RSpecs
Engineering Meeting

Aaron Helsinger
July 27, 2011
www.geni.net

Agenda

- Introduction and GENI RSpecs roll-out – Aaron Helsinger
- RSpec Converter at Orca – Ilia Baldine
- Network and Compute Ontologies – Ilia Baldine
- Invited Viewpoints on Ontologies
  - Max Ott (compute resources, wireless resources, …)
  - Mike Zink (storage resources)
  - Hongwei Zhang (wireless resources – LENS)
- Discussion: Ontologies
- Conclusion – Aaron Helsinger

RSpecs:
Introduction and Status

Aaron Helsinger – ahelsing@bbn.com

July 27, 2011

www.geni.net

Introduction Outline

- Review GEC10
- Status Overview
- AM API Revisions
- PGV2 Rollout and Experiences
- Future Work
- Goals for Today

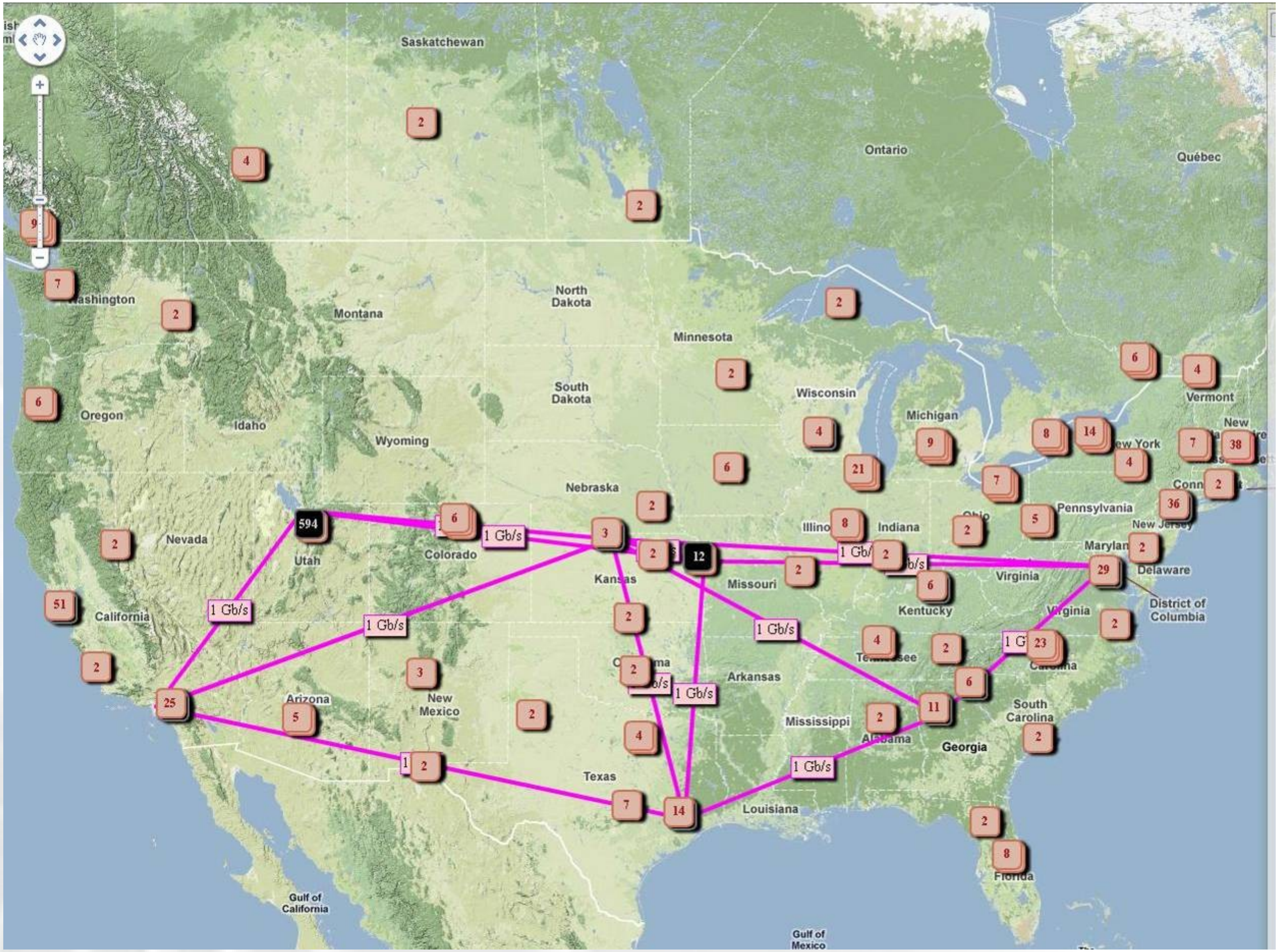
GEC10 Review

- Agreed on a need for a common language
  - A common language for describing resources helps new aggregates and experimenters find each other.
- Common object model is key
  - Ilia is driving the development of common ontologies/models
- Common format on the wire to simplify code
  - At GEC10 we adopted ProtoGENI V2 as the GENI standard format for communicating between GENI tools and services
- Aggregates may use translators

Status Overview

- Aggregate Manager API Draft Revisions
- ProtoGENI V2 (PGV2) Schemas
- Deployment of PGV2 Support
- Ontologies
  - Network discussed at GEC10, based on NML
  - Compute discussed via email since GEC10 (Orca, PG, PL, Kansei, Seattle and OMF). Discuss today.
  - Wireless discussed via email since GEC10 (Kansei & OMF). Discuss today.
  - Storage just beginning (Mike Zink). Discuss today.
  - Others (OpenFlow, measurement, component lifecycle, ?)

Resources available with GENI RSpecs



Status: Draft AM API Revisions

- Aggregate Managers must advertise what RSpec formats they support
- AM API must specify that the RSpec arguments and returns are in GENI standard format (PGV2)
- Jon Duerig of ProtoGENI proposed AM API revisions to support this and discussed on dev@geni.net
- Published to GENI wiki and dev@geni.net
  - http://groups.geni.net/geni/wiki/GAPI_AM_API_DRAFT
- Revisions remain in DRAFT form

AM API Revisions: Details

- All RSpec arguments and returns should be in GENI standard format (follow the PGV2 schema with extensions)
