5G ExPerimentation Infrastructure hosting CloudnativE Netapps for public proTection and disaster RElief

Margetis, G., Valera-Muros, B., Apostolakis, K. C., Díaz Zayas, A., Panizo, L., Tomás, P., Cordeiro, L., Henriques, J. & Stephanidis, C. (2022, December). Validation of NFV management and orchestration on Kubernetes-based 5G testbed environment. In *2022 IEEE Globecom Workshops (GC Wkshps)* (pp. 844-849). IEEE.

DOI: 10.1109/GCWkshps56602.2022.10008690

Validation of NFV management and orchestration on Kubernetes-based 5G testbed environment

George Margetis*, Barbara Valera-Muros†, Konstantinos C. Apostolakis*, Almudena Díaz Zayas†, Laura Panizo†, Pedro Tomás‡, Luis Cordeiro‡, Andre Gomes‡, Joao Henriques‡ and Constantine Stephanidis*§
*Institute of Computer Science, Foundation for Research and Technology-Hellas (FORTH-ICS), Heraklion, Crete, Greece
†ITIS Software, Universidad de Málaga, Málaga, Spain
‡OneSource, Coimbra, Portugal
§Department of Computer Science, University of Crete, Heraklion, Crete, Greece

***Abstract*—Prior work has shown that the integration of Kubernetes orchestration tools with present Network Functions Virtualization infrastructures toward Cloud-based 5G deployments may be the key to unlock beyond 5G communications. However, before we reach that point, further work is required to define, implement and validate practical Cloud-native augmentations that will enable virtual network functions management and orchestration via Kubernetes architectures in existing 5G platforms. In this paper, we present our approach in the context of the 5G-EPICENTRE project, drawing from ETSI-compliant reference implementation frames in order to enhance an experimental 5G testbed with capacity to host containerized network applications. We study how the proposal can be used to validate key performance indicators on high-demanding 5G applications, such as those characteristic of the Public Protection and Disaster Relief vertical.**

***Index Terms*—5G, NFV, MANO, Cloud-Native, Kubernetes**

I. Introduction

The fifth generation of mobile wireless networks (5G) promises both novel and improved capabilities for mobile services. At the core of the 5G evolution lies Network Function Virtualization (NFV) technology, which in turn, relies on elegant management solutions (Management and Orchestration - MANO), for orchestrating the necessary resources to provision network slices in accordance to tenant-specific service level agreements [1]. Through 5G, innovative solutions are enabled, which can address high-demand operations, such as those in the Public Protection and Disaster Relief (PPDR) vertical.

Due to the vital role of NFV and MANO technologies in 5G ecosystems, a common framework is defined by the European Telecommunications Standards Institute (ETSI) [2], specifying guidelines for the practical implementation of the NFV-MANO architecture. Hence, a variety of MANO solutions and projects have been developed, implementing MANO capabilities with support for the deployment of Virtual Network Functions (VNFs) on Virtual Machine (VM)-based virtualization environments and orchestration tools [3].

Recently, enhancements to this architecture have been proposed, most remarkably related to the deployment of VNFs in the Cloud as containers packaging microservices [4]. Although it adds complexity to the NFV-MANO architecture, this shift promises substantial optimizations to ETSI-compliant NFV-MANO systems [5]. Hence, the ETSI GR NFV-IFA 029 [6] updated the NFV-MANO architecture with the following components to support containerized network functions (CNFs):

- The Container Infrastructure Service (CIS), which provides the container runtime environment.
- CIS Management (CISM), which undertakes deployment and monitoring of containerized applications/resources.
- The Container Image Registry (CIR), which stores container images and exposes them to other functions.

There are different approaches in literature to implement the NFV-MANO enhancements to support CISM functionality for MANO of containerized workloads. While a comprehensive list of advantages and disadvantages to each one have been derived, no concrete conclusion has been drawn on the ideal mapping of CISM to the NFV-MANO reference architecture [7]. Kubernetes (K8s) [8], as the prime solution for containerized application management [9], is considered MANO-compliant, drawing concrete parallels to CIS, CISM and CIR through its exposed services and extensible master-worker architecture [7]. Thus, recent works [10]–[14] have explored K8s for fulfilling the responsibilities of the Virtual Infrastructure (VI) Manager (VIM) and VNF Manager (VNFM), while integrating support for CISM toward containerized 5G services provision. Based on gathered evidence, K8s is considered the most likely orchestrator for beyond 5G networks [11].

