5GTANGO: A Beyond-MANO Service Platform

C. Parada, J. Bonnet
{carlos-f-parada, jbonnet}@alticelabs.com
ALTICE LABS, Portugal

E. Fotopoulou, A. Zafeiropoulos
{efotopoulou, azafeiropoulos}@ubitech.eu
UBITECH, Greece

E. Kapassa, M. Touloupou, D. Kyriazis
{ekapassa, mtouloup, dimos}@unipi.gr
UNIVERSITY OF PIRAEUS, Greece

R. Vilalta, R. Muñoz, R. Casellas, R. Martínez
{ricard.vilalta, raul.munoz, ramon.casellas,
ricardo.martinez}@cttc.es
CTTC, Spain

G. Xilouris
{xilouris}@iit.demokritos.gr
NCSRD, Greece

***Abstract* — Unlike previous generations, 5G will be more than just a mobile network. 5G will have broader coverage, including humans, but also cars, robots, and things in general; and will target verticals like eHealth, Automotive, or Industry 4.0, just name a few. To deal with this, 5G will need to be faster, more efficient, reliable, flexible, agile, and, at the same time, cost less. For this to be possible, 5G has to engage with the best-of-breed of the emerging technologies, where NFV is definitely in the top list. ETSI NFV is today in an advanced stage of standardization. In particular, many MANO platforms are today available, with different levels of development, varying on the number of features and maturity levels. In this context, this paper describes the 5GTANGO Service Platform, an open source MANO framework currently under development in the scope of the 5GTANGO H2020 project, and whose main developments started in a previous H2020 project named SONATA. In particular, some features that go beyond the state-of-the-art are approached, either considering standardization or implementations available. Those features are Policy, SLA and Slicing.**

Keywords — 5GTANGO; NFV; MANO; Policy; SLA; Slicing.

I. Introduction

Founded in November 2012, ETSI NFV [1] standardization group has significantly progressed in defining a comprehensive framework for the virtualization of network functions (VNFs) and network services (NSs). ETSI NFV got a lot of traction from the beginning and quickly become the reference in the standardization developments organizations (SDO) arena.

At the same time ETSI NFV progressed in standardization work, proof-of-concept (PoC) activities have shown the impact fostered in the telecom market (more than 40 today). Recently, ETSI NFV has promoted PLUGTEST events, with the purpose of allowing vendors and open source NFV providers to validate their interpretation of the NFV standards, as well as to assess their interoperability levels. While standardization activities are progressing in a good pace, the implementation landscape is highly vibrant, in particular, in the Management and Orchestration (MANO) dimension. Here, for the first time, ETSI has created a working group to host the development of an open source tool called OSM (Open Source MANO).

Other open source implementations have also gained some momentum. Recently, ONAP (Open Network Automation Platform) got some highlight. This Linux Foundation managed project, has been created as the result of the *AT&T ECOMP* and *Open Orchestrator Project* (OPEN-O) merge. *Tacker*, an *Openstack* project, is another relevant MANO tool, as well as *Cloudify*, a very mature tool, although not fully ETSI-aligned. *OpenBaton* is another popular implementation from *Fraunhofer FOKUS*. Beyond those, there are many others not so popular, especially coming from research projects. All the projects above have strong and weak points, since the focus is always different. It is a project or leading community decision to go further on features they consider important, as a way to differentiate from the others. 5GTANGO is not an exception and tries to do the same. This paper describes the enhanced features that make this MANO better than other competitors.

In this paper, section I introduces the reader to the subject, while section II describes the basics of the 5GTANGO Service Platform (SP). Remaining sections go through the enhanced SP features. Section III approaches the Policy related features, while section IV refers to the SLA features and section V to the Slicing capabilities. Finally, section VI concludes.

II. Service Platform Overview

The Service Platform (SP), in the 5GTANGO architecture (see Fig. 1 [2]), provides the service and function orchestration features, plus all the needed complementary and supporting features, like slice management, policy and SLA management, user access management and infrastructure abstraction. The complementary features enhance the existing solution, as well as better support the verticals' requirements.

