DBcloud: Semantic Dataset for the cloud

Morsey, M.; Willner, A.; Loughnane, R.; Giatili, M.; Papagianni, C.; Baldin, I.; Grosso, P.; Al-Hazmi, Y.

DOI
10.1109/INFCOMW.2016.7562073

Publication date
2016

Document Version
Final published version

Published in
2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)

DBcloud: Semantic Dataset for the Cloud

Mohamed Morsey
Informatics Institute
University of Amsterdam, Netherlands

Alexander Willner & Robyn Loughnane
Next Generation Networks
Technische Universität Berlin, Germany

Mary Giatili & Chrysa Papagianni
NETMODE Lab
NTUA, Athens, Greece

Ilya Baldin
RENCI/UNC
Chapel Hill, NC, USA

Paola Grosso
Informatics Institute
University of Amsterdam, Netherlands

Yahya Al-Hazmi
Next Generation Networks
Technische Universität Berlin, Germany

***Abstract*—In cloud environments, the process of matching requests from users with the available computing resources is a challenging task. This is even more complex in federated environments, where multiple providers cooperate to offer enhanced services, suitable for distributed applications. In order to resolve these issues, a powerful modeling methodology can be adopted to facilitate expressing both the request and the available computing resources. This, in turn, leads to an effective matching between the request and the provisioned resources. For this purpose, the Open-Multinet ontologies were developed, which leverage the expressive power of Semantic Web technologies to describe infrastructure components and services. These ontologies have been adopted in a number of federated testbeds. In this article, DBcloud is presented, a system that provides access to Open-Multinet open data via endpoints. DBcloud can be used to simplify the process of discovery and provisioning of cloud resources and services.**

***Index Terms*—Linked Open Data, knowledge extraction, infrastructure federation, Interclouds, testbeds, RDF, OWL**

I. Introduction

Cloud infrastructures and services are becoming the state of the art foundation for supporting the operations of large Web infrastructures and applications, for processing big data and for providing users with ubiquitous and scalable computing and storage platforms. Cloud computing has several provisioning models, specifically Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). This operation framework is proving effective due to its scalability and the associated economic advantages. The cloud resource provisioning process passes through a number of phases, namely (i) discovery, (ii) selection, (iii) reservation, (iv) provisioning, (v) monitoring, (vi) control, and (vii) termination.

The process of modeling cloud infrastructures in a manner that supports effective matching of users' requests with available cloud resources is a challenging task. The issue becomes even more complex in the context of distributed cloud systems, i.e. Intercloud systems, particularly when these distributed systems utilize different modeling methodologies. Cloud resource matching and recommendation is therefore of great importance for assigning the right resources to the user.

Several approaches have been developed to address the issue of modeling cloud infrastructures. Most of these approaches adopt Extensible Markup Language (XML) as a standard for representing cloud resources and user requests, e.g. the Topology and Orchestration Specification for Cloud Applications (TOSCA) [20]. However, adopting a Semantic Web approach enables intelligent selection of cloud services (e.g. cloud IaaS), exploiting data semantics, knowledge inference and reasoning, thus facilitating the automation of operations related to the life cycle of cloud services (e.g. discovery).

A common model supports efficient matching of user requests with the available resources, as well as seamlessly facilitating the merging of different cloud resource descriptions. Semantic Web standards, e.g. Web Ontology Language (OWL) [19], play an important role in enabling the standardization of cloud models. The methodology used for the DBcloud system adopts Semantic Web technologies to develop a holistic cloud and Intercloud model. A comprehensive package of ontologies, Open-Multinet (OMN) [27], was developed for modeling cloud infrastructures. The work presented in the current paper constitutes further progress for the work described in [27]. The OMN ontology suite is used to convert data into Resource Description Framework (RDF) [8] triplets. For DBcloud, data was extracted from two major initiatives for federating experimental infrastructures, including cloud-computing testbeds. These initiatives are Global Environment for Network Innovations (GENI) [3] and Future Internet Research and Experimentation (FIRE) [10], which both currently use XML-based GENI Resource Specifications (RSpecs).

