SemNaaS: Add Semantic Dimension to the Network as a Service

Mohamed Morsey[1(✉)], Hao Zhu[1], Isart Canyameres[2], and Paola Grosso[1]

[1] Informatics Institute, University of Amsterdam,
1098 XH Amsterdam, The Netherlands
{m.morsey,h.zhu,p.grosso}@uva.nl
[2] i2CAT Foundation, 08034 Barcelona, Spain
isart.canyameres@i2cat.net

Abstract. Cloud Computing has several provision models, e.g. Infrastructure as a service (IaaS). However, cloud users (tenants) have limited or no control over the underlying network resources. Network as a Service (NaaS) is emerging as a novel model to fill this gap. However, NaaS requires an approach capable of modeling the underlying network resources capabilities in abstracted and vendor-independent form. In this paper we elaborate on SemNaaS, a Semantic Web based approach for developing and supporting operations of NaaS systems. SemNaaS can work with any NaaS provider. We integrated it with the existing OpenNaaS framework. We propose Network Markup Language (NML) as the ontology for describing networking infrastructures.Based on that ontology, we develop a network modeling system and integrate with OpenNaaS. Furthermore, we demonstrate the capabilities that Semantic Web can add to the NaaS paradigm by applying SemNaaS operations to a specific NaaS use case.

1 Introduction

Cloud infrastructures and services are becoming the de facto architecture to support operations of large web infrastructures, to process Big Data and to provide users with ubiquitous and scalable computing and storage services. However, still there are many less explored challenges for an optimal operation: one of them being the delivery of tailored network services. The underlying network connecting (cloud) data center or within a single site is still less malleable and programmable than the other parts of the infrastructure. New frameworks are emerging to define and create such dynamic network services; they effectively provide easy APIs to define services at the network level. These frameworks in essence support Network as a Service (NaaS) operations.

The emerging NaaS software systems require powerful and rich vocabularies, such as the ones that can be provided by Semantic Web ontologies. Those vocabularies should hide the network implementation details from the user. Thus NaaS users can fully express their needs in terms of services, without having to know the details of the physical network devices. OWL ontologies have several

F. Gandon et al. (Eds.): ESWC 2015, LNCS 9341, pp. 107–111, 2015.
DOI: 10.1007/978-3-319-25639-9_21

advantages as models for NaaS; i.e. they are easy to extend, they allow for automatic validation of both requests and provisioned services, and they enhance network resource discovery.

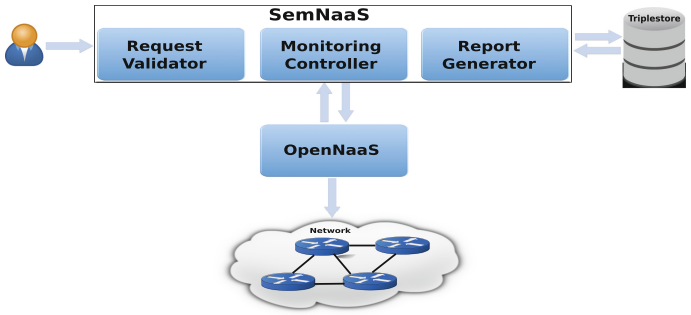

Fig. 1. The SemNaaS general system architecture.

2 SemNaaS Approach

SemNaaS[1] adds several potential improvements to OpenNaaS [2], namely (1) support of network request validation and network connectivity check; (2) system monitoring and failure condition detection; (3) capability of constructing distributed and interconnected OpenNaaS instances; (4) complex report generation.

2.1 Ontology

The ontology is at the core of the SemNaaS system, as it constitutes the main concepts required to create, define and link the various network nodes. We have reused and improved the Network Markup Language (NML) ontology [5]. NML constitutes the information model for describing and defining computer networks. In order to use NML in NaaS, we have devised more classes and properties, and enhanced the existing ones as well, in order to permit the creation of complex network topologies[2].

2.2 SemNaaS Architecture

The SemNaaS architecture is illustrated in Fig. 1. It consists basically of four components, which are (1) request validation and connectivity checking component; (2) OpenNaaS component, which is a pluggable component, that supports the network resources provisioning; (3) monitoring component; (4) report generation component.

[1] The source code is available at http://github.com/dana-i2cat/semnaas.
[2] The ontology is available at http://bitbucket.org/uva-sne/indl/src.

Request Generation and Validation. SemNaaS performs two levels of validation, namely request validation, and connectivity check. SemNaaS uses the Pellet reasoner [4] to validate the request against the NML ontology. In other words, it checks that the request adheres to the NML vocabulary, e.g. no component is declared as a port and as link at the same time, as those concepts are disjoint. The second level of validation, is the connectivity validation, in which SemNaaS utilizes SPARQL [1] to validate the request and detect if a network node is unreachable from another node. After a request is validated, it is passed to OpenNaaS for provisioning, and also its RDF data is stored in Virtuoso triplestore [3] for reporting.

