View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Ghent University Academic Bibliography

The Next Generation Platform as a Service

Cloudifying Service Deployments in Telco-Operators Infrastructure

Mimidis, A., Ollora, E., Soler, J.
DTU Fotonik
Lyngby, Denmark
joss@fotonik.dtu.dk

Bessem, S, Roullet, L.
Nokia-Bell Labs
Paris, France

Van Rossem, S.
Ghent University - imec
Ghent, Belgium

Pinneterre, S, Paolino, M., Raho, D.
Virtual Open Systems
Grenoble, France

Du,X., Chesterfield, J., Flouris, M.
OnApp
London, United Kingdom

Mariani, L., Riganelli, O., Mobilio, M.
University of Milano-Bicocca
Milano, Italy

Ramos, A., Labrador, I.
ATOS
Spain, Madrid

Broadbent, A., Veitch, P.
British Telecom
London, United Kingdom

Zembra, M.
Vertical M2M
París, France

Abstract - **5G standard emerges at a particular time in technology history when cloud transforms deeply al-most all industries and services: it becomes obvious that innovations have to be made cloud-native for being successful. 5G must become the ubiquitous fabric blending universal connectivity (to humans, robots, sensors…) with cloud versatility and scalability. For realizing this vision, another model than IaaS must be adopted, the Platform as a Service (PaaS), which should be built to support the telco-grade requirements and combine all sort of third-party applications. These are the core objectives of the Next Generation Platform as a Service (NGPaaS) project, a H2020 5G PPP Phase 2 project. The paper presents the project fundamentals, its architectural proposal and most relevant features.**

Keywords—telco; microservices; 5G; PaaS; DevOps; Dev-for-Operations; SDN; NFV; FPGA; vSwitch; IoT; cloud native

I. INTRODUCTION

With the promise of offering ultra-reliable, low-latency high speed communications, 5G is expected to enable a golden digital age of remote healthcare, autonomous cars and advanced robotics use-cases. 5G heralds an explosion of augmented and virtual reality (AR/VR) applications and accelerates the already rapid growth of the Internet of Things (IoT). But today's mobile networks are not set up in a proper way for handling 5G requirements, without needing extensive over-engineering. To make 5G possible, principles from the more scalable and flexible networks that deliver cloud-based services in IT companies need to be adopted for 5G deployments. This transformation is called *cloud native* [1]. For realizing this vision, another model than Infrastructure-as-a-Service (IaaS) must be adopted, a model derived from the cloud service providers themselves and made by developers for developers, known as *the Platform as a Service (PaaS)* concept [2]. PaaS promises to deliver network services and applications with higher agility and performance through "ancillary services" - scalability, high availability, state management, controllers, orchestrator… - provided once by the PaaS. Developers and service providers can therefore concentrate on their applications and businesses and improve time to market. An ideal 5G PaaS should not only facilitate *building*, *shipping* and *running* virtual network functions (VNFs) with "telco-grade" quality, it should also combine those VNFs with all sorts of third-party applications (from start-ups, free & open source, verticals…) for creating new more versatile and powerful cloud objects, breaking silos between connectivity and computing. One of the requirements to build a PaaS is to adopt the micro-service based architecture. The micro-service approach allows simplifying complicated software systems by breaking them into sub-components and distributing these components across many computing servers. In this approach, an application consists of many small independent

services, and each service is running on its own independent process. The introduction of micro-services in cloud infrastructure supports modularity, flexibility and distributed software components.

II. FROM PAAS TO NGPAAS

The current PaaS ecosystem has never been so rich and diverse, with a number of offerings from public clouds, such as Google Cloud Platform [3], Amazon Web Services [4] and Microsoft Azure [5]. The PaaSes available today offer a wide variety of configurations allowing for a large number of technologies to be utilized, such as containers (using Kubernetes [6] as the orchestrator) and VMs [7]. Beyond offering container or VM based platforms, cloud providers provide with other options such as serverless computing [8] (allowing for the execution of code in the cloud in response to events), cloud storage [9], databases [10] (Often including MySQL and NoSQL variants), load balancers [11], CDNs [12]. It is possible to largely categorize these offerings into broader categories such as Compute, Networking, Storage, Database or Containers for example. Between the major cloud providers, there is often little difference with the services they offer, with the choice between cloud provider often boiling down to the number of available datacenters, support offerings, hosting costs and other commercial decisions. The current drawback with the public cloud PaaS environments described above is that it is not possible to create 'build-to-order' PaaSes. To put another way, the desired PaaS must match the often numerous, but sometimes limited offering that the cloud providers have available. This can often mean that companies wanting to experiment with bleeding edge technologies (Such as Unikernels) will have to deploy their own PaaSes (Either on public or private IaaS) to fill this gap.

