ASTES

Advances in Science, Technology and Engineering Systems Journal Vol. 4, No. 1, 329-342 (2019)

www.astesj.com

Special Issue on Recent Advances in Engineering Systems

ASTESJ
ISSN: 2415-6698

Managing and Optimizing Quality of Service in 5G Environments Across the Complete SLA Lifecycle

Evgenia Kapassa*[,1], Marios Touloupou[1], Panagiotis Stavrianos[1], Georgios Xylouris[2], Dimosthenis Kyriazis[1]

[1]*University of Piraeus, Department of Digital Systems, Piraeus, Greece*

[2]*Institute of Informatics and Telecommunications, NCSR "Demokritos", Athens, Greece*

ARTICLE INFO

Article history:
Received: 20 December, 2018
Accepted: 15 February, 2019
Online : 28 February, 2019

Keywords:
Quality of service provisioning
5G networks
Service level agreements
Cost-efficient monitoring
Network service recommendation

ABSTRACT

The 5G is the fifth generation of mobile broadband, cellular technologies, and networks that promises a major change in mobility by evolving connected business realities. In such an emerging environment, reliable Service Level Agreements (SLA) and anticipation of breaches of Service Level Objectives (SLO) become compulsory. Thus, guaranteeing the required service quality, while also ensuring efficient recourse allocation becomes a challenge. In addition, 5G networks are expected to provide diverse Quality of Service (QoS) guarantees for a wide range of services, applications and users with a variety of requirements. However, there is an increased difficulty in translating user-friendly business terms into resource-specific monitoring attributes that can be used to manage resources in the 5G core network. To address these gaps, an SLA management framework, enabling QoS provisioning is introduced. The aforementioned framework will be supported by an adaptive monitoring algorithm, which removes the static time interval used in the monitoring system, in order to provide highly accurate information in real time, without the produce of unnecessary traffic to the network. The proposed architecture also incorporates a recommendation mechanism to determine the significance of various QoS parameters in order to ensure that relevant QoS metrics are included in the SLAs, using enriched metadata information from a Network Function Virtualization (NFV) Catalogue.

1. Introduction

The explosive growth in mobile data, forces network operators to transform their networks [1]. The emerging 5G in combination with Software Defined Networks (SDN) aims to comply with different quality of service (QoS) requirements in different application scenarios [2]. Nevertheless, promoting delay-limited QoS over emerging 5G wireless networks with limited resources for bandwidth-intensive and time-sensitive scenarios, presents many new challenges not arised in 4G networks [3,4]. Therefore, it becomes crucial to adopt high performance Virtual Network Functions (VNF) and Network Services (NS) that require a plethora of system resources [5,6]. However, this kind of provisioning in a virtualized environment is a challenging task. Various virtual services in fields like Augmented Reality (AR), Virtual Reality (VR), or autonomous guided vehicles (AGVs), require corresponding Service Level Agreements (SLAs) that capture different levels of guarantees as QoS requirements, such as performance and availability. As stated by many Telco's and cloud service providers, they are tempted by the promise of QoS such as high speed/performance, high reliability, low latency, increased capacity, availability and connectivity, as well as dynamic bandwidth allocation from 5G RANs to core networks [7]. As a result, the responsibility to provision the necessary infrastructure recourses and QoS assurances, lies with the 5G network operators and network service providers, in order to ensure this kind of demands. Consequently, there are challenges to address this topic which include a pro-active SLA management framework, an efficient and high adaptive monitoring system on top of the virtualized 5G infrastructure, while also support the provisioning of appropriate SLAs, in terms of business aspects and recourses allocation. One of the most significant challenges, is the role of an SLA Management Framework on top of the 5G Core network [8]. It should be pointed, that managing a virtualized network attempts to preserve acceptable quality, and at the same time push the customers to negotiate with the service providers precise QoS thresholds. Considering that the customer-related QoS requirements are indicated in an SLA, the thing that is vital for the service providers, refers to the assessment of the necessary resources, for each NS that is going to be deployed.

*Evgenia Kapassa, University of Piraeus, +30 2104142746 , ekapassa@unipi.gr

https://dx.doi.org/10.25046/aj040132

To this end, monitoring the infrastructure but also the running services, should be considered as the mediator for supporting QoS provisioning. In 5G networks, where QoS assurance through SLA enforcement is a crucial process, data monitoring is required to evaluate the health of the network [9]. Additionally, service provisioning and assurance tools in NFV should monitor a large number of end points [10]. Keeping that in mind, a frequency analyzer tool could be the key, where real time data and low running costs are essential in this kind of networks. Thus, monitoring and evaluation of monitoring data are considered as important aspects to track the overall progress and reliability of a running service. However, decisions on adaptive and pro-active tracking should be made without the possible absence of information that could negatively impact strategic decisions [11]. Such solutions should also be able to adapt the virtual service during runtime, while at the same time maintain all health and performance historical data as the service evolves [13].

At this point, questions such as how the QoS parameters would be described, or where the SLAs would be stored, and how the service providers could manage all this information arise. Furthermore, the efficient description of the QoS parameters comprises an issue of paramount importance, along with the storage and the management of the SLA descriptors from the service providers. Primarily, for fulfilling this demand in 5G infrastructures, the concept of the NFV Catalogues is introduced. Here comes the vision of the NFV Catalogues, going beyond a plain data store to the promotion of the service offerings and the facilitation of the commercial activity and fluent interaction among the different business stakeholders. The above-mentioned concept not only enables the main storage of the 5G infrastructure, but also, promotes the management and the exploitation of this information. As a result, recommendations mechanisms, or even predicting tools can be developed, providing the most suitable SLAs for the selected by the customers network service [12].

Considering the aforementioned set of challenges, in this paper, we are trying to portray an “SLA-Oriented Framework” which targets the profitable provisioning of QoS guarantees, in 5G environments. The proposed approach is based on mapping the high-level customers’ requirements to low-level recourse attributes, as shortly described in [14,15]. The proposed Framework adopts an Artificial Neural Network (ANN) approach, making it easily adaptable to different software components like VNFs and NSs. It is widely known that ANN have many benefits, but the main reason for selecting them is their fault tolerance. This kind of advantage is very important in the SDN environment, where networks are scaled across multiple machines and multiple servers. Additionally, we also introduce an extra set of mechanisms for suggesting the most important QoS parameters, for endorsing the most reasonable and profitable SLAs for diverse services and customers. Furthermore, we also introduce an adaptable “Monitoring ‘Framework, in order to supervise the established SLAs. The latter is based on a scheduling algorithm that provides decision logic on probe level as initially described in [17]. On top of them, an enhanced metadata “NFV-Catalogue” is adopted for valuable management of the NSs and the corresponding SLAs.

The remaining of the paper is organized as follows. Section 2 presents the related work and motivation of this work. Section 3 introduces the overall “SLA-Oriented Framework”, while in Section 4, 5 and 6 we describe in detail the three major architectural components. Section 7 states the evaluation of the proposed approach in a real 5G testbed, through an end-to-end scenario. Finally, in Section 8, we close up with ideas for future experiments and current study capabilities

2. Background and Motivation

2.1. SLAs & QoS assurance

It is doubtless that Active and advanced work on SLA management for the cloud an 5G infrastructure has been carried out. Therefore, there are many researches for solutions that handle QoS parameters and monitor in an efficient way the guaranteed SLOs. A framework is discussed in [16] to utilize QoS into grid applications. In this paper a performance model is used to calculate the response time and the pricing model to determine the cost of conducting a job. It should be noted that a baseline for our work, is presented in [19], where the architecture of an SLA management system is described. Another intriguing work is described in [20], which demonstrates the importance of the implementation of ANNs in the service- oriented field. In this approach, the ANN's main objective is to set technical goals in terms of quality goals for the design attributes of web service systems. Moreover, the LoM2HiS framework is presented in [21]. In this case, the authors provide a paradigm that implements the reverse process of the one we propose, where low- level specifications are translated into high- level requirements that are used in cloud SLAs. More details concerning the mapping of requirements can be found in [22,23]. Moving forward, to the next generation of networks, the 5G Network Slice Broker [24] is an innovative network element that builds on the capacity broker function block, for advanced RAN sharing considered by 3GPP. It maps incoming SLA requirements to physical resources in connection with network slice requests, having as a result to get a “slice” of the relevant elements of the Radio Access Network (RAN) [25].

In addition, a lot of work has been done to provide guaranteed QoS for enhanced user experience. There was a migration from QoS management at the user equipment level, to QoS management at the network level during the evolution of the QoS management mechanism in 3GPP networks, a shift which maintained also in 5G networks [26]. The QoS level provided in 5G systems should meet the requirements of future Internet industry and go beyond what can today be accomplished with any wireless communication technology. Essential QoS requirements in currently studied 5G systems include: a) maximum acceptable end-to-end latency (delay) less than 5ms and b) reliability around 10-9% or 99.999% [27, 28].

It is worth mentioning, that delay- limited QoS requirements are relatively difficult to guarantee, due to highly differing wireless channels. Alternatively, the statistical delay- limited QoS supply theory has been initially proposed and demonstrated to be a useful method for characterizing and implementing the delay- limited QoS guarantee for wireless real- time traffic [29]. In addition, several works have proposed solutions in the framework of the QoS scheduling [30], including multi-QoS scheduling as investigated in [31,32]. Instead, authors in [33,34] contended that existing QoS mechanisms do not endorse the implementation of specific policies for a group of network users.

It should be pointed out, that a preliminary description of our work can be found in [14,15,18], where the authors presented an approach for mapping the high-level end-user requirements the low-level policy parameters, and at the same time proposed a mechanism for suggesting the most important QoS parameters to the Service/Infrastructure Provider, in order to achieve better QoS assurance. In the present paper, we are going beyond an isolated SLA management framework, by trying to support business guarantees in the overall lifecycle of NSs. To do so, we co-operate the proposed "SLA Framework" with an "NFV-Catalogue", providing recommendations based on benchmarking results, as well as with an advanced "Monitoring Framework" for efficient SLA violations detection.

