UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

NOVI tools and algorithms for federating virtualized infrastructures

Lymberopoulos, L.; Grammatikou, M.; Potts, M.; Grosso, P.; Feteke, A.; Belter, B.; Campanella, M.; Maglaris, V.

DOI
10.1007/978-3-642-30241-1_19

Publication date
2012

Document Version
Final published version

Published in
The Future Internet

Link to publication

Citation for published version (APA):
Lymberopoulos, L., Grammatikou, M., Potts, M., Grosso, P., Feteke, A., Belter, B., Campanella, M., & Maglaris, V. (2012). NOVI tools and algorithms for federating virtualized infrastructures. In F. Álvarez, F. Cleary, P. Daras, J. Domingue, A. Galis, A. Garcia, A. Gavras, S. Karnourskos, S. Krco, M-S. Li, V. Lotz, H. Müller, E. Salvadori, A-M. Sassen, H. Schaffers, B. Stiller, G. Tselentis, P. Turkama, & T. Zahariadis (Eds.), *The Future Internet: Future Internet Assembly 2012: from promises to reality* (pp. 213-224). (Lecture Notes in Computer Science; Vol. 7281). Springer. https://doi.org/10.1007/978-3-642-30241-1_19

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)

Download date:25 Sep 2021

NOVI Tools and Algorithms for Federating Virtualized Infrastructures

Leonidas Lymberopoulos[1], Mary Grammatikou[1], Martin Potts[2], Paola Grosso[3], Attila Fekete[4], Bartosz Belter[5], Mauro Campanella[6], and Vasilis Maglaris[1]

[1] National Technical University of Athens
[2] Martel Consulting
[3] Universiteit van Amsterdam
[4] Eötvös Loránd Tudományegyetem
[5] Poznan Supercomputing and Networking Center
[6] Consortium GARR

Abstract. The EC FP7/FIRE STREP project *NOVI - Network Innovation over Virtualized Infrastructures* - explores efficient approaches to compose virtualized e-Infrastructures towards a holistic Future Internet (FI) cloud service. Resources belonging to various levels, i.e. networking, storage and processing are in principle managed by separate yet inter-working providers. In this ecosystem NOVI aspires to develop and validate methods, information systems and algorithms that will provide users with isolated slices, baskets of resources and services drawn from federated infrastructures. Experimental research accomplished thus far concludes the first phase of NOVI, with early prototypes of semantic-aware advanced control & management plane components being deployed and tested. The NOVI testing environment is based on combining PlanetLab and FEDERICA, two dissimilar virtualized experimental infrastructures with attributes widely anticipated in a FI cloud. This federated testbed is stitched at the data plane via the NSwitch, a distributed virtual switch developed within NOVI.

Keywords: NOVI, Future Internet, FIRE, Virtualization, Federation.

1 The NOVI Project: Goals and Objectives

We report in this paper experimental work within the *NOVI FIRE project* [1] towards a semantic-aware control and management plane for federating heterogeneous virtualized infrastructures and for establishing data plane connectivity amongst virtual resources offered by separate virtualized infrastructures. The goal is to offer automated advanced capabilities to users of the federation: intelligent resource mapping, policy-driven access and resource allocation, context aware resource discovery, transparent data plane connectivity and monitoring of combined user slices and substrate resources across domains. Experimental tool development and validation are performed in a testbed environment, based on two dissimilar virtualized infrastructures: *FEDERICA* [2] and *PlanetLab* [3]. The former, partially supported by the EC FP7/Capacities Programme, provides users with a combination of Virtual

F. Álvarez et al. (Eds.): FIA 2012, LNCS 7281, pp. 213–224, 2012.
© The Author(s). This article is published with open access at SpringerLink.com

Machines and Logical Routers, interconnected via Layer 2 VLAN technology extended over dedicated circuits provisioned by European National Research & Education Networks (NRENs) and *GÉANT* [4]; the latter is a popular experimental infrastructure, partially supported by the US NSF *GENI Programme* [5], that offers collections (slices) of virtual computing resources (slivers) within more than a thousand hosts, globally distributed over the legacy Internet. The selection of these platforms provides NOVI with a combined testbed, exposing experiments to a wide range of attributes as expected in a Future Internet federated cloud: FEDERICA combines commercial virtualization tools providing virtual machines and logical routers, interconnected with gigabit controlled connectivity; PlanetLab offers a highly distributed virtual machine selection interconnected over the existing public Internet, thus presenting distributed applications experiments with actual connectivity limitations and unpredictable wide area networking behavior. In conclusion, the combined PlanetLab and FEDERICA testbed for NOVI's experimental research captures basic features applicable in federated heterogeneous environments that are expected to serve a wide range of user communities.

