Towards Closed Loop 5G Service Assurance Architecture for Network Slices as a Service

Min Xie*, Wint Yi Poe†, Yue Wang‡, Andres J. Gonzalez*, Ahmed M. Elmokashfi§,
Joao Antonio Pereira Rodrigues¶, Foivos Michelinakis§
*Telenor Research, Telenor, Norway; †Huawei Technologies Duesseldorf GmbH, Munich, Germany;
‡Samsung Electronics, UK; §Simula Research Lab, Norway; ¶Nokia, Portugal

***Abstract*—5G intends to use network slicing to support multiple vertical industries. The dynamic resource sharing and diverse customer requirements bring new challenges towards service assurance (SA), such as automation and customer-centric. As a response to these challenges, this paper proposes a hierarchical, modular, distributed, and scalable SA architecture. This paper highlights an important key feature *SA coordination*, which is facilitated by three new SA functions, *SA interpretation*, *SA policy management*, and *data fabric*. Three closed-loops are introduced to coordinate and realize automation of service management. Challenges associated with realizing SA are briefly discussed and will be addressed by leveraging the 5G infrastructure developed within the H2020-ICT-17 project 5G-VINNI.**

I. Introduction

The diverse service requirements in 5G drive the shift towards a more dynamic, flexible, and cost-effective network. Network slicing is a key enabler by instantiating and operating multiple virtualized networks on a common physical network infrastructure [1]. The success of 5G with network slicing depends on how the customer requirements are met. Service Assurance (SA) is hence a key mechanism to guarantee such a success.

With 5G network slicing, SA becomes more challenging for a number of reasons. First, multiple customer-facing services (CFSs) with various service requirements will be served simultaneously. As a result, there is no longer an one-one mapping between CFS and the network like in the legacy networks. On one hand, a single CFS may be provisioned by multiple slices. On the other hand, multiple CFSs may share part of one slice. SA needs to gain a full understanding of such complex relationship and then customize SA solutions for each CFS. Second, 5G vertical customers may require low latency and high reliability, which are assured via real-time monitoring and automated management. However, most conventional SA solutions are realized offline in a manual and post-processing way with relatively simple analytics. The complexity and scale brought by virtualization and slicing make these SA solutions inadequate. In addition, most existing SA solutions are static and do not adapt to the network conditions whereas adaptivity is necessary to deal with the frequently changing network conditions in 5G. Last but not least, a CFS may span across multiple technology domains (access network, core network, transport network, private or public clouds for applications), supplied by multiple vendors and operated by multiple service providers. SA needs to coordinate them and generate an end-to-end (E2E) view for the customer.

As a result, SA for 5G Network Slicing is an open research issue which is not fully addressed. Despite the many standardization activities in defining the SA architecture, each of them has its limitations and does not solve the full challenges stated above. For example, although [2] specifies SA as part of closed-loop automation in the E2E scope considering the integration with the underlying management domain, the details of closed-loop automations inside each management domain have not been fully studied as of the time of writing. The work in [3] studies performance assurance at the Network and Network Slice layers, however, the customer-facing service level assurance is not fully addressed yet. Similarly [4] identifies Network Data Analytics Function (NWDAF) in 5G CN to assist SA, but the closed-loop SA at 5G core network (CN) is not considered in the current 5G architecture. Furthermore, works from [5] performs the same direction but with the focus on SLAs in cloud services. Our previous work [6], [7] studied the SA architecture of NFV, which does not fully applicable to network slicing.

This paper proposes a SA architecture to address the aforementioned requirements, by leveraging the network slicing architecture [8] available from an EU H2020 project 5G-VINNI[1]. In addition, open issues and critical challenges are also discussed and to be considered for the future work in realizing SA for network slicing.

II. Network Slicing Architecture Across 5G-VINNI Facility Sites

5G-VINNI is an EU project aiming to accelerate the uptake of 5G in Europe by providing an end-to-end (E2E) facility that validates the performance of new 5G technologies and use cases by operating trials of advanced vertical services. An 5G-VINNI facility site is the deployment of the 5G-VINNI architecture in one administrative domain (e.g. one operator). The 5G-VINNI facility builds vertical services based on network slicing and is designed to facilitate diverse vertical use cases.

