VNF Orchestration and Modeling with ETSI MANO Compliant Frameworks

Ales Komarek, Jakub Pavlik, Lubos Mercl, and Vladimir Sobeslav(✉)

Faculty of Informatics and Management, University of Hradec Kralove,
Rokitanskeho 62, Hradec Kralove, Czech Republic
{ales.komarek,jakub.pavlik.7,lubos.mercl,vladimir.sobeslav}@uhk.cz

Abstract. The goal of this paper is to compare existing NFV (Network Function Virtualization) management solutions and propose SaltStack based solution with the same goal. Modern approach to govern NFV automation in large scale IT infrastructures is done by virtualization of various network services. Hardware networking devices are gradually replaced by virtual network appliances for the lower acquisition and maintenance costs. Vendors virtualize their physical products so they can be used along or within cloud environments. The ETSI (European Telecommunications Standards Institute) standard MANO (NFV Management and Organization) aims to unify the management of physical and virtual network services and devices to single point of control for configuration and management.

The paper covers the major NFV frameworks that use MANO as their reference architecture with focus on underlying interface technologies. The main focus of the paper is NFV-MANO orchestration engine using SaltStack capabilities that can model and manage heterogeneous NFV and NFVI resources according to the ETSI NFV-MANO specification along software resources already present. This gives us unique ability to cover all data center resources, both hardware or software based in a single model.

Keywords: NFV · SDN · NFV-MANO · Cloud service · SaltStack

1 Introduction

The NVF frameworks leverage cloud technologies and network virtualization to offer services while reducing Capital and Operations expenditures and can achieve significant levels of operational automation. [7,11,14] Cloud technologies are being adopted, focusing primarily on information technology and dynamic applications. The cloud technology provides the ability to manage diverse workloads and applications dynamically [14]. The desired cloud capabilities are:

1. Real-time installation and decommission of Virtual Machines (VMs),
2. Dynamic assignment of applications and workloads to the running VMs,

O. Galinina et al. (Eds.): NEW2AN/ruSMART/NsCC 2017, LNCS 10531, pp. 121–131, 2017.
DOI: 10.1007/978-3-319-67380-6_11

3. Dynamic movement of applications and dependent functions to different VMs on servers within or across data centers in different geographical locations.

Efforts have not been underway to virtualize only compute resources but also network functions that had been realized in purpose-built appliances, specialized hardware and software (e.g., routers, firewalls, switches). Network function virtualization (NFV) is focused on transforming these appliances into software [18].

Companies are implementing the metadata driven application and service catalogs by centralizing the creation of models, rules, and policies [14,18]. All orchestrator applications and controllers ingest the metadata that governs their behavior from centralized store. Implementation of the metadata-driven methodology requires an enterprise-wide paradigm change of the development and delivery process. It demands a service agreement on the overall metadata-model across the business (product development) and the software developers writing code for orchestrator, business support system and operations support system. This agreement is a behavioral contract to permit both the detailed business analysis and the construction of software to run in parallel.

The software development teams focus on building service-independent software within a common operations framework for runtime automation. The Business teams can focus on tackling business needs by defining metadata models to feed the execution environment [14]. The metadata model content itself is the glue between design time and runtime execution frameworks, and the result is that the service specific metadata models drive the common service independent software for runtime execution.

Models are created and managed centrally so that all components, both simple and complex, are easily visualized and understood together, and validated properly prior to use. Once validated and corrected for any conflicts, the models are precisely distributed to the many points of usage. This improves scalability and reduces latency of delivery [6]. Models are created by many user groups (e.g., service designers, security engineers, operations staff, etc.). Various techniques are used to validate newly created models and policies and to help identify and solve any real- world problems [15].

2 NFV Orchestration Components

The European Telecommunications Standards Institute (ETSI) developed a Reference Architecture Framework and specifications in support of NFV Management and Orchestration (MANO) [3,7]. The main components of the ETSI-NFV architecture are: Orchestrator, VNF Manager, and VI (Virtualized Infrastructure) Manager [4]. The Metadata modeling tool is the most important external part of the MANO infrastructure [2] (Fig. 1).