2.2. Efficient Monitoring

The need for network monitoring is a key enabler for efficient network management. Several works address this domain. As stated in [35], OpenTM was presented, where the integrated features provided in the OpenFlow switches are used to directly and accurately measure the low overhead traffic matrix. OpenTM also utilizes the routing knowledge acquired from the OpenFlow controller to intelligently select the switches from which flow statistics can be obtained, decreasing the load on switching elements. Furthermore, the authors in [36] presented Flow Sense, a push-based approach to performance monitoring in flow-based networks, where they let the network inform regarding performance changes, rather than query for metrics information on demand. The key point is that control messages sent by switches to the controller contain information that allows performance estimation. Moreover, another well-presented monitoring framework for SDNs has been introduced in [37]. The authors propose a software-defined traffic measurement architecture that distinguishes the data plane from the control plane, namely OpenSketch. OpenSketch offers a simple three-stage pipeline (hash, filter and count) in the data plane that can be enforced with commodity switch components and support many measuring tasks. In the control plane, OpenSketch provides a measurement library that automatically configures the pipeline and allocates resources for different measurement tasks. What is more, in [38], authors have presented an extension of Prometheus.io, a monitoring framework implemented and integrated within SONATA project [39]. In short, the SONATA monitoring framework gathers and processes data from many sources, enabling the developer to activate measurements and thresholds to capture generic or service-specific behaviors. In addition, the developer can define rules based on metrics collected from one or more VNFs in one or more NFVIs to receive runtime notifications. Furthermore, authors in [40] proposed an approach for comprehensive and detailed monitoring of 5G mobile networks characterized by software using an IoT-based system. The corresponding monitoring framework is designed to collect any type of data either in text or in numerical form in a cloud database. Thus, by using this knowledge, SDN controllers make the decisions on network reconfiguration according to current conditions. A great work was published also in [41]. In this article, a Software Defined Monitoring (SDM) was proposed, while it highlights how SDMs can be used to solve the current limitations in legacy monitoring systems. The proposed approach is able to monitor both virtualized and physical network environments in an economical and efficient way. Initially, the authors' proposed SDM architecture was used only to monitor 5G backhaul network. Last but not least, an automatic monitoring management for 5G mobile networks was proposed in [42]. In particular, a 5G-oriented architecture was proposed to integrate SDN and NFV technologies to monitor and control the entire service life cycle taking into account network control plane information. This architecture automatically manages network resources to orchestrate network monitoring services, which are developed in their solution as VNF monitoring. VNF monitoring is described by the collection of information from different and diverse sources, such as network (i.e. physical or virtual) infrastructure data, network management services and user- to- network communication

2.3. NFV-Catalogues & Recommendations

The conception of the NFV Catalogues for the coherent storage of the exchanged entities of the 5G infrastructure originates from several applications into the new era of virtualization. The first attempt of delivering a distributed storage component with functional business and service layers for the VNFs/NS operators was from the [43]. The main view of this approach was to provide a digital marketplace that collects VNFs / NS to operate on commodity cloud infrastructures. On top of that, the continuous and real-time network information of the available VNFs/NS is exchanged between the several layers of the NFV Catalogue based on a set of RESTFul APIs for a functional service. Though, the first concrete approach was presented from in the framework of the T-NOVA project [44]. The authors presented the storage of the machine-readable descriptions of the VNFs/NSs and was covered solely from the NFV Catalogues as an integrated component. In parallel, the NFV Catalogues were fully aligned the functional components of the T-NOVA infrastructure, responsible for the charge of registering all business relationships and exposure of the related information for the billing component. Although, this approach presents a distributed storage approach with the several QoS/QoE metrics being disperse in the ecosystem, introducing latency in the diverse functionalities. The next milestone in the evolution of the NFV Catalogues was set one more time by the SONATA project. This approach was predominantly based on the efficient storage of the generalized package formats of the NFV landscape and their functional information. The followed approach was strictly correlated with the specification of the ETSI MANO for the diverse employed NFV Catalogues in the infrastructure [45]. What was beyond the specification comprised the introduction of the engagement of the information on the instantiated VNFs/NSs in accordance to the SONATA Service Platform policy updates.

In parallel with the vast development of the Web, the advent of the rapid growth of the available information was obvious to its users. Recommender Systems (RSs) pioneered the web with the aim of incorporating social information and at the same time delivering meaningful suggestions to the end user. While the research field of RSs has been skyrocketed in diverse domains, there is a gradual, yet slow, interest of the application of the RSs in the 5G ecosystems, through their pinpointing in network management applications [46-48]. Through the introduction of the virtualized era, telecom networks generate massive amounts of monitoring data consisting of observations on network faults, configuration, accounting, performance, and security. E- stream also included a predictive, automated network management recommendation with surprising results due to the ever- increasing complexity of the networks, correlated with particular business level constraints [49]. Through the exploitation of the profound streams of data and the efficient application of techniques in dimension reduction, E-stream was based on recommending actions with four different aligned factors, namely the context, audience, existing responses, and validation. The main factor of these recommendations was the applicable trading-policies for actuating the recommendation module and propelling the necessary actions to get the best price for each VNF be carried out. Yet, in the 5G telecom systems, there is a paramount absence of RSs in utilizing implicitly the plethora of the QoS/QoE metrics of the multiple diverse components, such as monitoring systems, policy and SLA modules, etc. Thus, RSs techniques and methods comprise a subset of tools that need to be examined thoroughly.

2.4. The challenges of 5G & SLAs

An SLA is a contract between a service provider and its end users that documents what services the service provider can provide and establishes the performance standards that the service provider is required to meet [50]. Since each component potentially impacts the overall behavior of the SDN, any high-level target specified for the service (e.g. performance, availability, security) potentially impacts all low- level components.

SLAs establish customer expectations regarding the service provider's performance and quality. In recent years, SLAs have set expectations for the performance of a service provider, set down penalties for missing targets and, in certain cases, rewards for exceeding those targets [21]. With SDN, network devices (routers and switches) can be managed using OpenFlow [51] and a separate set of application programming interfaces (APIs) can handle virtual network overlays. Once the network is virtualized, the SDN controller can set up network devices as fast as it can deploy new VMs. For instance, customers can take a VM image, deploy it to hardware, spin it, apply it via OpenStack Compute and set up the network around it in a short time via OpenStack Neutron (i.e. Orchestrated SDN form) [52]. In this scenario, SLAs can cover the time and cost of deploying new compute resources, as well as their related network resources [53].

Finally, the SLA lifecycle is an important part of the provision of services, in particular in SDN and 5 G networks. The SLAs lifecycle in the 5 G domain is managed by the accompanying 5 G service platforms and is slightly different from the traditional ones as described in [54]. SLA management is a dynamic process comprising four key stages: a) Architecture, b) Engagement, c) Operations and d) Termination as presented in Figure 1. The overall lifecycle is 5G-enabled, due to the fact that is fully aligned with 5G principles and is running in parallel with Network Service Lifecycle [55].

The first phase starts with the selection of a NS and the requirements definition by the developer. Typically, the Operator is the one responsible to examine those, take into consideration important business needs and implement SLA Templates, as initial offers to the NS customers. During the engagement phase, the selection of different NS results from business aspects, which are the basis for different QoS constraints, which can also be defined as requirements of the agreement.

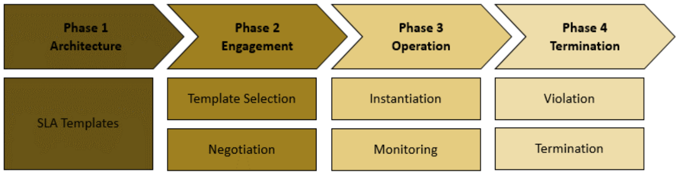


Figure 1 : SLAs Lifecycle in 5G.

The preference of an operator / network service provider depends on the desired network service (NS), its characteristics, budget constraints and so on. QoS expectations also drive end users to negotiate with their operators / service providers precise QoS levels. An SLA is created to describe the agreed QoS parameters after a successful negotiation process. After the successful NS selection and the corresponding SLA negotiation process, the operation phase takes place. This phase comprises the actual deployment of the NS, the population of the respective service with running data, the establishment of communication channels and additional operational activities. Moreover, the operation phase monitors the agreement with real- time data, for the purpose of avoiding or managing unexpected violations. Finally, termination phase deals with the end of the relationship between operator/service-provider and NS customer, including the end of the legal relationship. In general, the latter will continue for a few years after termination in accordance with mandatory laws and legislation. This last phase includes the evaluation of alternatives, settlement and termination commitments, export of data, customer care and diligence, and deletion of data. All the above should be considered either the Network Service was terminated, or the SLA was violated. More details for the aforementioned processes and how they are managed in the proposed architecture will be discussed in Section 3 of this article.

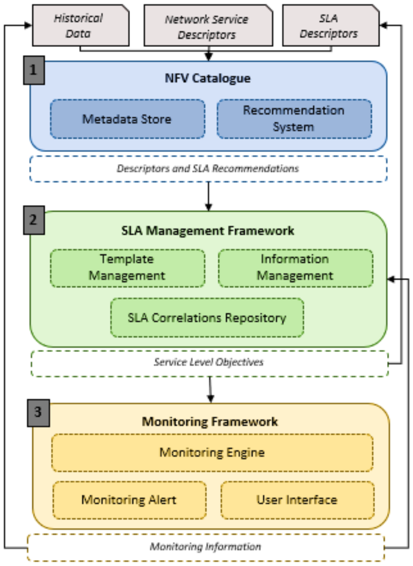


Figure 2: Overall framework architecture.

3. Proposed SLA-Oriented Framework

Considering the challenges presented in Section 2, the developed framework allows to manage the whole lifecycle of SLAs, from the template formulation to the violation detection. Figure 2 depicts the overall architecture of the developed framework. Taking into consideration the inter-connection between a distributed set of users and resources, the implementation of an "SLA-Oriented Framework" aims at governing this interaction. Due to the fact that all running service affects the performance of the service platform, any business parameter (i.e. high-level requirements) included in the SLA, would be linked to recourse demands – encapsulated in the respective policies (i.e. low-level requirements). Initially, a set of NSs, SLAs and historical monitoring data among with their metadata, are stored into the “NFV-Catalogue”. This preliminary information is used from the “Recommendation System”, to provide feedback toward the “SLA Management Framework” for an optimized SLA Template formulation. As soon as this information is available, the SLA Framework should be considered in two phases: a) the SLA Template Management and b) the SLA Information Management. During the first phase, an optimized SLA Template is prepared, by mapping the high-level expressed by the customer into low-level resource attributes needed by the service provider. On the other hand, during the second phase, where the NS instantiation takes place, our proposed framework oversees the obtain monitoring data from the “Monitoring Framework”, which lies on the bottom of the proposed architecture. The implemented “Monitoring Framework” provides support for the QoS management, while it provides adaptable monitoring feedback based on the infrastructure needs. The following sub-sections explain in detail each component.

