034115

PHOSPHORUS

Lambda User Controlled Infrastructure for European Research

Integrated Project

Strategic objective:
Research Networking Testbeds

Deliverable reference number D1.7

Network Service Plane Interfaces to G²MPLS

Due date of deliverable: 2008-11-30
Actual submission date: 2008-11-30
Document code: <Phosphorus-WP1-D1.7>

Start date of project:
October 1, 2006

Duration:
30 Months

Organisation name of lead contractor for this deliverable:
Rheinische Friedrich-Wilhelms-Universität Bonn

Revision 1

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	✓
PP	Restricted to other programme participants (including the Commission	
RE	Restricted to a group specified by the consortium (including the Commission	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Abstract

This prototype deliverable, named "Network Service Plane Interfaces to G²MPLS", describes the architecture and implementation details for the NRPS adapter / gateway used for the Harmony / G²MPLS interoperability. This includes the illustration of the scenarios, the description of the gateway functionality, and initial test results. The work presented in this deliverable is the outcome of the WP1-WP2 collaboration.

The developed functionality is supported by the Java package "harmony.thin.translator.g2mpls" and the C package "harmony.translator.g2mpls". Source code is publicly available from the project wiki site.

Table of Contents

Table of Figures

Table of Tables

0 Executive Summary

This deliverable, named “Network Service Plane Interfaces to G²MPLS”, is aimed to provide the implementation of the GMPLS driver extension for enabling network connections by using the northbound interface of the G.OUNI-C. Moreover, this deliverable describes the NRPS adapter for NSP interoperability for the East-West interface using the G.E-NNI-U/D API, provided by WP2.

G²MPLS and Harmony share the final goal of the automatic setup of inter-domain connections, when G²MPLS is run in the Phosphorus Overlay mode, i.e. when G²MPLS sets up just Network Services instead of Grid Network Services. In this context, the two Control Plane approaches have many commonalities and their interworking was the topic of deliverable D2.9 [PH-WP2-D2.9]. The current deliverable relies on the previous results reported in [PH-WP2-D2.9] and it deals with architectural considerations derived from two identified interoperability scenarios a) Northbound interaction, which is an overlay-style control of GMPLS/G²MPLS by Harmony/NRPS and b) East-West interaction, which implies a peering between G²MPLS and Harmony.

Signalling and routing specifications for both G²MPLS and Harmony system have been defined in [PH-WP2-D2.9], including internal entities and messages and compatibility issues. Detailed description of methods/system blocks used, list of messages and attributes exchanged for implementation purposes on Harmony system and G²MPLS also were reported. These consider all functions required for the signalling and routing interactions from Harmony to G²MPLS and vice versa.

In this deliverable the developed prototype is described and initial test results are given. The remainder of this deliverable is organised as follows. First, Section 1 gives a brief overview of the two involved systems Harmony and the G²MPLS control plane in the Phosphorus framework. Section 2 provides detailed information regarding the proposed gateway architecture that allows interoperability between the G²MPLS control plane and the Harmony system. Section 3 provides detailed information about the interoperability scenario. This detailed information is illustrated by the first test results. Finally, Section 4 summarizes our conclusions.

The developed functionality is supported by the Java package “harmony.thin.translator.g2mpls” and the C package “harmony.translator.g2mpls”. Source code is publicly available from the project wiki site.

1 Interworking Architectures

As a foundation for the following sections the two involved systems are described briefly. In Section 1.1 an overview of Harmony and its underlying data model is given. In Section 1.2 the G^2MPLS control plane is outlined and incorporated into the Phosphorus architecture.

1.1 Harmony

1.1.1 Overview

The Harmony system is a multi-domain, multi-vendor network resource brokering system. Harmony defines an architecture for a service layer between the Grid middleware and applications and the Network Resource Provisioning Systems (NRPS).

Harmony was developed based on two main assumptions: the system had to be multi-domain and had to be capable of creating end-to-end optical and layer 2 paths in a seamless way for the scientific applications at the end points. These conditions were set because the project aims to make existing provisioning systems interoperable and fill in the gap between these hardware-coupled software pieces and the Grid application middleware. The architecture of the system is built over a SOA model and is composed of the three layers: the Service plane, the Network Resource Provisioning plane and the Control plane.

Harmony service interface (HSI) is the component that enables the communication between the network service layer and the adaptation layer or the outward world. This service interface is common both for the network service layer and the adaptation layer. The HS contains three web services: Reservation-WS, Topology-WS and Notification-WS. Moreover, there is a common part of the three Web Services inside the whole interface. This common part defines the data type common for the three components.

Figure 1.1 shows a simple Harmony set up with one IDB at the NSP and one NRPS which is being connected to the NSP by means of one HNA (only one HNA allowed per NRPS). Names on the arrows are detailed hereby:

a1. Resource reservation requests Client-to-IDB (administrator or normal user or middleware).

a2. Topology requests Client-to-IDB (administrator only).

b1. Resource reservation requests to NRPS (normal operation).

b2. Topology requests IDB-to-IDB within the NSP (topology exchange).

b3. Resource reservation requests IDB-to-IDB within the NSP (topology exchange).

c1. Topology requests HNA-to-IDB within the NSP (topology exchange).

c2. Resource reservation requests HNA-to-IDB (request forwarding).

d. NRPS-dependent interface.

e. Network device dependent interface.

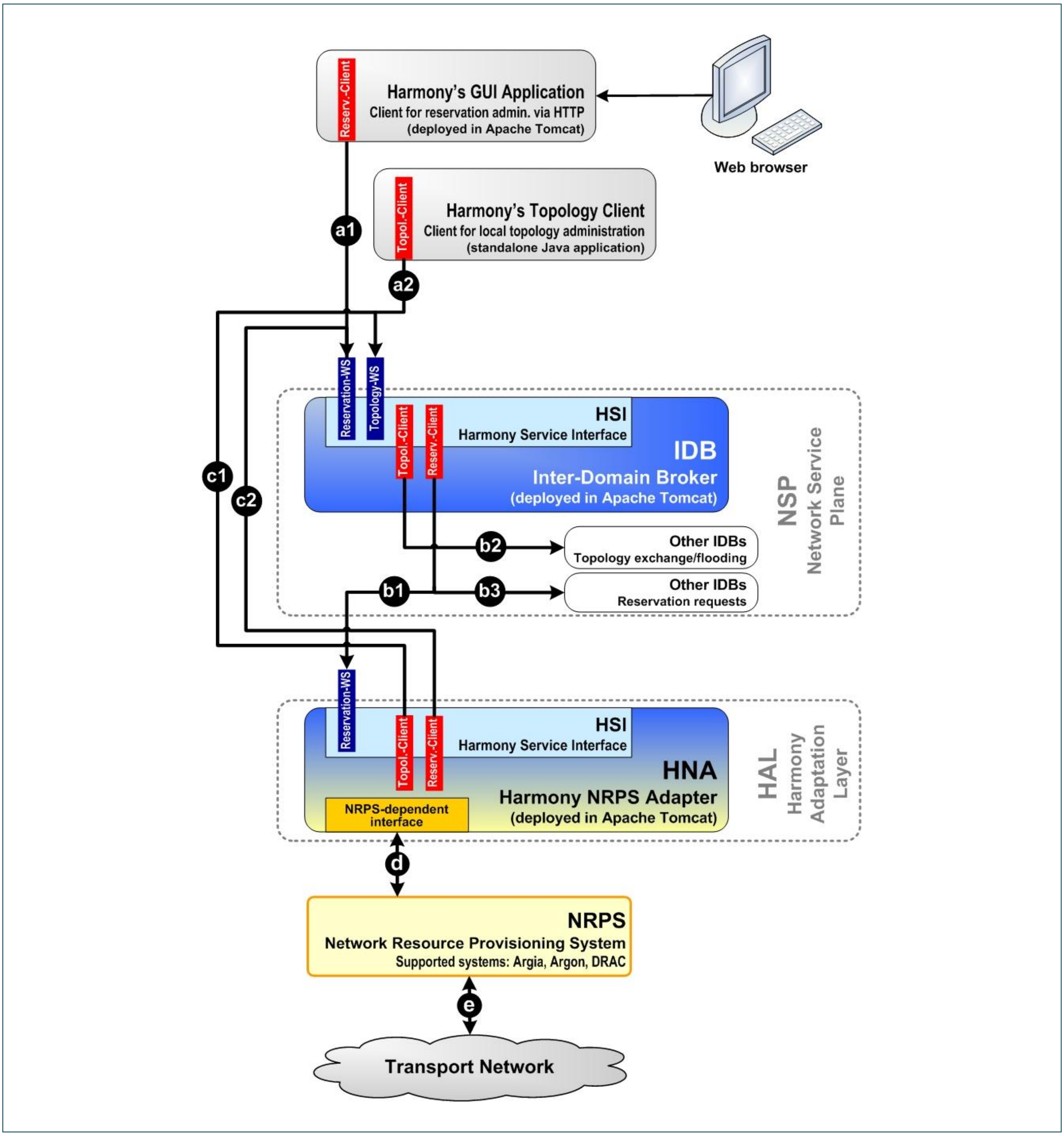

Figure 1.1: Simple Harmony system set up.