Monitoring Component. Once the request is sent, OpenNaaS determines whether the requested resources can be granted immediately or the user should wait till those resources can be granted. During resource operation, it passes through several states, e.g. active or inactive. However, the resource may experience failure conditions as well, e.g. network connectivity failure. Whenever a change occurs in the resource status, SemNaaS tracks that change, thus it always reflects the most recent status of OpenNaaS resources.

Report Generation Component. SemNaaS empowers OpenNaaS to generate complex reports about the whole resource reservation process. Those reports enable the system administrator to identify the problematic resources of OpenNaaS, and monitor the resource failure conditions. Furthermore, a system administrator can view all network resources which have been reserved at a certain point of time and still active, in order to release them, and allow reallocation of them to other users.

Interconnection of Distributed NaaS Instances. One major challenge OpenNaaS has faced was how to maintain the uniqueness of the IDs assigned to the various network components. In other words, each network component, e.g. network node, has an ID that is only unique within the boundaries of the OpenNaaS instance in which it resides, e.g. host with ID "host1". Since we use the NML ontology to describe networks provisioned by OpenNaaS, we should assign a unique URI to each component. For example, http://ivi.fnwi.uva.nl/sne/resource/host1, and http://www.i2cat.net/resource/host1, are URIs for "host1" located in two remote NaaS instances.

3 Evaluation and Use Case

The best way to show how SemNaaS improves the functionality of network resource provisioning and path finding of OpenNaaS is to follow the system operations in two use cases: user request processing and implementation of Virtual Routing Function. This request passes several phases to get fulfilled: specifically (1) the user sends a request, formulated in NML, with the required details, e.g. source and destination nodes, as illustrated in Fig. 2(a); (2) SemNaaS receives the request, and validates it; (3) SemNaaS forwards the path finding request to

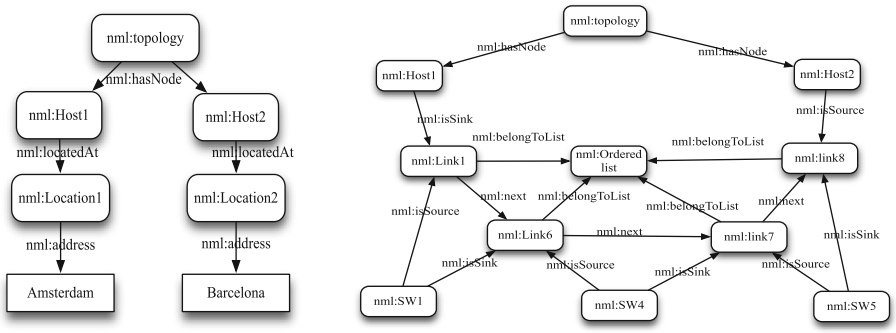

Fig. 2. An example request (a), and reply (b) in the virtual routing function use case.

OpenNaaS for fulfillment; (4) after OpenNaaS calculates the path between nodes, SemNaaS formulates the output response in NML as well.The Virtual Routing Function use case aims to implement inter-domain routing through OpenNaaS. The implementation of VRF is over an Openflow infrastructure. It extends the local routing of the Openflow controllers. VRF finds out a routing path according to the routing mode such as static, Dijkstra, and then invokes the controllers to dynamically create flow tables of the switches on the path. SemNaaS uses NML to describe resources in the request and reply of VRF. Figure 2(a) shows an example of a possible basic request in the VRF use case. The request contains a start host and a destination host, each one has different properties, e.g. IP address. Once the routing path is calculated and created, an understandable reply from OpenNaaS components is also important for user. As Fig. 2(b) shows the reply is described by NML as a routing path composed of a set of links and switches.

4 Conclusions

SemNaaS fuses Semantic Web with NaaS to develop an intelligent NaaS system, which improves NaaS systems with impressive features; e.g. connectivity checking.

References

1. SPARQL 1.1 Query Language. Technical report, W3C (2013)
2. Aznar, J.I., Jara, M., Rosello, A., Wilson, D., Figuerola, S.: OpenNaaS based management solution for inter-data centers connectivity. In: IEEE 5th International Conference on Cloud Computing Technology and Science, CloudCom, Bristol, United Kingdom (2013)
3. Erling, O., Mikhailov, I.: RDF Support in the virtuoso DBMS. In: Conference on social semantic web, vol. 113 of LNI, GI (2007)

4. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: a practical OWL-DL reasoner. Web Semant. Sci. Serv. Agents World Wide Web **5**(2), 51–53 (2007)
5. van der Ham, J., Dijkstra, F., Łapacz, R., Zurawski, J.: The network markup language (NML) a standardized network topology abstraction for inter-domain and cross-layer network applications. In: Proceedings of the 13th Terena Networking Conference (2013)