2.1 NFV Modeling

The modeling tools deliver product/service independent capabilities for the design, creation and lifecycle management of the target environments. Model

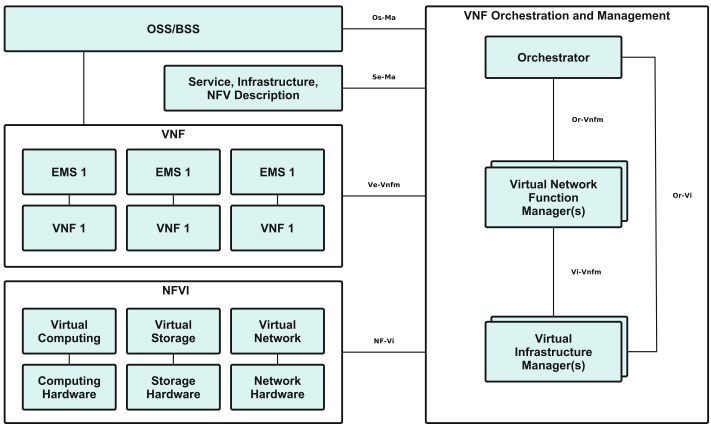

Fig. 1. ETSI MANO

design platforms are used to define and program the platform, and an execution time environment to execute the logic programmed in the design phase utilizing closed-loop, policy-driven automation [4,7]. The metadata model is managed through a design environment which guides a series of designers from 3rd party resource vendor onboarding through service creation, verification, and distribution. The modular nature of the metadata model provides a catalog of patterns that can be reused in future services. This provides approach to quickly extend resource functions to manageable services and sellable products, further realizing benefits in time to market. Orchestrator component supports companys metadata model design. Many different models of metadata exist to address different business and technology domains. Metadata modellers integrate various tools supporting multiple types of data input (e.g., YANG, HEAT, TOSCA, YAML, BPMN/BPEL, etc.). It automates the formation of an company internal metadata format to drive end-to-end runtime execution. In modeller the resource metadata is created in the description of the cloud infrastructure and the configuration data attributes to support the implementations. Subsequently, the resource metadata descriptions become a pool of building blocks that may be combined in developing services, and the combined service models to form products. In addition to the description of the object itself, modeling the management needs of an object using metadata patterns in modeller may be applied to almost any business and operations functions in company. For example, metadata can be used to describe the mapping of resource attributes to relational tables, through the definition of rules around the runtime session management of a group of related resources to the signatures of services offered by a particular type of resources.

2.2 NFV Orchestrator

Orchestrator expands the scope of ETSI MANO coverage by including Controller and Policy components. Policy plays an important role to control and manage behavior of various VNFs and the management framework. Orchestrator also significantly increases the scope of ETSI MANOs resource description to include complete meta-data for lifecycle management of the virtual environment (Infrastructure as well as VNFs).

2.3 NFV Manager

Network functions virtualization (NFV) defines standards for compute, storage, and networking resources that can be used to build virtualized network functions. The virtual network functions manager (VNFM) is a key component of the NFV management and organization (MANO) architectural framework.

The VNFM works with other MANO functional blocks, such as the virtualized infrastructure manager (VIM) and NFV orchestrator (NFVO), to help standardize the functions of virtual networking and increase the interoperability of software-defined networking elements. These standard components can help lower costs and increase new feature deployment and velocity by providing a standard framework for building NFV applications. In the virtualized network world of NFV, scalability and speed of service deployments create new challenges in the areas of network management and orchestration.

2.4 Virtualized Infrastructure Manager (VIM)

The virtualized infrastructure manager (VIM) is last key component of the MANO architectural framework. It is responsible for controlling and managing the NFV infrastructure (NFVI) compute, storage, and network resources, usually within operator's cloud based infrastructure. These functional blocks help standardize the functions of virtual networking to increase interoperability of software-defined networking elements. VIMs can also handle hardware in a multi-domain environment or may be optimized for a specific NFVI environment.

