

NFV Management and Orchestration: Analysis of OSM and ONAP

Nuno Simões

Department of Computer Science
Faculty of Sciences of the University of Porto
Porto, Portugal
up201801786@fc.up.pt

Abstract—Today, the new generation networks associated to the concept of Software Defined Network (SDN) and Network Functions Virtualization (NFV) is very relevant, particularly for telecom operators. In this paper the topic to be considered is the orchestration of the SDN/NFV networks, that is, Management and Orchestration (MANO). Taking into account the attention given to MANO, several aspects of this segment of NFV networks were analyzed. The MANO theme is analyzed in this article in the form of analysis of two solutions associated to this theme. The focus of the paper is an introductory analysis to ONAP and OSM, as an experiment and introductory paper to OSM and ONAP. Thus, two MANO solutions were analyzed: one with many followers, the OSM, and another more recent, but with plenty of potential and with a large community of participants and, consequently, users of its solution - ONAP. We show that ONAP is a much more complex tool, but, in turn, much more complete than the OSM, because it addresses other points that the OSM does not reach. Nevertheless, both solutions have their advantages and disadvantages.

Index Terms—NFV, SDN, MANO, ONAP, OSM, Orchestrator

I. INTRODUCTION

In order to fully understand the main theme of the article, it is important to introduce the SDN / NFV concepts. From the various topics related to telecommunications computer networks, some of the most popular are Software-Defined Networking (SDN) and Network Functions Virtualization (NFV). Regarding SDN, where there used to be only hardware and network protocols, there is now software that allows drivers to run. About NFV, this allows highly optimized packet processing of network functions. Both concepts allow the control and orchestration of architectures and are used to create, manage and scale new on-demand service platforms. These last described functions can be made fast and agilely [1]. The creation of an L3VPN service (Virtual Private Network over the network layer of the OSI model) is an example of the use of SDN/NFV¹.

The SDN had as starting point the academia and has been developed by researchers and architects of data centers. The NFV was created from a consortium of Service Providers. The concepts of SDN/NFV are related to the management of software networks and creation of network functions in a more

agile and fast way. Thus, the operators, above all, can more quickly meet the pretensions of the customers. The concept of SDN is based mainly on two concepts: Data plane and control plane. This way a control layer is added to the network management [2]. Comparing one concept with another, while the SDN aims to centralize network management, the NFV focuses on virtualizing network functions. This virtualisation aims to reduce physical equipment whenever possible, with its replacement being done for example by virtual machines (VM) [3].

Characteristics and themes associated with these two concepts began to emerge. Some of them are the virtualization of networks, the virtualization of some services, SDN controllers, NFV management and orchestration (MANO) (that will be detailed in the next section), reduction of physical space and reducing electrical energy [3].

The purpose of this paper is to analyze two MANO solutions: OSM and ONAP. This paper try to be an introduction to the theme of orchestration in NFV and, more specifically, an introduction to the two referred tools. Both tools come to try to help in the optimization and automation of tasks with regard to the management of networks through software. In particular, the tools can deal with the orchestration, policing and automation of physical and virtual infrastructures of computer networks.

The remainder of this paper follows the following organization. In the next section it will be presented the state of the art with regarding NFV Management and Orchestration (MANO). In the third section, two examples of MANO (the OSM, Open Source MANO, and ONAP) are presented, as well as a small comparison between each one. In the final section we will conclude the analysis carried out, as well as the presentation of some comparative points between both MANO solution.

II. STATE OF THE ART OF NFV MANAGEMENT AND ORCHESTRATION

As the meaning of the acronym MANO indicates, the NFV Management Orchestration is the NFV manager. NFV MANO is a working group of the *European Telecommunications Standards Institute Industry Specification Group*, commonly known as ETSI ISG NFV. NFV MANO manages and orchestrates all the resources in the cloud data center. This includes computing, networking, storage, and virtual machine resources

¹ATT and Ericsson presenting SDN-based L3VPN solution for Telco NFV needs: <https://cloudblog.ericsson.com/digital-services/att-and-ericsson-presenting-sdn-based-l3vpn-solution-for-telco-nfv-needs>

[4]. MANO is a critical point in the proper functioning of NFV Infrastructure (NFVI) and Virtual Network Functions (VNFs) and provides the requisite requirements for provisioning and configuration of VNFs [5].