1.1.2 Harmony Reservation Creation

The user has to be able to create, delete and query advance network resource reservations. Therefore, the IDB must provide these capabilities through its interface. The Reservation WS is the component that enables the user to create, delete and query advance reservations. The functions that the user or the grid middleware can invoke to deal with reservations are the following:

Availability request: Checks the availability of the resources required. When checking availability, the resources are not reserved; instead, the IDB returns a response indicating whether the reservation would be feasible or not. In case it is not feasible, additional information on alternative starting time is added to the response.

Reservation request: This request reserves network resources for one or more services. These resources are exclusively reserved between the specified start and end times.

Reservation status: Queries one or more reservations previously made in the network.

Cancel reservation: Deletes a reservation. Once this is done, all the resources that participate in the connection are released and tagged as available.

When a reservation request is received by the IDB, path computing method starts looking for feasible paths. Upon the completion of this task, the availability of the resources within all involved domains is requested via the *isAvailable* method. In case the resources are available, the *create reservation* method is invoked and the resources will be reserved. Otherwise, the resources are pruned and the path computer keeps on looking for feasible paths. I

1.1.3 Harmony Topology Knowledge

The peer-to-peer topology exchange is implemented using only the addOrEditDomain function. Therefore, all relevant information is contained in this function's DomainInformationType parameter as optional fields. Inside the DomainInformationType, only the DomainId and ReservationEPR fields are mandatory for several reasons. However, in the distributed mode of operation, where only the addOrEditDomain operation is used to advertise domain information stored in a DomainInformationType, the following fields should be set additionally:

The *Relationship* field (which defaults to "subdomain" if it is not set for backward compatibility reasons).

The *SequenceNumber* field is important in peer-to-peer scenarios to allow old information to be discarded.

The *InterdomainLink* field contains all information necessary for receivers to lookup the required topology information if the advertisements of both domains connected by an interdomain link are known. Note that it is not necessary for a domain to know the TNA associated with an interdomain link within its peer domain.

Name	Type	optional / unbounded	Description

DomainId	string	N/N	Unique identifier for the domain
Relationship	string	Y/N	Enumeration. Can be one of the self-explanatory values “subdomain” or “peer”.
SequenceNumber	int	Y/N	Increased by the origin domain, allows others to check if information has changed and is more current than previously stored information.
Description	string	Y/N	Short description of the domain.
ReservationEPR	anyURI	N/N	Endpoint reference of the domain’s Reservation-WS.
TopologyEPR	anyURI	Y/N	Endpoint reference of the domain’s Topology-WS.
NotificationEPR	anyURI	Y/N	Endpoint reference of the domain’s Notification-WS.
TNAPrefix	string	Y/Y	List of TNA prefixes the domain is responsible for.
InterdomainLink	Interdomain-LinkType	Y/Y	List of interdomain links originating from the domain.
avgDelay	Int	Y/N	Average delay for paths in this domain; path computer optimization.
maxBW	Int	Y/N	Maximum bandwidth for paths in this domain; path computer optimization.
Feature	string	Y/Y	List of features supported by this domain. Currently unused.

Table 1.1: Fields in the DomainInformationType.

Name	Type	optional / unbounded	Description
LinkId	string	N/N	ID of the link, must be unique only within same pair of domains.
SourceEndpoint	string	N/N	TNA of the source endpoint in the domain advertising this interdomain link.
DestinationDomain	string	N/N	Identifier of the destination domain this link connects to.

Table 1.2: Fields in the InterdomainLinkType.

1.2 G^{2}MPLS Control Plane

G^{2}MPLS is the Phosphorus framework that includes the following layers as shown in **Figure 1.2**:

- Grid layer,

- G^{2}MPLS Network Control Plane, Harmony System
- Transport Plane.

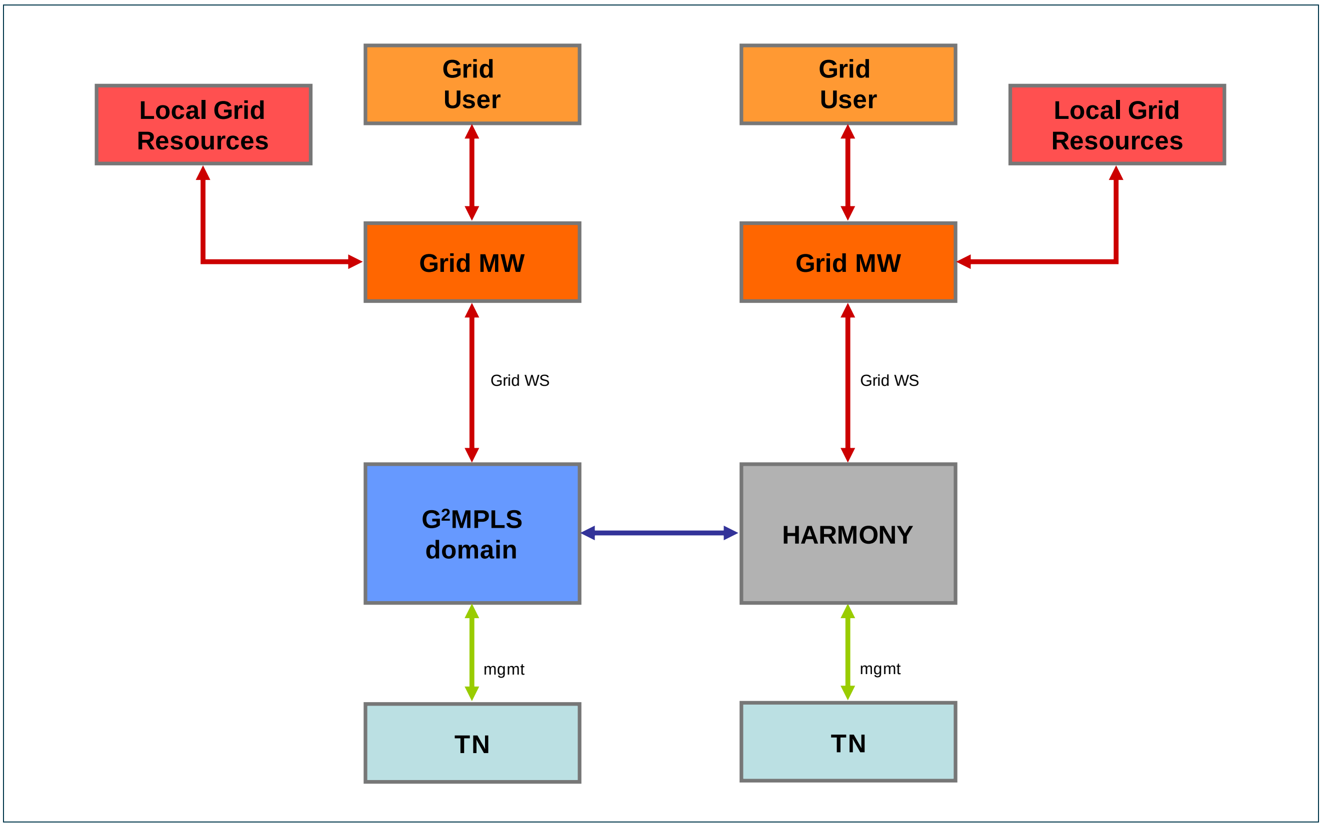

Figure 1.2: G^{2}MPLS positioning in the Phosphorus framework including the Harmony system.

The Transport Plane is the basic layer comprising all the data bearing equipments and their configuration interfaces.

G^{2}MPLS Network Control Plane is aimed to provide:

- *Discovery and advertisement* of Grid capabilities and resources of the participating Grid sites (*Vsites*);
- *Grid and Network Service setup* including:
  - *Coordination* with the Grid local job scheduler in the middleware responsible for the local configuration and management of the Grid job;
  - *Configuration* of the network connections among the sites participating to the Grid job;
  - *Management of resiliency* for the installed network services and possible escalation to the Grid middleware components that could be responsible for check-pointing and recovering the whole job;
  - *Advanced reservation*s of Grid and network resources;
- *Service monitoring* both for the Grid job and the related network connections.

The Grid layer is intended to comprise both Grid application and Grid middleware. Discussion on the architecture of the Grid layer is out of the scope in this document. The relevant aspect of this layer in a G^2MPLS perspective is the functionalities exported to/by the underlying network Control and Management Planes. G^2MPLS is primarily intended to interconnect remote instances of the Grid middleware, which are responsible for managing Grid resources localized in different sites (ref. **Figure 1.2**). In a more visionary scenario and in support of possible future applications, G^2MPLS could also interconnect directly Grid users/applications and Grid resources. Moreover, the southbound interface of the Grid layer represents a technological boundary, not necessarily an administrative boundary. This reference point cannot be addressed with standard IETF peer/integrated models, because any Grid component in the application or even in the middleware cannot peer with any Control Plane instance running on a network node.

Discussions on the G^2MPLS interworking with the Harmony system are provided in [PH-WP2-D2.9] and Section 2 onwards.

2 Gateway Architecture and Functionality

As shown in **Figure 2.1** the Harmony architecture can be extended by adding gateways to a Harmony Inter Domain Broker. In Section 2.1 the chosen architecture is described briefly and in Section 2.2 an overview of the implemented functionality is given.

