        <encodingType>ethernet</encodingType>
        <switchingCapabilitySpecificInfo>
          <interfaceMTU>9000</interfaceMTU>
          <vlanRangeAvailability>0,100-200</vlanRangeAvailability>
        </switchingCapabilitySpecificInfo>
      </SwitchingCapabilityDescriptors>
    </link>
  </port>
</node>
```

for an inter-domain link, this would show a different domain

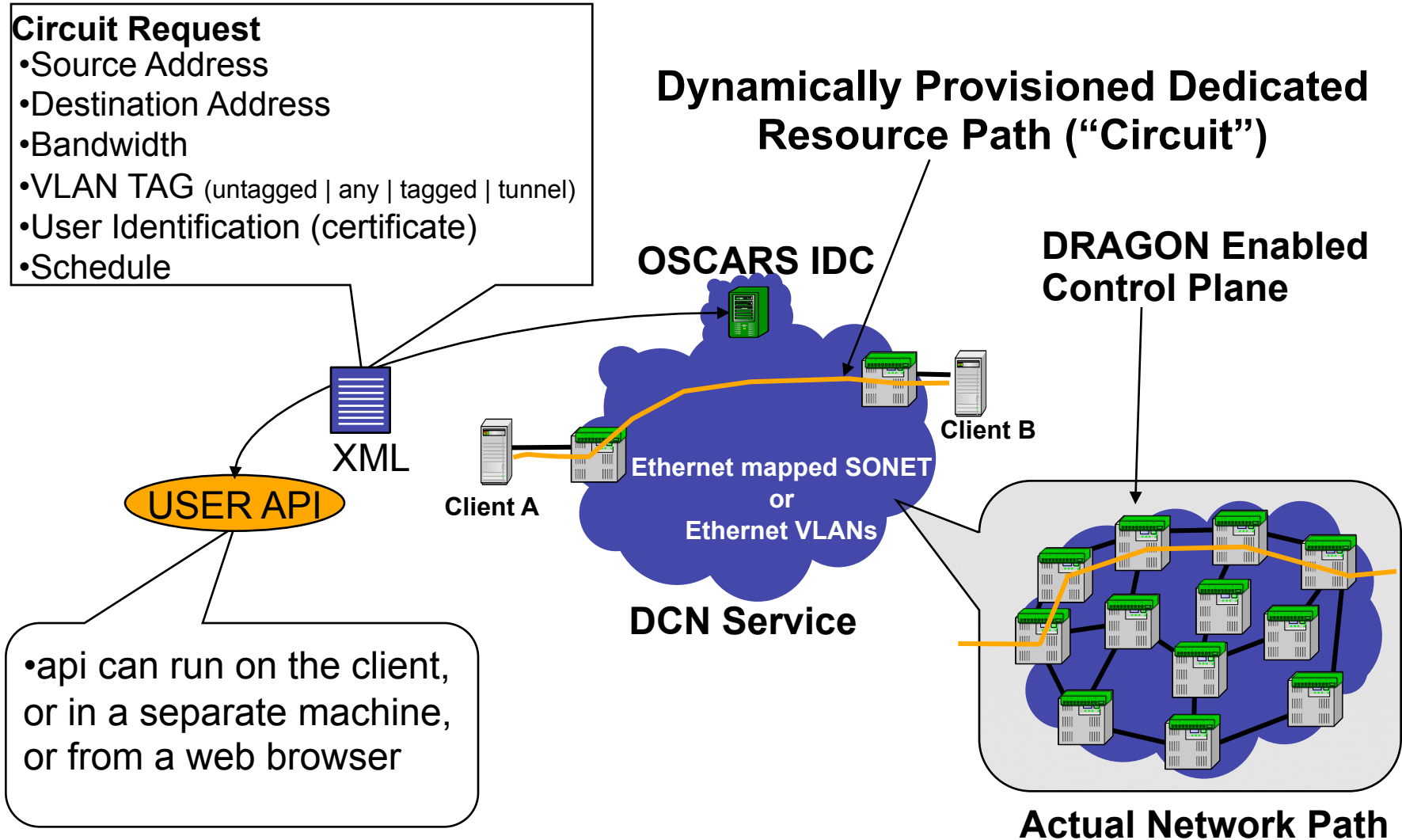
