Introducing Automated Verification and Validation for Virtualized Network Functions and Services

Manuel Peuster*, Stefan Schneider*, Mengxuan Zhao†, George Xilouris‡, Panagiotis Trakadas§, Felipe Vicens¶, Wouter Tavernier‖, Thomas Soenen‖, Ricard Vilalta**, George Andreou‡‡, Dimosthenis Kyriazis††, and Holger Karl* *Paderborn University (manuel.peuster@upb.de), †Easy Global Market, ‡NCSRD Demokritos, §Synelixis Solutions, ¶ATOS, ‖University of Ghent (imec), **CTTC, ††University of Piraeus, ‡‡Huawei

***Abstract*—Network function virtualisation (NFV) and software defined networks (SDN) will transform network management and operation tasks into agile development tasks. They will involve software artefacts which are managed and deployed as composite services using DevOps principles. Those softwarised networks rely on complex technology stacks, starting with low-level virtualisation technologies and ranging up to machine learning-based orchestration solutions. One of the main challenges in those environments is to verify that the deployed functions and services operate correctly and meet the quality goals, set by the stakeholders, before they are put to production.**

We tackle this challenge by introducing the novel concept of a verification and validation (V&V) platform for NFV, which enables automatic testing and qualification of single network functions and complex services. By adding such a platform to the NFV ecosystem, new business models emerge as we discuss in this article. We evaluate our proposed concepts by presenting a case study that uses our open-source V&V platform to verify and validate the behaviour and performance of a real-world network service.

I. Introduction

The upcoming 5th generation of networks (5G) is expected to be the backbone and enabler of many innovative services, ranging from those that require ultra-low latency to those with ultra-high bandwidth demands. Examples are the emerging vertical use cases for 5G, such as smart manufacturing (industry 4.0), immersive media, connected vehicles, or public protection and disaster relief, which cannot be efficiently implemented in legacy, general-purpose networks [1]. To tackle this, technologies like software-defined networks (SDN) and network function virtualisation (NFV) are emerging and will allow to apply agile methods and DevOps concepts to the networking domain [2].

However, the softwarisation of networks raises a series of questions about quality control and availability assurance: First, how to verify that all involved components of the technology stack, and especially the virtualised network functions (VNF), work correctly? Second, how to validate that complex service function chains (SFC), consisting of multiple, chained VNFs, correctly implement the intended service? Third, how to dimension virtualised resources to meet quality of service (QoS) goals? And finally, how to know about the aforementioned characteristics before a single VNF or service is deployed to production? These questions directly point to the challenge of testing all involved components of the NFV stack and motivate the need of automated verification and validation solutions for NFV.

A. Verification & Validation for NFV

In the software engineering community, *verification & validation (V&V)* is a way to determine whether a software product operates correctly and meets all predefined requirements [3]. The application of automated V&V concepts to the NFV domain promises to improve the reliability, interoperability, and quality of softwarised network solutions. It will also reduce the time-to-market for new VNFs and services even further. In addition, are V&V mechanisms important for the emerging vertical use cases that have their very own, strict, perhaps contradicting requirements. Approaches that allow to verify and validate wether a certain VNF or service (strictly) meets its requirements, if deployed in a given environment, will be a key enabler for wide adoption of softwarised 5G technologies and agile service deployments [1].

This leads to the question how automated V&V concepts can be applied to the NFV domain and how our networks can benefit from them? To answer this question, we take a closer look at a typical NFV scenario shown in Fig. 1. It shows a network service, consisting of four VNFs that are chained together to form an SFC. Each VNF is a virtualised entity (e.g., virtual machine or container) that is executed on top of a given (maybe distributed) NFV infrastructure (NFVI). The complete deployment is under control of the management and orchestration (MANO) system that requests the instantiation of the virtualised entities, controls the lifecycle of the VNFs, and manages their respective SFC. On top of the MANO layer, descriptors that specify how a MANO system should deploy and manage a certain service and its VNFs are shown.