The NFV MANO is divided into three functional blocks [4] [5]:

- NFV Orchestrator (NFVO): combines several functions to create end-to-end services;
- VNF Manager (VNFM): is responsible for the life cycle of network virtualization functions (VNF). Can manage only one or multiple instances of VNFs;
- Virtualized Infrastructure Manager (VIM): manages and controls NFV physical and virtual infrastructure resources across a single domain.

Along to the blocks, NFV MANO has four types of data repositories. These repositories are databases that have several NFV MANO information. The repositories are [5]:

- Network Services (NS) Catalog: are a set of predefined templates that indicate how services should be created and developed;
- VNF Catalog (VNFM): it is a set of templates that describe characteristics of development and operationalization of VNFs;
- NFV instances: are where all the information related to functions and services is stored;
- NFVI resources: are where all information related to NFVI is stored.

The use of the NFV concept foresees that its application can be gradual and with the existing equipment.

MANO is mostly used in aspects related to virtualization mechanisms, while network management is more directed to network services composed of VNFs and physical nodes. This network management also includes Operation System Support (OSS) and Business System Support (BSS), among others.

In Figure 1 we can observe the general architecture of NFV MANO [6]. In this Figure we can see the functional blocks and repositories already mentioned in this paper. With this image we can get a more visual idea of the MANO [6]. The EMS is the Element Management System.

To conclude the section, are presented some projects related to NFV MANO.

These examples can be [5]:

- CloudNFV: is an open platform for NFV implementation. This solution, based on cloud computing and SDN, is composed of three elements: Active virtualization, which is a data model; NFV Orchestrator, which has policing rules, for example; NFV management, which uses Management Information Bases (MIBs) that is a database to manage entities in a network communication, using Simple Network Management Protocol (SNMP);
- OpenMANO: is an open source project, led by the spanish operator *Telefónica* and aims to implement a MANO framework. The architecture of this solution is based on three main components: openmano, based on REST and that serves for the management of VNF; openvim,

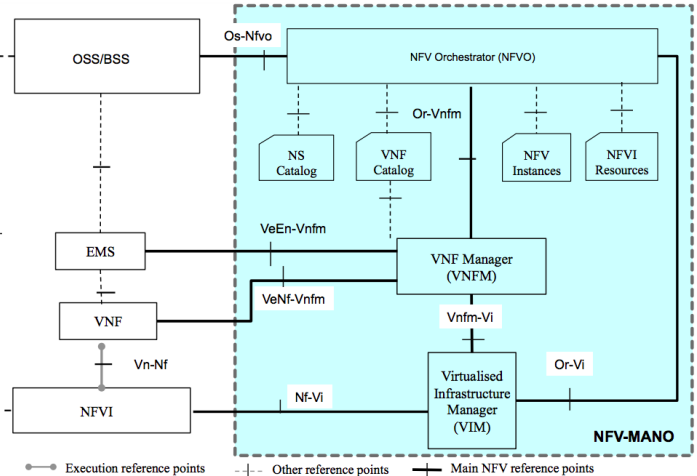


Fig. 1. NFV MANO architecture [6].

which is an implementation of VIM; GUI, which is the component of the Graphical User Interface.

Further to these two, there are also others such as ExperiaSphere, Zoom, OPNFV [5], Open Orchestrator Project (OPEN-O), Tacker, Cloudify and Open Baton [7].

This concludes the section on the state of the art.

III. OPEN SOURCE MANO (OSM) AND OPEN NETWORKING AUTOMATION PLATFORM (ONAP)

After the presentation of the adjacent concept, the main theme of the paper is presented. In this section, a presentation of the OSM and then of ONAP will be made.