- GetVersion allows advertising default and supported type, version, schema, namespace and extensions
  - Type/version like ProtoGENI 2 is an alias to the schema, namespace and extensions
- ListResources accepts an rspec_version option requesting a particular type/version as advertised in GetVersion
- CreateSliver accepts an RSpec in any advertised format and responds in kind

AM API Revisions: GetVersion from PLC

```
'ad_rspec_versions': [   {   'extensions': [   'http://www.protogeni.net/resources/rspec/ext/gre-tunnel/1',
                                             'http://www.protogeni.net/resources/rspec/ext/other-ext/3'],
                             'namespace': 'http://www.protogeni.net/resources/rspec/2',
                             'schema': 'http://www.protogeni.net/resources/rspec/2/ad.xsd',
                             'type': 'ProtoGENI',
                                'version': '2'},
                         {   'extensions': [],
                             'namespace': None,
                             'schema': None,
                             'type': 'SFA',
                                'version': '1'}],
  'default_ad_rspec': {   'extensions': [],
                          'namespace': None,
                          'schema': None,
                          'type': 'SFA',
                          'version': '1'},
  'request_rspec_versions': [   {   'extensions': [   'http://www.protogeni.net/resources/rspec/ext/gre-tunnel/1',
                                                  'http://www.protogeni.net/resources/rspec/ext/other-ext/3'],
                                  'namespace': 'http://www.protogeni.net/resources/rspec/2',
                                  'schema': 'http://www.protogeni.net/resources/rspec/2/request.xsd',
                                  'type': 'ProtoGENI',
                                  'version': '2'},
                              {   'extensions': [],
                                  'namespace': None,
                                  'schema': None,
                                  'type': 'SFA',
                                  'version': '1'}],
```

PLC advertises that it talks PGV2 or SFA. SFA remains the default.

