26-05-2010

Deliverable DJ1.4.1: Virtualisation Services and Framework Study

Deliverable DJ1.4.1

Contractual Date:	31-12-2009
Actual Date:	26-05-2010
Grant Agreement No.:	238875
Activity:	JRA1
Task Item:	T1
Nature of Deliverable:	R (Report)
Dissemination Level:	PU (Public)
Lead Partner:	JANET
Document Code:	GN3-09-225v1.0
Authors:	M. Campanella (GARR), P. Kaufman (DFN), F. Loui (RENATER), R. Nejabati (University of Essex), C. Tziouvaras (GRNET), D. Wilson (HEANET), S. Tyley (DANTE)

Abstract

This deliverable presents the results of an initial comparative study of existing infrastructure virtualisation technologies and frameworks. It also presents the results of an initial analysis of NRENs' requirements for using infrastructure virtualisation technologies in the near future. Furthermore, this deliverable defines virtualisation services within the context of GÉANT and proposes an approach for their implementation within GÉANT and associated NREN infrastructures. Finally, it provides a plan for proof of concept and validation of the proposed virtualisation approach over a small testbed.

Table of Contents

Table of Figures

Table of Tables

Executive Summary

In the context of network and computing infrastructure, virtualisation is the creation of a virtual version of a physical resource (e.g. network, router, switch, optical device or computing server), based on an abstract model of that resource and often achieved by partitioning (slicing) and/or aggregation. A virtual infrastructure is a set of virtual resources interconnected together and managed by a single administrative entity.

This document aims to investigate potential uses and benefits of infrastructure virtualisation services for the GÉANT and NREN communities. It proposes a multi-layer, multi-domain and multi-technology virtualisation architecture suitable for NREN requirements, based on tools and software that have already been developed or are currently under development within the European research community.

The deliverable first presents a comprehensive comparative study of existing major activities, research projects and technologies addressing infrastructure virtualisation. The projects considered include European projects (FEDERICA, MANTICORE, Phosphorous, 4WARD), US projects (GENI, PlanetLab/VINI/OneLab), a Japanese project (AKARI) and a commercial cloud project (Amazon virtualisation). All the projects include infrastructure virtualisation at national and/or international level and some of them involve National Research and Education Networks (NRENs) and international connectivity. The study tries to provide a consistently structured assessment of different projects addressing the following points:

- Overview of the project and its objective.
- A definition of infrastructure virtualisation as understood by the project as well as an architectural overview of its virtualisation approach.
- User community.
- Overview of existing features and implementation of virtualisation for Layer 1, Layer 2, Layer 3 and computing resources.
- Multi-domain support of the virtualisation technology.
- Testbed implementation and availability.
- Current status and roadmap.

The results of the study conclude that the European research community, helped by the drive and commitment of the NRENs, has managed to achieve significant progress on infrastructure virtualisation technologies through projects such as FEDERICA, MANTICORE and Phosphorus. These projects are complementary and, combined together, they can provide virtualisation of Layer 1, Layer 2 and Layer 3 networks as well as

computing resources. Any proposal for GÉANT virtualisation services should therefore build on the developments and achievements of these projects.

This deliverable also presents the results of an initial study of NRENs' requirements for using infrastructure virtualisation technologies in the near future. The analysis reported here focuses on the requirements of three NRENs only. This is a pilot analysis; its results are expected to create the foundation and framework for a more comprehensive study to be carried out during 2010 covering as many NRENs as possible, as well as GÉANT. To carry out the requirements survey, a questionnaire was prepared in four sections as follows:

- Existing use of virtualisation technologies and services.
- Other potential applications.
- Areas of specific or strategic interest for application of virtualisation.
- Risk analysis.

The results provide a summary of how different NRENs plan to use virtualisation over the coming 1-3 years, their experiences so far, and their views on the cooperative use of virtualisation in GÉANT. They also indicate the existence of a unanimous requirement for virtualisation by NRENs, with each stressing a different aspect of virtualisation and related services, i.e. Layer 1, Layer 2, Layer 3 and computing virtualisation.

The current virtualisation technologies resulting from the projects mentioned above are still in their research and development stage. It is therefore not realistic to propose a specific solution to the NREN and GÉANT community. This report doesn't aim to promote a specific solution or framework for a technological proof of concept for GÉANT virtualisation services. Instead, it aims to propose a solution for integrating and interworking existing virtualisation mechanisms and solutions at different layers, leaving the choice of suitable virtualisation technologies to individual NRENs, while enabling them to offer multi-domain, multi-layer and multi-technology virtualisation services.

This deliverable proposes an initial, multi-layer and multi-domain infrastructure virtualisation mechanism based on a combination of solutions and tools developed by relevant EU projects, namely FEDERICA, MANTICORE and Phosphorus. Without reinventing the wheel, the proposition is to integrate existing Layer 1, Layer 2, Layer 3 and computing virtualisation tools both horizontally and vertically.

Finally, this deliverable drafts a plan for a pilot prototype implementation and verification of the proposed integrated virtualisation service mechanism. Because of time and resource limitations within JRA1 Task 4, the prototype implementation and proof of concept will be carried out on a very small scale using existing resources within Task 4 participants' facilities. Two virtualisation frameworks and two small-scale testbeds have been selected for prototype implementation and proof of concept testing (as shown in Figure 5.1 on page 73): the University of Essex Layer 1 testbed (small scale) deploying Phosphorus (UCLP) and the HEANET Layer 3 testbed (small scale) deploying MANTICORE.

Network virtualisation is a relatively new concept. As with any innovation, there will undoubtedly be some aspects, both benefits and problems, that only emerge over time. Evaluating the advantages, disadvantages and risks of virtualisation compared to traditional operation, particularly with regard to security, will therefore form a key part of future JRA1 T4 work. At this stage in the development lifecycle, however, the majority agree

on the benefits and necessity of network virtualisation, as demonstrated by the EU projects reviewed in this document, none of which has so far reported significant drawbacks.

1 Introduction

Current developments and technical enhancements in transport network technologies, network management and control planes, multi-core processing, cloud computing, data repositories and energy efficiency are driving profound transformations of NRENs' (National Research and Education Networks') network infrastructures and their users' capabilities. These technological advances are driving the emergence of ever more demanding high-performance and network-based applications with strict IT (e.g. computing and data repositories) and network resource requirements. Examples of these applications include: ultra-high-definition remote visualisation and networked high-performance supercomputing infrastructures. These types of applications often require their own dedicated network and IT resources tailored to their strict computing and network resource requirements. As these types of collaborative and network-based applications evolve, addressing the needs of a wide range of users in the NREN community, it is not feasible (for scalability reasons, among others) to set up and configure dedicated network and computing resources for each application type or category. Consequently, NRENs need to deploy an infrastructure management mechanism able to support all application types optimally, each with their own access, network and IT resource usage patterns. Any solution providing such an infrastructure management mechanism has to address the following challenges:

- Increase in the number of users/applications and rapid increase in available bandwidth for users beyond 1 Gbps.
- Emergence of new scientific applications requiring 10 G or even 100 G connectivity e.g. LHC and Radio Astronomy.
- Partitioning of physical network and IT infrastructures for providing secure and isolated application-specific infrastructure.
- Migration towards a full range and large-scale convergence of IT and network services.
- Energy-efficiency in networking and computing.

A key issue in addressing these challenges is efficient network and computing resource utilisation and sharing within the current and future NREN infrstructure.

This deliverable aims to investigate potential uses of virtualisation to address the challenges listed above. In the context of network and computing infrastructure, virtualisation is the creation of a virtual version of a physical resource (e.g. network, router, switch, optical device or computing server), based on an abstract model of that resource and often achieved by partitioning (slicing) and/or aggregation. A virtual infrastructure is a set of virtual resources interconnected together and managed by a single administrative entity. The deliverable proposes a

multi-layer virtualisation architecture suitable for NREN requirements based on tools and software that have already been developed or are currently under development within the European research community.

The GÉANT3 JRA1 Task 4 work and, consequently, this deliverable, investigate the application of virtualisation technology for the GÉANT community within the framework of Infrastructure as a Service (IaaS) [IaaS]. IaaS is a promising paradigm that enables NRENs to provide infrastructure resources like routers, switches, optical devices, Internet Protocol (IP) networks, and computing servers as a service to their user communities. It comprises a set of software and tools that allows virtualisation of infrastructure by means of partitioning (slicing) and/or aggregation of infrastructure resources (i.e. network elements and computing resource). Resource virtualisation is an effective method of efficient infrastructure resource sharing among users and applications and therefore its immediate benefit for NRENs is to increase resource utilisation efficiency. Virtualisation can also potentially enable NRENs to offer remote access and control of virtual infrastructure elements (slices of real physical elements) to their user organisations through web services. By using virtualisation services, users can control their own virtual infrastructure. This provides an effective mechanism for secure and isolated application-specific virtual infrastructures to share physical infrastructure. Furthermore, virtualisation can potentially provide a new level of flexibility to the NRENs, as their infrastructure can scale up or down following user/application requirements, thereby minimising the cost of operating the infrastructure (both the capital and the operational expenditures).

This deliverable provides an initial report on a comparative study of existing virtualisation technologies and framework; an analysis of NRENS' requirements for infrastructure virtualisation; and a proposal for implementing a technological proof of concept for virtualisation services in GÉANT. To this end the deliverable has been organised as follows:

- Section 2 presents a detailed comparative study of the main current projects and initiatives addressing virtualisation technology.
- Section 3 presents the results of an analysis of three NRENs' requirements for deploying infrastructure virtualisation technologies in the near future.
- Section 4 proposes an initial multi-layer and multi-domain infrastructure virtualisation mechanism based on a combination of solutions and tools developed by relevant EU projects.
- Finally, Section 5 outlines a plan for prototype implementation and initial testing of the proposed virtualisation approach. This is expected to provide the foundation and basic information for the detailed design of a GÉANT3 virtualisation framework. In addition, T4 will undertake a more general evaluation of the advantages, disadvantages and risks of virtualisation compared to traditional operation.

1.1 Benefits and Challenges

Network virtualisation is a relatively new concept. As with any innovation, there will undoubtedly be some aspects, both benefits and problems, that only emerge over time. Evaluating the advantages, disadvantages and risks of virtualisation compared to traditional operation will therefore form a key part of future JRA1 T4 work. At this stage in the development lifecycle, however, the majority agree on the benefits and necessity of network virtualisation, as demonstrated by the EU projects reviewed in this document, none of which has so far reported

significant drawbacks. The assessment given in a recently published IETF Internet Draft [IETFProbStatement] is representative:

> Network virtualization guarantees isolation of network services by creating isolated logical network environments between users belonging to separate groups.
>
> In the current networks, the network service providers hardly offer resources encompassing the physical capability of the resources. However, by leveraging network virtualization, it is possible to provide high performance resources for users by logically aggregating multiple resources into single resource. Therefore, a logical network consisting of requested resources can guarantee users' performance requirements . . . coexistence of multiple logical networks is one of the fundamental motivations behind network virtualization. Legacy networks hardly provide multiple networks, but multiple virtual networks can be created in a shared physical network with the same resources.

At the same time, developers and researchers, JRA1 T4 among them, are not unaware of the challenges and technical issues to be addressed in order to achieve viable and successful realisations of network virtualisation. These include the following [NVChallenges]:

- **Interfacing**: providing a standard interface by which a user can request the creation and modification of virtual infrastructure provided by a physical infrastructure provider.
- **Resource discovery, brokering and monitoring**. Complex algorithms are required for efficient discovery and allocation of virtual resources (physical resource segments) to different virtual infrastructures as well as monitoring their performance.
- **Admission control and usage policing**. When establishing a virtual infrastructure, its user may require specific guarantees for attributes and characteristics of virtual infrastructure. Complex and accurate accounting, and admission control and distributed usage policing algorithms, are therefore needed to ensure the guaranteed performance can be delivered.
- **Security and privacy**. Isolation between coexisting virtual infrastructure can only provide a certain level of security and privacy through the use of secured tunnels, encryptions, and so on; but it does not obviate the prevalent threats, intrusions, and attacks to the physical layer and virtual nodes. In addition to that, security and privacy issues specific to network virtualisation must also be identified and explored. For example, programmability of the network elements can increase vulnerability if secure programming models and interfaces are unavailable. Further, virtualising infrastructure makes it possible to create and recreate individual computing resources on a large scale, and therefore to replicate vulnerabilities or otherwise exploitable infrastructure on an equally large scale.
- **Failure handling**. Failures in the underlying physical infrastructure components can give rise to cascading series of failures in all the virtual nodes directly hosted on those components. Detection, propagation, and isolation of such failures, as well as protection and restoration from them, are all open research challenges.
- **Interoperability issues**. End-to-end virtual infrastructures can span across multiple administrative domains, each using possibly heterogeneous networking and IT technologies and management frameworks. Enabling virtualisation in each of these technologies requires specific solutions for provisioning, operation, and maintenance. Interactions between such contrasting underlying

infrastructures, while providing a generic and transparent management interface for users to easily compose and manage irtual infrastructure, remains a daunting task.

2 Overview of Existing Virtualisation Technologies and their Usage

2.1 Introduction

This section provides a comprehensive overview of existing major activities, research projects and technologies addressing infrastructure virtualisation. The projects considered include European projects (FEDERICA, MANTICORE, Phosphorous, 4WARD), US projects (GENI, PlanetLab/VINI/OneLab), a Japanese project (AKARI) and a commercial cloud project (Amazon virtualisation). All the projects include infrastructure virtualisation at national and/or international level and some of them involve National Research and Education Networks (NRENs) and international connectivity.

Detailed descriptions of all the EU-based projects are provided except 4WARD, where the process of collecting detailed information is ongoing. For AKARI and GENI, the process for exchanging information between the relevant activities within those projects and GN3 JRA1 Task 4 is in its final stage. It is expected that a detailed overview of 4WARD, AKARI and GENI will be provided in the second deliverable, DJ1.4.2, due March 2012.

The review of each project has been organised to cover:

1. Introduction – an overview of the project and its objective.
2. Architecture overview – a definition of infrastructure virtualisation as understood by the project as well as an architectural overview of its virtualisation approach.
3. User community – a description of the user group(s) at which the project is aimed.
4. Mechanisms for providing virtualisation – an overview of existing features and implementation of virtualisation for Layer 1, Layer 2, Layer 3 and computing resources.
5. Multi-domain support – a statement of whether the virtualisation technology can be applied in a multi-domain environment.
6. Testbed implementation and availability – a description of the virtualisation testbed and test scenario, if they exist, is provided.
7. Current status and roadmap – roadmap and future plans with respect to virtualisation.
8. References – details of sources cited in the overview (these are also given in the References section at the end of the document on page 75.)

Finally this section concludes with a comparative table summarising the virtualisation capability and features of the projects reviewed.

2.2 FEDERICA

2.2.1 Introduction

FEDERICA (Federated E-infrastructure Dedicated to European Researchers Innovating in Computing Network Architectures) [FEDERICA] is a European Community 7th Framework Program co-funded project which started on 1st January 2008 and will last 30 months. The project is coordinated by GARR and involves 20 partners, including 9 National Research and Education Networks (NRENs), TERENA, DANTE, universities, research institutions and vendors.

The main goal of the FEDERICA project is to support research experiments on new Internet architectures and protocols. In particular, it is expected to contribute in the following activities:

- Creating a versatile, scalable, European wide "technology-agnostic" e-infrastructure that can interconnect with the Internet and other infrastructures. Dedicated "slices" of its infrastructure will be assigned to different user groups.
- Supporting research into the virtualisation of e-infrastructures and developing experience and solutions for the management and control of distributed networking and computing virtual resources.
- Identifying users and their requirements.
- Facilitating technical discussions amongst specialists, disseminating knowledge and NREN experience, and providing user training and support.
- Enabling the smooth implementation of a new inter-domain service layer.
- Providing preliminary information and results for the next generation of NREN networks, and linking with GÉANT.
- Contributing with real test cases and results to standardisation bodies, e.g., IETF, ITU-T, OIF, IPsphere.

On the other hand, the FEDERICA project will not consider the following objectives:

- Extended research, e.g., direct advanced optical technology evaluations (for example, 100 GE).
- Support of Grid applications.
- Develop architectures and concepts in the context of "Future Internet".
- Offer connectivity, in particular transit capacity between external entities (this service is available from the production NREN infrastructure).
- Offer raw computation power.

2.2.2 Architecture overview

The combination of virtualisation techniques with network control mechanisms is a unique aspect of FEDERICA. This enables the creation of dedicated "slices" of its infrastructure, which can be used in parallel by different research groups, while granting varying degrees of control and avoiding disruption.

Slices are created by leveraging virtualisation capabilities enabled in the physical infrastructure resources, i.e. FEDERICA physical switches and servers (as depicted in Figure 2.1). Each slice consists of a set of virtual networking and computing resources which can be configured and interconnected according to researchers' needs and requests. Though the capability to run virtual overlay networks, e.g., Layer 2 or Layer 3 VPNs (Virtual Private Networks), is already well consolidated in many production networks, in FEDERICA researchers will be allowed to access control and configuration capabilities down to the lowest possible network layer and even to add virtual resources to their slices. Moreover, they can use slices to carry out disruptive experiments generally not allowed in a large production networks.

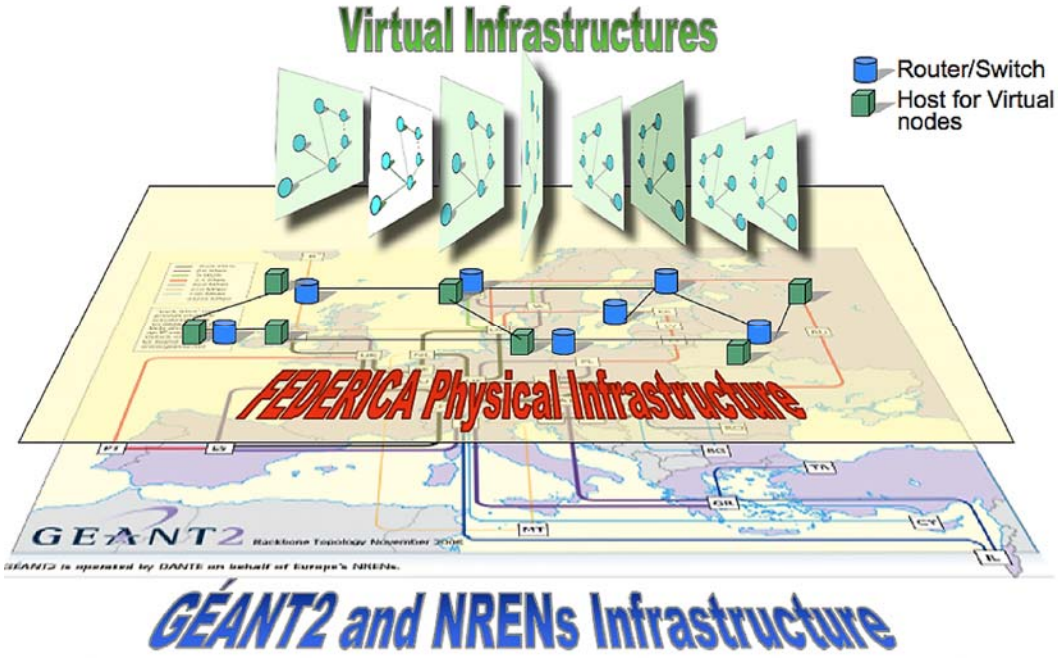

Figure 2.1: The FEDERICA infrastructure: graphical representation

The FEDERICA e-infrastructure is based upon Gigabit Ethernet circuits, transmission equipment and computing nodes capable of virtualisation.

The design of the FEDERICA e-infrastructure was based upon the following principles:

- Neutral and transparent to the types of protocols, services and applications that may be tested, while allowing disruptive experiments to take place without adverse effect on existing production networks.
- Able to be interconnected or federated with other e-infrastructures and also to host researcher hardware and applications in compliance with the policies provided the User Policy Board.

- Devoted to research on the Internet of the future and on virtual distributed systems and not intended to be used to provide raw computing power or permanent European connections.
- Take advantage of virtualisation both in computing and in networking, in order to allow slicing of physical resources.

The FEDERICA infrastructure is built over existing NREN networks and leverages GÉANT's end-to-end connection service GÉANT Plus. As shown in Figure 2.2, it consists of 12 Points of Presence (PoPs) interconnected by 18 x 1 Gigabit Ethernet links. Each PoP hosts one FEDERICA L2/L3 switch (Juniper MX480 or EX3200) and several computing elements, i.e. servers (Sun X2200 M2) locally connected to the switches by means of 8 x 1 Gigabit Ethernet links each. Further details of the FEDERICA e-infrastructure are available in Deliverable DSA1.1 "FEDERICA Infrastructure" [FEDERICADSA1.1].

