



UvA-DARE (Digital Academic Repository)

Software Defined Bearer Intercloud Networks

semantic-based network exchange for the IEEE P2302 Intercloud approach

Vij, D.; Bernstein, D.; Morsey, M.; Grosso, P.; Magedanz, T.; Willner, A.

DOI

[10.1109/ECTICon.2015.7207136](https://doi.org/10.1109/ECTICon.2015.7207136)

Publication date

2015

Document Version

Final published version

Published in

ECTI-CON 2015

License

Article 25fa Dutch Copyright Act

[Link to publication](#)

Citation for published version (APA):

Vij, D., Bernstein, D., Morsey, M., Grosso, P., Magedanz, T., & Willner, A. (2015). Software Defined Bearer Intercloud Networks: semantic-based network exchange for the IEEE P2302 Intercloud approach. In *ECTI-CON 2015: 2015 12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology : June 24-27, 2015, Rangsit University, Thailand* IEEE.
<https://doi.org/10.1109/ECTICon.2015.7207136>

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.

Software Defined Bearer Intercloud Networks

Semantic-based Network Exchange for the IEEE P2302 Intercloud Approach

Deepak Vij
USA
dvij_sj12@yahoo.com

Mohamed Morsey, Paola Grosso
Informatics Institute
University of Amsterdam
Netherlands
{m.morsey|p.grosso}@uva.nl

Alexander Willner
Architektur der Vermittlungsknoten
Technische Universität Berlin,
Germany
alexander.willner@tu-berlin.de

David Bernstein
Cloud Strategy Partners, LLC, USA
david@cloudstrategypartners.com

Thomas Magedanz
Next Generation Network Infrastructures
Fraunhofer FOKUS, Berlin, Germany
thomas.magedanz@fokus.fraunhofer.de

Abstract—In the contemporary cloud computing environment, the various cloud providers adopt different architectures for their systems and usually these are not compatible. The scalable exchange of information about these heterogeneous resources is often not addressed; still this represents an interesting challenge, since different Cloud systems and vendors have different ways to describe and invoke their services, to specify and communicate requirements. Hence a way to provide a common access to Cloud services and to discover and use required services in Cloud federations is highly desirable. The IEEE P2302 Intercloud approach is to define a scalable architecture and to adopt the OMN (Open-Multinet) ontology for supporting its operation. Our work specifically describes the inner workings of the Intercloud system in terms of the Intercloud Federation Application Programmers Interface (API), which are in turn based on the semantic definition of resources, Service Level Agreements (SLAs), and Bearer Network Provisioning Metadata.

Index Terms—Federation, Intercloud, SDN, Semantic, Ontology

I. INTRODUCTION

Cloud Computing has emerged as a model to provide access to large amount of data and computational resources by using seamless interfaces. Computing is being transformed to a model consisting of services and they are delivered without regard to where and how services are hosted.

The ease of management and configuration of resources and the low cost needed for setup and maintenance of services have made Cloud Computing widespread. Several commercial vendors now offer solutions based on Cloud architectures. More and more providers offer new different services every month, following their customer's needs. It is very hard to find a single provider which offers all services needed by end users, and we witness the emergence of federations of Clouds that can satisfy complex user needs.

Till recently, very few efforts have been done to define a unified standard for Cloud Computing, which is essential for the realization of Cloud Federations. This is a problem, since different Cloud systems and vendors have different ways to describe and invoke their services, to specify requirements and

to communicate. Hence a way to provide a common access to Cloud services and to discover and use required services in Cloud federations is highly desirable.

Our approach is to adopt a scalable architecture, namely the Standard for Intercloud Interoperability and Federation¹ (P2302) [1], that relies on a novel Semantic Web technologies, namely the Open-Multinet² (OMN) [2] ontology, for the definition of the Intercloud resources and services and implements a catalog for the IEEE Intercloud architecture.

In this paper, we describe the inner workings of the Intercloud system detailing the Intercloud Federation Application Programmers Interface (API) based on the semantic definition of resources, Service Level Agreements (SLAs), and Bearer Network Provisioning Metadata. Our work demonstrates how this API is used to dynamically provision a Software Defined Networking (SDN) based Virtual Private Cloud (VPC) using Virtual Private Networks (VPNs), creating a federating Bearer network needed for the transparent federation.

