

Service Development Kit for Media-Type Virtualized Network Services in 5G Networks

Refik Fatih Ustok, Ugur Acar, Selcuk Keskin, David Breitgrand, Avi Weit, Petros Drakoulis, Alexandros Doumanoglou, Nikolaos Zioulis, Dimitrios Zarpalas, Petros Daras, Francesco Iadanza, Francesca Moscatelli, and Giacomo Bernini

ABSTRACT

Network functions virtualization (NFV) attributes to the substitute of network functions on dedicated appliances such as load balancers and routers with the use of virtualized instances running as software. Any enterprise can easily implement a wide array of network functions using NFV while maximizing efficiencies and introducing new revenue-generating services that are significantly faster and easier than ever before. NFV is a key enabler of the coming 5G infrastructure, supporting the virtualization of various appliances in the network. The scope of this document is to provide a comprehensive overview of the development of NFV-based services, media services in particular, and their specific requirements that are addressed in existing service development kits (SDKs). By supplying an overview of available tools, the development workflow of an example use case on immersive media is demonstrated with an SDK that provides a set of open source tools facilitating the development of media applications.

INTRODUCTION

The telecommunication sector is expecting a phenomenal increase in the universal demand of mobile data volume [13]. This demand can be satisfied by the fifth generation (5G) wireless systems, which aim to improve the performance of networks with respect to latency, throughput, and energy consumption. To do so, the 5G systems require a new network architecture based on network functions virtualization (NFV) and software defined network (SDN) technologies that together create a network architecture that is more efficient, flexible, and programmable, and can support a wide range of devices, applications, and industries. At the foundation of NFV are virtual network functions (VNFs) that can conduct specific network functions including load balancing, firewalls, and so on.

Stimulated by media-intensive use cases, we observe that the function-as-a-service (FaaS) computing paradigm (also known as serverless) is of great advantage to these scenarios. FaaS allows event-driven on-demand VNF instantiation and execution in addition to its seamless elasticity contrary to a traditional virtual machine (VM)

oriented approach, where virtual appliances are continuously running and thus leading to low utilization. With FaaS the VNFs are deployed where they are needed, when they are needed, and for the exact duration for which they are needed, which is of crucial importance for the highly dynamic 5G network environment. Contrary to containers or VMs, the execution unit in FaaS is a function. This neat granularity helps the platform operators schedule the execution faster and makes the applications running in such platforms more responsive to load variations [12]. This responsiveness, together with the ephemeral style of the execution environment, result in more cost-efficient applications.

In addition to the serverless concept, unikernels have also attracted attention from the research community. Since unikernels present only strictly necessary code, they have enhanced security, a much smaller footprint, and far shorter boot time compared to traditional VM-based virtualization.

The future of telecommunication services requires an architecture that neatly harmonizes various VNF concepts (i.e., traditional VM-based, serverless, unikernels). This manuscript mainly focuses on development of media services in highly dynamic 5G networks based on our experience in the 5G-MEDIA project [2], and its objectives are given as follows:

- To introduce the readers to NFV-based service development and requirements
- To provide an overview on all available open-source service development kits (SDKs) to the best of our knowledge
- To present an SDK architecture that harmonizes various VNF concepts including serverless computing with mainstream standards and demonstrate tools for media type service development.
- To illustrate the workflow of the service development with a given use case on immersive media

NFV-BASED SERVICE DEVELOPMENT

Because telecommunication services are increasingly softwarized with the evolution of NFV technologies, telco providers can introduce new telco services much faster with the support of efficient software tools. These software tools must sup-

The authors provide a comprehensive overview of the development of NFV-based services, media services in particular, and their specific requirements that are addressed in existing service development kits (SDKs). By supplying an overview of available tools, the development workflow of an example use case on immersive media is demonstrated with an SDK that provides a set of open source tools facilitating the development of media applications.

	Open-source projects				
Features	SONATA	5G-CITY	5G-TANGO	ONAP	5G-MEDIA
All-in-one UI	No	No	No	No	Yes
Validator	CLI	CLI	CLI	CLI	CLI Web UI
MANO	OSM SON. MANO	OSM	OSM SON. MANO	ONAP	OSM
Emulator	vim-emu	vim-emu	vim-emu	svnm-simulator	vim-emu FaaS-vim
Profiler	Yes	Yes	Yes	No	Yes
FaaS support	No	No	No	No	Yes
DevOps support	Yes	Yes	Yes	Yes	Yes

Table 1. Comparison of the projects' SDKs.

port heterogeneous architectures in highly flexible platforms, while these platforms should support the life cycle of an NFV-based service, from development to operations. The SDK can be considered as a sandbox where service developers can develop, emulate, and test their NFV-based services, integrating development and operational aspects [7]. Development of an NFV-based service requires:

- Creating and validating descriptors, which are deployment templates that describe operational behavior requirements
- Emulating in a realistic network environment
- Injecting test traffic and monitoring change in its operational requirements before its deployment into a production environment

The heterogeneity of media-related VNFs poses a great challenge: to cater for all types of VNF concepts such as traditional VM-based, serverless, and unikernels. The majority of the SDK tools that are available are VM-oriented, and therefore they do not support serverless and unikernels. In the following section we compare existing SDK solutions to gain a better understanding of how they meet the service developers' needs.