A key goal of the 5G-EPICENTRE project [15] is to further experiment with such novel 5G architectures. The project targets the PPDR community due to its diversity of end-users, which translates into a variety of KPIs and mission-critical requirements that shall be addressed from a containerization perspective to avoid monolithic solutions that could only fit one type of service. Additionally, providing small and medium sized enterprises and developers with access to 5G and edge computing resources, the project aims at easing the technology adoption by first responders and crisis management teams.

In this paper, we present further evidence of the reconfiguration, adaptation, operation and evaluation of K8s-based 5G network applications (NetApps). We deploy the cloud native infrastructure on top of the NFVI, hence allowing the NFV-MANO underlay to act as a resource orchestrator to

The 5G-EPICENTRE project has received funding from the EU's Horizon 2020 innovation action program under Grant agreement No 101016521.

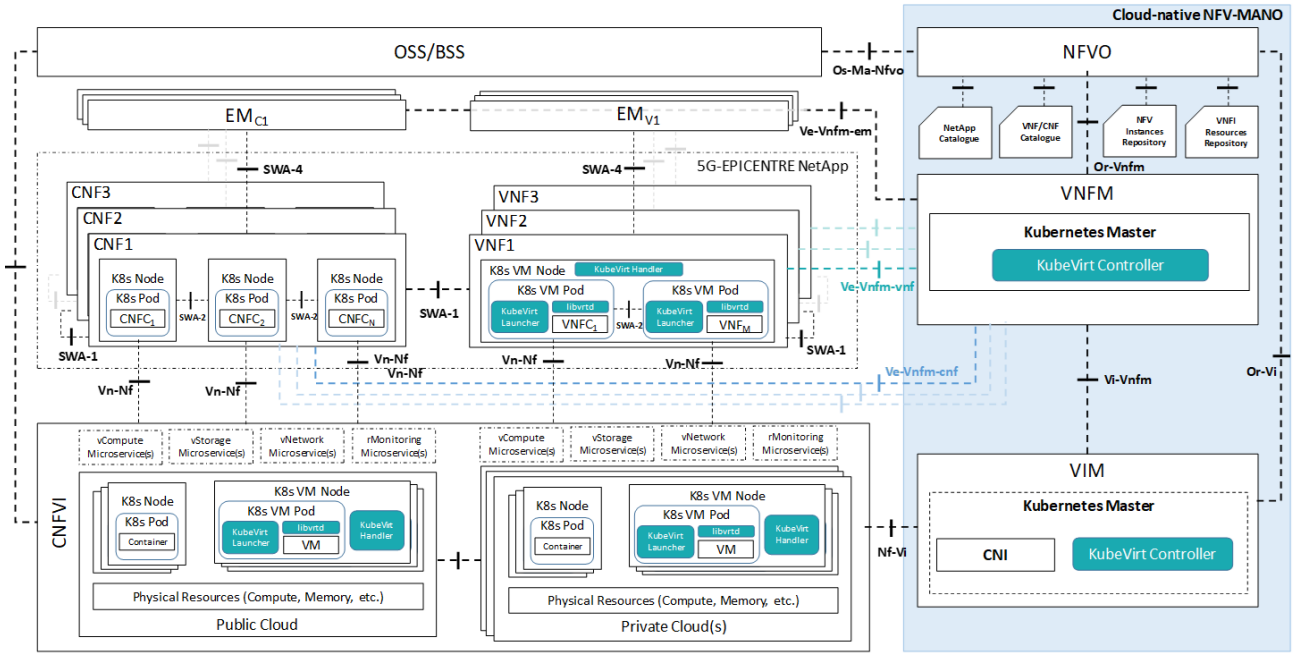

Fig. 1. Reference implementation proposal for cloud-native NFV-MANO with K8s and KubeVirt add-on.

allocate resources to K8s-managed clusters. In this way, both VI and VNF/CNF deployment, scaling and management can be automated to a significant extent. Further, we extend our testbed with the capacity to support MANO of both container-based and VM-based VNFs through K8s plug-ins, allowing the platform to adapt to the needs of heterogeneous work environments. Our implementation hence supports backward compatibility, making it easier to map the NFV-MANO architectural stack, augmented with CISM functionality, to K8s.

The remainder of this paper is organized as follows: Section II describes the conceptual work underpinning the cloud-native augmentation of a real 5G testbed and its implications on the NFV-MANO architecture. Section III describes the pragmatic 5G testbed infrastructure based on a multi-master K8s architecture, along with the extensions implemented for supporting backward compatibility with VM-based workloads. Section IV describes a novel 5G PPDR communications system used to validate the approach via experimentation, which is described in Section V. Section VI presents key performance indicators (KPIs) and discusses results. Section VII concludes our paper with insight on the proposed methodology.