The SP supports two distinct forms of deployment: as a test platform for the V&V (Validation and Verification) platform and as a production SP. These two forms of deployment imply a design that supports two very different sets of user roles,

978-1-5386-1478-5/18/$31.00 ©2018 IEEE

which will be addressed in detail next. As shown, the SP interacts with OSSs/BSSs of the platform owner, as well as the SDK (Software Development Kit) and the V&V platform. The entry point of the platform is the Gatekeeper, controlling who tries to access the platform and the privileges. This is crucial in the open 5G ecosystems, where entry barriers are disruptively lowered, without decreasing security. Assets submitted to the platform by authenticated and authorized users (a user Portal, not represented, is available for registration) are stored in a Catalogue for future use. When there's a service instantiation request (or any other kind of request that affects the service instance life-cycle), the Gatekeeper forwards to the MANO Framework all the necessary data (fetched from the Catalogue), which instruct the Infrastructure Abstraction for the required resource allocation at the Virtual Infrastructure Management (VIM) and WAN Infrastructure Management (WIM) levels. A successful request will store those records in the Repository.

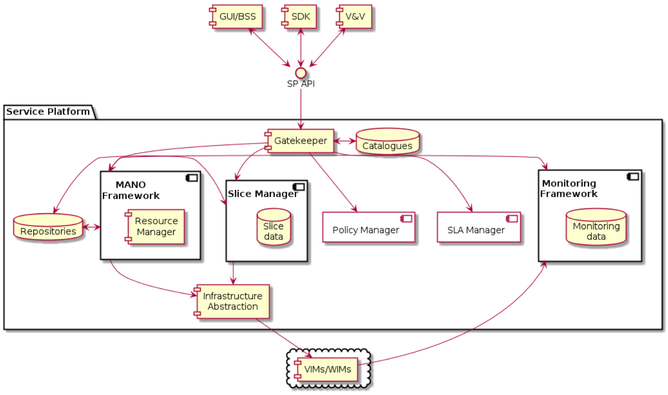

Fig. 1. 5GTANGO Service Platform high-level architecture [2].

The Gatekeeper validates and forwards requests for managing policies (see Section III) to the Policy Manager and for managing Service Level Agreements (SLAs, Section IV, below) to the SLA Manager. These two modules interact to provide the service platform owner and the Network Service's (NS) developer with added value about contracts with customers and NS monitoring parameters that can be also trimmed more adequately after some instantiations.

A feature that distinguishes 5GTANGO SP MANO from other platforms is the Function and Service Specific Manager plug-ins (F/SSM). This feature provides a plug-in mechanism that allows an authorized developer to create specific managers to his/her services or functions, thus changing the platform's default behavior. Examples of such plug-ins are placement, scaling or monitoring, which can be very specific to a service or function. The default (and thus necessarily generic to all services and functions that get instantiated) behavior of the SP will most probably be under-optimal for the kind of disruptive services expected to be developed and deployed over 5G. This bottom-up approach to services and functions management is, in our understanding, the correct way to approach the problem, but it comes with a cost we're lowering with the top-down approach of Policy Management: for example, global optimization or conflict resolution in the bottom-up/fully distributed approach is generically impossible to solve. We'll use both kinds of approaches to provide a flexible and yet efficient mechanism for service and function management.

Another feature to be highlighted is the ability for the Monitoring Framework to provide the developer of the service or function with streams of monitoring data of relative short duration or alerts, which allow to better trim the service or function he/she owns. This effectively closes the loop, from service and function on-boarding to performance analysis, allowing for the quick iteration of updated services and functions versions, a flexibility that is key in 5G ecosystems.

All these features, as well as many others, which cannot be all listed here due to space restrictions, are made open-source for the community to take advantage of [3].

III. Policy Manager

Policies in 5GTANGO are considering deployment and operational aspects of network services over programmable infrastructure offered mainly over network slices. Deployment policies regard the production of an optimal deployment plan, taking into account the set of constraints and objectives, as they are defined on behalf of network services developers and network services providers. Operational or runtime policies regard the runtime adaptation of network service mechanisms, in order to optimally support the overall performance achieved. Thus, it could be argued that the main objective of the introduction of policies management characteristics within the 5GTANGO SP, is the injection of intelligence in the various orchestration mechanisms. Such an injection is going to follow an evolutionary approach, since consumption of suggested actions by a Policies Manager can be realized for efficient deployment and operation of a network service, as well as dynamic management of a network slice.

Following, a short description of both policies aspects - deployment and runtime - is provided. In both cases, policies are realized through a Policies Editor that supports policies definition, refinement and mapping for a network service.