The paper is organized as follows. Section 2 presents our work on a domain-independent Information Model aiming to capture the main abstractions of shared resources and services within a NOVI federation. Section 3 presents the main components of NOVI's federated control and management plane and provides an overview of NOVI's distributed virtual switch (*NSwitch*) for data plane stitching. Section 4 presents the combined testbed for NOVI's prototype deployment and experimentation. Finally, section 5 concludes the paper and provides directions for future work in the remainder of the project.

2 NOVI Information Model

An agreed-upon Information Model (IM) provides consistent and shared semantics and descriptions of available resources and services in a federated environment. In NOVI we developed a novel IM and the associated data models as existing IM efforts, listed in NOVI Public Deliverable *D2.1: Information Models for Virtualized Architectures* [6] did not cover our two-fold objective: (a) to support the modeling abstractions to cater for a federation of infrastructures, e.g. the FEDERICA and PlanetLab platforms of the NOVI's testbed; (b) to include the necessary concepts so that can be used to model other Future Internet (FI) infrastructures that could participate in a NOVI-like federation.

We fully embraced a Semantic Web approach and defined data models using the *Web Ontology Language - OWL* [7]. This choice has been driven by the desire to support *reasoning* and *context awareness*, which in turn allow NOVI to create efficient and complex services with resources available within the federation.

The NOVI IM consists of three distinct but related ontologies; this modular approach is chosen on purpose to make the model more easily usable outside the project by parties interested in specific aspects. The NOVI IM defines a *resource ontology*, a *monitoring ontology* and a *policy ontology*.

The *Resource Ontology* provides the concepts and methods to describe the resources offered by Future Internet platforms and how they are connected together in a federated environment. This ontology provides the basis for topology and request descriptions and the terminology for describing physical nodes, virtual nodes, virtual topologies, etc. The Resource Ontology supports the operation of all the services of *NOVI's Federated Control & Management Architecture*, which will be presented in the Section 3 of this paper. For example, it is used to express requests within the *NOVI GUI* or by the *Resource Information Service* and the *Intelligent Resource Mapping Service* to communicate when coordinating the exchange of information about resources suitable for the embedding of virtual resources. The *Monitoring Ontology* extends the Resource Ontology to provide descriptions of the concepts and methods of monitoring operations, such as details about monitoring tools, how these relate to the resources, types of measurements that can be gathered etc. This ontology provides the primary support to the operation of the *Monitoring Service*. Finally, the *Policy Ontology* also extends the Resource Ontology by providing descriptions of the concepts and methods for the management and execution of policies defined within member platforms of a NOVI federation. This ontology supports the operation of the *Policy Service*. More information on the developed ontologies can be found in the project's public deliverable *D2.2: First Information & Data Models* [8].

3 NOVI Federated Data, Control and Management Plane Architecture

NOVI's novel algorithms, methods and services are initially based on the *Slice Federation Architecture - SFA* [9] as developed for the PlanetLab control & management plane federation. In SFA, a *resource specification - RSpec* is an XML-file describing resources bound and available to a user slice in terms of hardware characteristics, network facilities, constraints and dependencies on their allocation. NOVI extends SFA with advanced context-aware federation mechanisms (intelligent resource allocation, monitoring, policy management and virtualized resources discovery) and automating slice control & management operations anticipated within a complex NOVI federation.

The high level overview of the NOVI Data, Control & Management (C&M) architecture is shown in Fig. 1. It consists of three different layers:

1. At the bottom layer heterogeneous platforms (domains, infrastructures) contain the virtual resources to be allocated to user requests for combined slices. Data plane connectivity within a NOVI federated slice is achieved using *NOVI's Distributed Virtual Switch – NSwitch*
2. The middle layer components are used to provide basic C&M federation capabilities across platforms. In the figure we depict implementation choices referring to SFA (e.g. cross-domain authentication via synchronized registries and user-specified slice operations)
3. The top layer implements NOVI C&M services that aim at offering advanced capabilities to the federation users (e.g. intelligent resource mapping, policy-driven access and resource allocation, context aware resource discovery, transparent monitoring of combined user slices and substrate resources across domains). It leverages federation mechanisms of the middle layer (SFA), complementing them with advanced C&M functionality.

