PHOSPHORUS

Enabling Grid-Network Services via Control Plane: the Phosphorus G²MPLS way to the e-Infrastructures

Giacomo Bernini

Nextworks

On-demand network services for the Scientific Community workshop & demonstrations

Malaga (ES), June 7th 2009

Outline

- Issues of Grids & network in research networking
- G²MPLS in a small nutshell
  - Rationale and functions
  - Deployment models and network service types
  - Protocol extensions
  - Software prototypes
- Phosphorus G²MPLS test-bed
- Brief introduction to the upcoming G²MPLS demonstration

Issues of Grids & network in research networking

- BoD systems are moving to production in Research Networks
  - e.g. EU-GN2 AutoBAHN, ESNET-OSCARS
  - To provide dynamic connection services, e.g. for HPC/Grid centres
    - mostly WS-based & centralized
    - overlay-style approach for any BoD user
    - immediate and in-advance bandwidth reservations
    - No network recovery
- Still difficult to have combined reservation of network & Grid resources
  - Most of the production Grid middlewares (GLOBUS, UNICORE, gLite, etc.) just play on the Grid layer
    - pre-established and QoS-guaranteed connections between Grid sites (tens of Gbps CBR, minimum jitter and delay, etc.)
  - The mutual unawareness between the decision-making entities in the Grid and in the network layers leads to oversubscription of network resources
- A major step forward: the ***provisioning of network and Grid resources in a single-step***, through a set of seamlessly integrated procedures (Grid Network Service – GNS)

G²MPLS rationale

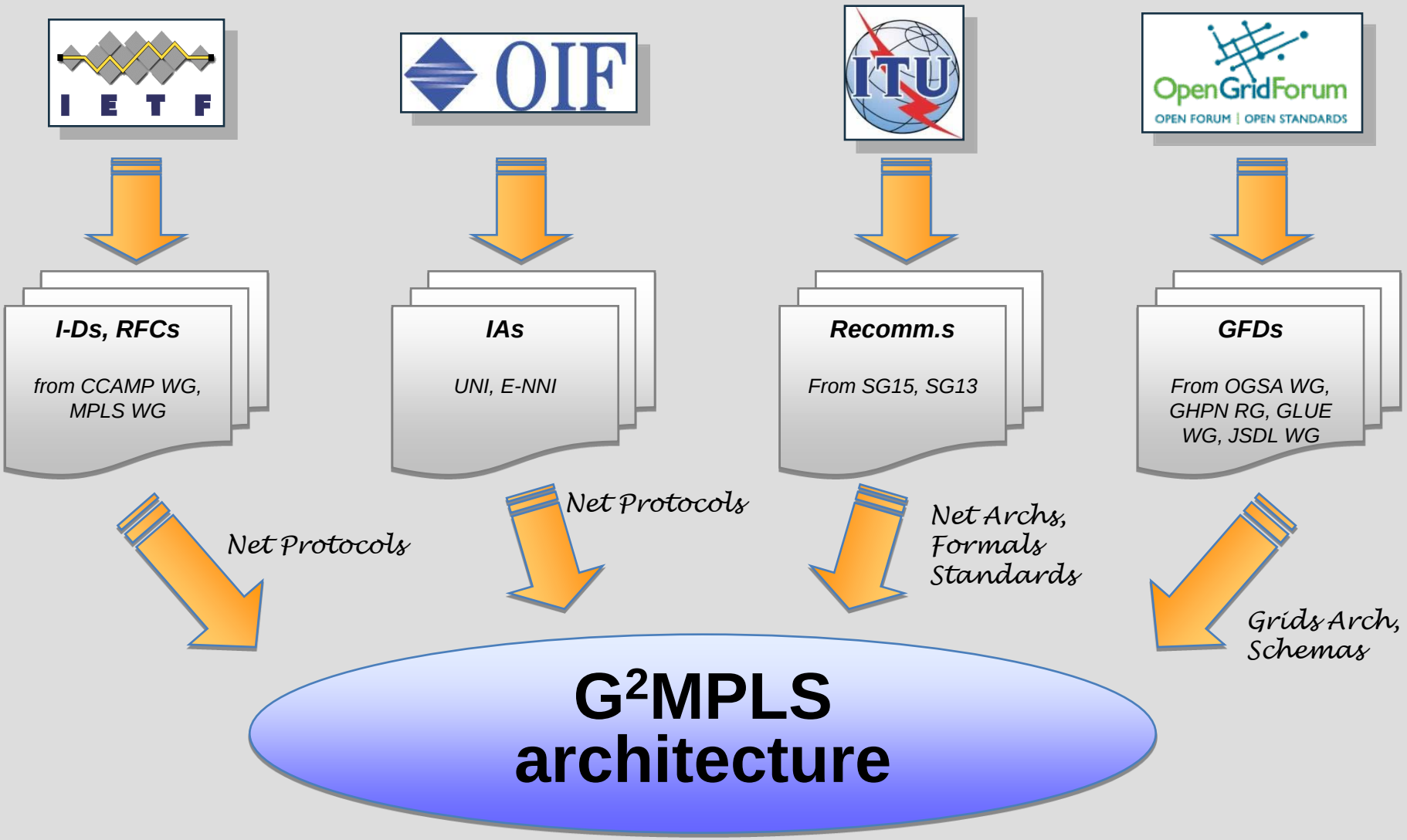
- G²MPLS is …
  - a Network Control Plane architecture that implements the ***Grid Network Services (GNS)***
    - **Single-step provisioning** of Grid and network resources
    - **Advance reservations** of Grid and network resources
  - an ***enhancement of the standard GMPLS*** for “power” users (HPC/Grids)
    - uniform interface for the Grid-user/applications (**G.UNI**)
  - basically, a ***superset of ASON/GMPLS***
    - Grid extensions to UNI, I-NNI, E-NNI, protocols and PCE
- G²MPLS is not …
  - an application-specific architecture
    - Support for ***any kind of end-user applications*** by providing network transport services and procedures that can fall back to the standard GMPLS ones
    - automatic setup and resiliency of network connections for “standard” users