The remainder of the paper is structured as follows. We give a brief overview of related work in the context of Intercloud architectures and ontologies in Sec. II. In Sec. III we outline the high level details for the overall IEEE P2302 Intercloud approach; in Sec. IV we focus on the upper OMN ontology. The proposed API for the semantic based Intercloud Federation is described in Sec. V. We conclude and describe future work in Sec. VI.

II. RELATED WORK

Studies such as [3] were the first to highlight that emerging Cloud Computing environments diverged in their approaches to solve federation problems and the associated interoperability issues. This topic has become so interesting that by now well over 20 proposals [4] have been made for Cloud Computing interoperability architectures. In particular several industry

¹<http://standards.ieee.org/develop/project/2302.html>

²<http://open-multinet.info>

associations and Standards Developing Organization (SDO) groups have been formed to define approaches and requirements for Intercloud architectures. Examples are the IETF Cloud Reference Framework[5], the Open Data Center Alliance (ODCA), the TeleManagement Forum (TMF) [6], and the ETSI Cloud Standards Coordination activities. Besides this, other work like the European Grid Infrastructure (EGI) Federated Cloud Task Force, the Open Grid Forum (OGF) Open Cloud Computing Interface (OCCI) [7], the National Institute of Standards and Technology (NIST) Cloud Computing definition[8], the International Telecommunication Union Telecommunication Standardization Sector (ITU-T) Focus Group on Cloud Computing[9], or the Topology and Orchestration Specification for Cloud Applications (TOSCA) [10] model defined by the Organization for the Advancement of Structured Information Standards (OASIS) are focusing on particular challenges in this context. However, none of them focus on semantic and distributed large-scale interoperability.

In order to provide an effective and efficiently scalable and interoperable description of resources, a number of approaches from the Semantic Web[11] can be adopted. Namely, the Resource Description Framework (RDF) is a framework for the description of knowledge, defined and maintained by the World Wide Web Consortium (W3C), providing an underlying graph based data model describing resources and their relationships. An extension to RDF, known as Resource Description Framework Schema (RDFS) is used to provide native predicates to define basic relationships among the described resources. A more expressive formalism is known as Web Ontology Language (OWL). Based on this, information can be easily accessed and retrieved via the SPARQL Protocol And RDF Query Language (SPARQL) and by defining rules using the Semantic Web Rule Language (SWRL) it is possible to automatically infer further semantic information for a given data set.

Work that has adopted these mechanisms to define, exchange, and manage network topologies are numerous. In particular the Network Mark-Up Language (NML) [12] is an information model designed to describe and represent computer networks. NML is built upon on the preliminary work of the Network Description Language (NDL) [13]. The model underwent a thorough review and accreditation process to eventually become an OGF standard. The NML developers kept it abstract, with the possibility of extension in order to for it be customizable for the emerging network architectures and novel use cases. Based on this, the Infrastructure and Network Description Language (INDL) [14] models computing infrastructures in a technology independent manner. INDL imports NML as well, which authorizes it to seamlessly represent the networking part of a computing infrastructure. This ontology adds more concepts and relationships that are special to the computing, processing and storage aspect of an infrastructure, e.g., ProcessingComponent, and MemoryComponent classes. INDL further addresses the modeling and representation of resource and service virtualization. In other words, it supports description, discovery, modeling, composition, and monitoring of resources.

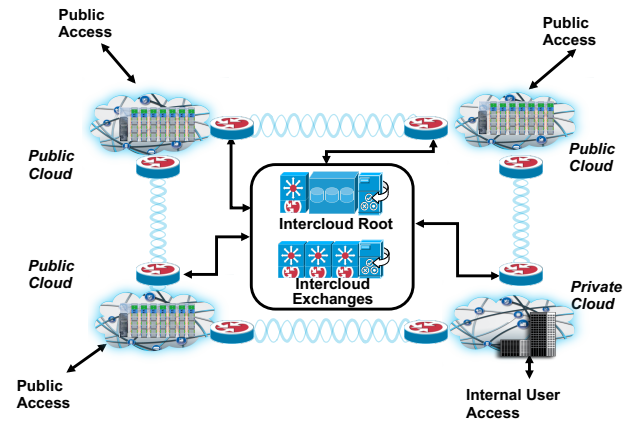


Fig. 1: IEEE P2302 Intercloud Overview

These existing achievements build the basis for the subsequently described semantic-enabled Intercloud approach to setup Intercloud network connections.