4. NFV Catalogue

4.1. Metadata Store

As an initial stage, the “NFV-Catalogue”, which is depicted in Figure 2, is positioned to address storage and management necessities of diverse stakeholders’ (i.e. NS developers, NS providers, customers etc.). The main view of the “NFV-Catalogue”, is to provide a repository for persistent storing of the developed VNFs/NS and their corresponding SLAs, attaching them with additional metadata, which are exploited to leverage its functionalities and interfaces for storing, searching and retrieving. Moreover, additional information, like metric significance outcomes and information related to policies and QoS need to be also stored as metadata of the corresponding VNFs. Thus, the “NFV-Catalogue” is deemed to be a multi-faceted data storage, addressing various stakeholder needs while also forming the main and centralized data storage of the 5G ecosystem. The functionality of the “NFV-Catalogue” is predominantly based on the metadata for NS Descriptors and SLA Descriptors. Prior to the attachment of the metadata, through a RESTful API of the “NFV-Catalogue”, the inspection of the validity of the document structure is a critical and necessary step. Since the documents are specified in machine-readable formats, the review of the format contributes to the eliminations of flaws in the “NFV-Catalogue”. Moreover, the attachment of metadata provides the ability of defining uniquely the individual stored machine-readable objects inside the data storage as depicted in Figure 3. Moreover, “NFV-Catalogue” is aligned with the principle of the persistent storage by extending this type of information with valuable fields for successful data integration, accuracy in the format of the document, confirmed time of creation, etc. In this way, it enables the development of enhanced operations for Creating, Retrieving, Updating and Deleting (CRUD) SLA Descriptors inside the “NFV-Catalogue”, while reassures the correct data format of the stored documents (e.g. SLA Templates).

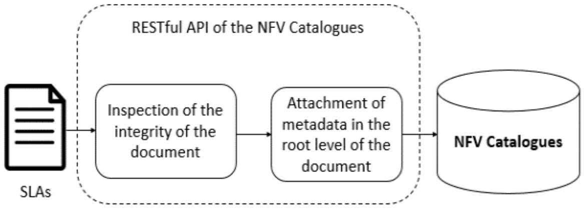

RESTful API of the NFV Catalogues
SLAs
Inspection of the integrity of the document
Attachment of metadata in the root level of the document
NFV Catalogues

Figure 3: SLAs and metadata storage.

Going beyond the conventional data storage, the presented “NFV-Catalogue” provide intelligent functionalities in a 5G environment. Since the types of information vary, one of the necessities that is satisfied by the Catalogues is the full-text search capabilities in structure-agnostic documents. Since the schema of the diverse documents (i.e. NS descriptors, SLA descriptors) is variable, the “NFV-Catalogue” provides searching capabilities without the necessity of indexes. Thus, it provides seamless retrieval abilities in deep-hierarchical machine-readable document structures. Furthermore, besides from the plain NoSQL document store for the diverse descriptors, “NFV-Catalogue” provides a scalable file system for hosting the artifact files, required for the instantiation lifecycle of the VNFs/NSs.

4.2. Recommendation System

After the successful storage of the NS Descriptors among with their SLAs, we consider a “Recommendation System”, which will optimize the SLA Templates creation, by providing important knowledge to the Operator/NS-Provider. Thus, through the plethora of the stored SLAs and the historic data of each end-user, it is undisputedly valuable to tap into them and provide optimum combinations of the available QoS parameters through recommendations. The basic principle of these recommendations is that paramount dependencies presented between the user-to-item activity. The aim of the “Recommendation System” is not to provide only specific and optimum combinations of QoS but also to allow users to profit from them. Thus, the recommendations comprise samples from the relevant actions that were followed in several signed SLAs from similar end-users. In the proposed framework, user-based Collaborative Filtering (CF) was used in order to detect similar users and promote recommendations in these terms [56]. The user is deemed that explicitly rated a combination of SLAs, along with the included QoS parameters. Moreover, what is of paramount importance is the metric of computing the correlations between the end users in the “Recommendation System”. More specifically, the Pearson correlation metric was the most appropriate for the best trade-off in the equation of quality-number of predictions [57]. Although, the aforementioned filtering suffers from two severe issues, the Data Sparsity and the Cold Start [58]. The former, comprises the phenomenon that end users provide a small amount of ratings, contributing to memory complexity and inevitability in training the “Recommendation System”. The latter refers to the difficulty in bootstrapping the “Recommendation System” for newly-introduced users and items. Despite this fact, the CF, in the proposed “SLA-Oriented Framework”, eliminating these challenges by introducing “trust relationships” of end-users. The main idea is the provision of the trust metric for each individual user-to-user and, on top of that, contributes to the predictions of the conventional CF, as depicted in Figure 4. The trust metric relevance of the users is denoted from the received implicit rating of each VNF/NS. With the instantiation/upload of a VNF/NS, the “NFV-Catalogue” automatically receives implicit rating of the respective entity. Thus, the selections can provide the relevance of the diverse users of the platform.

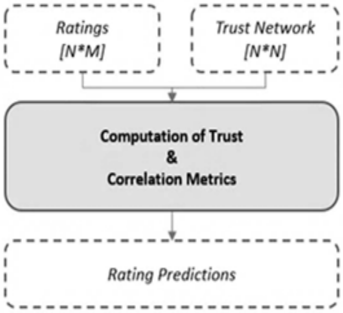


Figure 4: Proposition of trust metrics

5. SLA Management Framework

In the second stage of the proposed "SLA-Oriented Framework", the SLA Templates are generated, as an initial offer to the end-users. After the successful NS instantiation, along with the corresponding SLA, the signed Agreement starts to be monitored so it can fulfill the signed SLOs to the end-user. The current stage is splitted into two sub-phases, a) SLA Template Management, which takes place prior the NS deployment, and b) the Information Management, which takes place during the NS deployment. The internal architecture and workflow inside the "SLA Management Framework" it is depicted in Figure 5.

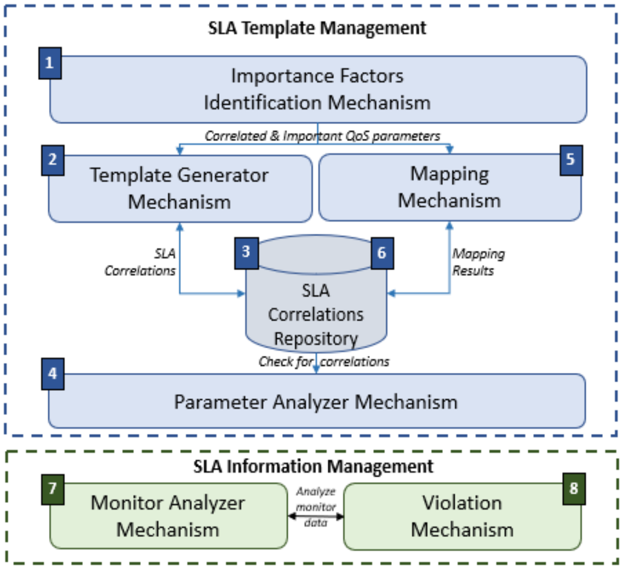


Figure 5: SLA management framework internal architecture

5.1. SLA Correlation Repository

Given the big amount of data generated and analyzed into the SLA Management Framework, an internal repository has been introduced, namely "SLA Correlations Repository", in order to store and manage all the necessary correlations, between end-users, network services, templates, agreements, violations, as well as recourse mapping results. In particular, the correlations between the high-level and the low-level requirements are stored for future analysis. In addition, it keeps track of all the correlations between the generated templates and the linked network services. At the same time agreements information are also located in the repository, along with the end-user's authentication details, as well as the violations records.

5.2. SLA Template Management

The SLA Template Management is the first phase of the SLA Management component while it is in charge of receiving the desired business guarantees (i.e. high-level QoS parameters) from various parties (e.g. NS provider, customer), and formulate an initially SLA Template. It is also responsible for mapping the high-level business guarantees to low level resource attributes, so they are able to be included in the Template, and afterwards monitored through the instantiated Agreement. The SLA Template Management consists of four mechanisms which are going to be further described in the following sub-sections.

Importance Factors Identification Mechanism

In the case of NSs, a challenge arises given that many different entities are setting their requirements for the overall service. Those entities may have specific preferences for resource attributes and potentially additional parameters that can be monitored (e.g. number of sessions) and thus be included into an SLA. To address the aforementioned challenge, we would need to develop a mechanism which would analyze monitoring data and performance information in order to identify dependencies between a VNF's metrics but also realize how these dependencies affect the overall performance of a specific NS. The latter would be reflected to the so called "importance" factors, while it could give feedback to the Operator, about what it is of crucial importance and need to be included in the SLA Template. This is in fact an on-line learning process that updates and dynamically evolves. To this end, the corresponding mechanism, estimates and defines the importance of various QoS parameters through correlation analysis of a) historical SLA parameters, b) predefined policies, c) historical monitoring data and d) recommended QoS metrics. The latter it is depicted in Figure 6.

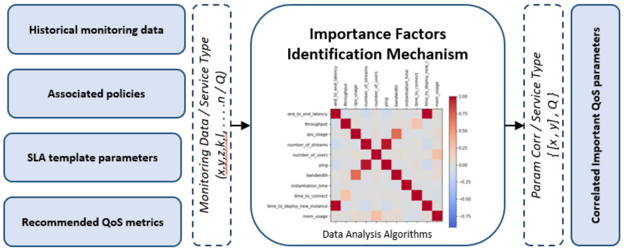


Figure 6: Importance factors identification mechanism

As one understands, the color of the correlogram is white when there is no correlation between the two variables (the correlation value is equal to 0 or close to 0). When the color is deep black it means that our mechanism has calculated a sufficiently large positive correlation, while the deepest grey color indicates there is a large negative correlation.