G²MPLS functional scope

- G²MPLS provides part of the functionalities related to the **selection and co-allocation** of both Grid and network resources
- Co-allocation functionalities
  - **Discovery and Advertisement** of Grid + network capabilities and resources of the participating virtual sites (Vsites)
  - **Service setup / teardown**
    - **coordination** with local job scheduler in middleware
    - **configuration** of the involved network connections among the participating Vsites

      (The network end-point – TNA – might not be specified, if Grid resources are specified)
    - **resiliency** mgmt for the installed network connections and possible recovery escalation to the Grid MW for job recovery
    - **advance reservations** of Grid and network resources
  - **Service monitoring**
    - retrieving the status of a job (*Grid transaction*) and of the related network connections

G²MPLS positioning w.r.t. standards

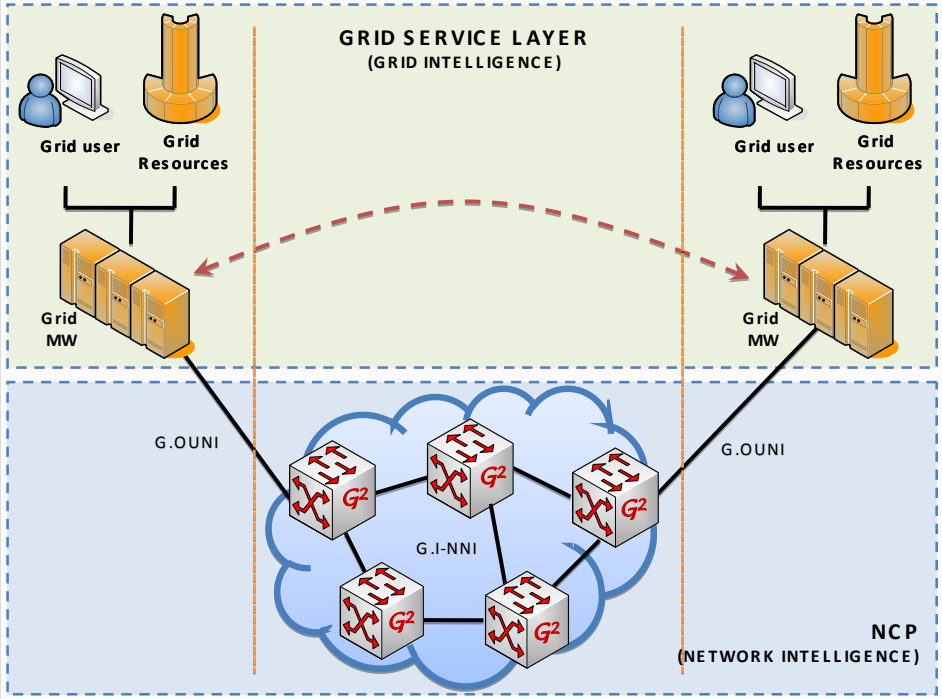


G²MPLS main features

- Switching Capabilities: LSC (with some WSON extensions), FSC (including Eth port-switching)
- G.UNI gateways to allow applications to “drive” its reference G.UNI-C and get dedicated dynamic circuits
- Support for application-specific information across the Control Plane
  - I-NNI, E-NNI and UNI routing extensions to distribute application node resources (grid)
  - I-NNI, E-NNI and UNI signalling extensions, for both Call and LSP signalling
- Support for advance reservations (in a distributed way)
- G.E-NNI as an inter-carrier i/f
  - Integration with AuthN/AuthZ frameworks
- Anycast circuits, i.e. “connect this local TNA with this amount of app (grid) resources”
- Indirect calls, i.e. “connect this TNA / amount of app (grid) resources (maybe elsewhere) to this TNA / amount of app (grid) resources (anywhere)”

