The Network Services Interface

An Overview and Demonstration of:

NSI Framework,

NSI version 2.0 implementations,

GLIF Automated GOLE

Jerry Sobieski
NORDUnet

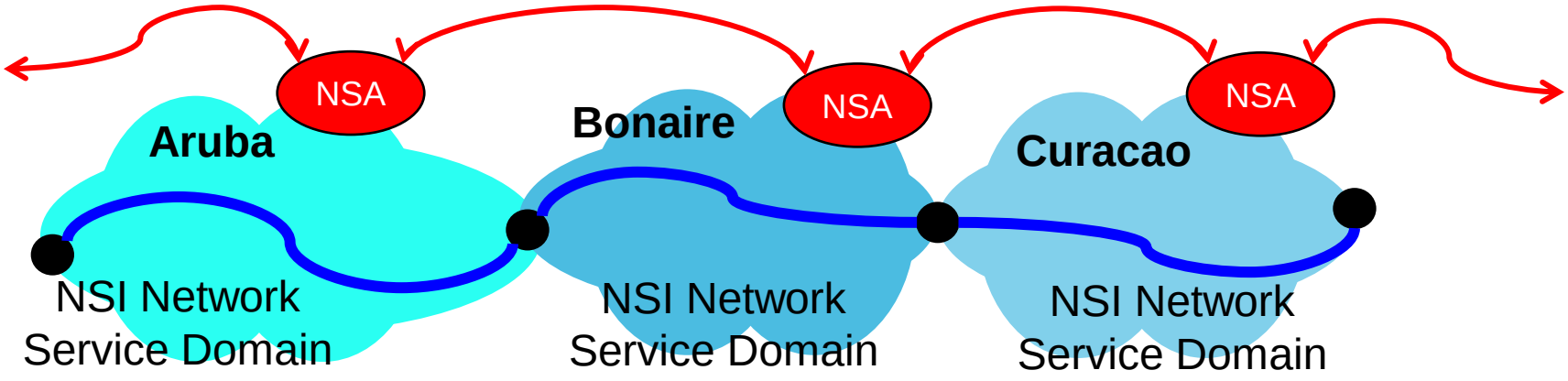
GLIF 2012
Chicago, US

What is NSI?

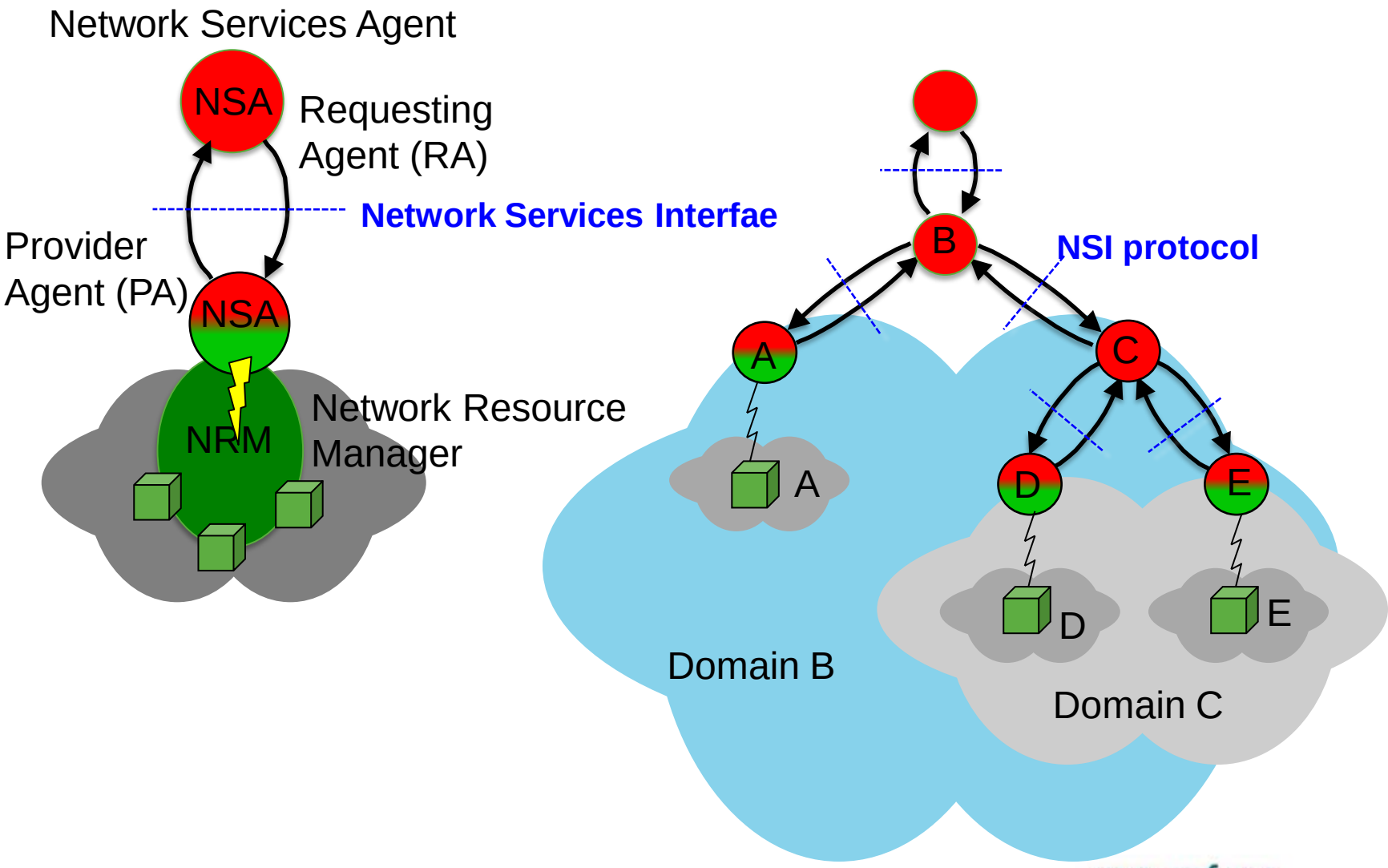
- Network Services Interface – “NSI”
- It is a framework, and a set of services within that framework, that:
  - Deliver predictable deterministic *connectivity* services – i.e. transport provisioning
  - Do so at a global scale
- NSI treats networks as “service” domains rather than collections of switches and cables:
  - It offers an abstract architectural model of the global network service infrastructure
  - It defines a set of services that work together to enable end to end transport provisioning

