GENI Aggregate Manager API

Tom Mitchell

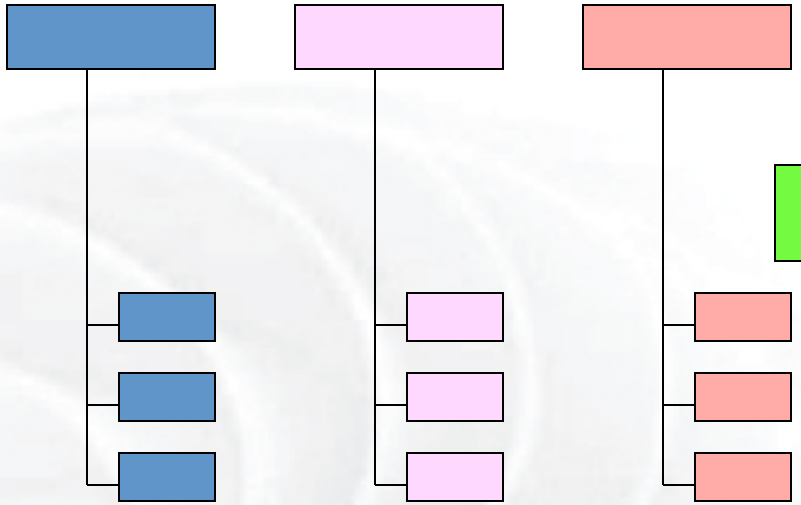
July 21, 2010

Sponsored by the National Science Foundation

Control framework interoperability
via the GENI Aggregate Manager API 1.0

Spiral 1: tightly coupled

Control frameworks / Clearinghouses

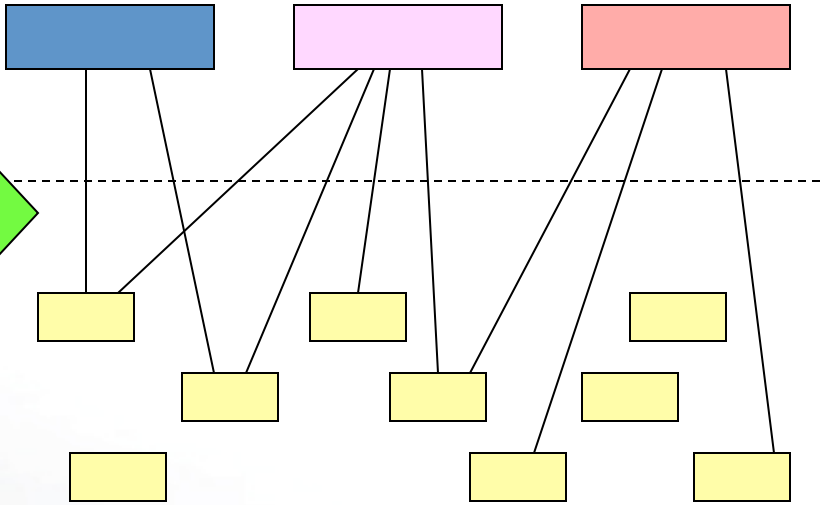


Aggregates

Each aggregate was tightly linked to exactly one cluster and control framework

Now: open & interoperable

Control frameworks / Clearinghouses



Aggregates

"GENI AM API" lets you mix and match aggregates with control frameworks

PlanetLab, ProtoGENI, OpenFlow are now interoperable

Sponsored by the National Science Foundation

Very recent news

- The API is implemented and works
- **PlanetLab**, **ProtoGENI**, and **OpenFlow** are now interoperable (shown in last night's demos)
  - OK, "almost totally interoperable" – still some details to fix
- **ORBIT** and **ORCA** will soon join in, we expect
- This has major importance for all of us
  - API lets you mix and match aggregates with control frameworks
- More info at the GENI Wiki (http://www.geni.net/)

Relationship to SFA

- SFA = Slice-based Federation Architecture
  - SFA 2.0 document is actively being hammered out by Larry Peterson, Rob Ricci, Jeff Chase, Max Ott, and Steve Schwab
  - Thus represents broad agreement among PIs of all GENI control frameworks
- GENI AM API = an implementation of one key interface in the SFA 2.0 draft
- API will probably track SFA as document evolves