It should be mentioned that policies are considered out of the proposed framework's scope, and that's why it is assumed that are able to be provided by an external Policy Management Framework. In order to correlate the QoS parameters and define the important ones, analysis libraries were used to analyze the performance measurements of the chained VNFs in a NS. Starting an offline learning process with the gathered data, the mechanism calculates and stores the dependencies in an internal NoSQL database. Then, providing REST APIs as well as a GUI, the end-user can request and get those dependencies of his/her developed NS, and as a result getting an inside knowledge of the NS's performance behavior. Thus, the "Importance Factors Identification Mechanism" can produce essential weight factors and classify parameter dependencies, to suggest and include relevant QoS parameters in the SLA templates. [59-61].

Template Generator Mechanism

As soon as the important QoS parameters are recognized, the "Template Generator Mechanism" takes action, which initially produces the SLA Templates requested by the service provider, and then it is responsible to establish the final Agreement. The "Template Generator Mechanism" as shown in Figure 7, can acquire a set of policies for a clearly defined NS and also historical data of the service provider through the "Monitoring Framework" (i.e. NS performance data, preferences of resource parameters) [62].

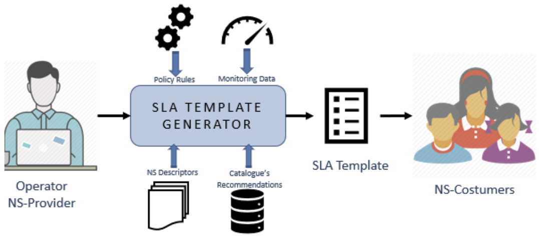


Figure 7: SLA template generator mechanism

In addition, the "Template Generator Mechanism" accesses the "NFV-Catalogue" to retrieve also the NS Descriptor for the corresponding NS, as well as QoS parameters recommendations from the Catalogue's RS. Finally, an also important input to the "Template Generator Mechanism", is the weight factors of the aforementioned recommended QoS parameters, obtained by the "Importance Factors Identification Mechanism". After gathering all the above-mentioned input, the "Template Generator Mechanism", triggers the analysis while it correlates the input in a way that it can formulate an SLA Template with some initial guarantees.

Mapping Mechanism

As soon as the initial Template is formulated, there is essential need to decompose the business guarantees to recourse attributes that can be monitored accordingly by the "Monitoring Framework". In other words, The SLA "Mapping Mechanism" (MM) is the component responsible to translate the high-level requirements described by the end-user into low-level metrics required by the service provider, and vice-versa. More specifically, the MM obtains a set of policy rules from an external Policy Management Framework, a set of low-level and high-level requirements described from the service provider but also from the customer. As a result, the produced output of the MM (i.e. output layer of the ANN) are explicit SLA business metrics, as depicted in Figure 8.

The MM is based on unsupervised learning, using an Artificial Neural Network (ANN) [63]. ANNs can be used to solve this translation problem by mapping service-specific SLOs to resource attributes directly. As they embody a black box approach, ANNs are ideal to be used in an environment where information is not easily transmitted from one entity to another. In addition, ANNs need no knowledge of the inner structure of the NSs [64]. However, it should be noted that they need a representative execution dataset, in order to detect complex, linear or non-linear dependencies.

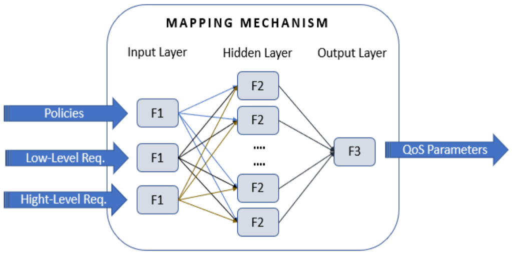


Figure 8: ANN based mapping mechanism

Parameter Analyzer Mechanism

The "SLA Parameter Analyzer" it is capable of deciding whether or not the MM should occur. The decision is made after the component searches into the "SLA Correlations Repository" and check whether the input parameters already correlate with the mapping results calculated and stored.

5.3. SLA Information Management

Once the selected NS has been successfully instantiated, the SLA Information Management should take action by continuous monitoring the service instance. This process aims to optimize the SLA formulation, manage the guaranteed terms in respect with the infrastructure conditions, while also check for any SLA violations. SLA Information Management consists of two mechanisms which are going to be presented explicitly in the next sub-sections. The first is called "SLA Monitor Analyzer Mechanism" while the second "SLA Violation Mechanism".

Monitor Analyzer Mechanism

Starting with the "SLA Monitor Analyzer Mechanism" the component acquires both historical monitoring data for the deployed NS and biases for resource parameters. Sequentially, it should decide whether there is any existed contrast between the mapping results and the runtime monitoring data. Specifically, the "SLA Monitor Analyzer Mechanism", examines the QoS parameters from the "SLA Correlations Repository", with the gathered monitoring data, while it afterwards measures the delta between their values. The MM will obtain the monitoring feedback as an additional dataset and re-train the ANN, in case the delta is greater than "0".

Violation Mechanism

As a final step, in the second stage of the proposed "SLA-Oriented Framework", we consider the "SLA Violation Mechanism". The corresponding mechanism is responsible to ensure that the newly deployed NS would not violate the corresponding SLA upon instantiation, while it successfully fulfills the signed business needs. Nevertheless, the mechanism is also responsible to identify any violations occurred and take the necessary actions (e.g. scale in, scale out). Specifically, upon receipt of the measured metrics through the Monitoring Framework, the "SLA Violation Mechanism" starts an ongoing process of re-adoption. In particular, the mechanism considers the mapping results stored in the "SLA Correlations Repository" and compare them with the real-time monitoring information. In anticipation of future SLA violation threats, the mechanism readjusts the low-level recourse parameters described for the SLA and push them back to the "Monitoring Framework". In this way, both the SLA is fulfilled, while at the same time the infrastructure does not waste recourses, but only in times of real need. Although, in case a violation is not prevented in time, an alert is sent from the "Monitoring Alert Mechanism". Upon receival, the "SLA Violation Mechanism" calculates the overall value of the specific metric and takes decision whether the SLA is violated or not. In case of an SLA violation, the customer is informed by an e-mail, SMS, or even a live push notification.

6. Monitoring Framework

In the third, and final, stage of the proposed "SLA-Oriented Framework", lies the "Monitoring Framework", where, its internal architecture is depicted in Figure 9. The proposed monitoring framework has adopted the SONATA Monitoring Framework [65], and then adapted accordingly in order to support the whole NS lifecycle in respect of the associated SLAs. The corresponding monitoring framework is consisted of: a) the "Monitoring Engine", which collects monitoring data provided by the NSs based on the signed SLA, b) the "Alert Manager", which is responsible to produce alert messages when a violation of a SLA rule is occurred, and c) a "User Interface", used for visualization of the collected monitoring data, while also visualization of each individual rule specified by the "SLA Management Framework".

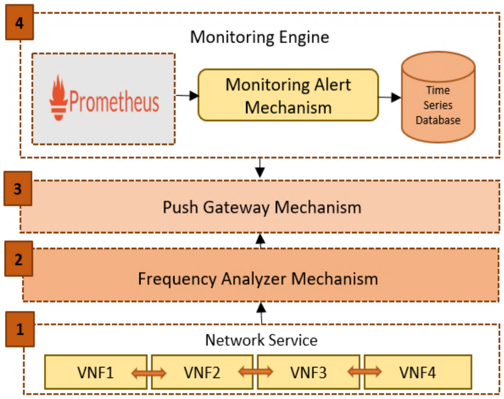


Figure 9: Monitoring Framework: Internal Architecture

6.1. Monitoring Engine

To begin with, a key mechanism of the third stage is the "Monitoring Engine", on the bottom of the proposed "SLA-Oriented Framework". We should point out that, the automation of monitoring an SLA, is a difficult task that demands precise specifications and an adaptable mechanism that collects the right measures and models. At the same time, evaluation of an SLA should occur in specific time frames or when some remarkable events happen. In 5G/SDN ecosystem, where chained VNFs in form of NSs are implemented and deployed on top of a service platform, it becomes essential to create a "Monitoring Engine" which is able to manage a variety of specifications and monitor accordingly the recourses of the virtualized infrastructure. While an SLA is already attached and instantiated through the NS, it is assumed that the desired guarantee terms have been granted to the customer. On the instantiation phase of the service, monitoring rules for the specific NS instance are automatically generated and pushed to the Monitoring Framework, through the "SLA Violation Mechanism". The "Monitoring Engine", which is based on a stable version of Prometheus Monitoring System [66] will undertake the collection of monitoring data from the running services. Prometheus scraps metrics for short- lived jobs either directly or through an intermediate push gateway. It stores all scraped samples locally and executes rules on these data to either add new time series from available data or generate alerts. The benefit of using Prometheus as "Monitoring Engine" is the fact that it was designed for reliability, and the ability to allow quick problems diagnosis.

Push Gateway Mechanism

As it is previously mentioned, Prometheus, can scrap monitoring data exported from the running service instances through the push gateway. To be more specific, the Push Gateway, is a subcomponent of Prometheus, acting like an intermediary service, allowing to forward the monitored data from the "Monitoring Engine" towards the "Monitoring Alert Mechanism", and thus publish them to the external components (i.e. SLA Management Framework, NFV-Catalogue). In our case, the usefulness of the "Push Gateway Mechanism" arrives during the NS scaling up [67]. Scaling up, means that a new VNF is about to start, relieving the service instance when it is actual needed. In this case, the new VNF identifies the "Push Gateway" in terms of authentication and push the monitoring data towards a recognized and reliable host. As a result of the approach, is the fact that the "Monitoring Engine" does not need to know and identify the NS instances, but vice-versa.

