FIRMA: A Future Internet Resource Management Architecture

Alexander Willner, Technische Universität Berlin, alexander.willner@tu-berlin.de, Germany
Thomas Magedanz, Fraunhofer FOKUS, thomas.magedanz@fokus.fraunhofer.de, Germany

Abstract

The Internet is broken and there are several approaches to fix it. In order to validate these novel ideas, they are often evaluated using large-scale experiments involving numerous heterogeneous resources. As a result, several Future Internet testbeds have been established and different competitive mechanisms to federate them are subject of current research. Even though the basic principles of these mechanisms are mostly equivalent, the current situation leads to a fragmented and complex environment for experimenters, testbed owners, and developers. Therefore, an extensible Future Internet Resource Management Architecture (FIRMA) with a protocol agnostic core is proposed, that offers interchangeable federation and experimentation interfaces concurrently, supports heterogeneous resources, and integrates existing testbed services. Both, the requirements for such an architecture and the applicability of an according implementation were evaluated within latest European Future Internet research projects.

1 Introduction

The Internet is broken[1] and there are several approaches to fix it. Basically, two strategies can be identified. On the one hand, evolutionary strategies try to incrementally improve the current Internet architecture while conserving the compatibility with existing mechanisms. On the other hand, clean-slate approaches tend to break the compatibility to the established Internet and try to fundamentally change the way devices communicate.

In both cases new developments need to be evaluated against specific requirements. Given the nature of the Internet, scalability and support for a wide range of heterogeneous resources are of particular interest. Briefly, the techniques that may be used for performance evaluation are analytical modeling, simulation, and measurement of real environments[2]. While former methodologies are suitable for systems with low to middle complexity, Future Internet approaches require large test domains with a diverse set of equipment. Therefore, various Future Internet experimental facilities have been established. These facilities enable experimenters to create their own environments, which can be used to evaluate the according system under test.

In this context, it is of peculiar concern to extend the potential list of offered resources for large-scale experiments. As a result, current research discusses architectures to federate different testbeds with each other in a standardized manner. This involves all areas of an experiment lifecycle, such as federated authentication and authorization, resource description, discovery, reservation, orchestration, provisioning, monitoring, and release as well as experiment control and measurement.

In the last years a variety of frameworks, protocols, and architectures have been developed for these purposes in the Future Internet Research and Experimentation[3] (FIRE) and Global Environment for Network Innovations[4] (GENI) initiatives. Currently, particular attention is being paid to the Slice-based Federation Architecture[5] (SFA) for resource provisioning, the cOntrol and Management Framework[6] (OMF) with its Federated Resource Control Protocol (FRCP) for experiment control, the ORBIT Measurement Library[7] (OML) for experiment measurement and the Teagle[8] framework components for generic resource federation. However, these mechanisms have been mainly designed independently from and therefore are not closely linked with each other. As a consequence, latest FIRE research focuses on unifying these technologies using existing developments.

Notwithstanding the above, the federation of facilities is also subject in similar contexts. In particular within the Future Internet Public Private Partnership[9] (FI-PPP) the federation of nodes is also envisaged based on Future Internet Core Platform[10] (FI-WARE) components. And the IEEE group Standard for Intercloud Interoperability and Federation[11] (P2302) is focusing on technologies to interconnect cloud environments.

Based on experiences gained from the afore-mentioned contexts and a conducted survey[12], several assumptions are being made. First, experimentation and federation mechanisms that are already in place, are unlikely to be replaced in the medium term. Second, a majority of the functionality of these technologies are in principle based on the same enabling mechanisms. Third, everything, also a service or a testbed itself, is a resource and can be federated. Fourth, existing testbed features (such as user databases or billing mechanisms) must be incorporated. Fifth, the concept of resource federation will be adopted by more fields of application in the future.