One major deliverable of 5G-VINNI is a design of architecture to support E2E network slicing [8], as shown in Fig. 1. A network slice (e.g., eMBB, mIoT, URLLC [1]) may be hosted within one Facility Site or traverse multiple Facility Sites, managed and orchestrated by the E2E service operation and orchestration. It is composed of basic sub-networks such as radio access network (AN), Transport network (TN) and Core network (CN), known

[1]https://www.5g-vinni.eu/

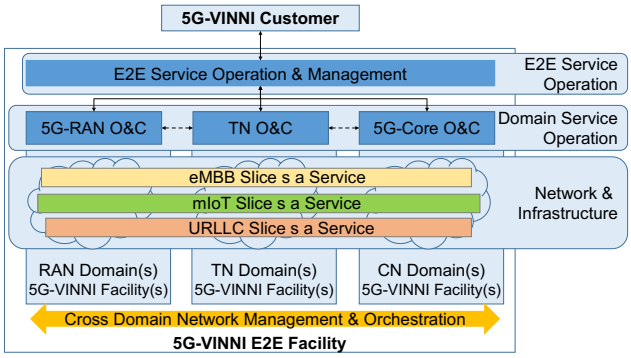

Figure 1. High-Level 5G-VINNI Network Slicing Architecture

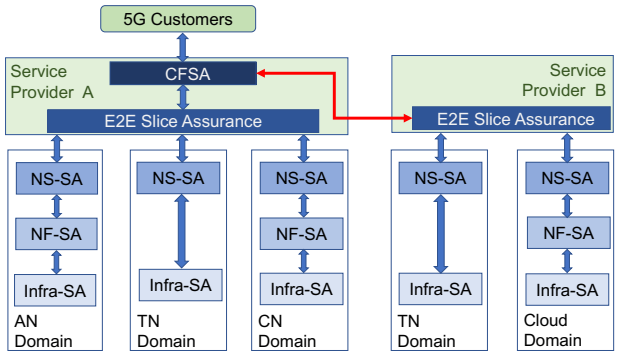

Figure 2. Service Assurance Architecture for Network Slicing

as *domain*. Each domain has its own orchestrator and controller (O&C) to manage the network service (NS) at the application level (independently of their deployment), including but not limited to domain level SLA management, QoS fulfilment, etc. The deployment of the network slice and constituent NSs relies on the underlying network functions (NFs) (e.g., physical NF (PNF) and virtual NF (VNF)) and infrastructure (data center (DC) and TN). The E2E service operation and management requires cross-domain interconnections to coordinate different domain O&C and then produce an E2E view of the entire slice across multiple domains.

To provide user friendly zero-touch orchestration, operations and management systems, 5G-VINNI offers Network Slice as a Service (NSaaS). The current network slicing architecture (*e.g.*, [1]) does not really solve the zero-touch automation issue due to the lack of SA. Furthermore, the hierarchical and multi-domain property of the 5G-VINNI architecture significantly increases the complexity to find an existing SA architecture that supports the objectives in SLA management and QoS fulfillment (even QoE assurance for future 5G users) in 5G network slicing. This paper highlights a SA architecture to automate provisioning of NSaaS within the 5G-VINNI Network Slicing architecture.

III. Service Assurance Architecture

Aligned with the slice architecture proposed by 5G-VINNI for 5G network slicing, a general SA architecture is proposed (Fig. 2). This SA is hierarchical and can be applied to slices spanning across multiple domains and multiple service providers (equivalent to the facility site in 5G-VINNI). It aims to assure the customer-facing service (CFS) offered to the 5G vertical customers.

A. Architecture Description

The *hierarchical* SA architecture consists of five layers (Fig. 2). The bottom three layers, Infrastructure-SA, NF-SA, and NS-SA correspond to the three NFV layers defined in the ETSI MANO framework [9], infrastructure, NF, and NS, respectively. The E2E Slice Assurance (E2E-SA) is responsible for assuring the network slices provisioning for the CFS, whose assurance is achieved by the CFS Assurance (CFSA). This hierarchy reflects how a CFS is constructed recursively from simpler components.

The top layer CFSA interacts with the 5G customers and can be offered by the service provider that receives service request from the 5G customers (*e.g.*, service provider A in Fig. 2). 5G customers usually request communications services rather than network slices. The CFSA translates the customer's service request, *e.g.*, service level agreement (SLA) and/or quality of experience (QoE) requirements, into the SLA suitable for individual slices that could be used by E2E-SA. If a CFS requires network slices provided by multiple service providers (*e.g.*, service provider A and B in Fig. 2), the CFSA decomposes the CFS-SLA into SLAs for each E2E-SA. Furthermore, CFSA receives and aggregates SA-related data from each E2E-SA (the red line from Service Provider B and blue line from Service Provider A) to generate an overall SA view for the CFS and assess if the CFS-SLA is guaranteed.

