Towards Edge-Cloud-Supported Monitoring at Cloud-Network Slice Granularity

Kevin B. Costa[1,2], Felipe S. Dantas Silva[1,2], Augusto V. Neto[1,3], and Fábio L. Verdi[4]

[1]Federal University of Rio Grande do Norte (UFRN), Natal/RN, Brazil
[2]LaTARC Research Lab, Federal Institute of Rio Grande do Norte (IFRN), Natal/RN, Brazil
[3]Instituto de Telecomunicações, Portugal
[4]Federal University of São Carlos (UFSCar), Sorocaba/SP, Brazil

***Abstract*—With the recent advances in the new network technologies and paradigms such as 5G, Cloud, Edge Computing, and IoT (Internet of Things), a set of management and resource optimization techniques are demanded. Recently, the slicing approach was employed as part of the 5G technology to maximize infrastructures capabilities by enabling softwarization and virtualization to offer network service and resource abstraction over several different physical networks, thus improving end-to-end service delivery in a flexible and customized manner. The accomplishment of the end-to-end slicing management procedures demands a set of data regarding elements that constitute the diverse entities involved in the communication (i.e., cloud, edge, applications, services, etc.). The fulfillment of this premise is achieved by employing the appropriate monitoring mechanisms in order to make the system intelligence aware of all information regarding the involved entities. This work introduces a monitoring architecture for deploying personalized monitoring schemes to fit different scenarios. The solution here presented allows to take management decisions in the core cloud, in the edge, or even establish an Edge-Cloud interplay model. We designed, implemented and deployed a proof-of-concept together with a set of evaluations, providing insights about the impact of having local versus remote monitoring over the end-to-end slice.**

***Index Terms*—5G, Cloud, MEC, RAN, Cloud-Network Slicing, Monitoring, Mobility.**

I. Introduction

The consumption of multimedia services by customers, which demand high quality, is growing every day, as suggested by [1]. In addition, according to the authors, a fundamental change is taking place in the way we manage networks that address issues related to abstraction, separation and routing mapping as well as control and service management aspects. In this regard, it is important to note that industry and academia are adopting network technologies that can support applications for future generations and with different service requirements [2].

Technologies and paradigms that are being adopted, both by industry and academia, namely 5G, Cloud Computing, Edge Computing, and network-slicing, emerge as a mean to subsidize limitations and needs regarding information processing [3], [4]. Thus, it is essential to chase solutions to problems that are related to such technologies. As [5] states, slicing allows network operators to provide differentiated services to customers, dynamically allocating dedicated parts of their networks. The concept of network slicing was mentioned a few years ago in the structure of 5G networks, as it aimed to offer enriched services that surpassed those of standard connectivity. Also, the motivation behind this context is the need to more accurately suit the requirements of vertical markets, namely entertainment, health, automotive, among others [5], [6].

It is also important to highlight the importance of Multi-access Edge Computing (MEC), as it is a distributed computing paradigm offering storage and computational resources that are processed directly at the edges of the network [7]. As [8], [9] suggest, this technology produces ultra-low latency, high bandwidth and real-time access to network resources. Moreover, MEC architectures raise the 5G and cloud capabilities within the Radio Access Network (RAN) closest to the end-users allowing faster content and service delivery, enhancing responsiveness from the edge through techniques like data offloading [10].

Recently, virtualization and softwarization techniques have leveraged 5G scenarios by enabling infrastructures to deploy services for computing, storage, and networking in a personalized and elastic way by extending the Cloud, Edge, and RAN capabilities through the resources provided by the network-slicing [11]. In addition, the joint adoption of virtualization, softwarization and cloudification [12] can maximize the delivery of end-to-end services that operate under multiple domains belonging to the same 5G cloud and network federation, thus enabling a Cloud-Network Slicing (CNS) scenario [13], [14]. Although domains are geographically distributed in a CNS-enabled infrastructure, they can be seen as a single structure capable of providing services as an expanded domain of resources.