Based on these assumptions, in this paper we propose an extensible architecture to manage resources within federated Future Internet environments. The main contribution is the definition of a notably reusable and extensible architecture for federated Future Internet resource provisioning, monitoring, and control. The long-term goal is to evaluate the architecture within all areas of the Future Internet experimentation life-cycle by developing a reference implementation, integrate the developments into a commercial Enhanced Telecom Operations Map[13] (eTOM) based context, to adopt it to the area of InterCloud federations, and finally to generic federated service life-cycles based on negotiated Service Level Agreements (SLAs). The proposed architecture will allow developers to efficiently implement and link existing and future protocols using already existing common denominators (e.g. for authorization handovers between protocols); researchers have a mechanism to compare Future Internet experimentation protocol characteristics against each other, students can learn details about different protocols using a single framework, and testbed providers have the ability to provide resources using several protocols at once.

The remainder of the paper is structured as follows. In Sec. 2 relevant work by others and the most important stances in the literature are presented. Based on this knowledge in Sec. 3 the treated research gap is shortly summarized and the proposed architecture is presented in detail. Finally, we close giving some conclusions and outline future work in Sec. 4.

2 Related Work

Within the Global Environment for Network Innovations[4] (GENI) different competing testbed control frameworks are under development. A comparison between these control frameworks has been published earlier [14]: the Open Resource Control Architecture[15] (ORCA) framework Shirako; PlanetLab Central (PLC) that is used for the PlanetLab[16] infrastructure; the DETER Federation Architecture[17] (DFA) used in the Trial Integration Environment Based on DETER[17] (TIED); the cOntrol and Management Framework[6] (OMF) developed within the Open Access Research Testbed for Next-Generation Wireless Networks[18] (ORBIT); as well as ProtoGENI[19] applied in Emulab[20]. Therefore, the need to federate several testbeds that are controlled by the different competing control frameworks has been identified already 2010[21]. As a result, the Slice-based Federation Architecture[5] (SFA) was defined and is still under revision to allow federation across facilities. SFA aligned testbeds can be controlled by various SFA compliant user tools such as the SFA Command-Line Interface (SFI), MySlice, and Omni. Resources are described using so called Resource Specifications (RSpecs). These are arbitrary XML documents and the formalization of resource descriptions are still subject of current research.

While SFA mainly addresses issues regarding the description, discovery and provisioning of resources, mechanisms are also needed to describe experiments and to control and monitor resources accordingly. For these purposes the OMF has been adopted. It takes an experiment description file, written in the OMF Experiment Description Language (OEDL), as an input to orchestrate experiments on OMF enabled testbeds. The underlying architecture is currently transitioning into a federation-enabled version initially called OMF-Federated[22] (OMF-F) that uses the Federated Resource Control Protocol (FRCP).

Furthermore, for experiment measurements the OML Measurement Stream Protocol (OMSP) is generally being used to collect and push monitoring information. An implementation is the ORBIT Measurement Library[7] (OML), which enables experimenters to instrument their application by defining measurement points inside their application source code. These data are transported as streams from the measurement points with the help of the OML client libraries and stored into an OML server.

Within the Future Internet Research and Experimentation[3] (FIRE) context the need to conduct experiments over federated testbeds has been identified as well. Starting with the Panlab II[23] project, possible heterogeneous resource federation scenarios in large scale experimental facilities had been analyzed[24]. Based on this, the Teagle federation architecture was proposed which contains several components: The involved testbeds configure a Resource Adapter[25] (RA) for each of their heterogeneous resources and exposes them using the Directory Enabled Networks New Generation[26] (DEN-ng) data model by running a Panlab Testbed Manager[25] (PTM); the experimenters create their own Virtual Customer Testbeds (VCTs) using the Virtual Customer Testbed Tool[27] (VCTTool) and based on this they can describe and control experiments using the Federation Computing Interface[28] (FCI).

