Challenges of an Information Model for Federating Virtualized Infrastructures

Jeroen van der Ham*, Chrysa Papagianni‡, József Stéger†, Péter Mátray†,
Yiannos Kryftis‡, Paola Grosso* , Leonidas Lymberopoulos‡,
*System and Network Engineering research group, University of Amsterdam, The Netherlands
Email: vdham@uva.nl, p.grosso@uva.nl
†Department of Physics of Complex Systems, Eötvös Loránd University, Hungary
Email: matray@complex.elte.hu, steger@complex.elte.hu
‡National Technical University of Athens, Iroon Polytechniou 9, Athens, 15780
Email: chrisap@noc.ntua.gr, leonidas@netmode.ntua.gr, ykryftis@netmode.ntua.gr

***Abstract*—Users of the Future Internet will expect seamless and secure access to virtual resources distributed across multiple domains. These federated platforms are the core of the Future Internet. It is clear that information models, and concrete implementation in data models, are necessary prerequisites for all federative operations, information exchanges, and service support.**

This position paper describes our approach to the development of an information model for federating virtual infrastructures. Our basic assumption is that semantic resource descriptions with context-awareness, in the form of Semantic Web descriptions, better support services in federated platforms. We use our experiences in the development of two ontologies for computer networks and for network monitoring, NDL and MOMENT, to support and guide our development. The requirements of our envisaged Information and Data models are driven from a concrete use-case, using PlanetLab and FEDERICA as examples of virtualized platforms in a Future Internet federated environment.

I. INTRODUCTION

There are currently different kinds of virtual infrastructures available for scientists to perform distributed experiments. A federated architecture that allows users to pick and choose from these different platforms would be very desirable, and it can offer richer and enhanced services. Crowcroft *et al.*[1] have identified horizontal and vertical aspects to be considered when constructing such federations of virtual infrastructures. The horizontal aspects mainly address scalability issues by peering among similar infrastructures. The vertical federation aspects concern the combination of infrastructures offering services at different layers. In this case users can for example combine network resources from one platform with compute resources in another allowing them to access the lower network layers. Vertical integration is still a challenge as a Future Internet combined offering.

The EU Project NOVI, Networking innovations Over Virtualized Infrastructures, aims to create a blueprint of Future Internet (FI) federated infrastructures. An important aspect of this blueprint is the development of an information and the data models that support the federation of FI platforms. In particular such information and data models should capture the concepts and semantics of resources and services offered by the virtualization platforms.

In NOVI the initial validation will use two platforms, PlanetLab and FEDERICA, described in more detail in section II. The NOVI models will nonetheless be generic and usable by other infrastructures that could, at a later stage, be included in the NOVI federation. We describe the envisioned role of the Information Model in the planned NOVI architecture (section III). In section IV we describe a concrete use-case to highlight the role of the information model, together with the various services and functionalities it will support. We then give an overview in section V of existing information models that can provide a starting point for the definition of the NOVI information model. We conclude with an overview of the planned work toward the definition and implementation of the NOVI information model (sec. VI).

II. NOVI PLATFORMS

The NOVI project aims to define a generic architecture for federation for any kind of infrastructure. In the initial prototype we limit ourselves to the FEDERICA and PlanetLab infrastructures.

A. PlanetLab

PlanetLab[2] is a global research platform supporting deployment and testing of novel distributed protocols and applications. It comprises more than a thousand of nodes distributed worldwide, across more than 500 sites.

Each site provides at least two nodes. Every PlanetLab node runs a Linux-based operating system that provides a slice abstraction. A *slice* is network-wide resource container that hosts experiments[3]. Slices usually span several PlanetLab nodes. The part of a node that is available to the slice as a virtualized resource is called a *sliver*. Slices have a finite lifetime and must be periodically renewed to remain valid. A slice is composed of Linux Vserver-based Virtual Machines.

The current PlanetLab control framework relies on the PlanetLab Central software (PLC) and the Slice-Based Facility Architecture[4], [5]. The PlanetLab control framework defines a clearinghouse as well as aggregate and slice manager functions. PlanetLab Central allows researchers to create and control their slices, and test-bed operators to remotely manage the nodes, in a client-server fashion via XML-RPC and HTTPS. PlanetLab uses a resource specification (RSpec) to model its resources. RSpec files are written in XML, describing slice resources, in terms of the hardware characteristics, networking facilities, constraints and dependencies on the allocation of those resources.

