

NGPaaS Framework for Enriched and Customized Virtual Network Functions-as-a-Service

Paul Veitch¹, Adam Broadbent¹, Angelos Mimidis², Jose Soler², Marco Mobilio³, Alessandro Tundo³, Michell Guzmán³, Steven Van Rossem⁴,

¹BT, ²DTU Fotonik, ³Unimib, ⁴Ghent University-imec,

Abstract— This paper describes how the novel Next Generation Platform-as-a-Service (NGPaaS) framework can facilitate major benefits for Network Operators and Vertical Service Providers (VSPs) who wish to leverage Virtual Network Functions-as-a-Service (VNFaaS) capabilities. Network Operators can benefit by providing an on-demand PaaS with required features for the VSPs, thus generating new revenue streams but with low operational overhead due to the high degree of automation. VSPs can benefit from the PaaS-oriented approach, by being able to flexibly on-board new VNF types and “value-added” service capabilities like monitoring, healing and profiling, to deliver customized service blueprints to meet the needs of their end customers. The paper outlines the design of an early prototype, built on the Open-CORD platform and using industry-standard Virtualised Network Functions (VNFs).

Keywords- NGPaaS, 5G, VNFas-a-Service, Monitoring, CORD.

I. INTRODUCTION & RELATED WORK

The Horizon 2020 Phase 2 “Next Generation Platform as a Service” (NGPaaS) project[1] has made significant progress in developing a novel and unique approach to building and customizing PaaS(es) “on demand” to meet specific requirements of distinct vertical use cases. Many of the use cases arising from vertical segments like Fixed Telco, Internet-of-Things (IoT) and Mobile 5G require specific attributes such as “Telco-grade” monitoring, reliability and data-plane performance. These do not inherently exist within “State-of-the-art” PaaS solutions. NGPaaS has been addressing these shortfalls by enabling enhancements and customizations to existing PaaS(es) to support the rich and diverse range of requirements across as wide a spectrum of use cases as possible[2].

An increasingly evident trend that will become an integral part of future user-centric 5G platforms and systems will be the dynamic and flexible nature of service management. This is considered in terms of available features, ongoing version control and maximized technology choice to avoid vendor lock-in. NGPaaS addresses this ambition by including a novel “Dev-for-Operations” layer within the baseline architecture (Fig. 1). Dev-for-Operations is an extension of DevOps concepts and principles to enable distinct stakeholders in the eco-system - such as Vertical Service Providers (VSPs) and Software Vendors- to participate in service on-boarding and in-life management. These interactions should work *across organizational boundaries*, rather than between teams within the same organization, as is already the case with traditional DevOps practices[3].

Further detail on the spectrum of use cases that can be catered for in NGPaaS, is provided in [2]. In this paper, we focus specifically on a wholesale business use case for Telco Operators and their enterprise customers, called Virtual Network Functions-as-a-Service (VNFaaS)[2]. At a simple level, this use case enables the hosting of VNFs (routers, firewalls, inspection devices, etc.) on NFV Infrastructure (NFVI) owned by Telco operators. VSPs can leverage this hosted capability, precluding the need to deploy specific hardware appliances on the premises of their end customers. Going beyond a simplistic “Infrastructure-as-a-Service” (IaaS) approach to supporting the VNFaaS use case, the paper clearly demonstrates how a PaaS approach – as afforded by the NGPaaS framework – can provide significant benefits, including but not limited to:

- Ability to flexibly on-board a wide range of selected VNF types and other Value-Added Services (VASS).
- Ensure a wide a range of service combinations (e.g. defined and catalogued as “blueprints”) can be flexibly defined to meet the needs of different end users.
- Ability to incorporate Telco-grade enhancements to baseline service on an “as-needed” basis, e.g. monitoring, alerting, healing, scaling, etc.
- Increased automation for lifecycle management, helping to meet Key Performance Indicators (KPIs) for reduced service launch times, manual operations, etc.