III. IEEE P2302

Instant interoperability and federation are rarely part of communication networks at the beginning. The telephone system is one example where geographical areas were not able to interoperate; human interaction was required to manually switch the phone systems. Until the relatively recently 1970's international "direct dials" were not available. In an analogous evolution, online services such as Prodigy, AOL, and CompuServe had no interoperability between them. When content was posted on one service it could not be read by clients from another service; and the same hold true for e-mail. Interoperability problems were solved in a highly scalable way in the phone system and in the Internet by developing new protocols, such as Signalling System 7 (SS7) [15] or the Autonomous System (AS) numbering and the Border Gateway Protocol (BGP) respectively.

In the context of Clouds Federation, the IEEE P2302 Working Group is pursuing the Standard for Intercloud Interoperability and Federation (SIIF) [1]. Starting from the first Intercloud blueprint[16] works and a number of publications, including security considerations[17], a first version of the standard has been elaborated. SIIF aims at defining a topology, a set of functionalities and a governance model to support Cloud interoperability and federation among different platforms. The scope of the resulting architecture is to ease the building of Intercloud solutions, enabling communication among different platforms thanks to a shared set of standards and resources definitions. The NIST included the IEEE Intercloud architecture in their Cloud Computing Technology Roadmap[8].

Topological elements that are illustrated in Fig. 1 include roots, clouds, exchanges (which mediate governance between Clouds), and gateways (which mediate data exchange between Clouds). Functional elements include name spaces, presence, messaging, resource ontologies, and trust infrastructure.

This architecture in conjunction with a semantic information model forms the basis for the development of a common understanding between different Cloud systems.

IV. OPEN-MULTINET ONTOLOGY

The previous work to semantically describe networks and infrastructures, has been extended and consolidated by the international Open-Multinet³ Forum, which is now being followed up by the W3C Federated Infrastructures Community Group⁴.

The scope of this group is to define upper ontologies to semantically describe federated infrastructures, and to support the management of the whole resource life cycle. A set of terms have been specified, i.e., concepts, attributes and relationships, which are used to represent the knowledge of this specific domain [18]. In other words, the upper ontologies cover only a specific knowledge field, namely the management of resources within and between autonomous administrative domains [19].

The ontology bundle consists of nine ontologies, specifically the *omn* upper ontology and eight descendant ontologies⁵. Those descendant ontologies are 1) *omn-federation*; 2) *omn-life-cycle*; 3) *omn-resource*; 4) *omn-component*; 5) *omn-service*; 6) *omn-monitoring*; 7) *omn-policy* and, 8) *omn-domain*.

The OMN upper ontology defines the abstract terms (concepts and properties) required for describing and modeling Cloud systems and for modeling federated Cloud environments as well. It includes classes “Attribute”, “Component”, “Dependency”, “Environment”, “Group”, “Layer”, “Reservation”, “Resource”, and “Service”. Fig. 2 shows the key concepts and properties of the *omn* upper ontology.

A. OMN Classes

The *omn* upper ontology contains the following classes:

- *Resource*: a stand-alone entity, which can be provisioned, i.e., granted to a tenant and/or controlled via APIs e.g., network node.
- *Service*: is a manageable entity, which can be controlled and/or used via APIs or capabilities it supports, e.g., Broker or SPARQL endpoint.
- *Component*: constitutes a part of a “Resource” or a “Service”, e.g., a port of network node.
- *Attribute*: it helps to describe the characteristics and properties of a specific “Resource”, “Group”, “Resource”, or “Component”, e.g., Quality of Service (QoS).
- *Group*: is a collection of “Resource”, “Service” or “Group”.
- *Dependency*: it describes a unidirectional relationship between “Resource”, “Service”, “Component” or “Group”. This class opens up the possibility to add more properties to a dependency via annotation.

³<http://open-multinet.info>

⁴<https://www.w3.org/community/omn>

⁵All ontologies can be found at <http://github.com/open-multinet/playground-rspecs-ontology>

- *Layer*: describes a place within a hierarchy a specific “Group”, “Resource”, “Service” or “Component” can adapt to.
- *Environment*: the conditions under which a “Resource”, “Group”, or “Service” is operating. For example concurrent virtual machines.
- *Reservation*: a specification of a guarantee. Moreover, It is a subclass of the “Interval” class of the W3C Time ontology[20]. For example Start and end times.