COMPARISON OF AVAILABLE SDKS

The 5G research program of the 5G Public-Private Partnership (5G-PPP) has a very wide scope extending next generation telecommunications. The 5G-PPP Phase 1 project SONATA [3] addresses the significant challenges associated with both SDN and NFV by the development and deployment of an orchestration platform for the complex services envisioned for 5G networks and empowered by these technologies. The SDK of SONATA has been used as the baseline to design and implement the SDK of many projects. The Phase 2 5G-PPP projects 5G-TANGO, 5G-CITY, and 5G-MEDIA also propose an SDK. While SONATA and 5G-TANGO [15] proposed a programmable management and orchestration (MANO) framework, all other projects utilized Opensource MANO (OSM). Outside the 5G-PPP, there are also other open source projects offering SDK for NFV-based service developers. One example is ONAP, which is a Linux Foundation project providing a platform for orchestrating and automating physical and virtual network elements, with full life cycle management. ONAP utilizes its own NFV MANO system, which is also considered alternative MANO to OSM. The FP7 project UNIFY provides an SP-DevOps toolkit, aimed at

post-development troubleshooting. Kubernetes, which is a cluster management system aimed at optimizing the utilization of the cluster resources, has been supported as a virtual infrastructure manager (VIM) by many open source projects such as ONAP (since Beijing) and 5G-TANGO. The 5G-MEDIA has utilized Kubernetes to achieve virtualization technology neutrality for the FaaS framework, which further elevates the level of abstraction for developers as further explained later.

Also, to the best of our knowledge, only 5G-MEDIA provides all available SDK tools in a unified user interface, which advances user-time efficiency and provides a smooth user experience. We summarize the comparison of open source SDKs in Table 1. In addition to open source SDKs, there are also commercial solutions available provided by prominent technology companies such as Nokia and VMware. While most of the commercial solutions provide powerful visual composers, VMware's Project Maestro supports multi-vim management and third party integrations.

SDK TOOLS

In 5G, many things, including a platform, infrastructure, and software, will be offered as a service. Service life cycle management (LCM) is a way of managing the life cycle of network services and is enabled by SDK tools. Using SDKs, services can be validated, tested, and reconfigured, or new service versions can be created. SDKs can provide several tools, for instance, validator, editor, catalogue, NFVI emulator, service monitoring and profiling tools, packaging tools, and deployment tools. In this section, we explain these tools as presented in Fig. 1, which together support developing complex media-type services incorporating various types of multiple VNFs [13].

VALIDATOR

The media-type services are VNF chains, and their orchestration realizes their end-to-end functionality [13]. The Service Orchestrator (SO) constructs and maintains each chain using the information in the corresponding network service descriptor (NSD). While the descriptors can be written in various formats, in order to have a MANO-agnostic structure, the descriptors are written according to the European Telecommunications Standards Institute (ETSI) NFV SOL001 specification [4], which is based on the TOSCA format [5] and specifies a novel generalized and extendable format for representing VNF descriptors (VNFDs) and NSDs. While all descriptors are yaml definitions, VNFDs are packaged in the Cloud Service Archive (CSAR) format [5], which specifies the structure of the VNF package. All available SDKs include validation tools to validate descriptors against TOSCA format. Furthermore, while most of these validation tools are offered in Table 1 as command-line interface (CLI), in the 5G-MEDIA project the validator is also presented as a web UI, which could improve the user experience by simplifying the validation process.

PRIVATE 5G APPS AND SERVICES CATALOGUE

Currently available open source SDKs (i.e., SONATA, ONAP, 5G-TANGO, 5G-MEDIA, 5G-CITY) provide a catalogue where descriptors