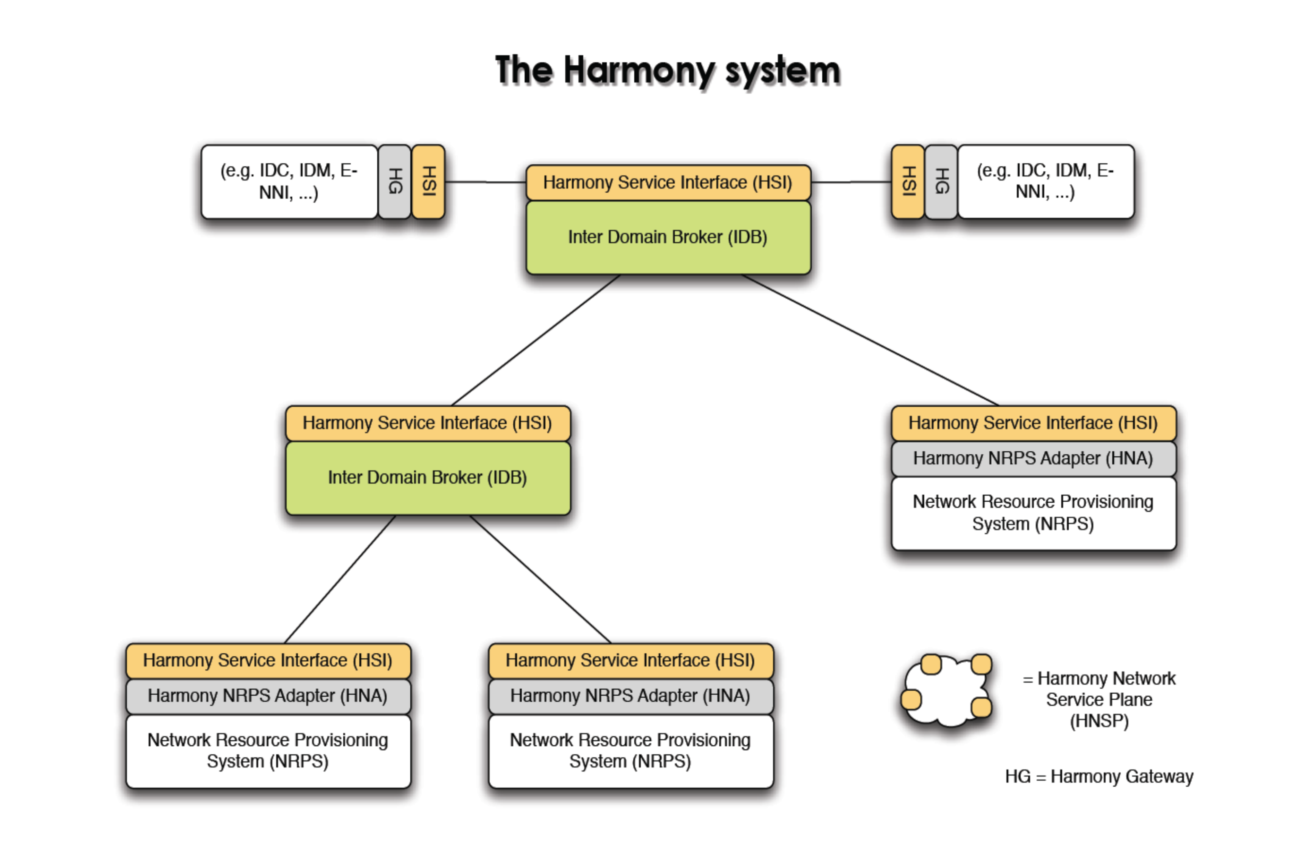

Figure 2.1: Overview of the Harmony architecture and possible gateway extensions.

2.1 Proposed Architecture

Different interoperability scenarios have been identified and discussed for the communication between Harmony and G^{2}MPLS in the previous deliverable D2.9 (cf. [PH-WP2-D2.9]). Two main cases identify the scenarios describing different interaction scenarios between the main roles assigned to each of the actors for northbound interfacing and East-West interfacing:

- *Northbound interfacing*
  - *Case A*: standard GMPLS as the only controller of a domain interfaced to the Harmony/NSP through a gateway module,
  - *Case B:* standard GMPLS as “slave” controller of a domain mastered by an NRPS;
- *East-West interfacing*
  - *Case C: full-fledged G²MPLS as the only controller of its Grid & network resources and peering with a neighbouring Harmony NSP*

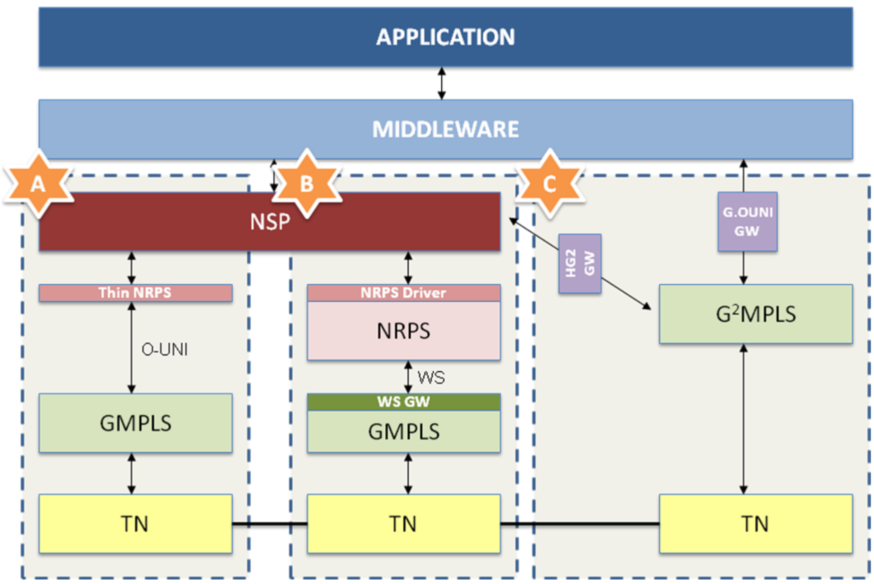

Figure 2.2: GMPLS/G²MPLS interworking with Harmony.

2.1.1 Interworking through the Northbound Interface

Harmony northbound services establish and release network circuits and retrieve topology and connections status information. Therefore, the external interface with G^2MPLS is limited to operate just with network resources as a standard GMPLS Control Plane. [PH-WP2-D2.1] and **Figure 2.2** shows the possible options with use cases.

In the first identified case (A), an NRPS driver acting as a Thin-NRPS translates incoming requests from the Network Service Plane (NSP) into standard O-UNI RSVP messages, establishing Soft Permanent Connections (SPC) between defined end points which are represented as TNAs. This approach has already been implemented and described in [PH-WP1-D1.3] and tested in [PH-WP1-D1.6].

In the second identified case (B), an NRPS driver translates incoming requests from the NSP into proprietary web service messages for G^2MPLS running in the overlay mode. Since this solution would neither gain many benefits nor allow bilateral translation of requests (with respect to the third case (C)), this approach was no longer under consideration.

2.1.2 Interworking through East/West Interfaces

The third case (C) describes the interworking between G^2MPLS and Harmony systems through the G²MPLS east/westbound interfaces when the G²MPLS domain and the NSP are considered to be at the same level. Nevertheless, transactions can just involve network services since even if G²MPLS controls network and Grid resources, the scope of Harmony is limited to support just network services. The completion of the Grid transactions is up to the Grid Middleware layer through dedicated communications.

The chosen solution for the current implementation incorporates a gateway that implements a web client/server able to parse and generate Harmony Web Services requests and responses and translate the calls into the corresponding CORBA methods that trigger the G^2MPLS services (**Figure 2.3**).

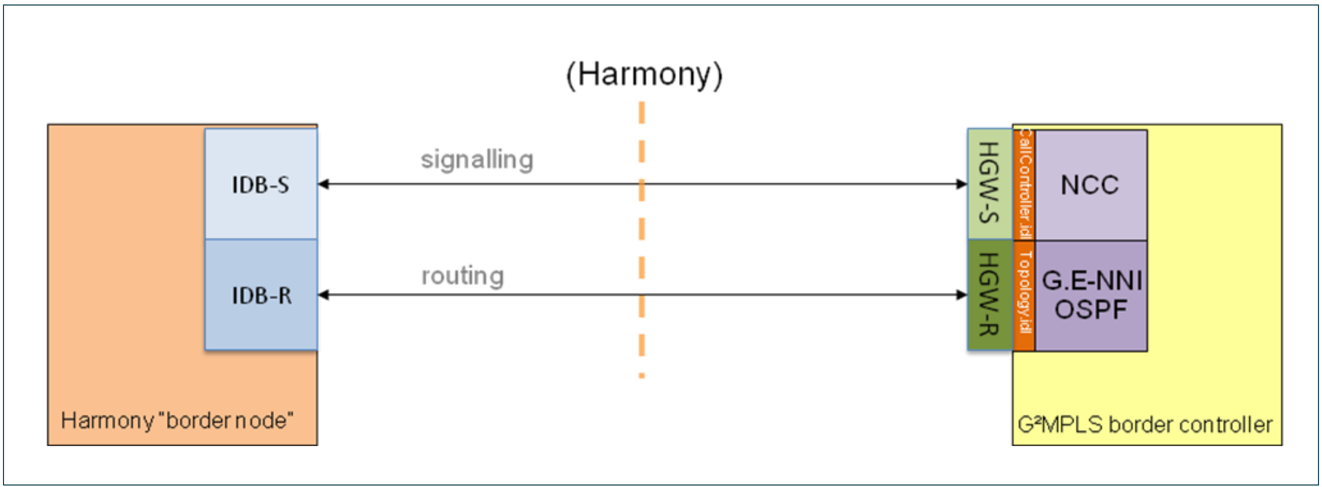

Figure 2.3: GMPLS/G²MPLS interworking with Harmony. The implementation of this solution is described in the next sections of this deliverable.