NSI Framework

- The “NSI Framework” is an OGF standard:
  - It defines an abstract model of the global network service “Topology” over which Connections are established
  - It defines a “Network Service Agent” (NSA) that represents each network service domain
  - It defines a high level protocol model between NSAs to enable inter-domain NSI Services.
  - It defines an abstract model of a network “Connection”

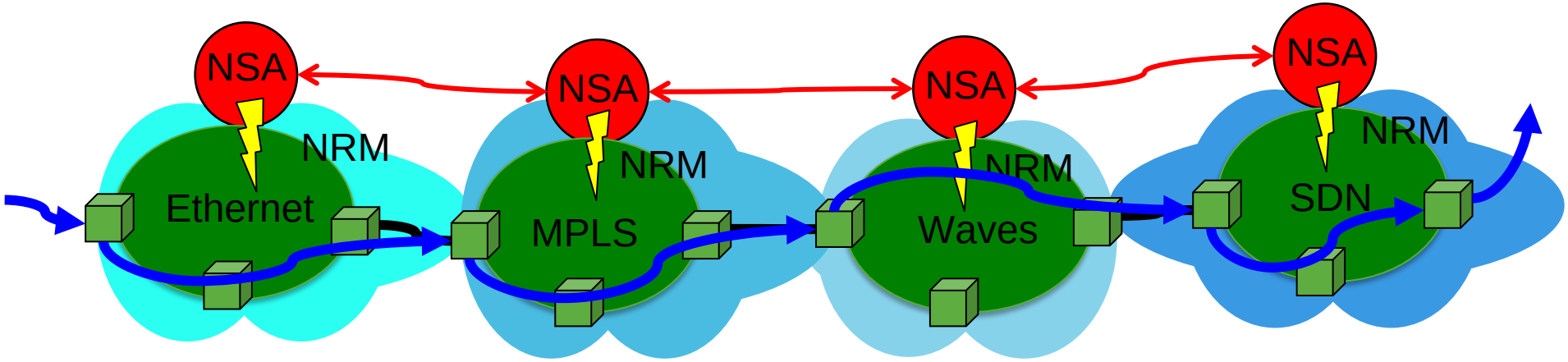


A Basic Overview of NSI Architecture



NSI is “Technology Agnostic”

- The NSI architecture does not specify transport technologies used within each domain
  - The Network Resource Manager handles internal provisioning
- It is therefore highly amenable to multi-layer, multi-domain, multi-service data transport environments -> i.e. modern networks.



NSI is “Secure by design”

- NSI protocol interactions are authenticated and authorized at each service boundary for all inter-domain requests.
- NSI does not expect or require a network to advertise any internal topology information
  - Each domain announces only the structure they wish external parties to perceive
  - NSI uses announced “inter-domain adjacencies” to perform inter-domain path finding
- NSI has been very careful to prevent internal information from leaking out via protocol responsing.

Why NSI? Why now?

- Emerging applications require predictable, deterministic, and dynamic network transport resources
- Provisioning of said resources must be *automated* in order to scale in a global environment,
- Must be *inter-domain* in order to provide global reach
- And they must be *secure* to be globally deployable

- ***We need a single ubiquitous CONSENSUS protocol!***

- Much prior work has been done to develop effective protocols for automated provisioning … But:
  - These protocols have been largely experimental prototype
  - They generally do not address deployment issues such as security, privacy, policy, …
  - ... They are rarely interoperable

  – resulting in limited and isolated islands of functionality.

Why NSI? Why now?

- The NSI Framework and protocol(s) are the product of an ***open*** and ***community-wide*** process within the Open Grid Forum.
- The result is a ***consensus standard*** that:
  - Meets a wide range of needs…
  - Is widely understood and widely supported,
  - And will therefore be widely deployed in networks and widely adopted by applications.