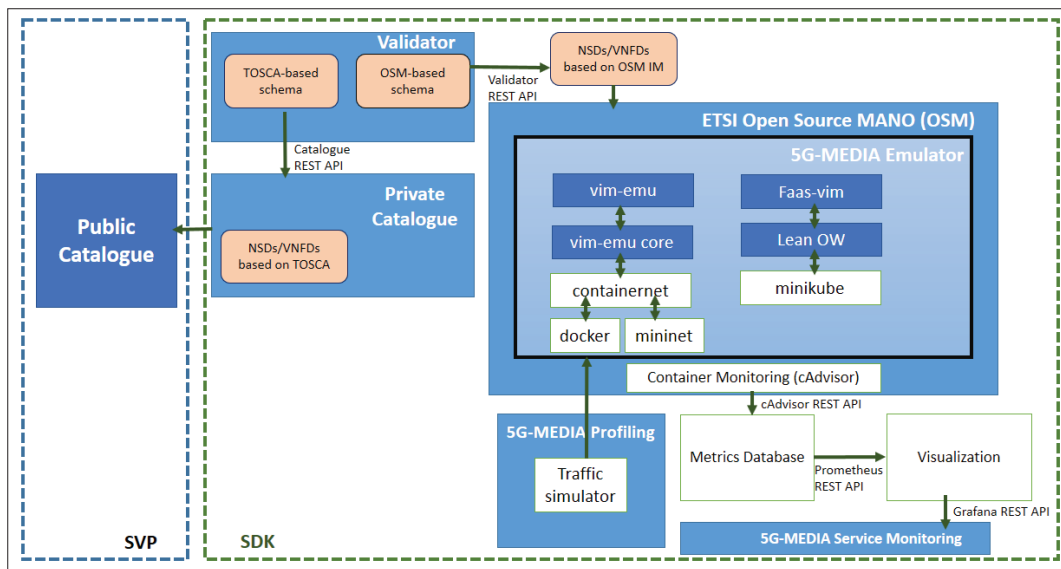


Figure 1. 5G-MEDIA SDK architecture.

of available applications and NSs are stored. The 5G Apps and Services Catalogue, for example, in the 5G-MEDIA and 5G-CITY projects, can be deployed as a public or private instance. In particular, the private catalogue is presented within the SDK, and it provides the application programming interfaces (APIs) and functionalities needed for all the management operations on NSDs and VNF Packages like onboarding, removal, queries, and enabling and disabling specific descriptors or packages. The catalogue North-bound Interface is aligned with the ETSI NFV SOL005 [10] specification concerning NSD and VNF Package management. The catalogue implements a set of MANO plugins that support OSM for the onboarding and deletion of descriptors and packages toward the OSM inner catalogue. Once new applications and services are finalized in the SDK, they can be exported to the public catalogue deployed in the core of the SVP, which stores all available applications and NSDs from all platform users.

NFVI EMULATOR

The NFVI Emulator facilitates local prototyping and testing of real network functions in emulated network topologies running on developers' machines. The emulation platform vim-emu is developed as part of OSM's DevOps to support network service developers in testing their real network functions (non-FaaS), which are packaged as Docker containers. The core of vim-emu is based on Containernet (<https://containernet.github.io>, accessed 21 January 2020), which is a network emulator that supports using Docker containers as hosts in emulated network topologies. Due to the highly dynamic network environment in 5G networks, the NFVI emulator for 5G networks also needs to address the emulation of FaaS applications, which achieve high utilization as they are deployed only where and when they are needed. The emulator of the 5G-MEDIA SDK provides an emulation environment for realistic end-to-end multi-point-of-presence (PoP) scenarios utilizing multi-vim environments [13]. It leverages the vim-emu for non-FaaS VNFs [13] and the FaaS-vim for the

FaaS VNFs [13] to equip the developer with an emulated environment for services in 5G networks. The FaaS-vim is presented in the following section in detail.

SERVERLESS COMPUTING

A cloud-native transformation is happening in the NFV orchestration field, and its impact on the telecom ecosystem is expected to be profound. At the core of this transformation, there is a micro-services architecture with container (e.g., Docker) and container orchestration (e.g., Kubernetes) technologies powering up the microservices approach. The 5G-MEDIA project not only aims to leverage a classic cloud-native approach, but attempts to make a next big step beyond the current state of the art by pioneering FaaS, also known as "serverless," for VNF orchestration for media-intensive applications in 5G networks. Serverless should be broadly understood as a framework that can automatically and efficiently scale an application down to zero in the absence of workload, while providing almost instantaneous event-driven elasticity to handle workload fluctuations. FaaS is a particular approach to implementing serverless technology where a unit of application workload is a function developed in a high-level language and automatically embodied by the framework into a particular runtime (e.g., a container). We argue that FaaS might be a useful paradigm for event-based, session-oriented workloads, such as media-intensive ones. Implementing VNFs using FaaS allows executing them where they are needed in 5G multi-access edge computing (5G MEC), for the duration they are needed, and only when they are needed. Therefore, under some scenarios, FaaS can improve cost efficiency and simplify the DevOps cycle and ongoing operation. Our approach to harmonizing the serverless computing paradigm with ETSI MANO standards is described in [11]. We only briefly recap the motivation and the main principles in this article for completeness, with our main focus being the SDK tools that facilitate the serverless computing devops cycle for typical 5G use cases.

A cloud-native transformation occurs in the NFV orchestration field, and its impact on the telecom ecosystem is expected to be profound. At the core of this transformation, there is a micro-services architecture with container (e.g., Docker) and container orchestration (e.g., Kubernetes) technologies powering up the microservices approach.