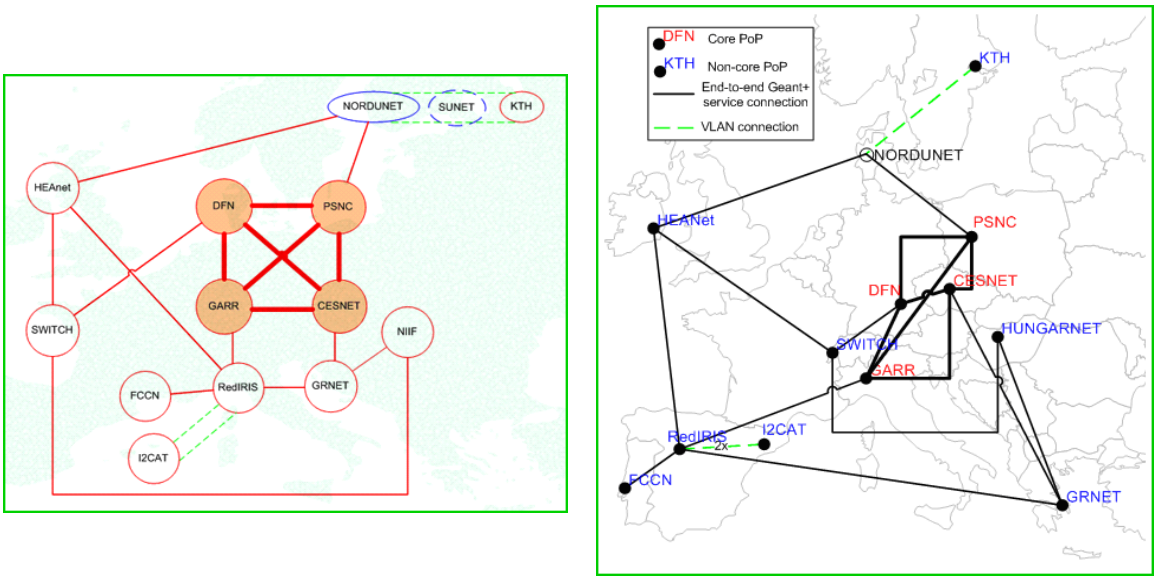

Figure 2.2: The FEDERICA infrastructure topology: simplified schema (left) and map view (right)

The FEDERICA infrastructure is aimed at providing slices to users. Each slice is composed of a set of virtual nodes, e.g., virtual switch and logical routers implemented in the FEDERICA switches, and virtual machines hosted in the virtualisation nodes (VNs), i.e. servers with virtualisation capabilities. Virtual nodes within a slice are interconnected by means of virtual Ethernet circuits, which can span multiple physical switches and links. These circuits are provided by using VLANs or MPLS L2 LSPs, and can also be set up with guaranteed bandwidth, hardware-based traffic shaping, and QoS functionality available in switches and nodes.

Slices are accessible either directly or through a gateway host from anywhere via the Internet. The user does not need then to move from their office to use a slice. At the present time, access to resources is undertaken through SSH and Keyboard Video Mouse (KVM) redirection tunnels, but the plan is to eventually allow slices to be controlled through web services. The FEDERICA infrastructure also has monitoring facilities that can be utilised to collect relevant information for diagnostic purposes. Monitoring information may be requested on the resources used in a slice.

2.2.3 FEDERICA-related research activities

Different FEDERICA-related frameworks, that is, projects based on similar concepts and implementations and research activities on the future Internet, are currently ongoing. In particular, because of some similarities, it is important for FEDERICA to liaise with PlanetLab [PlanetLab] and the GENI/VINI [GENI] [InVINIVeritas] initiatives, while both the IaaS framework [IaaS] and the IPsphere [IPsphere] framework are highly relevant to FEDERICA's aims, enabling the re-use and complementation of many of their components.

2.2.3.1 *IPsphere*

In particular, IPsphere Forum liaison and potential inter-working between the IPsphere Forum and FEDERICA are ongoing activities coordinated by FEDERICA, JRA2 T3 and NA3.

The IPsphere framework has been developed to enable providers to concurrently optimise flexibility and efficiency by focusing on the translation of a generalised service offering into a set of generalised resource commitments to meet the overall service goals. This "meet-in-the-middle" approach to service management allows providers to address their respective priorities in whatever manner they determine to be necessary and according to their specific circumstances. The IPsphere framework is network independent (and so technology and vendor agnostic), multi-domain, and constitutes a Service-Oriented Architecture (SOA) layer that can be integrated in any existing framework.

The IPsphere framework has strong commonalities with the service layer being explored in FEDERICA for the orchestration of virtualised resources from multiple stakeholders. In particular, integration between the IPsphere service framework and the FEDERICA resources management system (e.g., the MANTICORE implementation in the IaaS Framework) can be developed in order to allow users of IPsphere to compose resources coming from FEDERICA into an IPsphere service.

This integration requires FEDERICA use case(s) to be translated into one or more service template(s) in IPsphere, and FEDERICA virtualisation building blocks to be translated into IPsphere element templates.

The system needs to control (logical) router devices (initially based on Juniper architecture). Besides a ready-made (logical) IP network, users will also be able to integrate logical routers into their own configurations and profit from the logical resources. To achieve this objective, the current version of IaaS framework (i.e., the IaaS framework based on UCLP) will be enhanced to support the logical routing feature and the Application Programming Interface (API) written in Extensible Markup Language (XML), starting from the Juniper system. In the future this IP (Layer 3) web service will be integrated with existing or new web services for Layer 0, 1 and 2 and, if needed, layers above Layer 3.

Moreover, there is an opportunity to use the IPsphere framework to provide more sustainability in the tool being used, as well as to ease interoperability and network integration in many different environments, and FEDERICA is potentially a use case or a set of use cases of IPsphere Forum. In other words, FEDERICA could leverage the IPsphere Forum by defining its own set of use cases with very specific technical and business definitions, which could be different from those used by commercial operators.

The first version of the prototype for the interoperability between FEDERICA's IP network slices and other IP domains by means of the IPsphere framework has been developed. A detailed description of the FEDERICA and IPsphere framework inter-working can be found in FEDERICA deliverable DNA2.2 "FEDERICA User Community and Requirements" [FEDERICADNA2.2]; details concerning the prototype development can be found in "Prototype for the interoperability between FEDERICA slices and other IP domains by means of the IPsphere Framework" [FEDIPsphere].

2.2.4 User community

The target users of the FEDERICA infrastructure are researchers actively engaged in research on networking, who use networks not just as the tool, but primarily as the subject of their work. User groups will include EC projects, research groups in universities or research centres, equipment manufacturers and telecommunications research labs, or even individuals (e.g., PhD students). Training for users is also included in the project. The FEDERICA infrastructure will cover a significant part of Europe through the participating NRENs. Access to the infrastructure is not limited to the participating countries, but can be granted to any user with an Internet connection.

Users of the FEDERICA infrastructure can be divided into contributors and consumers:

- Contributors are able to modify, in a controlled way, their allocated virtual slice, i.e. its properties, configuration and software. Users wishing to test new network protocols or technologies might request a number of virtual routers or switches, or virtual machines interconnected by Ethernet circuits in a suitable topology. Virtual machines are available to the user for software and configuration upload including open source router and end node images.
- Consumers are the users, who are simply using a FEDERICA slice or layer to do higher-layer or application-layer testing. In this case, for example, users can request a slice including a working IP-routed network configured according to their needs.

Access to the infrastructure is normally free of charge and subject to compliance with an Acceptable Use Policy. A User Policy Board is responsible for accepting and prioritising users' requests. Each user group is assured testing privacy and results will not be accessible by other users. The FEDERICA project requests explicit feedback from its users.

The project had its launch event at the end of November 2008. Users and projects that have been approved by the FEDERICA User Policy Board (UPB) and are currently ongoing are:

- OneLab [OneLab] and monitoring testing (ELTE Hungary).
- Openflow tests (Stanford, Germany, Sweden, Italy).
- Monitoring (Czech Rep. – Internal).
- Phosphorus project [Phosphorus].

Pending requests from Ireland, Italy, Spain and Germany are currently under evaluation. Many of the requests relate to interconnection capabilities between initiatives and laboratories; some others concern optical testing.

Further details on user proposals can be found in Deliverable DNA2.2 “FEDERICA User Community and Requirements” [FEDERICADNA2.2].

External projects and users are encouraged to make use of the FEDERICA infrastructure. Proposals need to be submitted to the User Policy Board (UPB) for a usage agreement (see the FEDERICA User Information Kit [FEDERICAUIK] for details). The procedure is required so that the resources can be allocated in the best way to ensure separation, reproducibility and optimisation of the infrastructure. It is also necessary for new users to sign the Acceptable Use Policy, and to allow a few days for the slice to be set up.

2.2.5 Mechanisms for providing virtualisation

2.2.5.1 *Implementation of virtualisation on Layer 3*

Layer 3 virtualisation is achieved by:

- Junos virtualisation features available in the FEDERICA L2/L3 switches. In particular, there are two different levels of L3 virtualisation:
  - Virtual router, which basically makes it possible to run multiple and parallel instances of the same routing protocol (i.e., routing instances in the Junos terminology).
  - Logical system, which consists of the virtualisation of all router processes, e.g., remote access and control, MIBs, etc., and not only routing protocol processes. Logical system effectively corresponds to slicing a physical router into multiple virtual ones.
- Software routers, i.e., a virtual machine with some software installed that makes it work as a router. The software is installed in Virtual Machines (VMs) and hosted in the Virtualisation Nodes (VNs). Examples of such software are XORP, Quagga, VYATTA, BIRD.

2.2.5.2 *Implementation of virtualisation on Layer 2*

Layer 2 virtualisation is achieved by:

- Junos virtualisation features available in the FEDERICA L2/L3 switches. In particular, virtual switches can be created in order to make a set of physical or logical (VLAN) interfaces act as though they belong to the same Ethernet switch.
- VMware ESXi virtual switches (vSwitches), which are created by the virtualisation software installed in VNs. vSwitches are software Ethernet switches which link the VN physical interfaces with the virtual NICs of the Virtual Machines hosted in the VNs. Each VN can host multiple vSwitches and vSwitches inside VNs can be interconnected.
- Software switches and bridges installed in Virtual Machines and hosted in VNs, e.g., Linux brctl package.

2.2.5.3 *Implementation of virtualisation on Layer 1*

Layer 1 virtualisation is currently not available in FEDERICA.

2.2.5.4 *Implementation of computing virtualisation*

VMware ESXi enables virtualisation in computing nodes (VNs). Multiple Virtual Machines can be hosted within the same VN.

2.2.5.5 *Management of virtualised infrastructure*

The FEDERICA Network Operations Centre (NOC) is in charge of managing the FEDERICA e-infrastructure along with NREN staff for local operation support. Currently, the infrastructure is mainly managed by traditional and well-known techniques and tools already used in the NREN production networks, such as:

- Command Line Interface and web-based management tools available in Junos for switching equipment and in VMware ESXi for servers.
- SNMP- and perl-scripting-based monitoring system G3 [MonWithVR, ExtendingMANTICORE], which has been developed by CESNET (the Czech NREN operator) and customised to monitor the FEDERICA physical and virtual infrastructure.
- RT (Request Tracker) open source Trouble Ticket System.

However, software prototypes for slice-oriented provisioning, management and monitoring tools are currently under development within the project. In particular, an evolution of MANTICORE [ExtendingMANTICORE] is being developed mainly by i2cat in order to provide a unified software solution that delivers Infrastructure as a Service to router devices based on IaaS framework architecture. It will allow the infrastructure owner to manage the physical infrastructure by creating virtual slices and enabling infrastructure end users to control them. It will offer an intuitive GUI to manage all the devices and a persistent database for the information. Currently, a first version of the tool has already been internally released. This release offers the possibility to create and manage virtual IP networks on Juniper router/switches and VMware servers. The provisioning and management of Virtual Machines leverage VMware VI API as the communication interface with VMware ESXi.

Some screenshots of the MANTICORE extended GUI are shown in Figure 2.3.

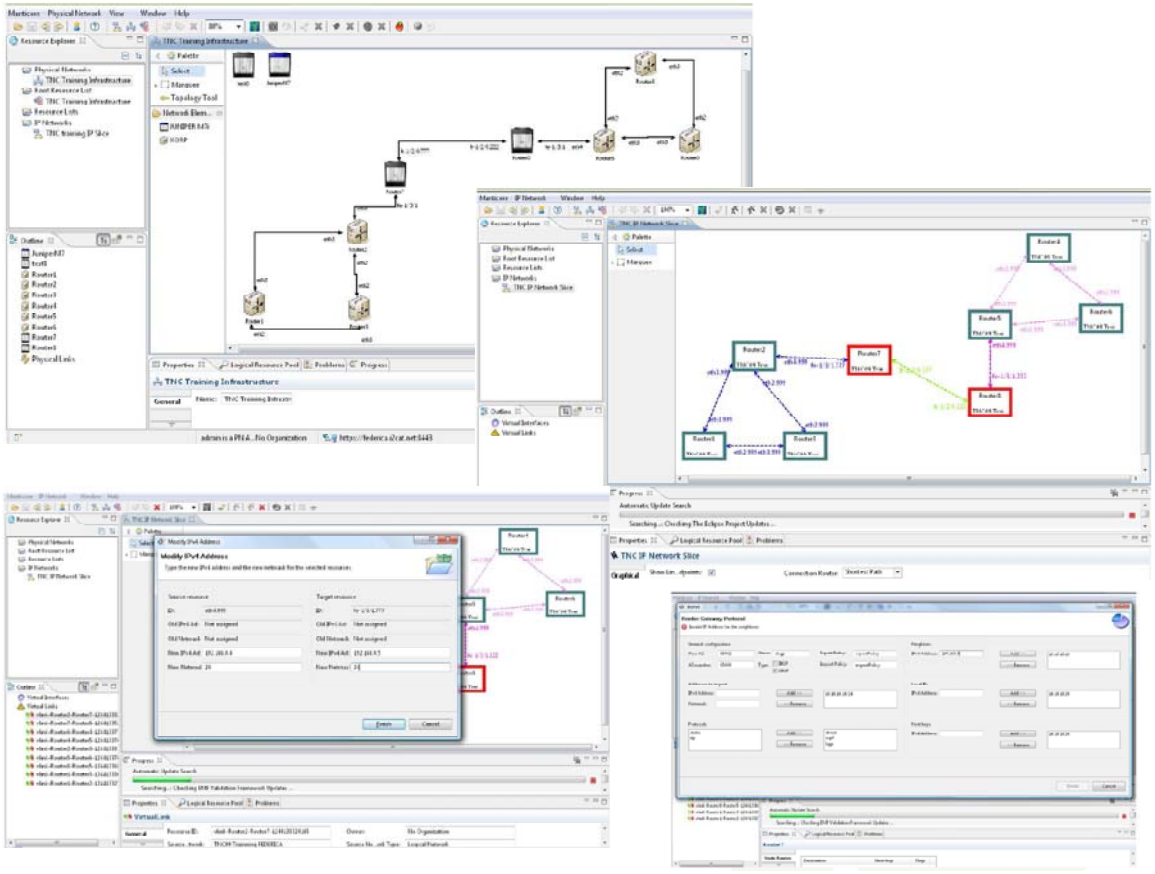

Figure 2.3: GUI screenshots of MANTICORE extended tool for FEDERICA resource management

2.2.5.6 *Control of virtualised infrastructure*

Within a FEDERICA slice, users are allowed to gain full control of the virtualised equipment, i.e., the virtual systems constituting the slice. Depending on the type of virtualised equipment, this capability allows users to:

- Leverage the control plane implementations available in the virtual systems which have been requested, configured and assigned to the users, e.g., Junos virtual routers or switches instantiated in Juniper boxes or other software-based equipment installed in VMware Virtual Machine such as Quagga or Xorp.
- Set up, deploy and test their own control plane within a slice by installing the developed control plane in user-provided software equipment installed in VMware VMs.

2.2.5.7 *Implementation of user interface*

As depicted in Figure 2.4, for the time being, FEDERICA users can access their own slices through a Virtual Slice Management Server (VSMS). This server is an authentication and authorisation proxy and it allows users to access the slice management network, i.e., a LAN which includes all the management interfaces of the slice virtual devices, e.g., CLI of logical or software router or the interface where KVM redirection of a Virtual Machine is running.

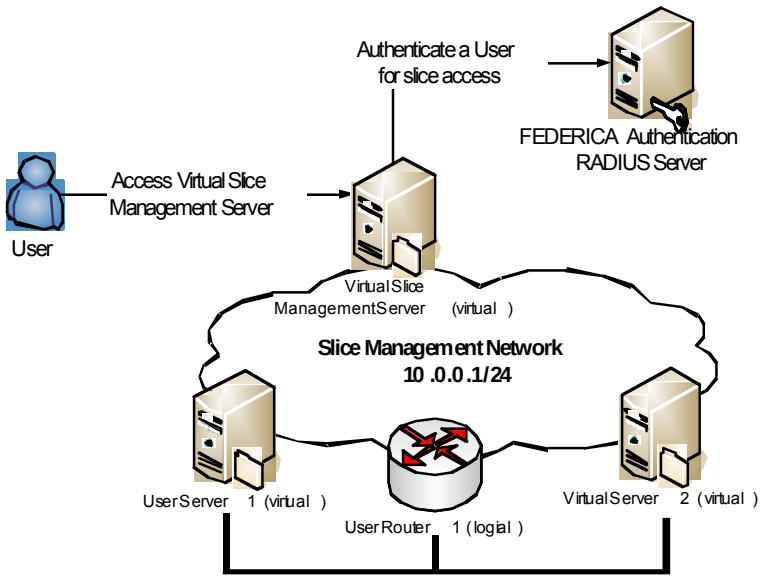

Figure 2.4: The FEDERICA user interface

2.2.6 Multi-domain support

FEDERICA slices fully support virtualised multi-domain environments. In particular, either a single slice can be configured as a multi-domain network, or different single-domain slices can be interconnected to build a multi-domain environment.

Moreover, even if the FEDERICA physical infrastructure is a single-domain network, in principle, and upon user policy board (UPB) approval, it is open to be interconnected with other e-infrastructure and testing facilities in order to set up physical multi-domain scenario.

2.2.7 Testbed implementation and availability

The FEDERICA infrastructure is mainly aimed at providing testing infrastructures and testbed environments to its users. For the time being there is no testing slice pre-configured in the infrastructure that can be used as an example by potential users. Projects and research groups interested in requesting a slice are asked to follow the application procedure described in the FEDERICA UPB User Information Kit [FEDERICAUIK], which also includes further details concerning access rules and guidelines, Acceptable Use Policy and resource description.

2.2.8 Current status and roadmap

Currently, the physical infrastructure has been set up almost everywhere, and only very few PoPs are still missing. The core infrastructure has been operational since the beginning of 2009, and there are already 5 slices configured for testing and first users. The main activity now consists in ensuring the infrastructure is used

and collecting feedback from the user community. Moreover, outcomes and software prototypes from the service and research activities are expected to ease and improve the infrastructure management and slice provision process.

2.2.9 References

The FP7 project FEDERICA is partly supported by the European Commission under the Grant Agreement No. RI-213107. The authors acknowledge the fundamental contribution of all project partners.

[ExtendingMANTICORE]	A. Berna, E. Grasa, S. Figuerola, "Extending MANTICORE to Manage IP and Virtual Machine Slices in the FEDERICA project", Proceedings. of TERENA Networking Conference 2009, TNC2009, June 2009, Malaga, Spain http://tnc2009.terena.org/core/getfile.php?file_id=319
[FEDERICA]	The FEDERICA Project https://www.fp7-federica.eu
[FEDERICADNA2.2]	Deliverable DNA2.2: "FEDERICA User Community and Requirements" http://www.fp7-federica.eu/documents/FEDERICA-DNA2.2.pdf
[FEDERICADSA1.1]	Deliverable "DSA1.1: "FEDERICA Infrastructure" http://www.fp7-federica.eu/documents/FEDERICA-DSA1.1.pdf
[FEDERICAUIK]	FEDERICA User Information Kit - http://www.fp7-federica.eu/users/users.php
[FEDIPsphere]	J. Pons-Camps, S. Figuerola, E. Grasa, "Prototype for the interoperability between FEDERICA slices and other IP domains by means of the IPsphere Framework", Proceedings. of TERENA Networking Conference 2009, TNC2009, June 2009, Malaga, Spain http://tnc2009.terena.org/core/getfile.php?file_id=410
[GARR]	GARR, the Italian Academic & Research Network http://www.garr.it
[GENI]	http://www.geni.net
[IaaS]	Infrastructure as a Service http://www.iaasframework.com
[InVINIVeritas]	"In VINI Veritas: Realistic and Controlled Network Experimentation", A. Bavier, N. Feamster, M. Huang, L. Peterson, J. Rexford. SIGCOMM'06, September 11–15, 2006, Pisa, Italy. http://conferences.sigcomm.org/sigcomm/2006/discussion/showpaper.php?paper_id=1
[IPsphere]	http://www.ipsphereforum.org
[MANTICORESvc]	E. Grasa , X. Hesselbach, S. Figuerola, V. Reijs, D. Wilson, J.-M. Uzé, L. Fischer, T. de Miguel, "'The MANTICORE project: Providing users with a Logical IP Network Service"', TERENA Networking Conference 2008, Bruges, Belgium http://tnc2008.terena.org/schedule/presentations/show.php?pres_id=98
[MonWithVR]	J. Navrátil, T. Košnar, J. Furman, T. Mrázek, V. Krmíček, "Monitoring Of Overlay Networks With Virtual Resources", Proceedings of TERENA Networking Conference 2009, TNC2009, June 2009, Malaga, Spain http://tnc2009.terena.org/core/getfile.php?file_id=409
[OneLab]	OneLab http://www.onelab.eu/
[Phosphorus]	www.ist-phosphorus.eu
[PlanetLab]	PlanetLab http://www.planet-lab.org/

2.3 MANTICORE

2.3.1 Introduction

The MANTICORE project is intended to provide a web services-based Resource Management System (RMS) that offers a logical IP network. The system needs to control (logical) router devices, initially based on a Juniper architecture. Besides providing a ready-made (logical) IP network, MANTICORE will enable users to integrate logical routers into their own configurations and profit from the logical resources. To achieve this objective, the current version of IaaS Framework (Infrastructure as a Service Framework, based on UCLP) will be enhanced to support the logical routing feature and the XML API, starting from the Juniper system. In the future this IP (Layer 3) web service will be integrated with existing or new web services for Layers 1, 2 and, if needed, layers above Layer 3.