2.2 Gateway Operation

2.2.1 Gateway Functionality

Some of the functionalities provided by Harmony and G^2MPLS can find their equivalent from one to the other. Thus, for these functions, the gateway just makes a simple translation of the parameters required and forwards the request. Table 2.1 shows which operations are offered by both sides and which are also supported in the HG^2GW.

Operation	Harmony IDB	G²MPLS	HG²GW
createReservation (Harmony) createCall/callSetTna/callSetUp (Net) (G²MPLS)	✓	✓	✓
cancelReservation (Harmony) callSetDown (G²MPLS)	✓	✓	✓
getStatus (Harmony) callGetDetails (G²MPLS)	✓	✓	✓

isAvailable (Harmony)	✓		✓ (always true)
getReservations (Harmony) getCalls (G^{2}MPLS)	✓	✓	✓
activate (Harmony)	✓		
addOrEditDomain (Harmony) netNodeGet/tnaIdsGetAllFromNode/... (G^{2}MPLS)	✓	✓	✓
createCall/callSetTna/callSetUp (Grid) (G^{2}MPLS)		✓	

Table 2.1: The operations on the Harmony IDB, G^{2}MPLS, and the HG^{2}GW that are currently implemented.

Basic operations such as a connection creation request, connection deletion request and get the status of the connections are available in Harmony and in G^{2}MPLS. Also a topology discovery service can be mapped from one language to the other. Finally, G^{2}MPLS allows Grid Network Services (GNS) requests, which is not supported by Harmony; therefore this feature is not implemented in HG^{2}GW.

In order to achieve full inter-operability between Harmony and G^{2}MPLS the translator must provide signalling and routing services. Following the pattern implemented in the Internet2 (IDC) – Harmony collaboration, the first service implemented in the HG^{2}GW is the signalling service.

2.2.2 Signalling

The signalling service enables the creation and deletion of connections between the two systems, as well as provides the capability of querying for the status of a connection involving resources located in both Harmony and G^{2}MPLS domains.

From the Harmony point of view, the HG^{2}GW is seen as a domain under the control of the main IDB and the gateway is the entity which talks to the G^{2}MPLS domain. As commented previously, as a first inter-operability step, the HG^{2}GW maps the signalling methods from Harmony to G^{2}MPLS and vice-versa.

The methods available at the translator are: create reservation, cancel reservation and get the status of a reservation. The correspondence between Harmony methods and G^{2}MPLS is depicted in next table (c.f. [PH-WP2-D2.9]). The other methods used in Harmony to establish a path, *isAvailable*, is not mapped in the translator, since G^{2}MPLS has not a corresponding method. In order to solve this lack of compatibility, when an availability requests reaches the translator, a dummy response is generated and forwarded to the main IDB be a preceding “thin translator” (cf. Section 2.2.3).

Harmony method	G^{2}MPLS calls
createReservation(parameters)	CallController::Mgmt::createCall(parameters) CallController::Mgmt::callSetTna(parameters) CallController::Mgmt::callSetUp(parameters)

	*****wait() CallController::EastWest::callRaiseMsgEvent(<SetupConfirm>)
cancelReservation(parameters)	CallController::Mgmt::callSetDown(callId) *****wait()
getStatus(parameters)	CallController::Mgmt::callGetDetails(callId)
getReservations(parameters)	CallControler::Mgmt::getCalls()

Table 2.2: Signalling method mappings.

2.2.3 Routing

Regarding the routing messaging, both Harmony and G^{2}MPLS use mechanisms to exchange topology information of their domains. Harmony uses the addOrEditDomain method and G^{2}MPLS uses netNodeGet, tnaIdsGetAllFromNode, teLinkGetAllFromNode and teLinkGetCom CORBA Calls.

The routing service between Harmony and G^{2}MPLS is not provided by the translator at this first step, since the implementation of this service add a huge complexity and it is still under study. The main issues and possible solutions are listed in Phosphorus public D2.9. An overview about the translation of the topology format from one side to the other and the mapping of messages are shown in **Figure 2.4**.

The current prototype is divided into two modules: the “Harmony Thin G^{2}MPLS Translator” that sends preconfigured addOrEditDomain requests and forwards createReservationRequests, and the “Translator” that incorporates the domain into G^{2}MPLS and translates reservation requests and responses (cf. **Figure 2.5**).

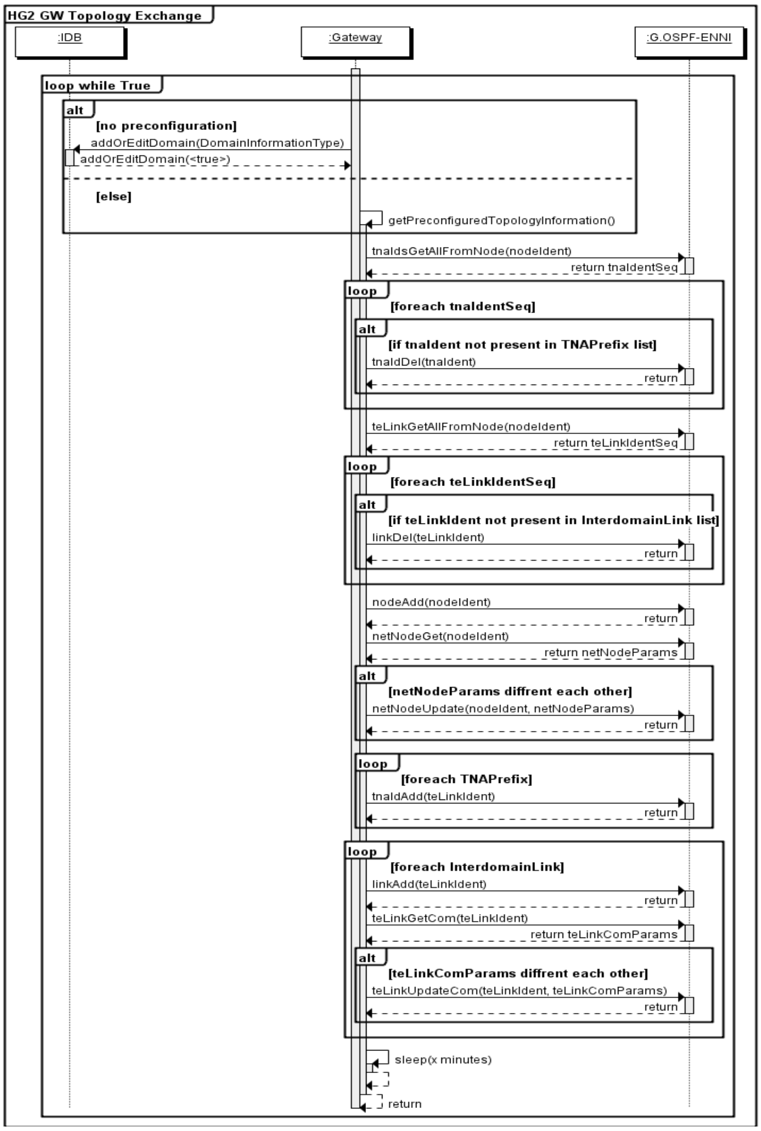

Figure 2.4: Sequence Diagram of the HG^{2}GW Topology Exchange – Harmony to G^{2}MPLS.