Adopting Semantic Web technologies for modeling cloud infrastructures has several advantages:

1) A common standardized model is used to describe cloud and Intercloud infrastructures.
2) Different resources and descriptions can be semantically related and connected.
3) Pitfalls in models can be detected early, before provisioning.
4) Complex queries can be carried out to discover resources.
5) Once cloud resources are semantically described, they can be interlinked to other Linked Open Data (LOD) cloud data sets, which dramatically enriches resource descriptors.

The significance of DBcloud is demonstrated by an example in Section VI where the system is used for resource discovery and selection over federated testbeds, focusing on cloud resources.

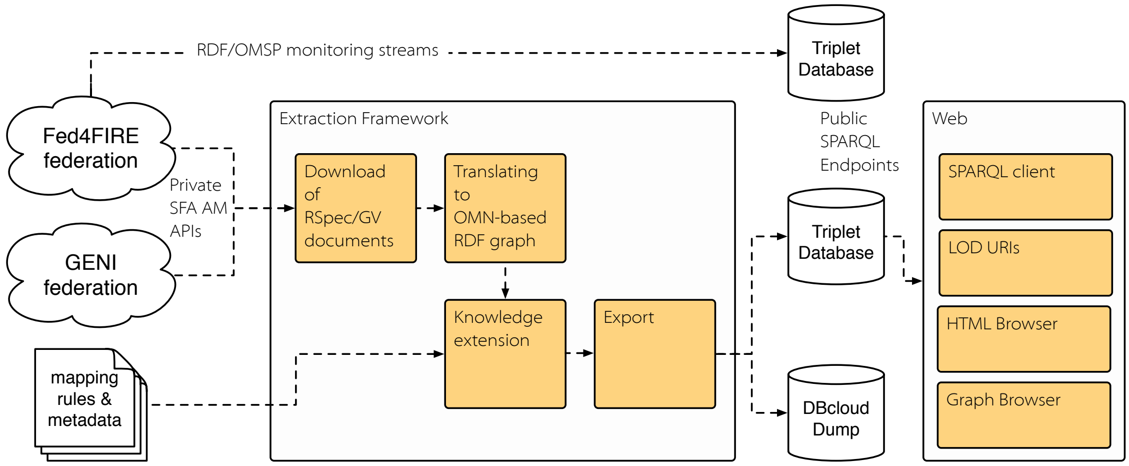

Fig. 1: Overview of the DBcloud extraction framework (analog to [15])

This application shows how Semantic Web technologies both standardize cloud resource descriptors and make the matching process between the request and resource more efficient.

The remainder of the paper is structured as follows. A brief overview of related work in the context of description and discovery of computing and cloud resources is given in Section II. In the subsequent Section III, the OMN ontology is presented, before a description of DBcloud in Section IV. The performance of translation of data for DBcloud is discussed in Section V and inference rules in Section VI. Finally, conclusions, considerations and descriptions for future work are given in Section VII.

II. Related Work

Numerous disciplines have adopted Semantic Web technologies, including future Internet and cloud infrastructures and services, in order to model computing infrastructures and improve their operation. Haak et al. [11] proposed an ontology-based optimization methodology that enables cloud providers to detect the best resource set that satisfies a user's request. However, the approach concentrates on the optimization of resource composition and overlooks Intercloud environments.

Haase et al. [12] introduced an approach for administering enterprise cloud environments. Semantic Web technologies are adopted to solve challenges related to the administration of enterprise cloud frameworks. Haase et al. proposed a Semantic Web–based product called eCloudManager, which incorporates an ontology for modeling its cloud data. However, the system and its ontology only focus on the management aspect of cloud systems, and this data is not open for usage.

Pedrinaci et al. [21] introduced Linked USDL, a vocabulary that utilizes research conducted on Semantic Web services and applies it to USDL [7, 18]. Linked USDL is quite comprehensive for describing services in order to support automated processing. As it focuses only on services, it is not efficient for describing cloud infrastructures.