A. Open Source MANO (OSM)

Open Source MANO, or OSM, is mostly developed in Python and runs on Linux operating systems [8]. The OSM was released in 2016 [9]. The most recent version, that has been released to date, is 5 [10]. It is important to note that OSM 5 already includes references to network slice and orchestration, two very popular themes nowadays regarding to SDN/NFV [11].

The OSM uses three functional blocks (NFVO, VNFM and VIM) of the NFV MANO to perform the configuration and abstraction of VNFs, and orchestration and management of the infrastructure [12].

Some of the operations supported by the OSM are [12]:

- An orchestration service for VNF;
- The possibility of performing complex services;
- Support for different VIMs;
- Support for SDN controllers;
- Support for monitoring tools.

The OSM proposal is to map the ETSI ISG NFV architecture into an Open Source implementation available on the OSM website itself [10]. In the Figure 2, the mentioned OSM mapping can be consulted.

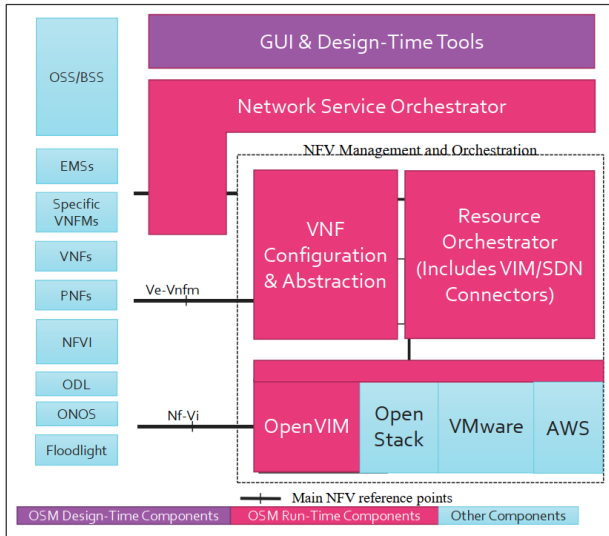


Fig. 2. Mapping from OSM to ETSI NFV MANO [13].

The use of the OSM can have several objectives. In addition to providing management and orchestration components to the NFV, the use of the OSM can also be made to facilitate the management, development and design of VNFs and network services. Another goal may be the evolution of the reference architecture proposed by ETSI ISG NFV. Thus, the use of OSM can become massive [8]. Besides this, it can also grow with the implementation of 5G. The OSM is one of the MANO quite associated with 5G [9].

Regarding the OSM requirements and installation, this solution has some features. According to some references in the documentation, OSM already uses the container concept via docker. This is because the website² references a *dockerized* installation of OSM. In addition, since version 4, OSM has been targeting the cloud-native strand, like so many other technologies nowadays. Still since version 4 they have a lighter orchestrator and a new GUI.

On the installation requirements, according to the documentation², only one machine, or VM, with the characteristics shown in table I are required.:

TABLE I
 REQUIREMENTS FOR OSM INSTALLATION

	Minimum	Recommended
CPU	2	2
RAM (GB)	4	8
Storage (GB)	20	40

In terms of operating system, Ubuntu 16.04 is referred to. Now about the installation, it appears to be quite simple, with all the steps being properly explained on the website². There

²OSM Release FOUR: https://osm.etsi.org/wikipub/index.php/OSM_Release_FOUR

is also a reference to the components that we want to include in the installation.

After the installation is complete, the web interface login page can be verified. Once the installation is complete, 10 docker containers are installed.

In addition to the installation on a machine or VM, it is also described how to configure OpenStack, VMware vCloud or Amazon Web Services to use OSM. In the end, a simple VNF can be started according to the steps described on the website².

Giving the presentation of the OSM is finished, ONAP will be presented in the next subsection.

B. Open Networking Automation Platform (ONAP)

According to ONAP’s own documentation, its solution comes to close a failure that existed at the level of telecom operators, cable and cloud. According to ONAP, there was a lack of a common platform, which offered different types of services and could also be competitive and compensatory [14].