E2E-SA is responsible for the network slice provided by one administrative provider, *e.g.*, in service provider A or B. The E2E term is used because one slice often spans multiple technology domains, such as AN, CN and TN. Each domain has its own SA and realized by NS-SA in Fig. 2. Similar to CFSA, E2E-SA decomposes the slice SLA into the SLA of each domain, and gathers and aggregates SA-related data from each domain to generate an E2E view of the network slice within the provider's domain. The similar relationship exists between NS-SA and NF-SA, and between NF-SA and Infrastructure-SA.

Apparently, SA is realized in a *distributed* way to allow for flexibility and scalability, *e.g.*, each layer and domain has its own SA, which is referred to as *local SA*. Local SA can evolve independently as some local SA are developed faster than others, *e.g.*, infrastructure-SA and NF-SA is more developed than E2E-SA [7]. The domain can flexibly construct its SA. For instance, the TN (*e.g.*, Software Defined WAN) does not have NFs and thus contains infrastructure-SA and NS-SA (Fig. 2). Distributed SA can separate the layer- and domain-SA issues from the E2E-SA and CFSA issues. Local SA has more detailed and in-depth knowledge of the assured entity and thus could make decisions more quickly and even accurately, especially when edge clouds are deployed in 5G. More importantly, with local SA, the complexity of assuring CFS is significantly reduced. In this way, the abstraction feature inherited from NFV is well maintained such that the changes in one layer or domain do not affect other layers. For example, if the infrastructure layer is switched from virtual-machine(VM)-based to container-based, the corresponding change of orchestration and assurance from

OpenStack-based to Kubernetes-based is agnostic for upper layers. On the other hand, distributed SA may suffer from performance degradation for the network slice and CFS, especially when local SAs operate independently. Therefore *coordination* is demanded. The higher-layer-SA is responsible for coordinating the lower-layer-SAs, *e.g.*, by properly and effectively aggregating SA data from lower-layer-SAs. How coordination is achieved relies on the functional components of each SA layer.

B. Functional Components

Each SA layer contains at least seven functional components, including four basic SA functions and three enhanced SA functions (Fig. 3). Like conventional SA, there are four basic SA functions, *monitoring*, *data collection and storage*, *analytics*, and *reporting*. Three enhanced functions are introduced to support coordination and enable automation: *SA interpretation*, *SA policy management*, and *data fabric*.

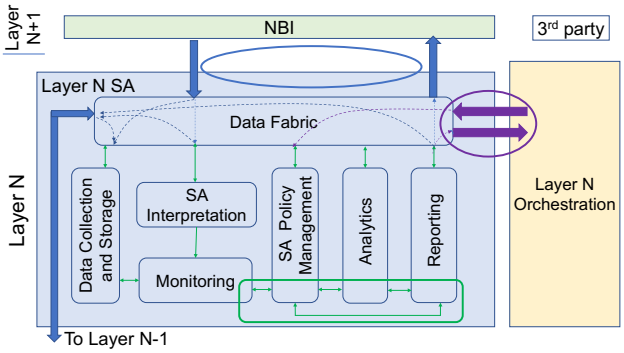

Figure 3. Service Assurance Framework in Layer N

SA interpretation is a translator that takes the higher-layer SA requirements as input and generates two outputs: i) the SA requirements that can be understood by the monitoring function; ii) the SA requirements to the lower-layer-SA. For example, at CFSA, the SA interpretation needs to translate the CFS-SLA into a series of metrics and relevant information used by CFSA monitoring to schedule monitoring tasks. Depending on the SLA content, CFSA runs different tasks to assure reliability, performance, or security. In addition, the CFS-SLA is also broken down into slice-SLA and passed down to E2E-SA, using breakdown mechanism such as the one suggested by the TM Forum in [10]). If the 5G customer imposes QoE requirements, then more advanced mapping algorithms are needed to translate the QoE requirements into the quality of service (QoS) requirements for E2E-SA. The similar procedure runs at E2E-SA, NS-SA and NF-SA except for Infrastructure-SA, which is the bottom layer.