Subsequently, the Federation for FIRE[12] (Fed4FIRE) project has been established. It currently gathers the requirements within the FIRE community and defines an architecture for the heterogeneous federation of Future Internet experimentation facilities on a larger scale. This includes the compatibility between FIRE and GENI facilities. It has been identified, that federated testbeds must support all the functions of the experiment life-cycle (cf. **Figure** 1): resource description, resource discovery, resource requirements, resource reservation, resource provisioning (direct or orchestrated), experiment control, facility monitoring, infrastructure monitoring, experiment measuring, permanent storage, and resource release; as well as federated identity management, authorization, and SLA management. The survey has revealed, that currently SFA has widely been adopted for resource discovery and provisioning. Other commonalities are the use of Zabbix, Nagios, Zenoss, Ganglia, and OML for monitoring and OMF/FRCP, VCTTool/FCI and the Network Experimentation Programming Interface[29] (NEPI) for experiment control. However, no dominating technologies could be distinguished for any of these categories. Furthermore, different possible federation architectures have been evaluated. Based on identified characteristics, the heterogeneous federation approach was recognized to be the most suitable one for the Future Internet experimentation facilities under evaluation. In this architecture all testbeds run their native testbed management software.

At the present time, also the recently established Future Internet Public Private Partnership[9] (FI-PPP) project Experimental Infrastructures for the Future Internet (XIFI) is in its requirement engineering phase. Within XIFI a

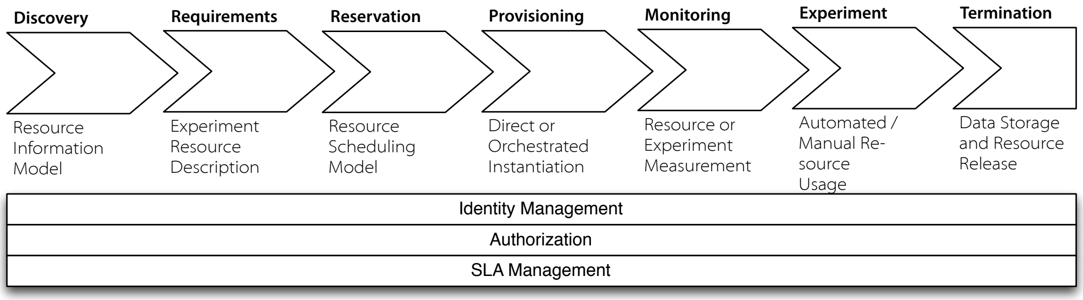

Figure 1: The Experiment Life-cycle based on [12]

sustainable pan-European federation of Future Internet test infrastructures will be established. It becomes apparent, that on the one side this federation will technically be based on Generic Enablers (GEs) developed by the FI-PPP project Future Internet Core Platform[10] (FI-WARE). On the other side, interoperability with the FIRE community and testbeds is aspired. Therefore, the federation mechanism, that is going to be developed within the XIFI project must take both approaches into account.

Within the European Institute of Innovation and Technology Information and Communication Technology Labs (EIT ICT Labs) context federation is also perceived as a catalyst to increase the utility of a testbed. In particular within the Fanning out Testbeds-as-a-Service for the EIT ICT (FanTaaStIC) project new ways of creating a sustainable business model, including an eTOM based operational concept for the implementation, are getting explored.

Analog to these efforts, current research also strives to federate resources within single administrative domains in another context[30]: In 2009 a “Blueprint for the InterCloud”[31] was published to define protocols and formats for Cloud Computing interoperability. A taxonomy and survey of already existing InterCloud architectures was recently published[32] and different Standards Developing Organizations (SDOs) are working on according documents. One example is the IEEE project Standard for Intercloud Interoperability and Federation[11] (P2302).

3 Own Approach

We have shown that several approaches to federate and control resources across multiple administrative domains already exist. It becomes explicit that on the one hand, a single mechanism can’t be used within every context. On the other hand, most of them share many similar concepts and information about the very same resources. Furthermore, within the Future Internet research area, multiple requirements arise to which either no solution is specified yet or different independent tools must interact with each other to fulfill them.

This leads to a wide range of interesting research questions, in particular with reference to the experiment life-cycle. Within the scope of this paper, we’ll restrict the research topic to define an architecture that is extensible enough to support potentially all aspects of the life-cycle.