Fig. 1 shows that NFV scenarios are different from typical software projects because they do not only contain the actual application code that implements the VNF's packet processing capabilities. They also contain many additional artefacts, like descriptors for VNFs and services, LCM scripts, or disk images for different NFVI technologies. All of them are key artefacts for the proper operation of a service and thus need to be tested. We highlight this with different scopes in the shown scenario, which are then mapped to the corresponding test method specification published by ETSI [4]. Each of these scopes deals with components within the scope itself; external reference points, e.g., lines that leave a scope, are not considered. Table I shows the test types involved in each of the scopes and gives brief examples. The mapping to the ETSI specifications as well as to the different test types highlight

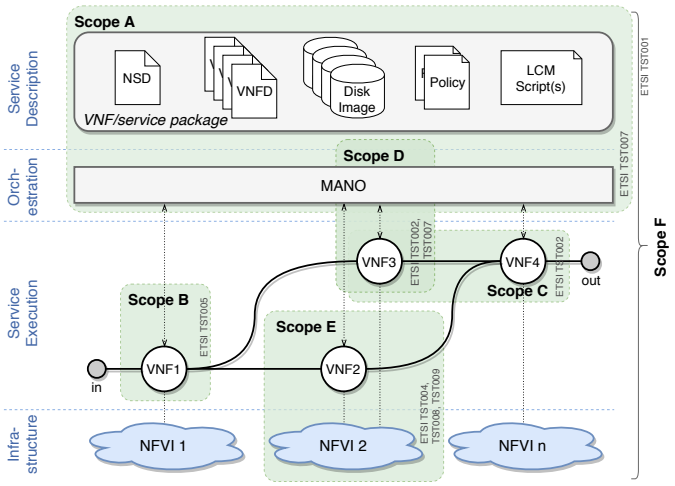

Fig. 1: Typical NFV scenario. The dashed boxes highlight the different scopes in which V&V concepts and methods should be applied (cf. Table I).

that V&V solutions for NFV have to be generic enough to support a wide variety of tests to verify and validate VNFs and services ent-to-end.

B. Contributions

This article builds upon our V&V concepts, initially presented in [5], [6], and uses those ideas to introduce the first integrated and fully-automated platform to apply V&V methods to the NFV domain. After discussing state-of-the-art solutions in Section II, we present our V&V platform and its integration into the NFV ecosystem in Section III. In this article, we focus not only on the tight integration into the NFV ecosystem but also on novel business aspects and refined roles, which goes beyond our previous work. Finally, we present a case study that, for the first time, uses our V&V platform in a real-world scenario, as shown in Section IV. We provide detailed insights into our platform, which is part of the open source 5GTANGO NFV framework [7]. This gives the reader a blueprint to implement fully-automated V&V solutions for NFV.

II. State of the Art

Recently, verification and validation of VNFs and services (or SFCs) has started to attract the attention of researchers, standardisation bodies, and vendors. Besides theoretic approaches, like NetKAT [8] which relies on Kleene algebra to provide formal verifications of packet-processing functions, the community is mostly seeking for practical end-to-end NFV testing solutions, which is what we address in this article.

Standardisation bodies, like ETSI and IETF are actively publishing reports and specifications about NFV testing [4], [9]. An outstanding example for this are ETSI's TST documents [4] which report, e.g., on "pre-deployment validation" of NFV environments and services (TST001), "interoperability testing methodologies" (TST002), or "MANO interoperability testing" (TST007). The given specifications are highly relevant for our work and the presented prototype is designed to be compatible to them, e.g. in terms of used package formats. Nevertheless, ETSI's documents only specify guidelines, methodologies, high-level architectures, and abstract test suites but they lack concrete real-world examples or proof-of-concepts. This article provides those while focusing on clause 7 ("pre-deployment validation of VNFs") and clause 8 ("pre-deployment validation of network services") of ETSI TST001 [4]. In addition, the general V&V platform concept can also be used for, e.g., interoperability testing of MANO systems, which is out of scope of this article. Finally, none of the mentioned standardisation activities provides the means and tools to execute the test scenarios they specify, which prohibits their wide adoption. Our novel V&V platform fills this gap and can be used to implement and execute tests, e.g., based on the ETSI specifications.

NFV testing solutions of vendors are still limited to specification documents, such as Cisco's "Third-Party NFV Ecosystem Certification Test Plan" [10], and do not include fully-automated, end-to-end test platforms, as we propose in this article. We even argue that V&V solutions for NFV must be open source to provide the necessary level of transparency to be successful and widely adopted. Existing open source solutions, such as OPNFV (https://www.opnfv.org/), provide first usable test tools, such as *Yardstick* for NFVI testing or *Qtip* for component benchmarking. But each of these tools focuses on a specific area within the NFV ecosystem, e.g., data plane testing and not on automated, end-to-end testing. However, they can be integrated into and controlled by the proposed V&V platform as a specific test case.