NSI Version 1.0

- The NSI Framework - done
- NSI Connection Service v1.0 - done
  - Basic point-to-point Connections and lifecycle management
  - Initial NSI topology model
  - Basic chain and tree segmentation
  - A State Machine for the protocol
  - Basic protocol:
    - Reserve, Provision, Release, Terminate, Query
  - Book ahead scheduling
- Testing in the GLIF Automated GOLE
  - Continues to run today:
  - **~ 2.5 x 10^5 NSI Connections in 2012 (!!)**

NSI Version 2.0

- Add “Modify()” primitive
- New state machine:
  - Split into three separate state trails: Reservation State, Connection State, Hardware State.
- Incorporate NML topology specifications
- Add “any point” capability to Reserve() req.
- Add type-value pairs for STP specification
- Incorporate EROs
- Unidirectional Connections and STPs
  - Resolved pathfinding ambiguity
- Discovery service

NSI v2.0 Announced Implementations:

- ***DRAC*** – SURFnet (Amsterdam, NL)
- ***G-LAMBDA-A*** - AIST (Tsukuba, JP)
- ***G-LAMBDA***-K – KDDI Labs (Fujimino, JP)
- ***DynamicKL*** – KISTI (Daejeon, KR)
- ***AutoBAHN*** – GEANT (Poznan, PL)
- ***OpenNSA*** – NORDUnet (Copenhagen, DK)
- ***OSCARS*** – ESnet (Berkeley, US)

NSI v2.0 Development Timeline

- "First Look"
  - GLIF 2012 – Chicago – Oct 2012
  - Very early versions, specification still wet …
- "Ramp Up"
  - Supercomputing 2012 - Salt Lake City – Nov '12
  - Deployed over Automated GOLE and beta testing
- "Bonzai Pipeline"
  - Techs in Paradise – Honolulu – Jan 2013
  - Full scale conformance testing – pre-production
- Production (!)
  - March 2013 – NORDUnet, SL?, GN3+?

NSI Futures – V3+

- Still adding new capabilities to CS
  - Point to Multipoint, AnyCasting
  - “Switching Points”
- Topology Service
  - Bootstrapping, discovery and update
- More sophisticated path finding
- Conformance suite
- Verification/Fault localization Service
- SDN integration
  - Open Transport Switch adaptations for NSI Networks
  - STPs currently represent simple flow spaces – reconcile OpenFLow and STPs (and NML)

GLIF Demo Presentations:

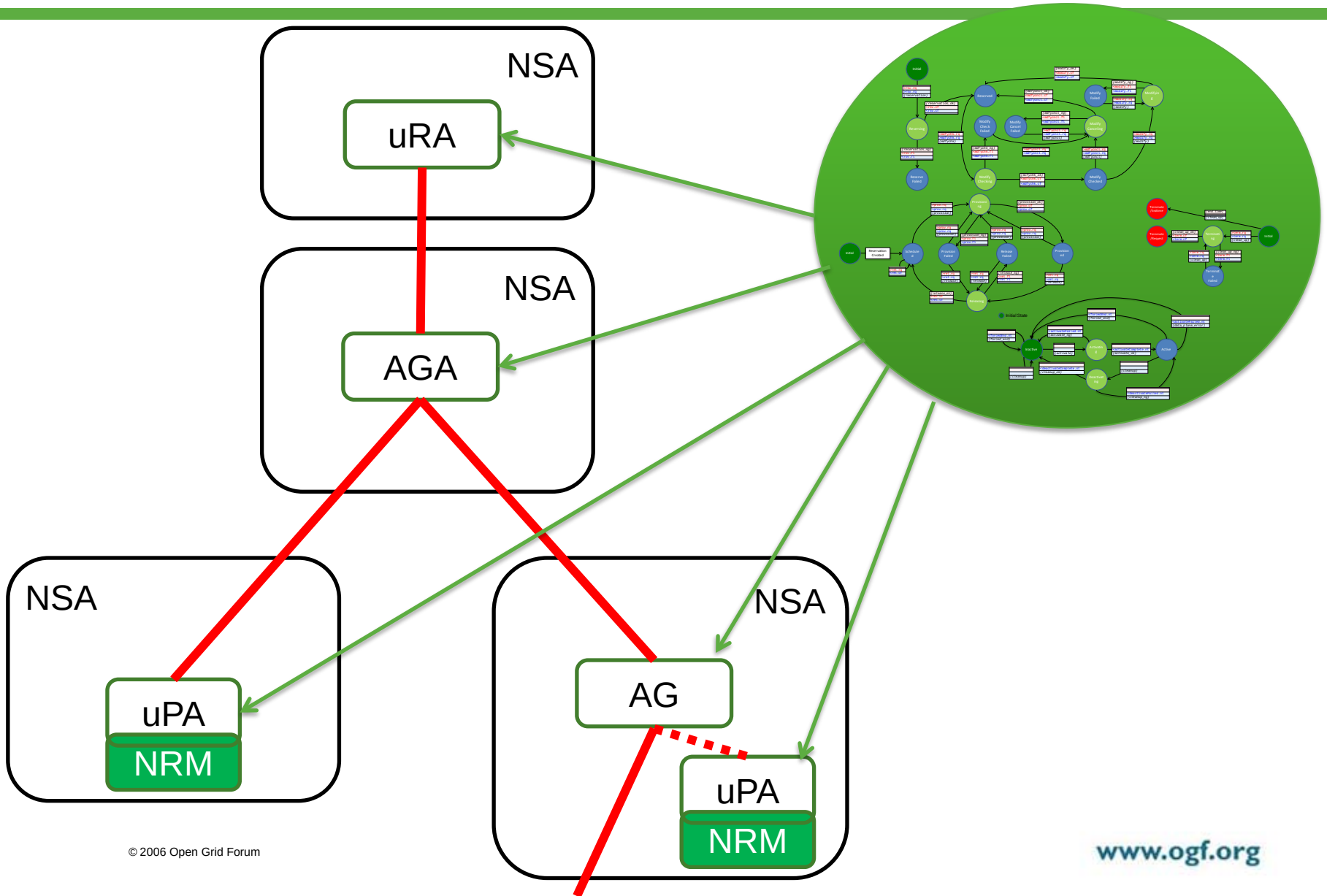
- NSI v2.0 Protocol Evolution
  - State machine, primitive extensions
- NSI v2 Topology NML Intergration
  - Standard interoperable topology
- NSI v2 Migration and deployment
  - Version discovery
  - Security profiles and authorization
- AutoGOLE (v1) Status Monitor