MANTICORE is privately funded by its partners, who include some NRENs and some commercial organisations (including the router vendors Juniper and Cisco). It began in 2007 and completed its first phase with a prototype in 2008. The current phase of the project, MANTICORE II, began in late 2008 and is working to complete a downloadable version in early 2010.

MANTICORE can be seen, at its simplest, as a provisioning system for virtual networking infrastructure – but, by taking a coordinated approach across multiple layers, some ambitious results are possible. In particular, by combining Layer 2 and Layer 3 provisioning, and codifying good addressing and routing practice in the provisioning, it is possible to provide a solution to the routing integrity problems that academic networks now encounter as they make connections between departments in different institutions. For further information, see “Routing Integrity in a World of Bandwidth on Demand” [Routing].

The Layer 3 (routing integrity) problem arises with systems that only cover point-to-point connectivity, as the user is connecting separate, perhaps unrelated, networks that do not have coordinated addressing or routing. Where a user requires many-to-many (IP) connectivity, with some specialised topology, MANTICORE supports this by integrating Layer 3 and Layer 2 (and, perhaps in the future, Layer 1) into the concept of a logical network.

2.3.2 Architecture overview

MANTICORE is a tool for provisioning logical networks. These are made up of logical routers (e.g. software logical routers that can be defined in Juniper M- and T-series routers) and point-to-point links (e.g. Ethernet VLANs or IP tunnels). The resulting virtual network has addresses assigned, and a routing policy defined. The routing policy defines not just how traffic routes internally, but also how the virtual network interfaces with the outside world, including the end user’s own LAN (as well as any connections to the Internet or other networks).

MANTICORE builds on the Infrastructure as a Service (IaaS) framework, which provides a soft interface to infrastructure resources by means of web services. Various projects extend IaaS in different ways, some of which can be seen in Figure 2.5. MANTICORE complements these by providing an interface to IP networks, treating such networks as one more piece of infrastructure with its own particular characteristics, and taking

advantage of the rest of the IaaS framework to compose the parts (such as optical links) that make up an IP network.

The two main use cases that can be seen for a logical IP network service are:

- Provisioning (logical) IP network(s) for the service provider itself. Here, streamlining and standardising the way IP networks are made is the main driving force. As soon as standardisation is in place, it becomes easier for a network provider to provide logical IP networks to clients.
- Provisioning of IP-related resources towards users or other service providers. Users (be they humans or machines) might want to build their own IP network, and thus a service provider must be able to provide (logical) IP resources to individual users. This service might not be directly needed now, but it is foreseeable that it will be.

MANTICORE sees these as implementations of the same type of network. The internal architecture of MANTICORE is made up of web services (WS), each of which specialises in a particular area, such as:

- (Logical) Router web service: a logical or physical device that has logical or physical ports and the ability to route traffic according to certain rules.
- Routing services: allow traffic in a router or network to be routed between internal entities (like other Router WSs) or external entities (like users or external networks).
- Lower-layer web services, which can provide connectivity at sub-Layer 1, Layer 1 and Layer 2 between (user/router) ports. Examples include, optical patch panels (OPP), lightpaths, and Ethernet or L2 MPLS VLL web service.
- An IP network web service, which integrates/combines the above services.

The next section covers routing services in more detail.

2.3.2.1 *The routing services*

The aim is to provide a programmatic way to build a logical IP network with its own consistent routing policy, so that users can take advantage of its network for their own applications without adversely impacting the existing network on which they operate.

Every IP network is different, and providing a generic framework for the specification of any IP network is no small task. In order to provide a useful service, the logical networks that may be defined must be very flexible, with few or no restrictions on network topology, devices that may be connected, or routing protocols that may be supported.

However, when creating an entire logical network from whole cloth, there are some characteristics that can be relied upon, and then used to advantage.

MANTICORE treats a single logical network as a single entity for the purposes of routing, and draws a distinction between internal routing and external routing. In accordance with best practice, it uses internal routing only to exchange information on the routers that make up the logical network. Because the information to be exchanged is minimised, the resulting routing table is small, and because MANTICORE does not interact

with "outside" devices, it has fewer concerns with the security and integrity of its internal routing. This lends itself very well to the use of well known IGPs (Interior Gateway Protocols) such as OSPF and ISIS (and, since the protocol chosen only needs to run inside the logical network, MANTICORE is free of many of the restrictions on that choice of protocol, and can choose it based on device support or any other particular characteristics the user might have in mind).

Every other route, then, MANTICORE sees as external – not only other networks, but even those belonging to end hosts that must be distributed within the network and (in aggregate form) beyond. This is, perhaps, counterintuitive; it's reasonable to see end hosts connected to a router as "within the network". MANTICORE's premise is that if a route is not necessary in order to connect the routers together, then it can be left outside the IGP. Since a number of router manufacturers now implement "indirect next-hop" for Border Gateway Protocol (BGP), it is possible for BGP to converge just as soon as the underlying IGP converges. In any event, MANTICORE must assume that BGP will converge no quicker than the IGP that glues it together. Advantage can be taken of such architecture in two ways.

First, the configuration of the IGP is made very simple – in some cases, trivially simple. The user may still wish to specify parameters such as link metrics (either directly, or indirectly through, e.g., bandwidth and latency restrictions) but the interfaces on which the IGP must run are restricted to the loopbacks of each logical router, and the links which connect the logical routers to each other. This information can be deduced directly from the logical network layout specified by the user.

Second, if MANTICORE is disciplined in using BGP to distribute information on external (including end-user) routes, then it can take advantage of a very well understood interface for connecting networks together. This gives the maximum flexibility in connecting existing networks – even if they don't use BGP themselves. Once the routing is understood and specified, it can be implemented by means of a static route, even if BGP is used to distribute the information internally. The user may pick whatever method of routing is appropriate to their network.

Instead of requiring the end user (with extensive, specialist routing knowledge) to provision routers, links between routers, bring up various routing protocols, compose an appropriate routing policy, implement it, and document it with RPSL, MANTICORE requires only the layout of the network and their preferred routing policy. It then creates the logical network in accordance with that specification, instead of requiring the layout and policy to be specified after the fact. In this way, it can provide a full logical network service to the user without requiring them to be unnecessarily familiar with the minutiae of logical links, routers and internal routing – and it can provide a working network, interfacing with other external networks, that is not subject to the routing integrity problems associated with connecting existing disparate networks over a point to point link.

2.3.3 User community

MANTICORE is specified to support a range of users, from large IP Network Operations Centres (NOCs) with many customers, to end users who have specific needs that are best accomplished by a specialised IP network.

At the level of the physical infrastructure administrator (e.g., a NOC) MANTICORE provides administrative access to the physical infrastructure, so that the NOC can partition the infrastructure appropriately, and grant access to the infrastructure to another user.

There can be a second level of user – for example, the IT department at an educational institution – which provides access to certain parts of that infrastructure to its end users, along with certain restrictions on configuring that infrastructure.

At the end-user level, the user may configure the infrastructure with which they have been provided, within the limits of the permission allowed. This may or may not extend to the configuration of certain routing protocols, firewall facilities, or other features.

2.3.4 Mechanisms for providing virtualisation

2.3.4.1 *Implementation of virtualisation on Layer 3*

Section 2.3.2 “Architecture overview” on page 19 fully covers Layer 3 virtualisation issues.

2.3.4.2 *Implementation of virtualisation on Layer 2*

Interfaces with different layers are implemented in the IaaS framework by means of additional web services. Within MANTICORE itself, different web services (Virtual Resource Services (VRSs)) are made for each of the physical ports or logical interfaces that a user can access – for example, an Ethernet Resource WS for Ethernet interfaces.

MANTICORE uses these WSs, and the rest of the IaaS Framework, in order to create the Layer 2 infrastructure over which the logical Layer 3 network runs. The actual underlying network is more or less independent, providing it can be controlled by the IaaS framework.

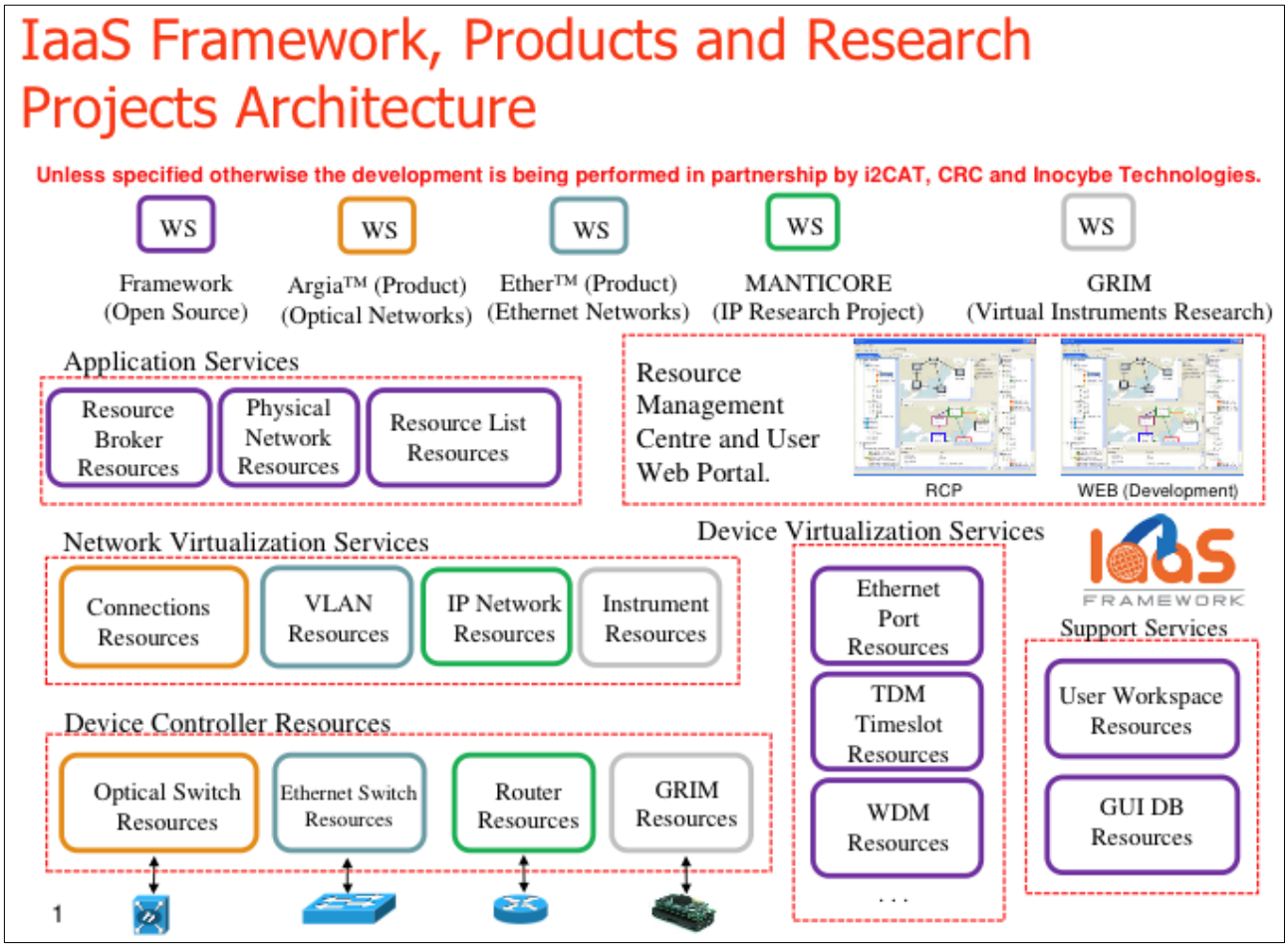

Figure 2.5: Architecture of the IaaS Framework & framework-based products/research projects

2.3.4.3 *Implementation of virtualisation on Layer 1*

At this time MANTICORE does not support Layer 1 networking. However, integrating lower-layer networking is a part of the overall vision, so provided that an IaaS Framework exists for the lower-layer network, there is no reason why such a Layer 1 network cannot be used within MANTICORE.

2.3.4.4 *Implementation of computing virtualisation*

MANTICORE does not define or include virtualised computing infrastructure; its scope is the connectivity between such resources, and providing a good method for describing and implementing the connectivity that they require.

2.3.4.5 *Management of virtualised infrastructure*

Similarly to Phosphorus, MANTICORE inherits the UCLP architecture of the management of services using web service-based technology. In the first instance, the administrator selects the resources to be assigned to

each logical router instance, and creates the logical router instances themselves. These are presented to a superuser in an institution, who can give an appropriate level of access (at their discretion) to end users.

The separation of the virtualised infrastructure is enforced using the router's own mechanisms. In principle, a router that does not offer software-based virtual routers can be used, offering exactly one instance. When used with, for example, Juniper M-series routers, MANTICORE takes advantage of the inbuilt features of JUNOS to create logical routers and separate the authentication and integrity of each.

2.3.4.6 *Control of virtualised infrastructure*

While the user may have a login on the logical routers, the MANTICORE model, at the moment, strongly encourages that changes are not made to the logical network except by the MANTICORE software. This is because MANTICORE manages the full IP configuration in the logical network, and any inconsistent changes may cause problems. Further, in the future, MANTICORE will provide a flexible interface to configuring routing characteristics, and implement these as a standard best practice configuration, which could be compromised by manual changes.

2.3.4.7 *Implementation of user interface*

MANTICORE provides a GUI to both the NOC and the end user, which allows them to provision their logical network. The procedure is as follows:

- The NOC performs initial setup, adding the physical routers to the system by means of the GUI.
- The NOC provisions the basic logical routers using the GUI.
- The NOC assigns particular resources (logical routers, interfaces and links) to the user within the GUI. At this point the user has control over their infrastructure.
- The user configures the links between their logical routers, as they require, within the GUI.
- The user provides addressing and routing information to the GUI.
- MANTICORE pushes out a configuration to the physical routers involved, setting up the network as requested by the user and authorised by the NOC.

In essence, the NOC and the user can draw a logical network on the screen, which is then implemented by MANTICORE.

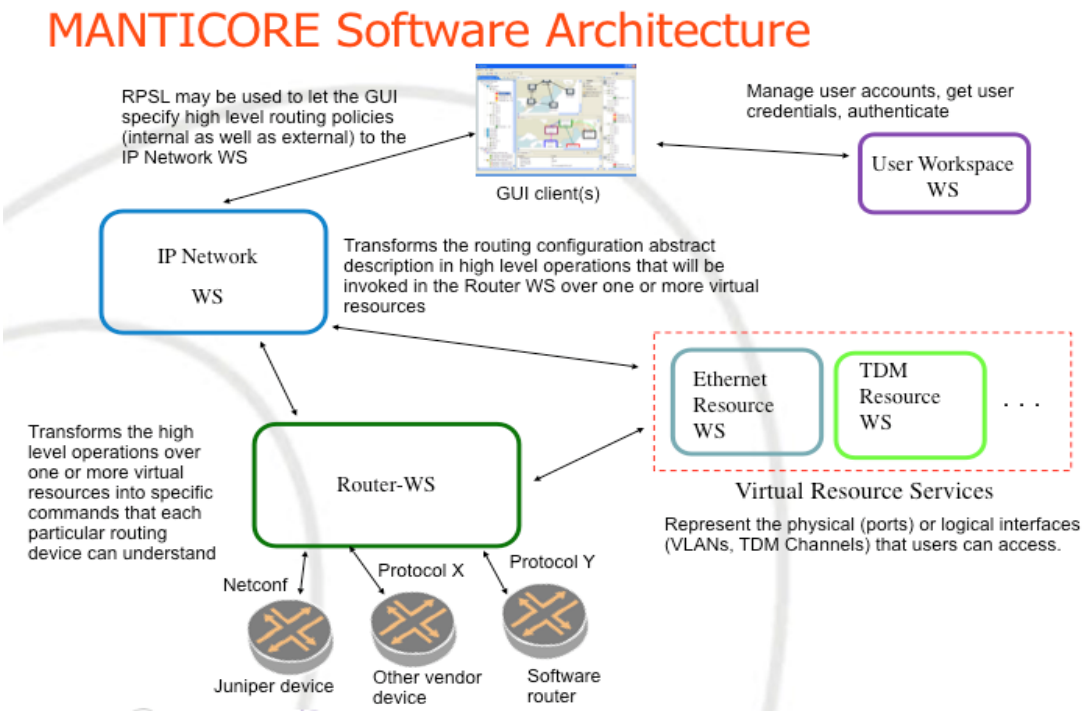

Figure 2.6: MANTICORE Software Architecture

2.3.5 Multi-domain support

MANTICORE, at this time, provides a single-domain network. In principle, it is possible to use resources from multiple physical domains, providing that access is granted, as MANTICORE makes no distinction.

The virtual infrastructure that is created operates within the context of a single installation of the MANTICORE system. However, at the IP layer, it is possible (indeed, expected) that each logical network will have its own routing policy and may well be in a different administrative domain (e.g. a separate BGP autonomous system).

2.3.6 Testbed implementation and availability

There is no public testbed available for MANTICORE at this time. However, the NRENs supporting the project are donating resources for the testing and demonstration of the project as part of its development.

2.3.7 Current status and roadmap

The first implementation of MANTICORE, a prototype, was completed in 2008; this led directly to the MANTICORE II project which is now underway. This is being developed privately by the project partners, with the expectation of running a test programme in late 2009 and early 2010, and releasing the software in the second quarter of 2010.

MANTICORE has an extensive programme of enhancements that will be implemented in a future MANTICORE III project. The project partners are investigating the best way to approach this, including the possibility of working with other projects, and the possibility of FP7 funding.

2.3.8 References

[Routing] D. Wilson, "Routing Integrity in a World of Bandwidth on Demand", TNC 2006 http://www.terena.org/events/tnc2006/programme/presentations/show.php?pres_id=242

2.4 Phosphorus (UCLP)

Much of the text in this section has been taken from the Phosphorus website www.ist-phosphorus.eu [Phosphorus] and the relevant communication magazine paper [UCLPv2]

2.4.1 Introduction

Phosphorus is a 3-year FP6 project, started in 2006. Although it ended in September 2009, it still provides relevant material for an overview of infrastructure virtualisation. It addresses some of the key technical challenges involved in enabling the on-demand end-to-end (E2E) high-bandwidth network services across multiple domains required for scientific and collaborative applications. The Phosphorus network concept and testbed will make applications aware of their complete resources (computational and networking) environment and capabilities, and enable them to make dynamic, adaptive and optimised use of heterogeneous network infrastructures connecting various high-end computing resources.

The Phosphorus project objectives include developing integration between application middleware and transport networks, based on three planes:

- Service plane:
  - Middleware extensions and APIs to expose network and Grid resources and make reservations of those resources.
  - Policy mechanisms (AAA) for networks participating in a global hybrid network infrastructure, allowing both network resource owners and applications to have a stake in the decision to allocate specific network resources.
- Network Resource Provisioning plane:
  - Adaptation of existing Network Resource Provisioning Systems (NRPS) to support the framework of the project.
  - Implementation of interfaces between different NRPSs to allow multi-domain interoperability with Phosphorus' resource reservation system.
- Control plane:
  - Enhancements of the GMPLS Control Plane to provide optical network resources as first-class Grid resource.

- Interworking of GMPLS-controlled network domains with NRPS-based domains.

The Phosphorus project deals with two types of networks with respect to control and management:

- Networks controlled by a GMPLS (Generalised Multi-Protocol Label Switching) control plane or its Grid-enabled version G^{2}MPLS (Grid GMPLS).
- Networks controlled by a Network Resource Provisioning System (NRPS). NRPSs are static or semi-static network management systems providing a mechanism for network resource (i.e. bandwidth and connectivity) provisioning (i.e. brokering, scheduling and reservation). Different types of NRPSs have been developed by industry and research institutes such as ARGON, DRAC, UCLPV2 and ARGIA.

One of the major outcomes of the Phosphorus project is the development and implementation of a novel network service plane to perform the adaptation between the Grid layer (application middleware) and the network layer controlled by NRPS or GMPSL/G^{2}MPLS. The Network Service Plane (NSP) provides the solution for a multi-domain environment with several coexisting NRPSs, each of them controlling its own domain as shown in Figure 2.7 (taken from Phosphorus project deliverable D1.4 "Definition and development of the Network Service Plane and northbound interfaces development" [PhosphorusD1.4]).