In the research community, Pelay et al. [11] recently proposed a solution to check the consistency of VNF descriptions on real deployments, introducing the concept of augmented network topology. Besides this, several approaches for VNF and service performance benchmarking or profiling have been proposed [12]–[14]. Even though such frameworks provide support for automated performance measurements of VNFs and services, they lack the support for describing and implementing general-purpose test cases, e.g., functional tests on different NFVI stacks. However, especially [13], [14] could be integrated with and triggered by our proposed V&V platform as additional performance testing solutions.

Our initial work [5], [6] on the V&V platform concept focused on high-level ideas and possible architecture realisations without providing concrete results. This article goes beyond this and provides not only a detailed concept, including workflows, entities, and roles, to integrate a V&V platform into the NFV ecosystem. It also presents a case study that validates the applicability and usefulness of our approach and demonstrates how a multi-VNF network service can be tested (functional and performance) end-to-end without any human interaction.

III. A V&V-enabled NFV ecosystem

We introduce the novel concept of adding a *V&V platform* to the NFV ecosystem that offers the service of verifying and validating VNFs or services against a pre-defined or custom set of tests. To do so, a V&V platform uses a test infrastructure

TABLE I: Mapping of V&V concepts to different scopes as shown in Fig. 1.

	static analysis	model checking	unit tests	integration tests	smoke tests	system tests	performance tests	security tests	compliancy tests	stability tests	Example(s)
Scope A	•	•							•		Descriptor validation (syntax and sematics)
Scope B			•		•		•	•			Unit tests to test the rules of an intrusion detection system (IDS)
Scope C				•			•	•			Verify the interoperability (packet output of VNF_i can be processed by VNF_j)
Scope D				•	•			•	•		Testing configuration, management, and monitoring interfaces of a VNF
Scope E							•	•	•		Testing compatibility and performance of a VNF on a specific NFVI
Scope F					•	•	•	•		•	Verifying the ent-to-end deployment of the SFC for given MANO and NFVI

which is similar to the production environment and may be based on different NFVIs controlled by different virtualised infrastructure managers (VIM) and MANO solutions. Having this, a customer of the V&V platform can submit VNFs and services to the platform, the platform verifies and validates them against different test cases in a variety of environments, and finally returns the test results to the customer. An example for this is a VNF that is submitted in one or multiple compliant VNF packages and then tested against different environments, e.g., using OpenStack or Kubernets as infrastructure controlled by different MANO systems, e.g., 5GTANGO, OSM, or ONAP.

This novel V&V platform is operated by a *V&V provider*, tightly integrates into the NFV ecosystem, and adds new roles and business models to it, as we show in Fig. 2. This starts with the *VNF and service developer* who uses NFV-enabled service development kits (SDK) [15] to create new VNFs and services. These VNFs and services might then be uploaded to public catalogues owned by *catalogue operators* to share them with their potential customers, the *service providers*. The *service providers* pick up existing VNFs and services and deploy them into production using service platforms (SP) operated by *SP operators*.

In the "developer-centered business model", we consider the case where the *developer* submits the developed artefacts to the V&V platform (1.1) to have them tested before they are shared (1.3). This also gives early feedback about the compatibility of the developed artefacts to a variety of environments, which the *developer* might not be able to test, e.g., in a lab (1.2). In the "catalogue-centered business model", the *catalogue operator* is, in turn, interested in verifying and validating all artefacts uploaded to the corresponding catalogues. This can be done by first sending the uploaded artefacts to the V&V platform (2.1) and only storing them in the catalogues if all tests have passed (2.2), e.g., to ensure that no malicious VNFs or services enter the catalogues. The "service platform-centered business model" is similar to the second one, but here the *SP operator* has an interest in getting artefacts verified and validated before they are on-boarded to the service platform (3.1). In this model, the pre-testing of artefacts (3.2 and 3.3) mitigates the risk of on-boarding incompatible or broken VNFs and services to a production