It is a good and sound approach to sustainability to integrate a serverless framework with Kubernetes, because this allows for portability, performance, and cost efficiency across different cloud providers. We just need to keep in mind that a single serverless framework can work with multiple container orchestrators and vice-versa.

Serverless computing is a powerful new paradigm greatly facilitating two aspects of cloud-native development. First, it masks complexity associated with deploying application components. A developer develops code of a function in a high-level language, such as Python, Node.js, Go, or Java, and the burden of packaging, instantiating, and de-instantiating this component is shouldered by a serverless framework. Second, serverless frameworks are event-driven by design with functions executing when they are needed, where they are needed, and for the time they are needed (the latter also allows to deliver on a long standing promise of cloud computing to charge customers for actual usage only). Our motivation is to:

- Make these advantages available to VNF developers
- Allow network service developers to mix and match serverless and non-serverless VNFs within a single service that would be orchestrated in a completely standardized way or at least with minimal changes to the existing standards

These are generic problems transcending a scope of the 5G-MEDIA project, so the lessons we learned and the architectural patterns we developed can be carried on to other use cases in the nascent 5G ecosystem.

It is a good and sound approach to sustainability to integrate a serverless framework with Kubernetes, because this allows for portability, performance, and cost efficiency across different cloud providers. We just need to keep in mind that a single serverless framework can work with multiple container orchestrators and vice versa. In 5G-MEDIA, we used a mature open source serverless framework, Apache OpenWhisk (<https://openwhisk.apache.org>, accessed 21 January 2020), that provides a high-level function abstraction to a developer and supports using multiple container orchestrators as back-end where containers — automatically created from these functions — execute. In particular, Apache OpenWhisk supports Kubernetes out of the box.

It is important to differentiate between containers and container orchestrators. Usually, the high-level serverless functions are automatically transformed into containers at runtime (by injecting the code into a predefined container template); then they are orchestrated by container orchestrators such as Kubernetes. Therefore, although serverless functions can use containers at runtime, they are not equivalent because they operate at different levels of abstraction.

Our approach to harmonizing high-level serverless abstraction with ETSI MANO is illustrated in Fig. 1. The 5G-MEDIA platform offers a new VIM, “FaaS VIM,” that implements OSM NBI and communicates with Apache OpenWhisk on the SBI. In turn, OpenWhisk communicates with Kubernetes to offload a serverless function to Kubernetes. In our SDK, we developed a small footprint Apache OpenWhisk, called LeanOW [14], so that it can be delivered to a developer with minimal resources. We use Minikube as the Kubernetes back-end for LeanOW.

ETSI MANO standards originated with VMs. Obviously, there is a semantic gap that we had to bridge between serverless and traditional VM

oriented VNFs to make them seamlessly work together. Here are some observations. The ETSI MANO information model proved to be rich enough that we could use the VNFDs information model and extend it to serverless actions. In fact, the only thing that we had to add was a new type of hypervisor, “FaaS,” to differentiate between FaaS and non-FaaS VNFs. When NFVO (we created a proof of concept with OSM) instantiates a service, FaaS VNFs are instantiated through the FaaS VIM. The FaaS VIM comes with three specialized OpenWhisk actions (transparent to the developer): offload, status, and delete. When FaaS VIM is required to create a new FaaS VNF, the offload action is invoked by the FaaS VIM with the metadata of the VNFD. The offload action uses this metadata to create a Kubernetes pod from the serverless function corresponding to this VNF (the serverless function is pre-created via Apache OpenWhisk CLI). When the pod becomes ready, the VNF record information for this FaaS VNF is updated in the NFVO via the NFVO’s periodic refresh task. When the refresh task executes, it asks FaaS VIM about all previously instantiated FaaS VNFs that NFVO believes to be VMs similar to other VNFs. FaaS VIM invokes the status action and passes back metadata of the container, which executes a serverless function back to the NFVO. When NFVO requires deletion of a FaaS VNF, the FaaS VIM invokes a delete action.

One specific problem with integrating serverless computing with the ETSI MANO standard is Day 1, Day 2 configuration. First, serverless frameworks do not support this out of the box. Second, we found standard mechanisms, such as charms, too VM oriented and too slow for serverless VNFs. To overcome this, we developed a specialized base image that results in two containers per pod when a serverless VNF is instantiated. One container is the serverless function itself, and another container is a private configuration service (implemented as a REST endpoint) for this VNF. The two containers share a volume, and Day 1, Day 2 configuration is pushed to this shared volume via the private configuration service of the base image. A VNF developer is equipped with an entire tool chain to create a serverless function, to instantiate it in the FaaS VIM of the SDK (as in Fig. 1) that offloads it to Minikube, to configure Day 1, Day 2 configuration, to test the VNF performance, and to finally delete it.