All of these benefits will be equally applicable to the more generic implementations and guises of Network Function as a Service, as outlined in the ETSI Industry Specification Group Use Cases[4], and extensible to other 5G use cases[5]. An obvious question that arises is whether the VNFaaS use case, along with other Telco, IoT and 5G use cases can be supported on *existing* PaaS solutions in the marketplace, i.e. does the state-of-the art not facilitate these requirements already? Put simply the answer is no. As explained in [1,2], existing public cloud providers like Microsoft Azure, AWS and Google Cloud certainly have an array of platform features and tools, but these are strongly oriented towards streamlining web and IT software development. They do not readily cater for “Telco-grade” use cases, and in any case no single PaaS solution would readily match the diverse range of requirements across the spectrum of Telco, IoT and 5G vertical markets.

Another possible approach is to follow the ETSI reference architecture which adds a single PaaS layer above NFVI[6], however this could prove restrictive and inflexible to meet the required “dynamic” nature of many of the use cases addressed

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>).

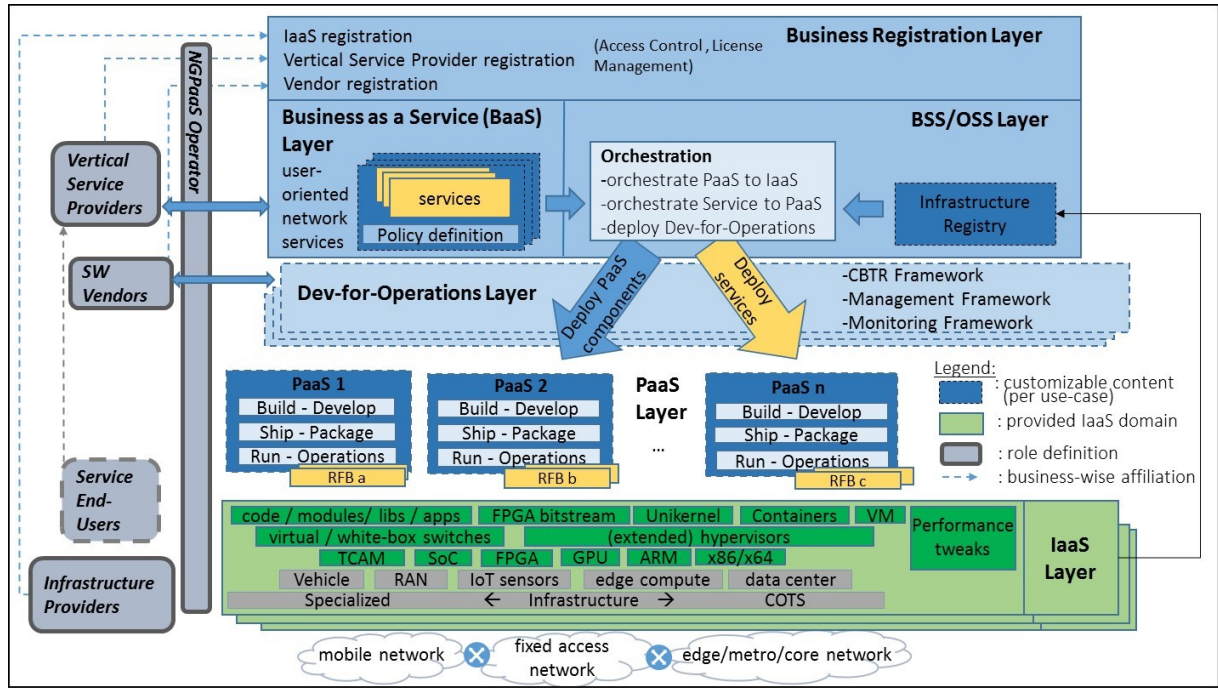


Fig. 1. The NGPaaS Baseline Architecture.

by NGPaaS (i.e. building to order) as well as meeting all the requirements as already highlighted. NGPaaS has therefore adopted the principle of build-to-order, and where necessary customizing, stand-alone PaaS(es) to meet the needs of those distinct use cases. The baseline PaaS is carefully selected to match as closely as possible some of the key feature requirements needed by the use cases, but with the necessary extensions to fine-tune and meet KPIs of the use cases. Two contrasting examples are shown in Table I.