Status: GENI RSpec format support

- ProtoGENI V2 schema live as of ~ 4/21/11
  - Documented at: http://www.protogeni.net/trac/protogeni/wiki/RSpec
- Aggregates advertising support for PGV2
  - PlanetLab (SFA-internal translator)
  - PlanetLab Europe (SFA-internal translator)
  - ProtoGENI Utah
  - ProtoGENI Kentucky
  - GPO lab: pgeni3
- Aggregates that talk PGV2 by default: none
- Aggregates with support in process
  - Other ProtoGENI sites
  - Other PlanetLab and MyPLC sites – latest SFA includes support
  - Orca – Ilia will discuss his translator
  - OpenFlow aggregates
    - GPO has contributed code that is in testing, using internal translator

Status: Deployment Details

- ProtoGENI support
  - PGV0.2 or V0.1 remains the default
  - GENI standard is an option and will become the default
- PlanetLab support
  - SFA native format remains the default, GENI standard is an option
  - See next slide
- OpenFlow support
  - Current format will remain the default, with GENI standard as an option
  - Code in testing
  - See following slides

GENI RSpec support at PlanetLab

- PlanetLab (Tony Mack) and PG (Jon Duerig) collaborated
- Found that PL concepts all map to existing PG concepts – no new schema extension needed
- Solution
  - Implement a translator internal to PlanetLab (SFA module) that converts PGV2 to internal data structures
- SFA continues to add tags to support additional features (eg init-script, services)

GENI RSpec support at OpenFlow

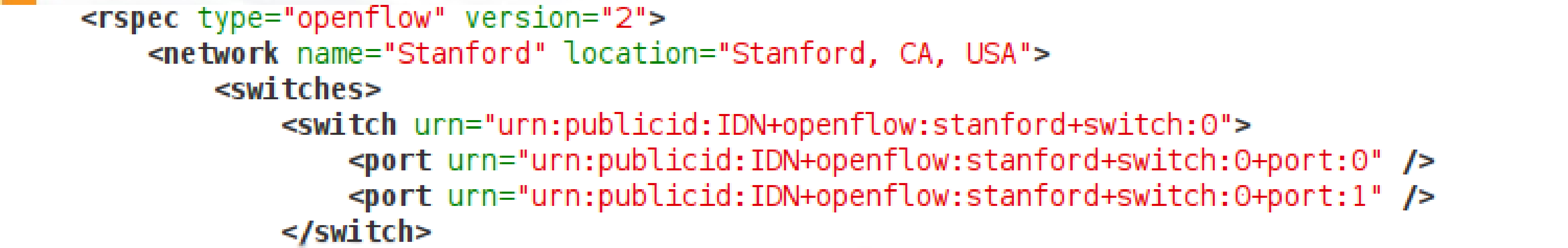
- Identified concepts that overlap, and some new concepts
- Aim for minimal disruption to existing structures
- Chris Benninger of U Victoria implemented the change
- Solution
  - Build an OpenFlow extension to describe ‘flowspace’, etc
  - 1 schema per RSpec type (ad, request, manifest)
  - OpenFlow extension includes PG base namespace
  - OpenFlow AM specifies it talks PGV2, but lists the OpenFlow extension in the extensions list
  - OpenFlow AM translates internally