NSI V2 – Protocol Evolution

NSIv2 State Machine

- Modify() command
  - Important “minor” feature required in V2
  - Turns out to have myriad complexities…
- The State Machine is a virtual map that describes how the protocol events progress a connection through its life cycle
  - Modify() caused SM explosion to track state of morphing connection…
  - Fix provisioning messages to address notification issue
  - Resolve “reservation” vs “data plane” vs “protocol” issues…

NSA Roles



OpenGridForum

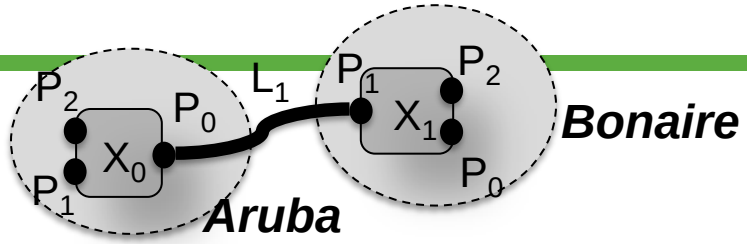
OPEN FORUM | OPEN STANDARDS

NSI V2 – NML Topology Integration

NSI v2 Topology descriptions

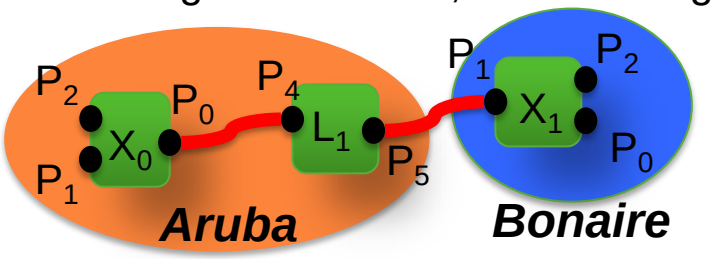
- In order to choose a path across the world we need a map of the network infrastructure – the Topology
  - The network service domains
  - And their juxtaposition to one another
    - **The NSI Topology model**
  - We can complement this basic information with increasing detail…
    - **The NML Topology Ontology**
- The global topology is too large and too detailed to maintain centrally
  - We want a distributed process for describing and announcing topology
- **OGF Network Markup Language (NML)** is a sister standard to NSI for describing network topologies

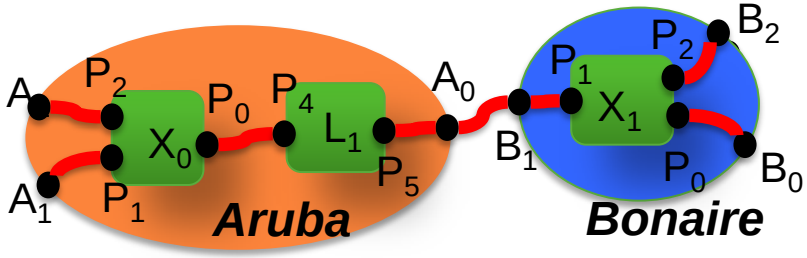
Basic NSI Topology Model

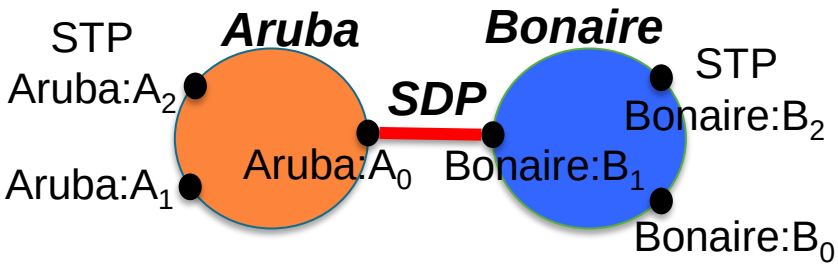


Conventional physical infrastructure – Networks, switches, ports, and links

The physical topology is translated to a derivative "resource graph" consisting of resources, and stitching relations







The NSI model assigns ownership of all physical components to one network or the other.

Technology agnostic inter-domain Service Termination Points ("STP"s) are defined and mapped logically to internal physical components. External relations are NSI Service Demarcation Points ("SDP"s).

By hiding all internal structure and only exposing inter-domain STPs and their peering SDP relations, we arrive at the basic NSI Topology Model of networks, STPs, and the SDPs that indicate inter-domain adjacency.

NSI NML snippet

The following is a snippet of NML describing the UvALight network:

uvalight:topology a nml:Topology; NSI Network
nml:hasInboundPort **uvalight:netherlight-uvalight** ;
nml:hasOutboundPort **uvalight:uvalight-netherlight**;
nsi:managedBy **uvalight:nsa** .

uvalight:nsa a nsi:NSA, NSA
nsi:csProviderEndpoint
”http://nsa.uvalight.nl:9080/NSI/services/ConnectionService/ ”
nsi:managing **uvalight:topology** .

STP
uvalight:netherlight-uvalight a nml:PortGroup, owl:NamedIndividual;
SDP
nml:isAlias <**urn:ogf:network:netherlight:2012:netherlight-uvalight**>;
nmleth:vlans "**1780-1783**" .