B. OMN Properties

The properties of the *omn* upper ontology are as follows:

- *adaptableTo*: relates a “Layer” to another “Layer” to which it adapts. It is also the inverse of “adaptableFrom”.
- *adaptsFrom*: determines the “Group”, “Resource”, “Service” or “Component” from which another “Group”, “Resource”, “Service” or “Component” adapts. “adaptsTo” is its inverse property.
- *fromDependency*: relates a “Group”, “Resource”, “Service” or “Component” to the “Dependency” to which it belongs. It is the inverse of property “toDependency” as well.
- *hasAttribute*: the Attribute associated with a “Component”, “Resource”, “Service” or “Group”, and it is the inverse of “isAttributeOf”.
- *hasComponent*: links a “Component”, “Resource”, “Service” to its subcomponent. Further, it is the inverse of property “isComponentOf”.
- *hasGroup*: connects a “Group” to its subgroup and it is the inverse of “isGroupOf”.
- *hasReservation*: relates “Group”, “Resource” or “Service” to its “Reservation”. “isReservationOf” is its inverse.
- *hasResource*: declares that a specific “Group” has a “Resource”. Moreover, it is the inverse of “isResourceOf”.
- *hasService*: declares that a “Group”, “Resource” or “Service” provides a “Service”. “isServiceOf” is its inverse.
- *withinEnvironment*: defines the “Environment” in which a “Group”, “Resource”, “Service” or “Component” operates.

In conjunction with the IEEE P2302 Intercloud architecture, this semantic information model provides a basis to setup large-scale interoperable Clouds that can be interconnected using dedicated Bearer networks on demand, i.e. VPCs.

V. INTERCLOUD FEDERATION API

In order to dynamically provision a VPCs it is essential to implement a federation signaling strategy. In an example scenario for federation of cloud computing resources, let us take Virtual Machines (VMs) as an example resource to be federated. A Requesting (or “Home”) Cloud needs more VMs to satisfy a user demand and so inquires to the Intercloud Exchange to which it is connected to try to obtain more VMs through the Intercloud.

In such a federated environment, the resource may not be physically running on the Requesting Cloud but it appears to be in a transparent manner as if the resource is Requesting Cloud’s own computing resource. The VMs need to be provisioned in

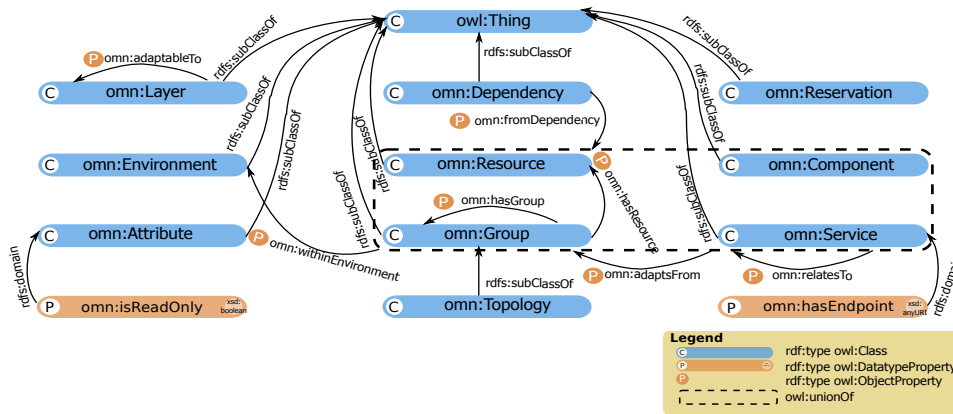


Fig. 2: The key concepts and properties of the omn upper ontology.

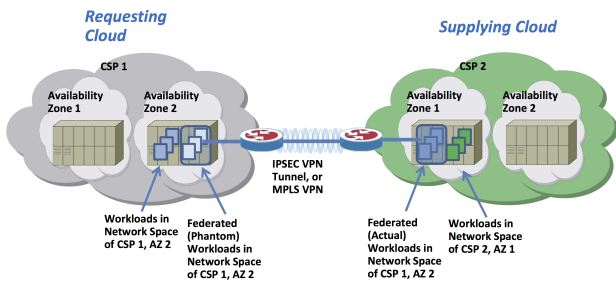


Fig. 3: Federation using Virtual Private Cloud over Dynamically Provisioned Virtual Private Networks

the Supplying Cloud in such a manner so that they appear to be “local” in the Requesting Cloud. Such dynamic provisioning has very recently been demonstrated using SDNs as shown in Fig. 3.