The VIM is responsible for managing the virtualized infrastructure of an NFV- based solution. VIM operations include:

1. Keeps an inventory of the allocation of virtual resources to physical resources. This allows the VIM to orchestrate the allocation, upgrade, release, and reclamation of NFVI resources and optimize their use.
2. Supports the management of VNF forwarding graphs by organizing virtual links, networks, subnets, and ports. The VIM also manages security group policies to ensure access control.
3. Manages a repository of NFVI hardware resources (compute, storage, networking) and software resources (hypervisors), along with the discovery of the capabilities and features to optimize the use of such resources.

3 Existing NFV Orchestration Platforms

Following chapter summarizes the most usual frameworks for VNF management according to ETSI MANO [7,8,12]. The ETSI NFV-MANO standard defines high-level modle, but lacks details and standards on the implementation for both managers as well as the interfaces. As a result some of the covered solutions are in fact customized with some parts based on proprietary solutions.

3.1 Open Source MANO

Open source implementation with RIFR.ware as orchestrators and OpenMANO and OpenVIM as VNF and NFVI resources [8,17]. RIFT.ware, developed by RIFT.io is the industrys first open source, NSV platform. As discussed in the previous section, network functions need to incorporate the practices of hyper-scale data centers to meaningfully reduce cost and deliver agile, dynamic service provisioning. As many of these practices are common across all network functions, it follows that the data center platform needs to follow the evolutionary trend of the past decades by incorporating new, web-scale techniques. RIFT.ware redefines the networking platform by combining functions expected in legacy hardware with the technology of cloud infrastructure (Fig. 2).

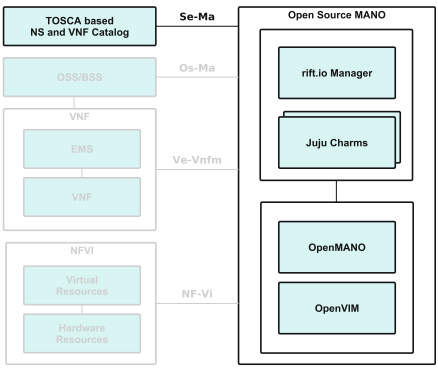

Fig. 2. OpenSource MANO

To bring hyper-scale capabilities and economics to cloud and network function builders, service providers, and enterprises, RIFT.ware simplifies the development, deployment, and management of cloud-scale network applications and VNFs. The OpenMANO is a reference implementation of an NFV-O (Network Functions Virtualization Orchestrator) [8]. It interfaces with an NFV VIM through its API and offers a northbound interface, based on REST (OpenMANO API), where NFV services are offered including the creation and deletion of VNF templates, VNF instances, network service templates and network service instances.

The OpenVIM is reference implementation of an NFV VIM (Virtualized Infrastructure Manager). It interfaces with the compute nodes in the NFV Infrastructure and an openflow controller in order to provide computing and networking capabilities and to deploy virtual machines. It offers a northbound interface, based on REST (OpenVIM API), where enhanced cloud services are offered including the creation, deletion and management of images, flavors, instances and networks. The implementation follows the recommendations in NFV- PER001.

3.2 OPEN-Orchestrator

GS-O is responsible for Global Service Orchestration to provide End-to-End orchestration of any service on any network [16]. GS-O allows the user to describe the end-to-end (global) service in a single Global Service Description. GS-O decomposes the Global Service Description to SDN Service Descriptions and NFV Service Descriptions, and then forwards these requests to the correct instances of SDN-O and NFV-O.

This NFV-O function is based on ETSI NFV MANO architecture & information model as a reference [16]. It orchestrates Network Services composed of Virtualized Network Functions and is an integral part of OPEN-O. The NFV-O function includes Network service life cycle manager, NFV resource manager, and NFV monitor. Using the south bound SBI, NFVO cooperates with several NFV SDN controllers, Multi-vendor VNFMs and different VIMs.