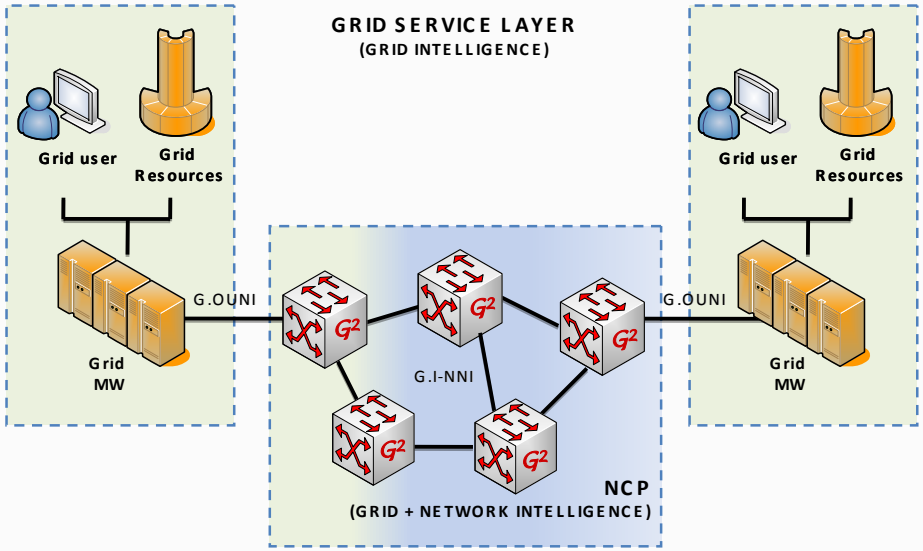
G²MPLS Control Plane models

G²MPLS Overlay model



- **Grid scheduler:**
  - configuration / monitoring of Grid resource
  - configuration / monitoring of net. resource
- **G²MPLS**
  - e2e bearer for network and Grid resources information
  - configuration of just network service

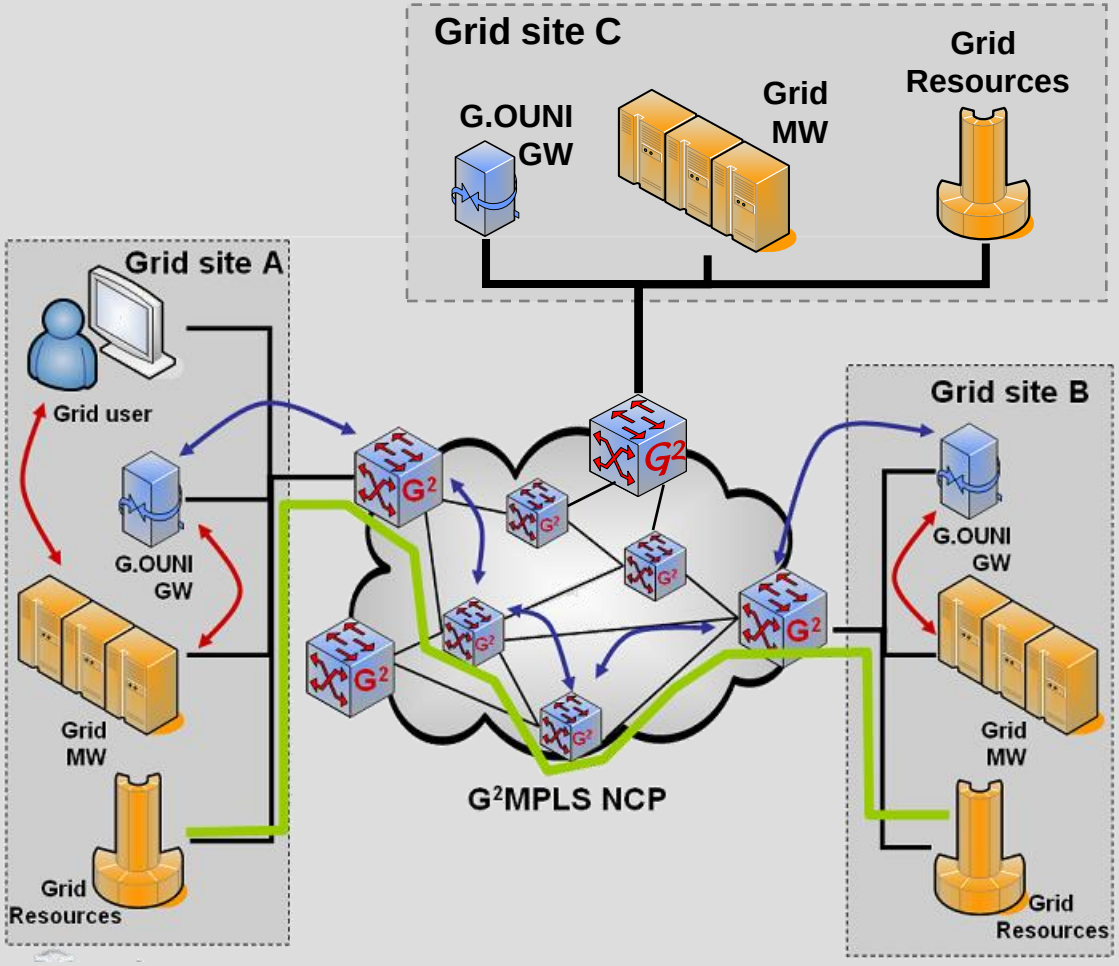
G²MPLS Integrated model



- **Grid scheduler:**
  - workflow coordination services
- **G²MPLS**
  - e2e bearer for network and Grid resources information
  - selection of the job providers (Grid and Net)
  - co-allocation of Grid and Net resources