SA policy management manages the policies used in other SA functions, including but not limited to monitoring, analytics, and reporting policies. The monitoring policies decide how to plan and schedule monitoring tasks, such as the task type (pre-deployment validation, performance monitoring, security monitoring, etc.), the monitoring tools (passive or active, etc.), the entities and metrics to be monitored, the frequency to measure, the number and placement of the selected monitoring tools. If the selected monitoring tool is external, *e.g.*, a virtual probe which is a special VNF, then it needs to be attached to the monitored entity. Accordingly, the monitoring policy should be taken into account in the lifecycle of the monitored entity, which is part of orchestration. The analytics policies decide the SA problem needed to solve, select an analytical model to solve the given SA problem, and set up the data set for the analytical model. One critical policy is on *aggregation* that guides what and how data are aggregated and correlated.

Reporting policies are required to classify analytical results and data into two classes and then send to different receivers. The first class corresponds to SA problems that have been solved locally and the root cause has been identified. Then a report is generated and sent to the orchestrator at the same layer. Recommendations may be enclosed to guide the orchestrator to take corrective actions. Meanwhile, another report is sent to the lower-layer-SA, which will investigate if the identified root cause is linked to its own entity. The second class is for SA problems that cannot be solved locally. A report is generated to escalate the problem and sent to the higher-layer-SA for further analysis.

Data fabric provides ways for SA functions to consume data. Both internal SA functional components and external components (higher-layer-SA and lower-layer-SA and orchestrator at the same layer) consume the data generated by this SA via the data fabric. Externally, it allows Data Collection and Storage receive external data such as the SA reports from both the lower-layer-SA and the higher-layer-SA and vice versa. SA Interpretation receives SA requirements from the higher-layer-SA and passes SA requirements to the lower-layer-SA. SA policy management adjusts its policies based on the requests received from the orchestrator at the same layer. Reporting creates reports, which are then sent to other-layer-SA and the orchestrator via the data fabric too. Internally, analytics consumes the stored data through this fabric.

C. Interactions

The proposed SA architecture and functions address the requirements on scalability, coordination, interpretation, and customization. Another vital requirement is automation, which is facilitated through three closed loops.

First, an *internal closed-loop* is created between SA functions (green circle in Fig. 3), centered at the SA Policy Management. Due to highly dynamic network and infrastructure conditions and customer requirements, many policies need to adapt to achieve a good performance. For example, if real-time Analytics results indicate that the required SLA is not met, then troubleshooting is triggered to diagnose the root cause. Monitoring policies will be updated to support the troubleshooting request, *e.g.*, by initiating a new monitoring task or increasing the monitoring frequency. On the other hand, if the service operation behaves well for a sufficiently long period, then the monitoring policy could reduce the monitoring frequency to save resource [7]. This closed-loop aims to automate the policy management.

Second, an *external closed-loop* is formed between SA and the orchestrator at the same layer (purple circle in Fig.

3). SA sends reports to the orchestrator if anomalies are detected and diagnosed. The orchestrator takes corrective actions to resolve the SA problem. In case the corrective actions cause impact on SA, the orchestrator will notify SA. For example, some life cycle management (LCM)-related corrective actions like VNF scaling or migration often result in changes of monitoring policies. Once the SA Policy Management is notified of such changes, it adjusts the monitoring deployment policy and guides Monitoring to scale or migrate the impacted monitoring tools. This closed-loop aims to automate the LCM of the assured entity, which is also the most representative automation considered in the literature.

Third, a *cross-layer closed-loop* is established between SA layers (blue circle in Fig. 3). Adjacent SA layers exchange data and reports regarding the their SA status. Specifically, the lower-layer-SA reports unsolved SA problems to the higher-layer-SA, which plays a role of coordinator and gains a wider view than the lower-layer-SA to solve the problem. In return, the higher-layer-SA sends its analytical results to the lower layer and helps to enhance the detection and diagnosis capability of the lower-layer-SA. This closed-loop aims to automate the coordination between SA layers.

To successfully form the three closed loops, three types of interfaces are demanded: *internal* (green lines), *external with orchestrator* (purple lines) and *external cross-layer* (blue lines), as illustrated in Fig. 3. Data Fabric acts as an access point to simplify the interface design. 3rd party applications (*e.g.*, advanced analytics applications) may also access the SA data and provide advanced insight to SA via the Data Fabric.

IV. Challenges

As SA is under-developed, many challenges exist while implementing SA in practice.