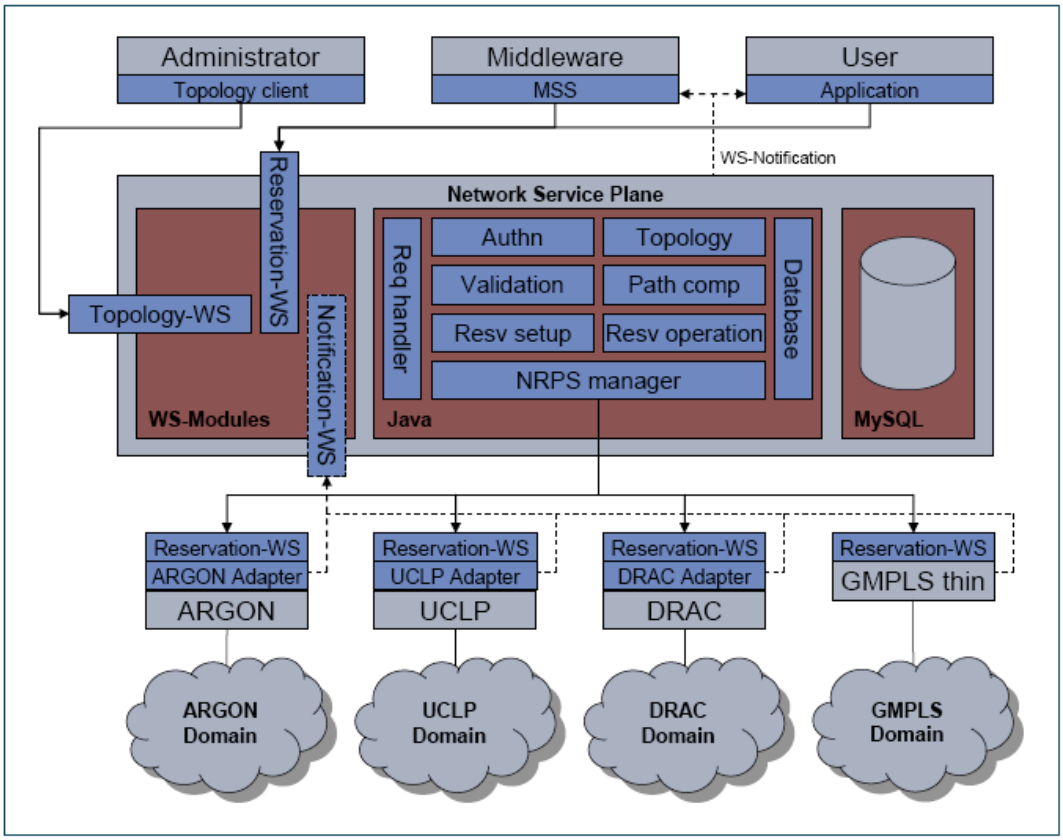

Figure 2.7: Overview of Phosphorus NSP

The Phosphorus project deploys an advanced heterogeneous testbed comprising domains with different NRPSs. However, the main focus for both NSP and NRPS developments in Phosphorus is on support for User-

Controlled Lightpath Provisioning (UCLP) because of its advanced capability and features, including virtualisation.

2.4.2 Architecture overview

Phosphorus provides the capability for virtualisation of the optical network infrastructure. It deploys the User Controlled Lightpath Provisioning (UCLP) system [UCLP] and its commercial variation Argia [Argia] as a tool for optical network virtualisation. This section focuses on the architecture and capabilities of the UCLP tool.

UCLP is a network configuration and provisioning tool able to partition switches and routers into virtual network resources that can be controlled by end-users. Users using UCLP can create and fully control complex virtual private optical networks across multiple administrative domains. These networks can be switched, routed or a combination of both; they may also comprise a variety of network devices and communication links, including lightpaths, switches, routers, instruments and sensors. These components are exposed and managed through web services. In UCLP, network elements and connections are represented as a web service defined by Web Service Description Language (WSDL) [WSDL]. UCLP web services provide flexible control and management plane functionalities, which can facilitate construction of complex network architecture and services.

UCLP is a distributed network control and management tool allowing users to provision lightpaths across multiple administrative domains. Authorised users can obtain network resources from a pool of available resources from different administrative domains and manage them.

UCLP (specifically UCLP version2) follows a service-oriented architecture (SOA) as shown in Figure 2.8 (taken from “UCLPv2, a Network Virtualization Framework built on Web Services” [UCMPv2]). The architecture comprises of three main layers: resource management layer, resource virtualisation layer and high-level services layer. Each layer provides a set of services based on web service (WS).

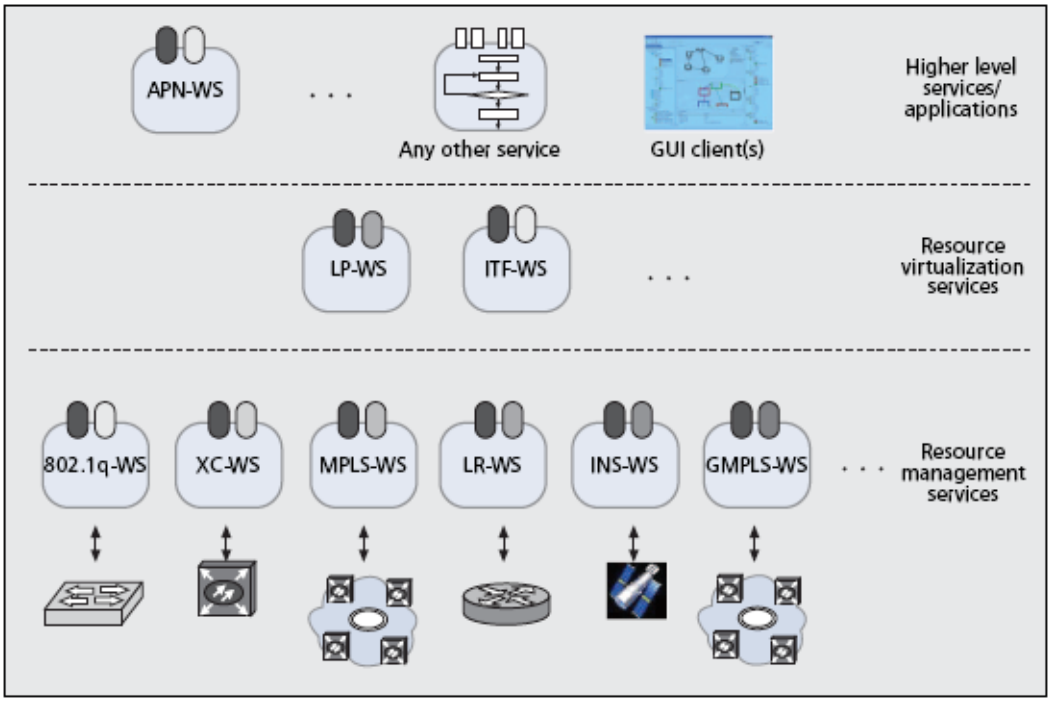

Figure 2.8: UCLP service-oriented architecture

Resource management layer

This layer comprises of a group of web services that manage and control physical layer resources. For each type of resource (e.g. router, network element) to be supported in UCLP, there must be a dedicated web service for its control and management. The resource management layer can comprise the following web services:

- Cross connect (XC) web service: manages devices that create Layer 1 cross connects such as fibre switches, wavelength switches and SDH/SONET cross-connects.
- VLAN (802.1q) web service: enables UCLP to control Layer 2 VLAN devices.
- Logical router web service: manages part or segment of a router when a router is partitioned to several independent logical devices.
- GMPLS/MPLS web service: enables UCLP to interface with a GMPLS/MPLS-enabled domain.
- Instrument web service: for controlling instruments such as sensors and network attached storage devices.

Of all the above, currently only the XC web service is supported by UCLPv2; the rest are included in the roadmap of UCLPv2's development. The SOA architecture and web service technology in UCLPv2 allow any arbitrary device control WS to be implemented in future.

Resource virtualisation layer

This layer provides a layer of abstraction between the higher layer services and the physical layer. It comprises two fundamental web services: the lightpath web service and the interface web service. These two web services provide a virtualised view of a network or a segment of a network where the user doesn't see details of

the physical layer but instead sees a set of lightpaths and their associated end points. A user can mix and match these paths and create the desired VPN topology.

- Lightpath web service: a lightpath is a reserved, private link between two interfaces of network elements e.g an SDH circuit. This web service provides a set of functionality for creating, managing, monitoring and deleting individual lightpaths or a set of lightpaths.
- Interface web service: this web service provides abstraction of interfaces (ports) of physical layer network elements.

High-level services layer

Using UCLP, heterogeneous networks can be represented as a pool of web services representing lightpaths and their associated end interfaces. These web services can be bonded, partitioned, sub-leased or connected together to build complex user-defined network topologies across multiple domains without knowing the technological details of the physical layer. As such, users can use these powerful features and create complex high-layer services based on the virtualisation service provided by UCLP. Examples of these services are bandwidth-on-demand services, routing services, etc.

2.4.3 User community

The ultimate goal of the Phosphorus project is to enable end-to-end dynamic service provisioning across the European and worldwide heterogeneous research network infrastructure and to disseminate the enabling technologies to the European NRENs and their users such as supercomputing centres. Furthermore, Phosphorus will provide applications with the ability to treat the underlying network as a first-class Grid resource.

In summary, the targeted user communities for Phosphorus, and specifically for UCLP, are NRENs serving e-science applications with a need for supercomputing processing power interconnected by a high-capacity optical network. Using UCLP/Argia, NRENs and/or e-science application users are able to build their own application-specific infrastructure on top of the existing physical infrastructure (i.e. network and computing infrastructure).

2.4.4 Mechanisms for providing virtualisation

2.4.4.1 *Implementation of virtualisation on Layer 3*

UCLP and its variations don't support Layer 3 virtualisation.

2.4.4.2 *Implementation of virtualisation on Layer 2*

In theory, and in terms of architecture, UCLP can support Layer 2 virtualisation and abstraction of services. It is perfectly possible to support Layer 2 devices through VLAN web services. However, currently there is no implementation of such services.

2.4.4.3 *Implementation of virtualisation on Layer 1*

UCLP supports a wide variety of Layer 1 services and devices such as fibre switching, wavelength switching and SDH/SONET circuit switching, including advanced capability such as support of interoperability with GMPLS-controlled layer devices or domains. Through this capability, UCLP is able to fully support virtualisation and abstraction of Layer 1 optical network. It provides basic services such as virtual connection or Layer 1 router and also complex services such as user-controlled and -defined virtual network creation and management. Users using UCLP can bond, partition, sub-lease or interconnect Layer 1 resources without dealing with physical and technological details, and create complex Layer 1 virtual resources, services or virtual network topologies.

2.4.4.4 *Implementation of computing virtualisation*

UCLP and its variations don't support computing virtualisation and abstraction of its services. However, it may be possible to achieve virtualisation of computing resources by implementing extended instrument web services and also more features within the virtualisation layer for partitioning and aggregating computing resources.

2.4.4.5 *Management of virtualised infrastructure*

With UCLP architecture, the management of services and virtualised resources is centralised and delegated to their creators (users). UCLP WS-based technology, SOA-based architecture, advanced Graphical User Interface (GUI) and, more importantly, high-layer services provide a set of web-based tools for users to manage their virtual network-based resources from several domains. Furthermore, within the Phosphorus project a complex network service plane (NSP) has been developed that facilitates interaction and communication between UCLP and different types of NRPSs, the GMPLS control plane and also Grid middleware.

2.4.4.6 *Control of virtualised infrastructure*

UCLP is a centralised management system, which enables users to create and control their own virtual infrastructure using resources from the physical infrastructure. The UCLP management system is static and self-contained, and doesn't allow any other management system or control plane to control the virtual infrastructure. Virtual infrastructure owners are only able to control their own infrastructure using management features provided by UCLP. The basic management service, which is provided by UCLP, is the resource management services or network element web services, which constitute the basic building block of other complex management services.

2.4.4.7 *Implementation of user interface*

UCLP currently supports an advanced GUI that enables users to use the functionality offered by UCLP. However, the UCLP comprises a set of web services and any user, client, software or tool can communicate with it and its web services using SOAP protocol and web-service technologies.

2.4.5 Multi-domain support

UCLPv2 support for multi-domain operation is two-fold. First, it is possible to create a virtual infrastructure using physical resources from multiple domains. Second, using UCLP, it is possible to create multiple isolated virtual domains on top of the same physical infrastructure. However, one must consider that the first case (creating a virtual infrastructure using physical resources from multiple domains) is only possible if all participating domains are managed by UCLP.

Furthermore, within the framework of Phosphorus, a system called Harmony has been developed which allows individual network domains controlled by different control plane or management systems such as UCLP and GMPLS to interact with each other and provide transient connectivity services for each other. Harmony is a network resource brokering system that provides co-allocation of heterogeneous (optical) network resources in multi-domain and multi-technology environments with advance reservation functionalities. This is shown in Figure 2.9 (taken from "UCLPv2, a Network Virtualization Framework built on Web Services" [UCMPv2]).

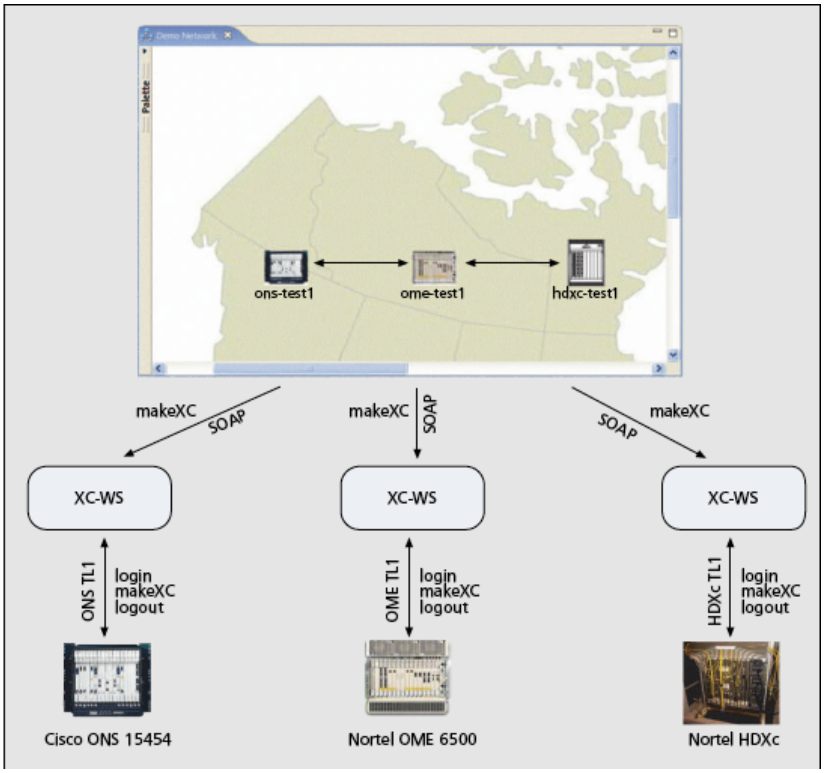

Figure 2.9: UCLP GUI-based management system

2.4.6 Testbed implementation and availability

A UCLP testbed has been implemented within the Phosphorus project framework. It is currently operational and available to Phosphorus members for testing and evaluation. The UCLP testbed comprises 9 domains in Europe interconnected by GÉANT. The Phosphorus project's full testbed will not be available after the final demonstration of the project (end of September 2009). However, a 3-domain testbed, comprising the University

of Essex, PSNC and I2CAT interconnected by GÉANT, is planned to host the UCLP demo capability after the end of project.

2.4.7 Current status and roadmap

Different variations of UCLP have been implemented by a few research institutes as well as university research groups. The latest version is UCLPv2, which incorporates most of the features and functionality mentioned above, and mainly supports Time-Division multiplexing (TDM) switching technologies (SDH/SONET). Also, there are variants/versions of UCLP tailored for specific purposes, for example, for Grid applications.

UCLP is an open source development. However, since 2006 (UCLPv2), there haven't been any official releases and it has become part of a commercial product called Argia [Argia]. Argia is the evolution and extension of UCLPv2. Its current implementation is based on the Globus Toolkit 4 [Globus] and supports the following network technologies: photonic switches, ROADM, TDM and untagged Ethernet. The services that it provides include:

- Virtualisation emulation and permission management for optical NEs: ability for the optical infrastructure owner to partition their physical devices into different resources and assign them to different users, who could control their part of the optical NE independently from others.
- Dynamic point-to-point immediate connection setup: users can invoke this service to create immediate end-to-end connections over the resources they have been assigned.

2.4.8 References

[Argia] E. Grasa, S. Figuerola, A. Forns, G. Junyent, J. Mambretti, "Extending the Argia Software with a Dynamic Optical Multicast Service to support High Performance Digital Media", accepted for publication in Elsevier journal of Optical Switching and Networking Volume 6, Issue 2, Recent trends on optical network design and modeling – selected topics from ONDM 2008, April 2009 http://www.sciencedirect.com/science?_ob=ArticleURL&_udi=B7GX5-4VXMPRK-1&_user=10&_rdoc=1&_fmt=&_orig=search&_sort=d&_docanchor=&view=c&_searchStrId=1077462195&_rerunOrigin=google&_acct=C000050221&_version=1&_urlVersion=0&_userid=10&md5=d6a9240fe4221f8d93569fc5868870be

[Globus] http://www.globus.org/

[Phosphorus] www.ist-phosphorus.eu

[PhosphorusD1.4] Phosphorus project deliverable D1.4 "Definition and development of the Network Service Plane and northbound interfaces development" http://www.ist-phosphorus.eu/files/deliverables/Phosphorus-deliverable-D1.4.pdf

[PhosphorusPresn] "Phosphorus: Lambda User Controlled Infrastructure for European Research" http://www.ist-phosphorus.eu/files/press/Phosphorus-general_presentation.pdf

[UCLP] http://www.canarie.ca/canet4/uclp/uclp_software.html

[UCLPv2] E. Grasa, S. Figuerola, A. López, G. Junyent, M. Savoie, "UCLPv2, a Network Virtualization Framework built on Web Services", IEEE Communications Magazine, Feature Topic on Web Services in Telecommunications part II, pp. 126-134

[WSDL] www.w3.org/TR/wsdl

2.5 4WARD

Most of the information in this section is extracted from public deliverables and documents, particularly the 4WARD Project website [4WARD] [4WARD-VNet]. Note that at the time of writing, limited information was publicly available for 4WARD; hence some sub-sections have not been completed.

2.5.1 Introduction

4WARD [4WARD] is a 7th European Community Framework Program project started on 1st January 2008. It aims to increase the competitiveness of the European networking industry and to improve the quality of life for European citizens by creating a family of dependable and interoperable networks providing direct and ubiquitous access to information. These future wireless and wireline networks will be designed to be readily adaptable to current and future needs, at acceptable cost. 4WARD's goal is to make the development of networks and networked applications faster and easier, leading to both more advanced and more affordable communication services.

One of the basic tenets of 4WARD is that the Future Internet shall allow multiple networking solutions to coexist, not only in the link and the application layer, as in the Internet today, but also in the network and transport layers. Network Virtualisation is ideally suited to allow the coexistence of different network architectures, legacy systems included. Virtualisation is thus not only an enabler for the coexistence of multiple, possibly revolutionary, architectures, but also provides a smooth path for the migration towards more evolutionary approaches. This way, virtualisation can help to keep the Internet capable of evolving and innovation-friendly, particularly since it can mitigate the need to create broad consensus regarding the deployment of new technologies among the multitude of stakeholders that make up today's Internet. By decoupling the infrastructure from the services, virtualisation can provide the opportunity to roll out new architectures, protocols, and services without going through the slow and difficult process of creating such consensus.

Virtualisation further provides a general approach for network service providers to share a common physical infrastructure. This is particularly beneficial in network domains where the deployment costs per user are predominant and an encumbrance for frequent technology replacement as is the case for instance in access networks.

The goal of Network Virtualisation (VNet) is to develop a systematic and general approach to network virtualisation. The problem space is divided into three main areas:

1. **Virtualisation of Network Resources:** While the virtualisation of some types of resources, such as servers and links, is well known and already widely used today, VNet aims for a generalised approach that allows the use of a broad variety of resources as part of a unified virtualisation framework. Virtualisation of both wireless and wireline resources will be studied. The performance of shared resources and the secure separation of virtual networks sharing a resource will be important issues. The secure, flexible, and efficient exploitation of wireless spectrum and access infrastructure is

expected to significantly improve cost-efficiency and utilisation of expensive wireless infrastructures. To allow for the integration of a variety of resources into the common framework, VNet aims to develop standardised interfaces for management and control of the virtualised resources.

2. **Provisioning of Virtual Networks:** Based on a substrate of virtualised network resources, and using their control and management interfaces, VNet aims to develop a systematic approach to instantiating complete virtual networks using the virtual resources, allowing the on-demand deployment of new virtual networks on a potentially large scale. The virtualisation framework includes the discovery of available physical and virtual resources, as well as the scalable provisioning, control, and aggregation of resources to form complete networks.
3. **Virtualisation Management:** Once a virtual network has been instantiated, management mechanisms are required to deal with the virtual resources it is based upon. These mechanisms should support the deployment, control, and dynamic re-allocation of resources on demand during the lifetime of the virtual network. A particular challenge is the dynamic management of volatile and mobile resources that may enter or leave the virtual network at any time.