In the following sections, we present the main differential features proposed for a Next Generation Platform as a Service.

A. Microservice-based Modularity

Building further on cloud-native design principles [13], we consider a network service to be a chain of VNFs, each implemented using one or more softwarized components, mapped to virtualized compute, network and storage offering resources. The total offered network service is therefore decomposed into *workloads*, microservice-based software components which each process a part of the whole service functionality. For example, a workload typically consists of an application program running in a compute node, where a number of users connect to and interact with the running applications. The modular workload approach is now further extrapolated to the platform itself, which deploys the workloads. The ancillary services offered by the PaaS to build, deploy and run network services must encompass a very broad spectrum of possible virtualization technologies, operational support functions and infrastructure types. To tackle this, we implement also the PaaS as a combination of microservices, and move away from a monolithic locked-in platform, with a fixed set of imposed features. Instead, the Next Generation PaaS (NGPaaS) can be custom composed, choosing more freely the functionalities and technologies needed to support a certain business case.

A two-phased orchestration mechanism is now enabled (as also depicted in Fig. 1):

1. The PaaS-related ancillary services (e.g. deployment, auto-scaling, monitoring framework) are orchestrated to a set of allocated resources.
2. The deployment of a requested workload is now delegated to the pre-deployed ancillary services of one or more PaaS(es).

Our proposed NGPaaS orchestration builds further on the microservice-based platform approach, where both PaaS and workload components are mapped to virtualized resources.

B. Build to Order

For supporting the components modularity feature required to solve the “one size does not fit all” dilemma, the “build-to-order” paradigm is adopted instead. Based on it, those components that best match the target service requirements (business, latency, throughput …) will be selected, as enabled by the platform “composability” feature. A component is an extension of the Reusable Functional Block (RFB) introduced in the 5G-PPP Superfluidity phase-1 project [14]. Building further on previous section, both PaaS ancillary services and user-oriented workloads are now decomposed into RFBs. They can be used as building blocks to create a tailored implementation for any use-case.

C. Build-Ship-Run

An RFB is defined by some metadata and an associated execution environment. The execution environment performs the abstraction between the functions and the infrastructure. The metadata support the deployment flexibility (e.g. composition, recursion). In addition to the Superfluidity definition, we attach additional procedures to the component, termed as BUILD, SHIP and RUN as inspired by the Open Container