In order to explain this further, we assume the Exchange, Requester/Home Cloud, and Supplier Clouds have already taken care of connectivity housekeeping including establishment of the Trust Infrastructure and Services Infrastructure with endpoints on the respective Gateways. Additionally, we further assume completed registration of Network Management points with the Exchange.

The next level of detail is the “registration” of the resources which the Supplier Cloud has, along with the subsequent registration of the need for resources by the Requester/Home cloud. This is accomplished using the Semantic Resource Definition using an OWL specification against a common “Ontology Schema” for resources, SLAs and Bearer Networks. Suppliers of resources declare the offered resources, the SLA provided, and the Bearer Network within which they can be supplied.

Requesters of resources declare the desired resources, the required minimum SLA, and the Bearer network within which they can be consumed as illustrated in Fig. 4.

Examples of Bearer Network Declarations include details such as Network Performance specifications; Type of Network specifications; Security of Network specifications etc. Next,

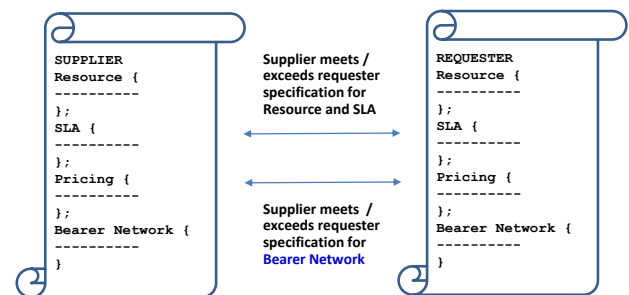


Fig. 4: Supplier and Requester Semantic Resource Definitions

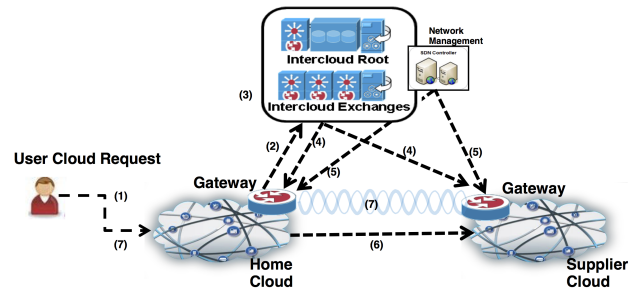


Fig. 5: Federation Steps

the Supplier Cloud contacts an Exchange it is affiliated with and supplies it with Semantic Resource Definitions indicating inventory for Federation. Following are the detail step by step communications protocol used by the Intercloud topology elements in the dynamic Federation scenario, as illustrated by Fig. 5.

1(a). User Requests Cloud Resources by issuing existing cloud specific User-to-Network Interface (UNI) API call to the Home Cloud.

1(b). If the Home Cloud is able to fulfil the entire User Request locally, it does so and returns the User Request API. The flow stops here as there is no need for the remaining Intercloud federation steps.

2(a). When the Home Cloud wishes to use Intercloud federation to fulfil certain resources of the User Request, the

Home Cloud uses a specific interface to its associated Gateway communicating the User Request.

2(b). The Gateway constructs the canonical Resource Description Declaration Including Resource, SLA, and Bearer Network sections. The Resource Description Declaration shall be expressed in the “Home Cloud Form”, e.g., with any and all of the acceptable options and quantifiers for the Home Cloud. For example, Bearer Network alternatives, acceptable geographies, etc.

2(c). Subsequently, the Gateway serializes this into the Federation API format (e.g., as appropriate for the Services Framework) including pre-pending Home Cloud credentials and other conventions such as time stamping, deadline requirements etc., and finally invokes the Federation API onto the implicit Gateway at the Exchange.

3(a). The implicit Gateway residing at the Exchange accepts the Federation API, extracts the contents and forms appropriate Exchange-Internal APIs for the Solver/Matching process.