ONAP is a solution with orchestration capabilities, which can be applied to both real and virtual elements. The ONAP solution is modular and supports YANG and TOSCA³ data models. It is important to notice that YANG and TOSCA are data modeling language. It is stated in the document [14] that ONAP can be integrated with several VIMs, VNFMs, SDN Controllers and even devices that may already be somewhat obsolete. This type of integration can lead to financial savings and reuse of existing material.

Design-Time consists of 2 subsystems⁴:

- Service Design and Creation (SDC);
- Policy.

The SDC has 4 main components⁵:

- Catalog: is where data arrives using Design Studio;
- Design Studio: is used to create, modify, and add Resource and Service definitions to Catalog;
- Certification Studio: is the test and experiment point of SDC for future releases;
- Distribution Studio: is the intermediate point between the Certification Studio and the placement of assets into production. This component is not explicit, but it is represented in "Recipe/Eng. Rules Policy Distribution".

Regarding the architecture of ONAP, it is represented in a high level perspective in Figure 3. Note that in the Figure 3 the catalog is what makes the connection between Design-Time, which allows the reuse of service models already developed for example, and Run-Time, where the rules and policies coming from Design-Time are executed.

The ONAP runtime is where the ONAP Policy, Command, or Request, and Inventory components are located [14].

³TOSCA vs. Netconf a Comparison: <https://www.sdxcentral.com/nfv/definitions/tosca-vs-netconf-comparison/>

⁴ONAP Architecture: <https://wiki.onap.org/display/DW/Architecture>

⁵Service Design and Creation (SDC): <https://wiki.onap.org/pages/viewpage.action?pageId=1015837>

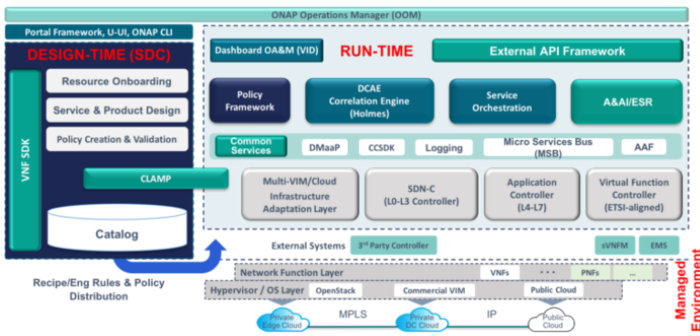


Fig. 3. ONAP high-level architecture [14].

On ONAP deployment, there are three possibilities for doing this⁶ component by component, via Heat (OpenStack) or via Kubernetes/ONAP Operations Manager (OOM)⁷. Depending on the chosen option, other platforms may be used. If we choose Kubernetes we can use⁸: Amazon EC2, Cloudify, Google Cloud Engine, among others. In the following table are presented the requirements of each of the options⁶. It is important to note that ONAP allows each component to be installed independently. For example, the Service Orchestrator or Policy Framework can be installed as an isolated component. So there is no need to install all the components to run/test just one.

TABLE II
 RESOURCES NEEDED TO INSTALL ONAP

	Kubernetes	Heat	Single component
VM	4	20	1
vCPU	32	88	1
RAM (GB)	128	176	2
Storage (GB)	160	1760	20

Note that the requirements presented in the Table II are for the Beijing version of ONAP. For the Casablanca version of ONAP the requirements in terms of infrastructure are less onerous⁹. About the versions of ONAP, the developers expect to release of two versions per year¹⁰. For the moment, the Casablanca version is already in use, but still at a very early stage¹¹.

The ONAP automation cycle can be seen in Figure. 4.

ONAP is based on a unified architecture [15]. Basically, this cycle can have the following path: Design > Create > Collect > Analyze > Detect > Publish > Respond [14].