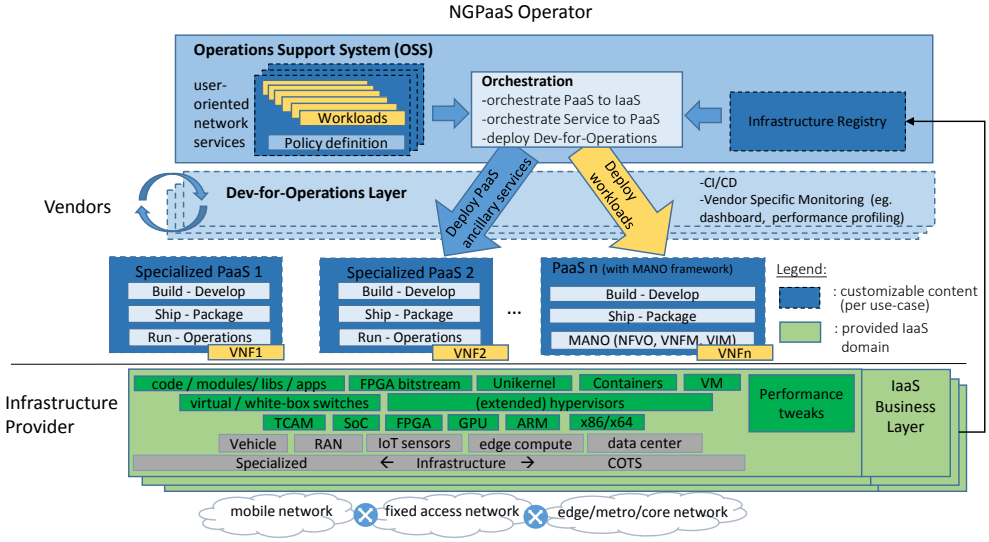

Fig. 1. NGPaaS architecture view. PaaS ancillary services and user-defined workloads are orchestrated to a wide spectrum of virtualized infrastructure. The OSS and Dev-for-Operations layers enable a cloud-based development and operational model.

Initiative [15] following object-oriented programming:

- BUILD: procedure in charge of component creation; it selects the RFB recipe, selects target execution environment among the inventory of all possible targets, connects to source/libraries, and builds the component (compilation, composition).
- SHIP: procedure in charge of securely (source-tracked, encrypted) transferring the component from the "production" site to the "execution" site; it requires security management (signing, hashing, version control), registry storage and synchronization;
- RUN: procedure in charge of deploying and running the component on its final target; it is commonly called orchestration (for placement and execution), service-chaining and monitoring (reliability, self-healing, horizontal and vertical scaling…); it is usually based a *service blueprint* defining its operation. The RUN environment can be an MANO (Management and Orchestration) platform on its own [16].

D. Towards Telco-grade features

New enhancements to usual PaaS infrastructures will be tackled by NGPaaS, as explained below.

1) Hypervisors/Unikernel acceleration extensions

In the telco-grade PaaS it is important to reduce access and transmission latency for network packets. Building the fastest packet forwarding hypervisor platform suitable for virtualizing network functions is conducted during NGPaaS. The MicroVisor [17-18] platform is a lightweight hypervisor that is optimized to process network and storage I/O extremely efficiently. Some low-level improvements to the existing I/O handling logic in the MicroVisor that are planned in the NGPaaS project are the addition of a full zero-copy packet forwarding mechanism between network function service domains and physical network paths. This is done through careful handling of shared guest and hypervisor memory, as well as interaction with the hardware Direct Memory Access (DMA) engines of the supported NICs. This effort also helps significantly to reduce CPU overhead. from the unikernel aspect the MiniOS will be utilized for various tasks, including optimized driver domains and ClickOS-based lightweight virtualized network functions [19]. The MiniOS is customized with the specific driver, relevant for the hardware type that will be controlled, and then spawned as needed for each virtualized resource. Embedding NIC hardware driver support into MiniOS to enable mapping of network interfaces as either a paravirtual VIF (Virtual Interface) or a hardware-based virtual function to a MiniOS domain will also be conducted to enhance the ClickOS framework.

2) FPGA virtualization

The PaaS of the future has to cope with requirements coming from network operators and 5G applications (automotive, IoT, augmented reality, etc.), which demand high performance and programmability as well as very low latency and power consumption. FPGAs fits these requirements very well, and for this reason are today gaining momentum in all kinds of virtualized computing environments (server, cloud, NFV, etc). In NGPaaS, the VOSYS FPGA Virtualization Manager [20] solution will be enhanced with support for Single

Root I/O Virtualization (SR-IOV) paravirtualization [21] and with a dedicated communication mechanism. The former adds the possibility to connect multiple virtual machines to the same accelerator, therefore improving the efficiency and the density of the system The VOSYS FPGA Virtualization Manager communication mechanism [20], on the other hand, enables commands coming from the higher level of the orchestration stack to reach the FPGA accelerators, hence enabling the possibility of remote control, monitoring and management of the accelerated VNFs.