3.3 Cloudify

The Blueprint Composer is an editor for composing blueprint YAML files dynamically using a modern Drag and Drop Interface [5]. Cloudify's Manager comprises of Cloudify's code and a set of Open-Source applications. An elaborate explanation on these applications is provided here (Fig. 3).

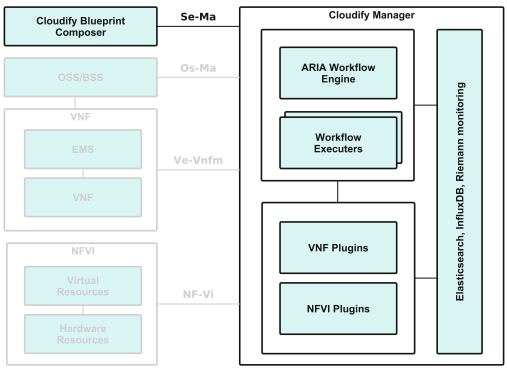

Fig. 3. Cloudify architecture

The way Cloudify users use Cloudify to deploy their application is by utilizing the power of blueprints. Blueprints are written in YAML format and their DSL is based on the TOSCA standard.

3.4 AT&T ECOMP

Capabilities are provided using two major architectural frameworks: (1) a Design Time Framework to design, define and program the platform (uniform onboarding), and (2) a Runtime Execution Framework to execute the logic programmed in the design environment (uniform delivery and lifecycle management) [1,10]. The platform delivers an integrated information model based on the VNF package to express the characteristics and behavior of these resources in the Design Time Framework. The information model is utilized by Runtime Execution Framework to manage the lifecycle of the VNFs. The management processes are orchestrated across various modules of ECOMP to instantiate, configure, scale, monitor, and reconfigure the VNFs using a set of standard APIs provided by the VNF developers (Fig. 4).

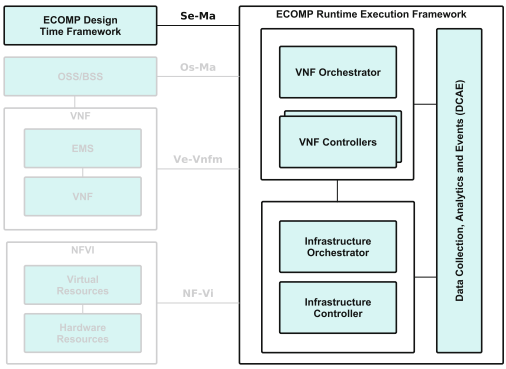

Fig. 4. AT&T ECOMP architecture

4 SaltStack Based NFV and VIF Orchestrator

In previous chapters we have covered the core components of the ETSI MANO standard and the most important implementations to the date. This chapter shows our proposed SaltStack based platform that govern resources according to the ETSI NFV-MANO model. The combination of orchestration and configuration features puts SaltStack in unique position to model, manage, and orchestrate various software stacks, network or storage devices and other infrastructure resources (Fig. 5).

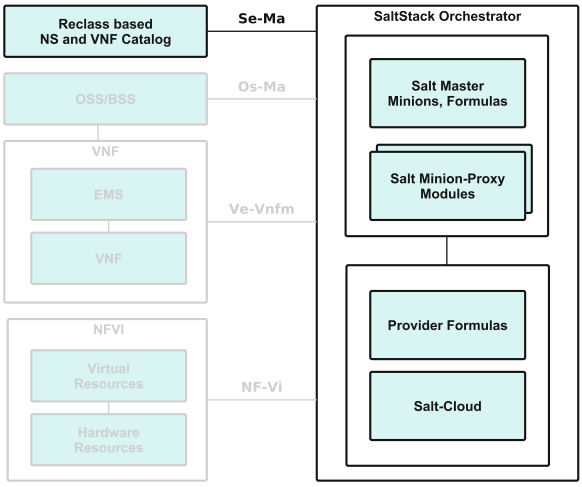

Fig. 5. SaltStack NVF-MANO orchestrator

4.1 NFV Modeller

The modeller delivers design models that are used to define and program the platform. Our modeller delivers ID centric system with simple classification using Reclass hierarchical database. All resources, both hardware and software are modeled the same way. The result model is simple data structure that can be interpreted by the Orchestrator.