Each CNS Instance (CNSI) is represented by a set of network and cloud resources, spread over several domains. CNSIs are enabled to meet a specific service of the distinct verticals and business models involved. It implies that the 5G infrastructure must also be able, autonomously, to optimize and manage communication and cloud resources, including the various CNS instances deployed throughout the federation [15].

A. Problem Statement

In a cloud-network, the resources offered to provide a particular service may belong to the various participating resource providers. While this architectural arrangement is highly relevant to the offering of ultra-high-definition services, given the advantages already introduced previously, network management procedures such as mobility control will require cooperation between the elements for orchestration and management of the infrastructure resources and services. It involves monitoring the entities (links, CNSIs, Base Stations, RAN, cloud), Key Performance Indicators (e.g., QoS/QoE), and network resources to fulfill the decisions made (e.g., handover decision) [16], [17].

In this sense, it is expected that future telco-cloud [18] scenarios will require the coexistence of different monitoring modes according to the tenant's interest. In these scenarios, characterized by the numerous demands imposed by the multi-tenancy concept, traditional monitoring schemes based only on (a) edge or (b) cloud premises need to evolve their operation to an edge-cloud interplay. Thus, such an approach will support cloud-network slicing operators to maximize third-party infrastructure for using their customized definitions.

The problem that this research addresses is within the context of slicing and its infrastructure monitoring. This demand, which is already a concern of the scientific community, has been discussed in previous works [19], [20], [21] as a relevant research challenge that is still open. This work is also in line with the NECOS Project [citar IEEE Commag aqui] that designed and implemented a Slice as a Service architecture including aspects of management, monitoring and orchestration. In this work, we advance the state-of-the-art in cloud-network slicing by proposing the EDgE ClouD SliCE-part monitoring (EDCS) approach, which harnesses edge-cloud interplay operating in a dynamic and distributed manner to monitor slice instances. The EDCS proposed approach makes it possible for either multiple cloud-network slice operators or automated network management mechanisms to operate their monitoring scheme over common sliced-defined resources.

B. Contributions

The main research contributions of this study are as follows:

(a) An end-to-end monitoring system for cloud-network slicing services;
(b) A slicing monitoring-as-a-service driven architecture;
(c) A full complexity abstraction of third-party monitoring mechanisms for network management applications.

C. Paper Organization

This paper is structured as follows: Section II outlines the most significant works in network-slicing monitoring. Section III introduces the proposed solution. Section IV describes the use case, the methodology, and setup of the experiments and examines the results. Section V summarizes the findings of our research and provides some recommendations for future research.

II. Related Work

This section provides an analysis and discussion on various research related to slicing monitoring. Also, it seeks to draw a comparison between the works to find a way to build the application proposed in this article.

Within this context, [22] presents an information-centric approach to deal with the monitoring of multiple sources in an IoT network. Furthermore, the authors suggest a model designed to provide generic and extensible data formats for different IoT objects. Therefore, the contributions of this work are: (i) propose a new model centered on information and its implementation architecture for the IoT slice monitoring problem; (ii) offer a solution to collect data from multiple sources, from edge devices to clouds at the same time; and (iii) propose a solution to several different types of data to allow the creation of API queries for different IoT contexts.

The work presented in [23] features a cloud-network slicing monitoring system with the capability of detecting the creation of new Cloud-Network slice instances. When a new instance appears, the system prepares in advance the entire monitoring configuration process for each one. At the end of this process, the tenant is responsible for indicating the desired monitoring KPIs. The authors highlight three essential concepts, namely: (i) automatic and dynamic assignment of tasks to monitor servers in message-queuing mode; (ii) load balancing between monitoring servers in the selection of tasks related to configuration; and (iii) exploring the parallelism in the construction of monitoring slices.

The authors of [24] propose an elastic monitoring architecture to collect monitoring KPIs related to physic and virtual infrastructur components. According to the authors, the main characteristic is that the solution is "enabled for elasticity", which means being able to trigger adaptations of dynamic monitoring components due to elasticity actions. In this way, slices or parts of slices are removed and updated according to elasticity-related events. Therefore, resource monitoring is performed both physically and virtually on the cloud network.