Frequency Analyzer Mechanism

After the previous discussion around pushing monitoring data through the "Push Gateway", a challenge arises, in terms of how often and which data are promoted to the engine. Having this in mind, an important parameter of define the above-mentioned challenge is the time interval used to evaluate the resource metrics and guaranteed SLOs (e.g. every two seconds or every two minutes). Although, too frequent pushes may affect negatively the overall system performance, whereas too infrequent pushes may cause heavy SLA violations, due to lack of monitoring metrics towards the "SLA Management Framework" [68]. To this end, an enhancement to Prometheus Monitoring Framework is introduced, namely the "Frequency Analyzer Mechanism", which is based to an adaptive monitoring algorithm. Thus, the "Frequency Analyzer Mechanism" acts as a middle agent between an active connection of the NS instance, the "Push Gateway" and the "Monitoring Alert Mechanism" [17]. Its purpose is to provide highly accurate information about the network's health, while at the same time avoid the production of unnecessary traffic in the network. It aims at adapting during runtime the monitoring time intervals in order to ensure that the data collected and transmitted to the SLA Management Framework, are fruitful and not all raw data. Moreover, it should be noted that the algorithm achieves significant reduction in resource consumption and also reduces the number of SLA violations, due to the pro-active nature of the mechanism.

Time Series Database

Finally, a key component of the Monitoring Framework, is also an internal database. A "Time Series Database" is used for storing and identifying the monitored information, by a metric name and a set of key-value pairs. Following the approach of a time series database, the advantage of having operators for calculating useful information of the monitoring data, is given to the "Monitoring Engine".

6.2. Monitoring Alert Mechanism

The previously discussed monitoring outcomes are going to be published to the external components (i.e. SLA Management Framework, NFV-Catalogue) though the "Monitoring Alert Mechanism". During our research we realized that a message queue system (MQ) was the most appropriate solution for the intercommunication of the aforementioned components. Therefore, RabbitMQ was integrated to the Monitoring Framework, as the message broker for asynchronous messaging [69]. Through the implemented "Monitoring Alert Mechanism", SLA monitoring rules and SLA violations are produced as alerts. Thus, the message (i.e. alert) is pushed to the "Monitoring Alert Mechanism", and all the RabbitMQ consumers are receiving the message for further actions. It should be noted that, one of the consumers is the "SLA Violation Mechanism". This mechanism, acts as an intermediate component between the "Monitoring Engine" and the end-user of the NS instance.

6.3. User Interface

For the visualization of the gathered monitoring data, Grafana is used as an open platform for visualize and beatify monitoring data analytics [70]. Grafana, features an advanced chart query editor that lets the user to quickly browse the metric space, add features, change operating parameters, and more.

7. Evaluation

In order to evaluate the performance of the proposed framework in terms of efficiency and ease of use, our approach was included in the innovative 5G infrastructure environment of the 5GTANGO Service Platform [71]. 5GTANGO project is an EU funded Innovation Action, that enables the flexible programmability of 5G networks with a modular Service Platform so it can bridge the gap between business needs and network operational management systems [72]. The 5GTANGO Service Platform offers the service and functional orchestration features, along with all the supplementary and supporting tools required, like the proposed "NFV-Catalogue", "SLA Management Framework" as well as the "Monitoring Framework".

7.1. Emulation Environment

During the evaluation of the proposed framework a challenge arised, as many of the mechanisms (i.e. Recommendation System, Importance Weight Factors Mechanism, Mapping Mechanism) need apriory behavior knowledge, in order to be able to deal with unknown VNFs/NSs and train their models properly. Moreover, this becomes even more necessary in the emerging DevOps environments, where new versions of NSs are directly deployed in production (i.e. working environment), and therefore no up-to-date monitoring data is available for the updated services. To deal with this challenge, we adopted the OSM supported VIM emulator [77], in order to run on top of it the 5GTANGO VNF/NS benchmarking framework, to automatically execute performance benchmarks of NFV network services and functions [78, 79]. The benchmarking tool automatically gathers performance information about a service, prior to its deployment without requiring dedicated testbeds, resulting to an offline profiling of the service, and the collection of benchmarking data, so they can used as a starting point of the service modeling.

7.2. Working Environment

As it is previously mentioned, the proposed "SLA-Oriented Framework" is implemented inside the 5GTANGOs' SP where the installation guide can be found in [80]. For evaluation purposes, we used the NCSR Demokritos' testbed in order to setup the SP and thus our proposed framework along it. NCSR Demokritos' testbed is the main node of the 5GTANGO infrastructure in Athens, providing the following infrastructure components: a) WAN network, b) Access network, c) datacenter (computing resources for NFVI realization), and d) end user devices and services. To be more specific, as Network Function Virtualization Infrastructure (NFVI) Queens OpenStack multi node deployment with provider networks configuration is used [82]. Also, service chaining is taken care by a Service Function Chaining (SFC) agent that interfaces with the Service Platform in order to fix the chaining between the Network Service components. For the Wide Area Network (WAN) part, the networking is managed by a WAN Infrastructure Manager (WIM) implemented by a Virtual Tenant Network (VTN) running on top of OpenDayLight (version Oxygene) [83]. Finally, the current study used a processing environment which consists of the following elements:

- One Dell R210 used as the Fuel jump host
    - 1xIntel(R) Xeon(R) CPU X3430 @ 2.40GHz
    - 4GB RAM
    - 1TB HDD
- One Dell T5500 functioning as the Controller
    - 2xIntel(R) Xeon(R) CPU X5550@2.67GHz
    - 16GB RAM
    - 1.8 TB HDD
- Three Dell R610 utilized as Compute Nodes
    - 2xIntel(R) Xeon(R) CPU E5620@2.40GHz
    - 64GB RAM
    - 1.8 TB HDD
- One Dell R310 used as NFVI-PoP
    - 1xIntel(R) Xeon(R) CPU X3450@2.67GHz
    - 16GB RAM
    - 465GB HDD

More details about the testbed's topology, hardware/software availability and network recourses can be found in [73]. It should be also noted that with regards to the source code availability, it is currently partially in open-source format, since not all functionalities have finalized yet.

7.3. Experimental Results

In this case study, different stakeholders take place to the overall workflow, in order to provide the necessary inputs. In this study, we consider a) Thomas – the Service Developer, b) Sally – the Commercial Offer Designer (COD), responsible for defining SLAs supporting the whole business, c) Bob – the network engineer of the Service Provider in charge of defining run-time policies, d) Robert – the engineer, in charge of providing monitoring data and e) End-user Customers, such as Brian. The deployment of a network service instance in a service provider's infrastructure comes with the definition of some requirements. This will ensure the performance estimation and the QoS requested by the customer. Quality of service is introduced in the SLA with the definition of SLOs, along with policies for managing the infrastructure accordingly, and thus enforces the respective SLA.

In order to perform a complete testing and evaluation of our approach, an elastic proxy network service is used as depicted in Figure 10, which is consisted of two chained VNFs: a) a HAProxy VNF, configured as a load balancer and b) a Squid VNF configured as a proxy server.

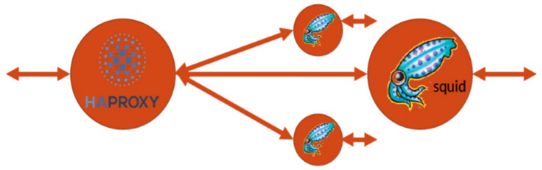


Figure 10: Elastic proxy network service

The end-users use the ingress interface of the HAProxy as proxy IP, and sequentially the HAProxy forwards the incoming requests to one of the Squids in its backend pool. To begin with, the NS-Descriptor, as developed by Thomas, the service developer, is onboarded to the proposed "NFV-Catalogue". The onboarding is made through a Rest API (i.e. Create operation), as mentioned in Section 3.1. Then, the Sally, the Commercial Offer Designer, is responsible to define the customers' facing characteristics of the service, in particular the SLAs. The SLA Generator requires four (4) parameters for a successful generation of the SLA template. The most critical one is the selection of the NS (i.e. the haproxy-squid NS) that is going to be correlated with the newly created SLA template. At the same time an SLA name among with a valid future expiration date and at least one SLO are needed. Once this information is gathered, the SLA Generator mechanism triggers the "Importance Weight Factors Mechanism" and the "Recommendation System". At this moment, the first one will feed the "Template Generator Mechanism" with feedback on *"relevant"* QoS parameters that need to be included in the template, and at the same time, Catalogue's "Recommendation System" will also provide recommendations based on previous created templates and relevant costumers' preferences. It is important to point out that, computing this kind of recommendations and similarities on behalf of Sally, enhance the generation process, by giving the chance to the Commercial Offer Designer to include in the template the most appropriate QoS parameters, and minimize the negotiation between Brian (i.e. the customer) and the service provider. Tables 1 presents high level parameters included in previous customers' SLAs, along with their calculated and monitored values, while Table 2 depicts the similarity between Sally and previous costumers of the haproxy-squid service. It should be mentioned that the values presented in Table 1, are real business requirements, as gathered from Communication Pilot use case of the 5GTANGO project [76].

Table 1: Previous customers low-level metrics

	Customer A	Customer B	Customer C
Availability	0,99.99 %	0.95 %	0.9 %
Jitter	10 ms	25 ms	50 ms
Packet Loss	0.1 %	0.05 %	0.5 %

Table 2: Sally's similarity with previous customers

	Sally	Customer A	Customer B	Customer C
Sally	1	82 %	50 %	30 %

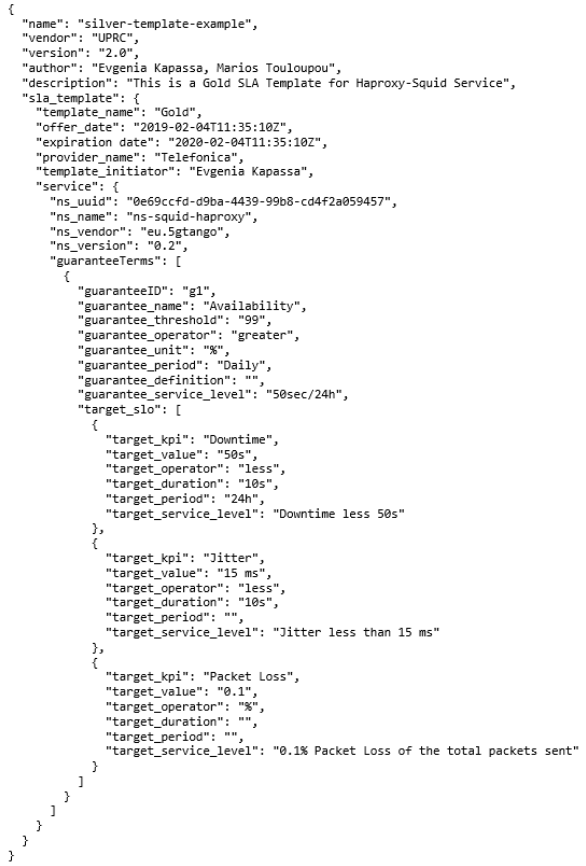
Based on the above results, the "Recommendation System" of the "NFV-Catalogue" and the "Importance Weight Factors Mechanism", Sally is recommended to include the following guarantees to the SLA Template, as presented in Table 3.