E. SDN Control layer

For the NGPaaS architecture, the SDN Control Layer allows for smooth integration of heterogeneous network infrastructure and the specific business and orchestration layer for the PaaS and grants differential Telco-grade features as well. A set of initial requirements to the SDN control functionality are considered:

- Modularity: the SDN controller entity would ideally be a flexible one, whose composition and features can be instantiated per use case or scenario.
- SDN/NFV integration: issues such as multi controller hierarchy, multi VIM access from a single controller, virtualization of the SDN controller function (as a VNF), controller-based Service Function Chaining are taken into account.
- Hardware acceleration and programmability: the integration of FPGA-based processing units will allow increased data processing speeds and new services. Enhancements to the control plane protocol support in order to access those units from a SDN controller would be an additional desirable feature.
- Policy enforcement: the capability to incorporate programmability features enabling the definition of network behavior (intent networking) would allow to easily map business related network constraints such as those in service level agreements (SLAs) into network (re)configuration thresholds and actions and to automatize its implementation.
- Domain: multi SDN controller coexistence in order to fulfill scalability, availability and resiliency (distribution), delay and consistency constrains together with multi-domain operation (inter and intra slicing).

F. Operational Framework

During the NGPaaS project, the operational support system will be further refined, focusing on a high reliability (aiming at telco-grade five nines [22]). This is done by developing ancillary services in the following areas.

1) Service Performance Profiling

The capability to gather monitored data of deployed services is the base for profiling VNF performance. The existing monitoring framework of a certain specialized PaaS can now be leveraged to support performance profiling: The PaaS requires the ability to plug in any VNF to test, and VNFs that can serve as workload generating traffic source/sink. Using automated configuration and monitoring, a service-specific set of performance metrics is captured and analyzed. While it is clear that 'profiling' builds further upon 'monitoring', it is also worth noting that valid VNF performance profiles can also enhance the operational framework, i.e. the way services are controlled. Current monitored performance can be compared with earlier profiled performance, where any delta can be a trigger for any problem mitigation. As a next step, the obtained VNF profiles will be used as input for service control mechanisms, such as auto-scaling. An illustration of this can be found in [23].

2) Healing

To ensure High Availability, NGPaaS includes a two-step self-healing process. The first step predicts failures and localizes the likely responsible resources by exploiting the KPIs obtained through monitoring. The second step activates countermeasures to prevent or heal failures.

Monitoring works at different abstraction levels simultaneously, so that KPIs about both resources and services can be collected. The self-healing process exploits online analytics techniques to analyze the KPIs and identify anomalies. Machine learning techniques allow to distinguish anomalies, which are symptoms of failures, from noise. Causal relationships among KPIs are exploited to localize the resources likely responsible of the failures. When failures are predicted, appropriate countermeasures are repeatedly activated from a catalog of rules. These map failure-prone situations to actions, until the system is back to normal operation. For example, if the metrics on the percentage of used memory and on the number of requests received per second, by an XDM server, are detected as anomalous, with the former metric increasing while the latter metric decreases, this may be interpreted as a memory issue in the XDM server. Countermeasures may consist in redirecting requests to a redundant server while rebooting the original one.

3) Gradual Software Updates

The NGPaaS framework and the deployed services are envisioned to be largely microservice

based. In this architecture, on-the-fly modification mechanisms such as partial updates or hot-swapping components are very useful enhancements. They can greatly improve the operational stability if executed properly and allow a quicker maintenance cycle and failover procedure. Taking the NGPaaS architecture as reference – see Fig. 1, we now describe the software upgrade process for NGPaaS. There are two main areas that can be upgraded: the platform services, such as the hypervisor or FPGA framework, and application services. Upgrades can be carried out in three different ways: live upgrade, rolling upgrade and cold upgrade. The software update will follow a set of steps including identifying service, gathering current service state, generating upgrade plan, upgrading depending on upgrade plan and additional configuration or rebooting if needed. For different services the customized upgrade plan will be generated. In NGPaaS, we look at upgrading each individual platform on its own, as dictated by the business logic layer. Information about the services provided and the dependencies that the services place on each of the systems will need to be encoded, if upgrade paths are to be dealt with automatically.