Semantic Web service discovery [14, 24] tackles the problem of automated discovery of Web services satisfying a set of given requirements. This discovery process adopts a matchmaking algorithm for finding potential Web services capable of solving the problem at hand. However, these methodologies are inapplicable for complex interconnected computing infrastructures.

Santana-Pérez et al. [23] proposed a scheduling algorithm suitable for federated hybrid cloud systems. This algorithm utilizes semantic technologies for scheduling and allocating tasks to the most suitable resources. The approach reuses and adapts the UCI project ontologies[1] as its information model. Although the ontologies cover a wide range of details, they do not cover Intercloud systems.

Le and Kanagasabai [9, 17] proposed ontology-based methodologies for discovering and brokering cloud services. These methodologies use Semantic Web technologies for user requirements and cloud provider advertisements, and then apply a matchmaking algorithm to find the best match between a requirement list and the list of advertised units. Multiple levels of matching are defined, ranging from an exact match to no match. Nevertheless, these methodologies concentrate only on IaaS provisioning. Furthermore, they neither provide their data as dumps nor as a SPARQL Protocol And RDF Query Language (SPARQL) [1] endpoint, which hinders data access and reuse.

III. Open-Multinet

OMN is an ontology suite for modeling and describing infrastructure components and services. OMN consists of an upper ontology and a set of eight descendant ontologies (cf. Figure 2): (1) *omn-federation*; (2) *omn-lifecycle*; (3) *omn-resource*; (4) *omn-component*; (5) *omn-service*; (6) *omn-monitoring*; (7) *omn-policy*; and (8) domain specific extensions called *omn-domain-xxx*.

The OMN upper ontology includes a set of classes that represent the general concepts required to model federated infrastructures and their respective components and services.

[1] http://code.google.com/p/unifiedcloud

The key concepts of the upper ontology include Resource, Service, Component, and Reservation. Resource represents a stand-alone component of an infrastructure that can be provisioned by a user, such as a network node. Service is a manageable entity that can be controlled or used via Application Programming Interfaces (APIs), e.g. an SSH login. Component is a part of a Resource or Service, e.g. a port of network node. Reservation represents a guarantee that a provisionable entity can be used for a certain time period. For this reason, Reservation is a subclass of the Interval class of the World Wide Web Consortium (W3C) Time ontology [13]. More details about the OMN ontology can be found in [27].

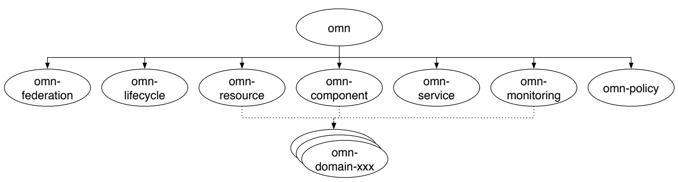

Fig. 2: Open-Multinet ontology hierarchy

OMN has an accompanying Java library[2] for converting GENI RSpecs, TOSCA and Yet Another Next Generation (YANG) [5] templates to OMN RDF models. The Java library uses Java Architecture for XML Binding (JAXB) to marshall and unmarshall XML files. It has a Representational State Transfer (REST) API and Web Graphical User Interface (GUI) to enable efficient and standardized translations of existing XML descriptions of cloud resources into OMN.

IV. DBcloud

Of central relevance to and inspiration for the current work, the DBpedia project [2] serves to extract structured knowledge from Wikipedia, making it freely available in a number of languages. The result is a database with tens of millions of RDF triples. Knowledge is structured following an ontology maintained by the community. The DBpedia data set is further linked to other LOD data sets. This structured information can then be browsed through on the Web, downloaded as an RDF dump or searched from the SPARQL endpoint. The SPARQL endpoint deals with an average of around three million queries per day [15]. A major application of DBpedia is in the area of Natural Language Processing (NLP) [16]. Making such a large amount of structured data available does, however, present a number of challenges, in particular inconsistency, ambiguity, uncertainty, data provenance and implicit knowledge [2]. There are also a number of similar projects whose goal is to extract information on a large scale from Wikipedia and other sources, including the Yet Another Great Ontology (YAGO) [25] developed at the Max Planck Institute for Informatics.