TABLE I. Telco PaaS and 5G PaaS Examples

PaaS	Services Supported	PaaS Extensions/KPIs	IaaS
1) Telco PaaS using Open Cord [7]	VNF-as-a-Service	-Adapt Open-Cord to generically support VNFaaS service descriptors (minimizes need to re-build if new VNFs introduced). -Integrate value-added functions like platform & service monitoring.	Private Cloud NFV Infrastructure
2) 5G PaaS using Kubernetes [8]	Mission Critical Push to Talk (MCPTT)-Emergency Services	-Adapt to support "Telco-grade" features like service chaining, and data-plane performance guarantees needed for 5G instantiation of MCPTT end-to-end network slice.	Hybrid: Private Cloud (for access) and Public Cloud (for core)

Both of these examples have been built as prototypes to demonstrate that a single/common framework can be applied to meet the needs of diverse business-oriented use cases, while using completely different PaaS and IaaS technologies. The remainder of the paper focusses on The "Telco PaaS" scenario. Section II provides a brief overview of the NGPaaS approach to building platforms and services. Section III

explains the VNFaaS use case in more detail, including how it is represented in NGPaaS, while Section IV outlines the testbed and initial results from the prototype. Finally Section V presents conclusions and future work.

II. NGPAAS FUNDAMENTAL PRINCIPLES

A fundamental building block of NGPaaS is the Re-usable Functional Block (RFB)[9]. In the ETSI model[10], RFBs cannot be decomposed into other VNFs since the composition stops at the level of VNF Component. In NGPaaS, the scope of RFBs is more arbitrary and generalized in that it covers *any deployable functional block* required to make up the service. This could cover images, install/configuration scripts etc, hence is not restricted to VM-based VNFs. In NGPaaS, we also allow for polymorphism, meaning in practical terms that an RFB can encapsulate something re-usable and modular at different levels of the architecture[9]. An obvious example here is that PaaS(es) like Open-CORD[7] or Kubernetes[8] can be subject to the same build/ship/run deployment cycle, as the services and components that run on top, so can also be considered RFBs.

A key component used by NGPaaS – nominally located in the Orchestration part of the BSS/OSS layer of Fig. 1 - is the RFB Decomposition and Composition Language 3D (RDCL3D) tool. RDCL3D is a web-based framework that allows for the composition of RFB graphs, which are then deployed on the infrastructure. Although originally developed in the 5G-PPP Superfluidity project[11], it has been further enhanced to meet the requirements of NGPaaS. As well as composing the RFB graphs, the RDCL3D tool is also responsible for "shipping" them to a RDCL agent which deploys the services/platforms composed by the graph. This

happens by executing per-RFB workflows, using dedicated Ansible roles. In simple terms, during “normal operations”, there is a sequence of process-oriented interactions within the NGPaaS architecture to deliver the required services on the required PaaS(es), as shown in steps 1-4 of Fig. 2:

- **Infrastructure Registration:** This provisioning process accommodates a diverse range of available heterogeneous infrastructure: Edge Cloud, Central Cloud, Bare Metal, Acceleration e.g. FPGA.
- **PaaS Orchestration:** Prior to a service request from the VSP, this step refers to the deployment of PaaS software (as selected by the operator) on the appropriate IaaS. The resulting PaaS capabilities should match the VSP’s use case requirements.
- **Service Orchestration:** In NGPaaS, a service is typically provided by deploying VNFs composed of RFBs whose target execution environment is included in the RFB’s metadata (e.g. specifying whether VMs, Containers, etc). A pre-deployed PaaS supporting the runtime aspects of the RFB will facilitate the actual service deployment.

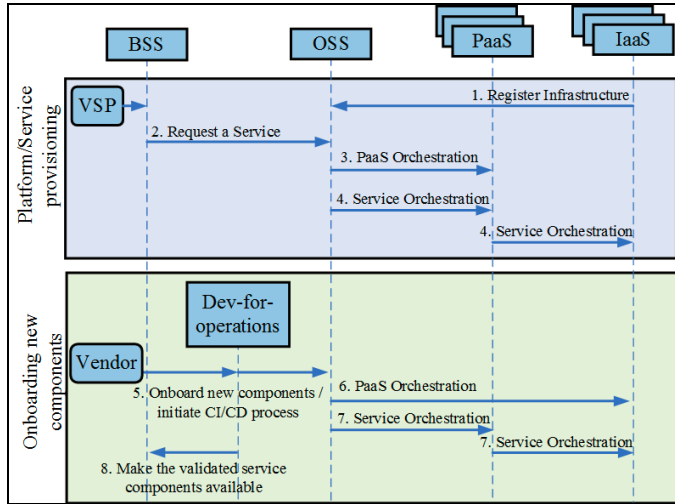


Fig. 2. NGPaaS Fundamental Process Interactions.