3(b). For example, the extracted Resource Description Declaration General Form might be used to create a SPARQL/SWRL query to the Solver/Matcher; the timestamp/deadline requirements might be used to advise what algorithms to use (find first suitable match, find best suitable match, etc.)

3(c). The Solver/Matcher seeks qualifying inventory (in this example from Supplier Cloud) to resolve the constraints/quantifiers and if found, constructs a canonical Resource Description Declaration including the specific Resource, SLA, and Bearer Network which the Supplier Cloud would deliver to fulfil the Home Cloud Resource Description Declaration. The Resource Description Declaration is in the “Supplier Cloud Form”.

4(a). If the Exchange did not resolve the inventory on any Supplier Cloud it returns an error to the Home Cloud Gateway, which unwinds this to return the User Request API with an error.

4(b). If the Exchange found inventory on a Supplier Cloud, the Implicit Gateway in the Exchange serializes that fulfilling Resource Description Declaration “Supplier Cloud Form” into the Federation API format including pre-pending Supplier Cloud credentials and other details (i.e., time stamping, inventory availability time or quantity information), and invokes them onto the Gateway at the Home Cloud.

4(c). The Exchange will also construct a Federation API for the Supplier Cloud, including Home Cloud Credentials, advising it of Resource Description Declaration Supplier Cloud Form that has been given to the Home Cloud. The Supplier Cloud may or may not choose to optimistically reserve or begin to provision those resources. At this stage, the User Request API still has not returned yet.

5(a). The Network Management (SDN Controllers) in the Exchange reaches out to the Network Management points in the Home Cloud and Supplier Cloud Gateways, respectively.

5(b). The Home Cloud and the Supplier Cloud Gateways may have self-provisioned based on the Bearer network information in the Federation API supplied in Step 4 earlier.

5(c). If not, the Network Management (SDN Controller) provisions the connectivity to the Bearer Network specified in the Resource Description Declaration Supplier Cloud Form.

5(d). If the Bearer Network cannot be provisioned, the Exchange returns an error to the Home Cloud Gateway, which unwinds this to return the User Request API with an error. In this case the Supplier Cloud should unwind any self-provisioned configuration they may have done in advance.

6(a). The Home Cloud Gateway serializes the Resource Description Declaration Supplier Cloud Form into the Federation API format including pre-pending Home Cloud credentials along with other details (i.e., time stamping, deadline requirements etc.), and invokes them onto the Supplier Cloud Gateway.

6(b). The Supplier Cloud provisions the resources, if not done optimistically earlier.

6(c). If the Supplier Cloud resources cannot be provisioned, it returns an error to the Home Cloud Gateway, which unwinds this to return the User Request API with an error. The Supplier Cloud should also unwind any optimistic provisioning it might have done earlier.

6(d). If the provisioning is successful, status is provided to the Home Cloud Gateway which unwinds this to return the User Request API with expected success results.

7(a). The Resources are provided over the Bearer Network.

7(b). Resources are seamlessly provided to the User Request as if they were supplied directly by the Home Cloud.

VI. CONCLUSION AND FUTURE WORK

The existing cloud computing environments possess heterogeneous architectures, and thus provide incompatible APIs for creating and managing their resources and services. In this paper, we propose an ontology based framework for supporting seamless access to heterogeneous cloud service providers. Our framework adopts the Open-Multinet ontology for supporting its functionality.

Furthermore, we detailed a use-case indicating the entire process involved by this framework. The process starts by specifying the resources, SLA and also the Bearer Networks via the proposed ontology schema. These specifications include all requirements this request has, e.g., network performance and network type. When the Home Cloud realizes that user request can not be fulfilled locally, it initiates the Cloud Federation process. Eventually, upon contacting the Supplier Cloud, and making sure that the required resources can be provisioned, the Home Cloud seamlessly grants the resources to the user.

Our short-term goal is to extend the OMN ontology by linking it to semantic cloud resource definitions such as the mOSAIC[21] ontology. In medium term, we want to concentrate on the implementation of a sophisticated solver/matching mechanism[22] that returns, based on an abstract Home Cloud request, a complete resource topology including the according bearer networks. Finally, the long-term objective is to implement and define a full standard on an Internet scale, semantic design to make the Intercloud reality.