The literature review reveals that the current efforts for delivering slicing monitoring are concerned with operating in the cloud premises. When solutions collect data from the edge, the data is sent to the core Cloud for further classification and organization. Only after that such data is delivered to the respective consumer applications. Thus, it is evident the lack of a solution capable of enabling 5G slicing-defined infrastructures with a slicing monitoring Edge-Cloud Interplay approach.

III. Towards the EDCS Solution

The EDCS proposed solution's architecture aims to tackle the current gap by allowing a distributed slice monitoring strategy that can (i) decouple the slice management application plane from the KPI monitoring plane (e.g., Netdata, Prometheus, among other technologies), creating a unified API in a per network-slice-part granularity; (ii) enable on-demand monitoring KPIs delivery comply with multiple network control application needs; and, (iii) offer specialized monitoring schemes, providing local, remote or hybrid (edge-cloud interplay) monitoring in a per network-slice-part granularity.

The EDCS architecture consists of an orchestrator to deploy and setup isolated modules prepared to monitor targeting resources (indicated by Tenants) of each slice-part of a single or multiple slices instances. The modules are responsible for collecting, filtering, and delivering the list of monitoring KPIs that the management applications subscribed without to bound these applications to specific technologies of the dense network ecosystems. In this scope, the EDCS proposal supports three monitoring modes:

- **Cloud-Edge Interplay Monitoring:** it consists of locally monitoring each slice-part of a given network slice (as long as the hardware resource where the slice-part is deployed has the computational architecture required to run the monitoring module on it). This scheme enables the monitoring of KPIs for management applications on the edge premises.
- **Full-Cloud Monitoring:** this scheme enables the monitoring of all slice-parts of a given cloud-network slice to be executed exclusively from the Cloud.
- **Full-Edge Monitoring:** this scheme enables monitoring all slice-parts of a given cloud-network slice to be executed exclusively from the network's edge.

The next subsections introduce the EDCS proposal architecture, its main components and interfaces.

A. *System Architecture*

The EDCS proposed monitoring system's architecture is composed of three main subsystems: (i) the Monitoring Orchestrator; (ii) the Message Broker; and (iii) the Slice-Part Adaptor (SP Adaptor). Figure 1 depicts the overall system and the relationship between the components.

1) Monitoring Orchestrator: As shown in Figure 1, the central entity of the EDCS monitoring system architecture comprehends the Monitoring Orchestrator, which is composed of the API, the Adaptor Builder, and the KPI Gathering.

The API component is responsible for handling requisitions supporting the three primary monitoring operations: (i) creating the monitoring for a cloud-network slice instances; (ii) updating the current monitoring scheme of a slice-part instance (for situations where a slice is modified and the monitoring scheme needs to be updated); and, (iii) removing the monitoring for a specific slice.

Through the API, the Slice Orchestrator System (following the approach adopted in the NECOS project [25]) delivers the necessary information that will drive the lifecycle of the system through the Adaptor Builder. An API component request example is provided in the Listing 1.

The Adaptor Builder is responsible for fetching the information about the slice-parts from the requisitions delivered to the API (from this point on, it will deploy Slice-Part Adaptors according to each slice-part specified in the request).

The configuration of the Slice-Part Adaptor consists of a requisition containing: (i) list of monitoring KPIs intending to be monitored; (ii) monitoring Agent of the slice-part (e.g., Netdata, Prometheus, Openflow, etc.); (iii) monitoring frequency of each monitoring KPI specified; and, (iv) the IP address/port of the application that will retrieve the monitoring KPIs and the Broker address.

