

# A Network Service Development Kit Supporting the End-to-End Lifecycle of NFV-based Telecom Services

Steven Van Rossem\*, Manuel Peuster<sup>†</sup>, Luís Conceição<sup>‡</sup>, Hadi Razzaghi Kouchaksaraei<sup>†</sup>, Wouter Tavernier\*, Didier Colle\*, Mario Pickavet\* and Piet Demeester\*

\*Ghent University imec, IDLab, Department of Information Technology.

Email: steven.vanrossem@ugent.be

<sup>†</sup> Paderborn University

<sup>‡</sup> Ubiwhere

**Abstract**—Ongoing evolutions in the Network Function Virtualization (NFV) area show that software will play an increasingly important role within the telecommunication industry. It enables telco providers to faster introduce telecom services, combined with new possibilities to optimize and fine-tune operational performance. New virtualization and softwarization methods support fine-grained scaling of resources and highly-customizable configuration settings. Additionally, multi-datacenter topologies are available to deploy the Virtual Network Functions (VNFs) a service consists of. In this context, specialized tools are needed for debugging and validating the deployment, placement, chaining, configuration and scaling of network services, before the operator deploys the service in production. This demo showcases an open-source Software Development Kit (SDK), built to support NFV-based services throughout their whole lifecycle. During the demo, several example VNFs will be loaded into the SDK environment to demonstrate the features a service developer is envisioned to use, while either creating or updating NFV-based telecom services.

**Index Terms**—NFV, SDK, DevOps, SONATA

## I. INTRODUCTION

NFV-based services process network traffic using interconnected Virtualized Network Functions (VNFs). VNFs can have a distributed deployment on multiple data centers at different locations, eg. closer to the network edge. The flow of network traffic is controllable, meaning there is a fixed chain or order in which the network functions process the packets. This differs from the application-layer based implementation of classic cloud applications. The 'classic' cloud ecosystem is basically a connection from a client to a server in a single datacenter, where the network infrastructure in between is not leveraged. To fill this gap, we have build an SDK environment to additionally validate: Service Function Chaining (SFC), custom placement on multiple Points of Presence (PoPs) and dedicated control functions to configure and scale VNFs in the service. This functionality builds further upon the available standards for Management and Orchestration (MANO) frameworks<sup>1</sup>, extended with an SDK

environment<sup>2</sup>. The code is open-source and freely available<sup>3</sup>.

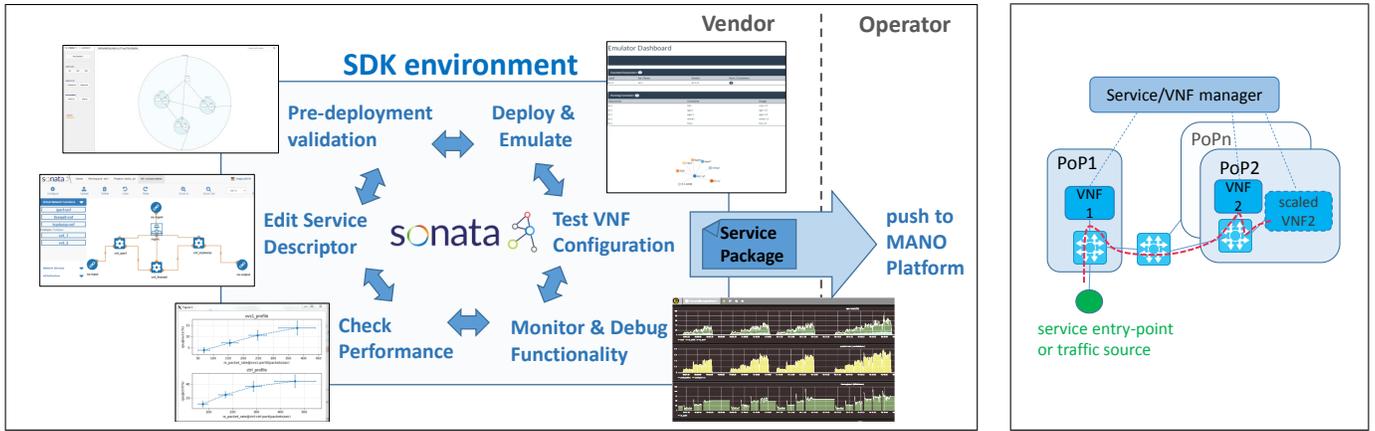
## II. TELCO-GRADE DEVOPS

This SDK effectively extends the MANO framework, as it helps to tightly integrate both operational and development aspects into one ecosystem (cf. the DevOps methodology). Earlier MANO-DevOps integrations such as [1] only focus on troubleshooting operational services. In fact, we can consider this SDK environment as a sandbox to develop and try-out NFV-based network services. It offers support for typical NFV-related actions such as: define and validate VNF forwarding chains, inject test traffic into an emulated network service, test management APIs of a VNF, identify bottlenecks and package the newly developed or modified service. The end-product of the SDK is a service package: a definition of the complete service, abstracting all the VNFs, network links and control mechanisms. This lowers the barrier between the proprietary domains of Vendors and Operators in a Telco context, as illustrated in Fig. 1a. Any vendor or service developer can use the SDK to create a VNF-based service, validate it and compile it into a package. This package is then ready for deployment on the operator's MANO framework. The relation between SDK and MANO framework is further discussed in [2]. The support for different MANO platforms (SONATA and OSM) is demonstrated in [3]. The modular design of the SDK allows that support for other package formats can be easily added. Additional security features such as package signing and user management are in place. The total feature set allows that the resulting service package can be trusted by the MANO operator to be fully functional and secure. The high-level development cycle is summarized in Fig. 1a. Bottom-up, a supporting toolset is developed, building further on the work presented in [4] and adding functionality beyond monitoring and debugging. We highlight the tools during the demo and shortly describe them in the sections below.