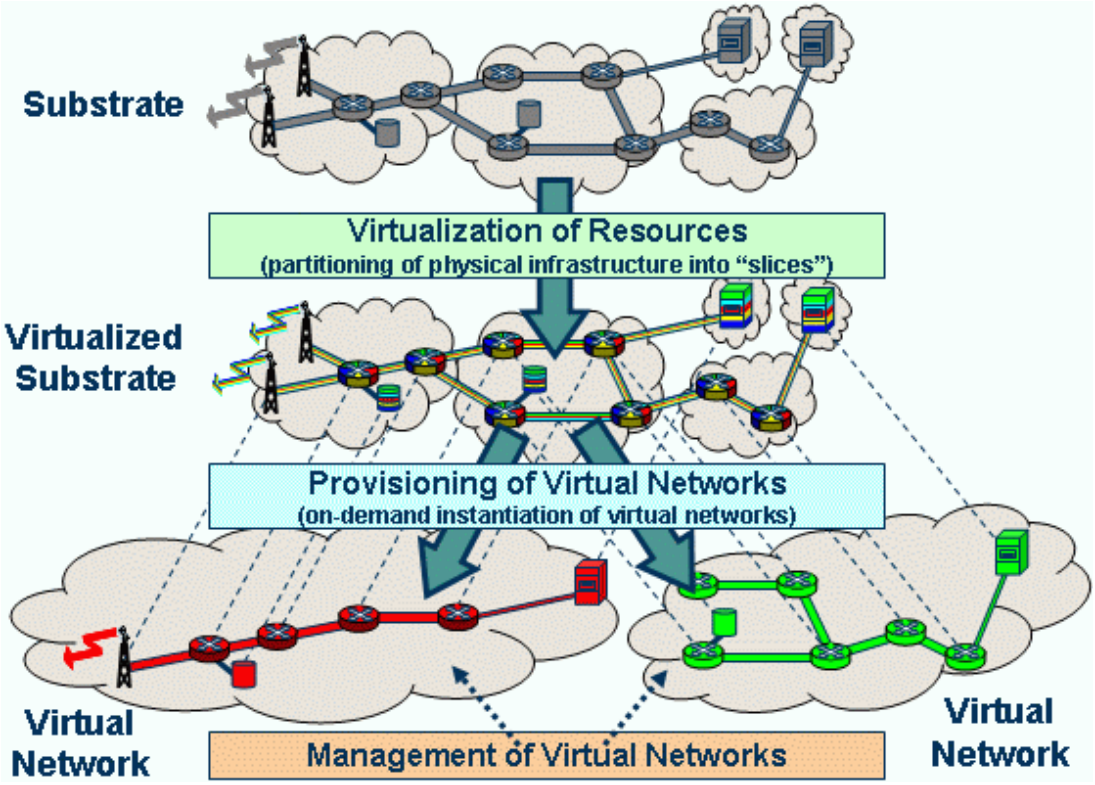

Figure 2.10: VNet virtualisation framework

2.5.2 Architecture overview

4WARD defines the following potential future business players who provide the end-user service:

- The Physical Infrastructure Provider (PIP) owns and manages the physical infrastructure (the substrate), and provides wholesale of raw bit and processing services (also known as slices), which support network virtualisation.
- The Virtual Network Provider (VNP) is responsible for assembling virtual resources from one or multiple PIPs into a virtual topology.
- The Virtual Network Operator (VNO) is responsible for the installation and operation of a VNet over the virtual topology provided by the VNP according to the needs of the Service Provider (SP), and thus realises a tailored connectivity service.
- The Service Providers (SPs) use the virtual network to offer their services. These can be value-added services, and then the SPs act as application service providers, or transport services, and then the SPs act as network service providers.

These roles must be understood as a technical concept, and therefore a single business entity could perform more than one task, e.g., one company can be a PIP and VNP at the same time; or the VNP and VNO roles could coincide.

On the technological side, a VNet Control Plane Architecture is defined, which provides the control and management functions. As there are different players involved, the control plane must perform a balancing act between the following conflicts:

- Information disclosure against information hiding.
- Centralisation of configuration and control against delegation and distribution of configuration and control.

VNet instantiation involves several interactions between players. A yet-to-be-standardised resource description model communicates the requirements between the different players.

End-user attachment can be achieved in two ways:

- The VNO/VNP issues a request to extend the VNet all the way to the end user, or
- The end user requests that a tunnel over existing substrate is constructed towards a ”VNet access point”.

While full experimental evaluation of results will mostly take place in the second year of the project, a number of small-scale prototyping activities have been implemented in order to support conceptual development and early validation. Several project partners are cooperating on such "bottom-up" activities using prototyping.

In order to enable remote control and management of virtualised resources, basic functions and primitives of a Virtualisation Management Interface (VMI) have been implemented. The primitives include the creation and destruction of virtual resources within a physical resource, the connection of nodes and links to interfaces, and functions for its configuration. In a testbed, the basic functions of the infrastructure and the VNet provider were implemented. Furthermore, the prototype offers on-demand operational and management capabilities, which are typically performed by the VNet operator. In line with some scenarios of the VNet architecture, the combined tasks of VNet provisioning and management are assigned to a single entity. The prototype mainly

provides a topology of virtual nodes based on specific requests of the VNet provider, followed by the instantiation (and optionally the on-demand management) of this network.

A further prototyping activity is focusing on software-based virtual routers. A crucial prerequisite for any network virtualisation architecture is virtual routers with a higher level of flexibility and programmability than today's commercial routers. Virtualised forwarding planes were evaluated in terms of performance, isolation, and fairness. The experiments were conducted on an SMP multicore platform running Xen and Click Modular Router for forwarding. It was shown that a virtual router platform based on commodity hardware can forward packets at very respectable rates up to 7.1 Mpps or 3.6 Gbps (on aggregate) for minimum-sized packets. Therefore, 4WARD designed and implemented such a platform that leverages modern and emerging hardware trends to provide:

- Highly configurable forwarding planes for advanced programmability.
- Optimised mapping of virtual router components to cache hierarchies.
- Hardware multi-queuing for sharing interfaces between virtual routers.

The platform enables the consolidation of virtual data planes onto a single forwarding domain, which results in significantly higher performance than forwarding in the separate guest domains, by avoiding the costly hypervisor domain switches per packets.

With memory access latency as the performance bottleneck for packet forwarding, packets as well as much of the data structures needed to process them should stay in cache memory as packets travel from an input to an output interface. Therefore, the notion of a forwarding tree has been introduced, which represents the set of packet processing elements necessary to move a packet from a single input to all possible outputs. The advantage of using a forwarding tree is that its elements can all be allocated to the same cache hierarchy, thus confining packets to this hierarchy and reducing main memory accesses.

In addition, forwarding trees are a smaller allocation unit than routers, providing more flexibility when exploiting hardware resources and implementing fairness. Since forwarding trees belonging to the same virtual router must send packets to the same output interfaces, a locking mechanism was used to grant exclusive access to these interfaces. Another critical issue for a virtual router platform is how to share interfaces between virtual routers. In order to provide fairness, hardware packet classification on the network interface cards has been enabled, allowing packets to be filtered into different queues, which can be subsequently polled by different virtual routers.

The tests demonstrated that it is possible to implement virtual routers that support high packet-forwarding rates combined with the flexibility and programmability afforded by general-purpose processors. As a preparation for integrated tests, the testbeds of several project partners have been interconnected. 4WARD is implementing a framework to allow sharing of resources for experiments across partner sites.

2.5.2.1 *Summary of results*

4WARD has:

- Defined a business-role architecture. This architecture contains, besides the users, the Physical Infrastructure Provider (PIP), the Virtual Network Provider (VNP), the Virtual Network Operator (VNO) and the Service Provider (SP).
- Defined a management and control plane architecture. The control plane operates out-of-band via an interface called Out-of-VNet-Access.
- Started to implement a prototype of the Virtualisation Management Interface (VMI).
- Started to implement a prototype of a software-based virtual router for performance and functional issues.

4WARD activities are not focused on a certain layer like L2 or L3.

As mentioned in Section 1 “Introduction” on page 4, it is expected that a detailed overview of 4WARD will be provided in the second deliverable, DJ1.4.2, due March 2012.

2.5.3 References

[4WARD]	4WARD Project http://www.4ward-project.eu/
[4WARDControlPlane]	Roland Bless, Christoph Werle,: “Control Plane Issues in the 4WARD Network Virtualization Architecture”, Proceedings of the KiVS Workshop on Overlay and Network Virtualization, Kassel, Germany, March 2009 http://eceasst.cs.tu-berlin.de/index.php/eceasst/article/view/225/216
[4WARD-VNet]	http://www.4ward-project.eu/index.php?s=overview&c=WP3
[DesigningaPlatform]	N. Egi, A. Greenhalgh, M. Handley, M. Hoerdt, F. Huici, L. Mathy, and P. Papadimitriou, "Designing a Platform for Flexible and Performant Virtual Routers on Commodity Hardware External link mark", Workshop on Overlay and Network Virtualization, Invited Paper, Kassel, Germany, Kassel, Germany, March 2009.
[ImplementingNV]	P. Papadimitriou, O. Maennel, A. Greenhalgh, A. Feldmann, and L. Mathy,. "Implementing Network Virtualization for a Future Internet External link mark", 20th ITC Specialist Seminar on Network Virtualization, Hoi An, Vietnam, May 2009
[NetworkVirtualization]	R. Bless, C. Werle,. "Network Virtualization from a Signaling Perspective External link mark",. Future-Net '09 International Workshop on the Network of the Future 2009 in conjunction with IEEE ICC 2009, Dresden, June 16th-18th, 2009.
[TowardsInterop]	Y. Zaki, L. Zhao, J. Jimenez, K. Mengal, A. Timm-Giel, C. Goerg,. "Towards Interoperability among Virtual Networks in the Future Internet",. ICT-MobileSummit, Santander, Spain, June 2009
[VirtualRouters]	N. Egi, A. Greenhalgh, M. Handley, M. Hoerdt, F. Huici, and L. Mathy, "Towards High Performant Virtual Routers on Commodity Hardware External link mark", ACM CoNEXT, Madrid, Spain, December 2008

2.6 GENI

Most of this summary, including the graphics, is extracted from “GENI: Global Environment for Network Innovations – Facility Design” [GENI-GDD-07-44], dated March 2007. Therefore, there is a risk of outdated

information. However, as mentioned in Section 1 “Introduction” on page 4, it is expected that a detailed overview of GENI will be provided in the second deliverable, DJ1.4.2, due March 2012. GN3 JRA1 Task 4 is currently in the process of establishing a channel of communication with the GENI team dedicated to virtualisation topics. The Memorandum of Understanding between GN3 and GENI, once signed, should make more tangible information available.

2.6.1 Introduction

GENI (Global Environment for Network Innovation) is a US programme funded by the NSF (National Science Foundation). It is an experimental facility designed to form a robust environment to allow computer networks’ researchers to experiment on a wide variety of problems in communications, networking, distributed systems, cyber-security, and networked services and applications with emphasis on new radical ideas. GENI will provide an environment for evaluating new architectures and protocols, over fibre-optic networks equipped with state-of-the-art optical switches, novel high-speed routers, radio networks, computational clusters and sensor grids.

GENI infrastructure presents some key characteristics in order to enable advanced research:

- Programmability: researchers can fully control GENI nodes’ behaviour.
- Virtualisation: researchers can simultaneously share the GENI infrastructure using their own isolated slice of resources.
- Federation: different parts of the GENI infrastructure are owned and/or operated by different organisations.
- Slice-based experimentation: each experiment will be implemented on a specific slice of the GENI resources.

This experimental facility should pave the way to:

- Long-running, realistic experiments with enough instrumentation to provide real insights and data.
- Propose an infrastructure that promotes and makes adhesion easy for real users into these long-running experiments.
- Enable large-scale growth for successful experiments, so good ideas can be validated on a large scale.

Ultimately, GENI’s goal is to avoid technology “lock in,” enable addition of new technologies as they mature, and potentially grow quickly by incorporating existing infrastructure into the overall “GENI ecosystem”.

A great number of projects are currently ongoing that are targeted at designing and operating prototypes of the GENI infrastructure. These projects are managed by the GENI Project Office (GPO).

2.6.2 Architecture overview

The high level GENI architecture can be divided into three levels, as shown in Figure 2.11:

- Physical substrate: represents the set of physical resources that constitute the GENI infrastructure, such as routers, links, switches.
- User services: represent the set of services that are available for the users in order to fulfill their research goals.
- GENI Management Core (GMC): defines a framework in order to bind user services with underlying physical substrate. In order to implement this, it includes a set of abstractions, interfaces and name spaces and provides an underlying messaging and remote operation invocation framework.

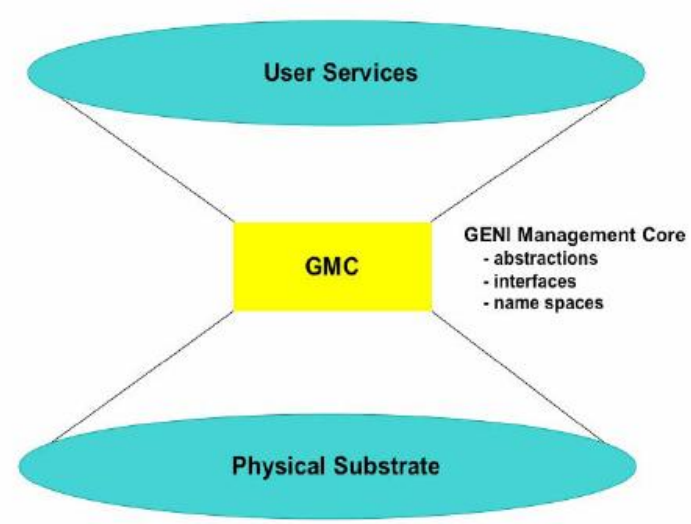

Figure 2.11: GENI architecture

The following sections provide more detail on the three levels of the GENI architecture.

2.6.2.1 *Physical substrate*

The physical substrate consists of an expandable collection of components. GENI components fall into one of the following categories:

- Programmable Edge Clusters (PEC): provide computational and storage resources as well as initial implementations of new network elements.
- Programmable Core Nodes (PCN): provide high-speed core-network data-processing functions.
- Programmable Edge Nodes (PEN): provide data-forwarding functionality at the boundary between access networks and a high-speed backbone.
- Programmable Wireless Nodes (PWN): implement proxies and other forwarding functionality within a wireless network.
- Client Devices: provide access to experimental services for end users.
- National Fibre Facility: provides 10 Gbps to 40 Gbps lightpath interconnection between PCNs.
- Tail circuits: interconnect GENI edge sites to the GENI core.
- Internet Exchange Points: interconnect the nationwide infrastructure to the commodity Internet.

- Urban 802.11-based Mesh Wireless Subnets: provide support for ad-hoc and mesh-network research.
- Wide-Area Suburban 3G/WiMax-based Wireless Subnets: provide open-access 3G/WiMax radios for wide-area coverage, along with short-range 802.11 class radios for hotspot and hybrid service models.
- Cognitive Radio Subnets: support experimental development and validation of emerging spectrum allocation, access, and negotiation models.
- Application-Specific Sensor Subnets: support research on both underlying protocols and specific applications of sensor networks.
- Emulation Grids: allow researchers to introduce and utilise controlled traffic and network conditions within an experimental framework.

2.6.2.2 *GMC*

The GENI Management Core (GMC) is a framework that defines a set of abstractions, interfaces and name spaces that binds together the GENI infrastructure. Because GENI's physical substrate and user services will develop and evolve rapidly as the facility is constructed and used, the GMC is designed to provide a narrowly defined set of mechanisms that both support and foster this development while isolating developmental change in one part of the system from that in other parts, so that independent progress may be made.

Abstractions

The GMC defines three key abstractions: components, slices, and aggregates. This sub-section introduces the abstractions; the following section describes the interfaces they support.

- **Components:** A component encapsulates a collection of substrate resources that can be either included on a single device or includes resources from many devices. Any resource can belong to only one component. Each component is controlled via a component manager (CM), over a well-defined interface. At the GENI facility it is possible to slice component resources among multiple users. This can be done either by virtualising component resources or by strictly partitioning them among the users. In both cases, the user is granted a sliver of the component. Each component is assigned a unique identifier as well as a human-friendly name.

- **Slices:** A slice is a set of slivers across a set of GENI components and an associated set of researchers that are implementing an experiment over these slivers. Each slice is assigned a unique identifier as well as a human-friendly name. Within the GENI framework, an experiment is a researcher-defined use of a slice.

- **Aggregates:** An aggregate is an unordered collection of components. Aggregates support hierarchy; an aggregate can contain other aggregates as well as components. Each aggregate has a unique identifier as well as a human-friendly name. Moreover, aggregates are controlled by aggregate managers.

Interfaces

GMC defines unique identifiers, called GENI Global Identifiers (GGID) for all the objects that constitute the GENI infrastructure, that is, components, slices and aggregates. A GGID is represented as an X.509 certificate.

Moreover, GMC defines two basic data types:

- **Resource specification (RSpec):** the data structure that describes GENI resources. It contains information about the resources that are encapsulated by components, their processing capabilities, their network interfaces and the instrumentation available on them.
- **Tickets:** granted by a component owner to a researcher, and later "redeemed" to acquire resources on the component.

GMC defines a series of operations for components, slices and aggregates. Some of them are mentioned below:

- Creating/modifying/deleting slices.
- Request for allocating a slice.
- Start/stop/delete a slice.
- Add/delete components in an aggregate.

2.6.2.3 *User services*

As shown in Figure 2.11, user services are built on top of the GMC and are the set of distributed services that enable GENI users to implement experiments on a given slice as well as to manage their allocated slices.

From the user services point of view, diverse user communities are defined in GENI. These communities are:

- Owners of parts of the substrate: define usage policies of the substrate and provide mechanisms for enforcing these policies.
- Administrators of parts of GENI: manage the GENI substrate ensuring proper operation.
- Developers of user services: create GENI services using the GMC interfaces.
- Researchers: use the GENI facility in order to conduct research. They can allocate resources on the GENI substrate and deploy specific software.
- End users not affiliated with GENI, but who access services provided by research projects that run over GENI.
- Third parties that may be impacted from GENI operation.

2.6.3 User community

As stated in Section 2.6.1 "Introduction" on page 39, the GENI infrastructure will provide the opportunity for researchers to experiment on a wide variety of innovative ideas on computer networks.

2.6.4 Mechanisms for providing virtualisation

2.6.4.1 *Implementation of virtualisation on Layer 3*

Each Programmable Core Node (PCN) includes a Packet Processing System (PPS) which is a collection of line cards, general-purpose processors, and programmable hardware (e.g., network processors and FPGAs) connected via a switch fabric. PPS implements a high-speed programmable device that supports multiple virtual routers, possibly belonging to different slices, within a shared platform. The term virtual router is used to denote any network element with multiple interfaces that forwards information through a network, while possibly processing it as it passes through. As such it also encapsulates the functionality of a conventional Ethernet switch. PPS design has two main goals:

- To provide the necessary resources to the researchers in order to build their own virtual routers that can operate at high speed.
- To ensure that virtual routers operating in different slices will run without interference.

The design of the PPS is quite different from the design of conventional routers and switches in that it must allow bandwidth to be flexibly allocated among multiple virtual routers and provide third-party access to generic processing resources that can be flexibly allocated to different virtual routers. Hence, PPS design requirements include open hardware and software components, scalable performance, stability and reliability, ease of use, technology diversity and adaptability, flexible allocation of link bandwidth and strong isolation between virtual routers.

2.6.4.2 *Implementation of virtualisation on Layer 2*

Each Programmable Core Node (PCN) also includes a Circuit Processing System (CPS), which is a layered collection of circuit-oriented elements, as shown in Figure 2.12.

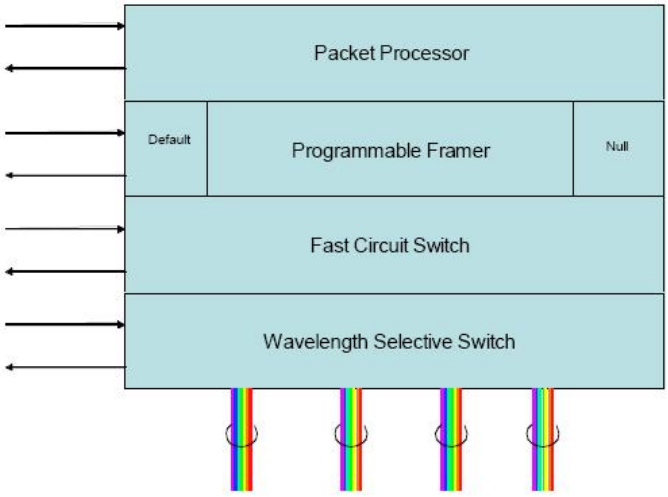

Figure 2.12: CPS design

Researchers can access CPS at whatever layer provides the appropriate of abstraction for their work. The layers of the CPS are described below:

- Wavelength Selective Switch (WSS): Data on one 10 Gbps wavelength can be switched to another wavelength, or delivered to the Fast Circuit Switch. Data carried on a wavelength is totally transparent for the WSS. User research equipment can connect directly to the WSS (shown on the left hand side of Figure 2.12).
- Fast Circuit Switch (FCS): Circuits are multiplexed using TDM onto a 10 Gbps wavelength. Virtual circuits of any bandwidth with granularity 1 Mbps can be established. Individual circuits can be assembled from multiple basic slots within and across wavelengths. The FCS will connect to the WSS via optical fibre. User research equipment can connect directly to the FCS (shown on the left hand side of Figure 2.12).
- Programmable Framer (PF): The framer will frame packets inside circuits. SONET is used for default framing format. The framer should have a null framing format in cases where the packets themselves carry sufficient information for recovery at the destination. The PF will connect to the FCS via electrical links or short-reach optics. User research equipment can connect directly to the PF (shown on the left hand side of Figure 2.12).
- Packet Processor (PP): this corresponds to the PPS subsystem described above. PP can bypass the PF layer and be connected directly to the FCS layer using optical fibres.