G²MPLS network service types

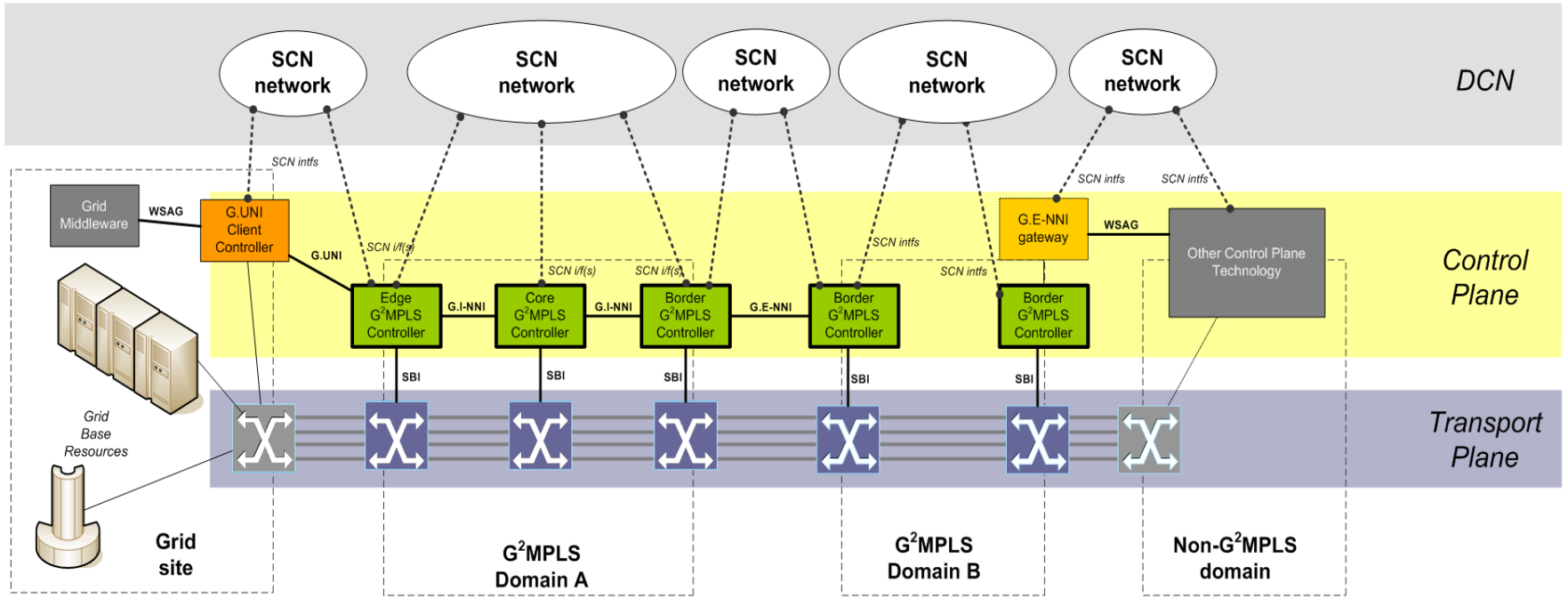
- **Unicasting**
  - GNS request A → B specified by Grid user
  - G²MPLS setup of the e2e call/connection
  - G²MPLS piggybacking of Grid information (resource and job)
- **Anycasting**
  - GNS request A→ *any (e.g. an amount of storage)*
  - ***G²MPLS chooses the "best" destination*** (as viewed by routing topology) and setup the e2e call/connections