SERVICE MONITORING

Various kinds of NS customizations and configurations are required by containerization and virtualization techniques used in 5G networks. Monitoring tools support speedy testing and verification of any modified parameter between the development process and the operations of an application so that any bugs or problems can be avoided before the operations period. Monitored metrics can be collected and consolidated into a local database by service monitoring tools, and these metrics can be queried from the emulated VNFs [13]. After obtaining the metric data in the local database, further analysis can be done to optimize or debug the performance of the monitored VNF or of a service. More detailed information on service monitoring tools can be found in [13].

PROFILING

The profiling tools offered by many open source SDKs listed earlier support load testing under several resource constraints on NSs that are deployed on the emulation environment. During these load tests, a set of various metrics can be monitored, and service developers can benefit from these monitoring metrics for finding bugs, detecting congestion, or investigating issues in their applications. To automate most of this workflow for media-intensive applications, the profiling tool generates one or more test containers to send various types of media traffic depending on the application to be tested. For instance, a 3D media file of pre-recorded content is used to test tele-immersive (TI) applications. These containers can be deployed on the emulator. Figure 2 presents the logical component diagram of the profiling tool. As shown in Fig. 2, the media file to generate load testing and the configurations (i.e., input/output ports, instance count, duration of the media file to be sent) are the inputs of the profiling tool.

DEVOPS AND UNIKERNELS

Unikernels bring many advantages to the NFV domain, where small footprint services can be quickly instantiated and can improve the reactivity of the 5G infrastructure. However, the usual unikernel development process requires deep knowledge of the functionality that is going to be packaged, of the intrinsic unikernel limitations, as well as of the possible specific unikernel API. The OSv catalog provides a long list of native applications [6] ready to be packaged and used; it also supports different runtimes (e.g., Java, Python, NodeJS) that can be used to package new services. In the context of media-type services, this last point is useful whenever a new application/media service has to be developed. Moreover, it fits perfectly into the DevOps within the SDK, as the continuous integration process can be made completely automatic and relies on the capstan tool from the Mikangelo EU project. With the SDK, the developer can ignore unikernels and just select the preferred option for compilation (e.g., with Java 8). Then they can invoke the SDK API to have the code shared on an SCM, with the specific Jenkins task automatically configured, and finally have the unikernel image available in the repository without further effort. Once on the repository, the unikernel image is finally ready for automatic deployment on the catalogue and for the subsequent instantiation orchestrated by OSM.

USE CASE: IMMERSIVE MEDIA APPLICATIONS AND VIRTUAL REALITY

The envisaged scenario of this use case incorporates a teleimmersive interactive multiplayer video game where players' physical appearances are embedded inside a common virtual environment. Players' silhouette and texture coverage is captured using a set of color-depth cameras, with the former being 3D reconstructed according to the method described in [8]. In addition to the small group of players, the system also makes provisions to accommodate a potentially large number of remotely allocated live spectators. These entities may be arbitrarily distributed across a wide geographic area, further pushing the envelope

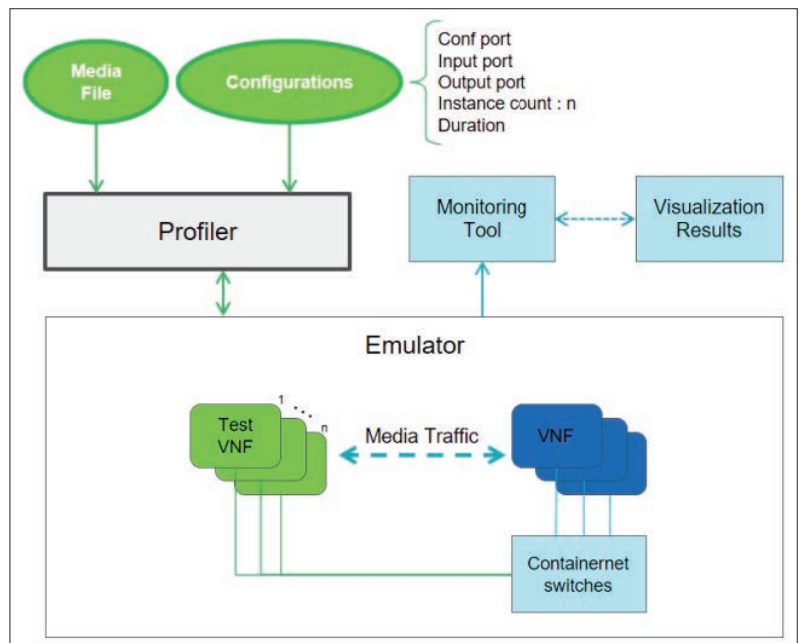


Figure 2. Profiler logical component diagram.

of the underlying infrastructure to include the need for very high bandwidth and near-real-time latency. Different parties may have different visual quality requirements, driven by devices' capabilities, network conditions, or subscription privileges. Thus, the production data streams need to be transcoded into various quality levels at real-time speed. In contrast to a typical 3D content delivery pipeline, the examined prototype utilizes many of the forthcoming 5G mannerisms to mitigate the conflicting nature of certain requirements in a resource-efficient way. The opted architecture steps on the serverless model [1], where transcoding components are built as VNFs. The instantiation of the NS is triggered through a REST call to the OSM which is handled by the FaaS-vim. The VNFs, which are packaged as docker images, are instantiated and deployed as OW actions (i.e., functions) on an underlying Kubernetes cluster. More on the envisaged system's architecture can be found in [9].