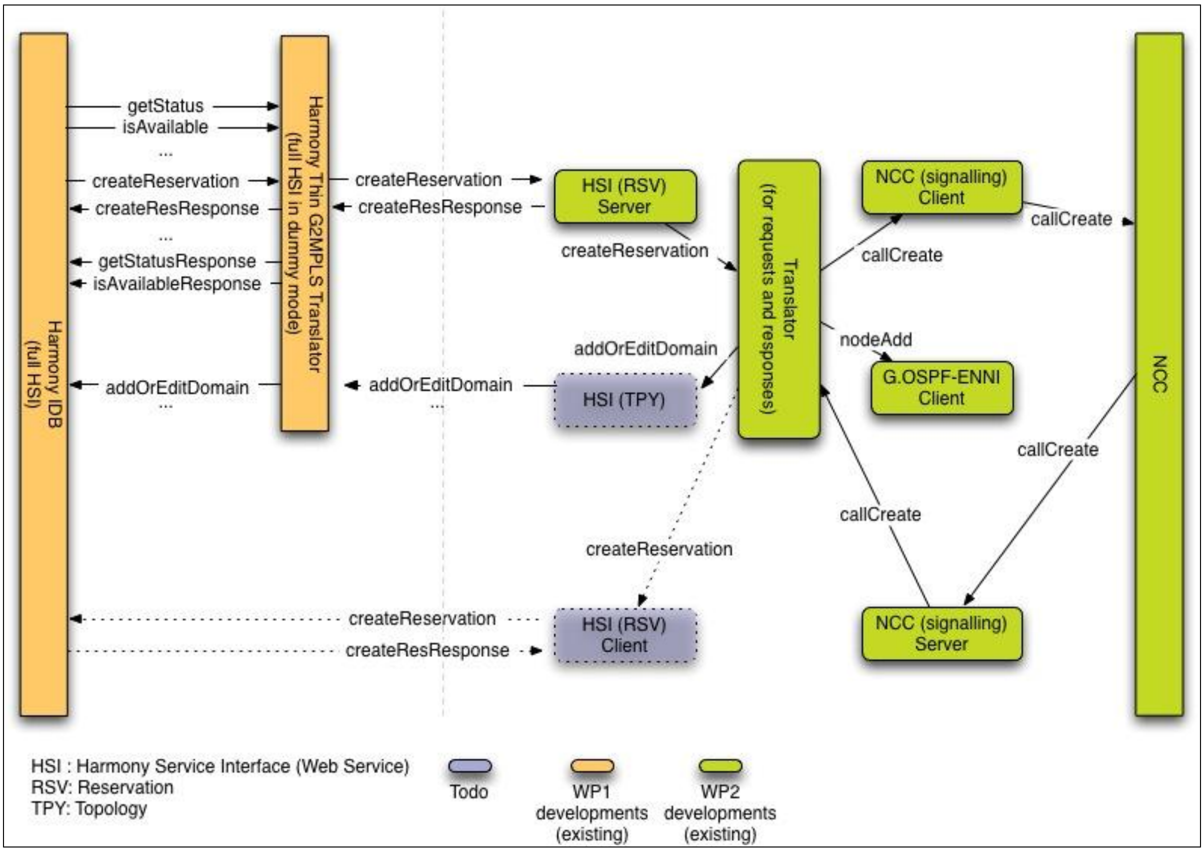

Figure 2.5: Two modules used in the translator architecture.

3 Interoperability Scenario

In order to test the implemented prototype a test-bed was created that is described in Section 3.1. The test workflow and scenarios done in this test-bed are described in Section 3.2 and some results are given in Section 3.3.

3.1 Test-bed Setup

Although the current test-bed in the Phosphorus project has inter-domain links between G^2MPLS domains and Harmony domains, there is not such a link exclusively dedicated for Harmony-to-G^2MPLS data-plane interconnection. WP1 and WP2 partners have agreed to create a new inter-domain link for this exclusive use. After several discussions, an agreement was achieved in which i2CAT (WP1) and PSNC (WP2) partners would request a new VLAN in their GÉANT2 link for this exclusive purpose. Hence, new VLAN tagged 180 via GÉANT2 network has been deployed.

In the i2CAT side, VLAN 180 is terminated in a port directly controlled by ARGIA, as untagged traffic. This port has been assigned TNA 10.3.1.5 in Harmony. On the PSNC side, VLAN 180 is terminated in a Calient DiamondWave FiberConnect device, from where is directly switchable to G^2MPLS data-plane. TNA for of this endpoint is 20.20.20.1, for G^2MPLS, and 10,1.2.1 for Harmony.

From Harmony's point of view, G^2MPLS domain is a cloud which signalling must be done via the Harmony-to-G^2MPLS Gateway (also known as HG^2GW). The G^2MPLS domain behind HG^2GW hosts all endpoints named with TNAs 10.1.2.0/24.

In the deployment of the interconnection link between both data-planes, three steps have been considered:

STEP0: the current status of the test-bed that supports the specification of the Harmony to G^2MPLS signalling. VLAN 638 between i2CAT and PSNC was allocated as data-plane of the Harmony domain at PSNC and permitted communication towards Harmony domain deployed at UESSEX. VLAN 641 was proposed as the new inter-domain link for Harmony to G^2MPLS data-plane interconnection. This VLAN tag (641) would finally be modified by VLAN providers, due to conflicts in the transient (transparent) domains. STEP0 setup is depicted in the following image,

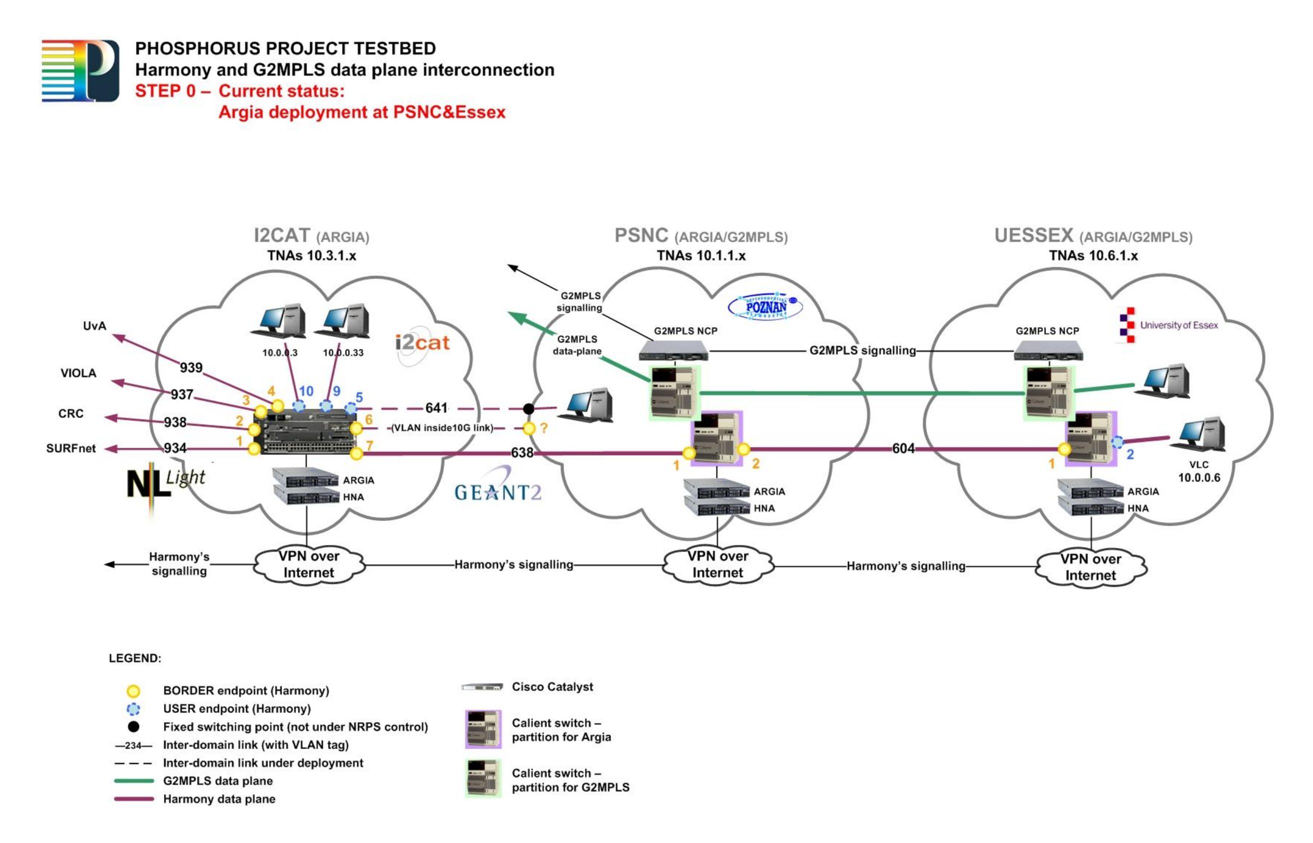

Figure 3.1: Harmony-G^{2}MPLS interconnection: step 0.

STEP1: Once the new VLAN was deployed, endpoint 10.3.1.5 at i2CAT (Harmony) was transformed to border endpoint, instead of user endpoint as it was up to the moment. This way, this endpoint could be used as inter-domain endpoint, and therefore, an inter-domain link could end on it. Otherwise, the IDB would not be able to keep it in its database. STEP1 setup is depicted in the following image,