⁶Setting Up ONAP: <https://onap.readthedocs.io/en/beijing/guides/onap-developer/settingup/index.html>

⁷ONAP Operations Manager Project: <https://wiki.onap.org/display/DW/ONAP+Operations+Manager+Project>

⁸ONAP on Kubernetes: <https://wiki.onap.org/display/DW/ONAP+on+Kubernetes>

⁹Setting Up ONAP: <https://onap.readthedocs.io/en/casablanca/guides/onap-developer/settingup/index.html>

¹⁰Release Calendar: <https://wiki.onap.org/display/DW/Release+Calendar>

¹¹Casablanca Maintenance Release: <https://wiki.onap.org/display/DW/Release+Planning#ReleasePlanning-CasablancaMaintenanceRelease>

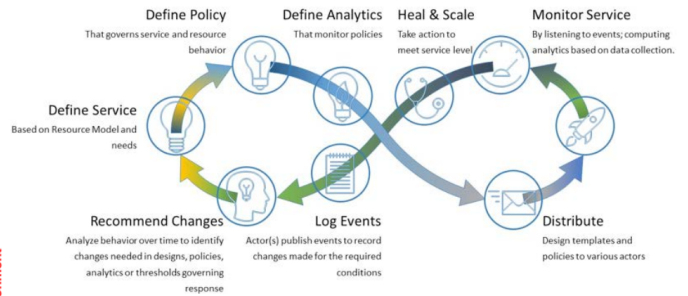


Fig. 4. ONAP Automation Cycle [14].

To conclude this section, in Figure 5 is presented an illustrative image of the execution of ONAP in a physical infrastructure composed by 2 servers. Initially, only one server was envisaged but, due to the limited number of pods (components) of ONAP¹², another machine was added. Both machines are running Ubuntu 16.04. One has 126GB of RAM and 366GB of storage and the other 39GB of RAM and 133GB of storage. At the top of Figure 5 is a list of some running pods. At the bottom of the same Figure is the portal of Kubernetes at Rancher. Rancher, that is a cluster Kubernetes management, is used since ONAP is instantiated on Kubernetes.

```

root@root:~# kubectl get pods --all-namespaces
NAMESPACE      NAME                                     READY   STATUS    RESTARTS   AGE
kube-system    heapster-7b48b96ec-tnmct               1/1     Running  0           12d
kube-system    kube-usa-665f570c6-dm6d5               3/3     Running  0           12d
kube-system    kubernetes-dashboard-ef5477c4b-nd5vk   1/1     Running  0           12d
kube-system    monitoring-grafana-7877679464-dm95c   1/1     Running  0           12d
kube-system    monitoring-influxdb-64664c6c5-965pw   1/1     Running  0           12d
kube-system    tiller-deploy-7042b6bdf-bmmt9         1/1     Running  0           12d
onap           dep-config-handling-service-16c8cc494-phdjt 2/2     Running  0           12d
onap           dep-dcaae-tca-analytica-5877b9b85-2g8m5 2/2     Running  0           12d
onap           dep-dcaae-vea-collector-664f808567-9465e 2/2     Running  0           12d
onap           dep-deployment-handler-55d8f4444b-ktne6 2/2     Running  0           12d
onap           dep-dcaae-vea-collector-664f808567-9465e 1/1     Running  4           12d
onap           dep-holmes-tui-mgmt-695667cc5b-9dm7c   1/1     Running  0           12d
onap           dep-inventory-52bb762b4f-2l9w5         1/1     Running  4           12d
onap           dep-policy-handler-4b708cb7f-d329q     2/2     Running  1491        12d
onap           dep-patg-write-6467b468ce-77msz       1/1     Running  4           12d
onap           dep-holmes-vea-collector-664f808567-9465e 0/1     CrashingBackoff 3150        12d
onap           dev-aaf-cm-ef5f9775-6e68x             1/1     Running  0           12d
onap           dev-aaf-cm-16055943bc-9778x          1/1     Running  0           12d
onap           dev-aaf-fe-6b49c449-57vap             1/1     Running  0           12d
onap           dev-aaf-gui-564576786-5b7xs           1/1     Running  0           12d
    
```