A. *Runtime Policies Management*

Several approaches are merging with regards to runtime policies management frameworks, considering formulation languages and design and implementation of runtime policies enforcement mechanisms. Policies formulation languages taken into account include a set of existing frameworks [1] and ongoing initiatives, such as SUPA specification [4], while runtime policies enforcement mechanisms include Drools [5].

In 5GTANGO, a policies formulation language along with a rule-based runtime policies management component is under development. The policies formulation language aims at providing a unified way for the description of runtime policies, targeted to network services and leading to the production of well-interpretable policy descriptors per network service. Based on this, for each NS, a set of rules per policy can be defined. Each rule consists of the expressions part, denoting a set of conditions to be met and the actions part denoting actions upon the fulfillment of the conditions. The conditions may regard resources consumption aspects, software-specific aspects, status of a VNF or a NS, while the set of actions may regard resources allocation/disposal aspects, live migration and mobility aspects, horizontal scaling aspects and network functions management or activation aspects.

A network service may be associated with a set of runtime policies, however only one can be active during its deployment and execution time. A runtime policy can be defined by the software developer of the NS, having detailed knowledge of the business logic of the developed software, or the network operator/service platform manager having knowledge of the available resources and the set of active SLAs. Knowledge of SLAs can lead to description of set of rules guaranteeing that such SLAs can be respected in the maximum possible way.

Policies enforcement is realized through a rule-based management system [6]. Based on the set of activated rules associated with the deployed network services and the real time collection of data, inference is realized leading to policies enforcement. Such enforcement takes place over the deployed network services, while the inference follows a continuous match-resolve-act approach. In the match phase, the set of defined conditions are examined, leading to the set of rules that have to be activated, while in the resolve phase, conflict resolution over the set of rules to be activated is taking place. The act phase concerns the triggering of the suggested actions aiming at the guidance of various orchestration components of the 5GTANGO service platform. For each rule, an inertial period is introduced denoting the time period that the same rule should not be re-triggered.

In more detail, the Policy Manager consists of (i) the working memory; facts based on the provided data, (ii) the production memory; set of defined rules, and (iii) an inference engine that supports reasoning and conflict resolution over the provided set of facts and rules as well as triggering of the appropriate actions. Data is fed to the working memory through the monitoring mechanisms that are responsible to collect data based on a set of active monitoring probes and to support a set of data management and processing operations. The production memory is also fed by policies associated with the deployed network services, as provided through the policies editor. The Fig. 2 depicts the Runtime Policies Management Framework.

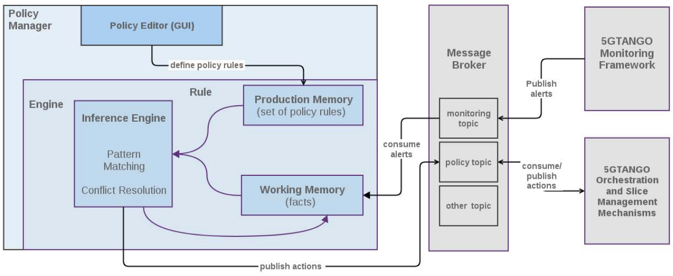

Fig. 2. Runtime Policies Management Framework.

During inference, triggering of actions is realized, taking into account the provided data. Such actions are tackled as suggestions that can be considered by various orchestration mechanisms for enhancing their effectiveness. These suggestions are made available in the Service Platform message broker, and thus can be dynamically consumed. Based on the adopted and successfully realized actions, policies validation is also taking place, leading to insights to network operators regarding the feasibility of the defined rules.

B. Deployment Policies Management

Deployment of network services over the created network slice is going to be based on the production of an optimal deployment plan, taking into account high level objectives of a services provider and set of constraints defined by software developers and service providers. High level objectives may regard - among others - performance, cost, energy efficiency, security and privacy aspects. Constraints may also regard - among others - minimum resources allocation, locality, time criticality, security aspects. Hard and soft constraints may be defined, the latter ones being desirable, however, not obligatory to be fulfilled, relaxing somehow the overall optimization problem to be solved.

The objective of the deployment policies management framework within 5GTANGO is to provide a flexible approach of definition of the main objectives and constraints by the various stakeholders and the implementation of a set of constraint satisfaction solvers providing optimal deployment plans. The selection of the more suitable constraint satisfaction problem can be realized taking into account the peculiarities and the size of the problem to be solved. Towards this direction, combination of constraint satisfaction solvers with heuristics/meta-heuristics solutions is envisaged to be realized within the framework of the project.