G²MPLS extensions to GMPLS

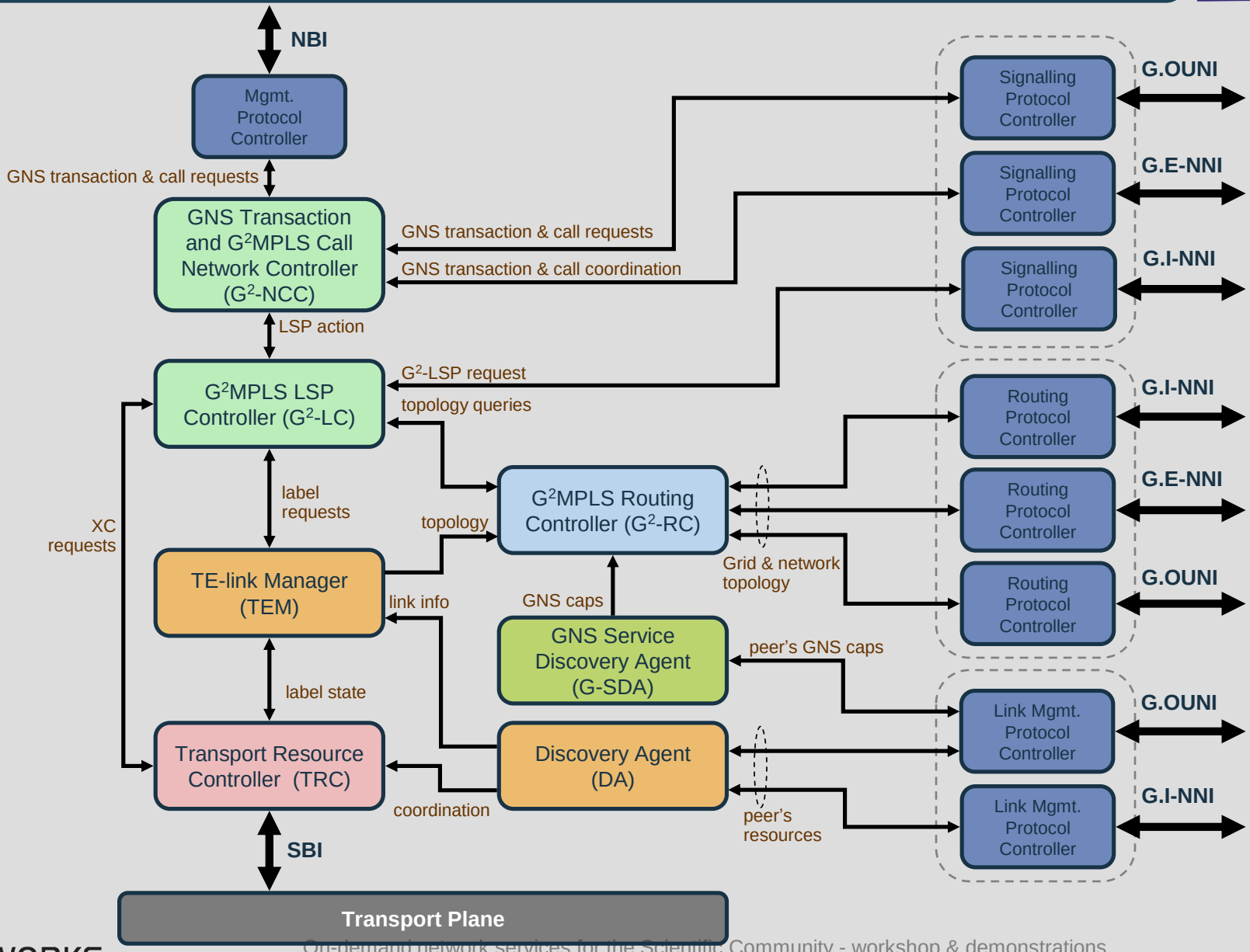
	Services at reference point		ASON / GMPLS	G²MPLS Overlay	G²MPLS Integrated
G.I-NNI	*Routing*	(TE) topology resource publication and discovery	✓	✓	✓
		Opaque piggybacking of discovered Grid resources (**GLUE**)		✓	✓
	Signalling	Connection (LSP) setup/tear-down/crankback	✓	✓	✓
		Connection (LSP) status inquire/notification	✓	✓	✓
		Opaque piggybacking of Grid job (**JSDL**) data		✓	✓
		Handling of Grid job (**JSDL**) data (anycast GNS)			✓
G.OUNI	*Routing*	Grid resources publication and discovery (**GLUE**)		✓	✓
	Signalling	NS (call) setup/tear-down	✓	✓	✓
		NS (call) status inquire/notification	✓	✓	✓
		Transparent handling of Grid job (**JSDL**) data		✓	✓
		(Advance) GNS setup/tear-down			✓
		(Advance) GNS status inquire/notification			✓
G.E-NNI	*Routing*	Network (TE) inter-domain resources publication and discovery	✓	✓	✓
		Inter-domain TE information feed-up/feed-down	✓	✓	✓
		Grid inter-domain resources publication and discovery (**GLUE**)		✓	✓
		Inter-domain Grid information feed-up/feed-down (**GLUE**)		✓	✓
	Signalling	NS (call) setup/tear-down	✓	✓	✓
		NS (call) status inquire/notification	✓	✓	✓
		Handling of Grid job (**JSDL**) data (anycast GNS)		✓	✓
		(Advance) GNS setup/tear-down			✓
		(Advance) GNS status inquire/notification			✓