RSpec Extension How to: Overview

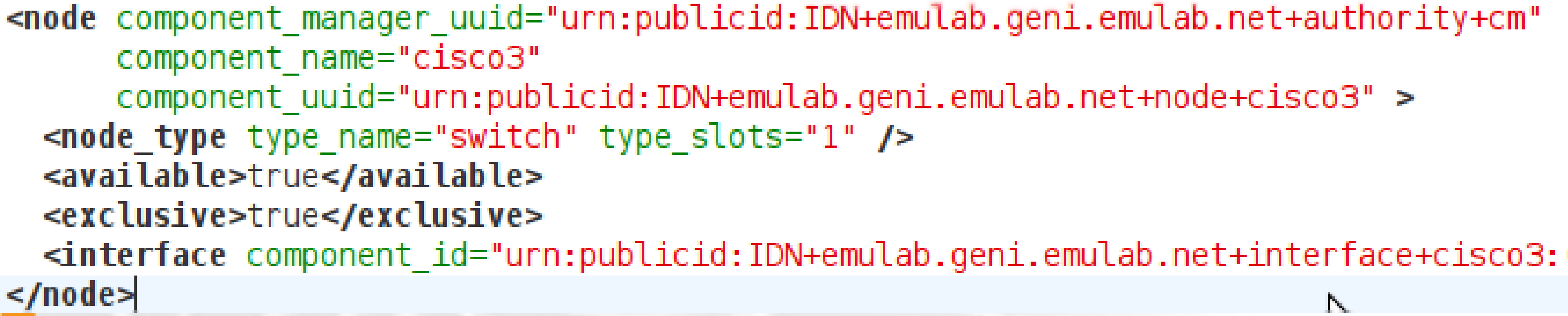
- Converting custom RSpecs to be GENI format RSpec (aka PGV2) extensions
  - Find similarities – use the PGV2 tags
  - Find differences – these are new tags for your extension
  - Write an extension schema – and likely 1 per type (ad/request/manifest)
  - Validate the schema against some sample RSpecs
  - Modify your AM code to add PGV2 support
  - Post the schema

RSpec Extension How-To: Step 1

- Find similarities from native to PGV2
  - For all of ad, request, manifest
  - If there’s a mismatch, wait for step 2 (next slide)
  - OpenFlow example



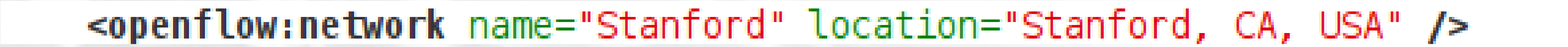
```
<rspec type="openflow" version="2">
    <network name="Stanford" location="Stanford, CA, USA">
        <switches>
            <switch urn="urn:publicid:IDN+openflow:stanford+switch:0">
                <port urn="urn:publicid:IDN+openflow:stanford+switch:0+port:0" />
                <port urn="urn:publicid:IDN+openflow:stanford+switch:0+port:1" />
            </switch>
```



```
<node component_manager_uuid="urn:publicid:IDN+emulab.geni.emulab.net+authority+cm"
      component_name="cisco3"
      component_uuid="urn:publicid:IDN+emulab.geni.emulab.net+node+cisco3" >
  <node_type type_name="switch" type_slots="1" />
  <available>true</available>
  <exclusive>true</exclusive>
  <interface component_id="urn:publicid:IDN+emulab.geni.emulab.net+interface+cisco3:
</node>
```