II. Towards a Cloud-native NFV-MANO Ecosystem

Our work focuses on the experimental 5G testbed at the University of Málaga (UMA) campus [16], a platform for 5G experimentation evolved under various 5G Infrastructure Public Private Partnership (5G-PPP) activities, primarily 5GENESIS [17] and 5G-EPICENTRE [15].

Figure 1 depicts the reference architecture for the implementation of the Cloud-augmented NFV-MANO with capacity to host 5G Cloud-native NetApps. The architecture accommodates mixed VM-container workloads through VNF Components (VNFCs) and CNF Components (CNFCs) being orchestrated inside K8s pods on worker nodes that integrate with K8s add-on components for VM life-cycle management inside a K8s environment.

The right part of Figure 1 shows the proposed augmentation of the UMA platform NFV-MANO block implementation, based on preliminary work [18]. Our proposal is able to integrate CISM functionality inside both/either the VIM and the VNFM, using K8s for management and control of both CNFs and VM-based VNFs. This augmentation warrants updates to the reference point for managing the life-cycle of the VNFs/CNFs and the (K8s-based) VNFM.

On the left side, a container cloud platform undertakes the responsibility of infrastructure virtualization (i.e., compute, network and storage) and exposing resource management and microservice support capabilities to the NetApp components. At the NFV-MANO block, a multi-master K8s architecture is followed, with a K8s master node being assigned to both the VNFM and VIM blocks responsible for management, scheduling and deployment of the NetApp container cluster (responsibility of the VNFM), alongside virtualized resource management (responsibility of the VIM).

A. Cloud-native NFV-MANO

In order to benefit from a Cloud-native implementation, a K8s environment is integrated with the existing testbed 5G components. In addition, proper tools and plug-ins are used to maintain backward-compatibility with VM-based deployments, as well as rendering the platform able to support mixed VM-container workloads. Through K8s, containers that package the application code can be orchestrated into pods

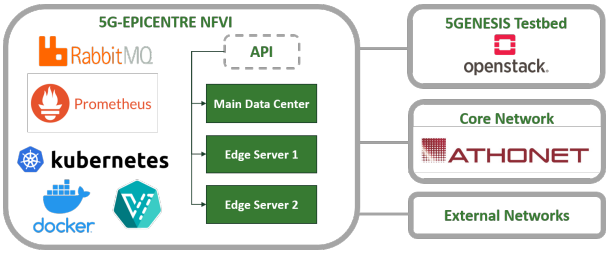

Fig. 2. UMA Platform.

assigned on a number of worker nodes forming a cluster [19]. The cluster can then be managed by one or more master nodes. Since K8s does not support VM orchestration, the KubeVirt VM management add-on [20] is used to enable both containers and VMs to be deployed inside the same cluster or node. The capabilities delivered by this plug-in to K8s-based infrastructures are considered a potential game-changer in the NFV-MANO architecture [21].

B. Cloud-native VNFI

Following the proposed enhancement of the NFV-MANO, A Cloud-native NFV Infrastructure (CNFVI) should be accommodated, including the Cloud resources and containerization layer on top [5]. In addition, VNFs consisting of one or more VNFCs should embrace a microservices-based architecture, which foregoes the one-to-one correspondence between VNFC and VM, and instead adopts a microservice-to-container analogy to form a CNFC, either by building new microservices, or by decomposing VNFs into smaller functional entities deployed as microservices [22]. These novel VNFs should remain aligned to the provisions of the ETSI VNF reference architecture [23]. As such, CNFs should expose interfaces toward the NFVI (Vn-Nf) to access its sliced resources; and the VNFM (Ve-Vnfm-vnf) to facilitate its life-cycle management (via K8s). In addition, CNFs should expose interfaces (SWA-1) toward one another to form 5G NetApps [15], [24].

III. UMA 5G Platform Enhancement

As stated in Section I, we extend the UMA testbed architecture with K8s orchestration to address high-demand operations of 5G applications and technologies, such as those from the PPDR vertical. The UMA platform is used as proof of concept of the feasibility of our containerization approach.