WORKFLOW OF EMULATING THE USE CASE ON 5G-MEDIA SDK

Figure 3 presents a regular workflow of media application development using SDK. The workflow comprises six steps, which start with the creation of the VNFDs and an NSD using the validator web UI or exporting them from the catalogue. Then in Step 1, the developer uses the validator web UI to update the descriptors and validate them against the TOSCA schema. If the descriptors are valid, they are onboarded to the private catalogue by simply clicking a button in the validator web UI (Step 2). Having the MANO plugin in the private catalogue, the descriptors are automatically onboarded to the OSM (Step 3) as soon as they are onboarded to the private catalogue. Then in the OSM, the media application is instantiated selecting the FaaS-vim (Step 4). After the media application (vTranscoder) is instantiated in the emulated environment, a traffic simulator is run to test the Lean OW action (Step 5). Finally, after the anal-

ysis of the service metrics in the monitoring tools during the traffic simulation (Step 6), the developer can optimize or fine-tune performance of their applications considering the simulation results.

ANALYZING THE MONITORING DATA

Figure 4 presents the monitoring results of traffic simulation of the given use case where two players play a video game in the emulator envi-

ronment. The specifications of the emulator environment include a Linux machine with Ubuntu 16.04.5 LTS, 1 CPU, 7 GB memory, and 100 GB disk. Please note that the metrics presented in the monitoring tools are not limited to Fig. 4, where we only demonstrate the CPU and memory usage due to page limitation. The figure shows an increase in the OW action CPU usage and memory usage for both vTranscoder VNFs of the application. (There is one vTranscoder VNF assigned to each player, and the players are represented in green and yellow.) As the simulation is completed, the values decrease to their stand-by values.

CONCLUSIONS

In this article, the development of NFV-based media applications and their specific requirements are discussed with an overview of available SDK tools. The development of a network service requires the use of several tools, which are encapsulated in most of the SDKs and explained in detail. The workflow of developing a network service on immersive media has been presented utilizing the 5G-MEDIA SDK, providing information on its profiling and demonstrating its monitoring results. Based on our experience during the project, the SDK provides an environment for rapid development of a service while automatizing the workflow to emulate them in developers' local environments. This allows media application developers to test and verify their service functionalities, and debug and fine-tune them before deploying to a production environment.

In our use cases, we learned that many media-intensive applications are short-session-based. Furthermore, in many such applications, in-app events should be handled on demand. One example is tele-immersive e-sport with short-lived sessions for gaming bouts and in-app events, such as a spectator joining/leaving or an event of interest requiring replay clip creation on the fly. Another example can be drawn from mobile journalism, where a journalist creates a contribution on the fly at the 5G edge with cognitive services such as speech to text for caption creation and image recognition. The cognitive services are applied on demand to a mobile contribution in progress. In these use cases, VNFs embodying the corresponding functionality should be started on demand, for the exact duration they are needed and where they are needed. This is exactly the promise of serverless technology to facilitate such scenarios. Therefore, in this article, serverless transformation of the application development with the FaaS approach and its integration into the ETSI MANO compatible NFV technology has been demonstrated. We argue that serverless computing offers multiple advantages to network service developers, and these advantages will become even more important with the maturing of the 5G ecosystem that would have to cater for highly dynamic workloads deployed on highly distributed infrastructure, where using VMs for every dynamic VNF might be both cost-inefficient and slow. Thus, we believe that future SDKs should support this novel paradigm, and we have shown in this article that it is possible to harmonize this novel model with the mainstream standards.