In parallel with the “usual operational” processes to build-to-order PaaS(es) and deploy associated service workloads, the Dev-for-Operations processes (steps 5-8 of Fig. 2) provide the DevOps-related tooling framework for vendors of PaaS and service components to continuously deploy and integrate new or updated components, made available for consumption as part of “usual operations”; Ref [3] provides further details.

III. VNF-AS-A-SERVICE AND NGPAAS METHODOLOGY

We now look in more detail at the VNFaaS use case, and consider how the NGPaaS framework outlined in the previous sections can enable various enrichments over a simple “baseline” service offering. Fig. 3 shows how a VNFaaS provider (e.g. a Tier 1 Telco) hosts VNFs on behalf of a VSP. These could be Routers or Firewalls (i.e. customer premise

equipment), which the VSP configures according to the provided service, for example business VPNs, Internet access, etc. The VNFaaS provider hosts the VNFs on behalf of the VSP who therefore does not need to ship hardware to the premises of his end customer[2].

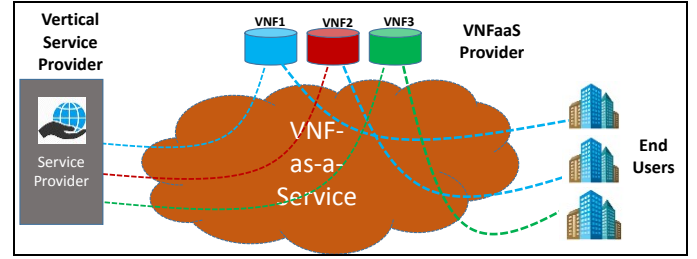


Fig. 3. VNF-as-a-Service Simplified Overview.

Going beyond a simple hosting service where available VNFs are pre-determined, in the NGPaaS framework, the Telco Provider offers a PaaS-oriented approach. Here, VSPs can onboard specific VNFs (e.g. vRouter, vFirewall) and also make selective use of Value-Added Service (VAS) capabilities which could be Telco-grade add-ons to the basic service (e.g. Monitoring, Alerting, Healing, Scaling). This leads to a very flexible “App Store” approach to defining service “Blueprints” based on re-usable and interchangeable components as shown in Table II.

TABLE II. VNFaaS Blueprint Examples

Service Components (VNFs, VASs)	VNFaaS Blueprint 1	VNFaaS Blueprint 2	VNFaaS Blueprint 3
Router (Vendor1)	✓		✓
Router (Vendor2)		✓	
Firewall (Vendor1)			✓
Firewall (Vendor2)	✓		
VAS- Monitoring	✓ (R,F)	✓	✓ (R)
VAS- Alerting	✓ (F)	✓	
VAS- Healing		✓	
VAS- Scaling		✓	

The interpretation of Table II is that VNFaaS blueprints can be defined by the appropriate selection of baseline VNFs and associated “Telco-grade” VAS extensions from simple Monitoring of the VNF, through to Alerting, Healing and Scaling functions. Blueprint 1 deploys a Router (Vendor 1’s VNF) plus Firewall (Vendor 2’s VNF) service chain, with monitoring selected for both VNFs, and alerting for the Firewall only. Blueprint 2 deploys a single Router with all of the available value-add services selected. Blueprint 3 meanwhile deploys a Router and Firewall service chain, with only the router selected for monitoring. The list of available Service Components can be extended (e.g. new VNF types/vendors introduced, or new VAS components), as well as the corresponding Blueprint permutations. The “Dev-for-Operations” aspect of NGPaaS is the means by which new assets can be added to the blueprint catalogue (as already illustrated at a high level in Fig. 2).