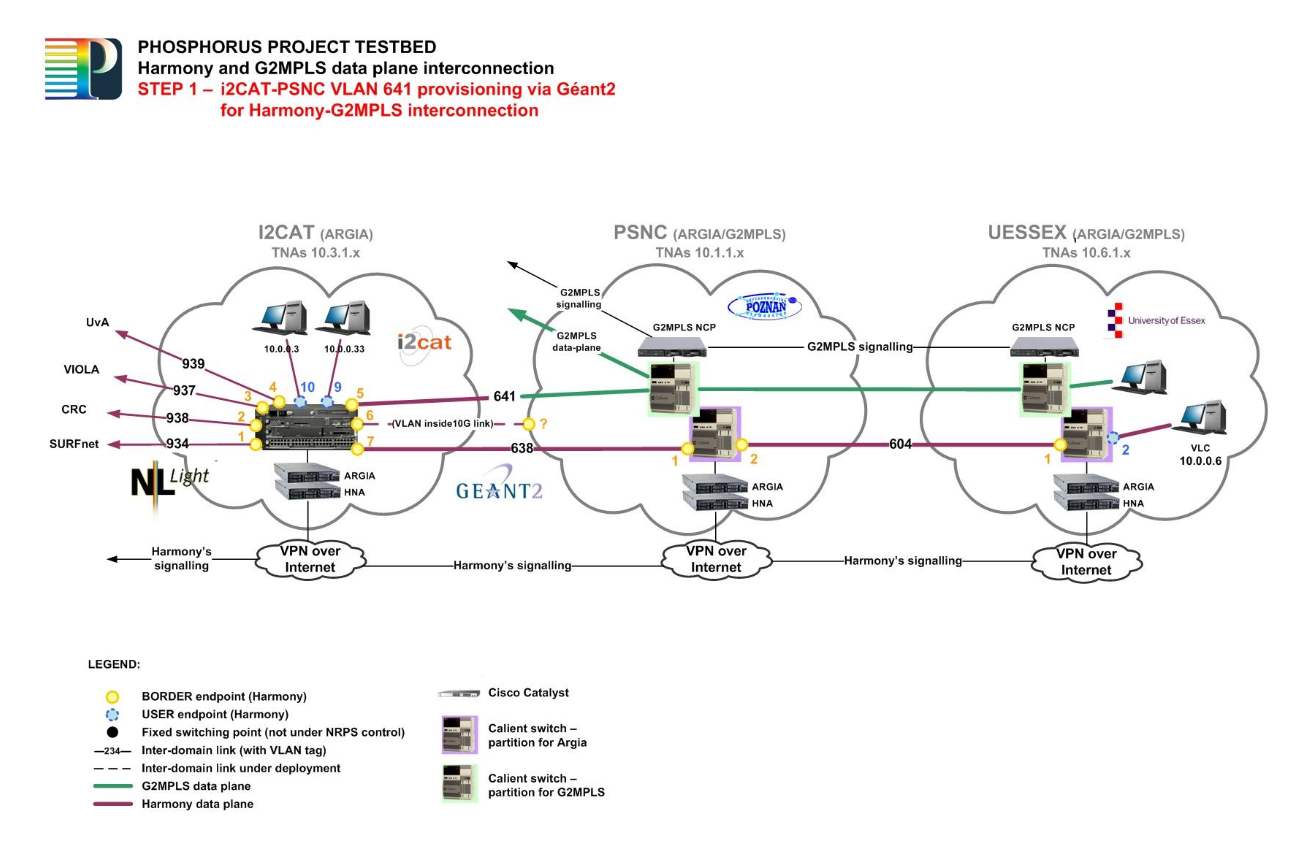

Figure 3.2: Harmony-G^{2}MPLS interconnection: step 1.

STEP2: From that moment on, data-plane was interconnected between Harmony and G^{2}MPLS. Thus, an interconnection of signalling planes was needed. The first approximation presented on STEP2 was not feasible, since the direct connection would mean that both Harmony and G^{2}MPLS should be able to understand each other's language. This reason lead to the STEP3 design described below. STEP2 setup is depicted in the following image,

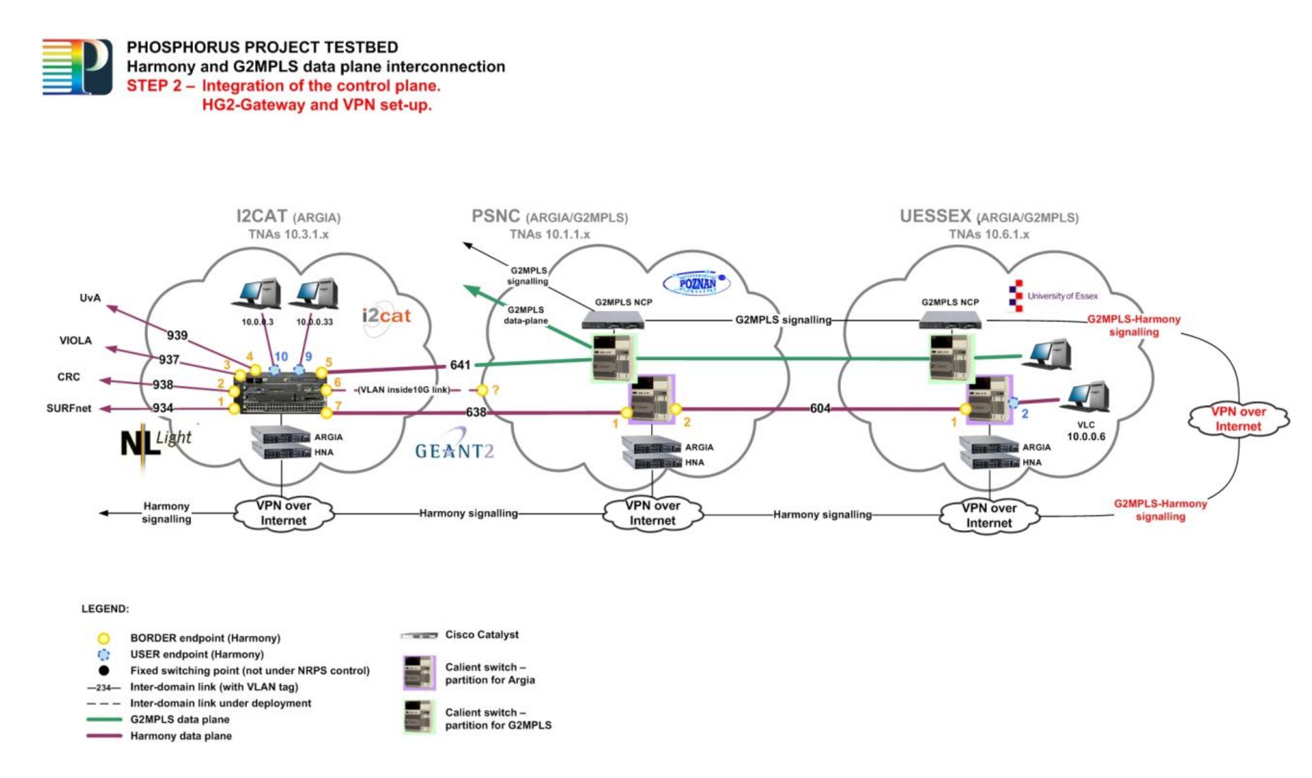

Figure 3.3: Harmony-G^{2}MPLS interconnection: step 2.

STEP3 (final setup): In this final step, a Harmony-G^{2}MPLS Gateway (HG^{2}GW) was implemented. HG^{2}GW is able to face Harmony on the one side and G^{2}MPLS on the other side and translate resource reservation requests coming from one to the other. This step required to set up the HG^{2}GW as the controller of a G^{2}MPLS domain, which was hosting all TNA name space created by addresses 10.1.2.0/24. STEP3 setup is depicted in the following image,

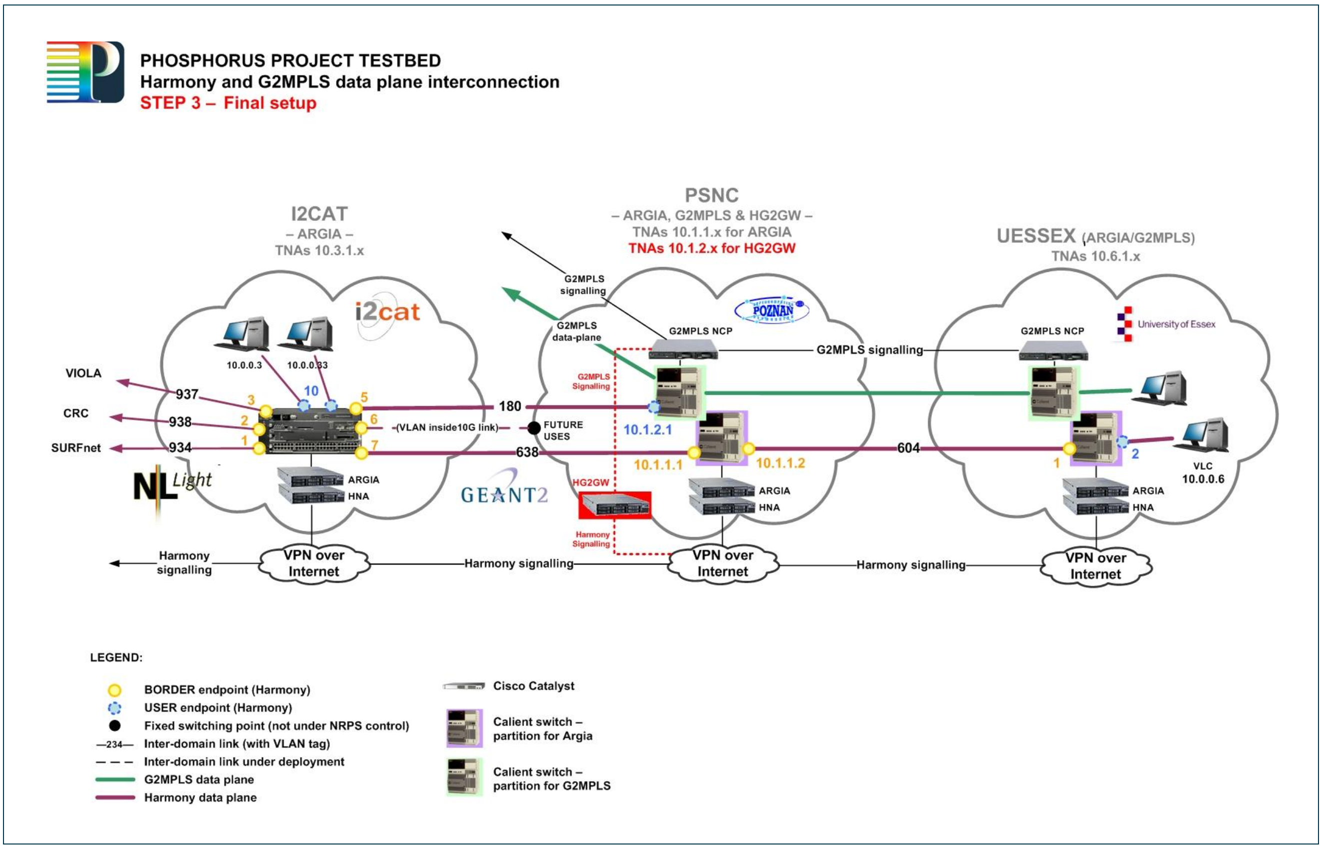

Figure 3.4: Harmony-G^{2}MPLS interconnection: step 3.