3.1 Architecture Overview

Related work focuses on loosely combining existing developments in order to cover the whole experiment life-cycle. While this approach reuses already established software components, it also introduces complexity including duplicated functionalities and synchronization overheads. Therefore, the approach is to identify shared functionalities of existing mechanisms in order to define an extensible architecture with a protocol agnostic core. That allows to re-use mechanisms that are common among them, to offer different protocols in parallel, and it should be applicable within a wide research field not limited by anticipated design decisions.

As a main achievement, **Figure** 2 shows the current version of the proposed architecture from a high level point of view and it will be described in further detail within the next sections. In general, it is based on established architectural design patterns such as Entity, Boundary, Interactor[33] (EBI); Data, Context and Interaction[34] (DCI); the Hexagonal Architecture[35]; and the Onion Architecture[36]. They all share a common objective, which is the separation of concerns[37]:

1. Independent of frameworks. The architecture does not depend on any external libraries or frameworks. Frameworks and libraries can be used as tools rather than inseparable part of the architecture.
2. Independent of interfaces. The User Interfaces (UIs) and Application Programmers Interfaces (APIs) can be changed without any affect to the architecture because they are not bound to it.
3. Independent of databases. The persistence technology can be changed without any affect to the architecture because the business rules are not bound to it.
4. Independent of services. The testbed premises and resources can be changed without any affect to the architecture because the business rules are not bound to them.
5. Testable. The business rules can be tested without any external elements following a Test Driven Development[38] (TDD) approach.