The UMA platform integrates a distributed K8s-based infrastructure with a multi-master multi-node architecture. As shown in Figure 2, this infrastructure is composed of three different physical servers, one for the main data centre and two edge nodes to distribute the services across locations based on the experimenters requirements. This framework follows the reference implementation proposed in Section II, combining the K8s orchestration with Docker and KubeVirt to manage both CNFs and VNFs. For the KPIs monitoring, an instance of Prometheus retrieves data from the NFVI and the core network. A RabbitMQ message broker connects to Prometheus and publishes data retrieved from the experiments via MQTT

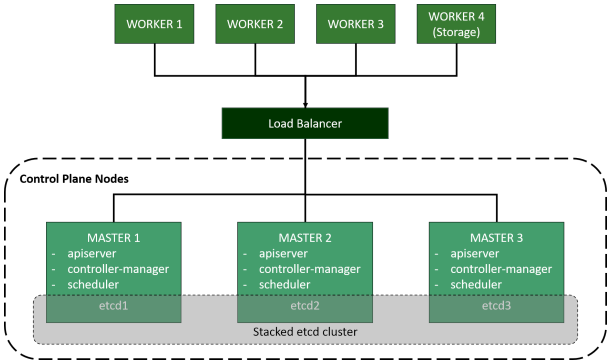

Fig. 3. UMA K8s-based Architecture.

queues. The NFVI deployed is connected to the 5G core network, external networks, and existing 5G resources from the 5GENESIS testbed [17]. The core network contains an Athonet 5G Core (5GC) [25] Standalone instance, with both user and control planes configured, that attaches to the existing 5GENESIS radio resources available at UMA's premises. It is worth noting here that, whereas Athonet's solution runs in our fully virtualized environment, it does not follow the CNF scheme. Although, Sections V and VI provide evidence of the improvements of partially containerizing the platform, such as the reduction of deployment times, which is critical for PPDR. Full containerization, including a container-compliant distribution of the 5GC, will be part of our future work.

Regarding the K8s-based architecture, as depicted in Figure 3, the control plane is composed of 3 master nodes with a stacked etcd topology. The cluster includes 4 worker nodes, one acting as the main persistent storage server. The multi-master approach guarantees high availability, addressing PPDR services' reliability requirements. The architecture integrates a high availability proxy acting as the K8s API, which balances the queries from researchers among the control plane nodes and the worker nodes. Storage is dynamically provisioned, and can be shared among workers to ensure scalability, geographical diversity, and minimal downtime in case of failure.

IV. Mobitrust NetApp

Mobitrust is a situational awareness NetApp supporting Command and Control Centre (CCC) operations to obtain full awareness from the field [26]. It provides capabilities to *Mobile CCC* and *CCC* for monitoring PPDR agents with End-User Devices (e.g., BodyKit (BK) Devices) in the field by monitoring, retrieving and collecting data from different types of sources: agent bio-sensors (e.g., ECG, SpO2, respiration rate), geographical/indoor positioning, internal communication systems, vehicles (e.g., ambulances), devices (e.g., drones), shared services (e.g., private websites or shared folders), as well as real-time text, audio, and video transmissions. Data is then relayed over 5G and processed in the CCC to be displayed in the frontend at the operator's request. Machine Learning algorithms detect critical looming situations (e.g., man-down situations, gunshots, environmental hazards, physical threats,

etc.). Such events trigger alerts, and for each alert, Mobitrust will recommend proactive actions to mitigate its effects. Figure 4 presents the Mobitrust Platform architecture.

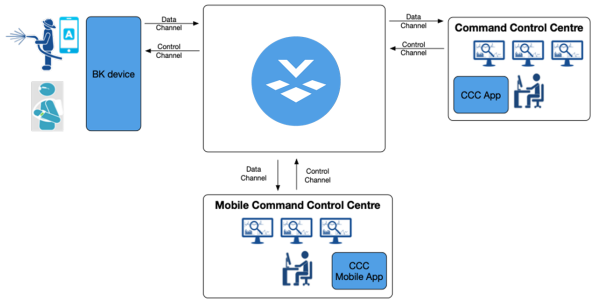

Fig. 4. Mobitrust Architecture.

A. Components (Functional Aspects)

The adoption of individual components loosely coupled as microservices streamlines their management in a Cloud infrastructure. Mobitrust components can be deployed as containers gathering all dependent libraries and configuration files into images. Figure 5 depicts the Mobitrust micro-services.