DBpedia adheres to LOD standards as per the five basic design principles for linked data proposed by Berners-Lee [4]. Data should (1) be available on the Web, (2) be machine readable, (3) be in a nonproprietary format, (4) use RDF standards, (5) and be linked RDF.

Similar to the DBpedia context, a large amount of semistructured information is available describing the GENI and FIRE testbed federations. This includes details about the testbeds involved and the heterogeneous resources offered, reservation information, and monitoring data. The information is encoded in different data formats and stored within the distributed infrastructures.

Inspired by the DBpedia approach, the proposed DBcloud extracts information from these federations and related testbeds and makes this information semantically accessible on the Web. The resulting knowledge base currently describes more than 100 aggregates, 2,500 nodes, 6,700 links, and about 11,000 interfaces. This results in 3.3 million statements, with the potential to grow by many times this amount.

In Figure 1 an overview of the DBcloud extraction framework is given. Its design follows the DBpedia extraction framework [15] and the result is currently available at http://lod.fed4fire.eu (cf. Figure 3) and is described using, among others, the Vocabulary of Interlinked Datasets. To gather the related information from the infrastructures, two different methods of the Slice-based Federation Architecture (SFA) [22] Aggregate Manager (AM) API are called at regular intervals. First, JSON-encoded metainformation about the testbeds is extracted by calling the GetVersion (GV) method. Second, RSpec-encoded information about published resources and their reservation information are extracted by calling the ListResources method, using X.509 certificates, which are trusted by each infrastructure involved, for authentication. The downloaded documents are then translated into a semantically annotated RDF graph using the omnlib Java library and the OMN ontology. To extend the knowledge encoded in this graph, the Apache Jena inference engine is used within this process by applying infrastructure-specific rules (as described in Section VI). Finally, after adding more static information about the federations, the resulting knowledge graph is written in a Sesame triplet database and a Turtle (TTL) serialized file. Information stored in the database is then available via a public SPARQL endpoint, with URIs following LOD principles. Further, there is a HTML rendering and a graph browser using LodView and LodLive [6].

As an example, Listing 1 shows such static information about the Federation for FIRE (Fed4FIRE) [26] federation. The federation consists of multiple federation members and can be rendered as HTML[3]. The basic information about Fed4FIRE and its members is encoded using well-known vocabularies and, by applying the *rdfs:isDefinedBy* property, further links to the corresponding DBpedia entry, which provides more detailed information.

While the RSpecs include information about current reservations, even more dynamic information about the testbeds, such as monitored resource utilization or availability, is exported