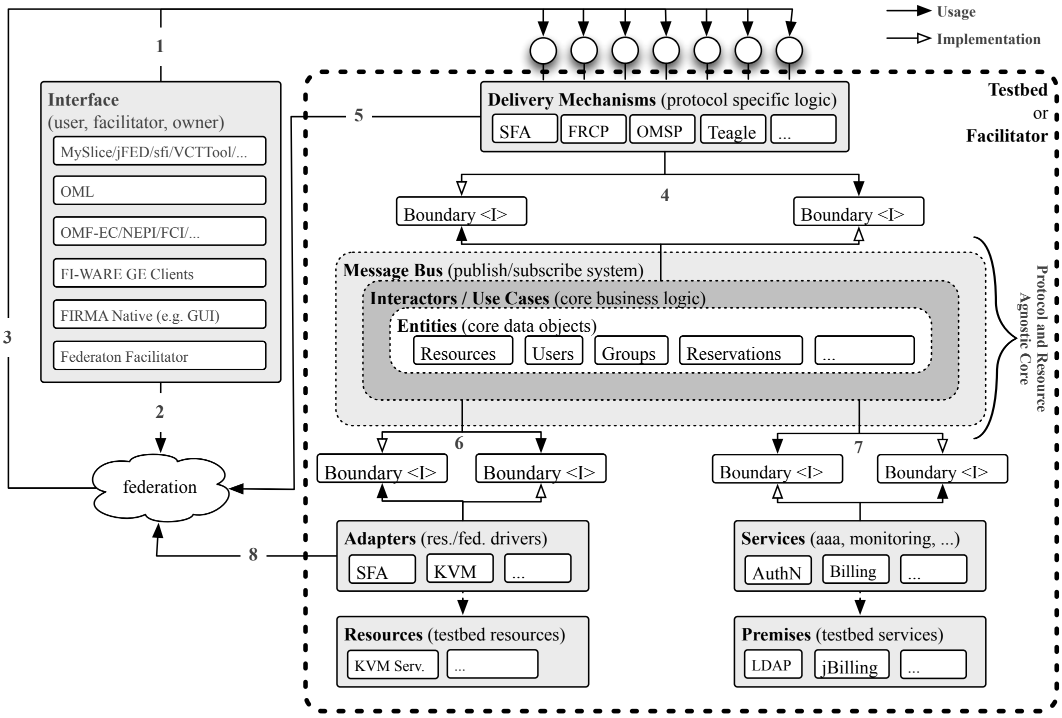

Figure 2: FIRMA Architecture Overview

3.2 Interface

The variety of user tools and developer toolkits are well established in their certain user community, and are under active development. Depending on the experimenters working environment specific tools will be used and it is unlikely that this would change in the midterm. Therefore, the main target is not to offer new user tools but to support these clients and to use them for acceptance testing and to ensure compatibility with other implementations. This includes SFA, FRCP, OML and Teagle compliant systems for Future Internet experimentation as well as the support for Graphical User Interfaces (GUIs) and native clients. Furthermore, Open Cloud Computing Interface (OCCI) compliant clients within the InterCloud and Generic Enabler (GE) interface compliant clients within the FI-PPP / FI-WARE contexts are possible candidates.
Related links: **1** (communication with FIRMA), **2** (communication with other federation partners), and **3** (communication with FIRMA from other clients within the federation).

3.3 Delivery Mechanism

The Delivery Mechanism is the single point of entry and responsible for incoming communication from testbed external components. In particular it offers multiple APIs at once for the different clients and handles the transport layer security. To acquire flexibility also in this components, the specific implementation of each protocol should be composed of four independent modules: **view** (abstracts the used transport protocol from the actual message), **view model** (the representation of the exchanged data), **controller** (protocol dependent business logic for incoming messages), **presenter** (protocol dependent business logic for outgoing messages).
Related links: **4** (bilateral publish/subscribe communication between the core system and the delivery mechanism), **5** (push communication to other federation partners, e.g. for pushing monitoring data).

3.4 Message Bus and Boundaries

To decouple the concrete implementations from each other - in order to make them exchangeable - the different modules communicate only via well specified interfaces within a message queue. Following the Dependency Inversion Principle[39] (DIP) makes the architecture reusable and is also important for the construction of code that is resilient to change. Empty arrows depict the implementation and filled arrows represent the usage of an interface by a module.

3.5 Core

In this module, the core business logic – or use cases – that are needed across several protocols are implemented. It is composed of separated protocol agnostic libraries that cover the management of users, groups, resources, policies, persistence, reservations, schedules or monitoring data. As a result, the Delivery Mechanisms will contain a limited set of business logic only, that is very specific for the according protocol. All other functionalities are implemented in the Interactors. For data exchange, they make use of plain business objects called Entities.
Related links: **6** (bilateral publish/subscribe communication between the core system and the adapters to provide

resources from own or federated testbeds), **7** (bilateral publish/subscribe communication between the core system and services to integrate testbed internal aids).

3.6 Adapters

The common adapter design pattern is used to unify the interfaces for heterogeneous resources. With restrictions it was also extensively used in the PII project and Teagle framework (RAs) and is comparable to device drivers. Each adapter encapsulates one or multiple resource instances of a single resource type by offering a unified interface for resource description, provisioning, control, monitoring, and release. On this level also another testbed or federation of testbeds is handled as a group of available resources and can recursively be abstracted.
Related links: **8** (providing resources from federated testbeds).

3.7 Services

Although the main use cases are implemented within the core, several functionalities can be exchanged by plugable extensions. Examples are persistence, authentication, authorization, accounting, or billing mechanisms. These extensions can again delegate parts of the functionality to testbed internal premises.

4 Conclusions and Future Work

We have given a brief overview of the current experimental Future Internet research landscape, with a focal point on testbed federation frameworks used within the GENI and FIRE initiatives. Based on this, we have identified research gaps with respect to the support of the whole experiment life-cycle by existing solutions. As the main contribution, the Future Internet Resource Management Architecture (FIRMA) was introduced and described in detail. The architecture is currently being implemented and preliminary results are encouraging.
As a result from this research, developers could use this architecture as a foundation for implementing and linking federation mechanisms within a single framework, testbed owners could provide resources using different interfaces at once to increase their sustainability, researchers would have an instrument to directly compare different protocols, and experimenters would have a single point of entry for their trials.
The short-term goals include the full implementation of plugins for the SFA, Teagle, and FRCP protocols. The medium-term planning treats the definition and integration of standardized data models and the integration of the architecture in the more commercialized contexts FI-PPP and EIT ICT Labs. Finally, the long-term objectives comprise the extension of the architecture's applicability from heterogeneous resource federation to generalized service life-cycle management.

