On Network Service Modeling in Softwarized Networks

Nathan F. Saraiva de Sousa, Christian Esteve Rothenberg (Orientador)
Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)
Caixa Postal 6101, 13083-970 – Campinas, SP, Brasil
{nsaraiva, chesteve}@dca.fee.unicamp.br

Abstract – The Software Defined Network (SDN) and Network Function Virtualization (NFV) architectures are composed of different functional blocks such as NFV Orchestrator, Virtualized Infrastructure Manager (VIM) and SDN controllers. These components are modeled as independent software with specific access, configuration, and management. They are connected mainly across of the Application Programming Interfaces (APIs) that each component provides. A critical challenge is an integration between these diverse components towards effectively deploy resource requirements, configuration parameters, management policies and performance metrics of network services. This integration is possible by definition of standard service modeling language. This work presents an overview of network service modeling on SDN/NFV domain, including the description and integration of modeling languages and template TOSCA, YANG and HOT. Besides, the paper compares two orchestration frameworks in term of service modeling. This paper serves as an initial point in the research about service modeling.

Keywords – NFV, modeling language, TOSCA, YANG, HOT.

1. Introduction

The Software Defined Network (SDN) and Network Function Virtualization (NFV) architectures orchestrate resources of compute, storage and network in complex environments. These architectures are composed of functional blocks with specific access, configuration and management and connected across of Application Programming Interfaces (APIs) [1]. The need for a seamless integration between the components suggests the definition of a resource and service modeling.

The modeling simplifies the understanding of functions and provides a generic way to represent resource and service. There are some template and data modeling for Virtual Network Function (VNF) and Network Services (NS) such as TOSCA, YANG, and HOT. The standardizing can enable interoperability of network services templates and addresses limitations arising in the deployment of services in heterogeneous landscape.

This paper introduces the problem statement of service modeling and interoperability between different solutions in the context of Network Function Virtualization. An integration of TOSCA, YANG, and HOT is proposed in the NFV architecture. We also define an use case and compare the network service modeling between two NFV orchestration solutions: OpenStack Tacker\(^1\) and OpenBaton\(^2\).

The work is organized as follows: Section 2. presents an overview of the service modeling related to virtual network functions and network services. As a Section 3., we propose a solution to integrate TOSCA, YANG and HOT in the NFV architecture. The Section 4. focuses on an use case that compares the service modeling between Tacker and OpenBaton using TOSCA language. Finally, we conclude the paper and present future works in the Section 5.

2. Service Modeling

One of the critical challenges, in order to obtain all Network Service Orchestration’s potential, is the development of models which catch the resource requirements, capabilities, optimization metrics and lifecycle of network services. In addition, the resources and services in SDN/NFV scenarios are provided by different components. Therefore, the standardization of service modeling is necessary towards creating well understood and open descriptors which enable the interoperability between entities and architectures.

Possible implementations of the descriptor are YAML, XML, JSON, using HOT, YANG and/or TOSCA modeling languages. Below we detail the main modeling languages adopted actually in SDN/NFV architectures and frameworks. In the next section, we identify the scope of each language in the NFV-Management and Orchestration(MANO) ETSI architecture.

\(^1\)https://wiki.openstack.org/wiki/Tacker
\(^2\)http://openbaton.github.io
2.1. TOSCA

TOSCA (Topology and Orchestration Specification for Cloud Applications) [2] is a language to describe a topology of cloud, their components, and relationships. It is interpreted by engines (e.g., ARIA) to realize lifecycle workflows. Besides, TOSCA addresses the automation of the application deployment and lifecycle management. TOSCA may be used in NFV domain for VNF definition, node monitoring and active policies like healing and scaling.

The TOSCA NFV profile specifies an NFV data model using TOSCA language. The deployment and operational behavior of each NS in NFV are defined in a service template and stored (on-boarding process) in a catalogue, for future instantiation. This profile defines the NFV specific types to fulfill the NFV requirements [2]. It is called network service descriptor (NSD) and describes a relationship between network functions (virtual and/or physical). A “service” for TOSCA is an application or application component, such as a load balancer, database or virtual switch. In a networking definition, a “service” defines the configuration of a network function, for example, Layer 3 Virtual Private Network (VPN) [3].

The TOSCA template consists basically of two parts:

- **Topology template**, which details the network functions and artifacts required to deploy them. Artifacts can be images, scripts, and applications.
- **Plans**, or workflows that define the process of deploying a VNF or NS, including the execution a bunch of tasks in correct order. The TOSCA template allows plans to be created in multiple workflow languages, e.g. BPMN, BPEL, Chef, Puppet.

Furthermore, the TOSCA template and its artifacts can be put in a CSAR (Cloud Service ARchive) zipped file and after on-boarded in a production environment. The TOSCA simple profile has a number of base types (node types and relationship types) such as compute, network, and database. It allows defining new types and extends the primitive types. This provides a large number of additional types defined by the community.

Many orchestration frameworks such as OpenStack Tacker, Open-O, Cloudify, and Open Baton have adopted the TOSCA like standard service modeling language. In contrast, there is no a standardization to create the artifacts to TOSCA templates. As a result, the interoperability of templates across vendors and orchestration solutions is limited.

2.2. YANG

YANG (RFC 6020) is an IETF data modeling language that allows the configuration and controls the state of the network elements. YANG is used to model the operations and content layers of NETCONF [4] and aims to be the successor of Simple Network Management Protocol (SNMP). YANG models could be applied in NFV applications and deployments mainly in the configuration of devices.