RSpec Extension How-To: Step 2

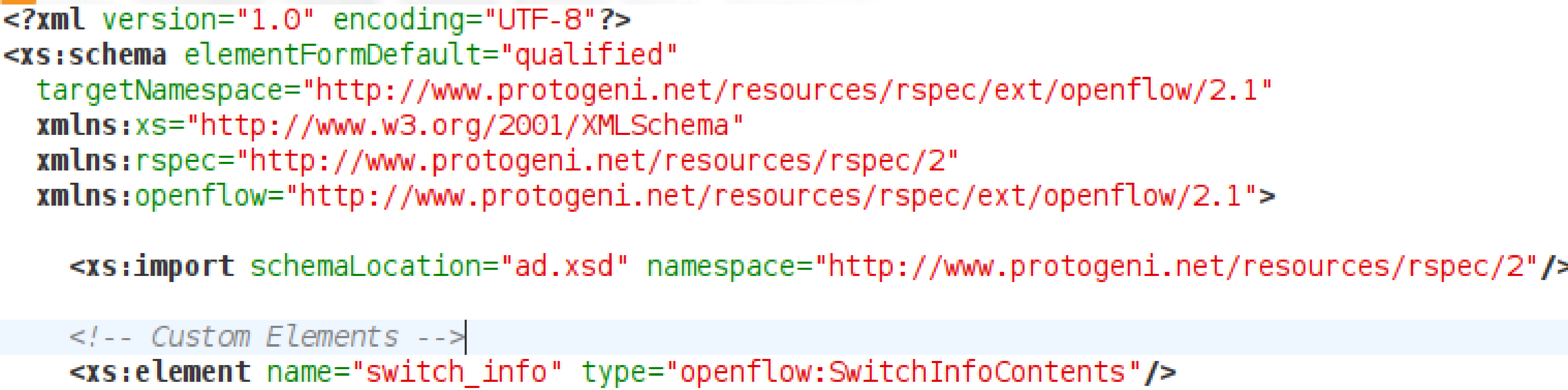
- Find Differences
  - Create new tags for concepts that don't fit
  - For example OpenFlow flowspace, network name
    - OpenFlow flowspace is a brand-new concept
    - No PG top-level 'network'. Add one.
  - Use that to build the schema. In the new extension namespace:



```
<openflow:network name="Stanford" location="Stanford, CA, USA" />
```

RSpec Extension How-To: Step 3

- Write an Extension
  - Holds the new tags
  - 1 for each of ad/request/manifest
  - Validate by creating sample RSpecs,
    - use ‘rspeclint’ tool from ProtoGENI to validate against multiple schemas



```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema elementFormDefault="qualified"
  targetNamespace="http://www.protogeni.net/resources/rspec/ext/openflow/2.1"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:rspec="http://www.protogeni.net/resources/rspec/2"
  xmlns:openflow="http://www.protogeni.net/resources/rspec/ext/openflow/2.1">

    <xs:import schemaLocation="ad.xsd" namespace="http://www.protogeni.net/resources/rspec/2"/>

    <!-- Custom Elements -->
    <xs:element name="switch_info" type="openflow:SwitchInfoContents"/>
```

RSpec Extension How-To: Step 4

- Add AM support
  - Code can inspect RSpec namespaces (or rspec_version option to ListResources) to determine desired RSpec format
  - Dynamically translate internal objects into correct format
- Publish GENI RSpec support
  - Be sure to return correct options in GetVersion
- Publish new extension schema

Status: Tool Support

- Omni 1.3
  - -t option to request a particular RSpec format
- Flack
  - Uses the new AM API options to request PGV2 at PlanetLab
  - Uses PGV2 where supported

Future Work

- More ontologies
- Complete PGV2 roll-out
- More translators as needed
- Adopt AM API revisions
- Maintain RSpec ontologies and schemas
  - Move to geni.net?

Goals for Today

- Adopt Network Ontology as published at https://geni-orca.renci.org/trac/wiki/NDL-OWL
- Adopt? Compute Ontology as published at http://groups.geni.net/geni/wiki/GEC11RSpec
- Discuss wireless, storage, other ontologies
- Discuss AM API revisions and GENI RSpec adoption

Agenda

- Introduction and GENI RSpecs roll-out – Aaron Helsinger
- RSpec Converter at Orca – Ilia Baldine
- Network and Compute Ontologies – Ilia Baldine
- Invited Viewpoints on ontologies
  - Max Ott (compute resources, wireless resources, …)
  - Mike Zink (storage resources)
  - Hongwei Zhang (wireless resources – LENS)
- Discussion: Ontologies
- Conclusion – Aaron Helsinger