The **Portal** and **Operational App** are the front-end components fitting desktops and smartphones devices. They provide a suitable user interface for leveraging the interaction with the Mobitrust operators. The **Message Broker** intermediates the communication between **End-user Devices**s and the backend services. **WebRTC Server** is the server supporting **End-user Devices**s streaming video and audio into **Portal** and **Operational App** components. The **End-user Devices** (e.g., bodykit, drones) contain sensors for collecting data and cameras for streaming multimedia to the CCC frontend by means of 5G communication. The **End-User Simulator** simulates an End-user Device (e.g., bodykit) and helps validate the Mobitrust deployment, by providing some behavior on authentication and simulating sensor values, producing messages and standard video. The **Gateway** intermediates

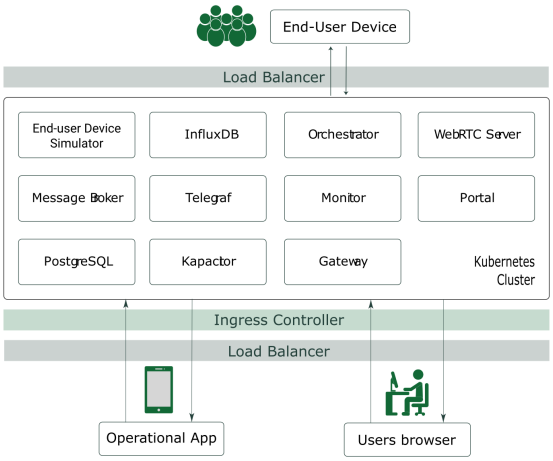

Fig. 5. Mobitrust K8s Deployment

the communication between CCC (**Portal** and **Operational App**) with the operations provided by the backend services. The **Orchestrator** provides the business logic for the running activities by managing users' authentication and authorization and coordinating the communication with **End-User Devices**. **PostgreSQL** is a relational database storing information, such as the users and their access control, as well as, the management information about **End-User Devices** and **WebRTC**. **InfluxDB**, **Telegraf** and **Kapacitor** are parts of the Tick Stack platform providing the capabilities to collect, compute and report Mobitrust related metrics. Those metrics are persisted by keeping **Telegraf** volume mount through out a Persistent Volume Claim (PVC). The **Ingress Controller** maps and paves the way how external users have access to internal services through a Load Balancer component. Finally, the **Monitor** component checks the Mobitrust bodykit availability status and triggers alerts to the CCC in case of unforeseen events.

B. Non-Functional Aspects

The Mobitrust architecture provides non-functional capabilities, streamlining the operation of scaling-out, load balancing, and availability of their components. The security of the application was another important non-functional aspect following some of the best practices highlighted by the 5G-EPICENTRE project [27], with all the services supporting private connections. Moreover, the key settings (e.g., private keys, passwords, users) are encrypted.

V. Experimental Work

The Mobitrust NetApp was deployed in the UMA testbed to integrate the wearable devices with a 5G network; demonstrate and validate NetApp capabilities over a K8s-based NFVI deployment; and collect KPI measurements regarding the platform performance and availability.

A. Kubernetes cluster deployment and 5G setup

Regarding K8s, a full stack of YAML files was developed supporting definition of the different resources, including configmaps, secrets, volume claims, services, deployments, and ingress controllers. Images for each Mobitrust component were pushed into a private Docker registry repository.

Use of a *secret* allowed implementation of optimal security practices and avoid including confidential data in the NetApp code. In this experiment, the Mobitrust TLS certificate were set for all services and ingress of the platform, by specifying the name of the secret. Secrets were defined for authorization purposes to pull images from the Docker registry repository.

The ingress service manages external access to the cluster services, exposing HTTP and HTTPS routes from outside to the application services. Persistent storage, demanded by PostgreSQL and Message Broker components, was accommodated by considering PVC. A configmap gathered the configuration settings for the Mobitrust components, including the name of the database, users, and passwords to access the container.

Deployments were defined by the names of the container images and their labels. The services expose specific ports

```
NAME                                READY   STATUS    RESTARTS   AGE
message-broker-fc65485b7-x42nt      1/1     Running   0          110s
mt-device-65b6946955-6bf2d          1/1     Running   0          33s
mt-gateway-6c77f466c5-g478x         1/1     Running   0          2m32s
mt-kpi-manager-7ff688d8f7-chcg4     1/1     Running   0          31s
mt-monitor-647df94599-w2qlp         1/1     Running   0          41s
mt-orchestrator-6566b465df-vhdbp    1/1     Running   0          46s
mt-portal-5dc7dbdbdf-xkncv          1/1     Running   0          2m8s
postgresql-85dd7c678c-8dsxl         1/1     Running   0          3m11s
tick-influxdb-85f94ff884-5lj9w      1/1     Running   0          86s
tick-kapacitor-5b84494d45-vtqjf     1/1     Running   0          34s
tick-telegraf-5788f79b8b-zhd9q      1/1     Running   0          36s
webrtc-67b444cdd6-ppntq             1/1     Running   0          115s
```