NSI Type-Value pairs

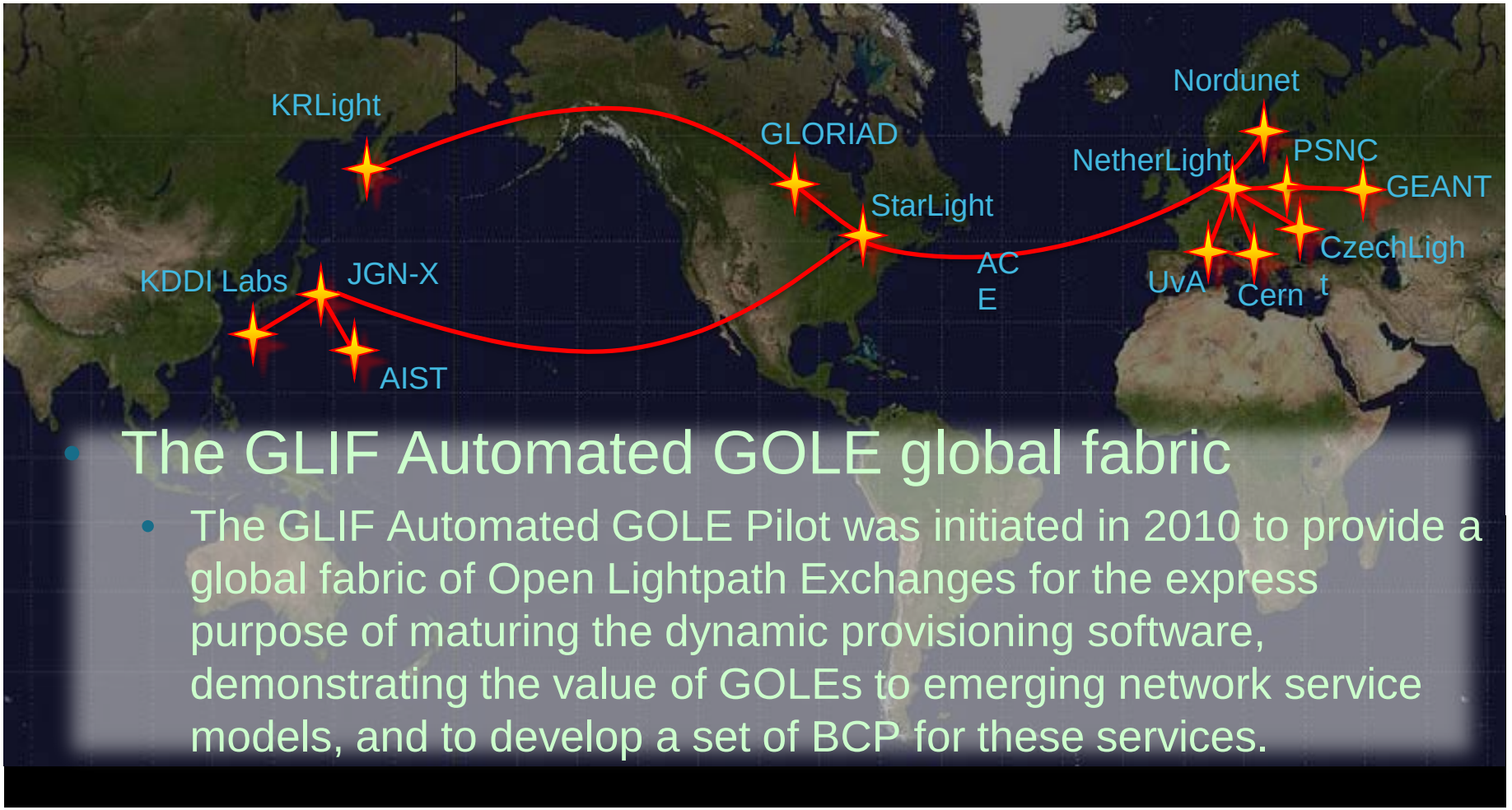
- Type Value pairs allow a mix of topological constraints to be specified in the NSI messages.

- `ReservationRequest:`
  - `SourceSTP -> Topology="northernlight", port="cph-ams", ethvlan=1780`
  - `DestSTP -> Topology="starLight", port="evl-80"`
  - `Bandwidth := 10000000000`

- Type Value pairs allow the Requester to specify a set of endpoints that are acceptable for the request.