4.2 NFV Orchestrator

Orchestrator is the central point of ETSI MANO infrastructure. SaltStack project began as a high-speed remote execution and communication bus designed to be used as a lightweight private cloud controller. Now it's full scale orchestration platform with configuration management capabilities. It uses modules to carry on the command execution and state layer with declarative resource description.

One of fundamental features of SaltStack is remote execution. Salt has two basic "channels" for communicating with minions. Each channel requires a client (minion) and a server (master) implementation to work within SaltStack. These pairs of channels will work together to implement the specific message passing required by the channel interface.

4.3 NFV Manager

SaltStack uses NAPALM (Network Automation and Programmability Abstraction Layer with Multivendor support) Python library that implements a set of functions to interact with different network device Operating Systems using a

unified API. It supports Arista EOS, Cisco IOS, Cisco IOS-XR, Cisco NX-OS, Fortinet Fortios, IBM, Juniper JunOS, Mikrotik RouterOS, Palo Alto NOS, Pluribus, and Vyos network operating systems. YANG (RFC6020) is a data modeling language, its a way of defining how data is supposed to look like. The napalm-yang library provides a framework to use models defined with YANG in the context of network management. It provides mechanisms to transform native data/config into YANG and vice versa. We use dedicated minion to handle network and storage devices by Salt Minion Proxy service, basically wrapping each external resource by NAPALM or other service wrapper.

4.4 Virtualized Infrastructure Manager (VIM)

Salt provides powerful interfaces to interact with cloud providers. These interfaces are tightly integrated with Salt, and new virtual machines are automatically connected to your Salt master after creation. The supported cloud environments include Scaleway Rackspace, Amazon AWS, Linode, Joyent Cloud, GoGrid, OpenStack, DigitalOcean, Parallels, Proxmox and LXC. You can define wide range of infrastructure resources, networks, volumes and computers. The SaltStack uses salt-cloud capability to govern major cloud providers for compute resources using Python lib-cloud library. More complex resources are managed by individual Salt formulas.

5 Conclusion

We have presented an overview of NFV-MANO framework that has been proposed by ETSI as well as representative platforms that realize the framework architecture. Other research projects as well as industry products focused NFV MANO have been surveyed, identifying their core technologies as well as their mapping to the ETSI NFV-MANO. Based on these, we have identified and discussed opportunities to govern NFV-MANO with SaltStack configuration management platform.

We have observed that ETSI has completed the first phase of MANO standard. But the proposed MANO framework still lacks details and standards on the implementation for both managers as well as the interfaces. As a result most of the covered solutions are in fact customized and based on proprietary solutions. In addition, while some of the identified challenges such as security are being considered in some of the projects and/or industrial products, others such as resource management-and in particular automated resource management-have not received significant attention yet. Our opinion is that the success of NFV will depend on the availability of mechanisms that are able to autonomously manage network and function resources.

Our proposed SaltStack based platform can be leaveraged for complete automation of vast number of hardware and software resources from single model without any vendor-specific constraints. The Salt defined resources can be used define and manage all VNF and VI aspects of modern IT infrastructures.

Acknowledgement. This work and the contribution were also supported by project of specific science, Faculty of Informatics and Management, University of Hradec Kralove, Czech Republic.