Listing 1: Monitoring create request example

```
{
   'slice_id': 1,
'slice_name': 'Default-slices',
'slice_parts': {
  'pcpe': [{
    'slice_part_name': 'wifi-slice',
    'slice_part_id': 5,
    'type': 'NET',
    'pcpe_address': '10.7.227.130',
    'monitoring_configuration': {
      'KPIs': {'Wifi Network Quality': '1 second', 'Wifi Network Bitrate': '1 second'},
      'monitoring_agent': 'node-exporter',
      'monitoring_location': [{'remote': '10.7.227.175'},{'Access credentials': ['admin':'admin']}],
      'delivery_location': [{'monitoring_delivery_place': 'local'},{'address': 'localhost:7564'}]
    }
  }],
  'edge': [{
    'slice_part_name': 'edge-dc-default',
    'slice_part_id': 2,
    'type': 'EDGE',
    'edge_address': '10.7.227.175',
    'monitoring_configuration': {
      'KPIs': {'Network Bandwidth': '1 second'},
      'monitoring_agent': 'Netdata',
      'monitoring_location': [{'remote': '10.7.227.175'},{'Access credentials': ['admin':'admin']}],
      'delivery_configuration': [{'monitoring_delivery_place': 'local'},{'address': 'localhost:8775'}]
    }
  }],
  'dc': [{
    'slice_part_name': 'core-dc-default',
    'slice_part_id': 1,
    'type': 'DC',
    'dc_address': '10.7.229.191',
    'monitoring_configuration': {
      'KPIs': {'Memory Available': '1 second', 'Network Upload': '1 second'},
      'monitoring_agent': 'prometheus',
      'monitoring_location': [{'remote': '10.7.229.191'},{'Access credentials': ['admin':'admin']}],
      'delivery_configuration': [{'monitoring_delivery_place': 'local'},{'address': 'localhost:9740'}]}}]}}
```

The KPI Gathering component is responsible for gathering monitoring KPIs from the Slice-Part Adaptors (received