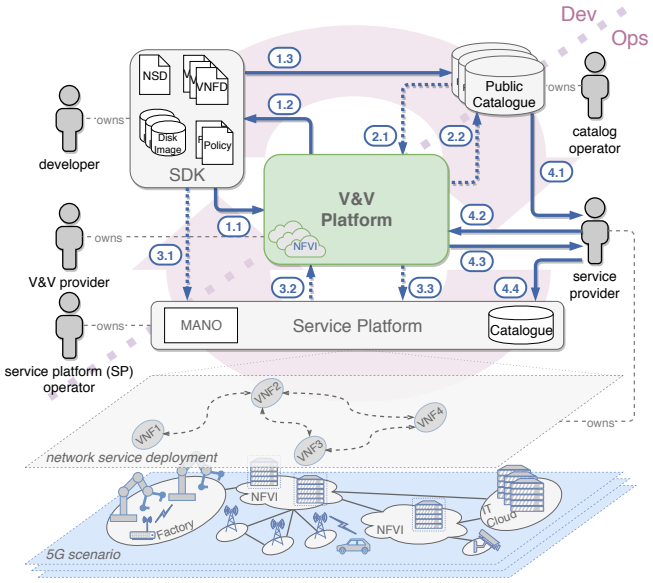

Fig. 2: End-to-end NFV scenario with our proposed V&V platform showing the involved roles and workflows

service platform. Finally, the *service provider* has an interest in using the V&V platform to test third-party VNFs and services, being the fourth model, called "service provider-centered business model". The *service provider* browses the available catalogues and selects the building blocks for his services (4.1). Even though the catalogues might already offer pre-tested VNFs and services, the *service provider* might still be interested in running those third-party artefacts against his own set of tests. He can do this by uploading those artefacts and his custom tests to the V&V platform (4.2), which verifies and validates them and sends them back (4.3). Finally, the *service provider* can decide if those artefacts fulfill his requirements and put them to production (4.4).

The presented V&V platform concept has the benefit that not every party needs to setup own testing infrastructure, which is costly and often not feasible. For example, most VNF and service developers do not have different NFVIs and MANO solutions available. *SP operators* and *catalogue operators*, in contrast, do not want to test new artefacts in their existing production infrastructure. A V&V platform allows them to

outsource these tasks and save resources by using the V&V platform's test resources on-demand.

Besides the potential resource savings, the time required to put new VNFs and services into production can be reduced as well. One reason for this is that *service providers* will know about the compatibility of the deployed artefacts beforehand and time-consuming bug-fixing tasks on freshly deployed services will be reduced. But more importantly, less re-testing and test repetitions are required if we assume that all roles in the described scenario trust the *V&V provider*. This is possible because verified and validated artefacts are annotated with the test results, all signed by the *V&V provider*. Then, every other role in the system can check the integrity of the results and reuse them without requiring new test runs.

IV. Case Study: The 5GTANGO approach

We designed and implemented an open-source V&V platform prototype, which is available as part of the 5GTANGO NFV framework [7], to evaluate the feasibility of the presented concepts. We use it to perform a case study in which we test a virtualised content delivery network service (vCDN) to give the reader detailed insights into the design and workflow of a V&V platform implementation.

A. Building a V&V platform

Fig. 3 shows the internal architecture of the 5GTANGO V&V platform and its surrounding building blocks. It is highly modularised and consists of the following main components that enable a fully automated V&V workflow: (i) The *V&V Gatekeeper*, exposing APIs towards the V&V platform users, allowing them to submit packages for verification and validation; (ii) the *Test Invoker*, responsible for the test case configuration, scheduling, and maintenance of the test state; (iii) the *V&V Catalogues* holding the artefacts to be tested, e.g., VNFs and network services; multiple repositories, i.e. (iv) the *Test Repository*, the (v) *Test Result Repository*, are used to store tests, test results, as well as raw monitoring metrics collected during the tests; (vi) the *Test Engine* responsible to control the execution of tests in the test queue using an extensible set of test plugins. The V&V platform uses the concept of plug-able (vii) *Test Execution Platform Drivers* to abstract and unify the interface towards the test execution platforms on which the VNFs or services under test (SUT) are deployed and the tests are actually executed. Finally, there is a set of tools for (viii) *Test Analysis*.