IV. SLA Manager

To enable the provision of a NS with the corresponding quality requirements tailored to it, SLA management should provide mechanisms to capture its network-related parameters, reflecting the corresponding quality levels [6]. These may be included in policy rules as the Service Provider is able to modify them accordingly, through updating existing ones and enrich SLAs with additional parameters. High-level parameters (performance, availability, security) specified for the NS in an SLA, are linked to low-level requirements [7] – encapsulated in the respective policies.

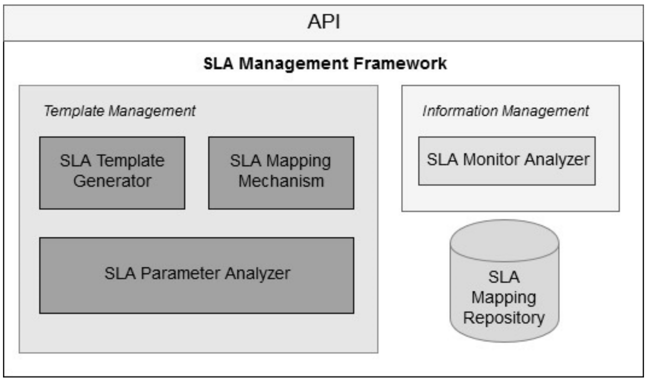

Fig. 3. SLA Management Framework.

The SLA Management Framework is depicted in Fig. 3 and consider two phases: Pre-deployment (Template Management) and Post-deployment (Information Management). It is worth noting that an SLA captures the “profile” of the user, as well as includes parameters tailored to the corresponding needs.

A. Template Management: From Generalized to Tailored

The SLA Template Management component is responsible for receiving the appropriate dataset from different stakeholders (such as the service provider and the end-user), analyze them, map them and formulate a tailored SLA Template. Furthermore, generation of a signed SLA in an automated and efficient way is also an action that will take place during the Template Management definition phase.

A vital part of the SLA Management Framework is the SLA Template Generator mechanism, which firstly generates the SLA Templates for the Service Provider, and secondly creates the SLA Agreement (signed SLA) by updating accordingly the template. The SLA Template Generator should be able to access both the NS-Descriptors and the associated set of policies for a specific NS, obtained by the Policy Manager, in order to create the templates in an automated way. The purpose of that is to be able to analyze and correlate the aforementioned information, in a way that can formulate a generalized SLA Template, applicable also to other services and verticals. At the same time, each generated SLA Template can be adjusted to each Service Provider's needs and requirements. Therefore, a Template can be changed from the Service Provider by adding more explicit parameters to it and generate a tailored SLA Template with personalized parameters. An SLA Template specification should contain initial Service Level Objectives (SLOs) which include a set of service level parameters, like availability, and each one has a target level to achieve. The key components of an SLA Template are presented in Fig. 4.

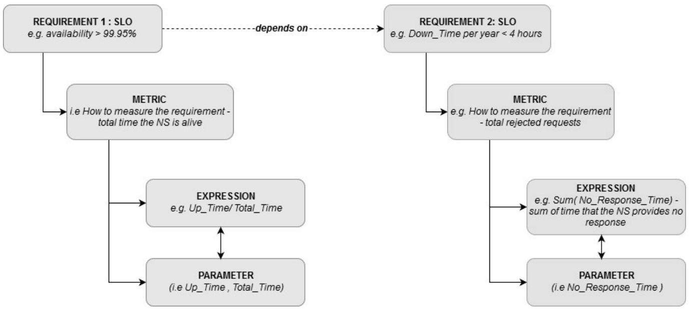
REQUIREMENT 1 : SLO
e.g. availability > 99.95%
depends on
REQUIREMENT 2: SLO
e.g. Down_Time per year < 4 hours
METRIC
i.e How to measure the requirement - total time the NS is alive
METRIC
e.g. How to measure the requirement - total rejected requests
EXPRESSION
e.g. Up_Time/ Total_Time
EXPRESSION
e.g. Sum(No_Response_Time) - sum of time that the NS provides no response
PARAMETER
(i.e Up_Time , Total_Time)
PARAMETER
(i.e No_Response_Time)

Fig. 4. SLA Template Key Components.