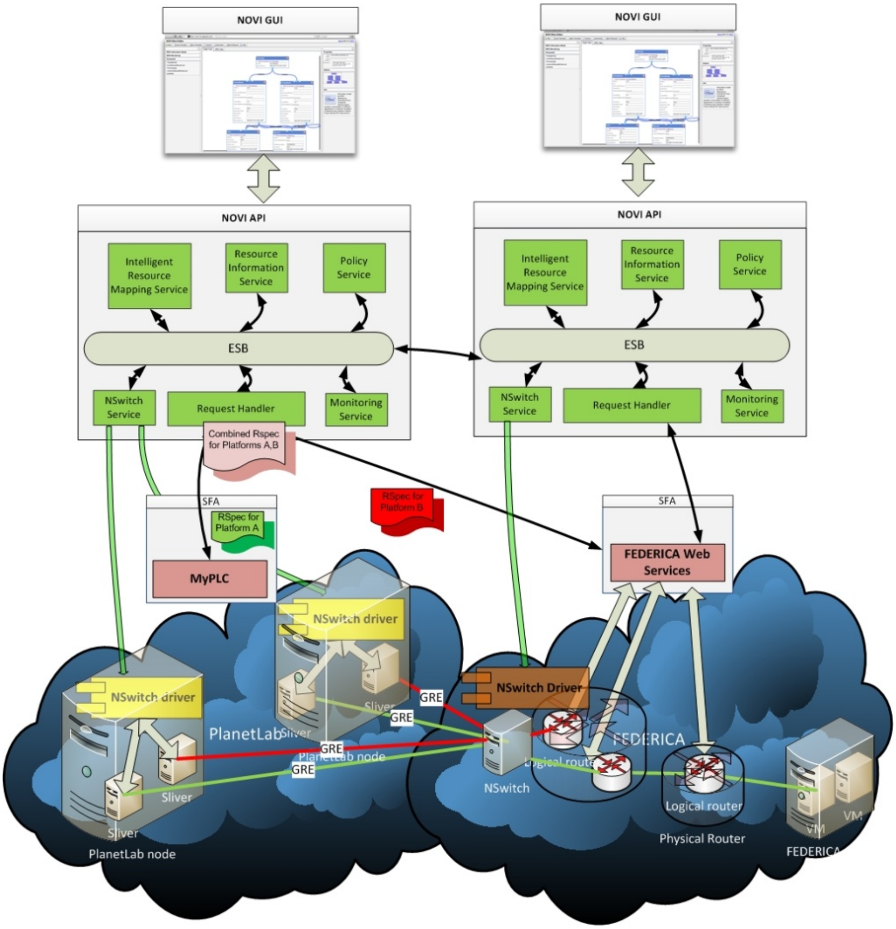

Fig. 1. NOVI Federated Data, Control & Management Architecture

For each infrastructure (platform) in the federation, as demonstrated within the NOVI testbed, separate NSwitch, SFA and NOVI C&M instances need to be deployed. In what follows we outline functionality of components within the latter.

3.1 NOVI API

The NOVI API provides the entry point for interacting with NOVI C&M services. It has three main tasks: (1) Accept requests from authenticated users containing resources requirements represented in NOVI Information model; (2) Handle and deliver the request to the appropriate component within NOVI Service Layer; (3) Provide user feedback on how their request is handled before the experiment starts being executed in its related NOVI slice.