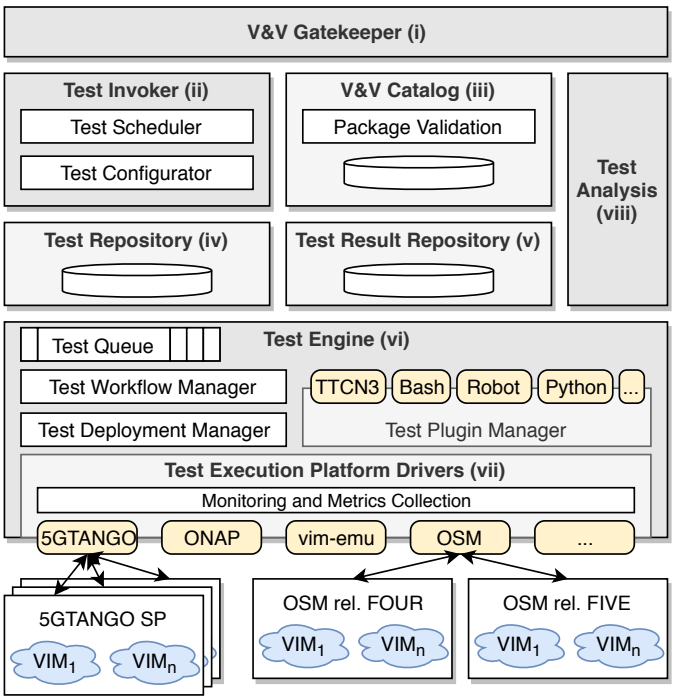

Fig. 3: 5GTANGO V&V platform architecture with several connected test execution platforms

B. V&V platform workflow

The workflow of the V&V platform looks as follows:

1) Test definition and implementation: Tests may be single test cases or a more complex battery of tests, i.e., a test suite. They can either be pre-uploaded to the V&V platform as standalone tests or uploaded side-by-side with a VNF or a service. The latter enables tests that are custom-tailored to a specific VNF or service supporting the business models defined in Sec. III. All tests are uploaded through the V&V Gatekeeper which is also responsible to distinguish between the different roles interacting with the platform, as described in Fig. 2.

A special challenge is the definition and implementation of tests to be executed on a single or multiple V&V platforms. Not to tie our implementation to any specific test definition approach, the 5GTANGO V&V platform offers a test plugin system as part of its test engine. The plugin mechanism utilises container technology, i.e., Docker, to realise test plugins. Containers allow packaging and integration of new test plugins, ranging from simple, script-based tests (e.g., Bash, Python) to more advanced testing technologies, like TTCN-3 or the Robot test automation framework. This extensible and modular design allows to always pick a suitable technology to build tests for all scopes described in Table I.

Each test implementation is accompanied by a test descriptor defining against which types of VNFs and services a test can be executed and which environments are needed. Such a test descriptor can be compared to a *Jenkinsfile*, known from general purpose continuous integration systems. We aligned the test descriptors with the ETSI data models for VNFs and services and published them along with our prototype [7].

2) Test management and execution: When a VNF or service is uploaded, the V&V platform needs to decide which tests should be executed. To automate this decision, we added a tagging system to our test descriptors as well as to the metadata of VNFs and services, which allows to flexibly categorise and match tests. We start with high-level test categories, like functional and performance tests; more detailed categories based on the scopes defined in Table I, down to detailed test categories, like latency tests, TCP throughput tests, and so on. Using the tagging approach, developers can also specify on which target environments a test should be executed, e.g., a

network service should be tested on 5GTANGO v4.1, OSM rel. FOUR and OSM rel. FIVE.

VNFs or services uploaded to the V&V platform are automatically matched against those tags, e.g., a firewall VNF could indicate that it can be tested using end-to-end throughput tests using arbitrary layer 2 traffic. Alternatively, customers of the V&V platform may manually select the set of tests to be executed. All test execution requests are then queued in the test engine and executed once the required testing resources become available. While first in, first out (FIFO) queuing may often be sufficient, more sophisticated queuing mechanisms are easy to realise (e.g., earliest deadline first or prioritising tests for premium users). An optimisation algorithm for this will get the available resources of the connected test platforms as well as the queued execution requests as inputs. It then needs to compute when and where each of the waiting tests should be executed. This creates new research opportunities for the NFV community, since it is desirable that a V&V platform optimally utilises the connected test execution infrastructure while ensuring that deadlines are met and test executions are properly isolated.