[2]https://github.com/w3c/omn

[3]http://lod.fed4fire.eu/id/fed4fire.eu

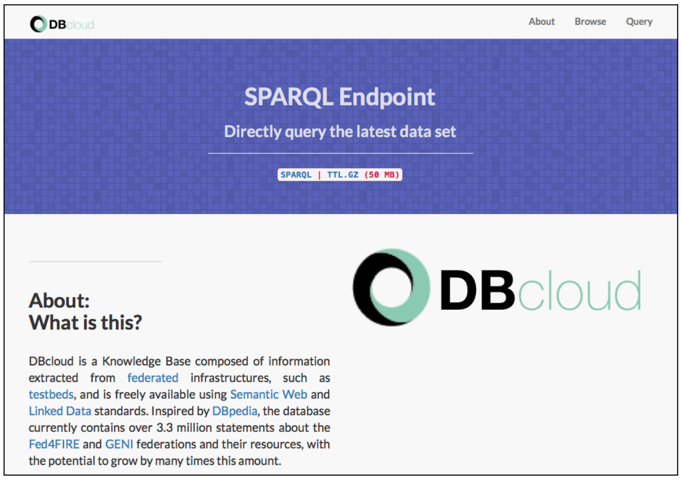

Fig. 3: DBcloud Web site

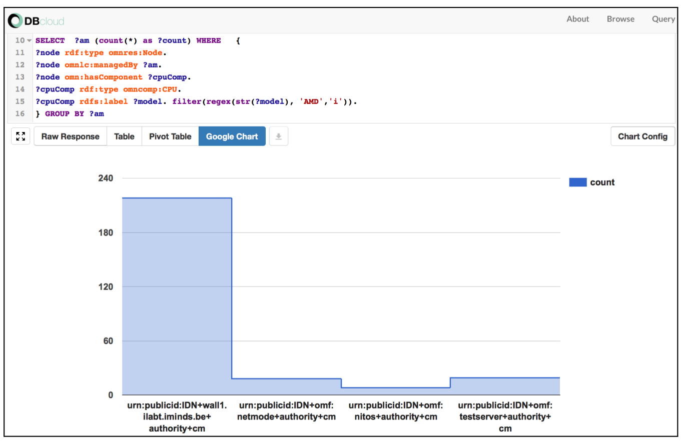

Fig. 4: Finding testbeds containing nodes with AMD hardware

from FIRE testbeds using the OML Measurement Stream Protocol (OMSP). By defining an own RDF/OMSP serialization, this information is translated within selected testbeds and stored in a second SPARQL database, as shown in Figure 1. As the same unique resource identifiers are used, this data can be used together with the first triplet database by testbed users and tool developers for sophisticated resource selection.

To demonstrate the benefits of the presented approach, Figure 4 depicts an excerpt of the user GUI, used to query and visualize information about the resources available within the federation. In the given query, the user has searched for nodes that contain CPUs manufactured by AMD and is interested in the URNs of the related testbeds. The user can then use a relevant tool to contact the AM API by its URN and make a reservation for theses resources.

Listing 1: Fed4FIRE federation description (excerpt)

```
<http://lod.fed4fire.eu/id/fed4fire.eu> rdf:type omn-federation:Federation ;
        dc:title "Fed4FIRE␣Knowledge␣Base"^^xsd:string ;
        rdfs:comment "The␣Fed4FIRE␣federation␣is␣a␣collaboration␣of␣
            ..." ;
        foaf:homepage <http://lod.fed4fire.eu/> ;
        void:sparqlEndpoint <http://lod.fed4fire.eu/sparql> ;
        omn-federation:hasFederationMember <http://lod.fed4fire.eu/id/
            fokus.fraunhofer.de> ,
              <http://lod.fed4fire.eu/id/i2cat.net> ,
              ...
```

V. Translator Benchmark

Initial data for DBcloud was taken from the actual Advertisement RSpecs from testbeds that are part of the FIRE and GENI projects. These XML files were then translated to RDF TTL using the OMN translator, with simple output as raw RDF. Of great importance to the potential scalability of DBcloud is the time taken for such translations, in particular with regards to the number of XML elements involved. At the time of writing, 100 Advertisement RSpecs had been extracted, of which six contained errors (e.g. not adhering to the RSpec XML Schema Definition (XSD) file) and could not be translated without manual changes. Tests were run on a MacBookPro with OS X Yosemite, a 2.8GHz Intel Core i7 processor and 8GB of RAM. Running a translation over all correct RSpecs produced median values of 24 milliseconds from XML to JAXB and 20 milliseconds from JAXB to RDF, giving a total median translation time of 44 millisecond from XML to RDF. As shown in Figure 5, translation times appear to be roughly linearly correlated with the number of XML elements translated, with a median of 180 elements and a maximum of 159,372 translated. This linear correlation indicates upwards scaling should be possible, although more data is required to confirm. At this stage, no major limiting factors have been identified and, given appropriate processing power, translation should be possible in most foreseeable use cases.