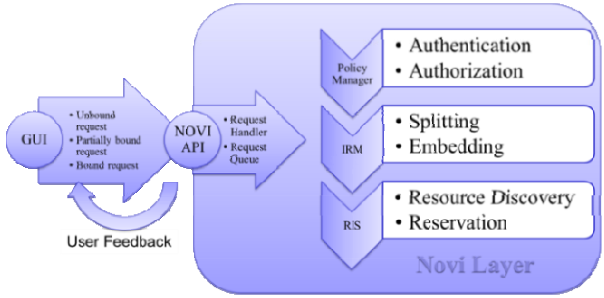

Fig. 2. Role of the NOVI API

As shown in Fig.2, the NOVI API receives requests from the NOVI GUI. The GUI is based on the *Ontology Instance Editor - OIntEd* [10], which was originally used to assist in the development phase of the NOVI IM and subsequently was customized to allow users to create and send requests for NOVI slices. In its current implementation, the NOVI GUI provides an intuitive drag-and-drop interface for this instantiation process and allows users to define relations between instantiated objects. For example, a user can define a virtual network topology along with the characteristics for requested resources. For every request, the GUI generates an OWL document based on the NOVI IM which is sent to the NOVI API by means of an HTTP post request.

Using the NOVI GUI (accessible online at *http://novi-im.appspot.com/*) the user can choose from the available ontologies in order to define the topology of the slice for his experiment.

3.2 Resource Information Service(RIS)

The *Resource Information Service (RIS)* acts as a single point of contact for other NOVI services to acquire information about the status of virtual and substrate resources. Resource discovery encompasses locating and retrieving information across the federated virtualized substrate network in a decentralized way with a scalable query process. RIS uses the *Request Handler* to communicate with the underlying platform, to reserve resources and to obtain the resource advertisements. It uses the *Monitoring Service* to query on the availability and the status of the resources and the *Policy Service* to get information related to the access rights or the users.

RIS exploits the features of the NOVI information model to improve the precision of resource discovery and to apply reasoning when selecting resources and services. It uses a database engine based on semantic web technologies, namely *Sesame* [11], for resource selection. The data are stored in the Sesame database as *RDF* triples [12]. The RIS uses the *AliBaba* tool [13] for the conversion of triples to Java objects. These Java objects that describe the concepts in the NOVI IM are used also by the other software components of the NOVI C&M architecture.

3.3 Intelligent Resource Mapping (IRM) Service

The *Intelligent Resource Mapping (IRM)* service for NOVI will enable embedding user requests for virtual topologies - resources (Virtual Networks - VNs) to the federated physical substrate network. This was initially formulated for a single domain (infrastructure) as *Virtual Network Embedding (VNE)*, an NP-Hard combinatorial problem [14]. In the NOVI federated profile, VNE had to be extended towards a multi-domain environment via graph spitting as in [15] and intelligent selection of intra-domain mapping.

Evaluation and testing of the embedding procedure for NOVI experiments require the appropriate representation of a VN request, formulated using the NOVI Information Model. The IRM gathers information from the *Resource Information Service (RIS)* and the *Monitoring Service* regarding available resources. As a first step, user requests for VN resources are apportioned to infrastructures that are members of a NOVI federation. Subsequently, single-domain VNE problems are formulated, resulting into sub-optimal allocation of virtual resources within the federated substrate.

A user may submit requests for standalone virtual resources, topologies of virtual resources and specific services regarding virtual resources/topologies. These requests may request specific mappings of virtual resources to substrate infrastructures. As specified by the *ProtoGENI RSpec* [16]. VN requests may contain a complete (pre-specified, bound), partial, or empty (free, unbound) mapping between virtual resources and available physical (substrate) resources.

3.4 Policy Service

The *Policy Service* is used to provide the functionality of a policy-based management system, where policies are used to define the behavior governing the managed environment. As reported in [17], we plan to extend the *Ponder2* policy framework [18] with functions to support enforcement of mission policies. These will be used to define the obligations of a member-infrastructure within a NOVI federation.

We currently provide support for (1) *Access Control* policies that specify what rights users have on specific resources and (2) *Event-Condition-Action* policies enforcing management actions upon events indicating failures or performance degradation. Events are received by the *Monitoring Service*. Implementation details are reported in *NOVI Public Deliverable D2.2* [8].

3.5 Monitoring Service (MS)

One of the main challenges for a *Monitoring Service (MS)* in a heterogeneous federated virtual environment is the diverse combination of monitoring tools deployed within different infrastructures. To address this, NOVI developed generic *Monitoring Ontologies,* enabling us to describe, parameterize and use diverse active and passive monitoring tools provided within constituent federated infrastructures. Thus, users are required to specify metrics to be measured and do not rely on monitoring tools.