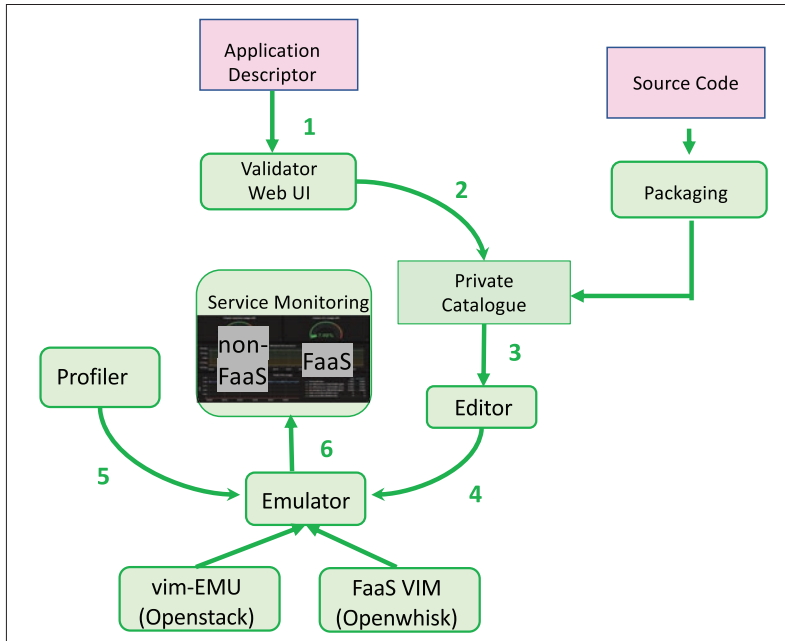


Figure 3. 5G-MEDIA SDK development life cycle.

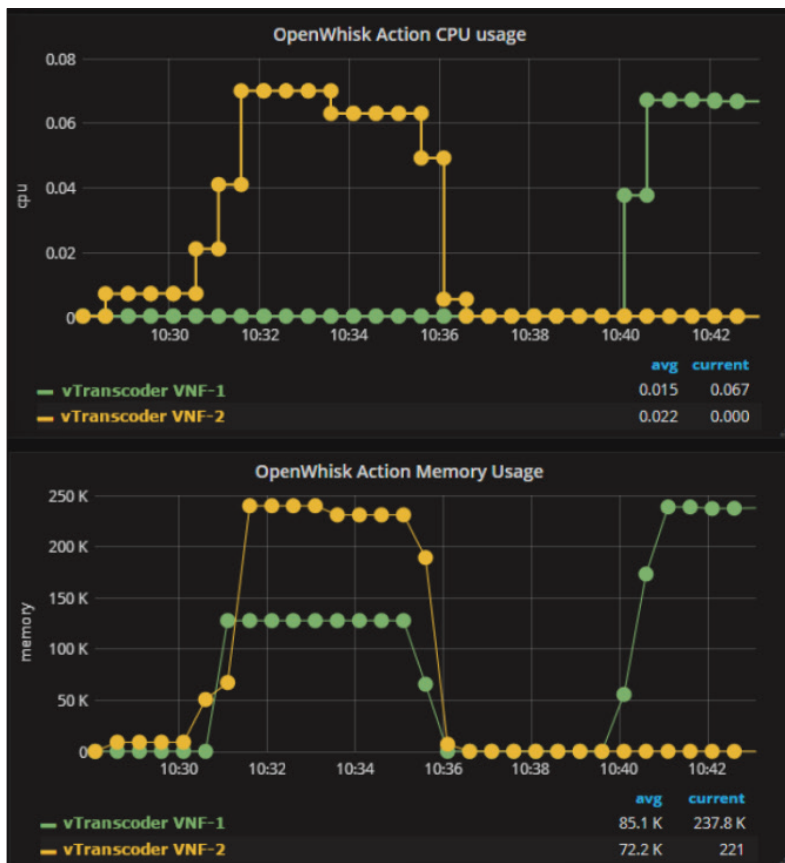


Figure 4. Monitoring graphs of the use case.

ACKNOWLEDGMENT

This work has been carried out as part of the 5G-MEDIA project, which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 761699.