Fig. 6. Mobitrust K8s Deployment

TABLE I
5G STANDALONE SETUP

Band	n78
Mode	TDD
Bandwidth	50 MHz
Carrier Components	1 Carrier
MIMO layer	2 layers
DL MIMO mode	2x2 Closed Loop
Max. Modulation	256 QAM
Subcarrier spacing	30kHz
Uplink/Downlink slot ratio	1/4
Scheduler configuration	Proactive scheduling

for different protocols, which were mapped to the exposed container ports. Some container settings were also defined through the use of environment variables. Figure 6 depicts the running pods as part of the Mobitrust K8s deployment.

B. 5G Setup

The Malaga testbed provides a 5G private network that supports the communication of **End-User Devices**. Table I summarizes the test setup based on a 5G SA deployment, with 4 5GNR TDD cells in FR1 band n78 at 3.5GHz with an associated channel bandwidth of 50 MHz per cell. A 2x2 MIMO with 256QAM modulation enables the selected scheduling configuration to reach 342 Mbps per carrier. The gNodeBs have activated a feature called proactive scheduling that enables the scheduler to generate a configurable number of additional uplink grants and thus, avoid the latency associated with the scheduling request procedure. The average measured latency is in the order of 10-12 ms.

C. Demonstration

The demonstration includes Mobitrust deployed in K8s; and a real user with a bodykit (End-User Device) plugged with a 5G modem, camera, and sensors. After the End-User device has turned on, it authenticates into the platform. The CCC operator connects to the **Portal** component and opens the Dashboard. This action is managed by the **Orchestrator** that pulls a request into **Message Broker** requesting for Standard Definition (SD) Multimedia streaming and sensor data from connected End-User devices while providing the mount point to consume the streaming data back to the **Portal**. Then, the **End-User Device** starts streaming multimedia from cameras to the **WebRTC** server and delivers data from sensors to the **Message Broker**. Finally, the CCC operator can watch video and data from sensors devices summarized in a Dashboard.

VI. EARLY KPIS AND RESULTS

This section presents early KPIs and results from the demonstration work. These KPIs attest to some of the foreseen benefits PPDR end users can expect from the combination of 5G and cloud-native technologies.

Mobitrust Deployment Time is the elapsed time between the K8s Mobitrust deployment starts until the last pod achieves the ready state. Figure 7 presents several measurements (in seconds) along with this demonstration.

Device Authentication Time is the elapsed time from the moment the End-User Device is turned on until the moment it receives the acknowledgment. Figure 8 presents the the measured values (in milliseconds).

Sensor Data Latency is the elapsed time between the moment messages are delivered from the End-User Device until they are received by the CCC (Mobile) Operator. Figure 9 presents the measured values (in milliseconds). Because sensor data arrives from field devices in bulk format, processing is needed in the server (counting towards the KPI value).

Incident Notification Time is the elapsed time between the moment events are identified until the CCC Operator or CCC Mobile Operator are notified. Figure 9 presents the KPI measured values (in milliseconds).

End to End SD Multimedia Latency is the elapsed time from the moment the End-User Device starts delivering SD Multimedia until it is displayed at the CCC Operator or CCC Mobile Operator screen. Figure 11 presents the KPI measured values (in milliseconds).

End to End High Definition (HD) Multimedia Latency is the elapsed time from the moment the CCC (Mobile) Operator request multimedia, until the moment their contents are displayed in the frontend. The encoding and compressing of HD streams is done by more efficient algorithms to the ones used for SD streams, in addition to an automated process that adjusts how the stream is sent to the CCC, taking into account the available resources of the network. Figure 12 depicts the KPI measured values (in milliseconds).

Results denote the platform and NetApp effectiveness to capture the aforementioned KPI measurements, whose values are in line with results expected in such a demonstration. These KPIs will inform tuning of future demonstrations, towards reaching the expected quality of the Mobitrust platform for the target PPDR end users.