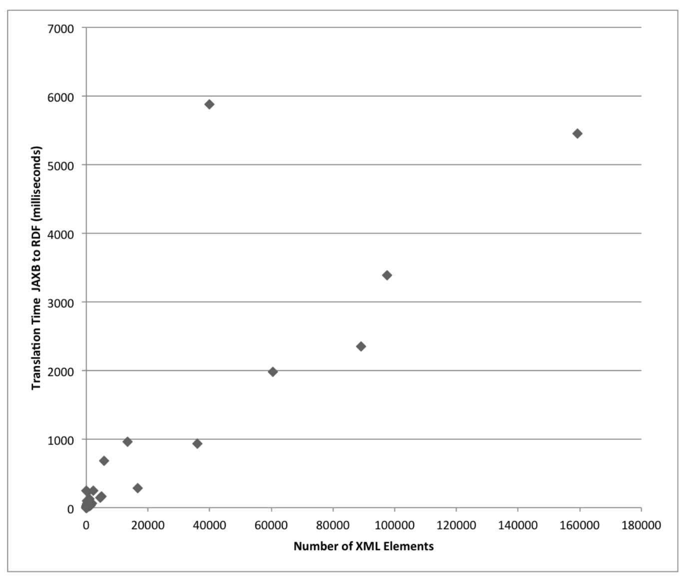

Fig. 5: JAXB to RDF translation times versus number of XML elements

VI. Knowledge Extension and Enquiry

As mentioned in Section IV, the knowledge graph may be extended by applying infrastructure-specific inference rules.

For example, infrastructure providers in the federation do not explicitly advertise the hardware configurations of their resources in the RSpec XML documents provided, and these are thus not translated into RDF. Instead, such information is encoded in each resource's *hardware type*, e.g. pcgen3, pcgen2 etc. Listing 2 represents a subset of the inference rules used to expand the knowledge base with CPU-related information regarding pcgen3 nodes. For a pcgen3 node, hardware properties per node include two Hexacore Intel E5645 (2.4GHz) CPUs, 24GB RAM, a 250GB hard disk, and one to five 1Gbit NICs.

Listing 2: Infrastructure knowledge (excerpt)

```
[rule1:
(?node omnres:hasHardwareType ?hwtype)
(?hwtype rdfs:label ?label)
regex (?label , "pcgen0?3.*")
makeTemp(?cpuComp)
->
(?cpuComp rdf:type owl:NamedIndividual)
(?cpuComp rdf:type omncomp:CPU)
(?cpuComp rdfs:label "Intel␣E5645␣CPU")
(?cpuComp omn:hasModelType "Intel␣E5645")
(?cpuComp rdfs:label "Hexa␣Core␣Processor")
(?cpuComp dbp:fastest "2.4"^^xsd:double)
(?cpuComp dbp:fastUnit <http://dbpedia.org/resource/GHZ>)
(?cpuComp omncomp:hasCores 6)
(?cpuComp dbp:arch <http://dbpedia.org/resource/X86-64>)
(?node omn:hasComponent ?cpuComp)
]
```

Having applied the above inference rules, a user may make a request for cloud resources with, for example, specific CPU requirements. In the sample SPARQL query provided in Listing 3, the user submits a request for two nodes (virtual machines) with a specific number of CPU cores and OS flavor (e.g., Fedora:6cores or Linux:6cores). The results are shown in Listing 4.

Listing 3: SPARQL query listing

```
SELECT ?resource1 ?resource2 WHERE {
?resource1 rdf:type omnres:Node .
?resource1 omnres:hasSliverType/omndpc:hasDiskImage/omndpc:
        hasDiskimageOS ?os1.
 FILTER (xsd:string(?os1) = "Fedora"^^xsd:string || xsd:string(?os1)= "
         Linux"^^xsd:string).
?resource1 omn:hasComponent ?cpuComp1.
    ?cpuComp1 rdf:type omncomp:CPU.
  ?cpuComp1 omncomp:hasCores ?cpuvalue1.FILTER (?cpuvalue1 = "6"
         ^^xsd:integer).
?resource2 rdf:type omnres:Node .
?resource2 omnres:hasSliverType/omndpc:hasDiskImage/omndpc:
        hasDiskimageOS ?os2.
 FILTER (xsd:string(?os2) = "Fedora"^^xsd:string ||xsd:string(?os2)= "
         Linux"^^xsd:string).
?resource2 omn:hasComponent ?cpuComp2.
    ?cpuComp2 rdf:type omncomp:CPU.
  ?cpuComp2 omncomp:hasCores ?cpuvalue2.FILTER (?cpuvalue2 = "6"
         ^^xsd:integer).
  FILTER (?resource1 != ?resource2)
} limit 1
```

