

Robert Ricci

GEC 10

March 16, 2011

# ProtoGENI Stitching Thoughts



# Philosophy

- Avoid global locks
- Hide as much from clients as possible  
(negotiation of parameters, ordering)
- Parties involved know about the technology
- All resource allocation is explicit and visible
  
- Make it easy to write clients

# Understanding Interconnections

- Links are the objects that cross aggregates
  - Links associated with some set (usually 2) of interfaces
- Our RSpecs have “External References”
- Think like a variable declaration (extern in C)
- Issue: keeping endpoint advertisements consistent
  
- We have a strawman extension for Tom’s common stitching schema

# Current Implementation

- Support cross-aggregate VLANs using ION and (almost) native layer 2
- We don't currently have a stitching topology service
  - We can build one, though we don't want to make it mandatory
  - Users can do this themselves
- We have a path computation function (SES)
  - Computes path only, does no setup
- Stitching workflow function
  - This lives inside the AMs
  - Can outsource to others if they want (eg. backbone AMs)
- Every AM involved is contacted explicitly
  - Can find this from SES output
  - User knows what resources they have reserved
- We don't support loose requests - stitching points are explicitly identified
- New API call at AM to facilitate stitching