The CPS design is planned to be implemented using commercially available hardware.

2.6.4.3 *Implementation of virtualisation on Layer 1*

Layer 1 virtualisation issues are covered in Section 2.6.4.2 “Implementation of virtualisation on Layer 2” on page 43.

2.6.4.4 *Implementation of computing virtualisation*

Computing and storage services are provided by Programmable Edge Clusters (PECs). GENI plans to deploy PEC components at 200 different sites on the GENI infrastructure.

PECs will consist of a rack equipped with commodity processors, high storage capacity, and connection to the local network infrastructure. Each PEC will run virtualisation software that will implement slivers as virtual machines (VM), each of which can be bound to some amount of processor, memory, disk, and link capacity under the control of the Component Manager (CM). Two different virtualisation technologies are expected to be deployed:

- Paravirtualisation, which gives slivers access to low-level hardware resources.
- Container-based virtualisation, which gives slivers access to a virtualised system call interface.

While PECs emulate computational clusters, they may also act as clients, individual servers, server farms, ingress routers for testing of innovative network architectures, etc. Storage capacity and computational

capability of PECs will differ significantly from site to site. At the low end, a PEC will include 8-12 processors and at the high end, a PEC might include 512-1024 processors.

2.6.4.5 *Management of virtualised infrastructure*

The GENI infrastructure will be managed by the GENI operator by means of an operator portal. According to the ITU FCAPS model, the following management functionality is planned to be offered to the GENI operators:

- Fault management. GENI operators will be able to detect and repair problems on the GENI infrastructure.
- Configuration management. GENI operators will be able to provision, configure and validate new components of the GENI infrastructure.
- Accounting management. GENI operators will ensure that only authorised users and experiments can use the GENI infrastructure as well as deploy policies on the usage of the infrastructure.
- Performance management. GENI operators will be able to monitor the utilisation and performance of the GENI components.
- Security management. GENI operators will receive security-related information on the usage of the GENI infrastructure and will be able to find out whether GENI is being attacked or the GENI Acceptable Use Policy is being violated.

2.6.4.6 *Implementation of user interface*

Researchers will interact with the GENI infrastructure via the researcher portal which allows researchers and developers to specify the characteristics of their experiments and manage the experiments themselves. More specifically the researcher portal is going to be the front-end of a set of services offering the following functionality:

- Resource allocation: defines how the resources are shared among experiments (acquired, scheduled, or released).
- Slice embedding: instantiates a researcher's slice over a number of components.
- Experimenter workbench: provides a set of tools to create, configure and control researchers' experiments.

2.6.5 Testbed implementation and availability

According to "The Global Environment for Network Innovations (GENI)" [GENI-Overview] dated April 2009, GENI is still in prototyping and design stage.

2.6.6 References

[GENI-GDD-07-44] L. Peterson (ed.), "GENI: Global Environment for Network Innovations – Facility Design", GDD-07-44, March 2007

[GENI-Overview] "The Global Environment for Network Innovations (GENI)", April 2009 http://www.geni.net/wp-content/uploads/2009/04/geni-at-a-glance-final.pdf

2.7 PlanetLab/VINI/OneLab

Most of the information in this section is reproduced from publicly available deliverables and documents, particularly "PlanetLab Architecture: An Overview" [PlanetLabArch] and "In VINI Veritas: Realistic and Controlled Network Experimentation" [InVINIVeritas].

2.7.1 Introduction

PlanetLab is an infrastructure that supports multiple distributed services running on hundreds of machines belonging to different organisations throughout the world. Its primary goal is to provide a global platform that supports broad-coverage services that benefit from having multiple points-of-presence on the network. PlanetLab is designed around five major principles:

- **Distributed Virtualisation:** PlanetLab simultaneously supports two usage models: short-term services and continuously running services. PlanetLab supports these usage models by providing distributed virtualisation, where each service can run in a slice of PlanetLab's global resources. Multiple slices can run concurrently on PlanetLab.
- **Unbundled Management:** PlanetLab decomposes the management function into a collection of largely independent infrastructure services, each of which runs in its own slice and is developed by a third party, just like any other service.
- **Chain of Responsibility:** PlanetLab takes advantage of nodes contributed by research organisations around the world. These nodes, in turn, host services on behalf of users from other research organisations. The PlanetLab Consortium (PLC) plays the role of a trusted intermediary, thereby freeing each owner from having to negotiate a hosting agreement with each user. An important responsibility of PlanetLab is to preserve the chain of responsibility among all the relevant principals. That is, it must be possible to map externally visible activity (e.g., a transmitted packet) to the principal(s) i.e., users, responsible for that packet.
- **Decentralised Control:** PlanetLab supports decentralised control, which in turn requires minimising the aspects of the system that require global agreement. It allows autonomous organisations to federate in the formation of a global facility and for these organisations to define peering relationships with each other.
- **Efficient Resource Sharing:** PlanetLab decouples slice creation and resource allocation. This means all slices are given only best effort promises when they are first created. They then acquire and release resources over time, assisted by available brokerage services.

2.7.2 Architecture overview

The PlanetLab architecture comprises 10 architectural elements. These elements consist of two major pieces: (1) a set of software modules that run on each node, and (2) a collection of global mechanisms that implement PLC.

- **Node:** A node is a machine capable of hosting one or more virtual machines (VM).
- **Virtual Machine:** A virtual machine (VM) is an execution environment in which a slice runs on a particular node.
- **Node Manager:** A node manager (NM) is a program running on each node that creates VMs on that node, and controls the resources allocated to those VMs.
- **Slice:** A slice is a set of VMs, with each element of the set running on a unique node.
- **Slice Creation Service:** A slice creation service (SCS) is an infrastructure service running on each node. It is typically responsible, on behalf of PLC, for creation of the local instantiation of a slice, which it accomplishes by calling the local NM to create a VM on the node.
- **Auditing Service:** PLC audits the behaviour of slices and to aid in this process, each node runs an auditing service (AS). The auditing service records information about packets transmitted from the node, and is responsible for mapping network activity to the slice that generates it.
- **Slice Authority:** PLC, acting as a slice authority (SA), maintains state for the set of system-wide slices for which it is responsible.
- **Management Authority:** PLC, acting as a management authority (MA), maintains a server that installs and updates the software (e.g., VMM, NM, SCS) running on the nodes it manages. It also monitors these nodes for correct behaviour, and takes appropriate action when anomalies and failures are detected
- **Owner Script:** A designated VM named `site_admin` is created on each node and initialised with the keys belonging to the site's technical and administrative contacts. An owner script running in this VM calls the node manager to express its preferences. This is to provide owners with as much autonomy as possible to communicate how they want their nodes managed.
- **Resource Specification:** A resource specification (RSpec) is an abstract object used throughout PlanetLab. It is the main object maintained by a node manager and slice creation service, and it is the primary component of the record maintained for each slice by a slice authority. An RSpec can be viewed as a set of attributes, or name/value pairs.

2.7.3 User community

PlanetLab is intended to provide services (i.e. isolated infrastructure slices) for researchers and users who want to deploy, test and evaluate a specific service, protocol or software at application, middleware or control protocol level. However, PlanetLab's rich features allow it to be used for any kind of service (including network service) that may require an isolated and managed slice of infrastructure.

2.7.4 Mechanisms for providing virtualisation

2.7.4.1 *Implementation of virtualisation on Layer 3*

PlanetLab uses a module called VNet for providing virtualised network access on PlanetLab nodes. VNet provides a restricted form of raw IP and raw packet sockets, ensures isolation of traffic between slices, and supports a unique interface for accessing IP addresses, while maintaining compatibility with standard Linux/BSD socket APIs.

VNet relies on Linux's Netfilter system to associate every inbound and outbound IP packet with a connection structure (VNet can support TCP, UDP,ICMP, and PPTP). VNet then ensures that slices send and receive only packets associated with connections that they own. The slices can only:

- Send packets associated with new connections or connections that they initiated.
- Receive packets associated with connections that they initiated or bound.

When an IP packet is sent through a socket, it passes through VNet and is associated with a new or existing connection. If the connection is not already bound to a slice, VNet allows the packet through and binds the connection to the slice that sent the packet. If the connection is bound to a slice, and it is not the slice that sent the packet, the packet is dropped and an error is returned to the sending application.

When an IP packet is received by the stack, it also passes through VNet and is associated with a new or existing connection. If the packet was expected, that is, if the connection was bound by a slice, or the connection was initiated by a slice, VNet allows the slice to receive the packet. Otherwise, the packet can only be received by the root slice.

Based on the VNet mechanism, the PL-VINI, which is a prototype of a virtual network infrastructure (VINI) that runs on the public PlanetLab, has been developed.

VINI is a virtual network infrastructure that allows network users to evaluate their protocols and services in a realistic environment that also provides a high degree of control over network conditions. VINI allows users to deploy and evaluate their ideas with real routing software, traffic loads, and network events. VINI supports the following features:

- Flexible network topology.
  - Virtual point-to-point connectivity.
  - Unique interfaces per experiment.
  - Exposure of underlying topology changes.
- Flexible Forwarding and routing.
  - Distinct forwarding tables per virtual node.
  - Distinct routing processes per virtual node.
- Connectivity to external hosts.
  - Allowing end hosts to direct traffic through VINI.

    - Ensuring return traffic from external services flows back through VINI.
- Support for simultaneous experiments.
    - Resource isolation between different simultaneous applications/users/experiments.
    - Distinct external routing adjacencies per virtual node.

PL-VINI enables arbitrary virtual networks, consisting of software routers connected by tunnels, to be configured within a PlanetLab slice. A PL-VINI virtual network can carry traffic on behalf of real clients and can exchange traffic with servers on the real Internet. Nearly all aspects of the virtual network are under the control of the experimenter, including topology, routing protocols, network events and, ultimately, even the network architecture and protocol stacks.

PL-VINI is an infrastructure that supports virtual networks. It provides some features that a wide range of network experiments should find useful, such as:

- Virtual network topologies, consisting of UML instances connected by virtual point-to-point Ethernet links, running within a PlanetLab slice. PL-VINI matches packets to virtual links (implemented by UDP tunnels) based on the Ethernet MAC header, and so has no inherent dependencies on Layer 3 protocols.
- Binding resources to an application/experiment to ensure that the virtual network can forward packets at a sustained rate and with low latency.
- Processes running in the same slice as an experiment can inject traffic into it using a TUN/TAP interface running on every PlanetLab node.
- Clients can inject traffic into a virtual network using OpenVPN, which runs on Linux, Windows 2000/XP and higher, OpenBSD, FreeBSD, NetBSD, Mac OS X, and Solaris hosts.
- Packets can exit an overlay and transit the public Internet via a NAT gateway.

2.7.4.2 *Implementation of virtualisation on Layer 2*

PL-VINI can emulate L2 functionality over IP and provide Layer 2 virtualisation.

2.7.4.3 *Implementation of virtualisation on Layer 1*

This feature is not supported.

2.7.4.4 *Implementation of computing virtualisation*

PlanetLab provides an effective mechanism for server virtualisation and isolation such that each part of the server can independently run a specific service.

2.7.4.5 *Management of virtualised infrastructure*

PL-VINI relies on control and management features provided by PlanetLab for creating and managing virtual infrastructures. As explained above, these include: node manager, slice creation, slice authority and management authority.

2.7.4.6 *Control of virtualised infrastructure*

PL-VINI provides the virtual network users with flexible amounts of control and realism in the infrastructure. This makes VINI an environment that is suitable for running controlled services and applications as well as for deploying long-running services. Control refers to the user's ability to introduce exogenous events (e.g., failures, changes in traffic volume, etc.) into the system.

2.7.4.7 *Implementation of user interface*

PL-VINI provides a set of unique UML (User-Mode Linux) based interfaces for each virtual infrastructure and its associated virtual nodes. Users can fully access their slice through these interfaces and configure their own slice or perhaps run their own control mechanisms over the slice.

2.7.5 Multi-domain support

PL-VINI benefits from a unique mechanism called multiplexed BGP. Using this mechanism, each virtual infrastructure can interconnect with other virtual infrastructures or with an external network domain.

2.7.6 Testbed implementation and availability

PlanetLab is a global testbed which is continuously growing by the addition of new servers and locations. The European part of PlanetLab is referred to as PlanetLab Europe (PLE) and supports similar features to PlanetLab. PLE is the result of the first phase of the OneLab project. PLE extends the PlanetLab service across Europe federating with other PlanetLab infrastructures worldwide [OneLab]. They are open for educational institutes and NRENs to join. Therefore PL-VINI can be tested over the PlanetLab or OneLab infrastructures for specific functionality that may be required by the GÉANT community.

2.7.7 Current status and roadmap

Europe is investing in an extension of PLE through the second phase of the OneLab project, which is expected to finish in 2010. The objective is to extend the testbed in terms of locations and number of servers, and also in terms of virtualisation features that will help create a more complex and heterogeneous virtual testbed/network [OneLab].

2.7.8 References

[InVINIVeritas]	"In VINI Veritas: Realistic and Controlled Network Experimentation", A. Bavier, N. Feamster, M. Huang, L. Peterson, J. Rexford. SIGCOMM'06, September 11–15, 2006, Pisa, Italy. http://conferences.sigcomm.org/sigcomm/2006/discussion/showpaper.php?paper_id=1
[OneLab]	OneLab http://www.onelab.eu/
[PlanetLab]	PlanetLab http://www.planet-lab.org/
[PlanetLabArch]	"PlanetLab Architecture: An Overview", L. Peterson, S. Muir, T. Roscoe and A. Klingaman, May 2006. http://www.planet-lab.org/files/pdn/PDN-06-031/pdn-06-031.pdf

2.8 AKARI

Note that at the time of writing this document, extremely limited information was publicly available for the AKARI project. For this reason, no concrete information on the project's achievements can be provided, only information regarding AKARI's vision. However, as mentioned in Section 1 "Introduction" on page 4, it is expected that a detailed overview of AKARI will be provided in the second deliverable, DJ1.4.2, due March 2012. GN3 JRA1 Task 4 is currently in the process of establishing a channel of communication with the AKARI team dedicated to virtualisation topics. The Memorandum of Understanding between GN3 and AKARI, once signed, should make more tangible information available.

2.8.1 Introduction

The Internet is the object of various projects whose aims are to identify the limits of the current networking models and propose alternatives to circumvent them. There are two approaches when dealing with "Future Internet" development. The first one is to develop enhancements with reference to the current situation. Therefore some limitations have their counterpart answers. For example, IPv6 was initially developed in order to cope with IPv4 address depletion, while mobile IP is meant to extend users' mobility. An alternative is the "Clean Slate" approach, where the IP protocol, whether IPv4 or IPv6, is not even part of the answer. AKARI is the Japanese initiative related to "Future Internet Networks", the GENI and 4WARD projects being the US and EU counterparts respectively.

AKARI's vision considers virtualisation and particularly network virtualisation as a major technology enabling the possibility to deploy various large-scale network ecosystems in parallel. As network virtualisation pushes the limit of node virtualisation up to the core infrastructure, it is now even possible to deploy several instances of potential Internet networks without relying on the Internet itself.

Network virtualisation would therefore promote:

- Competition between these potential network instances.
- Cooperation between these network instances.
- Easier comparison between these instances as they exist in parallel.

- Relevant experiment results as the technology can enable several large-scale testbed deployments worldwide without incurring additional infrastructure cost.

2.9 Cloud Services

2.9.1 Amazon Virtualisation

2.9.1.1 *Introduction*

Amazon provides a number of commercially available virtual computing services under the label Amazon Web Services (AWS). These include on-demand computing and storage services, plus a virtual IP network service for the specific purpose of including Amazon's computing services as part of an existing enterprise network.

2.9.1.2 *Architecture overview*

The Amazon Web Services offering comprises a number of different elements. Those most relevant to JRA1 T4's investigations include:

- Elastic Compute Cloud (EC2).
- Simple Storage Service (S3).
- Virtual Private Cloud (VPC).

This section will look most closely at EC2, which uses S3 for long-term storage, and will also note the characteristics of VPC.

Amazon EC2 separates the types of image that may be run from the instances themselves. A number of operating system environments (both free and chargeable) are made available as Amazon Machine Images (AMI), and the user may create as many short-term instances of these images as they require. The resources (such as CPU time) that an instance consumes are billed on an hourly basis, encouraging the short-term creation and termination of instances.

Users create an instance, after signing up for EC2, by choosing an AMI from those available (custom AMIs may be built, but only selected kernels may be used), "bundling" it so that the image may be launched repeatedly by multiple instances based on a single ID, and then launching one or more instances of it.

The instances are configured by the user to be launched in a region. While the user does not have visibility of the physical infrastructure inside a region, the regions are kept separate and user-identifiable by Amazon, and are dispersed geographically. This gives the user some control over the location, and therefore network distance, of the instance, and also gives some assurance that instances in separate regions are unlikely to be affected by simultaneous maintenance.

Amazon provides a command line interface to the EC2 system for creation of instances.

The network connectivity of instances has been quite specifically defined. Each instance receives, on creation, an RFC1918 IPv4 address, which is associated exclusively with that instance for its lifetime. At this time, a public IPv4 address is also associated with the instance, but this is implemented by means of Network Address Translation (NAT) under Amazon's own control.

For internal connectivity between instances, Amazon directs that the private IPv4 addresses should be used, to ensure that the least costly and most efficient path is implemented.

The public IPv4 address that is assigned by default could come from anywhere in the pool of addresses assigned to the EC2 service. Since instances may be short-lived, the public addresses that the user receives are likely to change over time. Where this is unsuitable, Amazon provides "Elastic IP addresses" – that is, IPv4 addresses that are associated with a user's account, and which may be assigned to that user's instances at the user's request. (To ensure that a user does not appropriate more addresses than they require, an hourly charge is imposed on addresses that remain unassigned to an instance.)

2.9.1.3 *User community*

The AWS offering appears to be aimed at a wide spread of users, and the individual services may be used somewhat independently for differing purposes.

The EC2 service provides computing power on demand, with an emphasis on the ability to scale up and down with the user's needs at a given moment.

The S3 service provides reliable storage over the Internet. The user community here ranges from power users that make heavy use of EC2, to home users who may use Jungledisk to create a network-attached drive on a Windows desktop machine for backups.

The VPC service is aimed at enterprises with an existing corporate network that require secure access to EC2 services from within their firewall.

2.9.1.4 *Mechanisms for providing virtualisation*

Implementation of virtualisation on Layer 3

Amazon VPC provides a particular, narrowly defined virtual networking service for the purpose of connecting EC2 resources to a corporate network by means of an IPsec tunnel instead of directly over the public Internet.

The user assigns a range of IPv4 addresses to the virtual private cloud, which may be further subnetted and assigned to EC2 instances. The documentation notes that if an IP range is subnetted, the subnets will be connected by means of a star topology centred on a logical router.

The user then has the facility to bring up an IPsec tunnel between this logical router and the user's own gateway. As part of the setup, sample configurations are provided for Cisco IOS and Juniper Junos-based routers.

Once configured, the EC2 instances are accessible via their RFC1918 IPv4 addresses, but not via the Internet. Instead, their external traffic is encrypted over the IPsec tunnel and is only accessible by means of the user's gateway. The gateway can be configured by the user to make the virtual private cloud available on the Internet network, behind their firewall, if that is appropriate.

Implementation of virtualisation on Layer 2

AWS does not provide for virtualisation on Layer 2.

Implementation of virtualisation on Layer 1

AWS does not provide for virtualisation on Layer 1.

Implementation of computing virtualisation

Amazon EC2 provides for computing virtualisation by means of short-term instances, with long-term reliable storage provided by Amazon S3, as described in Section 2.9.1.2 "Architecture overview" on page 52.

Management of virtualised infrastructure

Amazon provides a command line interface for the creation of virtual infrastructure, and also an API which is documented on the AWS website. The creation and ongoing existence of AWS elements attract an hourly charge which closely ties the cost of the service to the consumption of resources.

Control of virtualised infrastructure

Once an instance is created and booted, the user has full control of the operating system within, with the exception that there are specific limitations on the operating system kernels that can be run. A variety of Linux kernels for various distributions are available and are kept updated with recent patches. The user has full administrator/root access to their instances.

Implementation of user interface

The user interface for creation and manipulation of the instances is via the command line.

In order to log in and work with a given instance, the user may use secure shell (ssh) (for UNIX/Linux-based instances) or Remote Desktop Protocol (for Windows-based instances).

2.9.1.5 *Multi-domain support*

AWS does not provide for multi-domain support.

2.9.1.6 *Testbed implementation and availability*

AWS is a production service (although VPC is noted as being in beta). There is no testbed, but the production infrastructure is available for use, at a price.

2.9.1.7 *Current status and roadmap*

The service is in production, with VPC noted as being in beta and the user is required to apply for access. A pricing table for EC2 is available at [AWSPrice]. While future plans for the service are not revealed publicly, the documentation notes that the service does not currently support IPv6, and IPv6 support is being investigated.