Table 3: QoS recommendations for Sally

	Availability	Jitter	Packet Loss
Sally	99 %	15 ms	0.1

Afterwards, the generation of the SLA template, as an initial offer to the NS-costumers is triggered. The outcome is an SLA Descriptor that will be onboarded to the NFV-Catalogue, and it is based on WS-Agreement specification [74-75]. The main SLA Templates building blocks of the reference model include the root element, the SLA template and the service elements, as depicted in Figure 11.

Afterwards, Brian, the end-user customer, browses through the available NSs, select the haproxy-squid NS and instantiate it, in order to be deployed in the SP. During the instantiation process the customer triggers automatically the one-shot negotiation process through the "Mapping Mechanism", by selecting the previously generated SLA Template for the specified NS. Based on the business requirements of the customer, the SLA can be accepted, or a new negotiation process can be initiated. Once the SLA Manager has collected all the relevant datasets via the "Parameter Analyzer", checks if there is already a combination in the "SLA Correlations Repository", between the latter and the already existing mapping results, in order to decide whether the process of the "Mapping Mechanism" should be triggered or not. In case there is not a correlation yet, the operator's low-level requirements, the costumer's high-level business needs and the policies, are mapped in order to produce the actual QoS parameters that can finally be included in the SLA. The objective is to forecast the performance and the quality that is required, to be agreed and signed in the final SLA. Alternatively, if there is an already a combination between the input dataset (i.e. requirements obtained from the operator and the costumer) and the stored mapping results, the SLA Manager bypasses the mapping process and dynamically creates the final Agreement. In order to investigate this mapping, we needed to predict the performance of the Network Service on top of the infrastructure.



```
{
  "name": "silver-template-example",
  "vendor": "UPRC",
  "version": "2.0",
  "author": "Evgenia Kapassa, Marios Touloupou",
  "description": "This is a Gold SLA Template for Haproxy-Squid Service",
  "sla_template": {
    "template_name": "Gold",
    "offer_date": "2019-02-04T11:35:10Z",
    "expiration date": "2020-02-04T11:35:10Z",
    "provider_name": "Telefonica",
    "template_initiator": "Evgenia Kapassa",
    "service": {
      "ns_uuid": "0e69ccfd-d9ba-4439-99b8-cd4f2a059457",
      "ns_name": "ns-squid-haproxy",
      "ns_vendor": "eu.5gtango",
      "ns_version": "0.2",
      "guaranteeTerms": [
        {
          "guaranteeID": "g1",
          "guarantee_name": "Availability",
          "guarantee_threshold": "99",
          "guarantee_operator": "greater",
          "guarantee_unit": "%",
          "guarantee_period": "Daily",
          "guarantee_definition": "",
          "guarantee_service_level": "50sec/24h",
          "target_slo": [
            {
              "target_kpi": "Downtime",
              "target_value": "50s",
              "target_operator": "less",
              "target_duration": "10s",
              "target_period": "24h",
              "target_service_level": "Downtime less 50s"
            },
            {
              "target_kpi": "Jitter",
              "target_value": "15 ms",
              "target_operator": "less",
              "target_duration": "10s",
              "target_period": "",
              "target_service_level": "Jitter less than 15 ms"
            },
            {
              "target_kpi": "Packet Loss",
              "target_value": "0.1",
              "target_operator": "%",
              "target_duration": "",
              "target_period": "",
              "target_service_level": "0.1% Packet Loss of the total packets sent"
            }
          ]
        }
      ]
    }
  }
}
```

Figure 11: SLA descriptor example

The output is categorized between simple mapping results and complex ones. A simple mapping result maps "end-to-end", from low-level to high-level. For instance, mapping the low-level metric "downtime" to high level SLA parameter "availability". Complex mapping results include predefined formulations to calculate specific SLA parameters using low level resource metrics. Table 4 presents an example of a complex mapping result.

Table 4: Complex mapping result example

Low-Level Metric	SLA Parameter	Mapping Formulation
downtime, uptime	Availability (A)	$A = 1 - \frac{downtime}{uptime}$

In order to investigate the performance of the "Mapping Mechanism", the emulation environment provided a data set consisted of 360 data points. Of these, 50% was used in order to train the network model, for the haproxy-squid NS, as depicted in Figure 12, resulting into 180 data points. For validation purposes another 20% was used during training, meaning 72 data points. Finally, the overall network capability measured against the remaining 30% of the data set, (on which the model was not trained), simulating a real test situation.

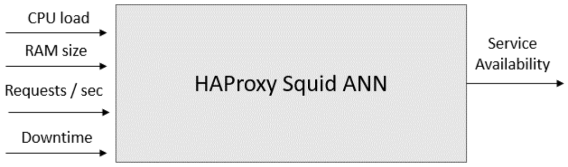


Figure 12: ANN model for HAProxy-Squid Network Service

Then, the remaining 30% was used to check the system's reliability in accurately predicting the NS's QoS levels in the Deployment Environment. In order to check the absolute differences between the ANN's prediction and the actual monitoring observation, the Mean Absolute Error (MAE) was used, as shown in (1), where 'n' is the number of data points, y_j represents the observed values and $\widehat{y}_j$ the predicted values. The MAE result is depicted in Table 5.

$$MAE = \frac{1}{n}\sum_{j=1}^{n}|y_j - \widehat{y_j}| \quad (1)$$

Table 5: Complex mapping result example

ANN Model	Neuron per Layer	Mean Absolute Error
HAProxy-Squid NS Availability	4 – 3 - 1	2.75 %

The aforementioned ANNs are feed-forward back-propagation networks, trained with the Levenberg-Marquardt algorithm [81]. The criterion for performance was the Mean Square Error (MSE) in the training set, while it was trained for 100 periods, for a training time of 1 minute. Putting all this together, we have the general formula for calculating the MSE in (2).

$$MSE = \frac{1}{n}\sum_{j=1}^{n}(y_j - \widehat{y_j})^2 \quad (2)$$

It should also be stated that, if the customer wishes to change the recourse parameters that the HAProxy-Squid service can handle, the 5GTANGO platform, which incorporates the service presented in this paper, offers the capability of re-triggering the "Mapping Mechanism" and retrain the model.

Next, the instantiation of the NS takes place. To this end, the deployment aims to enable execution of the NS according to the QoS requirements, while at the same time appropriate monitoring, allows the measurement of QoS parameters at both service and infrastructure levels targeting events of resource provisioning estimation and decision making. For this reason, the Monitoring Framework access the Point of Presence (PoP) that the NS is deployed into and gathers monitoring information for the haproxy-backend-downtime, jitter and packet loss, in order to measure its availability. At this point, the "Frequency Analyzer Mechanism" takes place, by adjusting the monitoring time intervals during runtime to ensure that the data collected and transmitted to the SLA management framework are meaningful. At first, data are collected and compared with linear increase of time intervals, until they rich an initial time threshold. In our case the collected monitored values of the HAProxy-Squid service downtime were below the certain threshold, indicating that the network has changed towards a better state. Therefore, after the service had pushed the monitored data, the "Frequency Analyzer Mechanism" multiply the time the change occurred, with β, in order to increase the data transmission interval. As a consequence, the new timeout is higher, and the probes will collect data with a linear increase over a longer time period, without wasting recourses.

As depicted in Figure 13, the X axis shows a linear increase in data collection time from the samples. The latter begins in the first second and increases linearly until it reaches the fifth second. while a change in the metric value occurs in the 5th sec. The "Frequency Analyzer Mechanism" commands the probes to send their data to the "Push Gateway", and at the same time increases the timeout by the current time* β (i.e. current time is the time when the significant change occurred).

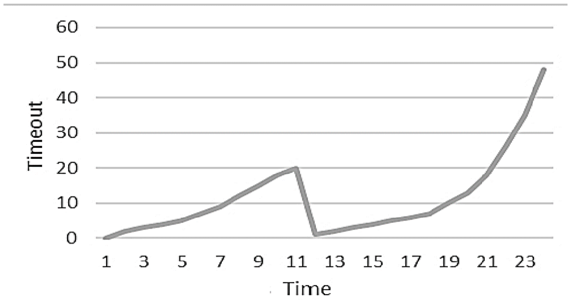


Figure 13: Time intervals adjustments

In our scenario, the equation returned the value of 18 seconds, meaning that the data collection will continue after a linear increase in time until the new timeout which set to 18th second is reached. Moreover, the monitoring process of the proposed approach was tested firstly using the standard Prometheus framework, and then by enhancing it with the aforementioned "Frequency Analyzer Mechanism". Figure 13 depicts the difference regarding the network workload (i.e. throughput in terms of data requests towards the push gateway per second). Finally, when the network service has completed its lifecycle, Brian, is responsible for terminating it. The termination process, as well as all the aforementioned procedures, are taking place in a user-friendly way, through a unified Portal.

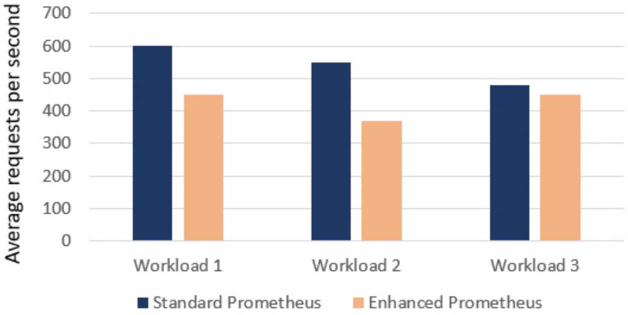


Figure 13: Network workload with and without using the "Frequency Analyzer Mechanism"

8. Conclusions

Based on the aforementioned evaluation, we presented a SLA Management Framework that is used to map high- level business parameters to low- level attributes of resources. This framework is integrated in the 5GTANGO Service Platform for the autonomous management of SLAs. We considered a generic approach that is based on ANNs in order to efficiently be used as a mediator for the network provider and the end-user. We considered an ANN based approach that can be used as a mediator between the end-user and the provider. Furthermore, we have introduced a mechanism to determine the importance of various QoS parameters so that the "relevant" ones be included in the SLAs for better QoS assurance. The proposed monitoring framework goes a step forward from the traditional implementation, by preventing any unnecessary traffic to the network, and also by providing real-time and high accurate information for better QoS assurance. Last but not least, the authors have presented a beyond a plain data storage to an enriched information – driven repository, the so-called "NFV-Catalogue". The aggregation of the stored information allows the mechanism to apply recommender system techniques, build on QoS predictions and SLA recommendation systems. Regarding the three major components of the proposed architecture (i.e. the NFV-Catalogue, SLA Management Framework and Monitoring Framework), we conclude to the following, based on the captured experimental results.