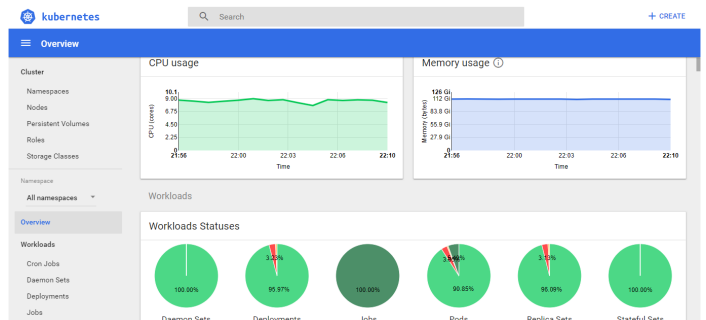


Fig. 5. Above, querying the list of installed pods and, in the bottom, the Kubernetes GUI.

The ONAP instantiation was not very simple. To deploy ONAP, OOM was used. The script used for the software requirements is what is presented on the website¹³. A change was made to the Rancher installation, i.e., the commands

¹²ONAP on Kubernetes on Google Compute Engine: <https://wiki.onap.org/display/DW/ONAP+on+Kubernetes+on+Google+Compute+Engine>

¹³oom_rancher_setup.sh: https://github.com/onap/logging-analytics/blob/master/deploy/rancher/oom_rancher_setup.sh

followed were similar to those presented on the website itself [16].

Unfortunately, I did not have the opportunity to test the OSM.

This concludes the section on MANO solutions.

IV. CONCLUSION AND FUTURE WORK

Very briefly, it is concluded that ONAP touches all points of Management & Control, while OSM only refers to the Orchestration, Management & Policy topic of Management & Control according to the Linux Foundation. This can be seen in Figure 6.

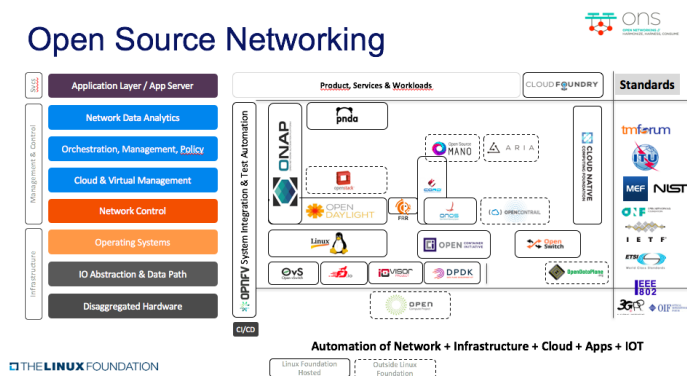


Fig. 6. Open Source Networking¹⁴.

From our experience, the installation of ONAP is not complicated. The only experimental factor made for this article was the instantiation of ONAP in a physical infrastructure. Thus, the ONAP conclusion is based only on this fact. The hard part may be finding the way, or the right directions, to do so. In our view, one of the main problems of ONAP, compared to OSM, are the requirements. As can be seen in the previous section, the requirements of ONAP are much higher than the characteristics of the infrastructure where this solution can be based. From this point of view, the OSM is more advantageous because it does not need so many resources, as indicated in the website².

Another important issue is documentation. In terms of documentation, both have plenty. The point is that although we have been more concerned about ONAP, it seemed to us that the OSM documentation was better structured. ONAP has plenty documentation, but it seems to be spread across several websites. It did not seem to us that it was present in a place and that it was well structured. This factor is also important during the development and use of any technological solution. But one of the reasons may be that ONAP is a much more complex tool than OSM.

¹⁴How ONAP Will Merge Millions of Lines of Code from ECOMP and Open-O: <https://www.sdxcentral.com/articles/news/onap-will-merge-millions-lines-code-ecomp-open-o/2017/04/>

It is important to note that there are many other Open Source solutions competing with OSM and ONAP. Nevertheless, these others have received less attention.

In terms of future work, and extending the subject of MANO to solutions other than OSM and ONAP, other possibilities could also be analyzed. It would also be interesting to present performance results between the different solutions. Thus, it could be verified that there would be a more efficient solution compared to others. An analysis to management orchestration taking in attention the 5G technology will be an interesting thematic to analyze.