A. Interfaces and Information Modeling

Open interfaces are critical to support the distributive and modular SA architecture with a high coordination. For the external interface between SA and orchestrator, Verizon's framework has defined reference points *Or-Sa, Vnfm-Sa*, and *Vi-Sa* for the bottom three layers NS, NF, and Infrastructure, respectively [11], but without detailed specifications. The interface between E2E slice orchestration and E2E-SA has not been defined. For the external cross-layer interface, in Fig. 3, a single north-bound-interface (NBI) is proposed to expose data between SA layers for the purpose of unifying the SA framework. However, there is no existing standardized interface between SA layers. The internal interfaces have not been studied as most SA vendors tend to supply the entire one-layer-SA in one module, which seems reasonable for lower-layer-SA such as Infrastructure-SA or NF-SA. Nevertheless, higher-layer-SA like E2E-SA and CFSA may be built from multiple functional modules to enable automation, as demonstrated in Fig. 3.

For CSFA with QoE consideration, intention-based knowledge is expected and needs to be represented by the corresponding interface. Proper information models (IMs) are required to capture the intention-based knowledge and define the attributes and supported operations of the assured entities. These models should be vendor-agnostic in order to facilitate the SA architecture with local SA supplied by multiple vendors. Furthermore, to allow for effective coordination between SA layers, the IMs have to reserve all key information but also maintain the abstraction property to hide the complexity and technology-specific details. The development of IMs are to be synchronized with the interfaces.

B. Coordination

Coordination is a key capability of the SA architecture, including cross-domain coordination and cross-layer coordination. The overall SA can be treated as a combinational coordination problem, which represents an unforeseeable challenge. In an extreme case, an alarm at the infrastructure layer may be propagated upwards and infect the top layer CFS, creating alarm explosion, known as *alarm storm*. Well-designed and well-executed coordination is expected to identify potential threats as early as possible and thus avoid alarm storms.

To achieve this, new mechanisms and methods are needed. Considering the amount and diversity of data generated to be coordinated and aggregated, advanced machine learning (ML) and artificial intelligence (AI) algorithms are considered as promising candidates. Correlation analysis is fundamental to aggregate data from different SA layers or domains. Impact analysis could be used to solve the alarm storm problem [12]. However, existing ML and AI algorithms are not specifically designed for the SA problems in network slicing and thus not optimal for coordination. More efforts in ML and AI need to be dedicated to the SA area.

C. Policy management

Policy management is the foundation for automation and the center of all three closed loops in Fig. 3. One main challenge is to develop new policies for SA, such as monitoring policies and classification policies for reporting. Some policies need input from multiple functions. For example, the placement of monitoring tools and monitoring granularity affect the performance of the analytics models, *e.g.*, prediction accuracy. Thus the monitoring policies should be designed taking into consideration of both monitoring cost and the data requirements from analytics models. Furthermore, these policies need to be optimized and adaptive to the highly dynamic network conditions based on the achieved service performance. However, policy update is resource-consuming. Therefore, a trade-off exists between enhancing service performance and reducing resource consumption. In addition, policy management is often implemented in a distributed way into individual domains and layers. Then coordinating policy management across domains and layers is critical for optimization.

D. Monitoring

The SA framework demands continuous monitoring of various entities and data flows, from infrastructure to CFS. Current monitoring in legacy networks does not meet the

demand. First, it generates coarse grained counters that are collected at particular entities without paying attention to the E2E performance. In order to save cost, detailed E2E monitoring is only initiated reactively upon the reception of customers' complaints. This monitoring strategy obviously cannot support the SA automation, which requires real-time monitoring. However, due to the high cost of monitoring every entity and event, fine-grained monitoring remains a challenge in practice to balance between the monitoring performance and the potential resource consumption. Second, current monitoring depends on static monitoring policies and lacks the ability to adapt to the dynamics of network conditions, a typical property of network slicing. Third, network slices have diverse service requirements and thus anticipate versatile monitoring solutions, *e.g.*, the monitoring tools and monitoring policies. A unified SA monitoring is expected to be flexible and supportive of monitoring multiple slices simultaneously.

At present, there is no agreement on how to handle monitoring in the early deployment of 5G. Some projects have indicated a general direction towards integrating monitoring into the MANO framework, *e.g.*, the two key MANO platforms ONAP[2] and OSM[3]. ONAP designs a specialized subsystem Data Collection, Analytics, and Events (DCAE) to collect measurement data at various NFs and report them to analytics engines. OSM recently added a monitoring module OSM MON to collect monitoring information at the infrastructure and VNF layer. However, none of the existing solutions tackles all the aforementioned challenges.