In the Telco PaaS implementation, services are deployed by means of blueprints being sent from the RDCL3D tool to

CORD. A visual representation of how VNFaaS Blueprint 3 has been modelled in RDCL3D as a “Service Graph” comprising RFBs is shown in Fig. 4. Router monitoring is achieved by connecting it to the same monitoring network as a monitor probe. The firewall is not required to be monitored in this particular example, so is not connected to this network.

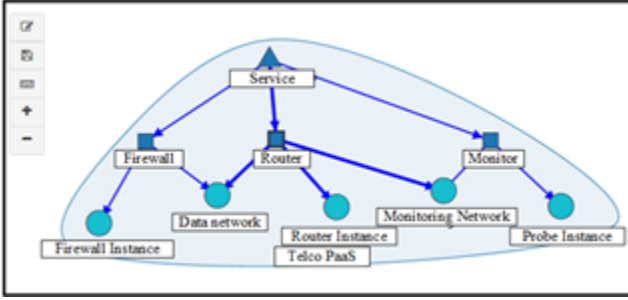


Fig. 4. VNFaaS Blueprint Modelled in RDCL3D as Service Graph.

While Fig. 4 shows the “graphical” composition of a particular service blueprint comprising RFBs, in Fig 5 we further expand on the methodology used to integrate all the key components into a working testbed.

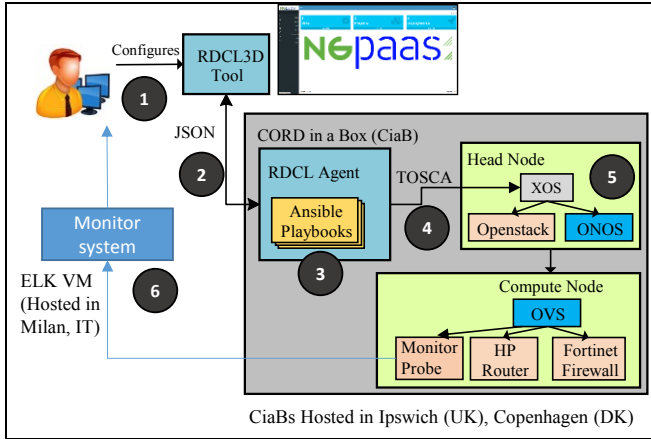


Fig. 5. Overview of Workflow Driving CORD via RDCL3D tool.

With reference to Fig. 5, having had blueprints defined by an administrator of the RDCL3D tool (1), the RDCL3D tool encodes these blueprints using JSON and sends them (via HTTP) to an RDCL agent (2), within the Cord-in-a-Box (CiaB) node. A sample JSON excerpt is shown below:

```
"metadata": {
  "flavor": "m1.cisco",
  "id": "150",
  "image": "CISCO2",
  "management": true,
  "networks": [
    "dataNetwork",
    "monitoringNetwork"
  ]
}
```

The RDCL agent translates the service blueprints into Ansible Playbooks and executes them locally in the CORD

node (3). The playbooks will trigger Ansible roles – developed within NGPaaS – and these generate CORD-compliant TOSCA recipes (4). Finally, the recipes instruct CORD to instantiate the desired service graph (5).

IV. TESTBED INTEGRATION AND RESULTS

The key hardware and software elements are a mix of open-source and vendor-proprietary components (Table III). While VNFs are vendor-proprietary, Open-CORD is open-source and ELK is an open-source stack used for monitoring and visualization. The ELK components¹ run inside a VM deployed in Unimib’s lab in Milan (part 6 of Fig 5), while the monitoring probe runs on the “Cord-in-a-Box” (CiaB) Telco PaaS deployments, external to, but alongside the VM-based VNFs. We have CiaB testbeds in BT labs in the UK and DTU labs in Denmark. Although the monitoring probe is deployed *external* to the target service, it can poll the target service for the required KPIs via suitable APIs. For “Service-level” monitoring - e.g. of VM-based VNFs - we make use of an API Anbased on the SNMP protocol (enabling collection of KPIs like CPU Usage, Memory, Network IO, etc). For “Platform-level” monitoring, we make use of a REST API to communicate with ONOS (the SDN controller part of Open-CORD) and collect Open VSwitch (OVS) statistics related to the compute node hosting the VNFs and monitor probe.