MS collects information about specific resources and measurement metrics. Monitoring can be performed on slivers (virtualized resources allocated to a user) or on the physical substrate resources (hosts, links, paths, etc.). It is possible to obtain *passive monitoring* information from resources or from repositories, and *active monitoring* information as requested. Depending on usage scenario, MS can support two main tasks: The first task is triggered by the *Resource Information Service* prior to resource allocation to collect monitoring and measurement information from the substrate, which can be used by the IRM service to ensure that the constraints defined in the resource requests are satisfied. The second task is used after the resource reservation, to perform slice monitoring for diagnostic and watchdog purposes, i.e. to check the current status of a given set of virtual resources across a NOVI federation.

MS supports three advanced high level monitoring tools, i.e. *SONoMA* [19], *HADES* [20], and *Packet Tracking* [21]. These tools enable users to measure key performance metrics of the network, for example the one-way delay, the round-trip time, the packet loss, or the available bandwidth. Obviously, the MS can obtain from hosts via command line SSH CPU utilization, memory consumption, disk usage etc.

In Fig. 3, we provide a screenshot of the MS GUI provided to users of NOVI, who can choose from available metrics and specify required and optional parameters. Note that users do not need to specify which monitoring tool will measure the selected metric, as these can vary across infrastructures (testbeds) in a NOVI federation. Measurements of selected metrics, associated with monitoring tasks, can be managed individually, independently from the other monitoring task. The monitoring tasks can be started, stopped or removed from the task list. The results of the measurements can be read from the console of the GUI, or uploaded to a database within the Resource Information Service, or trigger event-condition-action policies in the *Policy Service*.

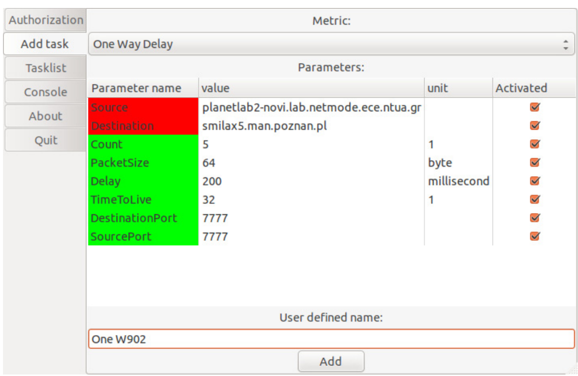

Fig. 3. GUI of the NOVI Monitoring Service

3.6 Request Handler Service

The main purpose of this service is to perform two types of operations: (1) Handling of resource allocation requests to the underlying platforms and (2) handling external calls coming from testbeds that are members of a NOVI federation.

For the first type, the NOVI IM needs to be translated into the underlying platform resource specification model. Given the key role played by SFA in the federation of PlanetLab and FEDERICA for NOVI experiments, a translation needs to be performed between NOVI IM concepts and the ones in *SFA RSpec v2* [16].

Translation in the opposite direction is needed to handle remote calls from the federated platforms. External calls from underlying platforms occur when the *Resource Information Service (RIS)* needs updates with new information, i.e. the presence of new resources or resource status notification updates received from the *Monitoring Service (MS)*. RIS will only store the static part of the information from the monitoring ontology, while the dynamic parts will be obtained by directly calling MS.

The *Request Handler*, as shown in Fig. 1, communicates via *RSpec* with a server running *SFA*. Since the SFA code was initially developed for PlanetLab, we just had to installa private SFA server for the PlanetLab part of our testbed. However, there is no SFA implementation for FEDERICA; thus we developed an appropriate *FEDERICA RSpec* and an *FEDERICA SFA Wrapper* service acting as FEDERICA's Aggregate Manager (see Fig. 1). More information can be found in *D2.2: First Information and Data Models* [8].

3.7 The NOVI Distributed Virtual Switch - NSwitch

The *NSwitch* distributed software complements NOVI's federation architecture by providing a unified way of interaction between heterogeneous domains at the data-plane. It enables a virtual entity in one domain to be connected at *protocol layer 2 (L2)* with another virtual entity in a remote domain taking into account concurrence, isolation, elasticity and programmability aspects.