E. Isolation

Isolation is a capital property of network slicing, and also vital for SA. There are different mechanisms to implement it. Techniques such as time and frequency splits and traffic prioritization can be used in RAN and Transport. In the virtualized environment, isolation can be done at several levels, e.g., datacenter, zones, physical host, virtual machines or dockers. Based on this resource sharing will exist at some point, and potential interference will exists, which significantly complicates SA. For example, if two CFSs share part of one slice, *e.g.*, a VNF, then the corresponding CFSA expects NF-SA to distinguish packet flows from these two CFSs, which results in higher monitoring cost and analytics overhead. Moreover, if network slice A is not perfectly isolated from slice B, then A may causes direct or indirect influence on B. When slice B experiences a SA problem, its E2E-SA needs to identify whether the problem is caused by its own components or by slice A, which is extremely challenging. Assuring isolation is the prerequisite for assuring services. However, isolation is a complex topic still under development.

F. Quality of Experience (QoE)

QoE assurance is a new trend in 5G. Assuring QoE is complicated for various verticals. First, most existing QoE models are applied to mobile broadband (MBB) services like voice, video, or gaming. The QoE models of many industrial verticals (*e.g.*, IoT and autonomous vehicles) are still undefined, *e.g.*, what and how network QoS and infrastructure performance contribute to the QoE of verticals customers. Second, most existing QoE models are built from a single end user's perspective, with consideration of subjective perception whereas CFS is facing vertical customers, each of which consists of a group of end users. Additional models are required to link the end-user-QoE with the vertical-customer-QoE, such as aggregation models. Third, new mapping models are needed to translate vertical-customer-QoE into SLA of individual slices if one CFS is provisioned by multiple network slices. The study of these topics has not been matured in the QoE society.

[2]https://www.onap.org/

[3]https://osm.etsi.org/

V. Conclusions

In this paper, we propose a SA architecture to address the main requirements raised up by network slicing features in 5G. The SA architecture is designed in a way to support automatic, scalable, flexible, and customized 5G service provisioning. Key elements of the architecture are discussed, including functional components and interfaces. It is recognized that there are many challenges in realizing the proposed SA, such as interfaces and information models, policy management, coordination, monitoring and isolation. An even bigger challenge is that these challenges are intertangled with each other, *e.g.*, policy management implementation also involves cross-layer and cross-domain coordination. In future, we consider to develop pragmatic approaches to test and validate our SA architecture by leveraging the 5G-VINNI E2E facility.

Acknowledgment

This work has been supported by the European Community through the 5G-VINNI project (grant no. 815279) within the H2020-ICT-17-2017 research and innovation program.

References

[1] 3GPP, "TR28.801: Study on management and orchestration of network slicing for next generation network (Rel 14)," 2017.

[2] ETSI ZSM, "ZSM 002 Draft: Zero-touch Network and Service Management (ZSM); Reference Architecture," Jan 2019.

[3] 3GPP, "TS 28.550 Management and orchestration; Performance assurance (Release 15," Dec 2018.

[4] ——, "TR 23.791, v16.0.0: Study of Enablers for Network Automation for 5G (Release 16)," Dec 2018.

[5] TM Forum, "IG1127: End-to-end Virtualization Management: Impact on E2E Service Assurance and SLA Management for Hybrid Networks," Apr 2015.

[6] M. Xie, C. Banino-Rokkones, P. Grønsund, and A. J. Gonzalez, "Service assurance architecture in NFV," in *2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Nov 2017.

[7] M. Xie, Q. Zhang, P. R. Grønsund, P. Palacharla, and A. J. Gonzalez, "Joint Monitoring and Analytics for Service Assurance of Network Slicing," in *2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, 2018.

[8] 5GVINNI, "D1.2 Design of network slicing and supporting subsystems v1," https://www.5g-vinni.eu/.

[9] ETSI, "ETSI GS NFV-MAN 001: Network Functions Virtualisation; Management and Orchestration," Dec 2014.

[10] TM Forum, "Frameworx Best Practice: Enabling End-to-End Cloud SLA Management," Oct 2014.

[11] Verizon, "SDN-NFV Reference Architecture, V1.0," February 2016.

[12] S. Papadimitriou, J. Sun, and P. S. Yu, "Local Correlation Tracking in Time Series," in *International Conference on Data Mining (ICDM'06)*, 2006.