To finally execute the tests, the test engine forwards the VNF or network service to be tested (the SUT) to the target test execution platforms. This is done through the test execution platform drivers which abstract and unify the access to different kinds of test platforms. Each test platform offers a particular configuration, e.g., different connected NFVIs, all known by the V&V platform. The test engine then instructs a selected test execution platform to deploy the SUT and may add additional test probes to the deployed service, e.g., traffic generators, to stimulate the SUT during test execution. This clear separation of concerns between V&V and used test execution platform highlights the modularity of the presented platform. In particular, the V&V platform itself does not need to know how to deploy a tested VNF or service. It acts as a meta orchestrator and delegates this tasks to the underlying test execution platforms and manages the test process once the underlying platform has successfully deployed the SUT. This ensures compatibility and extendability to a wide range of VNFs, services, and platforms.

3) Test result collection and management: During test execution, the test engine collects monitoring data from the test execution platforms and stores it in the test result repositories. This is done through the test platform drivers, which not only abstract the control interfaces of those platforms, but also connect to and translate from platform-specific monitoring solutions. Besides the raw monitoring data recorded during test executions, the test results produced by the tests themselves are stored in the result repositories. Those results can either be simple binary *pass*- or *fail*-like results or more complex results, like raw performance metrics, statistical information, or trained machine learning models. This also opens an interesting opportunity for further research: How to best represent and share NFV test results?

Furthermore, the authenticity and integrity of all test results are ensured by using the 5GTANGO package format which is compatible to ETSI's SOL004 [6] specification and allows to sign and store the results. Any other party can then decide which packages to accept (e.g., those which are verified by a trusted V&V and/or created by a trusted developer) by checking their signatures.

C. Verifying and validating a network service

To evaluate the proposed concepts, we use the 5GTANGO V&V platform to test an example service, a vCDN implementation, following the developer-centered business model described in Sec. III. The used vCDN service is a multi-VNF network service with a load balancer VNF (HAproxy) and one or multiple caching proxies (Squid) interconnected to a single SFC. Both VNFs are implemented as VMs and are compatible with OpenStack-based NFVIs. We use the ETSI-compatible 5GTANGO service description format to compose this service, which is then deployed using a 5GTANGO service platform registered as a test execution platform to the V&V.

We use three Dell RX730 servers, each with dual Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz CPU and 128 GB memory, to install our platform and execute the experiments on top of OpenStack Pike (based on OPNFV 5.0). The servers are interconnected by 10G Ethernet links using a Pica8 P-3297 switch. On top of this infrastructure, the V&V automatically deploys the example service to be tested and terminates it once all tests have been performed. In addition to the VNFs of the service, deployed with 1 vCPU and 4 GB memory each, the V&V instructs the test execution platform to deploy two additional VMs acting as traffic source (4 vCPU, 8 GB memory) and traffic sink (16 vCPU, 8 GB memory) for the tests.

We perform three types of tests. First, a series of functional tests, inspired by ETSI TST 001 [4], is used to ensure the correct instantiation and configuration of the service. More specifically, the VNF on-boarding, the VNF instantiation and configuration, the SFC and forwarding graph setup, horizontal scaling, as well as an end-to-end traffic forwarding is tested as shown in the V&V test report in Fig. 4. The report shows that all tests have passed except for the scaling test, which is expected as we intentionally use an example service that does not support horizontal scaling.

Second, a series of performance tests using throughput and end-to-end service latency as metrics is performed. The results of both tests are also shown in Fig. 4. We use the tools *Wrk* as traffic source and *Nginx* as traffic sink. Each test is configured to do 100 parallel HTTP requests, over 30 seconds using request rates between 3,000 and 24,000 requests/s. The results show that the throughput stagnates at about 13,000 requests/s and thus identifies a performance limit if the vCDN service runs with 1 vCPU and 4 GB memory per VNF. This test can be repeated with other resource configurations to learn more about the service's behavior under different resource assignments. The results show how the latency of the service increases under high load and can help developers to optimise it.