Acknowledgments

This work was carried out with the support of the European Fed4FIRE (Grant No. 318389) and OpenLab (Grant No. 287581) FP7 projects.

References

[1] D. Talbot, "The Internet is broken," *Technology Review*, 2005.

[2] R. Jain, *The art of computer systems performance analysis*. John Wiley and Sons Ltd. Chichester, UK, 1991.

[3] A. Gavras, A. Karila, S. Fdida, M. May, and M. Potts, "Future Internet Research and Experimentation: The FIRE Initiative," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 3, pp. 89–92, 2007.

[4] J. Duerig, R. Ricci, L. Stoller, and M. Strum, "Getting started with geni: a user tutorial," *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 1, 2012.

[5] L. Peterson, S. Sevinc, J. Lepreau, and R. Ricci, "Slice-based Federation architecture," GENI, Tech. Rep., 2009.

[6] T. Rakotoarivelo, G. Jourjon, M. Ott, and I. Seskar, "OMF: a control and management framework for networking testbeds," *Operating systems review*, 2009.

[7] O. Mehani, G. Jourjon, J. White, T. Rakotoarivelo, R. Boreli, and T. Ernst, "Characterisation of the Effect of a Measurement Library on the Performance of Instrumented Tools," Tech. rep. 4879. NICTA, Tech. Rep., 2011.

[8] S. Wahle, C. Tranoris, S. Fox, and T. Magedanz, "Resource Description in Large Scale Heterogeneous Resource Federations," in *Testbeds and Research Infrastructure. Development of Networks and Communities*, Springer, 2012, pp. 100–115.

[9] D. Havlik, S. Schade, Z. Sabeur, P. Mazzetti, K. Watson, A. J. Berre, and J. L. Mon, "From Sensor to Observation Web with environmental enablers in the Future Internet," *Sensors*, vol. 11, no. 4, pp. 3874–907, Jan. 2011.

[10] A. Glikson, "FI-WARE: Core Platform for Future Internet Applications," in *Proceedings of the 4th Annual International Conference on Systems and Storage*, Haifa, 2011.

[11] D. Bernstein and V. K. Deepak, "Draft Standard for Intercloud Interoperability and Federation (SIIF)," 2012.

[12] W. Vandenberghe, B. Vermeulen, P. Demeester, A. Willner, S. Papavassiliou, A. Gavras, A. Quereilhac, Y. Al-hazmi, F. Lobillo, C. Velayos, A. Vico-oton, and G. Androulidakis, "Architecture for the Heterogeneous Federation of Future Internet Experimentation Facilities," in *Future Network and Mobile Summit*, 2013.

[13] M. Kelly, "Report: The TeleManagement Forum's Enhanced Telecom Operations Map (eTOM)," *Journal of Network and Systems Management*, 2003.

[14] T. Magedanz and S. Wahle, "Control framework design for Future Internet testbeds," *e&i Elektrotechnik und Informationstechnik*, 2009.

[15] J. Chase, L. Grit, and D. Irwin, "Beyond virtual data centers: Toward an open resource control architecture," *Selected Papers from the International Conference on the Virtual Computing Initiative (ACM Digital Library)*, 2007.

[16] L. Peterson and T. Roscoe, "The design principles of PlanetLab," *ACM SIGOPS Operating Systems Review*, 2006.

[17] T. Faber and J. Wroclawski, "A federated experiment environment for emulab-based testbeds," in *Tridentcom: 5th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities and Workshops*, 2009, pp. 1–10.

[18] M. Ott, I. Seskar, R. Siraccusa, and M. Singh, "ORBIT Testbed Software Architecture: Supporting Experiments as a Service," in *IEEE Tridentcom: First International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities*, 2005, pp. 136–145.