4) SLA Assurance

A *Service Level Agreement* (SLA) is a document describing the level of service expected by a customer from a provider, laying out the metrics and thresholds by which that service is measured and the rewarding or penalties, if any, should the agreed-upon levels not be achieved. Traditionally SLAs in the telco market have been defined based on coverage, service availability and/or network metrics such us delay, jitter or packet loss. Within 5G, network services are expected to meet higher real-time QoS demands to support advanced vertical use cases e.g. Ultra-Reliable Low latency (URLL), Enhanced Mobile Broadband (eMBB) or Machine Type Communication (MTC) (for IoT). Current legacy OSS systems do not support that level of dynamicity, so SLAs assurance becomes more complex [24]. SDN/NFV enables certain level of flexibility, by programmability and automated life-cycle management of cloud-optimized network applications [25], but at the same time NFV introduces other challenges to achieve full SLA assurance, since service performance issues may come not only from the network, but also from the cloud/NFV infrastructure, or from the service development [26]. The current work in NGPaaS mitigates this by means of some of the processes explained previously, e.g. profiling, healing, SW updates, and as whole with introduction of DevOps in the telco world and its extension to a Dev-for-Operations model. A new SLA framework is proposed that automates the SLA workflow, from the SLA creation until evaluation and penalties/rewarding process, including automatic triggering of service scaling so as to close the loop and which is integrated with a multi-PaaS monitoring system and Dev-for-Operations layer including profiling tools.

III. NGPaaS Architectural Proposal in Brief

The main goal of the NGPaaS project is to define a new cloud-stack which allows an open collaboration between all roles involved in network service provisioning (such as vendors and service/infrastructure providers). The aim is to move away from a hierarchical cloud stack, where a fixed set of features is imposed to the service provider. In the NGPaaS model, there is no 'one-size-fits-all' solution. The NGPaaS platform should allow an easy collaboration where softwarized PaaSes and services are easy to integrate and couple to a wide infrastructure variety. Cloud-based systems are dynamic by nature: services are provisioned "on the fly" for customers, they react and scale in real-time according to the network state (e.g. migrate between hosts) and resources can be stopped or killed when no longer needed (cattle vs. pets model).

A. OSS Refactoring

We re-architect the OSS (Operations Support System), to cope with the cloud-driven evolution of the business and operational model. For instance, as deployed services move between hosts and scale up and down in response to workload variations, collection of resource utilization and reporting to the billing function of OSS should be able follow, in order to provide to the user with a "pay for what you use" or "pay as you go" billing model. Generally, current OSS models lack dynamic service configuration and hold parameters which are not assumed to change. The NGPaaS OSS model is refactored to support:

- Policy driven real-time service variation.
- The ability to respond at packet/flow timeframe.
- The ability to abstract and model services.
- The ability to manage micro-services.

B. Dev-for-Operations

The DevOps methodology [27] is primarily adopted in the IT industry to realize a closer and faster collaboration between development and operation teams within a single organization. In a

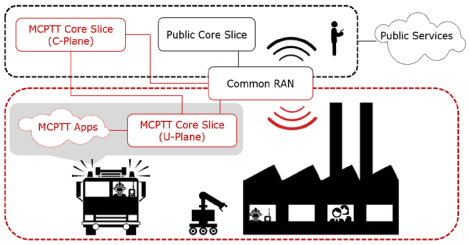

Fig. 2 NGPaaS 5G PaaS demonstrative use case.

telco-grade context, we aim to extend this "in-house" flow to a wider range of stakeholders. The target is to enable multi-sided interactions between the NGPaaS Platform itself and the main actors affiliated to it: the vendors supplying the SW components and the third party entities using those components to implement their specific services (e.g., virtual operators or vertical market players) [28]. The goal is to apply the DevOps methodology, but not only in the scope of a single organization, but also considering that work-teams can develop their activities in different administrative domains as usually happens in a telco-grade context. This is what we call "Dev-for-Operations". This has significant implications not only regarding the way the SW development, deployment and monitoring processes are implemented; also other aspects such as the smooth integration of components coming from different vendors, IPR management and, of course, the security aspects to enable a secure access to all the participant entities should be considered. Also, beyond the IT specific requirements usually associated to DevOps, it will be also necessary to support the specific telco-grade requirements (e.g., very strict Fault Tolerance and High Availability) in order to bring the agility, scalability, cost reductions and reliability from the IT DevOps approach to the new 5G telco paradigm [22]. To implement this, the NGPaaS architecture provides a customizable Dev-for-Operations layer, tailored to each vendor allowing custom access and execution rights. This way, the Dev-for-Operations layer serves as an interface for each vendor to implement a DevOps cycle with the NGPaaS Operator. Also, specific Dev-for-Operations instances managed by the NGPaaS Operator itself will be used to perform the multi-vendor integration functions. These layer instances can be deployed to allow customizable access to the PaaSes where the services are deployed. For instance, one important feature in this layer are the CI/CD components to enable the automatic integration and delivery of the SW components after the validation tests have been automatically executed. Another important feature is to expose monitoring data to vendors (the data could be filtered by the operator so no sensitive or irrelevant information would be provided). Also, as shown in Fig. 1, the OSS and Dev-for-Operations implement and further enhance the microservice-based orchestration process described earlier in this paper.