B. FEDERICA

FEDERICA[6] is a network virtualization platform, augmented with network and computing facilities hosted in European national resource and education network (NREN) points of presence. Resources of the FEDERICA infrastructure can be allocated to multiple independent slices and assigned to different network researchers.

A slice in FEDERICA is a set of computing and network virtual resources created by means of virtualization (VMs, Logical Routers, VLANs etc). Users have full control over the allocated virtual resources in their slice as well as access to related monitoring information.

During the first stage of the FEDERICA project, slices were manually created and managed. However the automation of slice creation/allocation, easy slice maintenance and user registration have also been addressed within the course of the project with introduction of the FEDERICA Slice Tool[7] and the User Portal. The FEDERICA User Portal allows users to create new slice requests and the FEDERICA Slice Tool acts as the management plane.

III. Role of the NOVI information model

As outlined above, information models are a necessary component for managing federations of heterogeneous virtualized infrastructures; they cater for the necessary abstraction of the managed entities within a federation of heterogeneous virtualized platforms. To accomplish both horizontal and vertical federations, we believe that information models for federated Future Internet platforms should support:

- *virtualization concepts*, which cater for virtualized resources that are key element in FI platforms;
- *vendor independence*, as virtualized infrastructures employ hardware and software from different manufacturers;
- *monitoring and measurement concepts*, with a uniform description for both the monitored entities and the measurement units within a federation;
- *management policies*, which can be used to define the behaviour governing the managed environment. It is desirable to provide support both for authorization policies, i.e. policies that define who is able to access which resource, and event-condition-action policies;

Furthermore we are convinced that such an information model must include:

- *semantics* to support context-aware decisions. Semantic Web can provide the basis for this;
- *description of stateful resources*. The capability to capture the state of the managed environment enables rollback to previous states and evaluation of the current state in respect to the desired state so as to take on appropriate actions.

We set out to build the NOVI information model by including all of these features.

The envisioned NOVI information model will ultimately support the control and management of the individual federated platforms, and the communication between them. In the NOVI federation architecture control and management rely on a number of components: the *Scheduler*, the *Intelligent Resource Mapping*, the *Monitoring Service* , the *Policy Management Service*, the *Service Discovery* and the *Database* service.

Figure 1 represents schematically these components.

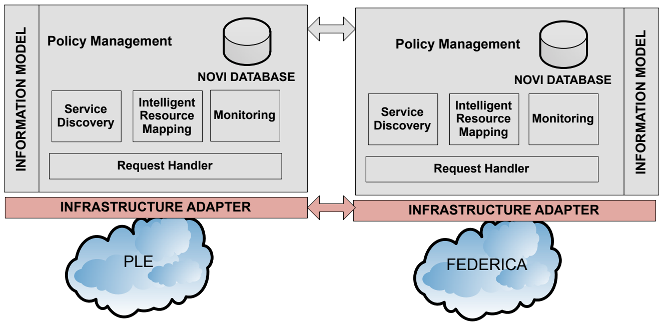

Figure 1. Preliminary concepts for the NOVI federation

The NOVI information model will contain a number of ontologies; e.g. we expect to need:

- a *resource mapping* ontology, which provides a description of the resource in the NOVI platforms. We will base this on the NDL models we will discuss later in section V;
- a *service* ontology, which provides the definition of the federated services;
- a *monitoring and measurement* ontology, which provides the definition of the concepts needed to perform monitoring in the federation. We will base this on the MOMENT models, also discussed later.

We can summarize and generalize the requirements on the information model with respect to each of the services as follows:

- The *Intelligent Resource Mapping Service* supports the embedding of user requests within a physical substrate. The IM must define all the elements and concepts that the Intelligent Resource Mapping Service uses to allocate, schedule and migrate resources in heterogeneous environments.
- The *Monitoring Service* enables NOVI users and administrators to retrieve information about the temporal behavior or the status evolution of specific resources. The IM needs to define proper descriptions and representations of monitored data, as different methods monitoring the same cut of concepts need to be standardized.
- The *Policy Management Service* provides the functionality of a policy-based management system, where policies

are used to define the behavior governing the managed environment. The IM must allow for the definition of policies at an abstract level, in such a way that as new platforms join the federation, there is no need for changes at the policy level.