[19] The GENI Project Office, "ProtoGENI Control Framework Overview," GENI, Tech. Rep., 2009, pp. 1–41.

[20] E. Eide, L. Stoller, and J. Lepreau, "An experimentation workbench for replayable networking research," in *4th USENIX Symposium on Networked Systems Design & Implementation*, 2007, pp. 215–228.

[21] S. Jeong and A. Bavier, "GENI Federation Scenarios and Requirements," GENI: Global Environment for Network Innovations, Tech. Rep., 2010, pp. 1–16.

[22] T. Rakotoarivelo, G. Jourjon, and M. Ott, *Technical Report: Designing and Orchestrating Reproducible Experiments on Federated Networking Testbeds*, Sydney, Australia, 2012.

[23] G. Tselentis, *Towards the Future Internet: A European Research Perspective*. IOS Press, 2009.

[24] K. Campowsky, T. Magedanz, and S. Wahle, "Resource management in large scale experimental facilities," English, in *Proceeding of IEEE Network Operations and Management Symposium (NOMS)*, IEEE, 2010, pp. 930–933.

[25] S. Wahle and B. Harjoc, "Pan-European testbed and experimental facility federation–architecture refinement and implementation," *International Journal of Communication Networks and Distributed Systems*, vol. 5, no. 1, pp. 67–87, 2010.

[26] J. Strassner, "DEN-ng: achieving business-driven network management," English, in *NOMS 2002. IEEE/IFIP Network Operations and Management Symposium. ' Management Solutions for the New Communications World'(Cat. No.02CH37327)*, IEEE, 2002, pp. 753–766.

[27] S. Wahle, "A Generic Framework for Heterogeneous Resource Federation," PhD Thesis, Technische Universität Berlin, 2011.

[28] C. Tranoris and S. Denazis, "Federation Computing: A pragmatic approach for the Future Internet," in *2010 International Conference on Network and Service Management*, IEEE, Oct. 2010, pp. 190–197.

[29] A. Quereilhac, M. Lacage, C. D. Freire, T. Turletti, and W. Dabbous, "Nepi: An integration framework for network experimentation," in *19th International Conference on Software, Telecommunications and Computer Networks (IEEE SoftCOM)*, 2011, pp. 1–5.

[30] Y. Demchenko, C. Ngo, C. D. Laat, J. Rodriguez, L. M. Contreras, J. A. Garcia Espin, S. Figuerola, G. Landi, and N. Ciulli, "Intercloud Architecture Framework for Heterogeneous Cloud based Infrastructure Services Provisioning On-Demand," in *IEEE International Conference on Advanced Information Networking and Applications*, IEEE, 2013.

[31] D. Bernstein, E. Ludvigson, K. Sankar, S. Diamond, and M. Morrow, "Blueprint for the Intercloud - Protocols and Formats for Cloud Computing Interoperability," in *2009 Fourth International Conference on Internet and Web Applications and Services*, IEEE, May 2009, pp. 328–336.

[32] N. Grozev and R. Buyya, "Inter-Cloud Architectures and Application Brokering: Taxonomy and Survey," *Software: Practice and Experience*, 2012.

[33] I. Jacobson, M. Christerson, and P. Jonsson, *Object-oriented software engineering - a use case driven approach*, 4. A. Reno: ACM Press, 1992.

[34] J. O. Coplien and G. Bjørnvig, *Lean Architecture: for Agile Software Development*. Wiley, 2010, p. 376.

[35] S. Freeman and N. Pryce, *Growing Object-Oriented Software, Guided by Tests*. Addison-Wesley Professional, 2009, p. 384.

[36] J. Palermo, *The Onion Architecture*, 2008.

[37] R. C. Martin, *Agile Software Development: Principles, Patterns, and Practices*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2003.

[38] K. Beck, *Test Driven Development: By Example*. Addison-Wesley Professional, 2002, p. 240.

[39] R. C. Martin, "The dependency inversion principle," *C++ Report*, vol. 8, no. 6, pp. 61–66, 1996.