Listing 4: Query results

```
RESULTS
urn:publicid:IDN+wall2.ilabt.iminds.be+node+n095-05a
urn:publicid:IDN+wall2.ilabt.iminds.be+node+n096-02
TIME EXECUTION:
0.016sec
```

VII. Conclusion and Future Work

The OMN ontology suite consists of an upper ontology and a set of eight descendant ontologies. Each ontology covers a specific aspect of infrastructure components and services. In this paper, the DBcloud system was introduced, which uses OMN to model cloud data, which in turn facilitates the process of resource lookup, matching and provisioning. The DBcloud system provides its data both via a public endpoint[4] and as data dumps. The resulting knowledge base contains more than 100 aggregates, 2,500 nodes, 6,700 links, and around 11,000 interfaces, constituting around 3.3 million triples. Furthermore, DBcloud mimics the design of the DBpedia system, i.e. it has a core extraction framework that produces triples corresponding to the whole resource life cycle, starting from resource advertisement and ending with resource release. Moreover, the DBcloud system incorporates a translator that converts the existing XML-based data into RDF-based data with the OMN ontology suite.

The DBcloud system utilizes Semantic Web technologies in order to improve the entire resource life cycle. In other words, various resources are described using the OMN ontologies, which ultimately improves the process of matching between a user request and the available resources. Since all data, including requested and free resources, is expressed in RDF, the matching process is very simple using SPARQL queries.

A short-term goal for future work is to support the publication of monitoring information, i.e. the information resulting from monitoring the provisioned resources along the various phases of the whole life cycle. This information is highly important as it helps to detect any problems with single or multiple resources, e.g. network link failure. It can further be used by the system administrator to monitor the entire provisioning process and consequently detect potential sources of problems that might hinder any future provisioning.

In the long run, the testbed data sets will be published to the LOD cloud, and interlinked with other data sets. Using the LOD cloud means that the metadata of resources is dramatically enriched. For instance, when network switch data is linked to the LOD cloud, the specifications registered for this switch will increase significantly, e.g. manufacturer or release date. Internet Topology Data Kit (ITDK)[5] is an example data set for potential interlinking.

VIII. Acknowledgments

This work is funded by the Dutch national program COMMIT, EU FP7 Fed4FIRE project and the US NSF GENI program.

References

[1] Carlos Buil Aranda et al. *SPARQL 1.1 Overview*. Recommendation. World Wide Web Consortium (W3C), Mar. 2013.

[4] http://lod.fed4fire.eu/sparql

[5] https://datahub.io/dataset/itdk

[2] Sören Auer et al. "DBpedia: A Nucleus for a Web of Open Data". In: *The Semantic Web (ISWC)*. Springer, 2007, pp. 722–735.

[3] Mark Berman et al. "GENI: A federated testbed for innovative network experiments". In: *Computer Networks* 61.3 (Mar. 2014), pp. 5–23.

[4] Tim Berners-Lee. *Linked data*. 2006.

[5] M Bjorklund. *RFC6020-YANG-A Data Modeling Language for the Network Configuration Protocol (NETCONF)*. Tech. rep. IETF RFC, October, 2010.[Links], 2010.

[6] Diego Valerio Camarda, Silvia Mazzini, and Alessandro Antonuccio. "LodLive, exploring the web of data". In: *8th International Conference on Semantic Systems (I-SEMANTICS)*. New York, New York, USA: ACM Press, Sept. 2012, p. 197.

[7] Jorge Cardoso et al. "Towards a Unified Service Description Language for the Internet of Services: Requirements and First Developments." In: *IEEE SCC*. IEEE Computer Society, 2010, pp. 602–609.

[8] Richard Cyganiak, David Wood, and Markus Lanthaler. *Resource Description Framework (RDF) 1.1 Concepts and Abstract Syntax*. Recommendation. World Wide Web Consortium (W3C), Feb. 2014.