- The *Virtualized Resource Discovery Service* selects resources capable of providing a specific service and is able to find resources efficiently based on their context. The IM provides descriptions of the state of the resources in relation to both their environment, and other internal parameters, i.e. the external and internal context. This is essential for selecting the resources in the most efficient way.
- The *Database Service* holds information on slices and slivers of the managed virtualized infrastructure. The database is used by all other services to acquire information about resources and associated policies.
- Furthermore the IM must support the *Request Handler Service*, which handles the requests coming from users or platforms.

IV. Requesting A Federated Slice: A NOVI Use-Case

A use-case can better illustrate how we envision the NOVI architecture and the information model. Below we describe a use-case in the form of a user request for a monitorable slice across federated platforms #1 and #2. To make our example more concrete we may assume that platform#1 is PlanetLab and platform#2 is FEDERICA, but the functionalities and behaviors we sketch are generic and applicable to any other Future Internet platform that is part of a NOVI federation.

1) John, who is a registered platform#1 user, requests a slice for a virtual topology. Borrowing the terminology used by the GENI initiative, user requests can be *bound* when virtual resources are mapped into physical ones by the user and *unbound* when there are unmapped links or nodes. John submits an unbound request at the NOVI service layer of platform#1. The requested topology supports custom routing protocols/functionality provided by ten fully meshed logical routers R1–R10. The traffic source and sink for the experiment are defined as two virtual servers, {S1: 2 CPU cores, 2048MB RAM, 10GB disk storage} and {S2: 2 CPU cores, 1024MB RAM, 50GB disk storage} linked to R1, R10 respectively. Morevoer John makes an additional request related to the quality of service, specifying a particular upper threshold regarding one way delay for the access links {S1-R1} and {S2-R10}.
2) NOVI's *Virtualized Resource Discovery Service* is responsible for finding the set of available susbtrate resources from the two platforms - #1 and #2. These resources must comply with the requirements set by John's request, with regards to functional characteristics of the requested resources e.g. node type.
3) Selection among the available susbtrate resources is optimized via the *Intelligent Resource Mapping Service* at platform#1. Appropriate mapping optimization algorithms are used in order to select substrate resources enforcing additional virtual topology constraints such as the one-way delay constraint for links {S1-R1} and {S2-R10}. According to the algorithm results, the S1, S2 virtual servers are mapped to the S1',S2' platform#1 substrate servers and the R1-R10 logical routers are mapped to the R1'–R10' platform#2 subrate routers. Substrate routers R2'and R3' are connected via switch SW1'.
4) The solution of the embedding algorithm now has all of the necessary details, which are needed to create the slice and allocate resources to it. The following two procedures are executed in parallel:
   a) NOVI's service layer at the platorm#1 side requests virtual server instantiation at S1' and S2' through platform's #1 Infrastructure Adapter. Additional monitoring components are installed in these substrate nodes.
   b) NOVI's service layer at the platorm#2 side requests logical router instantiation at R1'–R10' and the usage of SW1' as a transit node through platform#2's infrastructure adapter. Additional monitoring components are installed in R1' and R10'.
5) John is now ready to start experimenting with a combined platform#1 platorm#2 slice. John is able to monitor the combined NOVI slice. Monitoring serves the following purposes: John can follow the real-time progress and performance of his experiment. At the same time NOVI service layer is able to catch signals if something went wrong with the originally requested topology.
6) Both resources and the monitoring capabilities expire after a certain amount of time. John may update his slice to prolong reservation or he can decide to delete his slice before expiration. Otherwise, monitoring operations die out and allocated resources are released in a robust manner.

A. Interactions of the NOVI Components

Based on the above use-case we now elaborate on the way the different NOVI service components would interact at two stages during the lifecycle of a slice; during the slice request and slice creation phase and secondly in an event-condition-action situation occurring at runtime. In the NOVI innovation cloud, information passing between service components is represented according to the NOVI Information Model.

When a slice request arrives to the NOVI service layer via the *NOVI request handler* (step 1), the *Intelligent Resource Mapping Service (IRMap)* after having authorized the user's request by the *Policy Manager Service (PolMan)*, triggers the distributed resource discovery operation via the *Virtualized Resource Discovery Service (RD)* across all federated platforms (step 2). In John's case the *RD* service looks into specific attributes of the substrate resources, such as availability of CPU, RAM and storage at the substrate computing nodes and one-way delay for each of these servers link to an available substrate router. For that reason it invokes the *Monitoring*

Service (MonSrv) component to respond with the most up-to-date available resource characteristics to narrow the resource set to fulfill the requirements of the slice request. At this stage *IRMap* has all the necessary information to properly choose from the substate resources (step 3).