VII. ACKNOWLEDGMENTS

Research for this paper was partially financed by the Dutch COMMIT and the German DAAD UNIFI projects. We thank our project partners for their contributions and their collaboration to this research work.

REFERENCES

- [1] D. Bernstein and V. Deepak, "Draft standard for intercloud interoperability and federation (siif)," IEEE P2303, Tech. Rep., 2014.
- [2] A. Willner, C. Papagianni, M. Giatili, *et al.*, "The openmultinet upper ontology - towards the semantic-based management of federated infrastructures," in *10th Int. Conf. on Testbeds and Research Infrastr. for the Dev. of Netw. & Comm. (to appear)*, Vancouver, Canada: IEEE, 2015.
- [3] L. Mei, W. K. Chan, and T. H. Tse, "A tale of clouds: paradigm comparisons and some thoughts on research issues," in *Asia-Pacific Services Computing Conference (APSCC)*, IEEE, 2008, pp. 464–469.
- [4] N. Grozev and R. Buyya, "Inter-cloud architectures and application brokering: taxonomy and survey," *Software: Practice and Experience*, vol. 44, no. 3, pp. 369–390, 2012.
- [5] B. Khasnabish, C. JunSheng, S. Ma, *et al.*, "Cloud reference framework," Internet Engineering Task Force (IETF), Internet-Draft draft-khasnabish-cloud-reference-framework-02, 2014.
- [6] TM Forum, "Tr139, service delivery framework (sdf) overview, release 2.0," TM Forum, Tech. Rep., 2008.
- [7] T. Metsch and A. Edmonds, "Open cloud computing interface–infrastructure," in *Standards Track, no. GFD-R in The Open Grid Forum Document Series, Open Cloud Computing Interface (OCCI) Working Group, Muncie (IN)*, Open Grid Forum (OGF), 2010.
- [8] US Government Cloud Computing Technology Roadmap Volume III, *First working draft - technical considerations for usg cloud computing deployment decisions*, 2011.
- [9] ITU-T Focus Group on Cloud Computing, "Part 2: functional requirements and reference architecture," TELECOMMUNICATION STANDARDIZATION SECTOR OF ITU, Tech. Rep. Version 1.0, Feb. 2012.
- [10] D. Palma and T. Spatzier, *Topology and orchestration specification for cloud applications (tosca) version 1*, Nov. 2013.
- [11] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific American*, vol. 284, no. 5, pp. 34–43, May 2001.
- [12] J. van der Ham, F. Dijkstra, R. Lapacz, *et al.*, "Gfd.206: network markup language base schema," Tech. Rep., 2013.
- [13] J. Van Der Ham, "A semantic model for complex computer networks: the network description language," PhD thesis, Universiteit van Amsterdam, 2010, p. 164.
- [14] M. Ghijsen, J. van der Ham, P. Grosso, *et al.*, "A semantic-web approach for modeling computing infrastructures," *Computers and Electrical Engineering*, vol. 39, no. 8, pp. 2553–2565, 2013.
- [15] ITU-T, "Q.700: introduction to ccitt signalling system no. 7," ITU-T, Tech. Rep., 1993.
- [16] D. Bernstein, E. Ludvigson, K. Sankar, *et al.*, "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.
- [17] D. Bernstein and D. Vij, "Intercloud security considerations," in *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*, IEEE, 2010, pp. 537–544.
- [18] J. Heflin, "Owl web ontology language - use cases and requirements," World Wide Web Consortium (W3C), W3C Recommendation, 2004.
- [19] L. Yu, *Introduction to the Semantic Web and Semantic Web Services*. CRC Press, 2007, pp. 1–341.
- [20] J. R. Hobbs and F. Pan, "Time ontology in owl," Tech. Rep., Sep. 2006, p. 133.
- [21] F. Moscato, R. Aversa, B. Di Martino, *et al.*, "An analysis of mosaic ontology for cloud resources annotation," in *Federated Conf. on Computer Science and Information Systems (FedCSIS)*, IEEE, 2011, pp. 973–980.
- [22] A. Willner, B. Di Martino, G. Cretella, *et al.*, "Towards an ontology-based intercloud resource catalog - the ieee p2302 intercloud approach for a semantic resource exchange," in *4th Int. WS on Cloud Computing: Interclouds, Federations, and Interoperability*, Tempe, Arizona: IEEE, 2015.