The *NSwitch* was developed, deployed and tested over the PlanetLab – FEDERICA testbed above. Its functionality was driven by the need to combine virtual resources belonging to these two virtualization infrastructures that employ dissimilar communication protocols and hypervisors. Notably, PlanetLab does not provide users with data-plane connectivity options, using IP/BGP over the legacy Internet. By contrast, FEDERICA provides users with data-plane network virtualization choices, e.g. providing Juniper Logical Routers and Ethernet switches based on L2 VLAN technology. User-configurable VLANs are carried by SDH 1 Gbps circuits provisioned by NRENs and GÉANT into a controlled WAN environment, thus enabling repeatability of experiments over the FEDERICA infrastructure.

In order to map PlanetLab slivers into an L2 broadcast domain we adopted an approach similar to the one developed within the *VINI* [22] project in the US that introduced a set of extensions to the PlanetLab kernel and tools. VINI used an *Ethernet over GRE – EGRE* [23] mechanism to provide point-to-point virtual network capabilities to user configured virtual resources over the Internet. *NOVI's NSwitch* enhanced VINI's capabilities by introducing the *Open vSwitch (OVS)* [24] S/W in PlanetLab's host OS, thus enabling point-to-multipoint virtual links. OVS, compared to the VINI multiple bridges, provides better management of multiple EGRE tunnels within a host. Furthermore, distributed OVS instances can be centrally managed via the *OpenFlow protocol* [25]. This feature will be adopted in NOVI's *Spiral 2* phase.

On the FEDERICA side, L2 data plane connectivity is provided by means of VLANs used by Logical Routers, Switches and VMs. The *NSwitch* functionality provides the mapping of EGRE key values of packets originating from PlanetLab slivers to VLAN IDs.

3.8 Integration of NOVI C&M Services

In each platform (member of a NOVI federation) the *C&M Services* components of the top layer in Fig. 1 interact with each other and communicate (1) northbound with the NOVI GUI and (2) south-bound with the middle layer (SFA). The north-bound interface is the *NOVI API*, while the south-bound interface is the *Request Handler Service*. Intra-domain *C&M Services* within the top layer exchange messages via an *Enterprise Service Bus - ESB* [26]. Inter-domain C&M services can communicate (1) via the Request Handler using SFA services (e.g. for slice creation across domains) or (2) directly in a peer-to-peer mode via secure RPCs in cases that SFA mechanisms were deemed as inadequate (e.g. for remote interactions of monitoring services).

An example of C&M service integration is the *Slice Creation Use Case* detailed in NOVI Public Deliverable *D4.2: Use Cases* [27], which also provides an overview of initial usage scenarios of the project. In summary, an authenticated experimenter is authorized to use a set of resources across domains, as confirmed by the relevant per-domain *Policy Services*. He may then request a desired topology using the *NOVI GUI*. The virtual topology request is then passed to the *IRM* through the NOVI API. Prior to solving the inter-domain VNE, *IRM* contacts *RIS* to identify available resources that would fulfill the requirements imposed by the experimenter. *RIS* interacts with the *Monitoring Service* to obtain information regarding the status (e.g. availability, capacity, usage) of resources. Finally, when an appropriate mapping of virtual-to-substrate resources is identified, reservation requests in the form of *RSpecs* are sent by the *Request Handler* to the relevant testbed(s) slice manager(s).

NOVI developed a software integration framework for its C&M Services architecture. It follows the Service Oriented Architecture complemented with the Event Driven Architecture to enable synchronous and asynchronous communication between components. The integration framework was implemented using Java technologies. However it supports communication of components written in different programming languages via a range or specific bridges such as: *Jython* [28], a Python engine for Java; *JRuby* [29] for the Ruby language; *JNI* [30], a Java Native Interface API for components written in C/C++.

4 NOVI Experimentation Testbed

To test and validate NOVI's prototypes, a testbed environment was configured consisting of private PlanetLab and FEDERICA resources. This testbed enables NOVI software developers to run, test, refine and validate their software components and prototypes, according to the experimentally driven methodology followed in the project. NOVI developers are able to configure operational slices within the NOVI testbed, in isolation from production services of the two virtualization platforms.