REFERENCES

- [1] D. Breitgand et al., "Towards Serverless NFV for 5G Media Applications," *Proc. 11th ACM Int'l. Systems and Storage Conf.*, 2018, pp. 118.
- [2] S. Rizou et al., "Programmable Edge-to-Cloud Virtualization for 5G Media Industry: The 5G-MEDIA Approach," *IFIP Advances in Info. and Commun. Tech.*, vol. 585, Springer, 2020, pp. 95104.
- [3] S. Van Rossem et al., "Introducing Development Features for Virtualized Network Services," *IEEE Commun. Mag.*, vol. 56, no. 8, Aug. 2018, pp. 184-92.
- [4] ETSI GS NFV-SOL 001 v2.5.1, "Protocols and Data Models; NFV Descriptors Based on TOSCA Specification," ETSI NFV ISG, Dec. 2018.
- [5] OASIS TOSCA "Tosca Simple Profile for Network Functions Virtualization"; <http://docs.oasis-open.org/tosca/TOSCA-SimpleProfileYAML/v1.0/TOSCA-Simple-Profile-YAML-v1.0.pdf>, accessed 3 May 2018.
- [6] OSv apps; <https://github.com/cloudius-systems/osv-apps>, accessed 6 Feb. 2019.
- [7] S. Redana et al., "5GPPP Architecture Working Group: View on 5G Architecture," European Commission, June 2019.
- [8] D. Alexiadis et al., "An integrated Platform for Live 3D Human Reconstruction and Motion Capturing," *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 27, no. 4, 2017, pp. 798-813.
- [9] A. Doumanoglou et al., "A System Architecture for Live Immersive 3DMedia Transcoding Over 5G Networks" *Proc. IEEE Int'l. Symp. Broadband Multimedia Systems and Broadcasting*, 2018, pp. 11-15.
- [10] ETSI GS NFV-SOL 005 v2.5.1, "Protocols and Data Models; RESTful Protocols Specification for the Os-Ma-nfv Reference Point," ETSI NFV ISG, Sept. 2018.
- [11] F. Alvarez et al., "An Edge-to-Cloud Virtualized Multimedia Service Platform for 5G Networks," *IEEE Trans. Broadcasting*, vol. 65, no. 2, 2019, pp. 369-80.
- [12] P. Aditya et al., "Will Serverless Computing Revolutionize NFV?," *Proc. IEEE*, vol. 107, no. 4, 2019, pp. 667-78.
- [13] U. Acar et al., "Programming Tools for Rapid NFV-Based Media Application Development in 5G Networks," *2018 IEEE Conf. Network Function Virtualization and Software Defined Networks*, 2018, pp. 1-5.
- [14] D. Breitgand and P. Kravchenko, "Lean OpenWhisk: Open Source FaaS for Edge Computing," *Medium*; <https://medium.com/openwhisk/lean-openwhisk-open-source-faafored-edge-computing-fb823c6bbb9b>, accessed 6 Feb. 2019.
- [15] T. Soenen et al., "Empowering Network Service Developers: Enhanced NFV DevOps and Programmable MANO," *IEEE Commun. Mag.*, vol. 57, no. 5, May 2019, pp. 89-95.

BIOGRAPHIES

REFIK FATIH USTOK (fustok@netas.com.tr) is a research engineer at Netas Telecommunications. He received a Ph.D. in engineering and computer science from Victoria University of Wellington, New Zealand. His research interests include 5G networks, NFV, and media applications.

UGUR ACAR is a senior researcher and the R&D manager at Netas Telecommunications. He is currently pursuing his Ph.D. at Bogazici University, Turkey. His research interests cover cloud computing and big data.

SELÇUK KESKIN is a research engineer at Netas Telecommunications. He received a Ph.D. in computer engineering from Bahcesehir University in 2017. His research interests cover NFV and cloud computing.

DAVID BREITGAND is a research staff member at the IBM Haifa Research Lab. His research interests include end-to-end performance analysis and management of networked storage systems. He received a Ph.D. in computer science from the Hebrew University of Jerusalem.

AVI WEIT is a cloud research engineer at IBM Haifa Research Labs. His research interests include cloud computing, serverless platforms, and system management.

PETROS DRAKOULIS received his M.Sc. in digital media and computational intelligence from Aristotle University of Thessaloniki. In 2018, he joined the Visual Computing Laboratory of ITI-CERTH, Greece, as a research assistant and software developer. His main areas of interest include visual computing, machine learning, and graphics.

ALEXANDROS DOUMANOGLU received his Diploma degree in electrical and computer engineering from Aristotle University of Thessaloniki in 2009. Since 2012, he has been a researcher with ITICERTH. His main research interests include computer vision, pattern recognition, and 3D graphics.

NIKOLAOS ZIOULIS joined Aristotle University of Thessaloniki in 2012. He has been a researcher with ITI-CERTH since 2013. His interests include 3D processing and graphics, particularly performance oriented real-time computer vision, 3D capturing, and tele-immersive applications.

DIMITRIOS ZARPALAS joined ITI-CERTH in 2007, where he is currently a research associate. His current research interests include tele-immersion applications, 3D vision, and evaluation.

PETROS DARAS is a principal researcher at ITI-CERTH. His main research interests include multimedia processing, 3D reconstruction from multiple sensors, medical image processing, and bioinformatics. He received a Ph.D. in electrical and computer engineering from Aristotle University of Thessaloniki.

FRANCESCO IADANZA is a senior researcher at Engineering Ingegneria Informatica S.p.A. He received his M.Sc. in computer engineering from the University of Sannio, Italy, in 2001. His main research interests are system integration, security, DevOps, and NFV.

FRANCESCA MOSCATELLI is a system and software engineer at Nextworks. In 2015, she received her Italian Laurea degree cum laude in computer science from the University of Pisa. Her research interests include SDN/NFV. Currently, she is contributing in the ETSI NFV Specialist Task Force on OpenAPI specification for NFV.

GIACOMO BERNINI is an R&D manager at Nextworks. His research interests are NFV, SDN, 5G network architectures, and cloud computing. He has an M.Sc. in telecommunications from the University of Pisa.