3.2 Test/Interoperability Scenario Setup and Workflow

On the Harmony side, the interface is implemented using Web Services while on the G^{2}MPLS side the external interface uses CORBA IDLs. The two main objectives of HG^{2}GW are:

Mapping of request and response data structures (cf. Section 2.2)

Mapping of synchronous and asynchronous method calls.

The following signalling workflow has been extracted from [Phosphorus-D2.9], showing a standard workflow to administrate a reservation crossing from a Harmony domain to a G^{2}MPLS domain.

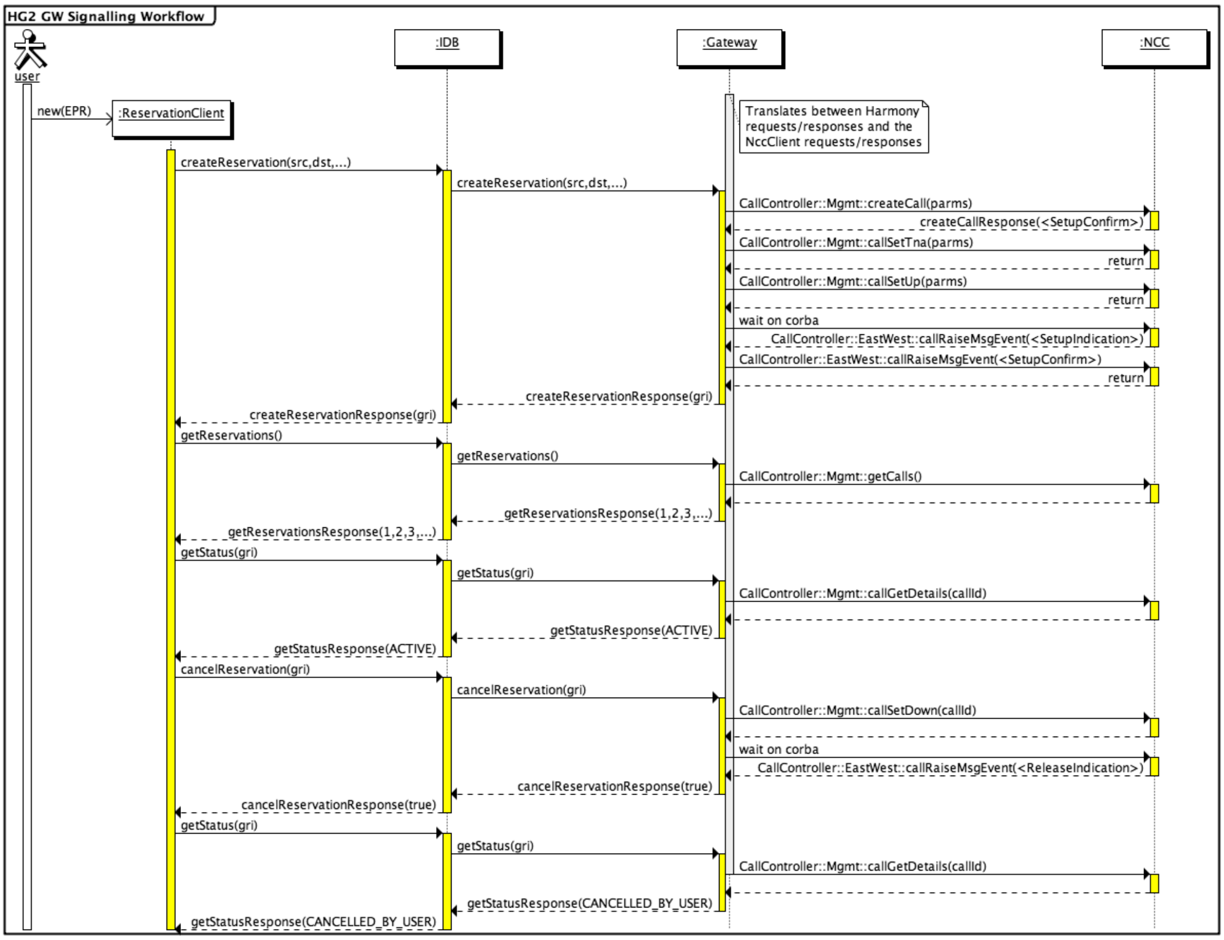

Figure 3.5: HG²GW Signalling Workflow – from Harmony to G²MPLS.

The flow shows the asynchronous method call mappings from one domain to the other:

createReservation: This call requests a path to the G^{2}MPLS domain. The gateway maps this method to the corresponding CORBA calls (createCall, callSetTna, callSetup) and waits to the reception of a callRaiseMsgEvent from the G^{2}MPLS Network Call Controller (NCC) since Harmony expects a confirmation Response of the success of the request.

getReservations: This call requests a list of the existing reservations. This request can be mapped to a single NCC request and returns a list of available reservations made in the G²MPLS domain.

getStatus: This call queries the status of a specific reservation to ensure that it is in an active (immediate reservation) or pending state (advance reservation). Again this request can be mapped to a single NCC request.

cancelReservation: This call cancels a previously requested reservation. Here again, the gateway maps the call into the corresponding CORBA call (callSetDown) and waits to the reception of a callRaiseMsgEvent acknowledging the cancelation. In general, a success message will be returned to Harmony.

3.3 Test Results

The HG^{2}GW has been tested in the Phosphorus test-bed by establishing a connection between Harmony (i2CAT) and G^{2}MPLS (PSNC) domains. Figure 3.6 shows the test-bed logical view used for the inter-domain tests.

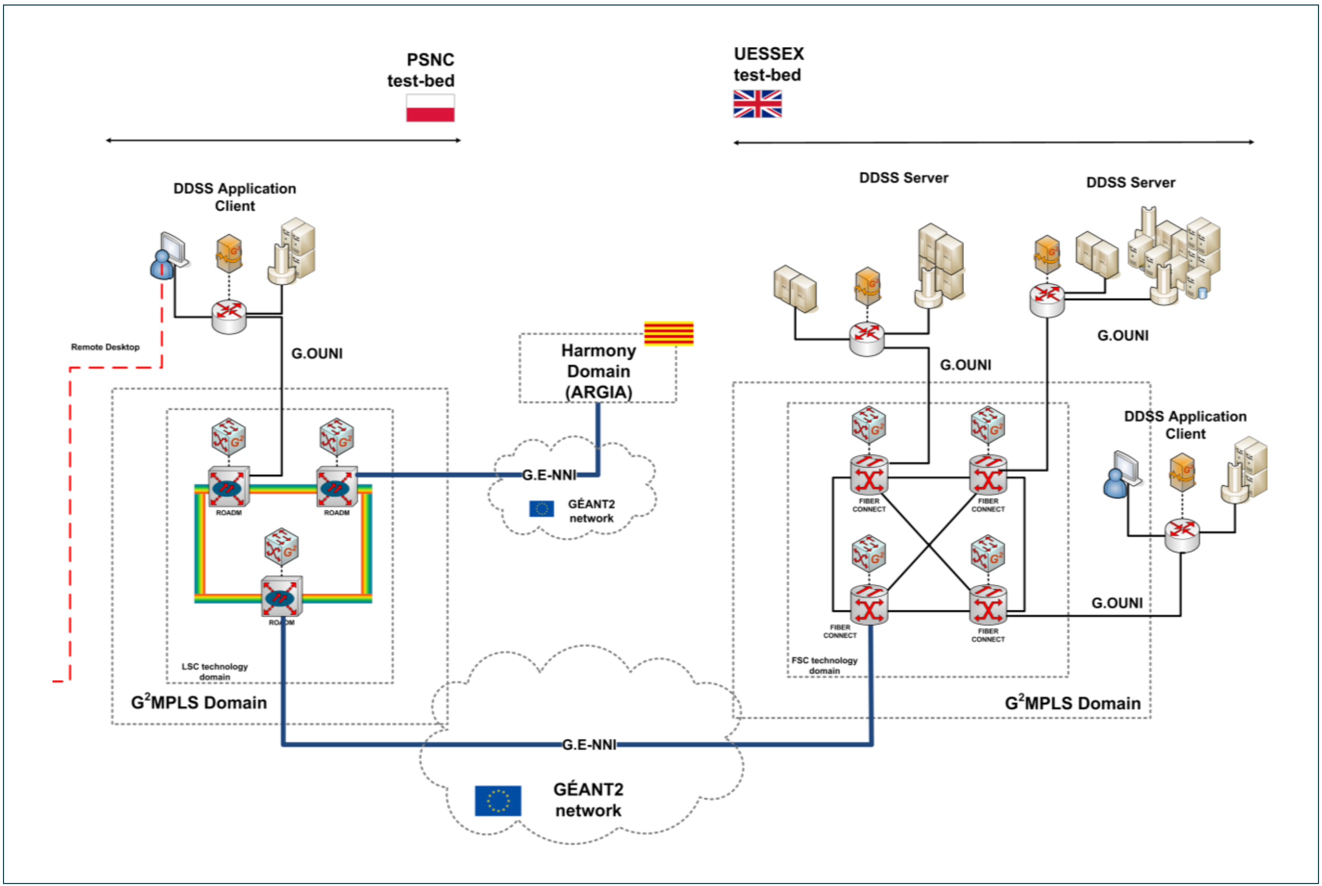

Figure 3.6: Test-bed logical view.