G²MPLS controller prototypes

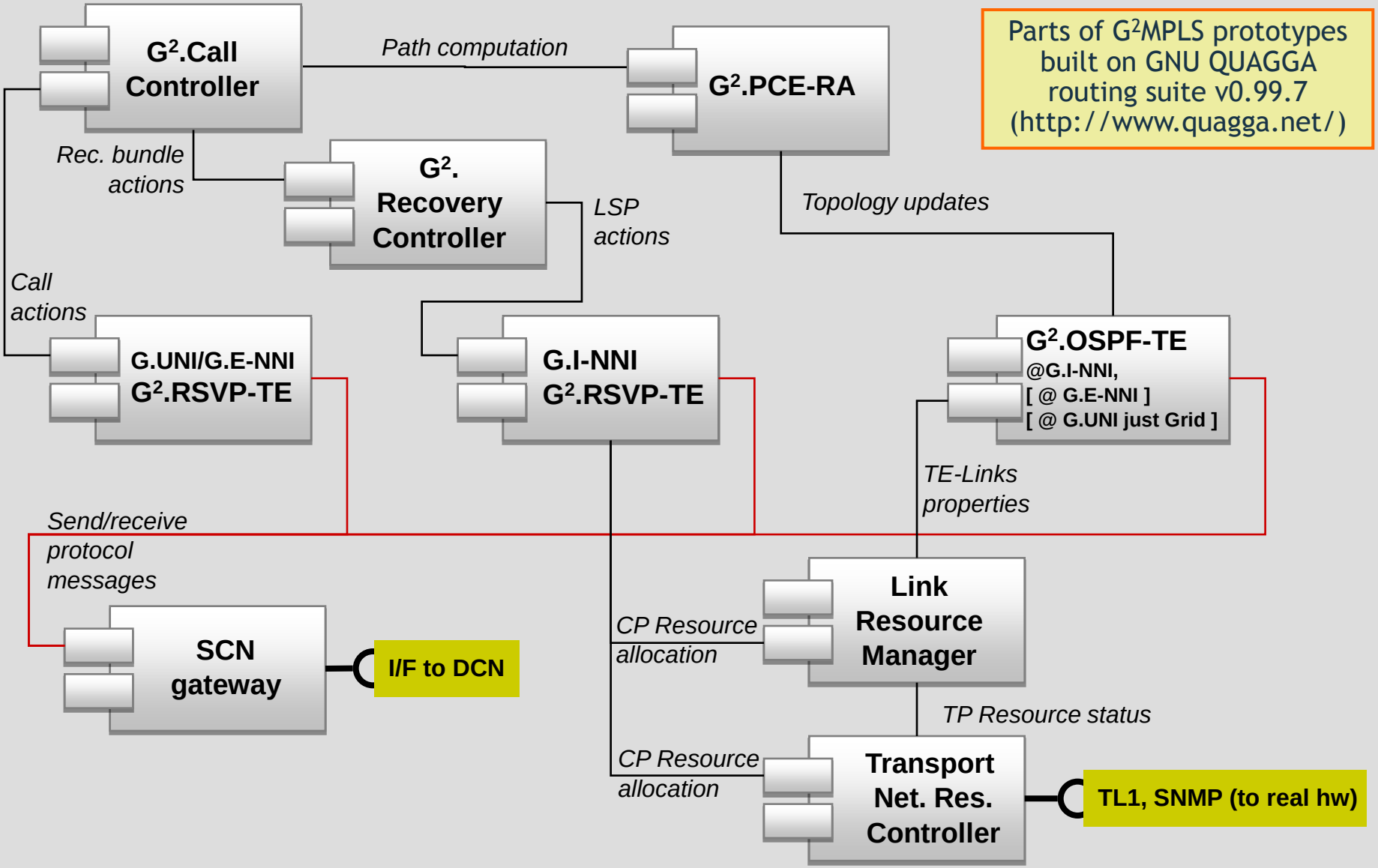


- Four different kinds of controllers can be run depending just on the node configuration (i.e. location in the network)
  - G²MPLS edge controller
  - G²MPLS UNI-C controller
  - G²MPLS core controller
  - G²MPLS border controller