TABLE III. Testbed Components

Testbed Component	Additional Information
Hardware/laaS	“Private Cloud” Intel Xeon x86 servers.
Telco PaaS	Open-CORD Version 6.0, deployed on x86 bare metal as “Cord-in-a-Box” (CiaB).
Cisco CSR1000v	IOS-XE 16.03.05, License- Evaluation
HPE VSR1000	Comware v7.10, License- Evaluation
Fortinet Fortigate	VM-64-KVM v5.4.2License- Evaluation
Monitoring Framework	Elasticsearch-Logstash-Kibana (ELK) stack Monitoring of VNF Stats/KPIs: SNMPv3 Monitoring of Open-Cord PaaS: REST API.

Fig. 6 displays the real-time dashboard output following successful provision of the service chain according to selected VNFaaS Blueprint 3 where a Router and a Firewall are deployed but only the Router is monitored. The graph labelled “ONOS Bandwidth”, shows the platform-level statistics, while the “CPU Load for Monitored VNFs” shows real-time CPU utilization for a Router, which, as shown in the “VNF System Description” immediately below, is a Cisco Cloud Services Router CSR1000v (pre-configured for SNMPv3). Hence in this particular example, we show how NGPaaS allows for a single dashboard “pane of glass” to combine the presentation of “Platform-Level” (ONOS Bandwidth) and “Service-Level” (Router’s CPU load) graphical displays.

It should be stressed that the combination of specific metrics displayed in the dashboard can be further customized to meet the VSP’s requirements. For example, with the baseline capability in place, further extensions and enhancements are possible which go beyond passive

¹ Logstash is a log pipeline tool collecting data from a probe, Elasticsearch is a NoSQL database for search/analytics, while Kibana is the Visualisation/dashboard layer.

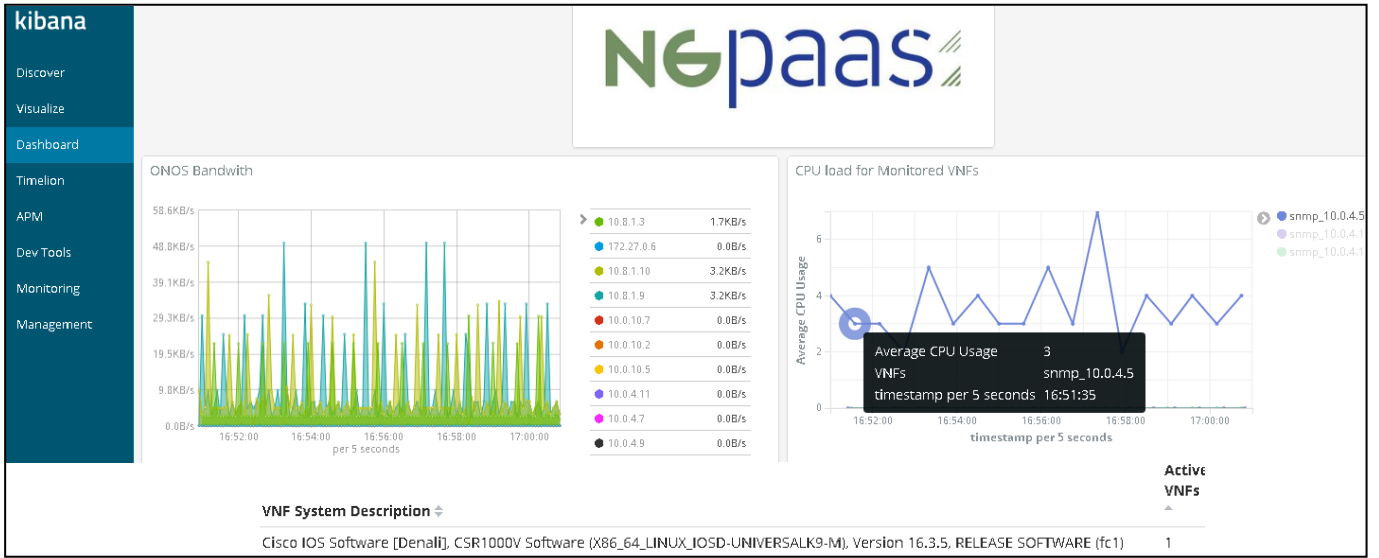


Fig. 6. Kibana Dashboard Visualisation Output for Testbed Prototype of VNF-as-a-Service

monitoring. Fig. 7 shows how an “Alerting” value-added service can respond to pre-set threshold rules on KPIs such as CPU utilization. The figure displays “anomaly markers”, generated in the event of a monitored router’s CPU load exceeding 50%, as well as a “Green to Amber” status change.