When testing a connection setup with direction Harmony → G^{2}MPLS, the HG^{2}GW prototype receives an incoming request (createReservation) from Harmony specifying as the source TNA the ingress TNA to the G^{2}MPLS domain and as destination TNA the end point of the connection or the G^{2}MPLS egress point. I2CAT Harmony IDB and PSNC G^{2}MPLS controllers are disposed as depicted in Figure 3.7 for test purposes. Since the HG^{2}GW has been implemented as part of the G^{2}MPLS stack, in these tests it has been placed at the same location as the G^{2}MPLS border controller that has a connection to the Harmony domain.

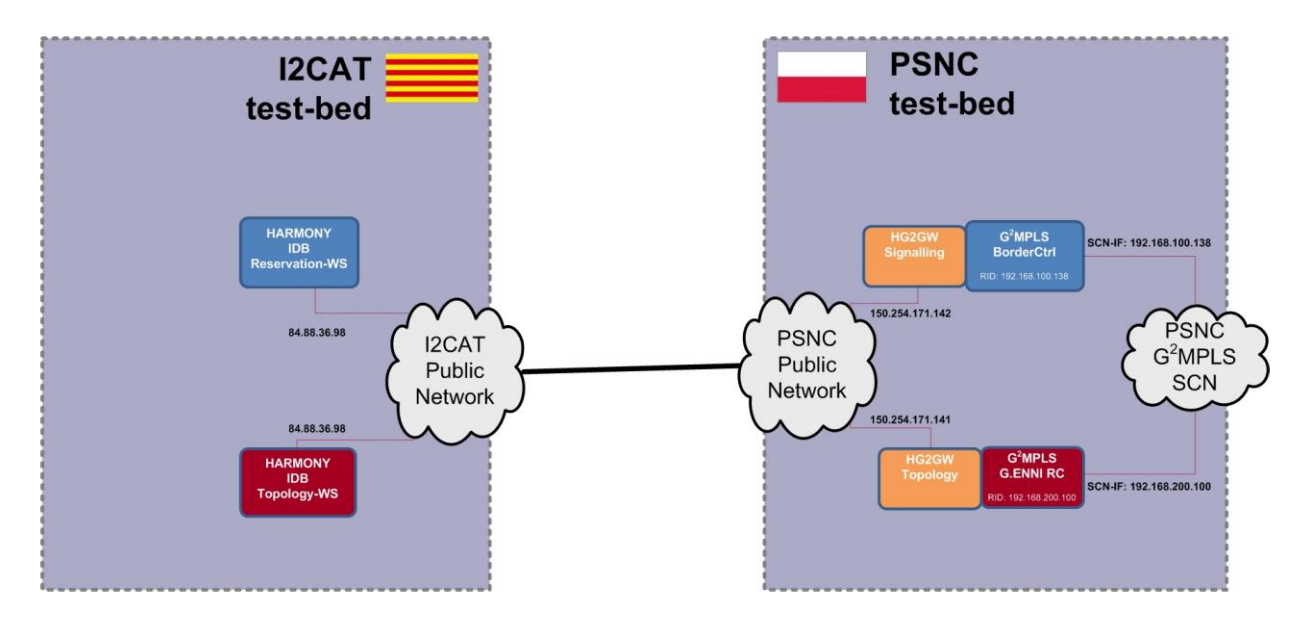

Figure 3.7: Harmony-G^{2}MPLS interconnection scenario.

As we can see in the following HG^{2}GW log, when the request is received, the HG^{2}GW maps the call parameters and sends the corresponding CORBA methods to the NCC. After failure or a confirmation of success, it acknowledges to Harmony informing about the status of the request.

```
2008/12/15 15:50:59 HG2GW: [DBG] nRes Call: createReservation

2008/12/15 15:50:59 HG2GW: [DBG] ServiceID = 1

2008/12/15 15:50:59 HG2GW: [DBG] srcTNA = 100.100.100.1

2008/12/15 15:50:59 HG2GW: [DBG] dstTNA = 20.20.20.1

2008/12/15 15:50:59 HG2GW: [DBG] Going to create a HG2GW CALL with localID 1

2008/12/15 15:50:59 HG2GW: [DBG] Sent callCreate through CORBA Mgmt interface

2008/12/15 15:50:59 HG2GW: [DBG] Created call with identity [OPERATOR SPECIFIC]

2008/12/15 15:50:59 HG2GW: [DBG]     - srcAddr        : (IPv4) 150.254.171.142/32

2008/12/15 15:50:59 HG2GW: [DBG]     - localId        : 0x0000000000000001
```

```
2008/12/15 15:50:59 HG2GW: [DBG] Sent callSetTna for iTNA through CORBA Mgmt interface

2008/12/15 15:50:59 HG2GW: [DBG] Sent callSetTna for eTNA through CORBA Mgmt interface

2008/12/15 15:50:59 HG2GW: [DBG] Sent callSetUp through CORBA Mgmt interface

2008/12/15 15:51:00 HG2GW: [DBG] Received callRaiseMsgEvent through CORBA EastWest interface

2008/12/15 15:51:00 HG2GW: [DBG] Sent callRaiseMsgEvent through CORBA EastWest interface

2008/12/15 15:51:00 HG2GW: [DBG] HG2GW call successfully created with localID 1

2008/12/15 15:51:00 HG2GW: [DBG] nRes Call: Returning createReservation
```

When a cancelReservation is received, the following HG^{2}GW log can be seen:

```
2008/12/15 15:55:13 HG2GW: [DBG] nRes Call: cancelReservation

2008/12/15 15:55:13 HG2GW: [DBG] ServiceID = 1

2008/12/15 15:55:13 HG2GW: [DBG] Going to cancel a HG2GW CALL with localID 1

2008/12/15 15:55:13 HG2GW: [DBG] Sent callSetDown through CORBA Mgmt interface

2008/12/15 15:55:13 HG2GW: [DBG] Received callRaiseMsgEvent through CORBA EastWest interface

2008/12/15 15:55:13 HG2GW: [DBG] HG2GW call successfully cancelled with localID 1

2008/12/15 15:55:13 HG2GW: [DBG] nRes Call: Returning createReservation
```

The other implemented calls (getStatus, isAvailable) perform in a similar way as createReservation

4 Conclusions and Outlook

This deliverable is focused on the implementation of the Harmony G²MPLS Gateway (HG²GW) to enable the automatic setup of inter-domain connections between Phosphorus Harmony and Phosphorus G²MPLS. The deliverable relies on previous deliverable D2.9 and describes the inter-working architecture and interoperability scenarios between the Harmony and G²MPLS systems. The document provides the first test results and identifies open issues that are drafted below.

Mainly two aspects are under current investigation that should be resolved with the future development. First, the implemented prototype must be enhanced to enable a bilateral signalling translation. Second, the automatic exchange and translation of dynamic routing information should be evaluated in more detail and must be implemented.

5 References

[PH-WP2-D2.1] Phosphorus deliverable D2.1, "The Grid-GMPLS Control Plane architecture".

[PH-WP2-D2.9] Phosphorus deliverable D2.9. "Design of Grid-GMPLS interworking with NRPS".

[PH-WP1-D1.3] Phosphorus deliverable D1.3, "NRPS southbound interfaces for standard GMPLS control plane".

[PH-WP1-D1.6] Phosphorus deliverable D1.6, "Planning and report on functional tests, and prototype for NRPS, GMPLS control plane and middleware interoperability".

6 Acronyms

E-NNI	External Network-to-Network Interface
G.OUN	Grid O-UNI
G²MPLS	Grid-enabled GMPLS
G-E.NNI	Grid E-NNI
G-I.NNI	Grid I-NNI
GMPLS	Generalized Multi Protocol Label Switching
GNS	Grid Network Services
GW	Gateway
HAL	Harmony Adaptation Layer
HG²GW	Harmony G²MPLS Gateway
HNA	Harmony NRPS Adapter
HSI	Harmony Service Interface
IDB	Inter Domain Broker
IDB-R	IDB for Routing
IDB-S	IDB for Signalling
NCC	Network Call Controller
NRPS	Network Resource Provisioning System
NSP	Network Service Plane
OSPF	Open Shortest Path First
O-UNI	Optical User Network Interface
SOA	Service-Oriented Architecture
TNA	Transport Network Address
WS	Web Service
WSDL	Web Service Description Language