The GLIF Automated GOLE Pilot Project

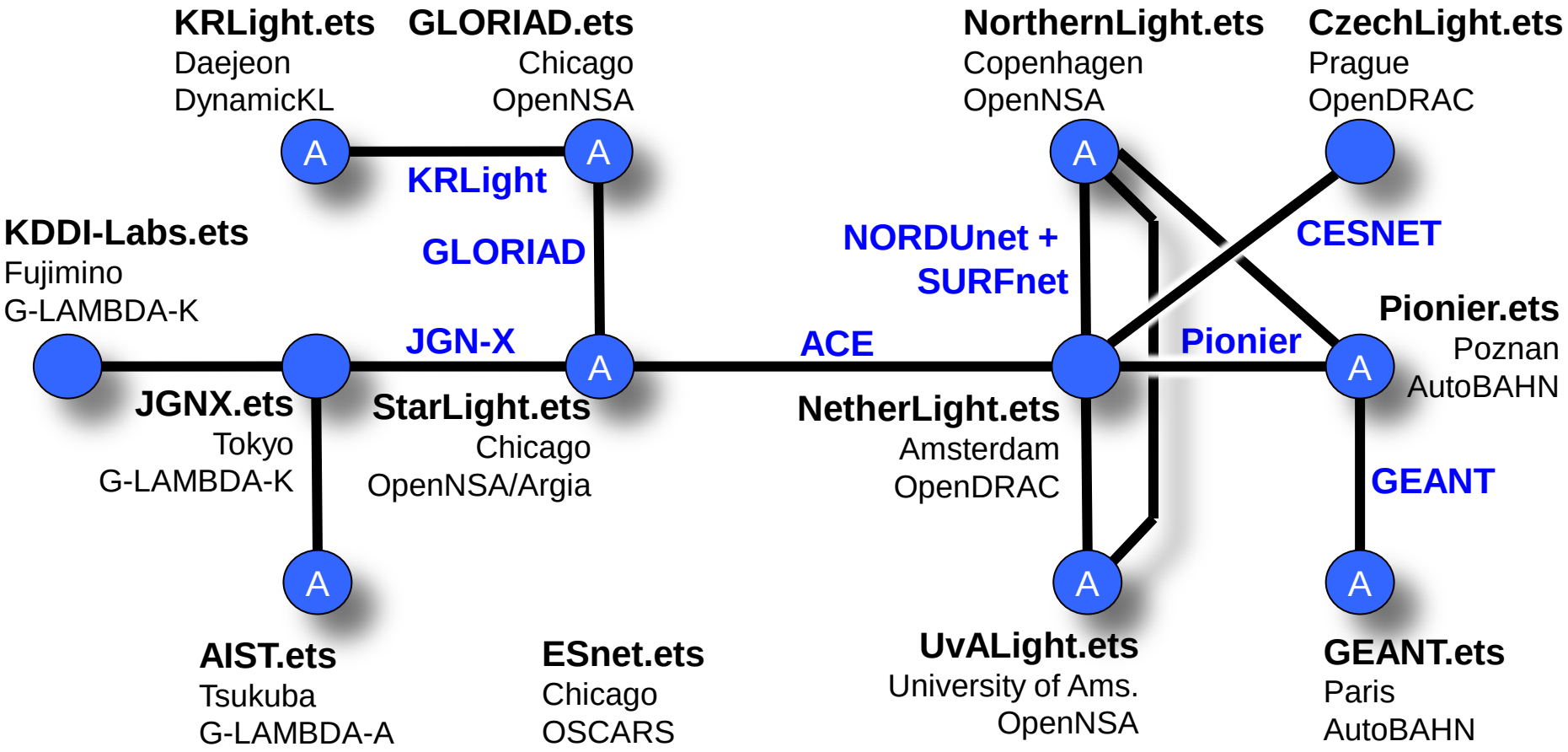
The Automated GOLE Fabric



- The GLIF Automated GOLE global fabric
  - The GLIF Automated GOLE Pilot was initiated in 2010 to provide a global fabric of Open Lightpath Exchanges for the express purpose of maturing the dynamic provisioning software, demonstrating the value of GOLEs to emerging network service models, and to develop a set of BCP for these services.

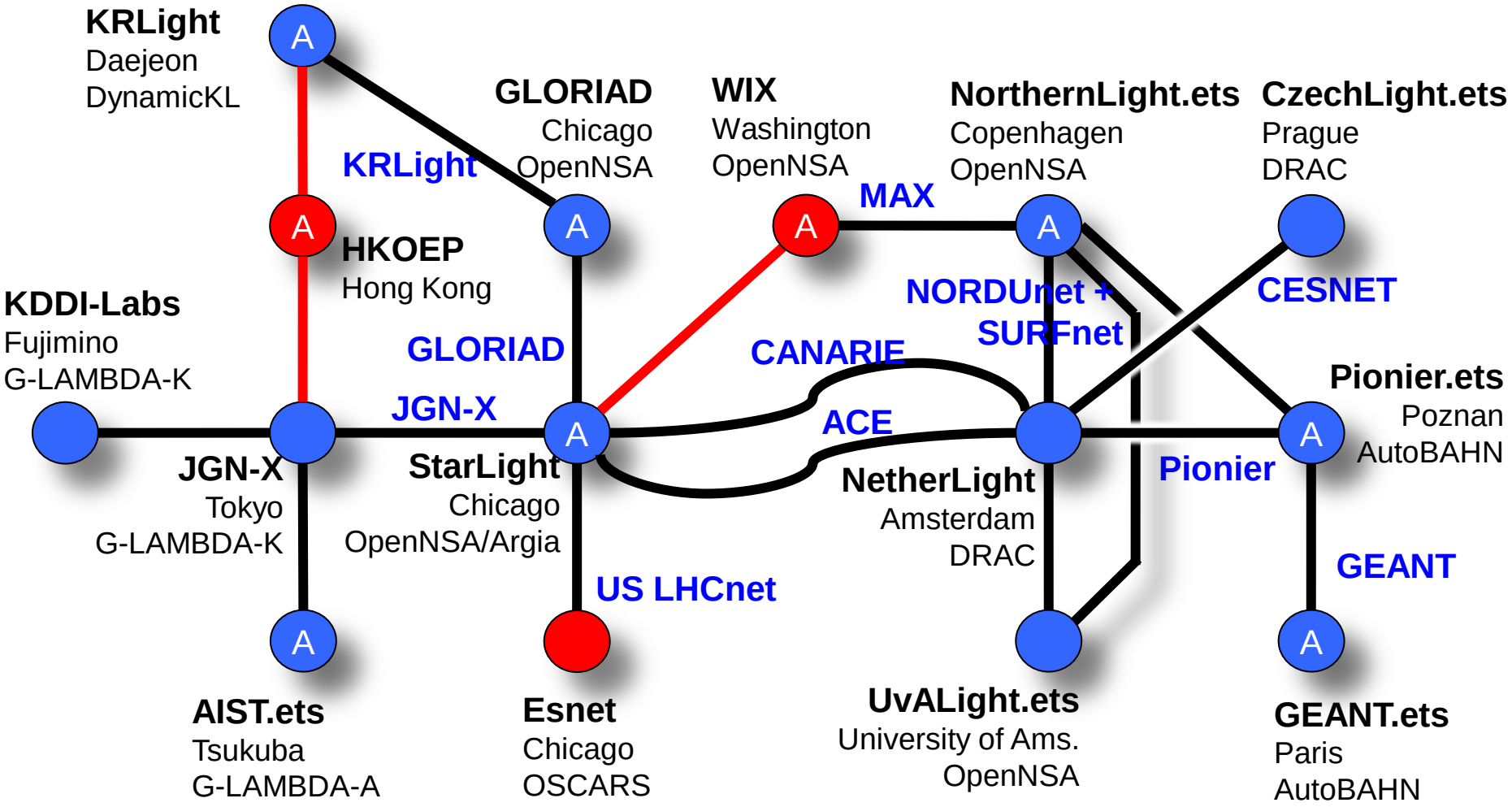
AUTOMATED GOLE + NSI
DEMO NETWORK 2012-10