[9] Amir Vahid Dastjerdi, Sayed Gholam Hassan Tabatabaei, and Rajkumar Buyya. "An Effective Architecture for Automated Appliance Management System Applying Ontology-Based Cloud Discovery". In: *10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid), 2010*. IEEE. 2010.

[10] Anastasius Gavras et al. "Future Internet Research and Experimentation: The FIRE Initiative". In: *ACM SIGCOMM Computer Communication Review* 37.3 (2007), pp. 89–92.

[11] Steffen Haak and Stephan Grimm. "Towards Custom Cloud Services – Using Semantic Technology to Optimize Resource Configuration". In: *Proceedings of the 8th Extended Semantic Web Conference, ESWC 2011*. 2011.

[12] Peter Haase et al. "Semantic Technologies for Enterprise Cloud Management". In: *International Semantic Web Conference*. 2010.

[13] Jerry R. Hobbs and Feng Pan. *Time Ontology in OWL*. Tech. rep. Sept. 2006.

[14] Martin Junghans, Sudhir Agarwal, and Rudi Studer. "Towards Practical Semantic Web Service Discovery". In: *ESWC (2)*. Vol. 6089. Lecture Notes in Computer Science. Springer, June 8, 2010, pp. 15–29.

[15] Jens Lehmann et al. "DBpedia – A large-scale, multilingual knowledge base extracted from wikipedia". In: *Semantic Web Journal (SWJ)* 5.1 (2014), pp. 1–29.

[16] Pablo Mendes, Max Jakob, and Christian Bizer. "DBpedia: A Multilingual Cross-domain Knowledge Base". In: *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. Ed. by Nicoletta Calzolari (Conference Chair) et al. Istanbul, Turkey: European Language Resources Association (ELRA), May 2012.

[17] Le Duy Ngan and Rajaraman Kanagasabai. "OWL-S Based Semantic Cloud Service Broker". In: *2012 IEEE 19th International Conference on Web Services, Honolulu, HI, USA, June 24-29, 2012*. 2012.

[18] Daniel Oberle et al. "A unified description language for human to automated services." In: *Inf. Syst.* 38.1 (2013), pp. 155–181.

[19] OWL Working Group and others. *W.: OWL 2 Web Ontology Language: Document Overview. W3C Recommendation (27 October 2009)*. 2012.

[20] Derek Palma and Thomas Spatzier. *Topology and Orchestration Specification for Cloud Applications (TOSCA) Version 1*. Nov. 2013.

[21] Carlos Pedrinaci, Jorge Cardoso, and Torsten Leidig. "Linked USDL: A Vocabulary for Web-Scale Service Trading." In: *ESWC*. Ed. by Valentina Presutti et al. Vol. 8465. Lecture Notes in Computer Science. Springer, 2014, pp. 68–82.

[22] Larry Peterson et al. *Slice-based Federation architecture*. Draft version. GENI, 2009.

[23] Idafen Santana-Pérez and Maria S Perez-Hernández. "A semantic scheduler architecture for federated hybrid clouds". In: *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*. IEEE. 2012.

[24] Michael Stollberg et al. "Two-Phase Web Service Discovery Based on Rich Functional Descriptions". In: *Proceedings of the 4th European Semantic Web Conference (ESWC)*. Ed. by Enrico Franconi, Michael Kifer, and Wolfgang May. Vol. 4519. Lecture Notes in Computer Science. Springer, 2007, pp. 99–113.

[25] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. "YAGO: A core of semantic knowledge. Unifying WordNet and Wikipedia". In: *Proceedings of the 16th international conference on World Wide Web*. ACM. 2007, pp. 697–706.

[26] Wim Vandenberghe et al. "Architecture for the Heterogeneous Federation of Future Internet Experimentation Facilities". In: *Future Network and Mobile Summit (FNMS)*. Lisboa, Portugal: IEEE, 2013, pp. 1–11.

[27] Alexander Willner et al. "The Open-Multinet Upper Ontology Towards the Semantic-based Management of Federated Infrastructures". In: *10th EAI International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities (TRIDENTCOM)*. Vancouver, Canada: ACM, Aug. 2015, p. 10.