VII. CONCLUSION

In this paper, we presented an approach towards a Cloud-native 5G experimentation infrastructure, and studied its implications on high-demand PPDR NetApp deployment. To arrive at this result, both the theoretical and pragmatic dimensions of a K8s-based 5G testbed architecture were elaborated. We hence aligned the architectural design of the testbed to standards and literature documentation, which helped identify a concrete reference implementation frame and practical, elegant solutions for both cloud-native NFV and MANO. In addition, we adapted this architecture to accommodate high reliability and availability requirements that are crucial in PPDR network

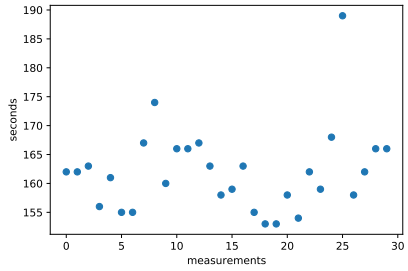

Fig. 7. Platform Deployment Time

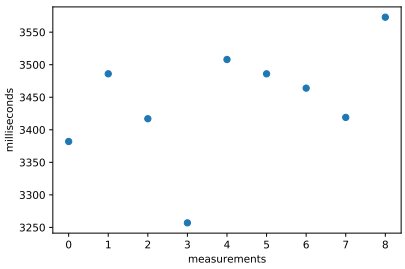

Fig. 8. Device Authentication Time

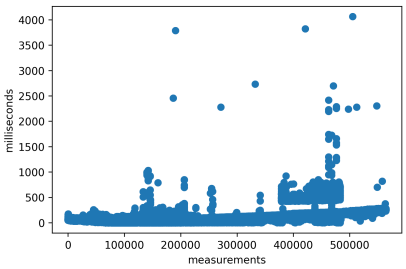

Fig. 9. Sensor Data Latency

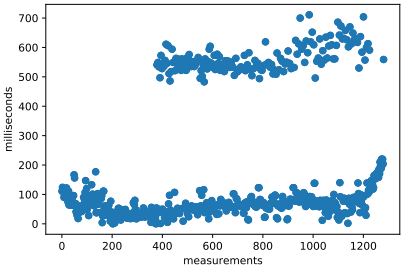

Fig. 10. Incident Notification Time

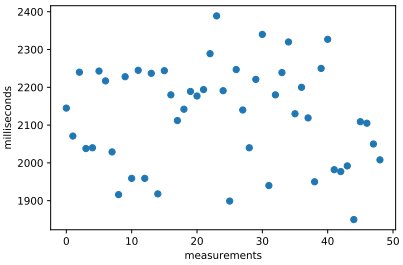

Fig. 11. End-to-End SD Multimedia Latency

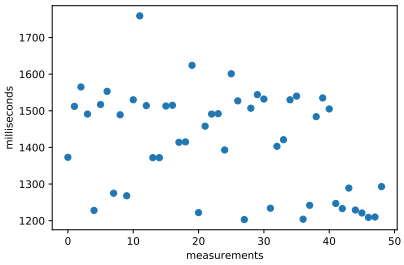

Fig. 12. End-to-End HD Multimedia Latency

application scenarios. Experimentation with such an application validates proper function of the 5G setup, and enables collection of preliminary indicators to the benefits of using 5G and cloud-native technologies for PPDR applications.

REFERENCES

[1] F. Z. Yousaf, M. Bredel, S. Schaller, and F. Schneider, "NFV and SDN—Key Technology Enablers for 5G Networks," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2468–2478, 2017.

[2] "GS NFV-MAN 001 V1.1.1 Network Function Virtualisation (NFV); Management and Orchestration," ETSI ISG NFV, Group Specification, Dec 2014.

[3] G. M. Yilma, Z. F. Yousaf, V. Sciancalepore, and X. Costa-Perez, "Benchmarking open source NFV MANO systems: OSM and ONAP," *Computer Communications*, vol. 161, pp. 86–98, 2020.

[4] S. Imadali and A. Bousselmi, "Cloud Native 5G Virtual Network Functions: Design Principles and Use Cases," in *IEEE 8th Int. Symp. on Cloud and Service Computing (SC2)*, 2018, pp. 91–96.

[5] Y. Chen and A. Bernstein, "Bridging the Gap Between ETSI-NFV and Cloud Native Architecture," in *Proc. SCTE/ISBE Fall Tech. Forum*, 2017, pp. 1–27.

[6] "ETSI GR NFV-IFA 029 V3.3.1 (2019-11) - Network Functions Virtualisation (NFV) Release 3; Architecture; Report on the Enhancements of the NFV architecture towards "Cloud-native" and "PaaS"," ETSI ISG NFV, Group Specification, Nov 2019.