OpenGridForum



AUTOMATED GOLE + NSI
DEMO NETWORK 2012-10

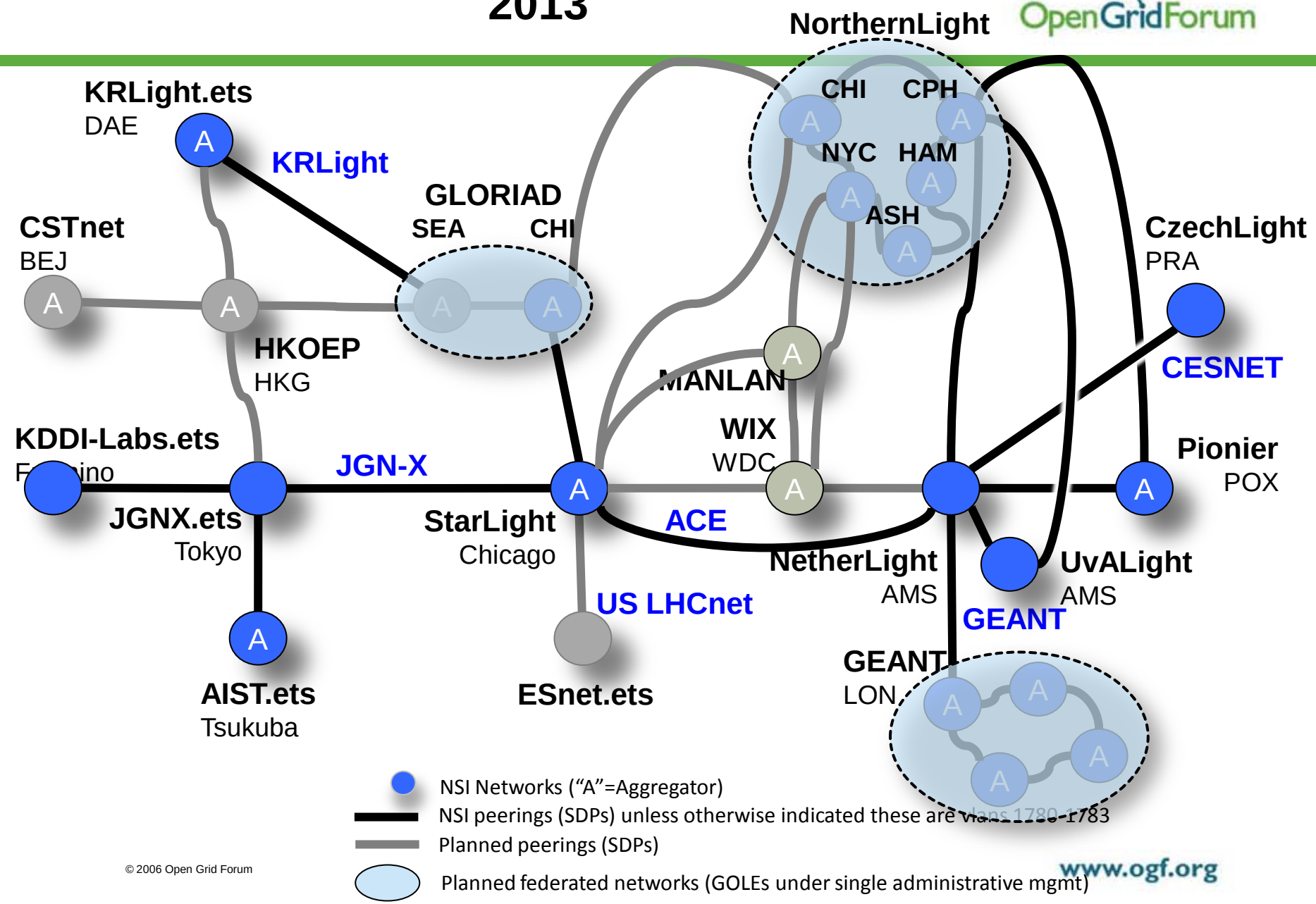
OpenGridForum



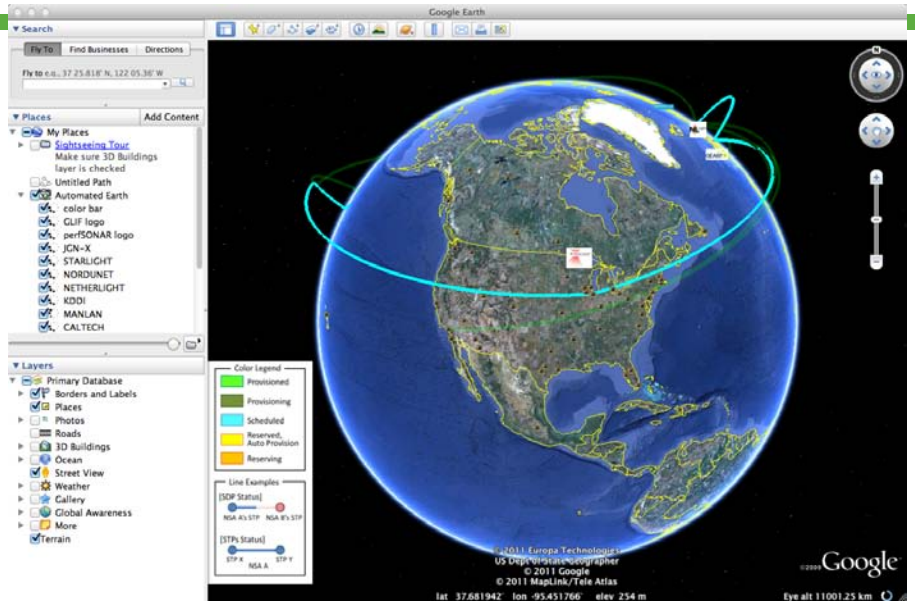
NSI Networks ("A"=Aggregator)

NSI peerings (SDPs) unless otherwise indicated these are vlans 1780-1783

[PLANNED] AUTOMATED GOLE + NSI NETWORK 2013

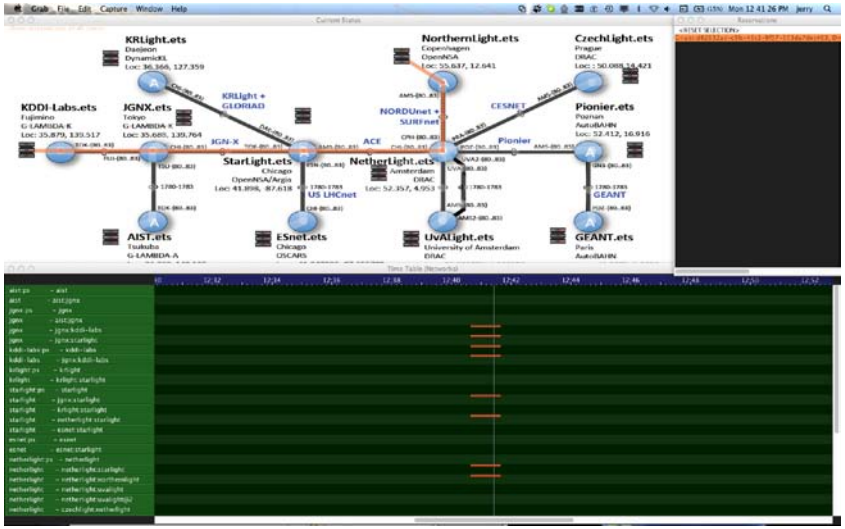


The visuals are stunning!



"Automated Earth" viz
(Takatoshi Ikeda, KDDI-Labs)

http://kote-ps-1.ps.jgn-x.jp/ps/autoearth-nsi/



"NSI Monitor" viz
(Tomohiro Kudoh, AIST)

Watch it Live!

- **Visualization**
- Java web start thing: http://163.220.30.174:8070/monitor.jnlp
- Google earth plugin: http://kote-ps-1.ps.jgn-x.jp/ps/autoearth-nsi/
- Google earth kml: http://kote-ps-1.ps.jgn-x.jp/ps/autoearth-nsiAutoMAP.kml

NSI Development & Road Map

- **OGF NSI-CS version 1.0 is in final draft now**
- Demos:
  - Sep 2011: First NSI CS Interop Plugfest – GLIF 2011 Rio de Janeiro, BR
  - Oct 2011: First NSI Transport Provisioning Future Internet Assembly 2011 Poznan, PL
  - **Nov 2011: Global NSI + AutoGOLE Demonstration Supercomputing 2011 Seattle, US**
- Futures:
  - NSI Topology – dynamic distributed topology exchange. Required to automated the local maintenance of local topology and to enable scalable global pathfinding.
  - NSI Performance Verification – An architecture for automated service verification and fault localization/remediation
  - Common Service Definitions – Enabling interoperable transport services

OGF NSI Working Group

- The OGF NSI WG is an *Open* working group
- This means if *you* have ideas you would like to see incorporated into the NSI framework and/or protocols, please get active in the process:
  - Contact one of the active WG members and pick their brain
  - Join the mailing list, lurk and get up to speed, then join the calls…
  - Contribute – ask, comment, propose…help us sort thru the issues to achieve clarity within the group and consensus within the broader community

Thank You!