

# NSI Implementation TF

---

John MacAuley, SURFnet

Inder Monga, ESnet

GLIF 2013



NSI 2.0 – Community effort  
Matrix,30 Biopolis



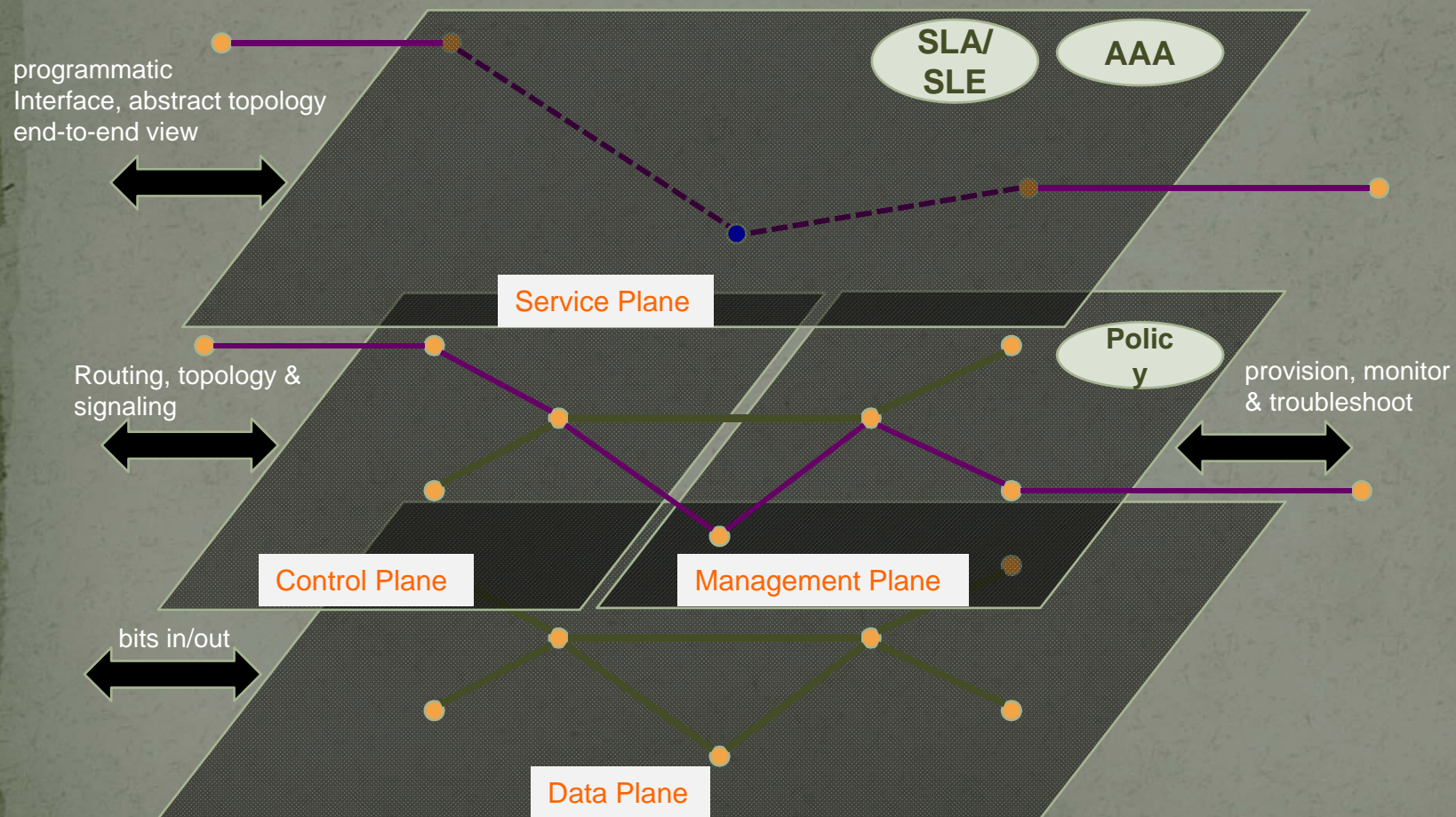


## NSI Implementation Figuring out the issues

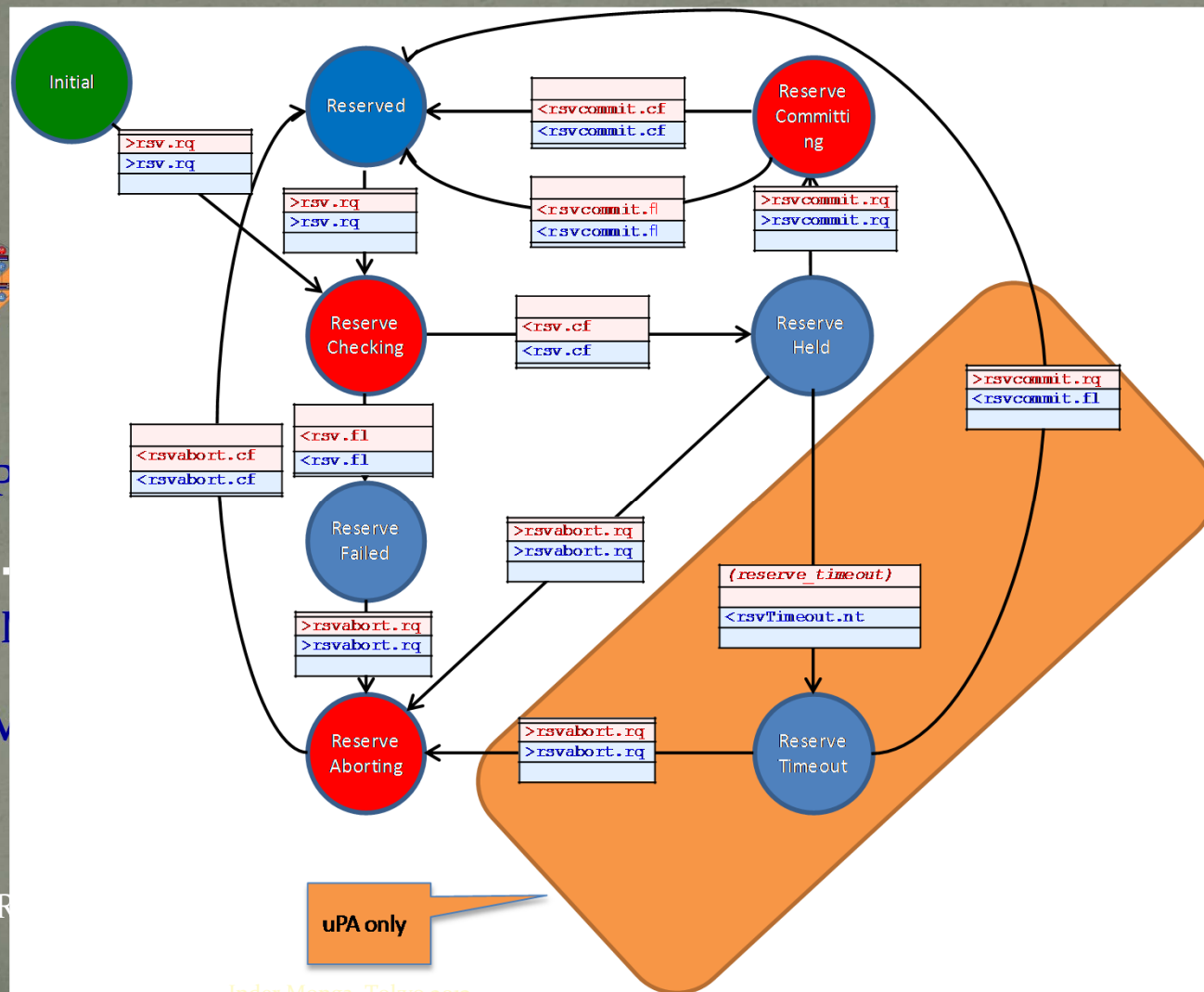
# NSI Overview



# Introducing the Service Plane Concept



# NSI Protocol Structure



Inder Monga, Tokyo 2015

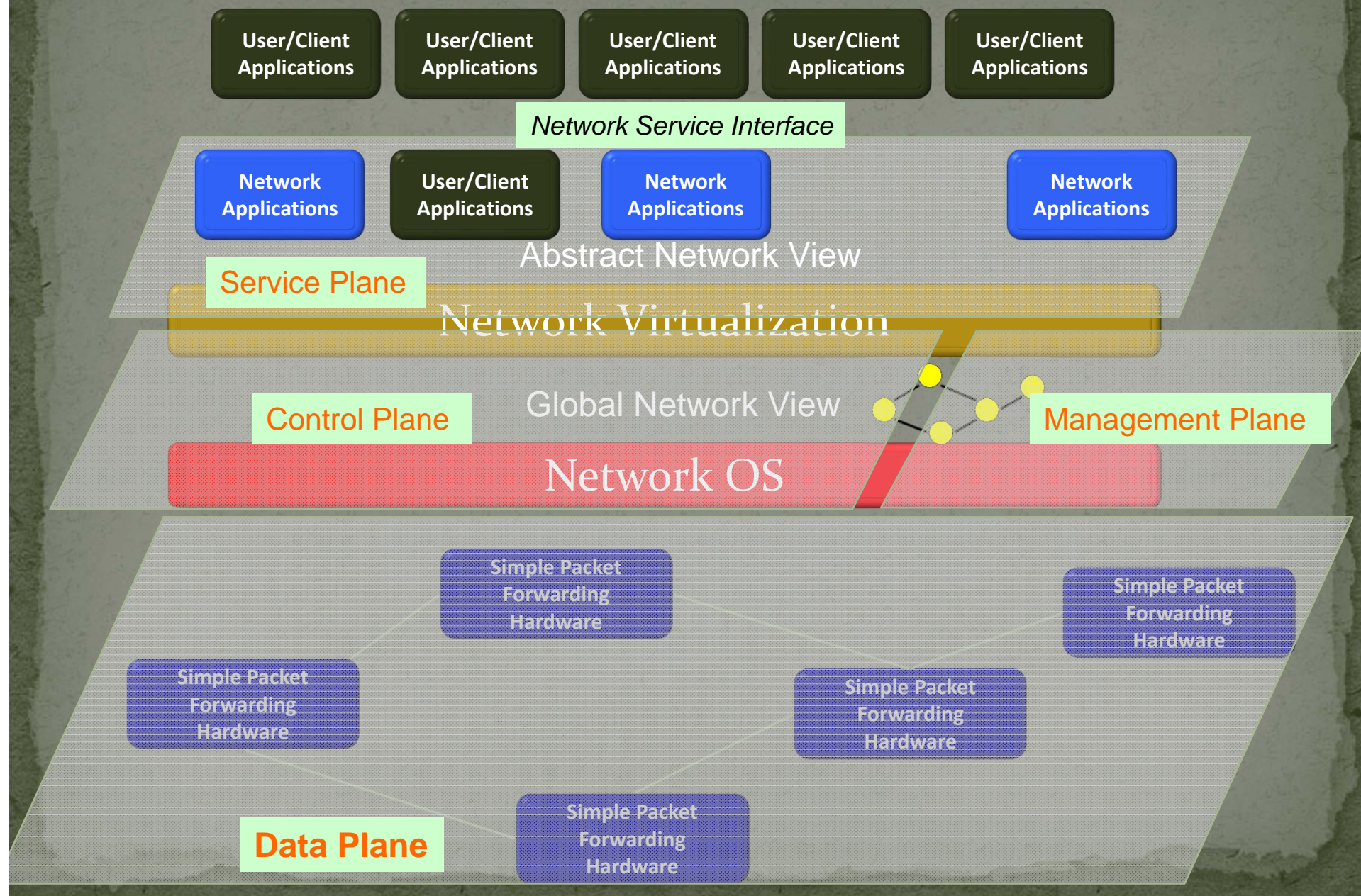


# NSI & SDN: Aligned architecturally

NSI model	Architecture/Function	SDN model
<ol style="list-style-type: none"><li>1. One NSA/network</li><li>2. Tree/Chain model of NSA interaction</li><li>3. b/w NSAs/domains</li><li>4. Resource policies enforced by NRM</li><li>5. Provisioning of end-to-end services</li><li>6. Inherits same challenges</li></ol>	<ol style="list-style-type: none"><li>1. Logically Centralized</li><li>2. Hierarchical/nested support</li><li>3. Trust in control plane</li><li>4. Policy Management central to operation</li><li>5. Control and Management functions</li><li>6. Control plane challenges: Security, partitioning</li></ol>	<ol style="list-style-type: none"><li>1. One logical Controller</li><li>2. Multiple hierarchical controller model (tree)</li><li>3. Required b/w controllers</li><li>4. Flowvisor, AM, other policy mechanisms</li><li>5. Provisioning of end-to-end data flows</li><li>6. Inherits same challenges</li></ol>



# NSI: Northbound API for SDN (for Multi-domain)?





# NSI Implementation TF

1. Security: Authorization and Authentication
2. Topology Distribution
3. Service Decoupling

# Authorization and Authentication

- Agreement within the NSI working group to use SSL mutual authentication (digital certificates) between NSA servers.
- Requires manual provisioning of certificates on the client and server sides.
- Current authorization model is based on transient trust (I will trust who you trust).