IV. PRACTICAL EXAMPLES

To highlight the flexibility and adaptability of the presented Build to Order principle in NGPaaS, allowing a customized PaaS to fullfil the requirements of specific business cases, three different PaaS targets are being considered by NGPaaS: Telco, 5G and IoT. In them, not only conventional Telco services, but also Vertical industries ones are targeted, on their one or as mixed compositions.

1) Telco PaaS

NGPaaS Telco PaaS focuses on demonstrating the NGPaaS principles for Telco-grade services with streamlined latency and availability as main differentiated indicators. Besides these main two, other aspects characterizing the NGPaaS Telco PaaS service are: Automatized Management, Security, Integrity, Resilience and Operation integrity. The NGPaaS consortium decided on a Virtual Network Function as a Service (VNFaaS) as demonstrative use case for the NGPaaS' Telco PaaS: a Telco provider delivers wholesale Layer 2 connectivity to Content Providers (CoPs), primarily focusing on end-to-end connectivity between the CoP and its customers. By levering the NFV and SDN paradigms, the Telco Provider allows the CoP to onboard and administrate specific VNFs (e.g. vRouter, vFirewall) on specific points of presence (PoP) between the CoP and its customers. The Telco PaaS will be based on the Central Office Rearchitected as a Datacenter (CORD) platform [29], with XOS [30], OpenStack, and ONOS as its individual components. CORD was selected as the base platform, since it is designed as an NFV-based substitute for the Central Offices (COs) of Telco providers, role very close to the aforementioned PoPs.

2) 5G PaaS

Exploiting an NGPaaS platform will enable service providers to deploy PaaSes to support mobile 5G with ease. One example that NGPaaS is going to explore is the use of this for MCPTT (Mission Critical Push-to-Talk) on a 5G Network (see Fig. 2). MCPTT is a service used by emergency services to allow them to provision and utilise a network slice to allow for reliable, high quality, high bandwidth and secure communications, without the fear of any outside impact from other network users. These communications will extend beyond just radio and

voice communication (such as adding temperature sensors / video in the fire brigades equipment).

3) IoT PaaS

NGPaaS architecture is particularly fitted to support IoT needs. In order to demonstrate the high flexibility and adaptativeness of NGPaaS towards IoT, one of the considered demonstration cases for IoT PaaS is IoT4Energy. This Use Case aims at showcasing both massive IoT for metering services (smartmetering & smartbuilding) as well as mission-critical IoTs such as Demand/Response (for smartgrid); on-demand applications can be automatically E2E (end-to-end) provisioned and activated by different stakeholders (such as Utility Companies, Facility managers or Property managers), together with the relevant KPIs for both devices and applications, taking into account application-based SLAs and device-based SLAs. IoT devices will be connected using either 5G or LPWA technologies. A dedicated IoT infrastructure (including IoT BaaS, Business as a Service, and IoT PaaS, Platform as a Service), based on an evolution of the CommonSense IoT Platform [31], will be deployed within the overall NGPaaS resources for control, orchestration, operations and security.