The SLA Mapping Mechanism (MM) is responsible for the mapping between the high-level requirements described by the end-user and the low-level requirements described by the service provider. The MM obtains a set of policies from the Policy Manager, a set of low-level requirements from the Service Provider and a set of high-level requirements from the end-user, based on the services being selected from the 5GTANGO Catalogue. The datasets are processed through an Artificial Neural Network (ANN) to compose, aggregate, and convert the low-level metrics to high-level SLA parameters [8]. Since ANN represents a black box approach, do not need any knowledge regarding the internal structure of the services. They only need inputs and outputs of the model, which are available as they are the prior input dataset of NSDs and policy rules [9].

The SLA Parameter Analyzer (SPA), after searching into the mapping repository, decides whether the process of the MM should take place or not, based on already stored correlations.

B. Information Management

The SLA Information Management (SIM) takes place after the successful instantiation of the NS, and is responsible for optimizing the mapping results.

The SLA Monitor Analyzer (SMA) component, compares the QoS parameters from the Mapping Repository, with the computed monitoring data, and then calculates the delta between them. If delta > x%, the MM will obtain the monitoring feedback as an additional dataset and repeat the mapping process in order to optimize it.

The SLA Mapping Repository (SMR) is responsible for keeping all the produced mapping results from the MM. More specifically, it will store the correlations between the high-level and the low-level requirements.

V. SLICE MANAGER

In 5G, network sharing goes beyond the traditional infrastructure sharing models used in previous generations, and is termed as Network Slicing, which consists on running multiple logical networks as virtually independent business operations on a common physical infrastructure in an efficient and economical way, so that specific different QoS network requirements can be tackled for different verticals use case needs (e.g. URLLC, eMBB, mMTC). Network Slicing are under standardization in different SDOs, and a lot of challenges are expected to be fixed before slicing becomes a reality.

One of the main challenges is the mapping of Network Slices instances to the physical and logical resources that will be allocated to them. While 3GPP proposes to focus only on the network functions and provide network slice blueprints [10]. Hence, resource allocation is not within its standardization activities. However, IETF and other SDOs are proposing to relate the requested network functions with specific and even deterministic network resources [11].

In this environment were slicing concept is still taking forms depending on the point of view of each organization, 5GTANGO will support flexible mechanism for orchestrating and lifecycle management of end-to-end Network Slices also taking into account verticals application requirements and SLA polices (as define previously). The objective is to align our proposed network slicing architecture with 3GPP focusing on compliance and support all the standardized features. However, it will also provide complex resource allocation schemas, fitted into dedicated network slice blueprints suitable for verticals industries. 5GTANGO network slicing will provide extended and novel isolation and customizable features, E2E telemetry and monitoring, and E2E orchestration. Novel slice operational features such as slice elasticity, and scalable slice resource scheduling will also be introduced.

5GTANGO will extend, implement and validate the architecture proposed in [12], where ETSI and 3GPP SDOs combine their specifications in order to offer Network Slicing. It proposes a Slice Manager, with internal architecture shown in Fig. 5. The Network Slice Manager is a SP functional block that interacts with OSS and is responsible for interacting with the SP MANO framework to control slicing. The MANO exposed REST APIs are consumed by the Network Slice Manager (reference point Os-Ma-nfvo).

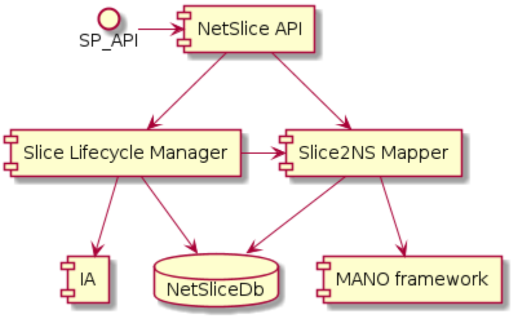

Fig. 5. Slice Manager Architecture.

Within the Network Slice Manager, one can identify two main components: a) the Slice Lifecycle Manager, which is responsible for assigning services and applications to network slices and for managing the lifecycle of these slices; and b) the Slice2NS Mapper, which is responsible for mapping network slices to NFV Network Services (NS).