Next the *IRMap* embeds the virtual network request to the available substrate network resources and triggers in a distributed fashion resource reservation via the appropriate *Infrastructure Adapters (IAs)* (step 4). *IAs* are also responsible for uploading any additional piece of software that is needed for monitoring. Upon success it creates the slice model (resource mapping ontology), which is registered to the *Database*. The next task of *IRMap* is to inform the *MonSrv* to start the monitoring of a list of properties and whether to signal certain monitoring events back to the *PolMan* (step 5). Finally the *PolMan* informs the user that the slice is ready to be used (step 6).

If the *MonSrv* indicates that a slice component fails or an event described by the user sets in (the delay exceeds a given threshold in the former use-case) the *PolMan* informs the user and applies the appropriate action (step 7), it will execute the update slice rule in this scenario. The slice is repaired through appropriate re-mapping of substrate resources.

V. Related Work

We envisage a federated environment where a large number of heterogeneous resources, belonging to different Future Internet (FI) platforms, need to be managed in such a way, so that their offered virtualized resources and services are shared amongst multiple users of the federation. To facilitate the management of FI resources and services within a multi-domain distributed environment, we require employing an Information Model -and its related Data Models- to capture the concepts and semantics of the resources and services offered by different virtualization platforms. In the remainder of this section, we provide an overview of four Information Models, supporting federation and/or virtualization concepts within the context of a federation of FI platforms.

The reported models are: The Common Information Model (CIM), Directory Enabled Networks - next generation (DEN-ng version 7), the Network Description Language (NDL) and the Monitoring and Measurement ontology (MOMENT). We will conclude this section with future work-plan directions, taking into account the pros and cons of the reported models.

A. Common Information Model

The DMTF has established the System Virtualization, Partitioning, and Clustering Working Group (SVPC WG)[8] to work on modeling of virtualized environments. This work is part of DMTF's Virtualization Management (VMAN)[9] initiative and aims to help vendors develop more robust, interoperable solutions for deploying and managing virtual appliances and environments. At a technical level, DMTF has described in DSP2013[10] how CIM can be used for system virtualization, including the schema additions for the general resource allocation pattern and the modeling of virtual and host computer systems. The elements described in [10] enable management of system virtualization environments including management of virtual computer systems and their associated virtual resources. According to DSP2013, the resources that make up the virtualization environment typically are supplied by one or more host computer systems. A virtualization layer (usually firmware or software, but possibly hardware) manages the lifecycle of a virtual computer system, which is composed of resources allocated to or assigned from the host computer system. A virtual computer system may be active and running an operating system and applications with a full complement of virtual devices defined and allocated, or it may be inactive with no software running and a subset of the virtual devices actually allocated.

The system virtualization model enables the client to manage the virtualization layer and the full lifecycle of the hosted virtual computer systems. Both host and virtual computer systems (also known in the industry as virtual machines) are represented similarly by instances of the *CIM_ComputerSystem* class. Computer system devices are modeled through instances of subclasses of the *CIM_LogicalDevice* class. The relationship between system and devices is modeled through the *CIM_SystemDevice* association. The relationship of virtual computer systems to their host system is modeled through the *CIM_HostedDependency* association.

B. Directory Enabled Networks - Next Generation

DEN-ng[11] is an object-oriented information model that describes the business, system, implementation and deployment aspects of managed entities as well as their relationships. It was originally designed to meet the needs of the TeleManagement Forum New Generation Operations Systems and Software (TMF NGOSS) architecture[12]. DEN-ng was created following the policy based management concept[13] employing policies to govern the behavior of the managed environment in a domain independent way.

To maintain compatibility with the existing DEN-ng model, the work reported in [14] updated the DEN-ng model with a model for virtual resources. The hierarchy of *Resource* is changed, including the *VirtualResource* and *NonVirtualResource* subclasses. The definition of *Resource* was updated to employ the former two subclasses. Its new definition in version 7 is ([14]): "A Resource is any virtual or non-virtual component or system. A Resource contains a set of physical and/or virtual entities that are of interest to the managed environment. A Resource may represent a limited or critical entity that is required by other entities."

Furthermore, the original DEN-ng definition of *Resource* is also used in DEN-ng v7 as the new definition of *NonVirtualResource*, whereas the original *Resource* hierarchies are moved under *NonVirtualResource*. Last, DEN-ng v7 includes a new definition for *VirtualResource*. The latter is defined as[14] "A VirtualResource is an abstraction that decouples the physical manifestation of a Resource from its logical operation". More information on DEN-ng version 7 can be found in [14].