In fact, the testbed uses the actual FEDERICA substrate and virtualization services, i.e. *Juniper MX480 Logical Routers* [31] and *VMWare ESXi* [32] Virtual Machines (VMs). By contrast, the public *PlanetLab* could not be used as is for NOVI's experiments that require S/W upgrades, embedding custom code to C&M tools within *MyPLC* [33] and root access rights to host hypervisors. Note that PlanetLab is a widely used federated infrastructure [34], consisting of PLC (PlanetLab Central), PLE (PlanetLab Europe) and PLJ (PlanetLab Japan), each with a single instance of MyPLC. Experiments affecting PlanetLab's host OS and C&M S/W are usually performed on private testbed installations and this practice was also adopted in NOVI.

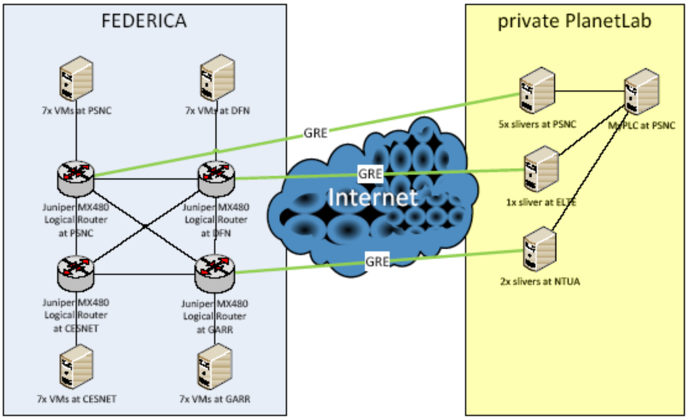

Fig. 4. Example of an experimentation slice in the NOVI testbed

Fig.4 presents the topology of one operational slice used to test control and management plane components, detailed in NOVI Public Deliverable *D4.1: Software Architecture Strategy & Developers' Guide* [35]. This slice is comprised of *three FEDERICA core PoPs* located in PSNC (Poznan, Poland), DFN (Erlangen, Germany) and GARR (Milano, Italy). These are connected over the Internet via *GRE tunnels* to private PlanetLab nodes in NTUA (Athens, Greece), ELTE (Budapest, Hungary) and PSNC (Poznan Poland).

To isolate the slice in Fig. 4 from other NOVI slices using the same FEDERICA core PoPs, *Logical Routers* are created on the *Juniper MX480* routers. The *open source MyPLC* (PlanetLab's C&M software) is deployed at PSNC, managing the private PlanetLab testbed.

An illustration of a typical slice deployed in the NOVI testbed is the *NOVI-MONITORING* devoted for validating NOVI's monitoring methods (active and passive) and their corresponding tools. Measurements assembled via this slice are depicted in Fig. 3.

5 Summary and Future Work

This paper reported a summary of NOVI's current research outcomes: The ontology-based NOVI information model, the advanced services within NOVI's federated Control & Management architecture and the distributed virtual switch architecture (NSwitch). More technical details on the aforementioned work can be found in NOVI's public deliverables and publications that are available at the project's website, http://www.fp7-novi.eu.

NOVI's research revealed a plethora of areas requiring further investigation. We list below some of them:

- *Information Model:* Need for constant updating of NOVI's IM evolutionary ontologies, e.g. incorporating NSwitch parameters. Short-term ontology enhancements based on feedback coming from validation of the integrated prototype implementation.

- *GUI*: Implementation of a user feedback mechanism and support for grouping of graphical objects, thus simplifying the level of information details of user requests.
- *Resource Information Service:* Validation of the distributed architecture model and support for more complex semantic queries that aim to provide efficient resource discovery mechanisms, towards facilitation of virtual network embedding processes.
- *Monitoring Service:* Implementation of a mechanism allowing different monitoring tools over dissimilar platforms to cooperate and contribute to multi-domain measurements of the same metric.
- *Policy Service:* Definition and deployment of role based access control (RBAC) policies and enhancement of the policy engine to support enforcement of inter-domain obligation Ponder2 policies
- *NSwitch:* Integration of the NSwitch control plane with the other components of NOVI's C&M plane.

We are currently at the end of *Spiral 1* of the project, having deployed a first version of an integrated prototype on the NOVI testbed (Section 4 of this paper). It is expected that the *Spiral 2* subsequent effort will complement functionality and performance of NOVI's prototypes, based on Spiral 1 results obtained from validation experiments on the NOVI experimental testbed.