2.9.1.8 *References*

[AWS]	Amazon Web Services http://aws.amazon.com/
[AWSPrice]	https://aws.amazon.com/ec2/#pricing
[AWSUserGuide]	Amazon Web Services User Guide http://docs.amazonwebservices.com/AWSEC2/2009-08-15/UserGuide/

2.10 Summary Comparison

The following table provides a summary of the virtualisation technologies described above. More specifically, the following aspects are considered for each virtualisation technology:

- Protocol dependency: states whether there is any protocol dependency for the users of the virtualised infrastructure.
- Network layer virtualisation: the OSI layers for which virtualisation is provided.
- Computing virtualisation: whether computing virtualisation is provided.
- Virtualisation technology: how virtualisation is achieved.
- Reason for deploying virtualisation: what is the added value that virtualisation offers.
- User community: the community that the virtualisation technology is targeting.
- Who manages the virtualised infrastructure. Two broad roles are identified:
  - Physical infrastructure owner – the party that owns the substrate infrastructure that is used for implementing virtualisation.
  - User – the party that exploits the subset of the physical infrastructure that constitutes the virtualised infrastructure.
- Management tools: what are the tools that are used for managing the virtualised infrastructure. It should be specified if these tools are used by the physical infrastructure owner or the user.
- Offered services: the services that are offered to the users.
- Potential use in a multi-domain environment: whether deployment of the virtualisation framework is possible in a multi-domain environment.

Virtualisation technology	Protocol dependency	Network layer virtualisation	Computing virtualisation	Virtualisation technology	Reason for deploying virtualisation	User community	Who manages the virtualised infra-structure	Management tools	Offered services	Potential use in multi domain environment
FEDERICA	None. A user can define its own networking technology	L3/2	Yes	Inherent virtualisation capabilities of L3/2 NEs (Junos and software router/switch) and servers (VMware ESXi)	Creation of parallel virtual environ-ments (slices) aimed at supporting research on networking	Network researchers	Physical infrastructure owner and/or users	Traditional tools. Tools for slice-oriented provisioning, management and monitoring are under development	Creation of L2/L3 VPNs (including virtual computing elements)	Open to be inter-connected/ federated with other e-infrastructure and service management frameworks, e.g. IPsphere
MANTICORE	IP	L3/2	No	Tool based on the IaaS framework in combination with inherent virtualisation capabilities of Juniper routers	Project focus		Users	From IP NOCs to end users	Creation of IP-based VPNs	

Virtualisation technology	Protocol dependency	Network layer virtualisation	Computing virtualisation	Virtualisation technology	Reason for deploying virtualisation	User community	Who manages the virtualised infra-structure	Management tools	Offered services	Potential use in multi domain environment
Phosphorus	IP	L1	No	UCLPv2 based on web service technology	Resource partitioning and network virtualisation through network resource slicing	NRENs and e-science community	Users	Web-based GUI	Static connectivity provisioning Static network topology creation and control Static network slicing	Yes
4WARD	None. The concept is independent of specific protocols	L3	N/A	N/A	Co-existence of multiple architectures and smooth migration path New business models	Addressing all users	Physical infrastructure owner	Implementa-tion of a "Virtualisa-tion Management Interface"	N/A	N/A

Virtualisation technology	Protocol dependency	Network layer virtualisation	Computing virtualisation	Virtualisation technology	Reason for deploying virtualisation	User community	Who manages the virtualised infra-structure	Management tools	Offered services	Potential use in multi domain environment
GENI	None. A user can define its own networking technology	L3/2/1	Yes	Virtualisation middleware (GENI Management Core – GMC) and inherent virtualisation capabilities of L3/2/1 NEs and servers	Project focus	Network researchers	Physical infrastructure owner	Management tools are under development. They are accessed by the physical infrastructure owner via the GENI operator portal	Researchers can define their own experiments over the virtualised infrastructure via the researchers portal	N/A
PlanetLab/ VINI/OneLab	IP	L3/2	Yes	PlanetLab and VINI virtualisation tools	Infrastructure slicing for protocol testing	Network, application and service researchers but not limited to any community	Slicing and creation of virtual infrastructure in a central authority basis. Management of each slice can be done by users through a	Specific management tool and interface is available	Multiple independent network and server slice over same infrastructure	Yes

Virtualisation technology	Protocol dependency	Network layer virtualisation	Computing virtualisation	Virtualisation technology	Reason for deploying virtualisation	User community	Who manages the virtualised infra-structure	Management tools	Offered services	Potential use in multi domain environment
							dedicated interface			
AKARI	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Amazon	IP	L3	Yes	Amazon specific tool (Amazon Web Services-based virtualisation tool)	Efficient sharing of resources. Increased utilisation of resources	Everyone. Commercial service	Users	CLI, API	Elastic Compute Cloud (EC2) Simple Storage Service (S3) Virtual Private Cloud (VPC)	No

3 Initial Requirements Analysis

3.1 Introduction

This section presents the results of an initial study of NRENs' requirements for using infrastructure virtualisation technologies in the near future. The analysis reported here focuses on the requirements of three NRENs only. This is a pilot analysis; its results are expected to create the foundation and framework for a more comprehensive study to be carried out during 2010 covering as many NRENs as possible, as well as GÉANT. The complete study results and their analysis will be presented in the second deliverable, DJ1.4.2, due March 2012.

3.2 Description of Survey and Participants

To carry out the requirements survey, a questionnaire was prepared by the JRA1 Task 4 team, and circulated among JRA1 Task 4 participants. Participants contacted the individuals in their respective NRENs who have responsibility for the production use of virtualisation, both internally and customer-facing, on the layers covered by the Task, and went through the questionnaire with them.

The results provide a summary of how different NRENs plan to use virtualisation over the coming 1-3 years, their experiences so far, and their views on the cooperative use of virtualisation in GÉANT. The participating NRENs were GRNET, HEAnet and RENATER.

3.2.1 Questionnaire and results

The questionnaire was made up of four sections:

- Existing use of virtualisation technologies and services.
- Other potential applications.
- Areas of specific or strategic interest for application of virtualisation.
- Risk analysis.

The questions, and the results, are given below. The figures in [square brackets] denote the number of respondents who gave the answer.

1. Existing use

1.1 Do you have existing server virtualisation in use?

"Server virtualisation" is the provision of multiple instances of an operating system installation (perhaps the same OS or perhaps different) on a single host – for example using User Mode Linux, VMware or Parallels.

1.1.1 [0] not at all
[1] used by NREN staff only
[2] part of the internal network infrastructure
[1] service provided to clients

1.1.2 Provide example applications.
[1] Usenet news feeder & reader
[1] IPv6 tunnel broker for SixXs
[1] Federation WAYF
[1] New OS release testing
[1] Core network services: DNS, email, Radius server-EduRoam, Directory Services
[1] User services: server hosting, e.g. GN2-JRA3 VM.
[1] PerfSONAR RRD-MA, EDUROAM, Gatekeeper H323, Meeting Maker Server, SIP router / monitor

1.1.3 What tools do you use to manage this?
[1] VMware ESX 3.5 vCenter
[1] VMware ESXi server
[1] Currently XEN and openVZ (KVM and others to be evaluated)
[1] In house tool under development

1.1.4 What is the user community using this at the moment?
[1] Own NOC, some researchers
[1] Customer NOCs, EU funded projects (e.g. GN2)

1.2 Do you have existing network equipment virtualisation in use?

1.2.1 [3] not at all
[ ] used by NREN staff only
[ ] part of the internal network infrastructure
[ ] service provided to clients

Comment: “We evaluated Junos but faced isolation problems between the guest and host OSes. To be reconsidered.”

1.2.2 Provide example applications.
[1] Virtual CPE for customers (planned but not currently implemented)

1.2.3 What tools do you use to manage this?
[1] MANTICORE, JUNOS logical routers

1.2.4 What is the user community using this at the moment?

[1] Plan to be own NOC

1.3 Do you have existing link/network virtualisation in use? This may include link/network virtualisation, or such technologies as layer 2 or layer 3 VPNs.

1.3.1 [0] not at all
[0] used by NREN staff only
[3] part of the internal network infrastructure
[3] service provided to clients

1.3.2 Is it integrated in some way with the routing/switching equipment virtualisation?
[1] Not at this time, planned to be
[2] No

1.3.3 Provide example applications.
[1] Underlying infrastructure for IP network
[2] Connecting customer branches
[1] Direct connections for researchers to other countries
[1] Provisioning of a customer's IP address to a VM hosted at the GRNET core.
[1] Dedicated "any to any" MPLS L3VPN and L2 pseudowire

1.3.4 What tools do you use to manage this?
[1] Bluenet tool, based on GRnet's tool
[1] In house built, fully automatic configuration/provisioning tool.
[1] CLI, SNMP polling/trap, Syslog with the use of 3rd party tools (Opensource / in house such as RANCID, in house SNMP pollers and weathermaps).

1.3.5 What is the user community using this at the moment?
[2] Own NOC
[1] customer IT departments, researchers
[1] L2VPN and L3VPN used by RENATER users having offices spread among several locations.

2. Other use by users

2.1 To your knowledge, do your users already use virtualisation services, other than the ones already mentioned in previous sections?

[1] VMware, server virtualisation

2.2 Do you, or your users to your knowledge, use any outsourced virtualisation services?

[1] Not substantially

3. Areas of interest

3.1 What are the aspects of virtualisation, based on the definitions above, that you are most interested in?

[1] Mainly for network equipment. Server virtualisation is fully deployed in GRNET.
[1] Rapid deployment
[1] Small incremental cost of additional virtual servers

[1] Ability to gain maximum use from hardware
[1] Ease of migration of applications to new hardware
[1] Deployment of virtual servers on behalf of clients
[1] L1/L2/L3 apects

3.2 In what timescale do you expect to implement these? What response times would you require from such a service?

[1] 2010 - 2012
[1] Less than a year.
[1] No particular timescale
[1] Expected response time for server provisioning: 10 minutes
[1] No particular requirement for response time.

3.3 Do you see requirements for virtualisation in a multidomain environment?

[1] Yes
Examples: for overflow from our own infrastructure
for providing geographically/topologically distant server resources to our clients
[1] Weak

3.4 Do you see requirements for a federated virtualisation environment?

[1] Yes.
[1] Moderate.
[0] No

3.5 Do you see requirements or advantages of federated virtual services that span multiple NRENs? If so, how might you expect to use it?

[2] We would use it in addition to our own infrastructure
[0] We would use it instead of our own virtual infrastructure
[0] We would not use it

3.6 In which type of resources/services would you be interested?

Type of resource/service	In addition to our own	Instead of our own	Would not use it
Server	[1]	[ ]	[1]
Layer 1 net equipment	[1]	[ ]	[1]
Layer 2 net equipment	[1]	[ ]	[1]
Layer 3 net equipment	[1]t	[ ]	[1]
Layer 1 VPNs	[2]	[ ]	[1]
Layer 2 VPNs	[3]	[ ]	[1]

Type of resource/service	In addition to our own	Instead of our own	Would not use it
Layer 3 VPNs	[2]	[ ]	[1]
Service	[2]	[ ]	[1]

3.7 How can virtualisation, at any level, help your user community?

[1] Advanced networking experiments

3.8 If you used it, would you facilitate access to your customers?

[0] We would use it for NREN internal operations only
[3] We would facilitate access to our customers

4. Risk analysis

4.1 Do you see risks in the area of

data protection
[1] Yes
[1] Low
operational complexities (including virtual resource management)
[1] Yes
[1] Moderate
deployment
[1] Low
stability and maturity of the virtualisation-enabling technologies
[1] Yes. Existing experience with Dell & VMware suggest that there is still a good deal of work to do in providing “turnkey” hardware/software packages which provide virtual server environments
[1] Moderate
performance
[1] Yes. Well known profiling tools exist to help mitigate this, with proper planning and experimentation before production.
[1] Low

3.3 Results Analysis

3.3.1 Findings about existing installations

The survey asked about current server virtualisation, and the uses to which it is put. One NREN said that it is used only by NREN staff, two that it is used as part of their internal network infrastructure, and one that it is used as a service provided to clients. Example applications included applications that would once have been situated on their own servers for reasons of security, such as Usenet News and an IPv6 tunnel broker, and core

network services such as DNS, email, eduroam radius and directory services, and also destructive testing of new OSs.

Tools used include VMware ESX3.5 vCentre, VMware ESXi server, XEN, openVZ and, in one case, an in-house tool that is under development.

One NREN reported its own NOC as part of the existing user community, while two reported external users including some researchers and EU-funded projects e.g. GN2.

None of the NRENs reported network (routing/switching) equipment virtualisation in use. One reported that they evaluated JUNOS logical routers, and faced isolation problems between the guest and the host – but that this is to be reconsidered. An example application provided is planned virtual CPE for customers, which is planned to be implemented using MANTICORE and JUNOS logical routers. The user community in this case would, in the first instance, be the NREN's own NOC providing services to their users.

All three NRENs reported some network link virtualisation in use, both as part of internal infrastructure and as services provided to clients. One reported that they plan to integrate it with network equipment virtualisation. Example applications include, in two cases, connecting customer branches, and in one case each, underlying infrastructure for their IP network, direct connections for researchers to other countries, provisioning of a customer's IP address to a hosted VM, and dedicated "any to any" Layer 3 VPNs and Layer 2 pseudowire connections.

Tools used include in two cases a tool based on GRNET's own in-house code, and also management tools including SNMP polling/traps and syslog with third-party open source or in-house tools. The user community for this application is, in two cases, the NREN's own NOC, and in one case each customer IT departments/researchers and remote users with offices spread among several locations.

In general there is little other use of virtualisation services by users that NRENs are currently aware of, other than those already mentioned above. One NREN reported use of VMware and server virtualisation independently among its users.

3.3.2 Findings about expected future work and interest

The survey asked about aspects of virtualisation that NRENs would be most interested in. The response in this case was quite spread among all layers that Task 4 covers. One NREN reported interest mainly in network equipment as server virtualisation is already fully deployed, while another mentioned rapid deployment and small incremental cost of additional virtual servers as well as ability to gain maximum use in hardware. One mentioned deployment of virtual servers on behalf of clients.

The timescales over which NRENs expect to deploy these services range from 2010-2012 in one case to less than one year in another, with no particular timescale mentioned by the third.

One expected response time for server provisioning was mentioned as 10 minutes.

One NREN expressed a requirement for virtualisation in a multi-domain environment, giving examples of overflow from their own infrastructure, and providing geographically or topologically distant server resources to their clients. The other NRENs did not express a strong requirement here.

One NREN sees a requirement for a federated virtualisation environment, and one sees a moderate requirement. Two NRENs stated that they would use federated virtual services spanning multiple infrastructures in addition to their own infrastructure; none said they would use it instead of their own infrastructure.

All three NRENs expressed an interest in Layer 2 VPNs, with two NRENs expressing interest in Layer 1 VPN, Layer 3 VPN and service virtualisation. Only one NREN expressed an interest in server and Layer 1-3 network equipment virtualisation. All these expressions of interest were in addition to their own infrastructure, rather than instead of it. All three NRENs said that they would facilitate access to the infrastructure for their customers.

3.3.3 Risk assessment

NRENs gave varied assessments of the risks that they feel they face in using virtualisation services across multiple domains. One noted data protection as a strong concern, while another felt this risk was low. One noted operational complexities as a strong concern, and another said this risk was moderate. Deployment was considered low risk. The stability and maturity of the technologies was a strong concern for one NREN, noting that existing experience had suggested there is a good deal of work to do in providing “turnkey” packages that provide virtual server environments; this risk was assessed as moderate by another. One NREN also noted performance as a concern, mentioning that well-known profiling tools exist to help mitigate this, with proper planning and experimentation before production.

4 Proposal for Technological Proof of Concept for GÉANT Virtualisation Service: an Integrated Approach

4.1 Introduction

This section defines virtualisation services within the context of GÉANT and proposes an approach to implementing a technological proof of concept within GÉANT and associated NREN infrastructures. The recommendations in this section are based on the results of the comprehensive study reported in Section 2 and the initial requirements analysis reported in Section 3.

JRA1 Task 4's proposal for a technological proof of concept for GÉANT virtualisation services is to take advantage of each of the existing relevant European projects and initiatives, while providing the capability to incorporate the outcome of future relevant projects and frameworks.

JRA1 Task 4 isn't aiming to promote a specific solution or framework for a technological proof of concept for GÉANT virtualisation services. Instead, it aims to propose a solution for integrating and interworking existing virtualisation mechanisms and solutions at different layers, leaving the choice of suitable virtualisation technologies for each domain to individual NRENs.

4.2 Layer-Based Virtualisation Services

JRA1 T4 has defined virtualisation at four different layers as described below:

- Computing virtualisation: binding together several computing servers or partitioning a server into several independent servers by means of an operating system.
- Layer 3 network virtualisation: creating Layer 3 (IP) related functionalities on any type of hardware. This includes partitioning a Layer 3 router into several independent routers to create a Layer 3 virtual network topology.
- Layer 2 network virtualisation: creating Layer 2 (Ethernet) related functionalities on any type of hardware. This includes partitioning a Layer 2 switch into several independent switches tor create a Layer 2 virtual network topology.

- Layer 1 (optical) network virtualisation: creating a Layer 1 virtual network topology by binding together Layer 1 resources (e.g., SDH timeslots, wavelength, fibre). This includes partitioning (slicing) of Layer 1 devices such as optical switches.

In each of the above layers, virtualisation can occur according to the user community's needs. Indeed, projects such as LHC, for instance, could request from an NREN and/or GÉANT a dedicated Layer 3 VPN. GRID5K is a French Grid Research Infrastructure that has its own physical optical VPN on top of RENATER infrastructure. The JIVE project relies on a set of "stitched lightpaths" that is also called a set or string of "Single Point of Failure".

As described in Section 2 on page 8, there are various initiatives and projects that feature virtualisation services and technologies. However, each of these projects is focused on a specific area and their solutions only deal with a restricted number of layers:

- The Phosphorus project is focused on Layer 1 virtualisation using an ARGIA/UCLP framework.
- The FEDERICA project deals with Layer 2, Layer 3 and computing virtualisation (and will address Layer 1 in the future).
- MANTICORE implements network virtualisation at Layer 3 through the use of Logical Routers.

4.3 An Integrated Approach to a Technological Proof of Concept for GÉANT Virtualisation Services

This section proposes an initial, multi-layer and multi-domain infrastructure virtualisation mechanism based on a combination of solutions and tools developed by the EU projects mentioned in Section 2. Without reinventing the wheel, the proposition is to integrate existing Layer 1, Layer 2, Layer 3 and computing virtualisation tools both horizontally and vertically.

Throughout this implementation, it is essential to observe the security impacts of the work, e.g., as related to the ISO framework FCAPS (Fault Detection, Configuration, Accounting, Performance, Security.) As discussed in Section 1.1, virtualisation presents security challenges on a number of levels: the isolation of the virtual infrastructures themselves; integrity and privacy of the data within the virtual infrastructures; and managing the possible large-scale replication of vulnerabilities or exploitable infrastructures.

We will also ensure that the IPv6 protocol is fully utilised. It will be supported in the administrative domain, the management of virtual infrastructure, and in the infrastructure itself. Where weaknesses in feature parity with IPv4 are encountered, these will be highlighted. Further, however, we will investigate how the features of IPv6, such as neighbour discovery or anycasting, can be used to improve the infrastructure.

4.3.1 Vertical Integration

Vertical integration refers to the integration of virtualisation tools at different layers. It aims to build an integrated tool that can provide virtual Layer 3 services over virtual Layer 2 services over virtual Layer 1 services.

4.3.2 Horizontal Integration

Horizontal integration refers to interfacing two virtual infrastructures provisioned by the same or different virtualisation tools belonging to the same or different domains (i.e. the federation of two virtual infrastructures). A generic functional architecture for the proposed solution in a single domain is shown in Figure 4.1. As shown in this figure, with the proposed mechanism it is possible to partition a multi-layer physical infrastructure into several independent virtual infrastructures. Virtual infrastructures, although sharing the same physical infrastructure, appear to their users as independent infrastructures with a specific topology, each of which can be controlled and managed independently and deploy different control mechanisms and protocols. Furthermore, it gives NRENs the capability to deploy virtualisation at only one layer (e.g. Layer 3) or at multiple layers depending on their preference or their preferred solution, while fully interacting with other virtualised domains.