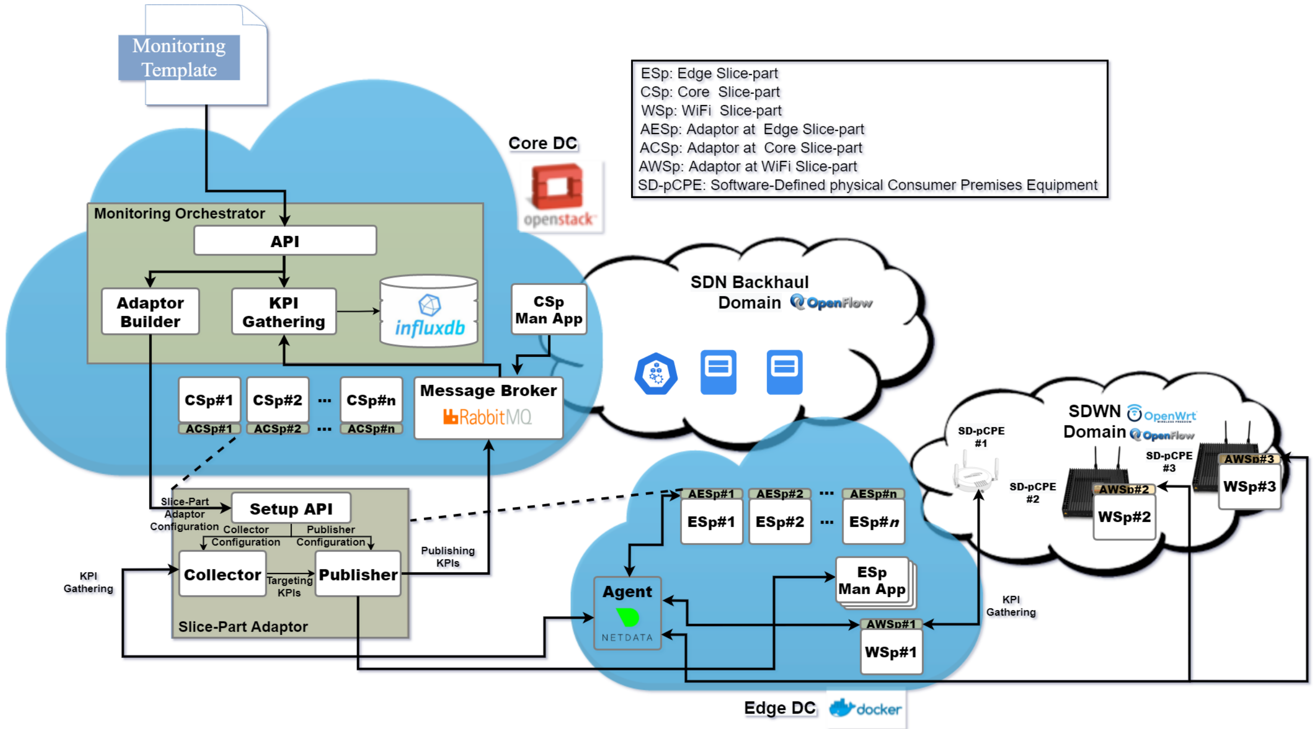

Figure 1: Proposed Architecture.

through the Broker) to the database. The organization of the monitoring KPIs stored is done by employing headers in its data (added inside the adaptors over the network when it collects the monitoring KPIs), namely: (i) Slice Identification Header (specified by the Slice Orchestrator); (ii) Slice-Part Identification Header (specified by the Slice Orchestrator); and, (iii) collection time (which contains a timestamp representing the instant of time that the monitoring KPI is fetched).

2) Message Broker: The Message Broker employs a publish-subscribe communication service between the Monitoring Orchestrator and all active Slice-Part Adaptors. The Message Broker organizes all slices in the network that are being monitored with the use of queues. Each slice-part is mapped to a queue, transporting all the monitoring KPIs related to all slice-parts from a specific slice (the slice-part Identification Header mentioned above will differentiate monitoring KPIs from different slice-parts in a queue). This approach allows temporal and spatial decoupling between these components.

3) Slice-Part Adaptor: The Slice-Part Adaptor is the component that will interact directly with the resources of the slice-part. Thus, it needs to be aware of the specific technologies used in the slice-part where it will be deployed. This information is given by the Slice Orchestrator through the Monitoring Orchestrator's API and then delivered to the Slice-Part Adaptor through the Setup API (in the instant that the Adaptor Builder configures it).

When the Adaptor Builder starts the configuration step, the Setup API will receive the necessary information and configure the Collector and the Publisher. The Collector is the component that will switch (according to the Setup API information) between the specific algorithms to access the APIs of the slice-part specific monitoring agent.

The Publisher is configured to deliver a list of monitoring KPI (after collecting and filtering procedures) to the specific local Management Applications, if there's any, or send it through the Message Broker to be delivered to a subscribed remote application and received and stored by the KPI Gathering.

IV. Testbed Evaluation

To validate the EDCS solution's architecture, a proof-of-concept was conducted atop a testbed embedding real-world devices and enabling technologies. The EDCS architecture implementation was done in Python programming language (version 3.6.9) and executed in a testbed consisting of an Edge server (Dual Core AMD Athlon(tm) II X2 B28, 16GB of RAM), a Cloud server (8 vCPUs and 8GB of RAM), and an off-the-shelf Wifi SD-pCPE (TP-LINK TL-WR1043ND v2, which is powered by Qualcomm Atheros QCA9558 @ 720 MHz chipset, 64 MB RAM and 8 MB flash, and running OpenWrt SO and OpenFlow v3).

Different scenarios were set to showcase the perspectives of using each of the three monitoring modes that our EDCS solution supports. To achieve this, the time-consuming and control signaling throughput measures are analyzed. In the the time-consuming, we collected the instant time that a deployed Slice-Part Adaptor takes (i) to gather a list of monitoring KPIs that the Adaptor Builder indicated, through the monitoring agent API (NetData); (ii) to process and filter the list of monitoring KPIs; and, (iii) to deliver the gathering monitoring KPIs to the specified destination, according with the scenario. The specifications of each scenario are bellow:

1) Full Cloud: the Adaptor Builder deploys all adaptor instances in slice-parts that provisioned at the Core Cloud DC (Cloud slice-part – CSp), so that to enable on-site monitoring. Edge slice-part (ESp) instances, as well as WiFi slice-part (WSp) instances, are subjected to remote monitoring (i.e., CSp-running adapators gather monitoring KPIs by invoking the edge-running monitoring agent API remotely). Finally, a Slice-Part Management Application is containerized in the Core Cloud DC, being the destination of a subscribing set of monitoring KPIs in a per Slice-Part Adaptor instance granularity.
2) Full Edge: the Adaptor Builder deploys all adaptor instances in the Edge DC, so that enabling on-site monitoring and management in a per-ESp instance granularity. A Slice-Part Management Application is containerized in the Edge DC, being the destination of local-measured KPI in a per Slice-Part Adaptor instance granularity. In this scenario, the WSp instance is provisioned at the Edge DC, since the SD-pCPE off-the-shelf resource patterns are not sufficient to support on-site WiFi slicing.
3) Cloud-Edge Interplay: in the Cloud-Edge Interplay model, ESp and CSp instances provide a distributed monitoring scheme. The workflow for the Cloud-Edge Interplay model defines that ESp monitoring instances (including WSp instances) carry out a first slice-part measurement data analysis, and then delivers monitoring KPI all the way to a CSp for executing, for instance, a second round data analysis. The evaluation time in this scenario comprehends the total time that each ESp-CSP coupled scheme takes between gathering targeting monitoring KPIs until delivering to the cloud management application. The purpose of this evaluation is to measure the impact of the distributed monitoring;

Each evaluated monitoring scenario is set to the same template definitions, consisting of two monitoring KPIs for the CSp, one for the ESp, and two for the WSp. Listing 1 shows the description template of a requisition made by the Slice Orchestrator for the Monitoring Orchestrator. The first analysis is the KPI delivery time in a per monitoring scheme granularity, shown in Figure 2 sketches.

As can be seen in Figure 2 and 3, the Full Edge monitoring mode raises the lowest delivery time rate and monitoring KPI network cost regarding the monitoring KPI. As expected, the Full Edge monitoring mode performs better since the monitoring agent runs aside the monitoring-approached components (i.e., AESp and WSp instances, and corresponding ESp management applications), without any Cloud DC intervention. The results show promising perspectives to afford management

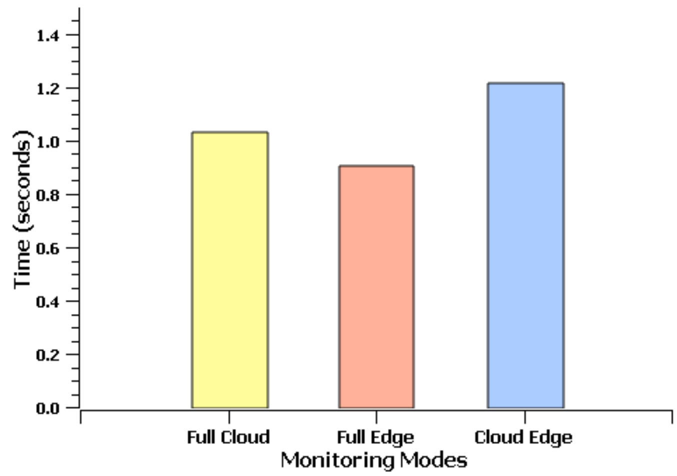

Figure 2: Delivery Time rate for each monitoring mode.

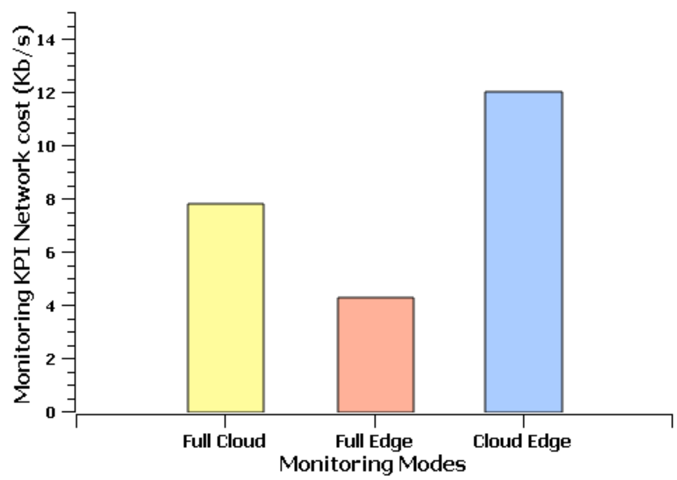

Figure 3: Monitoring KPI Network cost for each monitoring mode.

applications that need to take agile decisions, while enabling a scalable approach. On the other hand, the Edge-Cloud Interplay scenario adds an extra ESp-to-CSP signaling trip time, which results in the higher time rate of the experiments.