<sup>2</sup>Technical implementation details on the SDK can be found in the SONATA project's deliverables on <http://www.sonata-nfv.eu>

<sup>3</sup>All repositories are open-sourced under Apache v2 license and available via <https://github.com/sonata-nfv>

<sup>1</sup>ETSI NFV group: <http://www.etsi.org/technologies-clusters/technologies/nfv>



(a) SDK development flow with supporting tools to export a service package to the operator’s MANO platform. (b) Custom emulated topology to test VNF chaining and configuration.

Fig. 1: The SDK supports both formal and functional validation features.

### III. SERVICE DEPLOYMENT

To ensure a successful deployment of the service, we demonstrate the SDK support for following aspects:

A **service descriptor** defines the service in terms of resources, VNF images, networking, forwarding chains, alarm triggers and control functions. A web-based service graph editor allows visual editing of the VNF chains. This is compiled into a signed service package, ready for deployment on a virtualized infrastructure.

The SDK includes a **service emulator** which is capable of deploying container-based service chains. It is a light-weight environment, emulating arbitrary virtual multi-PoP infrastructure topologies on a developer’s machine [3] as shown in Fig. 1b. Both direct links and LAN networks between VNFs can be emulated with user-defined delays and loss-rates. A REST API and dashboard is available to control and verify the emulated service and topology. After deployment, the **service configuration** can be validated. The assigned IP and MAC addresses can be queried, the management interfaces of each VNF in the emulator can be tested, as well as the configuration of the traffic forwarding.

### IV. SERVICE VALIDATION

In the SDK’s development flow, the service is audited before and after deployment: A **pre-deployment, formal validation** of the forwarding graphs and generated service package is done, to make sure that the MANO platform is able to accept and deploy the network service. Validation issues such as invalid connection points, repeated paths and existence of cycles in the forwarding graphs provide hints about the service accuracy and eligibility for deployment. On the other hand, some of these validation issues may be ignored, e.g. a cycle in a forwarding graph may not be problematic if an involved VNF has the ability to steer traffic and influence the SFC. For this reason, all validation issues can be configured to be reported as errors, warnings or simply ignore them. From the SDK, the service package can be pushed to an emulation environment or to the operator’s MANO platform. **Post-deployment**

**functional validation** is done by analyzing monitored data. The SDK monitoring system is capable of gathering metrics regarding VNF resource usage and filtered flow rates. In addition, traffic generators or custom-built test scripts can be attached to the endpoints of the emulated service to inject test traffic. The monitoring functionality is extended with analysis tools for **VNF performance profiling** to aid in capacity and resource planning for virtualized services. By defining automated test runs and metrics to be monitored, the performance of the VNFs can be characterized under varying resource configurations. By exploiting the configuration APIs of the VNFs and the API of the emulator, possible **management strategies** such as scaling can be tested, as shown in in Fig. 1b.

#### A. Example Services

The SDK tools are applied to several example services like: An OpenFlow controller and OvS instance<sup>4</sup>, a vEPC service<sup>5</sup> or a vCDN service implemented using Squid. Configuration and deployment can be manipulated by a management function. The used toolset shortens the development time needed to create a fully validated NFV-based service package, ready to be deployed on a MANO platform.

### REFERENCES

- [1] W. John *et al.*, “Service provider devops,” *IEEE Communications Magazine*, vol. 55, no. 1, pp. 204–211, 2017.
- [2] H. Karl *et al.*, “Devops for network function virtualisation: an architectural approach,” *Transactions on Emerging Telecommunications Technologies*, 2016.
- [3] M. Peuster *et al.*, “A flexible multi-pop infrastructure emulator for carrier-grade mano systems,” in *IEEE NetSoft 2017*.
- [4] S. Van Rossem *et al.*, “Monitoring and debugging using an sdk for nfv-powered telecom applications,” in *IEEE NFV-SDN 2016*.

#### Acknowledgement

This work is done in the SONATA project, funded by the European Commission through the 5G-PPP program. We thank the SONATA consortium for all contributions ([www.sonata-nfv.eu](http://www.sonata-nfv.eu)).

<sup>4</sup>A demo movie is available on YouTube: <https://youtu.be/J14mb79Fwmc>

<sup>5</sup>vEPC based on [https://github.com/networkedsystemsIITB/NFV\\_LTE\\_EPC](https://github.com/networkedsystemsIITB/NFV_LTE_EPC)