PlanetLab API Status

- Code committed in trunk
- Upgrade option for MyPLC installations
  - Instructions on GENI Wiki
- PLC rollout expected, requires coordination
  - Significant change to certificates and credentials

ProtoGENI API Status

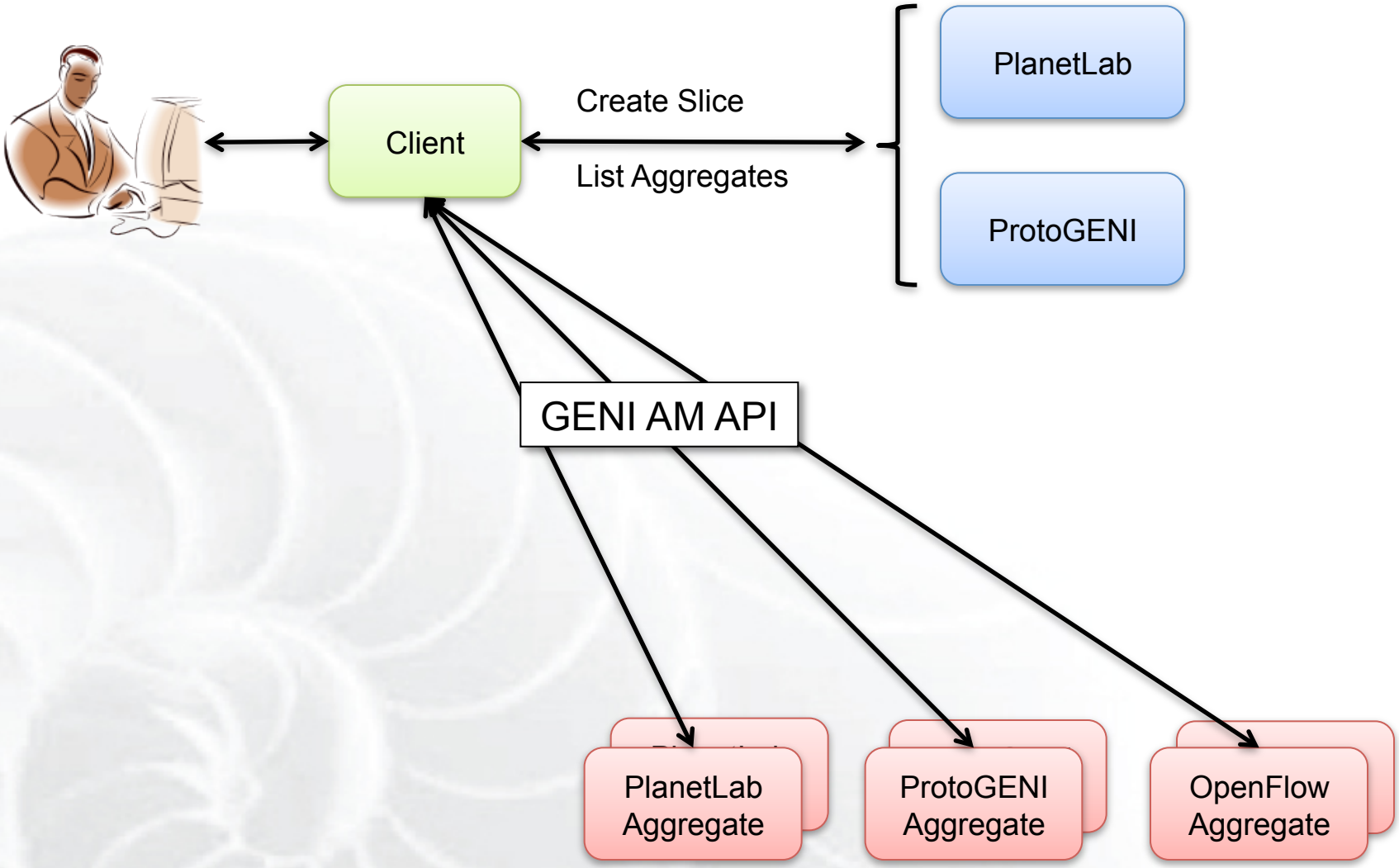
- Running in Utah
- Part of emulab-stable July 14, 2010
- Available now to ProtoGENI experimenters
- Upgraded MyPLC installations can share certificates with Utah for access
  - Ask us for help