C. Network Description Language

The Network Description Language (NDL)[15] is an information model developed by researchers at the University of Amsterdam. NDL comprises of a series of schemas that categorize information for network topologies, network technology layers, network device configurations, capabilities, and network topology aggregations. The main use-cases are supporting management of hybrid networks, generation of network maps, offline and multi-layer path finding, and network topology information exchange. NDL has been used primarily in the research community in the Netherlands: UvA, SARA and SURFnet. It also has been applied to the GLIF Optical Lightpath Exchanges[16]. NDL uses RDF (Resource Description Framework[17]) as syntax and the grounding of the model in the Semantic Web framework. NDL is a modular set of schemata, defining an ontology to describe computer networks. Its main components are

1) The topology schema, which describes devices, interfaces and connections between them on a single layer. The classes and properties in the topology schema describe the topology of a hybrid network, without detailed information on the technical aspects of the connections and their operating layer.
2) The layer schema, which describes generic properties of network technologies, and the relation between network layers. The NDL layer schema allows applications to describe multi-layer networks, like hybrid networks.
3) The capability schema, describing device capabilities.
4) The domain schema, which describes administrative domains, services within a domain, and how to give an aggregated view of the network in a domain.
5) The physical schema, which describes the physical aspects of network elements.

NDL has also been succesfully applied and extended in the GENI project[18] to support management and slice provisioning in the BEN network.

Currently, NDL does not support virtualization concepts and monitoring/measurement data, however, its support for describing network topologies and device configurations could be used as a building block for the Information Model for a federation of FI platforms.

D. The Monitoring and Measurement Ontology

The main objective of the MOMENT project was to design and develop a Mediator architecture offering a unified interface for measurement services, able to use all data and functionalities from the existing measurement infrastructures[19].

The project's principal innovation was to create a measurement-specific ontology, which allows semantic representation and retrieval of measurement and monitoring information, as well as providing the flexibility of a service oriented architecture for future Internet applications. Based on the data scheme of various measurement facilities, they built up a Network data ontology that represents the semantic content of network measurement data.

The MOMENT Mediator serves as a common platform to federate network measurement data provided by various sources, like measurement infrastructures or measurement tools. One of the key problems tackled was the heterogeneity of these sources: data are represented in infrastructure-specific ways with no standardization of data types, units and structures. The MOMENT ontology is the key to generate mappings between the different information models to enable the successful interoperability of these originally independent systems.

The MOMENT ontology builds on existing ontologies (e.g. OGF-NMWG[20]). Furthermore, MOMENT project describes the approach used to construct a taxonomy of possible anonymization approaches, within the context of Internet measurement related data collections' query and retrieval.

E. Summary

We can see in the above overview that none of the aforementioned Information Models fulfils all the requirements of an Information Model catering for a federation of FI platforms. We will thus further investigate:

- Which of the overviewed models is able to provide us with an advanced information modeling framework that will enable us to capture all the necessary abstractions for federating a diverse number of heterogeneous virtualized infrastructures.
- What are the extensions in existing efforts that need to be designed and implemented in order to support a federation of heterogeneous virtualized infrastructures.

We believe that a key requirement within a Future Internet environment is to support context-awareness for managing federations of FI virtualized infrastructures. The NOVI Information Model should support mapping to and between the models of the different platforms. NDL and MOMENT are considered as good candidates supporting this key requirement, as they are based on Semantic Web languages and support the semantics needed for both high-level and more detailed descriptions required for richer decisions.

VI. Summary and Future Work

The NOVI information model and the associated data models are a fundamental building block to support control and management of virtualization platforms participating in the NOVI federation.

The research directions that we will explore in depth in the coming time are related to the three main components of the NOVI information model we listed in section IV:

1) the definition of the semantics needed to handle heterogenous resources. This will become the resource mapping ontology;
2) the definition of the resource selection algorithms which use the semantics in the service ontologies;
3) the definition of semantics for NOVI's measurement metrics and units, including attributes such as accuracy of measured data in heterogeneous infrastructures.