The Slice Lifecycle Manager is responsible for the entire lifecycle management of the created network slice, from instantiation to termination, including scaling, healing or any kind of runtime modification, like migration, reconfiguration, etc. The Slice Lifecycle Manager instantiates a network slice by using a Network Slice Template (NST) previously on-boarded, and assigns it to the customer. The network slice instantiation triggers the instantiation of underlying NSs with the appropriate flavor and performance.

The Slice2NS Mapper function maintains an association between the on-boarded NST and underlying NSDs. The NST includes a list NSDs that need to be instantiated to comprise the instantiation of the network slice, considering the applicable deployment flavor and performance requirements. It also deals with the combination of NSs. This function is also responsible to maintain the mapping between the slice and comprising NSs for any LCM operation that needs to be performed.

Once resources have been properly virtualized and allocated to a network slice instance for a particular tenant, slice isolation between tenants is of extreme importance. Slice isolation implies that each tenant shall perceive its slice as the only one in the system, and shall not be able to modify any behavior of a slice that is not owned by itself. It is essential to consider slice isolation in terms of resources, performance and operations. Several mechanisms for isolation have been proposed [13], which focus (among others) on standardized methods for designing inter-slice access and management, accessing common network functions shared between isolated slices, creation of new slices without violating current level of isolation between existing slices (especially in the 5G RAN), and finally, accounting and non-repudiation for slices' users and operators. It is worth to mention that isolation cannot always resort on the use of physically separate resources; establishing hard quotas on physical resources only changes the resource granularity, therefore, interesting solutions for slice isolation must not generally prevent dynamic sharing.

VI. Conclusions

In this paper, the 5GTANGO Service Platform tool has been presented, highlighting the features that we consider being beyond-MANO, which are the strongest points when compared to other competitors. The features approached in particular were the SLA, Policy and Slicing. The SLA feature allows the establishment of contracts between NFV customers and providers, in order to ensure that the expected service levels are actually delivered. The Policy feature allows the creation of an internal policy-driven engine, which is able to take decisions like scaling, healing, etc, based on monitoring or other kind of data. Finally, the Slicing feature allows the creation of virtual networks (slices), allowing his entire lifecycle management. We believe that this set of enhanced features, together with a strong ETSI NFV compliance with standards, makes the 5GTANGO Service Platform an extremely competitive solution within the overcrowded open source MANO arena.

Acknowledgments

This research has been co-funded by the projects: H2020 5GPPP 5GTANGO and Spanish MINECO DESTELLO.

References

[1] ETSI NFV, http://www.etsi.org/technologies-clusters/technologies/nfv.

[2] 5GTANGO, D2.2: Architecture Design, November 2017.

[3] SONATA GitHub, SONATA Project, https://github.com/sonata-project.

[4] SUPA Policy-based Management Framework, https://datatracker.ietf.org/doc/draft-ietf-supa-policy-based-management-framework/.

[5] Drools Rules Based Management System, https://www.drools.org/.

[6] P. Gouvas, E. Fotopoulou, A. Zafeiropoulos, C. Vassilakis, "A Context Model and Policies Management Framework for Reconfigurable-bydesign Distributed Applications", Procedia Computer Science 2016.

[7] E. Kapassa, M. Touloupos, A. Mavrogiorgou, D. Kyriazis, "5G & SLAs: Automated proposition and management of agreements towards QoS enforcement", in press.

[8] J. Bendriss, IGB. Yahia, P. Chemouil, D. Zeghlache, "AI for SLA Management in Programmable Networks", Design of Reliable Communication Networks, 2017.

[9] G. Kousiouris, D. Kyriazis, S. Gogouvitis, A. Menychtas, K. Konstanteli, T. Varvarigou, "Translation of Applicationlevel Terms to Resourcelevel attributes across the Cloud Stack Layers" , IEEE Symposium on Computers and Communications (ISCC) 2011.

[10] 3GPP TR 28.801 Study on management & orchestration of network slicing for next generation network, 2017

[11] ETSI, "NFV-eve 012 report on network slicing support with ETSI NFV architecture framework," 2017.

[12] Chatras at all, "NFV enabling network slicing for 5G." Innovations in Clouds, Internet and Networks (ICIN), 2017 20th Conf on. IEEE, 2017.

[13] Kotulski, Zbigniew, et al. "On end-to-end approach for slice isolation in 5G networks. Fundamental challenges." Computer Science and Information Systems (FedCSIS), 2017 Federated Conference on. IEEE.