REFERENCES

- [1] S. Van Rossem, W. Tavernier, B. Sonkoly, D. Colle, J. Czentye, M. Pickavet, and P. Demeester, "Deploying elastic routing capability in an SDN/NFV-enabled environment," in *Network Function Virtualization and Software Defined Network (NFV-SDN), 2015 IEEE Conference on*. IEEE, 2015, pp. 22–24.
- [2] V. Wijekoon, T. Dananjaya, P. Kariyawasam, S. Iddamalgoda, and A. Pasqual, "High performance flow matching architecture for openflow data plane," in *Network Function Virtualization and Software Defined Networks (NFV-SDN), IEEE Conference on*. IEEE, 2016, pp. 186–191.
- [3] B. B. Westcon, "What is the difference between SDN and NFV?" available at <https://blogbrasil.westcon.com/qual-a-diferenca-entre-sdn-e-nfv>.
- [4] S. Central, "What is nfv mano?" available at <https://www.sdxcentral.com/nfv/definitions/nfv-mano/>.
- [5] R. Mijumbi, J. Serrat, J.-L. Gorricho, S. Latré, M. Charalambides, and D. Lopez, "Management and Orchestration Challenges in Network Functions Virtualization," *IEEE Communications Magazine*, vol. 54, no. 1, pp. 98–105, 2016.
- [6] M. Ersue, "ETSI NFV Management and Orchestration - An Overview," in *Proc. of 88th IETF meeting*, 2013.
- [7] C. Parada, J. Bonnet, E. Fotopoulou, A. Zafeiropoulos, E. Kapassa, M. Touloupou, D. Kyriazis, R. Vilalta, R. Muñoz, R. Casellas et al., "5GTango: A Beyond-Mano Service Platform," in *2018 European Conference on Networks and Communications (EuCNC)*. IEEE, 2018, pp. 26–30.
- [8] A. S. Saito and F. L. Verdi, "High availability support management in nfv environments using osm."
- [9] M. Bhagwat, D. Clarke, P. Eardley, A. E. Armengol, G. de Blas, P. Gronlund, A. Hoban, S. Manning, T. Nakamura, F. J. Ramn Salguero, and A. Reid, "OSM: Experience with NFV Architecture, Interfaces and Information Models," *Open Source MANO, Technical Overview*, 2018.
- [10] OSM, *Open Source MANO*, OSM, available at <https://osm.etsi.org/>.
- [11] OSM, *OSM PoC 1: DevOps in Service Chains and 5G Network Slices*, OSM, available at https://osm.etsi.org/wikipub/index.php/OSM_PoC_1_-_DevOps_in_Service_Chains_and_5G_Network_Slices.
- [12] G. Venâncio, V. F. Garcia, L. da Cruz Marcuzzo, T. N. Tavares, M. F. Franco, L. Bondan, A. E. Schaeffer-Filho, C. R. P. dos Santos, L. Z. Granville, and E. P. Duarte Jr, "Simplifying Lifecycle Management of Network Virtualized Functions."
- [13] A. Israel, A. Hoban, A. Sepúlveda, F. Salguero, G. de Blas, K. Kashalkar, M. Ceppi, M. Shuttleworth, M. Harper, M. Marchetti et al., "OSM release three," *Open Source MANO, Technical Overview*, 2017.
- [14] ONAP, *ONAP Architecture Overview*, ONAP, available at https://www.onap.org/wp-content/uploads/sites/20/2018/06/ONAP_CaseSolution_Architecture_0618FNL.pdf.
- [15] F. Slim, F. Guillemin, A. Gravey, and Y. Hadjadj-Aoul, "Towards a dynamic adaptive placement of virtual network functions under onap," in *Network Function Virtualization and Software Defined Networks (NFV-SDN), 2017 IEEE Conference on*. IEEE, 2017, pp. 210–215.
- [16] Rancher, *Rancher: Single Node Install*, Rancher, available at <https://rancher.com/docs/rancher/v2.x/en/installation/single-node/>.