Open Access. This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

[1] NOVI FP7 STREP Project, http://www.fp7-novi.eu
[2] Szegedi, P., Figuerola, S., Campanella, M., Maglaris, V., Cervelló-Pastor, C.: With Evolution for Revolution: Managing FEDERICA for Future Internet Research. IEEE Communications Magazine 47(7), 34–39 (2009)
[3] PlanetLab, http://www.planet-lab.org
[4] GÉANT, http://www.geant.net/pages/home.aspx
[5] Global Environment for Network Innovations (GENI), http://www.geni.net/
[6] D2.1: Information Models for Virtualized Architectures, http://www.fp7-novi.eu/index.php/deliverables
[7] Web Ontology Language (OWL), http://www.w3.org/TR/owl-features
[8] D2.2: First Information and Data Models, http://www.fp7-novi.eu/index.php/deliverables
[9] Slice Federation Architecture, v2.0, http://groups.geni.net/geni/attachment/wiki/SliceFedArch
[10] Ontology instance editor - OIntEd, http://novi-im.appspot.com
[11] Openrdf Sesame, http://www.openrdf.org
[12] Resource Description Framework - RDF, http://www.w3.org/RDF/
[13] Alibaba, http://www.openrdf.org/alibaba.jsp

[14] Mosharaf Kabir Chowdhury, N.M., Boutaba, R.: Network Virtualization: State of the Art & Research Challenges. IEEE Communications Magazine 47(7), 20–26 (2009)

[15] Houidi, I., Louati, W., Ameur, W.B., Zeghlache, D.: Virtual network provisioning across multiple substrate networks. Elsevier Computer Networks 55, 1011–1023 (2011)

[16] RSpec, http://www.protogeni.net/trac/protogeni/wiki/RSpec2rac/protogeni/wiki/RSpec

[17] Lymberopoulos, L., Grosso, P., Papagianni, C., Kalogeras, D., Androulidakis, G., van der Ham, J., de Laat, C., Maglaris, V.: Managing Federations of Virtualized Infrastructures: A Semantic-Aware Policy Based Approach. In: Proc. of 3rd IEEE/IFIP International Workshop on Management of the Future Internet, Dublin, Ireland, May 27 (2011)

[18] Ponder2, http://ponder2.net

[19] Hullár, B., Laki, S., Stéger, J., Csabai, I., Vattay, G.: SONoMA: A Service Oriented Network Measurement Architecture. In: Korakis, T., Li, H., Tran-Gia, P., Park, H.-S. (eds.) TridentCom 2011. LNICST, vol. 90, pp. 27–42. Springer, Heidelberg (2012)

[20] https://wiki.man.poznan.pl/perfsonar-mdm/index.php/Hades_MA

[21] Santos, T., Henke, C., Schmoll, C., Zseby, T.: Multi-hop packet tracking for experimental facilities. In: SIGCOMM 2010, New Delhi, India, August 30-September 3 (2010)

[22] VINI, http://www.vini-veritas.net

[23] Farinacci, D., Li, T., Hanks, S., Meyer, D., Traina, P.: RFC 2784, Generic Routing Encapsulation (GRE) (March 2000)

[24] Open vSwitch, http://openvswitch.org/

[25] OpenFlow, http://www.openflow.org

[26] Chappell, D.: Enterprise Service Bus. O'Reilly (June 2004) ISBN 0-596-00675-6

[27] D4.2: Use Cases, http://www.fp7-novi.eu/index.php/deliverables

[28] Jython, http://www.jython.org

[29] JRuby, http://jruby.org

[30] Java Native Interface, http://java.sun.com/docs/books/jni/

[31] Juniper MX480, http://www.juniper.net/us/en/local/pdf/brochures/1500027-en.pdf

[32] VMware ESXi, http://www.vmware.com/files/pdf/ESXi_architecture.pdf

[33] MyPLC, http://www.planet-lab.org/doc/myplc

[34] PlanetLab federation, http://www.planet-lab.org/federation

[35] D4.1: Software Architecture Strategy and Developers' Guide, http://www.fp7-novi.eu/index.php/deliverables