The main challenge for us lies in the need to support a multiplicity of services within the NOVI federation. Therefore, we need semantics in the information model on multiple levels. For example, in supporting the Request Handler Service we must account for different levels of complexity in requests:

- *Named Request*: The user can request specifically identified resources, where the mapping between the virtual and physical has already been done by the user. In this case the Discovery Service is only responsible for checking the state of the requested resources, and their availability.
- *Mixed Request*: The user can request a virtual topology, where NOVI discovers available resources and provides the instantiation of the topology. This is the case in the use-case we have highlighted above. QoS constraints can also be expressed in this type of request. For example, in order to deliver a specific network or computing performance, the NOVI services will transform this to indicate the need for a certain number of resources.
- *Service Request*: The user can request a service, which must be mapped to a topology by the NOVI Discovery Service, before the Intelligent Resource Mapping Service can instantiate resources. This requires more complex decisions, involving identifying the resources capable of supporting the requested type of service.

All of these components need to be captured in the description we provide in the service ontology.

Finding solutions to these challenges, in the form of semantically rich information models, will provide a novel and original way to support federation and Future Internet services. This new approach fully based on context-awareness makes it possible to deliver complex and rich services.

Acknowledgments

This work was partially supported by the European Commission, 7^{th} Framework Programme for Research and Technological Development, Capacities, Grant No. 257867 – NOVI.

References

[1] J. Crowcroft, P. Demeester, J. Magen, P. Tran-Gia, and J. Wilander, "Towards a collaboration and highlevel federation structure for the fire facility," in *Working Group on modular federation of FIRE Facilities*, 2009.

[2] "Planetlab," http://www.planet-lab.org.

[3] "Understanding and resolving conflicts on planetlab," http://www.cs.princeton.edu/~llp/policy.pdf.

[4] "Sfa draft 1.0," https://svn.planet-lab.org/browser/sfa/trunk/docs/sfa.pdf.

[5] "Sfa draft 2.0," http://groups.geni.net/geni/wiki/SliceFedArch.

[6] "Federated e-infrastructure dedicated to european researchers innovating in computing network architectures," http://www.fp7-FEDERICA.eu.

[7] S. F. A. Berna, E. Grasa, "Extending manticore to manage ip and virtual machine slices in the federica project," in *TNC*, 2009.

[8] "Distributed Management Task Force," http://www.dmtf.org/about/working-groups.

[9] DMTF's virtualization management (VMAN) standard. [Online]. Available: http://dmtf.org/standards/vman/

[10] "DMTF CIM System Virtualization Model White Paper," http://www.dmtf.org/sites/default/files/standards/documents/DSP2013_1.0.0.pdf.

[11] Directory Enabled Networks – Next Generation. [Online]. Available: http://autonomic-management.org/denng/index.php

[12] J. C. Strassner, *Policy Based Network Management*. USA: Morgan Kaufman Publishers, 2004.

[13] M. Sloman, "Policy driven management for distributed systems," *Journal of Network and Systems Management*, vol. 2, pp. 333–360, 1994.

[14] Y. Choi, J. Li, Y. Han, J. Strassner, and J. W.-K. Hong, "Towards a context-aware information model for provisioning and managing virtual resources and services," in *Proceedings of the 5th IEEE international conference on Modelling autonomic communication environments (MACE'10)*, 2010, pp. 100–112.

[15] J. van der Ham, P. Grosso, R. van der Pol, A. Toonk, and C. de Laat, "Using the network description language in optical networks," in *Tenth IFIP/IEEE Symposium on Integrated Network Management*, May 2007. [Online]. Available: http://staff.science.uva.nl/~vdham/research/publications/0606-UsingNDLInOpticalNetworks.pdf

[16] Global Lambda Integrated Facility (GLIF). [Online]. Available: http://www.glif.is/

[17] G. Klyne and J. J. Carroll, "Resource escription Framework (RDF): Concepts and abstract syntax," World Wide Web Consortium (W3C), W3C Recommendation, 2004. [Online]. Available: http://www.w3.org/TR/rdf-concepts/

[18] I. Baldine, Y. Xin, D. Evans, C. Heerman, J. Chase, V. Marupadi, and A. Yumerefendi, "The missing link: Putting the network in networked cloud computing," in *International Conference on the Virtual Computing Initiative (ICVCI 2009)*, Oct. 2009. [Online]. Available: http://www.cs.duke.edu/nicl/pub/papers/missing-link.pdf

[19] *MOMENT Project Deliverable, D3.2 Final specification of the unified interface*.

[20] "MWG public subversion repository," http://anonsvn.internet2.edu/svn/nmwg/.