V. CONCLUSION

To avoid 5G being an isolated industry providing basic connectivity for the cloud applications and services boom, we propose to build the Next Generation PaaS (NGPaaS). The NGPaaS envisages 5G as: a build-to-order platform, with components, features and performance tailored to a particular use case; developed through a "Dev-for-Operations" model that extends the IT industry's DevOps approach to support a multi-sided platform between operators, vendors and verticals; and with revised Operational and Business Support Systems (OSS/BSS) to reflect the new parameters and highly dynamic environment. NGPaaS can enable 5G to become central to a cooperative future with cloud developers, by removing the technological silos between the telco and IT industries and has the potential of unleashing the NFV promise for a true revolution on the way: i) operators manage their network and offer their services, and (ii) vendors design and architect their applications.

Acknowledgement: This work has been performed in the framework of the NGPaaS project, funded by the European Commission under the Horizon 2020 and 5G-PPP Phase2 programmes, under Grant Agreement No. 761 557 (http://ngpaas.eu).

REFERENCES

[1] https://www.cncf.io/ , fetched on January 2018 (foJ2018).

[2] S. Kolb, C. Röck, "Nucleus - Unified Deployment and Management for Platform as a Service", Otto-Friedrich-Universität Bamberg, 2016.

[3] https://cloud.google.com/products, foJ2018.

[4] https://aws.amazon.com, foJ2018.

[5] https://azure.microsoft.com/en-gb, foJ2018.

[6] https://cloud.google.com/kubernetes-engine, foJ2018

[7] https://azure.microsoft.com/en-gb/services/virtual-machines, foJ2018.

[8] https://aws.amazon.com/lambda, foJ2018.

[9] https://cloud.google.com/storage, foJ2018.

[10] https://cloud.google.com/sql, foJ2018.

[11] https://azure.microsoft.com/en-gb/services/load-balancer foJ2018.

[12] https://azure.microsoft.com/en-gb/services/cdn, foJ2018.

[13] Adam Wiggins "The Twelve-Factor App" [online] Available: http://12factor.net/, foJ2018.

[14] Superfluidity, "Deliverable D3.1: Final system architecture, programming interfaces and security framework specification', December 2016.

[15] https://www.opencontainers.org/ , foJ2018.

[16] ETSI –NFV, Management and Orchestration. ETSI GS NFV-MAN 001 V1.1.1 , Dec 2014

[17] P. Barham, et alt.. "Xen and the art of virtualization." ACM SIGOPS Operating Systems Review 37, no. 5 (2003): 164-177.

[18] X. Ragiadakou M. Alvanos J. Chesterfield J. Thomson M. Flouris "Microvisor: A scalable hypervisor architecture for microservers" 2015.

[19] J. Martins, "ClickOS and the Art of Network Function Virtualization", Proc. NSDI 2014, pp. 459-73, 2014-Apr.

[20] M. Paolino, S. Pinneterre, S.D. Raho. "FPGA virtualization with accelerators overcommitment for Network Function Virtualization", 2017 International Conference on Reconfigurable Computing and FPGAs.

[21] PCI-SIG speccs group. https://pcisig.com, fetched on 12th January 2018.

[22] F.J. Ros, P. M. Ruiz, "Five Nines of Southbound Reliability in Software-Defined Networks", ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN 2014), USA, 2014

[23] S.Van Rossem,W. Tavernier, D. Colle, M. Pickavet, P. Demeester." Introducing Development Features for Virtualized Network Services." IEEE Communications Magazine. In press.

[24] TMForum whitepaper: "OSS of the Future" (2017).

[25] A. Abdelwahab, B. Hamdaoui, M. Guizani and T. Znati, "Network function virtualization in 5G", in IEEE Communications Magazine, vol. 54, no. 4, pp. 84-91, April 2016.

[26] R. Mijumbi, et alt. "Management and orchestration challenges in network functions virtualization. Communications Magazine, IEEE. (2016).

[27] G. Kim, J. Humble, P. Debois and J. Willis, "The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations", IT Revolution Press, October 2016

[28] M. K. Weldon,'The Future X Network: A Bell Labs Perspective', Chapter 13, March 2016

[29] https://opencord.org/, foJ2018.

[30] https://wiki.opencord.org/display/CORD/XOS%3A+The +CORD+Controller, foJ2018.

[31] http://www.vertical-m2m.com/commonsense , foJ2018.