NETCONF protocol defines the rules to configure the network devices. NETCONF uses well-known technologies such as RPC and XML. It gains importance in NFV domain, where broader solutions to deploy NS and VNF is needed. Besides, it can act as an element management system (EMS) in the NFV architecture and provide a way of sharing information between the OSSes and the network devices. Furthermore, YANG can be used to define the format of event notifications emitted by network elements, virtual or physical, and this can define an action to be invoked on network elements via the NETCONF protocol.

2.3. HOT

Heat Orchestration Templates (HOT) define a topology infrastructure for a Cloud application, using a readable and writable way of representation. Inside the HOT are also defined policies, specific configuration (e.g. CPU and memory) and relationships between resources. The OpenStack/HEAT orchestration engine orchestrates the actions defined on HOT templates in the correct order.

HOT templates are usually defined in YAML. It describes and automates the deployment of cloud resources and services exclusively in OpenStack infrastructure. This is a limitation in terms of portability, however, the HEAT can translate from TOSCA template to HOT.

3. Integration TOSCA, YANG and HOT

Operation/Business Support System (OSS/BSS) is not tightly integrated into the NFV-MANO architecture. There is no a standard modeling in reference point between OSS/BSS and the MANO block.
Different modeling language and template can help in the process of integration between the diverse components of the NFV architecture. TOSCA, NETCONF/YANG, and HOT can work together in distinct steps and layers of the service life-cycle management process. The Figure 1 shows where each template can act in NFV stack.

A TOSCA template can trigger NETCONF/YANG and HOT configuration during the instantiation of an NS or VNF. An advantage of NETCONF/YANG is that the same approach can be used for runtime configuration of both virtual and physical network functions. Meanwhile, the HOT template is charge of virtualizing the compute, storage and network. Once the VNF instanced, the OSS can work with VNFs or their corresponding EMSs to configure the VNFs at runtime using NETCONF/YANG [3].

4. Use Case

4.1. Description

We deploy an use case in order to compare the interoperability between network service descriptors using the same modeling language and in different orchestration platforms. The use case consists of a network service composed by two VNFs interconnected across a public interface. The Figure 2 describes the network service.

The chosen orchestration platforms were OpenStack Tacker and OpenBaton. Tacker [5] is an official OpenStack project building a Generic VNFM and a NFVO to deploy and operate Network Services and Virtual Network Functions (VNFs) on an NFV infrastructure platform like OpenStack. It is based on ETSI MANO Architecture and provides a functional stack to Orchestrate Network Services end-to-end using VNFs.

OpenBaton [6] is platform complaint with NFV-MANO ETSI launched by Fraunhofer FOKUS and TU Berlin. It implements an NFV Orchestrator and a generic VNF Manager but is capable of interoperating with VNFMs implemented by third parties.

Both solutions support TOSCA templates and have already stable release. Besides, they have a large documentation. The platforms were installed in virtual machines with 8 GB RAM and 4 CPUs.

The objectives are to identify the difference between the descriptors of the two solutions and point the limitations to each descriptor.

4.2. Modeling

The Tacker and OpenBaton follow the TOSCA Simple Profile for Network Functions Virtualization (NFV) Version 1.0. The Tacker used the OpenStack as VIM and the OpenBaton used a VIM of test, installed by default.

Initially, we identify the involved components in the composition of network service. The NS has the following components:

- Two virtual network functions named VNF1 and VNF2.
- Each VNF has one virtual deployment unit (VDU1 and VDU2).
- Each VDU has one connection point (CP1 and CP2).
- There are two virtual links (VL1 and VL2) connecting the VDUs to the public interface.

The next step is to create the full NSD. Each descriptor has many lines, so we presented in the
Figure 3 only the part reference to VDU1 of each descriptor.

![Figure 3. Comparative between the Tacker and OpenBaton descriptors.](image)

In order to onboard the descriptor, the way was different for each platform. In the Tacker was used its CLI and onboarded two VNFDs and after the NSD. In the OpenBaton was necessary to create a CSAR (Cloud Service Archive) file with all the descriptors and file, after that the dashboard was used to onboard the file.

The VNFs and NSDs were validated and instanced in each platform. The complete NSD are available in the URL: https://goo.gl/wFZKkm.

4.3. Analysis

During the analysis of the descriptors created for each platform, we identify differences between them, even they have used the same version of TOSCA. The Tacker and OpenBaton can onboard the NSD using CLI or Dashboard.

The OpenBaton have some limitations in its template such as:

- *Topology_Template* includes only the component *Node_Templates*, not implement the *substitution_mappings*, *input* and *output*;
- *Interfaces* field has only one option at the moment: *lifecycle* (instantiate, configure, start, stop, and terminate);
- Only one type of image is supported (*tosca.artifacts.Deployment.Image.VM*);
- The option of import new descriptors is not supported;

As for Tacker, some limitations were found:

- *In topology_Template* does not include *substitution_mappings* and *output*;
- It uses NSD and VNFD in separate files. This is a limitation to scale and to complex services.
- Only works in OpenStack Cloud.

Besides, the Tacker and OpenStack descriptors have fields that there are no or are different in each solution. For example, the fields *endpoint*, *vnf-PakageLocation*, and *deploymentFlavours* only belong to the OpenBaton.

5. Conclusion

SDN/NFV solutions are responsible for orchestrating complex and heterogeneous environments composed of multi-domain and multi technologies. In this scenario, there are many challenges such as end-to-end automated management, security, resource and service modeling, and interoperability.

The work currently focused on an initial analysis of service modeling in the NFV domain and summarized the main templates and data modeling for Network Service. We presented a proposal of integrating of the TOSCA, YANG, and HOT in the NFV stack.

We also presented an use case with the objective of identifying the difference between service descriptor of two orchestration solution: Tacker and OpenBaton. We note that there are significant differences between the solutions even using the same modeling language and this difficult the interoperability. There is no doubt that is necessary a standardization process towards enabling interoperability of network services templates.

References