References

1. AT&T: ECOMP (Enhanced Control, Orchestration, Management and Policy) Architecture White Paper. http://about.att.com/content/dam/snrdocs/ecomp.pdf. Accessed 21 Apr 2017
2. Arai, T., Gokurakuji, J., Oohira, M.: MANO technology supports implementation of intelligent network operations management. NEC Tech. J. **10**(3), 19–22 (2016)
3. Arai, T., Yoshikawa, N., Mibu, R.: Technology systems for SDN/NFV solutions. NEC Tech. J. **10**(3), 15–18 (2016)
4. Chayapathi, R.: Network Function Virtualization (NFV) with a Touch of SDN. Addison-Wesley Professional, Boston (2016)
5. Cloudify - Pure-Play Cloud Orchestration and Automation Based on TOSCA. http://getcloudify.org/. Accessed 20 Apr 2017
6. Coughlin, M., Keller, E., Wustrow, E.: Trusted click: overcoming security issues of NFV in the cloud. In: Proceedings of ACM International Workshop on Security in Software Defined Networks, pp. 31–36 (2017)
7. Ersue, M.: ETSI NFV Management and Orchestration - An Overview. https://www.ietf.org/proceedings/88/slides/slides-88-opsawg-6.pdf. Accessed 20 Apr 2017
8. github.com - nfvlabs/openmano: Openmano. https://github.com/nfvlabs/openmano. Accessed 02 Apr 2017
9. Goranson, P., Chuck, B.: Software Defined Networks: A Comprehensive Approach. Morgan Kaufmann, Burlington (2014)
10. Jarich, P.: AT&T, ECOMP and the Increasingly Difficult Pace of Virtualization. https://networkmatter.com/2017/02/08/att-ecomp-and-the-increasingly-difficult-pace-of-virtualization/. Accessed 21 Apr 2017
11. Kim, M.S., Do, T., Kim, Y.H.: TOSCA-based clustering service for network function virtualization. In: 2016 International Conference on Information and Communication Technology Convergence, ICTC 2016, pp. 1176–1178 (2016)
12. Kreutz, D., Ramos, F.M.V., Verissimo, P.E., Rothenberg, C.E., Azodolmolky, S., Uhlig, S.: Software-defined networking: a comprehensive survey. Proc. IEEE **103**(1), 14–76 (2016)
13. Malik, A., Ahmed, J., Qadir, J., Ilyas, M.U.: A measurement study of open source SDN layers in OpenStack under network perturbation. Comput. Commun. **102**(3), 139–149 (2016)
14. Mijumbi, R., Serrat, J., Gorricho, J.L., Latre, S., Charalambides, M., Lopez, D.: Management and orchestration challenges in network functions virtualization. IEEE Commun. Mag. **54**(1), 98–105 (2016)
15. Ming-Wei, S., Mohan, K., Taesoo, K., Gavrilovska, A.: S-NFV: securing NFV states by using SGX. In: Proceedings of 2016 ACM International Workshop on Security in Software Defined Networks, pp. 45–48 (2016)
16. Open Orchestrator. https://www.open-o.org. Accessed 19 Apr 2017
17. RIFT.io - Build Deploy Carrier-Grade Virtualized Network Services. https://riftio.com/. Accessed 21 Apr 2017
18. Sonkoly, B., Szabo, R., Jocha, D., Czentye, K., Kind, J., Westphal, F.J.: UNIFYing cloud and carrier network resources: an architectural view. In: 2015 IEEE Global Communications Conference (GLOBECOM), pp. 1–7 (2015)

19. TOSCA Simple Profile for Network Functions Virtualization (NFV) Version 1.0. http://docs.oasis-open.org/tosca/tosca-nfv/v1.0/tosca-nfv-v1.0.html. Accessed 04 Apr 2017
20. Tracker - OpenStack. https://wiki.openstack.org/wiki/Tacker. Accessed 18 Apr 2017
21. Yousaf, F.Z., Goncalves, C., Moreira-Matias, L., Perez, X.C.: RAVA - resource aware VNF agnostic NFV orchestration method for virtualized networks. In: IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC (2016)