[7] "NFV Release 4 FEAT17 – CNF management concepts." ETSI NFV IFA & SOL WG, Presentation, Mar 2020.

[8] Kubernetes. [Online]. Available: https://kubernetes.io/

[9] L. Abdollahi Vayghan, M. A. Saied, M. Toeroe, and F. Khendek, "Deploying Microservice Based Applications with Kubernetes: Experiments and Lessons Learned," in *IEEE 11th Int. Conf. on Cloud Computing (CLOUD)*, 2018, pp. 970–973.

[10] O. Arouk and N. Nikaein, "Kube5G: A Cloud-Native 5G Service Platform," in *IEEE Global Communications Conf. (GLOBECOM)*, 2020, pp. 1–6.

[11] A. Pino, P. Khodashenas, X. Hesselbach, E. Coronado, and S. Siddiqui, "Validation and Benchmarking of CNFs in OSM for pure Cloud Native applications in 5G and beyond," in *Int. Conf. on Computer Communications and Networks (ICCCN)*, 2021, pp. 1–9.

[12] H. Lim and Y. Kim, "A Design of Service Function Chaining with VNF and CNF on Cloud Native Environment," in *Int. Conf. on Information and Communication Technology Convergence (ICTC)*, 2021, pp. 1467–1469.

[13] J. Lee and Y. Kim, "A Design of MANO System for Cloud Native Infrastructure," in *Int. Conf. on Information and Communication Technology Convergence (ICTC)*, 2021, pp. 1336–1339.

[14] A. Khichane, I. Fajjari, N. Aitsaadi, and M. Gueroui, "Cloud Native 5G: an Efficient Orchestration of Cloud Native 5G System," in *IEEE/IFIP Network Operations and Management Symp. (NOMS)*, 2022, pp. 1–9.

[15] K. C. Apostolakis *et al.*, "Cloud-Native 5G Infrastructure and Network Applications (NetApps) for Public Protection and Disaster Relief: The 5G-EPICENTRE Project," in *Joint European Conf. on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, 2021, pp. 235–240.

[16] A. Díaz Zayas *et al.*, "A Modular Experimentation Methodology for 5G Deployments: The 5GENESIS Approach," *Sensors*, vol. 20, no. 22, 2020.

[17] H. Koumaras *et al.*, "5GENESIS: The Genesis of a flexible 5G Facility," in *IEEE 23rd Int. Work. on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2018, pp. 1–6.

[18] D. Arampatzis *et al.*, "Unification architecture of cross-site 5G testbed resources for PPDR verticals," in *IEEE Int. Mediterranean Conf. on Communications and Networking (MeditCom)*, 2021, pp. 13–19.

[19] J. Shah and D. Dubaria, "Building Modern Clouds: Using Docker, Kubernetes & Google Cloud Platform," in *IEEE 9th Annual Computing and Communication Work. and Conf. (CCWC)*, 2019, pp. 0184–0189.

[20] KubeVirt. [Online]. Available: https://kubevirt.io/

[21] "Edge Computing for 5G Networks." 5G-PPP Technology Board WG & 5G-IA's Trials WG, 5G-PPP White paper, Mar 2020.

[22] S. R. Chowdhury *et al.*, "Re-Architecting NFV Ecosystem with Microservices: State of the Art and Research Challenges," *IEEE Network*, vol. 33, no. 3, pp. 168–176, 2019.

[23] "ETSI GS NFV-SWA 001 V1.1.1 (2014-12) - Network Functions Virtualisation (NFV); Virtual Network Functions Architecture." ETSI ISG NFV, Group Specification, December 2014.

[24] C. Patachia-Sultanoiu *et al.*, "Advanced 5G Architectures for Future NetApps and Verticals," in *IEEE Int. Black Sea Conf. on Communications and Networking (BlackSeaCom)*, 2021, pp. 1–6.

[25] Athonet 5G Core. [Online]. Available: https://athonet.com/products/athonet-5g-core/

[26] J. Henriques, A. Gomes, and L. Cordeiro, "IoT for Improving First Responders' Situational Awareness and Safety on Federated 5G Testbeds." Zenodo, Jun. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6783271

[27] J. Henriques *et al.*, "The 5G-EPICENTRE Approach for Decreasing Attack Surface on Cross-Testbeds Cloud-native 5G Scenarios," in *IEEE Int. Mediterranean Conf. on Communications and Networking (MeditCom)*, 2021, pp. 7–12.