All things considered, provisioning of resources in a virtualized 5G infrastructure is a challenging task, that still needs a lot of investigation. Therefore, we plan to extend the framework in order to enable Quality of Experience (QoE) enforcement. This kind of enforcement could be done by adopting the infrastructure recourses accordingly during runtime, considering parameters based on Catalogue's recommendations as well as monitoring feedback. Moreover, we tend to enhance the SLA violations management, by providing violations prediction models, in order to prevent day zero violations. Additionally, the currently proposed framework is able to monitor and manage business guarantees in a single` 5G environment. Therefore, we envision to manage a 5G network with multiple domains, enabling higher level of integration, and at the same time adapt the proposed architecture from a wide range of verticals, enabling higher level of abstraction.

Conflict of Interest

The authors declare no conflict of interest.

Acknowledgment

This work has been partially supported by the 5GTANGO project, funded by the European Commission under Grant number H2020ICT-2016-2 761493 through the Horizon 2020 and 5G-PPP programs (http://5gtango.eu).

References

[1] F. Hu, Opportunities in 5G Networks: A Research and Development Perspective, CRC Press, 2016.

[2] A. Osseiran et al., "Scenarios for 5G mobile and wireless communications: the vision of the METIS project," in IEEE Communications Magazine, 52(5), 26-35, 2014. doi: 10.1109/MCOM.2014.6815890

[3] H. Su, X. Zhang, "Cross-Layer Based Opportunistic MAC Protocols for QoS Provisionings Over Cognitive Radio Wireless Networks," in IEEE Journal on Selected Areas in Communications, 26(1), 118-129, 2008. doi: 10.1109/JSAC.2008.080111

[4] D. Wu, R. Negi, "Effective capacity: a wireless link model for support of quality of service," in IEEE Transactions on Wireless Communications, 2003. doi: 10.1109/TWC.2003.814353

[5] C. Liang, F. R. Yu and X. Zhang, "Information-centric network function virtualization over 5g mobile wireless networks," in IEEE Network, 29 (3), 68-74, 2015. doi: 10.1109/MNET.2015.7113228

[6] Q. Zhu, X. Zhang, "Game-theory based power and spectrum virtualization for maximizing spectrum efficiency over mobile cloud-computing wireless networks," in 49th Annual Conference on Information Sciences and Systems (CISS), 2015. doi: 10.1109/CISS.2015.7086818

[7] Samsung Developers, "5G Requirements", Online: https://developer.samsung.com/tech-insights/5G/5g-requirements

[8] Huawei Technologies Co. Ltd, "5G Network Architecture: A High-Level Perspective", Online: https://www.huawei.com/minisite/hwmbbf16/insights/5G-Nework-Architecture-Whitepaper-en.pdf

[9] E. Casalicchio, V. Cardellini, G. Interino, M. Palmirani, "Research challenges in legal-rule and QoS-aware cloud service brokerage", in Future Generation Computer Systems, 78(1), 211-223, 2018. doi: https://doi.org/10.1016/j.future.2016.11.025.

[10] Anuta Networks, "Top 6 Challenges for Service Assurance in NFV", Online: https://www.anutanetworks.com/top-6-challenges-for-service-assurance-in-nfv/

[11] A. Tabebordbar, A. Beheshti, "Adaptive Rule Monitoring System" in 1st International Workshop on Software Engineering for Cognitive Services, 2018. doi: 978-1-4503-5740-1

[12] L.B. López, J. M. Vidal, L.G. Villalba, "An Approach to Data Analysis in 5G Networks", in Entropy, 19(2), 74, 2017. doi: 10.3390/e19020074

[13] Centina, "Service Assurance Critical to SDN/NFV Success", Online: http://www.centinasystems.com/service-assurance-critical-sdnnfv-success/

[14] E. Kapassa, M. Touloupou, A. Mavrogiorgou, D. Kyriazis, "5G & SLAs: Automated proposition and management of agreements towards QoS enforcement" in 21st Conference on Innovation in Clouds, Internet and Networks and Workshops, 2018. doi: 10.1109/ICIN.2018.8401587

[15] E. Kapassa, M. Touloupou and D. Kyriazis, "SLAs in 5G: A Complete Framework Facilitating VNF- and NS- Tailored SLAs Management," in 32nd International Conference on Advanced Information Networking and Applications Workshops, 2018. doi: 10.1109/WAINA.2018.00130

[16] S. Benkner and G. Engelbrecht, "A Generic QoS Infrastructure for Grid Web Services," in Advanced Int'l Conference on Telecommunications and Int'l Conference on Internet and Web Applications and Services, 2006. doi: 10.1109/AICT-ICIW.2006.16

[17] M. Touloupou, E. Kapassa, A. Kiourtis and D. Kyriazis, "Cheapo: An algorithm for runtime adaption of time intervals applied in 5G networks," in Fifth International Conference on Software Defined Systems. doi: 10.1109/SDS.2018.8370420

[18] N. Sfondrini, G. Motta, L. You, "Service level agreement (SLA) in Public Cloud environments: A Survey on the current enterprises adoption," in 5th International Conference on Information Science and Technology, 2015. doi: 10.1109/ICIST.2015.7288964

[19] Xi Zhang, Jia Tang, Hsiao-Hwa Chen, Song Ci and M. Guizani, "Cross-layer-based modeling for quality of service guarantees in mobile wireless networks," in IEEE Communications Magazine, 44(1), 100-106, 2006. doi: 10.1109/MCOM.2006.1580939

[20] L. Zhu, X. Liu, "Technical Target Setting in QFD for Web Service Systems Using an Artificial Neural Network," in IEEE Transactions on Services Computing, 3(4),338-352, 2010. doi: 10.1109/TSC.2010.45

[21] V. C. Emeakaroha, I. Brandic, M. Maurer, S. Dustdar, "Low level Metrics to High level SLAs - LoM2HiS framework: Bridging the gap between monitored metrics and SLA parameters in cloud environments," in International Conference on High Performance Computing & Simulation, 2010. doi: 10.1109/HPCS.2010.5547150

[22] T. Cucinotta et al., "Virtualised e-Learning with real-time guarantees on the IRMOS platform," 2010 IEEE International Conference on Service-Oriented Computing and Applications, 2010. doi: 10.1109/SOCA.2010.5707166

[23] G. Kousiouris, D. Kyriazis, S. Gogouvitis, A. Menychtas, K. Konstanteli, T. Varvarigou, "Translation of application-level terms to resource-level attributes across the Cloud stack layers," in IEEE Symposium on Computers and Communications, 2011. doi: 10.1109/ISCC.2011.5984009

[24] K. Samdanis, X. Costa-Perez, V. Sciancalepore, "From network sharing to multi-tenancy: The 5G network slice broker," in IEEE Communications Magazine, 54(7),32-39, 2016. doi: 10.1109/MCOM.2016.7514161

[25] X. Foukas, G. Patounas, A. Elmokashfi, M. K. Marina, "Network Slicing in 5G: Survey and Challenges," in IEEE Communications Magazine, 55(5), 94-100, 2017. doi: 10.1109/MCOM.2017.1600951

[26] V. Tikhvinskiy, G. Bochechka, "Prospects and QoS Requirements in 5G Networks", in Journal of Telecommunications and Information Technology, 1, 23-26, 2015. doi:

[27] ETSI, "ETSI TR 102 889-2 V1.1.1, Technical Report", Online: https://www.etsi.org/deliver/etsi_tr/102800_102899/10288902/01.01.01_60/tr_10288902v010101p.pdf

[28] G.C. Madueño, C. Stefanović, P. Popovski, "Reliable Reporting for Massive M2M Communications With Periodic Resource Pooling" in IEEE Wireless Communications Letters, 3(4), 429-432, 2014. doi: 10.1109/LWC.2014.2326674

[29] J. Tang X. Zhang, "Quality-of-Service Driven Power and Rate Adaptation over Wireless Links," in IEEE Transactions on Wireless Communications, 6(8), 3058-3068, 2007. doi: 10.1109/TWC.2007.051075

[30] A. Asadi, V. Mancuso, "A Survey on Opportunistic Scheduling in Wireless Communications," in IEEE Communications Surveys & Tutorials, 15(4), 1671-1688, 2013. doi: 10.1109/SURV.2013.011413.00082

[31] T. Guo, R. Arnott, "Active LTE RAN Sharing with Partial Resource Reservation," in IEEE 78th Vehicular Technology Conference), 2013. doi: 10.1109/VTCFall.2013.6692075

[32] K. Hammad, A. Moubayed, S. L. Primak and A. Shami, "QoS-Aware Energy and Jitter-Efficient Downlink Predictive Scheduler for Heterogeneous Traffic LTE Networks," in IEEE Transactions on Mobile Computing, 17(6), 1411-1428, 2018. doi: 10.1109/TMC.2017.2771353

[33] J. Pérez-Romero, O. Sallent, R. Ferrús, R. Agustí, "On the configuration of radio resource management in a sliced RAN," in IEEE/IFIP Network Operations and Management Symposium, 2018. doi: 10.1109/NOMS.2018.8406280

[34] I. da Silva et al., "Impact of network slicing on 5G Radio Access Networks," in European Conference on Networks and Communications, 2016. doi: 10.1109/EuCNC.2016.7561023

[35] A. Tootoonchian, M. Ghobadi, Y. Ganjali, OpenTM: Traffic Matrix Estimator for OpenFlow Networks, Springer, 2010

[36] C. Yu, Lumezanu, Y. Zhang, V. Singh, G. Jiang, H.V. Madhyastha, FlowSense: Monitoring Network Utilization with Zero Measurement Cost, Springer, 2013.