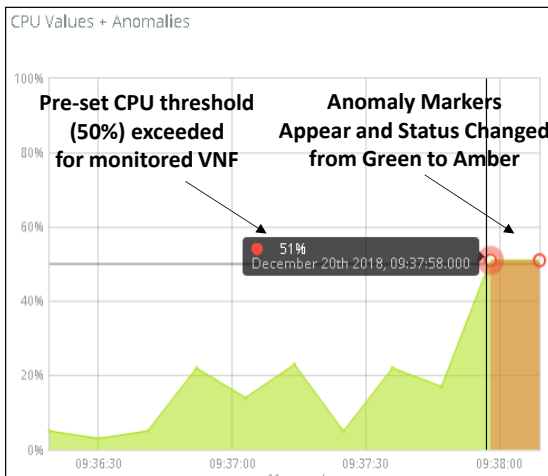


Fig. 7. Alerting-as-a-Service for a VNF’s CPU Load

V. CONCLUSION AND FUTURE WORK

This paper has explained how the novel Next Generation Platform-as-a-Service (NGPaaS) framework provides major benefits – in terms of flexibility and customization - for Telco Providers and Vertical Service Providers (VSPs). We have demonstrated many of the key features of NGPaaS as applicable to the VNF-as-a-Service use case, by building a testbed comprising both open source and vendor-proprietary components. Beyond the test results presented in this paper, and as was indicated in Table II, NGPaaS will permit even further enrichment of the VNFaaS baseline capabilities. Healing can be applied based on pre-defined rules for

anomalous behavior to trigger a remedial action, e.g. VNF reboot. Scaling can be applied based on pre-profiling[12] to determine whether deviations from “expected” behaviors could be a result of traffic conditions, and thus trigger an “up-scale” or “down-scale” of VNF resources (e.g. extra memory, vCPUs, etc). These additional enhancements will be the basis of future work, as well as addressing the end-to-end integration of the Telco PaaS scenario with IoT and 5G business scenarios to demonstrate how NGPaaS simultaneously caters for multiple and diverse use cases.

REFERENCES

- [1] “A Vision for the Next Generation Platform-as-a-Service”, Steven Van Rossem et al. IEEE 5G World Forum, Santa Clara USA, July 9th-11th 2018.
- [2] “A Network Operator’s Perspective on Next Generation Platform as a Service”, BT White Paper, Paul Veitch, June 2018.
- [3] “Next Generation Platform-as-a-Service (NGPaaS) From DevOps to Dev-for-Operations”, Atos White Paper, I. Labrador et al, October 2018.
- [4] ETSI General Specification NFV001, v1.1.1
- [5] Network Functions Virtualisation: Network Operator Perspectives on NFV priorities for 5G”, ETSI White Paper, February 2017.
- [6] Report on the Enhancements of the NFV architecture towards “Cloud-native” and “PaaS”, ETSI GR NFV-IFA 029 V0.4.0, November 2018.
- [7] “Central Office Re-Architected as a DataCenter”, <https://opencord.org/>.
- [8] “Kubernetes- Production-Grade Carrier Orchestration” <https://kubernetes.io/>
- [9] “Next Generation PaaS- A Cloud Native 5G Solution”, L. Rouillet, SDN NFV World Congress, The Hague, October 2018.
- [10] ETSI NFV ISG, “Network Functions Virtualisation (NFV); Network Service Templates Specification”, ETSI GSNFV-IFA 014 V2.1.1 (2016-10).
- [11] Superfluidity, “Deliverable D3.1: Final system architecture, programming interfaces & security framework specification”, Dec 2016.
- [12] “Introducing Development Features for Virtualized Network Services”, S. Van Rossem et al. IEEE Communications Magazine, Aug 2018.