# Authentication and Authorization: Key Operational Issues

- Issues with complexity, cost, and logistics of managing client-side certificates for a large number of NSI clients.
- Requirement for client friendly authentication mechanisms such as HTTP BASIC and OAuth2.
- Further research is required on the topic of end user trust across domains.



# Topology Distribution (alpha!)

## Topology Service in nutshell

- ✍ NML/ NSI compatible TS service.
- ✍ Simple RESTish style! Over HTTP(s).
- ✍ Supports simple PUB/ SUB model.
- ✍ Easy to build clients; cURL just works fine.
- ✍ Has the hooks for feature extensibility (things I couldn't do due time constraints)

Ahmed El-Hassany <a.hassany@gmail.com>

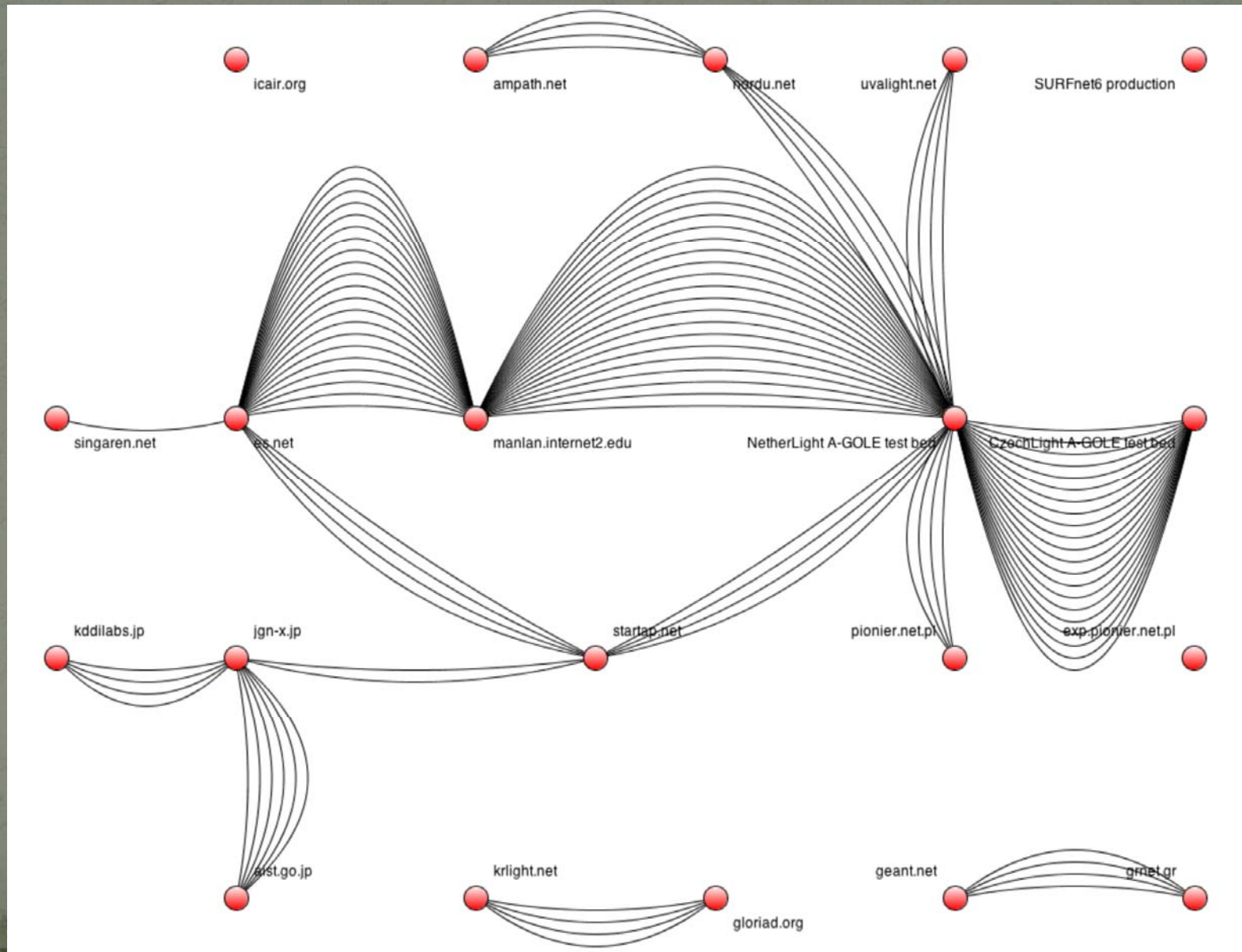


# Topology:

## Key Operational Issues

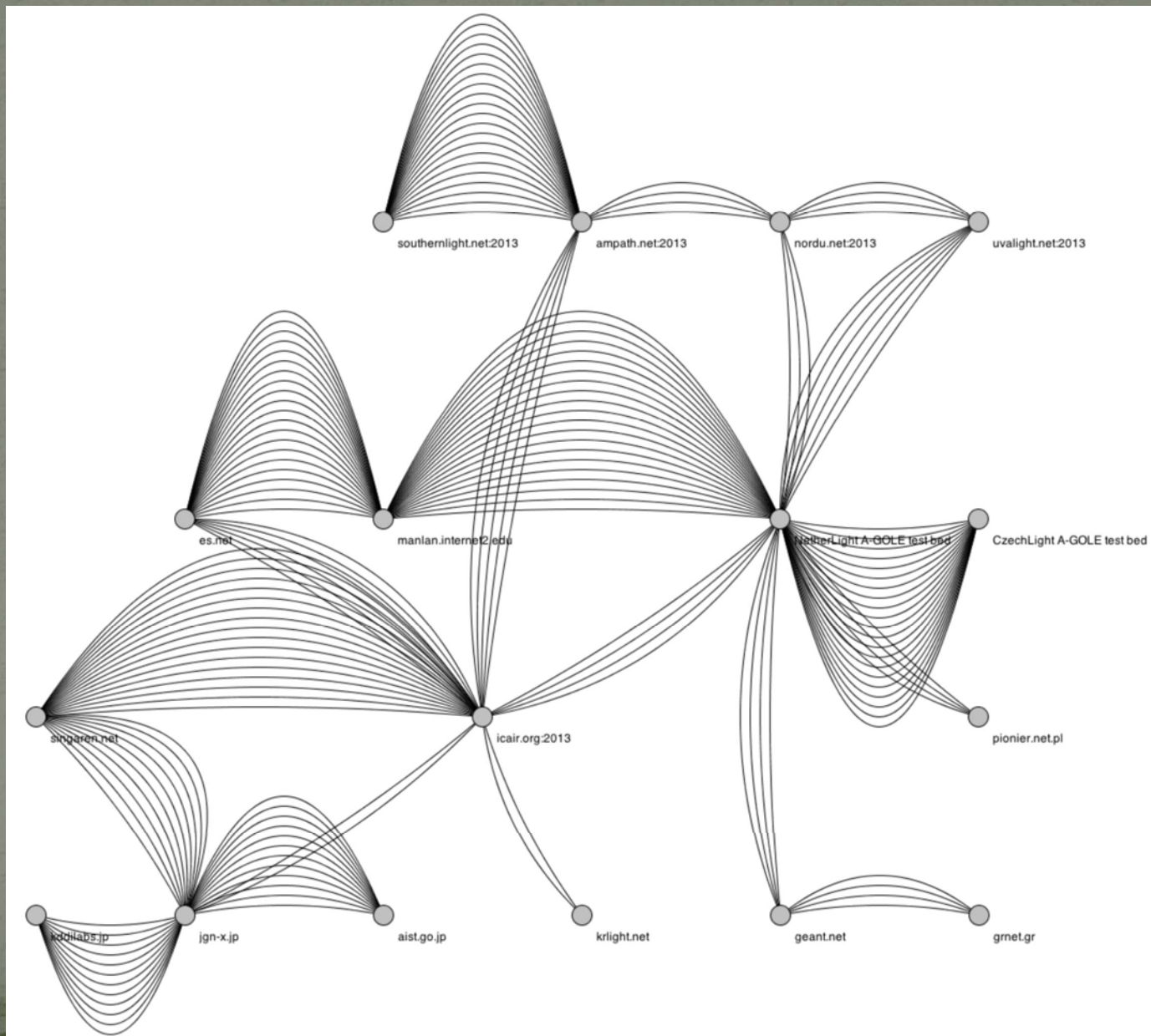
- Large operational overhead associated with maintaining accurate topology files and the interconnection of inter-domain ports.
- Some automatic generation currently performed.
- Root cause analysis needs to be performed on activities leading up to GLIF demo.
- Policies and procedures will need to be defined to help ease the administration overhead.
- Requirements for automated topology discovery mechanisms should be investigated.