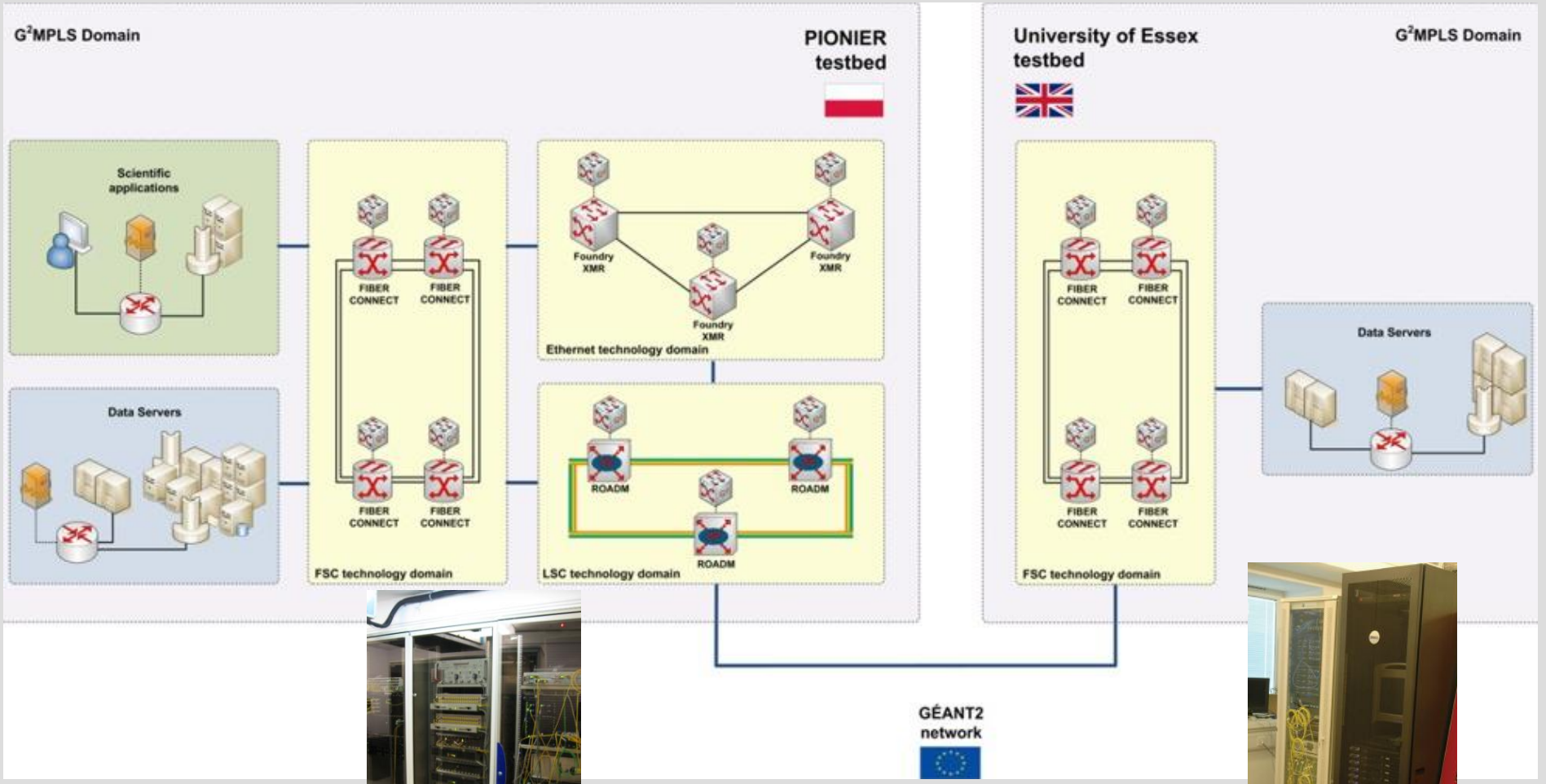
G²MPLS controller functional decomposition



G²MPLS main software modules



Permanent G²MPLS test-bed in Phosphorus [1]

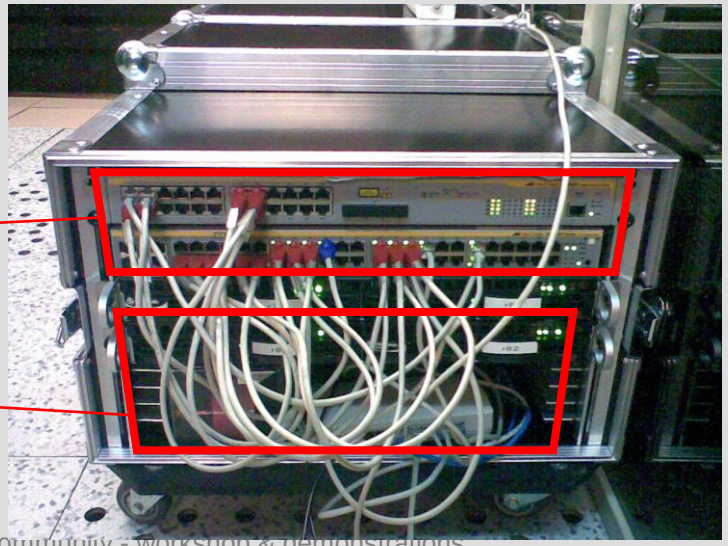


Permanent G²MPLS test-bed in Phosphorus [2]

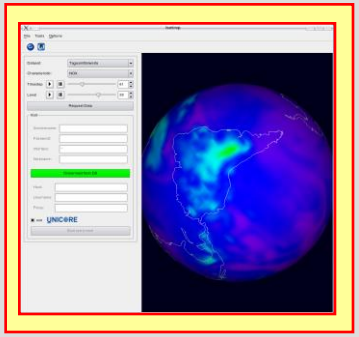
- 2 sites interconnected through the GÉANT2
  - PSNC-PIONIER
  - UESSEX-Photonic Networks Lab
- Multiple administrative domains
- 3 Switching Capabilities
  - LSC (ADVA FSP 3000RE-II ROADMs)
  - FSC (Calient Diamond Wave Fiber Connect)
  - FSC Eth (Allied Telesis AT-8000S + Allied Telesis AT-9424T)
- Different e-Science applications integrated with G²MPLS
  - Distributed Data Storage Systems (DDSS): [***unicast*** & ***anycast***]
  - Collaborative Data Visualisation (KoDaVis) for atmospheric simulations: [***unicast*** & ***anycast***]
  - Wide In Silico Docking On Malaria (WISDOM) for large-scale molecular dockings on malaria study: [***unicast***]
  - *DDSS and G²MPLS publicly demonstrated at SC'08 & ICT'08*
- Interoperation with Harmony in other Phosphorus local test-beds (I2CAT, VIOLA, SURFNET)
  - a dedicated gateway (***HG²-GW***) to handle reservations (signaling) & topology (routing)