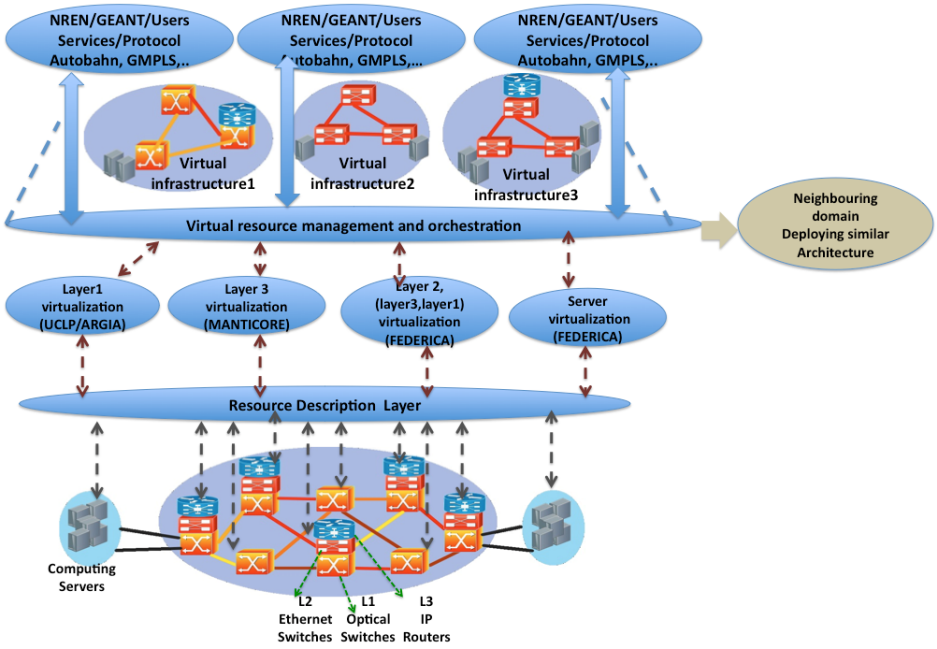

Figure 4.1: Generic functional architecture for the proposed multi-layer virtualisation mechanism

4.4 Integration Building Blocks

The proposed architecture comprises three buildings blocks:

- Resource description layer.
- Virtualisation layer.
- Virtual resource management and orchestration layer.

Each of these is described below.

4.4.1 Resource Description Layer

The resource description layer uses a semantic description language such as Network Description Language (NDL [NDLRef]) or Virtual Resource and Interconnection Network Description Language (VxDL [VxDLRef]) to homogeneously describe the functionality, capability and connectivity of the heterogeneous physical infrastructure elements (computing and network). This provides a standard information model covering different network/computing technologies and elements at different layers, with which any virtualisation mechanism can interface (using a standard interface mechanism) to retrieve information about the underlying physical infrastructure.

4.4.2 Virtualisation Layer

The virtualisation layer uses existing UCLP/ARGIA, FEDERICA and MANTICORE tools and software to virtualise Layer 1, Layer 2 / computing and Layer 3 resources. The choice of the specific tool and framework is dependent on the requirements of the individual NREN (e.g. Layer 1 or Layer 3 virtualisation) and the underlying infrastructure technology. This layer must also provide a standard interface where the virtual resource management and orchestration layer can exchange information about the status of virtual resources as well as the setting up and tearing down of virtual resources and connectivity.

4.4.3 Virtual Resource Management and Orchestration Layer

The virtual resource management and orchestration layer is the most important part of the proposed architecture. It orchestrates virtualisation at all three layers and enables the architecture to provision and control the multi-layer virtualised infrastructure. Furthermore this layer is the key enabler for multi-domain virtualisation by providing the interface to the “Virtual Resource Management and Orchestration Layer Resource Management” of a neighbouring domain for sharing virtual resources. Finally and most importantly, this layer provides an interface which allows users, applications and infrastructure owners to control and manage individual virtual infrastructures or request the creation/modification of a virtual infrastructure.

5 Next Steps

5.1 Introduction

The next main step for JRA1 Task 4 is to undertake a proof of concept to validate the proposed integrated virtualisation approach over a small testbed and analyse its outcome. This is expected to provide the foundation and basic framework for the detailed definition and design of the proposed architecture and its associated building blocks as defined in Section 4.

In addition, T4 will evaluate the advantages, disadvantages and risks of virtualisation compared to traditional operation, particularly with regard to security, an evaluation for which input and evidence will become firmer and more readily available as the concept becomes more mature, the technologies progress beyond the research and development stage, and the work of the standardisation bodies is finalised.

The question of security will be addressed more fully in the next version of the deliverable (due March 2012), in the next version of the questionnaire and use case as well as in the technical text. By that time the security-related findings of the IETF virtual network research group (VNRG) should also be available.

As well as continuing to gather information about 4WARD, AKARI and GENI, T4 will extend its review of projects that feature infrastructure virtualisation to cover additional EU projects that have just started or are due to start in 2010.

This section outlines the proof of concept, and introduces some of the new EU projects.

5.2 Proof of Concept and Prototype Implementation

Because of time and resource limitations within JRA1 Task 4, the prototype implementation and proof of concept will be carried out on a very small scale using existing resources within Task 4 participants' facilities. Two virtualisation frameworks and two small-scale testbeds have been selected for prototype implementation and proof of concept testing (as shown in Figure 5.1 below): the University of Essex Layer 1 testbed (small scale) deploying Phosphorus (UCLP) a the HEANET Layer 3 testbed (small scale) deploying MANTICORE.

The testing and prototype implementation for the proof of concept will be in seven phases as follows:

Phase	Name	Description
1	Prototype test scenario	Define a virtualisation scenario involving Layer 1 and Layer 3 technologies.
2	Testbed setup	Set up a small-scale Layer 1 testbed (very limited number of nodes) in the University of Essex and a small-scale Layer 3 testbed (very limited number of nodes) in HEANET.
3	Resource description layer prototype	Extend NDL or VxDL to describe the resources and their connectivity in both testbeds.
4	Resource description layer interface prototype	Develop a common interface for interconnecting MANTICORE and UCLP to the resource description layer.
5	Initial functionality test	Carry out an initial functionality test of MANTICORE and UCLP interoperation with resource description layer and physical infrastructure. In this phase, the HEANET testbed will be interconnected to the University of Essex testbed over GÉANT with very limited bandwidth.
6	Virtual resource management and orchestration layer prototype	Implement a very limited virtual resource management layer, which is able to provide Layer 3 virtualised network (HEANET) over Layer 1 virtualised infrastructure (University of Essex).
7	Verification and functionality test	Test the functionality of the proposed approach in detail. Identify issues and shortcomings for consideration at the subsequent detailed architecture design stage.

Table 5.1: Proof of concept testing and prototype implementation

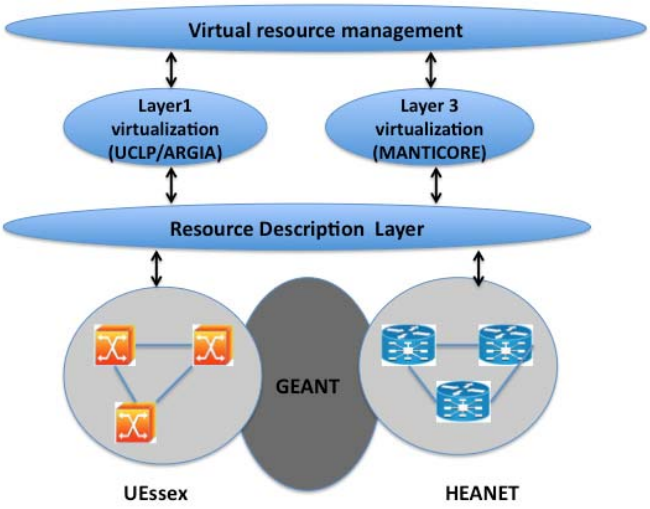

Figure 5.1: Architectural diagram of virtualisation test prototype

5.3 Additional Projects

Recently several significant projects and initiatives that address different aspects of infrastructure virtualisation technologies and their application in the future Internet have been funded by the European Union's Seventh Framework Programme for Research and Technological Development (FP7). These projects include:

- **OpenFlow in Europe: Linking Infrastructure and Applications (OFELIA)**. The aim of the OFELIA project is to create a unique experimental facility based on OpenFlow technology that allows researchers not only to experiment on a test network but also to control the network itself precisely and dynamically. The project is due to start in September 2010. There is no publicly available documentation at this stage.

- **MANTHYCORE**. The objective of MANTHYCORE is to provide the European research community with IP networks as a service over the NRENs' e-infrastructure for the benefit of their research activities. The project is due to start in September 2010. There is no publicly available documentation at this stage.

- **Generalised Architecture for Dynamic Infrastructure Services (GEYSERS)**. GEYSERS' vision is to qualify optical infrastructure providers and network operators with a new architecture, to enhance their traditional business operations. Optical network infrastructure providers will compose logical infrastructures and rent them out to network operators; network operators will run cost-efficient, dynamic and mission-specific networks by means of integrated control and management techniques [GEYSERS].

The projects above have either just started or will start later in 2010 and have strong core research activities on infrastructure virtualisation. JRA1 T4 participants are also members of these projects, so T4 is well placed to monitor their progress and outcome and will adopt their solutions where appropriate. A detailed overview of these projects and their approach toward virtualisation will be provided in the next deliverable, due March 2012.

6 Conclusions

This deliverable has reported on a comparative study of several major infrastructure virtualisation frameworks, both existing and under-development, in Europe, USA and Japan. From the results of the study it is evident that the European research community, helped by the drive and commitment of the NRENs, has managed to achieve significant progress on infrastructure virtualisation technologies through projects such as FEDERICA, MANTICORE and Phosphorus. These projects are complementary and, combined together, they can provide virtualisation of Layer 1, Layer 2 and Layer 3 networks as well as computing resources. JRA1 Task 4's proposals for GÉANT virtualisation services therefore aim to build on the developments and achievements of these projects.

A questionnaire was designed and distributed to a select group of NRENs to collect their requirements for virtualisation services and their expectations of a virtualised infrastructure service-provisioning system. The initial results, reported here, indicate a unanimous requirement for virtualisation by NRENs, with each stressing a different aspect of virtualisation and related services, i.e. Layer 1, Layer 2, Layer 3 and computing virtualisation.

The current virtualisation technologies resulting from the projects described in this report are still in their research and development stage. It is therefore not realistic to propose a specific virtualisation technology solution to the NREN and GÉANT community. Instead, this report has proposed an integrated architecture approach that allows the different virtualisation technologies deployed across the NRENs and GÉANT to be integrated, offering a multi-domain, multi-layer and multi-technology virtualisation service. This approach enables each NREN to adopt one or multiple virtualisation technologies, depending on their requirements, and to offer to its users inter- and/or intra- domain as well as multi-layer infrastructure virtualisation services.

Evaluating the advantages, disadvantages and risks of virtualisation compared to traditional operation will form a key part of future JRA1 T4 work. At this stage in the development lifecycle, however, the majority agree on the benefits and necessity of network virtualisation, as demonstrated by the EU projects reviewed in this document, none of which has so far reported significant drawbacks.

References

[4WARD]	4WARD Project http://www.4ward-project.eu/
[4WARDControlPlane]	R. Bless, C. Werle,: “Control Plane Issues in the 4WARD Network Virtualization Architecture”, Proceedings of the KiVS Workshop on Overlay and Network Virtualization, Kassel, Germany, March 2009 http://eceasst.cs.tu-berlin.de/index.php/eceasst/article/view/225/216
[4WARD-VNet]	http://www.4ward-project.eu/index.php?s=overview&c=WP3
[Argia]	E. Grasa, S. Figuerola, A. Forns, G. Junyent, J. Mambretti, “Extending the Argia Software with a Dynamic Optical Multicast Service to support High Performance Digital Media", accepted for publication in Elsevier journal of Optical Switching and Networking Volume 6, Issue 2, Recent trends on optical network design and modeling – selected topics from ONDM 2008, April 2009 http://www.sciencedirect.com/science?_ob=ArticleURL&_udi=B7GX5-4VXMPRK-1&_user=10&_rdoc=1&_fmt=&_orig=search&_sort=d&_docanchor=&view=c&_searchStrId=1077462195&_rerunOrigin=google&_acct=C000050221&_version=1&_urlVersion=0&_userid=10&md5=d6a9240fe4221f8d93569fc5868870be
[AWSREFERENCE]	Amazon Web Services Details of document that the reference marker is referring to (Including URL is available) e.g. S. Ubik, V. Smotlacha, S. Trocha, S. Leinen, V. Jeliazkov, A. Friedl, “MS.3.7.5: Report on Passive Monitoring Pilot” http://aws.amazon.com/
[AWSPrice]	https://aws.amazon.com/ec2/#pricing
[AWSUserGuide]	Amazon Web Services User Guide http://docs.amazonwebservices.com/AWSEC2/2009-08-15/UserGuide/
[DesigningaPlatform]	N. Egi, A. Greenhalgh, M. Handley, M. Hoerdt, F. Huici, L. Mathy, and P. Papadimitriou, "Designing a Platform for Flexible and Performant Virtual Routers on Commodity Hardware External link mark", Workshop on Overlay and Network Virtualization, Invited Paper, Kassel, Germany, Kassel, Germany, March 2009.
[ExtendingMANTICORE]	A. Berna, E. Grasa, S. Figuerola, “Extending MANTICORE to Manage IP and Virtual Machine Slices in the FEDERICA project”, Proceedings of TERENA Networking Conference 2009, TNC2009, June 2009, Malaga, Spain http://tnc2009.terena.org/core/getfile.php?file_id=319
[FEDERICA]	The FEDERICA Project https://www.fp7-federica.eu
[FEDERICADNA2.2]	Deliverable DNA2.2: “FEDERICA User Community and Requirements” http://www.fp7-federica.eu/documents/FEDERICA-DNA2.2.pdf
[FEDERICADSA1.1]	Deliverable “DSA1.1: “FEDERICA Infrastructure” http://www.fp7-federica.eu/documents/FEDERICA-DSA1.1.pdf
[FEDERICAUIK]	FEDERICA User Information Kit - http://www.fp7-federica.eu/users/users.php

[FEDIPsphere]	J. Pons-Camps, S. Figuerola, E. Grasa, “Prototype for the interoperability between FEDERICA slices and other IP domains by means of the IPsphere Framework”, Proceedings. of TERENA Networking Conference 2009, TNC2009, June 2009, Malaga, Spain http://tnc2009.terena.org/core/getfile.php?file_id=410
[GARR]	GARR, the Italian Academic & Research Network - http://www.garr.it
[GENI]	http://www.geni.net
[GENI-GDD-07-44]	L. Peterson (ed.), “GENI: Global Environment for Network Innovations – Facility Design”, GDD-07-44, March 2007
[GENI-Overview]	“The Global Environment for Network Innovations (GENI)”, April 2009 http://www.geni.net/wp-content/uploads/2009/04/geni-at-a-glance-final.pdf
[GEYSERS]	http://www.geysers.eu/
[Globus]	http://www.globus.org/
[IaaS]	Infrastructure as a Service http://www.iaasframework.com
[IETFProbStatement]	S. Jeong, M.-K. Shin, T. Egawa, H. Otsuki, “Network Virtualization Problem Statement”, draft-shin-virtualization-meta-arch-01.txt, Internet Draft, March 4, 2010 http://www.rfc-editor.org/rfc/internet-drafts/draft-shin-virtualization-meta-arch-01.txt
[ImplementingNV]	P. Papadimitriou, O. Maennel, A. Greenhalgh, A. Feldmann, L. Mathy, "Implementing Network Virtualization for a Future Internet External link mark", 20th ITC Specialist Seminar on Network Virtualization, Hoi An, Vietnam, May 2009
[InVINIVeritas]	“In VINI Veritas: Realistic and Controlled Network Experimentation”, A. Bavier, N. Feamster, M. Huang, L. Peterson, J. Rexford. SIGCOMM’06, September 11–15, 2006, Pisa, Italy. http://conferences.sigcomm.org/sigcomm/2006/discussion/showpaper.php?paper_id=1
[IPsphere]	http://www.ipsphereforum.org
[MANTICORESvc]	E. Grasa , X. Hesselbach, S. Figuerola, V. Reijs, D. Wilson, J.-M. Uzé, L. Fischer, T. de Miguel, “‘The MANTICORE project: Providing users with a Logical IP Network Service”’, TERENA Networking Conference 2008, Bruges, Belgium http://tnc2008.terena.org/schedule/presentations/show.php?pres_id=98
[MonWithVR]	J. Navrátil, T. Košnar, J. Furman, T. Mrázek, V. Krmíček, “Monitoring Of Overlay Networks With Virtual Resources”, Proceedings. of TERENA Networking Conference 2009, TNC2009, June 2009, Malaga, Spain http://tnc2009.terena.org/core/getfile.php?file_id=409
[NDLRef]	https://noc.sara.nl/nrg/ndl/
[NetworkVirtualization]	R. Bless, C. Werle, "Network Virtualization from a Signaling Perspective External link mark",. Future-Net '09 International Workshop on the Network of the Future 2009 in conjunction with IEEE ICC 2009, Dresden, June 16th-18th, 2009.
[NVChallenges]	N. M. M. K. Chowdhury, R. Boutaba, "Network virtualization: state of the art and research challenges", Communications Magazine, IEEE, vol.47, no.7, pp.20-26, July 2009
[OneLab]	OneLab http://www.onelab.eu/
[Phosphorus]	www.ist-phosphorus.eu
[PhosphorusD1.4]	Phosphorus project deliverable D1.4 “Definition and development of the Network Service Plane and northbound interfaces development” http://www.ist-phosphorus.eu/files/deliverables/Phosphorus-deliverable-D1.4.pdf
[PhosphorusPresn]	“Phosphorus: Lambda User Controlled Infrastructure for European Research” http://www.ist-phosphorus.eu/files/press/Phosphorus-general_presentation.pdf
[PlanetLab]]	PlanetLab http://www.planet-lab.org/

[PlanetLabArch]	"PlanetLab Architecture: An Overview", L. Peterson, S. Muir, T. Roscoe and A. Klingaman, May 2006. http://www.planet-lab.org/files/pdn/PDN-06-031/pdn-06-031.pdf
[Routing]	D. Wilson, "Routing Integrity in a World of Bandwidth on Demand", TNC 2006 http://www.terena.org/events/tnc2006/programme/presentations/show.php?pres_id=242
[TowardsInterop]	Y. Zaki, L. Zhao, J. Jimenez, K. Mengal, A. Timm-Giel, C. Goerg,. "Towards Interoperability among Virtual Networks in the Future Internet", ICT-MobileSummit, Santander, Spain, June 2009
[UCLP]	http://www.canarie.ca/canet4/uclp/uclp_software.html
[UCLPv2]	E. Grasa, S. Figuerola, A. López, G. Junyent, M. Savoie, "UCLPv2, a Network Virtualization Framework built on Web Services", IEEE Communications Magazine, Feature Topic on Web Services in Telecommunications part II, pp. 126-134
[VirtualRouters]	N. Egi, A. Greenhalgh, M. Handley, M. Hoerdt, F. Huici, and L. Mathy, "Towards High Performant Virtual Routers on Commodity Hardware External link mark", ACM CoNEXT? , Madrid, Spain, December 2008
[VxDLRef]	http://www.ens-lyon.fr/LIP/RESO/Software/vxdl/home.html
[WSDL]	www.w3.org/TR/wsdl

Glossary

AMI	Amazon Machine Images
API	Application Programming Interface
AS	Auditing Service
AWS	Amazon Web Services
BGP	Border Gateway Protocol
CLI	Command Line Interface
CM	Component Manager
CPS	Circuit Processing System
CPU	Central Processing Unit
E2E	End to End
EC2	Elastic Compute Cloud
FCS	Fast Circuit Switch
FP7	EU's Seventh Framework Programme for Research and Technological Development
FPGA	Field Programmable Gate Array
G^{2}MPLS	Grid GMPLS
GGID	GENI Global Identifier
GMC	GENI Management Core
GMPLS	Generalised Multi-Protocol Label Switching
GPO	GENI Project Office
GUI	Graphical User Interface
IaaS	Infrastructure as a Service
IGP	Interior Gateway Protocol
IP	Internet Protocol
IPsec	Internet Protocol security
JRA1 T4	Joint Research Activity 1 Future Network Task 4 Current and Potential Uses of Virtualisation
KVM	Keyboard Video Mouse
MA	Management Authority
MIB	DefinitionManagement Information Base
NAT	Network Address Translation
NIC	Network Interface Controller
NM	Node Manager
NREN	National Research and Education Network
NRPS	Network Resource Provisioning System
NSF	National Science Foundation (US)

NSP	Network Service Plane
OPP	Optical Patch Panels
PCN	Programmable Core Nodes
PEC	Programmable Edge Clusters
PEN	Programmable Edge Nodes
PF	Programmable Framer
PIP	Physical Infrastructure Provider
PLC	PlanetLab Consortium
PLE	PlanetLab Europe
PoP	Point of Presence
PP	Packet Processor
PPS	Packet Processing System
PWN	Programmable Wireless Nodes
RMS	Resource Management System
RSpec	Resource Specification
RSpec	Resource Specification
RT	Request Tracker
S3	Simple Storage Service
SA	Slice Authority
SCS	Slice Creation Service
SNMP	Simple Network Management Protocol
SOA	Service-Oriented Architecture
SP	Service Provider
ssh	secure shell
UCLP	User-Controlled Lightpath Provisioning
UDP	User Datagram Protocol
UML	User-Mode Linux
UPB	FEDERICA User Policy Board
VLAN	Virtual Local Area Network
VM	Virtual Machine
VMI	Virtualisation Management Interface
VMM	Virtual Machine Manager
VN	Virtualisation Node
VNO	Virtual Network Operator
VNP	Virtual Network Provider
VNRG	IETF Virtual Network Research Group
VPC	Virtual Private Cloud
VPN	Virtual Private Network
VRS	Virtual Resource Service
VSMS	Virtual Slice Management Server
vSwitch	Virtual Switch
WSDL	Web Service Description Language
WSS	Wavelength Selective Switch
XC	Cross Connect
XML	Extensible Markup Language