# State of Demo Topology (09/29/2013)





# State of Demo Topology (10/03/2013)

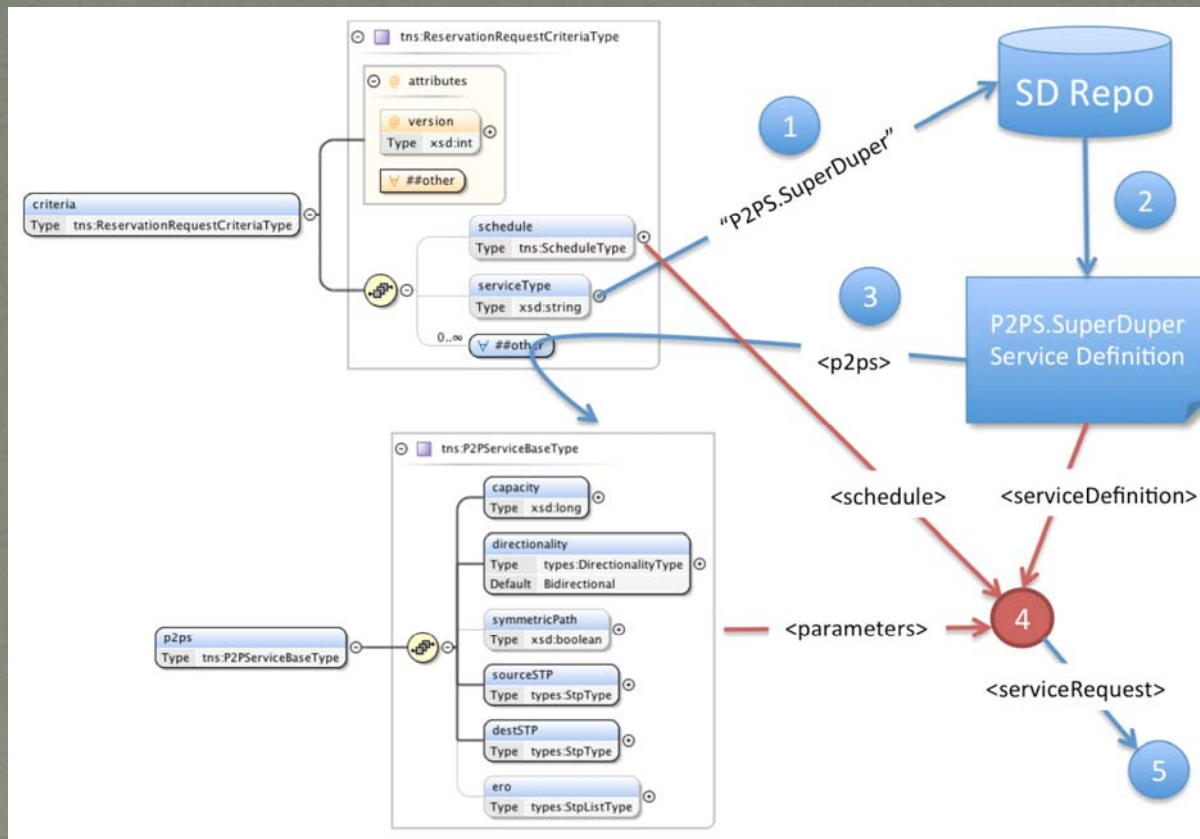


# Service Decoupling

- Removal of service specifics from NSI CS v2.0 protocol
- Original Idea
  - Service Definition = Different Data planes only
- New Idea
  - Service Definition = New services, multiple services in one request
- Three basic service building blocks are now available in NSI CS v2.0
  - Point-to-Point service
  - Ethernet Transport Service
  - Ethernet VLAN Transport Service
  - new services can be defined as needed
    - Multipath service
    - Point-to-Multipoint service
    - Protection service



# Actions when receiving a reserveRequest



## Steps:

1. When reserveRequest arrives extract the *serviceType* value.
2. Fetch the Service Definition corresponding to the *serviceType*.
3. Extract the specific service elements from *criteria* as specified in SD.
4. Use the Service Definition to validate request.
5. Process using both the supplied service parameters and additional information as needed from the Service Definition document.

# Thank you!