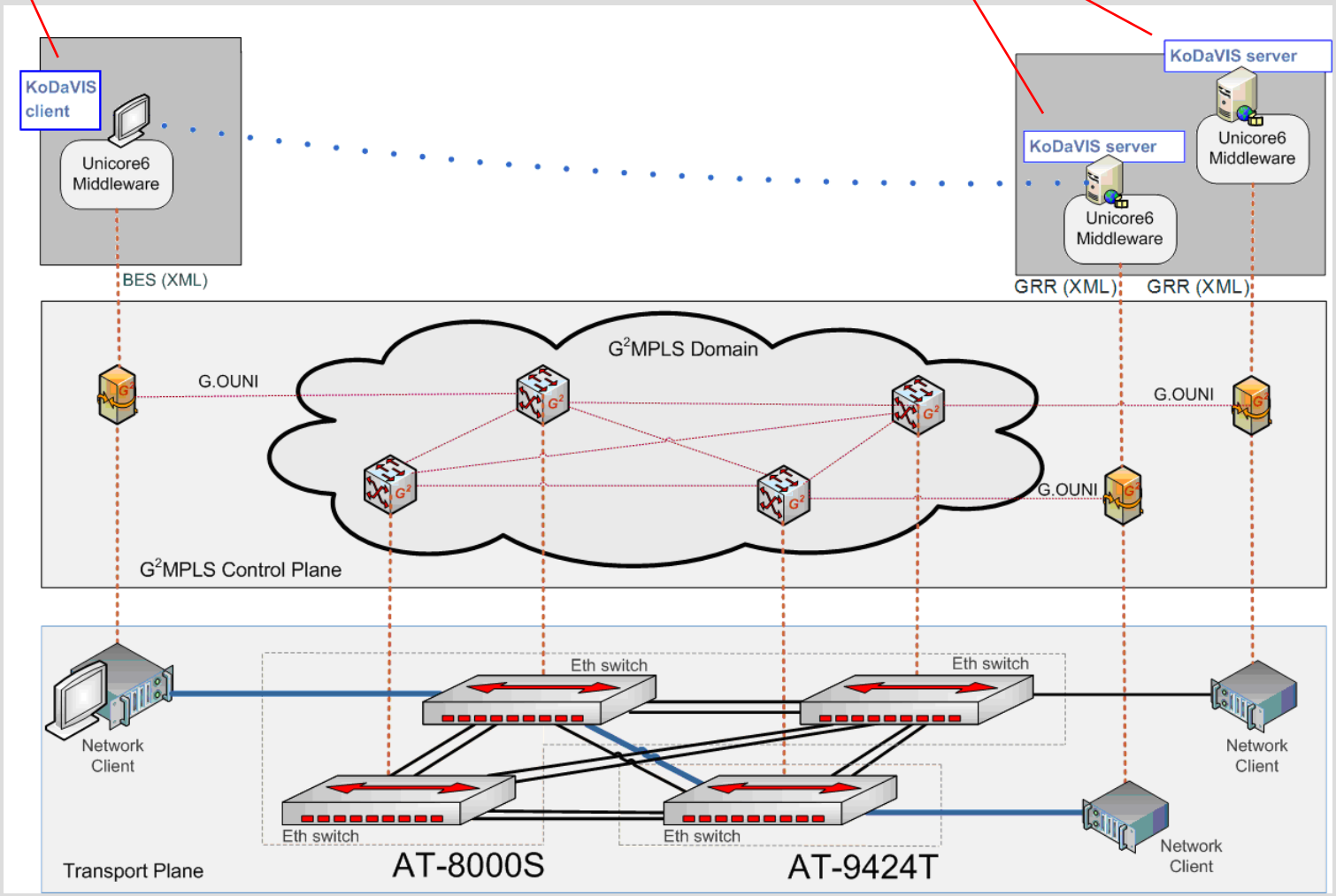
G²MPLS Demonstration @ TNC'09

- Scope: *Grid Network Services (GNS) by G²MPLS with a distributed computation grid application (KoDaVis + UNICORE6)*
  - 1 administrative domain
  - Transport Plane on demo site because of remote connectivity issues
    - Allied Telesis AT-8000S (partitioned in 3 sub-nodes)
    - Allied Telesis AT-9424T
  - Two full chain demo scenarios (KoDaVis-UNICORE6-G²MPLS)
    - *G²MPLS* Overlay (unicast)
    - *G²MPLS* Integrated (anycast)
  - Anycast computing server selected through the dynamic FreeJobSlots parameter



G²MPLS testbed for TNC'09





PHOSPHORUS

Questions?

Giacomo Bernini

g.bernini@nextworks.it

Further details publicly available on

- http://www.ist-phosphorus.eu/deliverables.php