[37] M. Yu, J. Yu, L. Rui, M. Rui, Software Defined Traffic Measurement with OpenSketch, USENIX Association, 2013.

[38] P. Trakadas et al., "Scalable monitoring for multiple virtualized infrastructures for 5G services" in The International Symposium on Advances in Software Defined Networking and Network Functions Virtualization, 2018. doi: http://hdl.handle.net/1854/LU-8569066

[39] SONATA Project Consortium, "SONATA NFV: Agile Service Development and Orchestration in 5G Virtualized Networks", Online: http://www.sonata-nfv.eu/

[40] T. Maksymyuk, S. Dumych, M. Brych, D. Satria, M. Jo, "An IoT Based Monitoring Framework for Software Defined 5G Mobile Networks" in Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication, 2017. doi: 10.1145/3022227.3022331

[41] M. Liyanage et al., "Software Defined Monitoring (SDM) for 5G mobile backhaul networks," in IEEE International Symposium on Local and Metropolitan Area Networks, 2017. doi: 10.1109/LANMAN.2017.7972144

[42] A. H. Celdrán, M.G. Pérez, F. J. García Clemente, G. M. Pérez, "Automatic monitoring management for 5G mobile networks" in Procedia Computer Science, 110, 328-335, 2017. https://doi.org/10.1016/j.procs.2017.06.102.

[43] N. Nikaein. et al., "Network store: Exploring slicing in future 5g networks" in Proceedings of the 10th International Workshop on Mobility in the Evolving Internet Architecture, 2015. doi: 10.1145/2795381.2795390

[44] T-NOVA Project Consortium, "T-NOVA: Network Functions As-a-Service Over Virtualized Infrastructures", Online: http://www.t-nova.eu/

[45] M. Ersue, "ETSI NFV Management and Orchestration - An Overview", Online: https://www.ietf.org/proceedings/88/slides/slides-88-opsawg-6.pdf

[46] R.E. Núñez-Valdéz, et al., "Implicit feedback techniques on recommender systems applied to electronic books" in Computers in Human Behavior, 28(4), 1186-1193, 2012. https://doi.org/10.1016/j.chb.2012.02.001.

[47] K. Oku, R. Kotera, K. Sumiya, Geographical Recommender System Based on Interaction Between Map Operation and Category Selection, ACM, 2010.

[48] J. Serrano-Guerrero, E. Herrera-Viedma, J.A. Olivas, A. Cerezo, F.P. Romero, "A google wave-based fuzzy recommender system to disseminate information in University Digital Libraries 2.0" in Information Sciences, 181(9), 1503-1516, 2011. doi: https://doi.org/10.1016/j.ins.2011.01.012.

[49] F. Zaman, G. Hogan, S. V. Der Meer, J. Keeney, S. Robitzsch, G. Muntean, "A recommender system architecture for predictive telecom network management," in IEEE Communications Magazine, 53(1), 286-293, 2015. doi: 10.1109/MCOM.2015.7010547

[50] Y. Koh, R. Knauerhase, P. Brett, M. Bowman, Z. Wen, C. Pu, "An Analysis of Performance Interference Effects in Virtual Environments," in IEEE International Symposium on Performance Analysis of Systems & Software, 2007. doi: 10.1109/ISPASS.2007.363750

[51] ONF, "Software-Defined Networking (SDN) Definition", Online: https://www.opennetworking.org/sdn-definition/

[52] OpenStack Project, "OpenStack Networking ("Neutron")", Online: https://wiki.openstack.org/wiki/Neutron

[53] B. Hoff, “IT service-level agreements and SDN: Assuring virtualization performance”, Online: https://searchnetworking.techtarget.com/tip/IT-service-level-agreements-and-SDN-Assuring-virtualization-performance

[54] SLA-Ready, “Cloud SLA lifecycle”, Online: http://www.sla-ready.eu/cloud-sla-lifecycle

[55] S. Van Rossem et al., "A network service development kit supporting the end-to-end lifecycle of NFV-based telecom services," in IEEE Conference on Network Function Virtualization and Software Defined Networks, 2017. doi: 10.1109/NFV-SDN.2017.8169859

[56] C. Saluja, “Collaborative Filtering based Recommendation Systems exemplified”, Online: https://towardsdatascience.com/collaborative-filtering-based-recommendation-systems-exemplified-ecbffe1c20b1

[57] J. Jin, S. Zhang, L. Li, T. Zou, "A Novel System Decomposition Method Based on Pearson Correlation and Graph Theory" in IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS), 2018. doi: 10.1109/DDCLS.2018.8515967

[58] X. He, L. Liao, H. Zhang, L. Nie, T. Chua, “Neural Collaborative Filtering”, in Proceedings of the 26th International Conference on World Wide Web, 2017. doi: 10.1145/3038912.3052569

[59] Z. u. Rehman, F. K. Hussain, O. K. Hussain, "Towards Multi-criteria Cloud Service Selection," in Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, 2011. doi: 10.1109/IMIS.2011.99

[60] Z. ur Rehman, O. K. Hussain, S. Parvin, F. K. Hussain, "A Framework for User Feedback Based Cloud Service Monitoring," in Sixth International Conference on Complex, Intelligent, and Software Intensive Systems, 2012. doi: 10.1109/CISIS.2012.157

[61] T. Halabi, M. Bellaiche, "Evaluation and selection of Cloud security services based on Multi-Criteria Analysis MCA," in International Conference on Computing, Networking and Communications, 2017. doi: 10.1109/ICCNC.2017.7876216

[62] B. K. Tripathy, A. G. Sethy, P. Bera, M. A. Rahman, "A Novel Secure and Efficient Policy Management Framework for Software Defined Network," in IEEE 40th Annual Computer Software and Applications Conference, 2016. doi: 10.1109/COMPSAC.2016.31

[63] S. Hussain, R. Atallah, A. Kamsin, J. Hazarika, “Classification, Clustering and Association Rule Mining in Educational Datasets Using Data Mining Tools: A Case Study” in Silhavy R. (eds) Cybernetics and Algorithms in Intelligent Systems, Springer, 2019. doi: https://doi.org/10.1007/978-3-319-91192-2_21

[64] G. Kousiouris, D. Kyriazis, S. Gogouvitis, A. Menychtas, K. Konstanteli, T. Varvarigou, "Translation of application-level terms to resource-level attributes across the Cloud stack layers," in IEEE Symposium on Computers and Communications, 2011. doi: 10.1109/ISCC.2011.5984009

[65] SONATA Project Consortium, “D4.3 Service Platform First Operational Release and Documentation”, Online: http://www.sonata-nfv.eu/sites/default/files/sonata/public/content-files/deliverables/SONATA%20D4.3%20Service%20platform%20operational%20release%20and%20documentation.pdf

[66] rometheus Authors, “Prometheus: From metrics to insight”, Online: https://prometheus.io/

[67] S. Dutta, T. Taleb, A. Ksentini, "QoE-aware elasticity support in cloud-native 5G systems," in IEEE International Conference on Communications (ICC), 2016. doi: 10.1109/ICC.2016.7511377

[68] C. Vincent et al., “Towards autonomic detection of SLA violations in Cloud infrastructures”, in Future Generation Computer Systems, 28(7), 1017 – 1029, 2012. doi: https://doi.org/10.1016/j.future.2011.08.018.

[69] Pivotal Software, “RabbitMQ”, Online: https://www.rabbitmq.com/

[70] Grafana Labs, “Grafana: The open platform for beautiful analytics and monitoring”, Online: https://grafana.com/

[71] C. Parada et al., "5Gtango: A Beyond-Mano Service Platform," in European Conference on Networks and Communications, 2018. doi: 10.1109/EuCNC.2018.8443232

[72] 5GTANGO Project Consortium, “5GTANGO: 5G Development and Validation Platform for global Industry-specific Network Services and Apps”, Online: https://5gtango.eu/

[73] 5GTANGO Consortium, “D6.1 Infrastructures, Continuous integration approach”, 2017, Online: https://5gtango.eu/project-outcomes/deliverables/38-d6-1.html

[74] A. Andrieux et al., “Web Services Agreement Specification (WS Agreement)”, Grid Resource Allocation Agreement Protocol (GRAAP) WG, 2011, Online: https://www.ogf.org/documents/GFD.107.pdf

[75] R. Kabert, G. Katsaros, T. Wang, "A RESTful implementation of the WS-agreement specification", in Proceedings of the Second International Workshop on RESTful Design, 2011. doi: 10.1145/1967428.1967444

[76] SONATA Project Consortium, “Real Time Communications”, Online: https://5gtango.eu/index.php/about-5g-tango/47-real-time-communications

[77] M. Peuster, H. Karl, S. v. Rossem, “MeDICINE: Rapid Prototyping of Production-Ready Network Services in Multi-PoP Environments” in IEEE Conference on Network Function Virtualization and Software Defined Networks, 2016. doi: 10.1109/NFV-SDN.2016.7919490

[78] M. Peuster, H. Karl, “Profile Your Chains, Not Functions: Automated Network Service Profiling in DevOps Environments” in IEEE Conference on Network Function Virtualization and Software Defined Networks, 2017. Doi: 10.1109/NFV-SDN.2017.8169826

[79] M. Peuster, H. Karl, “Understand Your Chains: Towards Performance Profile-based Network Service Management” in Fifth IEEE European Workshop on Software Defined Networks, 2016. doi: 10.1109/EWSDN.2016.9

[80] 5GTANGO Project Consortium, “5GTNAGO Service Platform Installation Guide” Online: https://sonata-nfv.github.io/component_installation

[81] G. Kousiouris et al., “Distributed Interactive Real-time Multimedia Applications: A Sampling and Analysis Framework” in 1st International Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems, 2010. doi: https://eprints.soton.ac.uk/id/eprint/272323

[82] OpenStack Project, “OpenStack Queens Expands Support for GPUs, Containers to Meet Edge, HA, AI Workload Demands”, Online: https://www.openstack.org/software/queens/

[83] OpenDaylight Project, “OpenDaylight”, Online: https://www.opendaylight.org