switching technology specific info for Ethernet, a range of VLANs

2 URNs identifying link endpoints for a single link

## *End-to-End Slices Across Network Aggregates*

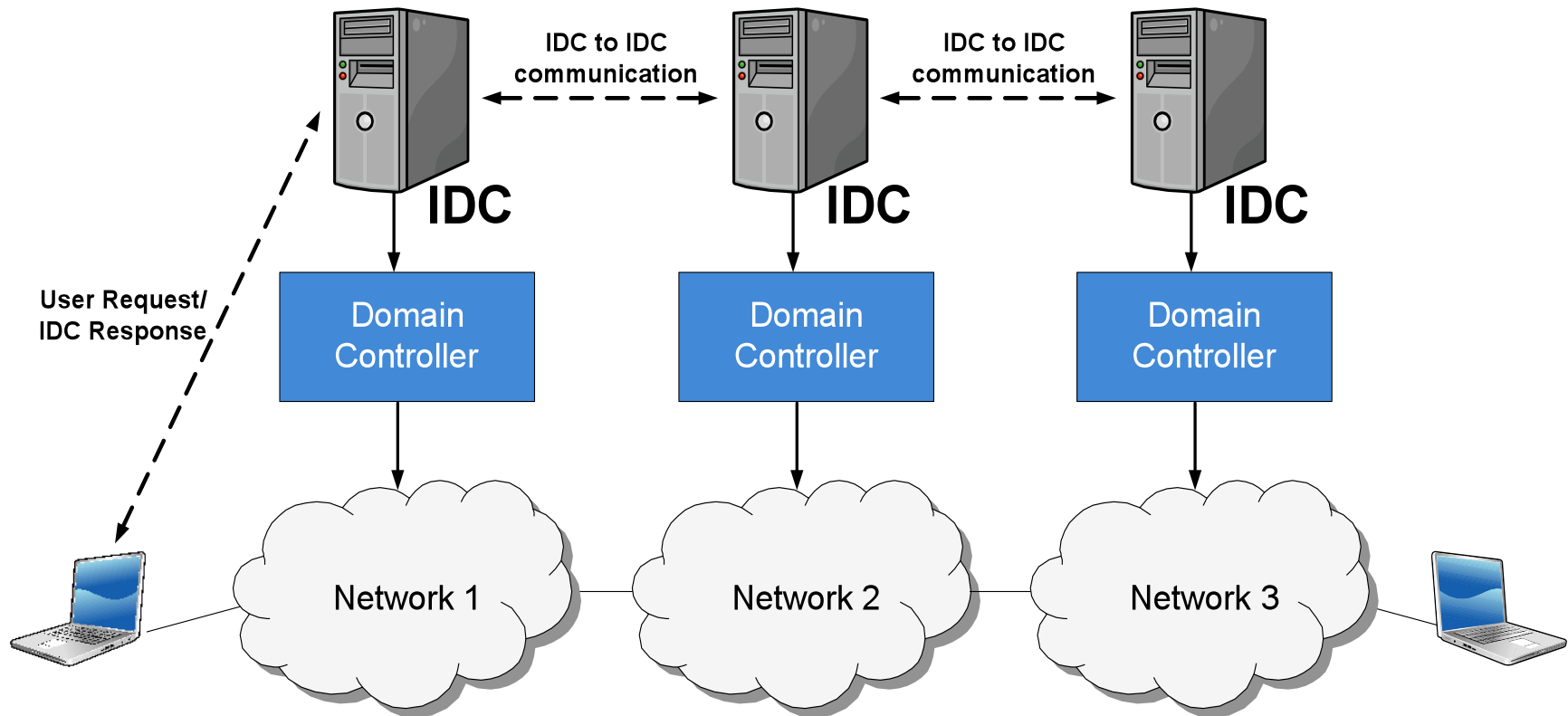
- Inter-domain Path Computation Example
- Inter-domain Circuit Provisioning Example

# Dynamic Network Services Intra-Domain





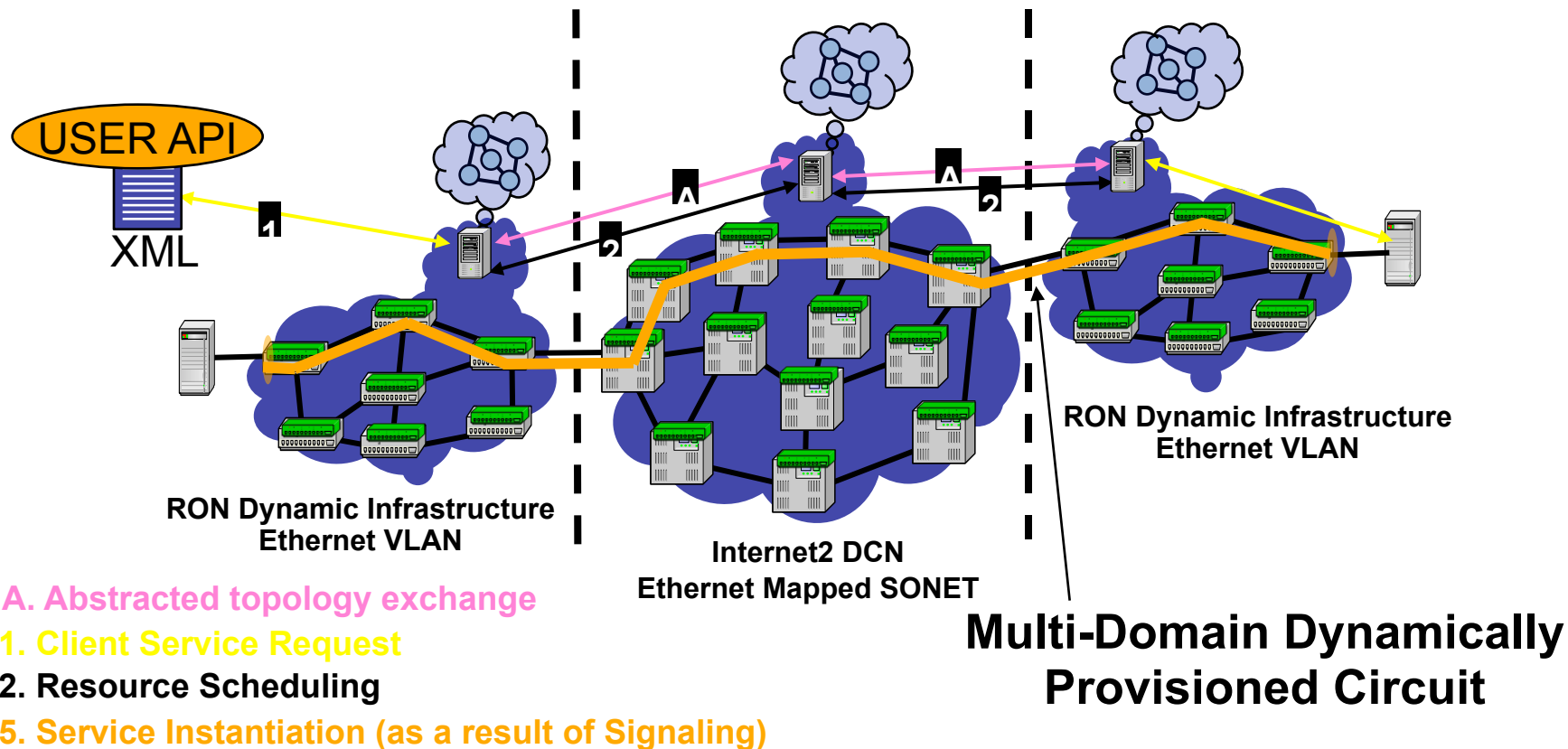