Besides the test results, the report in Fig. 4 also shows how the package of the tested service is referenced and its integrity is ensured by using a checksum. It is worth noting that all these tests are performed in a completely automated manner,

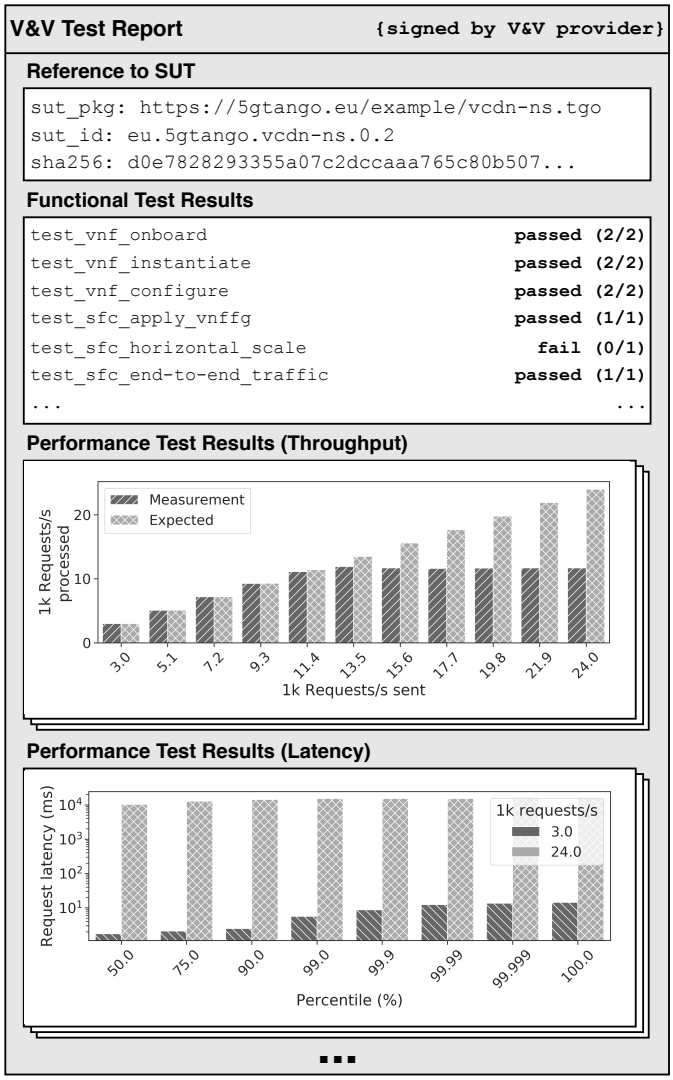

Fig. 4: V&V test report referencing the service under test and showing test results of different test types

without human interaction, after the service developer uploads the example service to our V&V platform.

V. Conclusions and Future Work

Verification and validation plays an important role in future, softwarised networks. It is still a novel discipline and existing solutions mostly focus on small parts of the overall technology stack, which is not enough as the presented analysis of test scopes shows. Using the concept of a trusted V&V platform as part of the NFV ecosystem, we presented the first end-to-end approach for automated verification and validation in the NFV domain, opening the door for new business models and opportunities. Our approach is complementary to most existing testing solutions and allows to integrate them by using modular, plugin-based designs. Further, it can be used to realise real-world implementations of NFV test specifications defined by standardisation bodies.

Following the presented concepts, new research questions about flexible, platform-independent test definition approaches, optimised test scheduling, automated test selection and execution algorithms, as well as generic test result representation formats emerge.

Acknowledgments

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. H2020-ICT-2016-2 761493 (5GTANGO), and the German Research Foundation (DFG) within the Collaborative Research Centre "On-The-Fly Computing" (SFB 901).

References

[1] IEEE 5G Initiative, *5g and beyond technology roadmap*, https://5g.ieee.org/images/files/pdf/ieee-5g-roadmap-white-paper.pdf, Accessed on 11-13-2018.

[2] H. Karl *et al.*, "Devops for network function virtualisation: an architectural approach," *Transactions on Emerging Telecommunications Technologies*, vol. 27, no. 9, pp. 1206–1215, 2016.

[3] B. W. Boehm *et al.*, *Software engineering economics*. Prentice-hall Englewood Cliffs (NJ), 1981, vol. 197.

[4] ETSI GS NFV-TST 001, *Network functions virtualization (nfv); pre-deployment testing; report on validation of nfv environments and services*, Accessed on 11-13-2018.

[5] M. Zhao *et al.*, "Verification and validation framework for 5g network services and apps," in *2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, 2017, pp. 321–326.

[6] P. Twamley *et al.*, "5gtango: an approach for testing nfv deployments," in *2018 European Conference on Networks and Communications (EuCNC)*, Jun. 2018, pp. 1–218. DOI: 10.1109/EuCNC.2018.8442844.

[7] 5GTANGO project consortium, *5gtango development and validation platform for global industry-specific network services and apps*, https://5gtango.eu, Accessed on 11-13-2018.