OpenFlow API Status

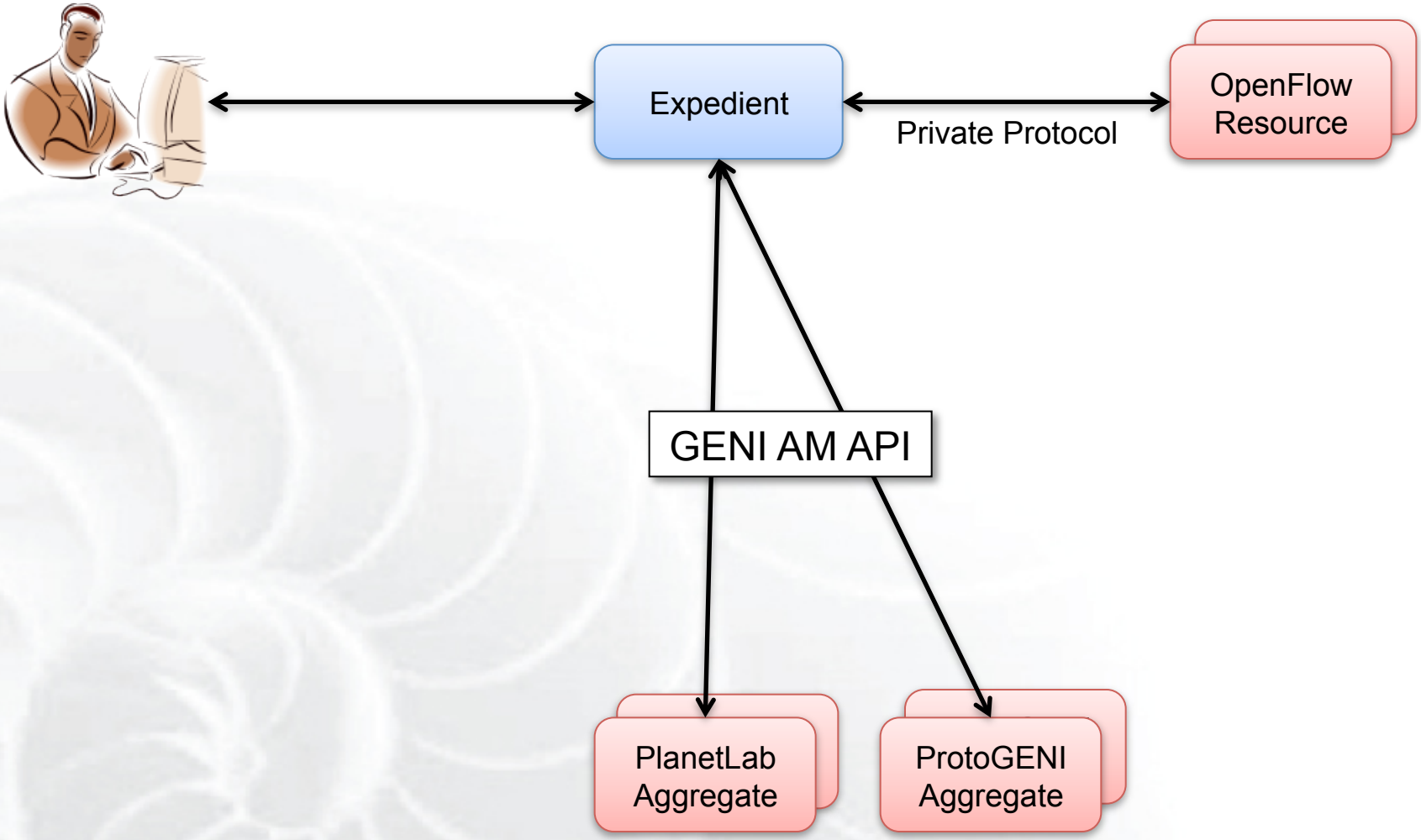
- Implemented as an Expedient Plugin
- Interoperability demonstrated last night
- Alpha testing now; full deployment by GEC 9
- More info:

  http://yuba.stanford.edu/~jnaous/expedient/

Using the GENI AM API with PlanetLab and ProtoGENI



OpenFlow Control Framework



GENI Aggregate Manager Operations

```
GetVersion()

ListResources(string credentials[], struct options)

CreateSliver(string slice_urn, string credentials[],
             string rspec, struct users[])

DeleteSliver(string slice_urn, string credentials[])

SliverStatus(string slice_urn, string credentials[])

RenewSliver(string slice_urn, string credentials[],
            string expiration_time)

Shutdown(string slice_urn, string credentials[])
```

Technical Details

- Draws on existing PlanetLab and ProtoGENI underpinnings

- API underpinnings
  - SSL with client authentication
    - Both server and client are identified by a certificate
  - XML-RPC
    - Programming language independent communication
  - URNs identify users, slices, aggregates, etc.
  - Certificates identify entities
  - Credentials authorize entities

URNs

- General Form:

  urn:publicid:IDN+*toplevelauthority*+*resource-type*+*resource-name*

- Examples

  urn:publicid:IDN+plc:gpo+user+jane
  urn:publicid:IDN+openflow:stanford+slice+myslice
  urn:publicid:IDN+emulab.net+authority+sa

- More Info

  http://www.protogeni.net/trac/protogeni/wiki/URNs

Certificates

- Standard X.509 certificates
- Issued by Control Framework or Clearinghouse
- Requirements:
  - URN in Subject Alt Name

```
                cd:bd:46:68:9f:cd:f7:a0:7b:1d:b6:e7:9b:b9:9a:
                30:66:b6:29:b7:9b:8c:31:8f
            Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Alternative Name:
                URI:urn:publicid:IDN+geni.net:gpo:gcf+user+alice
    Signature Algorithm: md5WithRSAEncryption
        60:cd:f2:ce:6b:14:7b:5d:2d:d3:ba:4a:32:d6:94:46:ac:ef:
        ce:95:28:21:d6:4e:07:7a:7f:12:1e:3d:8c:43:9d:34:fd:dc:
        a1:13:69:9c:e3:ff:85:bd:cd:d7:8f:66:dc:b9:d9:89:3c:7e:
```

Credentials

- ProtoGENI credentials
- W3C XML Signature Syntax and Processing

- Contains:
  - **Owner** - the actor
  - **Target** - the object to be acted upon
  - **Privileges** - what the owner can do to (or with) target

- Credentials can be delegated

Sample Control Framework

- gcf: A Sample GENI Control Framework
  - Provides a working example for testing and comparison
  - Includes a reference implementation of certificates, credentials, and credential verification
  - Documentation by example
  - gch: A simple GENI Clearinghouse
    - Not for production use
    - No bells or whistles
  - gam: A simple GENI Aggregate Manager
    - Manages fake resources
    - All state is volatile

Available for download at the GENI Wiki (http://www.geni.net/)

Sample API Client

- omni: A GENI AM API Client
  - Communicates with different control frameworks in their native tongue
  - Interacts with multiple compliant aggregates
  - Uses native RSPECs
  - Command line client requiring manual work

Available for download at the GENI Wiki (http://www.geni.net/)

Future Topics

There is more to do!

- RSPECs
  - The API is intentionally agnostic to RSPECs
  - This puts extra burden on clients to understand a heterogeneous set
- Stitching
  - The API is intentionally agnostic to stitching
  - The API is intended to allow experimentation with stitching approaches
- Scheduling
  - Time is buried in RSPECs
  - No reservations
  - Anticipated in a future revision

What should you do?

- **Experimenters / demo teams**
  - Between now and GEC 9 . . . use it!
  - Work with Mark Berman to learn how you can use it
  - He will engage software people as needed, to help you
- **Aggregate owners and prototyping teams**
  - Consult with your current control framework leader about when & how to upgrade
- **ORBIT and ORCA cluster members**
  - Don't panic – we haven't forgotten about you!
  - GPO will work with ORBIT and ORCA for compatibility in coming months
- **Community**
  - Download, read, comment on and use the API

Questions?

Comments?