# Inter-Domain Circuit Provisioning



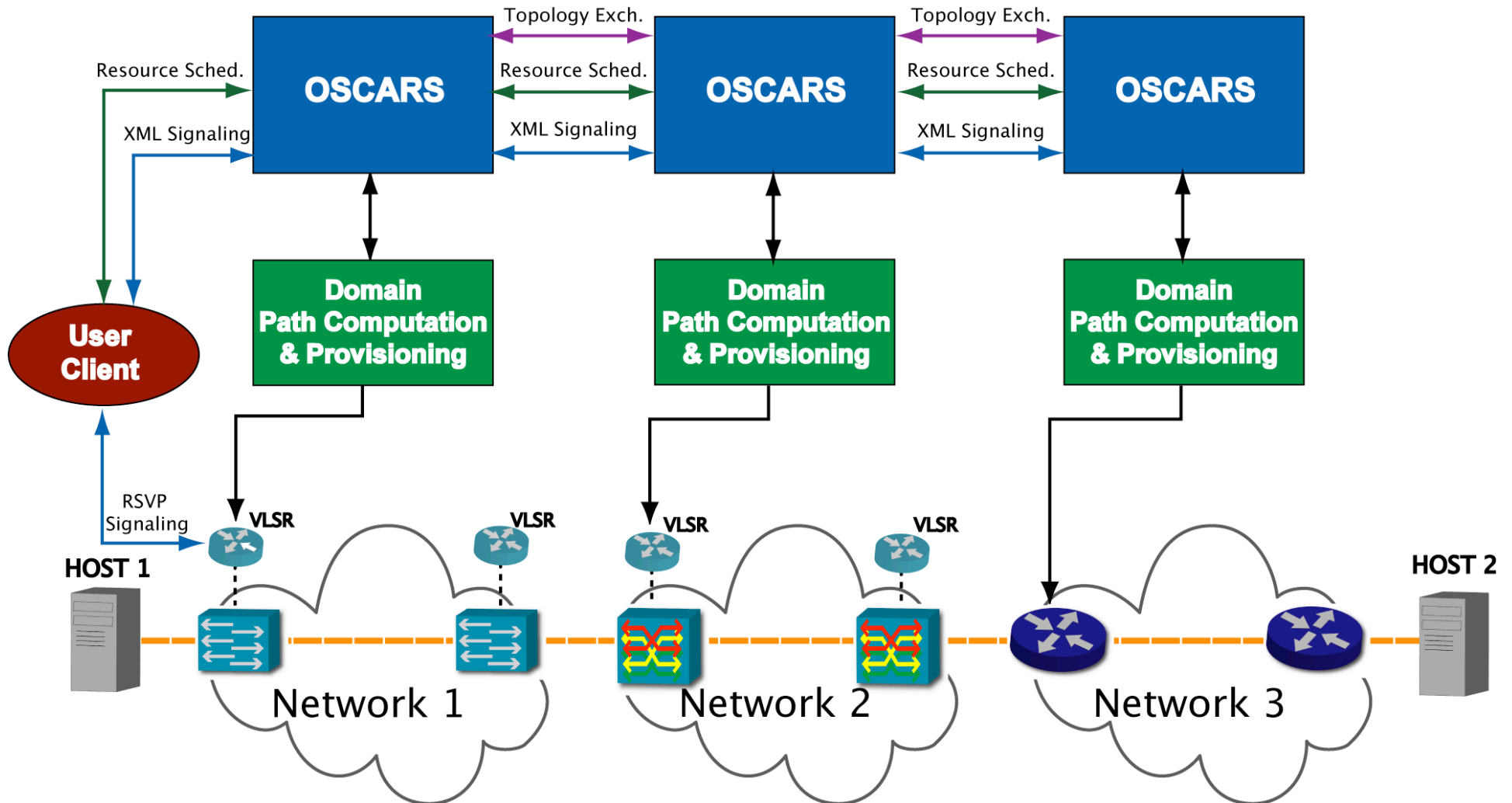
# Dynamic Network Services

## Inter-Domain

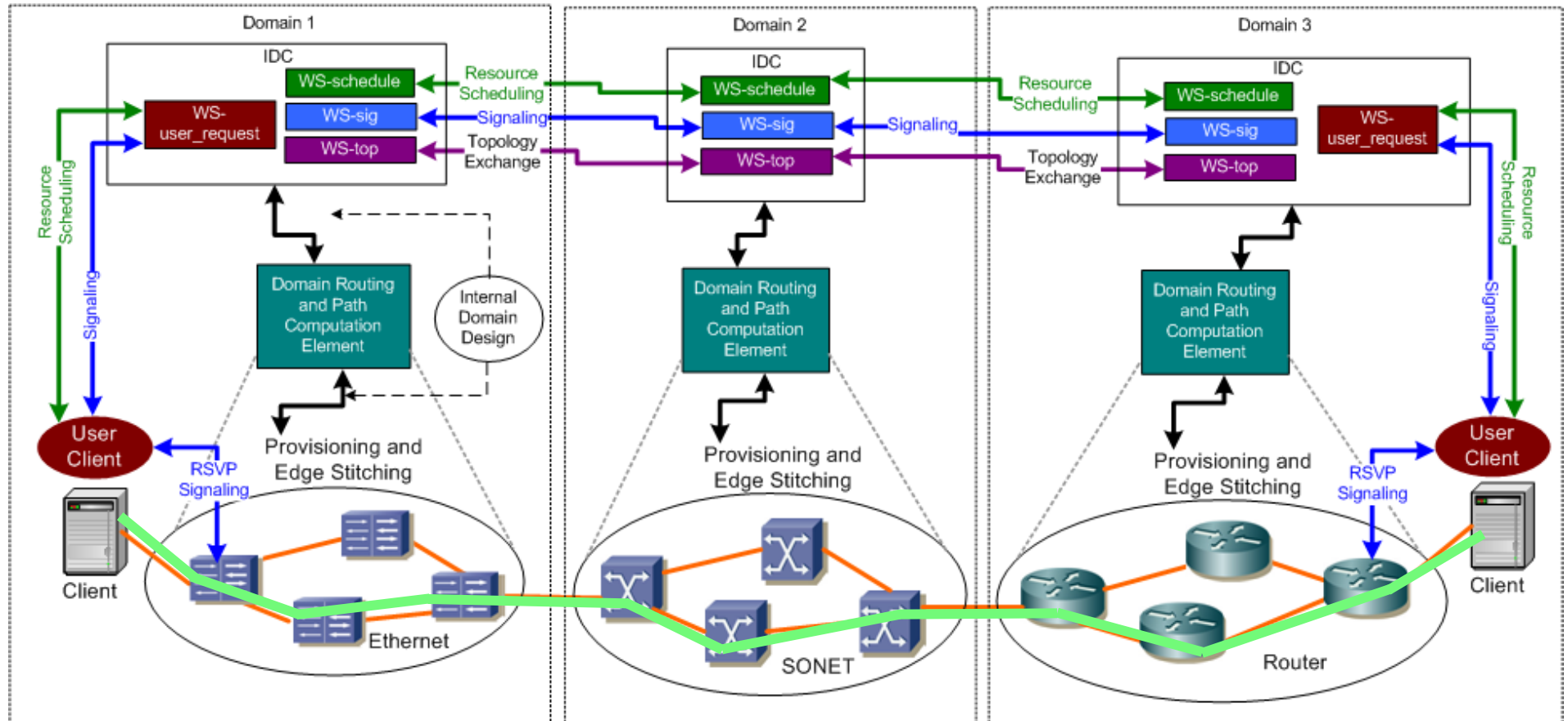
- No difference from a client (user) perspective for InterDomain vs IntraDomain



# OSCARS-DRAGON Integration



# IDC - Web Service Based Definition



- Four Primary Web Services Areas:
  - Topology Exchange, Resource Scheduling, Signaling, User Request

## *Discussion*

- How Do We Specify (Network) Resources?
  - OGF26 Network Markup Language Working Group
  - Multilayer NDL presentation by Freek Dijkstra
    - » covers introduction to G.805, generalized method to model transport networks
    - » covers development of the Network Description Language
  - <https://forge.gridforum.org/sf/go/doc15666?nav=1>

## *Discussion*

- Exploring Cross-Aggregate Network Slices
- Goal:
  - support non-IP compatible traffic flows
  - construct “slices” that cross aggregate boundaries
    - » could be VLANs, tunnels, ...
- What can/can't we complete before the end of Spiral 1?
  - end of September 2009
- What are the issues/challenges that we face?
  - GENI Clearing House service?

***Thanks!***

<http://groups.geni.net/geni/wiki/Mid-Atlantic%20Crossroads>

<http://geni.maxgigapop.net>

(under construction)