[8] D. Kozen, "Netkat: A formal system for the verification of networks," in *Proc. 12th Asian Symposium on Programming Languages and Systems (APLAS 2014)*, vol. 8858, Springer, 2014.

[9] M.-K. Shin *et al.*, *Verification of nfv services : problem statement and challenges*, Working Draft, Internet-Draft, Accessed on 11-13-2018. [Online]. Available: http://www.ietf.org/internet-drafts/draft-irtf-nfvrg-service-verification-05.txt.

[10] Cisco Systems Inc., *Third-Party NFV Ecosystem Certification Test Plan v1.6*, Accessed on 24-01-2019, 2018. [Online]. Available: https://pubhub.devnetcloud.com/media/nfv/docs/ThirdPartyNFVEcosystemCertificationTestPlanv1_6.pdf.

[11] J. Pelay *et al.*, "Verifying the configuration of virtualized network functions in software defined networks," in *NFV-SDN*, IEEE, 2017, pp. 223–228.

[12] L. Cao *et al.*, "NFV-VITAL: A framework for characterizing the performance of virtual network functions," in *NFV-SDN*, IEEE, 2015, pp. 93–99.

[13] R. V. Rosa *et al.*, "Take your vnf to the gym: a testing framework for automated nfv performance benchmarking," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 110–117, 2017, ISSN: 0163-6804. DOI: 10.1109/MCOM.2017.1700127.

[14] M. Peuster and H. Karl, "Profile your chains, not functions: automated network service profiling in devops environments," in *2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, 2017.

[15] S. V. Rossem *et al.*, "Introducing development features for virtualized network services," *IEEE Communications Magazine*, vol. PP, no. 99, pp. 2–10, 2018, ISSN: 0163-6804. DOI: 10.1109/MCOM.2018.1600104.

Author Biographies

Manuel Peuster received his MSc degree in computer science from Paderborn University in 2014, where he is currently doctoral researcher in

the Computer Networks group. His research interests are NFV, SDN, and performance benchmarking.

Stefan Schneider received his MSc degree in computer science from Paderborn University in 2017, where he is currently doctoral researcher in the Computer Networks group. His research interests are NFV, SDN, and machine learning.

Mengxuan Zhao joint Easy Global Market as a research engineer in 2015 after her PhD in computer science from University of Grenoble. Her research interests are data management and standardization in IoT, as well as NFV.

George Xilouris holds an MSc in automation systems since 2000. He is fellow researcher at Media Networks Lab, at the Institute of Informatics and Telecommunications at NCSR Demokritos. His current interests are next generation and software networks.

Panagiotis Trakadas is collaborating with Synelixis Solutions as a Project Manager in EU-funded projects. He is also an Associate Professor at TEI of Sterea Ellada. His research interests include routing and virtualization technologies in next generation networks.

Felipe Vicens is a member of ATOS Research & Innovation department Telecom team. He has a long experience in networking, virtualisation and cloud environments. His current interests are in 5G, SDN, and cloud-native.

Wouter Tavernier received his MSc degree in Computer Science in 2002 from Ghent University. He obtained a PhD in 2012 and is currently professor at the same university. His interests focus on performance aspects of SDN, NFV and large-scale routing.

Thomas Soenen obtained his MSc degree in Physics and Astronomy in 2012 from Ghent University. Currently, he is a researcher at IDLab at Ghent University - imec. His interests focus on new network paradigms such as SDN and NFV.

Ricard Vilalta (MSc in telecommunications engineering 2007, PhD in 2013) at the Universitat Politcnica de Catalunya (UPC). He is a senior researcher at CTTC and he is an active contributor in several standardization bodies such as ONF, ETSI and IETF.

George Andreou is senior software engineer at Huawei Technologies Ireland. His interest are big data, machine learning, and SDN/NFV.

Dimosthenis Kyriazis Dimosthenis Kyriazis is an Assistant Professor in University of Piraeus. His research focuses on virtualization technologies in distributed infrastructures, ranging from 5G environments to cloud and edge computing, while also analysing topics related to data management and analytics.

Holger Karl received his PhD in 1999 from Humboldt University Berlin; afterwards, he joined Technical University Berlin. Since 2004, he is Professor for Computer Networks at Paderborn University. His main research interests are wireless communication and architectures for the Future Internet.