The Full-Cloud scenario raises a slight time increase with regards to the Full-Edge experiment, since the measurement workflow (gathering slice-part monitoring KPI, processing all them, and then delivering to the targeting cloud management application) is done entirely at the Core Cloud DC premises. However, this scheme counts with the Message Broker intervention, which adds latency to the monitoring KPI gathering approach along with the additional networking latency to cross the backhaul infrastructure.

According to the outcomes of our testbed experiments, the concepts behind the EDCS monitoring approach have proven to be functional, and supports the three monitoring modes that this paper proposes. Although not subjected to studies in our set of experiments, it is well-known that management cost exponentially increases according with the network density and complexity. Therefore, our testbed experiments suggest that on-site monitoring and decision functions impact the performance of management applications.

V. Concluding Remarks and Future Work

In this work, we introduce a new system architecture tailored to the monitoring of end-to-end cloud-network slices, which we denote to EDCS. The EDCS proposed system harnesses the Edge-Cloud continuum, for enabling three different monitoring modes to be provisioned: Full-Edge, Full-Cloud, and Edge-Cloud Interplay. Thus, EDCS enables slice-part instances to be monitored either in local, remote or distributed premises. To achieve this, the design of the EDCS architecture follows a modular approach of components that interwork through internal interfaces, and exposes functions to outside applications through external interfaces.

As future work, we will integrate the EDCS approach into the NECOS platform. Afterwards, we will assess the performance of the EDCS-enabled NECOS platform so as to showcase the impact regarding the increased density of slice instances and services settings.

VI. Acknowledgment

This research was partially supported by the H2020 4th EU-BR Collaborative Call, under the grant agreement no. 777067 (NECOS–Novel Enablers for Cloud Slicing), funded by the European Commission and the Brazilian Ministry of Science, Technology, Innovation, and Communication (MCTIC) through RNP and CTIC.

References

[1] A. A. Barakabitze, A. Ahmad, R. Mijumbi, and A. Hines, "5g network slicing using sdn and nfv: A survey of taxonomy, architectures and future challenges," Computer Networks, vol. 167, p. 106984, 2020.

[2] F. S. Dantas Silva, E. Silva, E. P. Neto, M. Lemos, A. J. Venancio Neto, and F. Esposito, "A taxonomy of ddos attack mitigation approaches featured by sdn technologies in iot scenarios," Sensors, vol. 20, no. 11, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/11/3078

[3] A. Gupta and R. K. Jha, "A survey of 5g network: Architecture and emerging technologies," IEEE Access, vol. 3, pp. 1206–1232, 2015.

[4] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions," IEEE Communications Surveys Tutorials, vol. 20, no. 3, pp. 2429–2453, 2018.

[5] V. Q. Rodriguez, F. Guillemin, and A. Boubendir, "5g e2e network slicing management with onap," in 2020 23rd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN). IEEE, 2020, pp. 87–94.

[6] R. Casado-Vara, A. Martin-del Rey, S. Affes, J. Prieto, and J. M. Corchado, "Iot network slicing on virtual layers of homogeneous data for improved algorithm operation in smart buildings," Future Generation Computer Systems, vol. 102, pp. 965–977, 2020.

[7] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," IEEE internet of things journal, vol. 3, no. 5, pp. 637–646, 2016.

[8] F. Ramalho and A. Venâncio Neto, "Virtualization at the network edge: A performance comparison," in 2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM). IEEE, 2016, pp. 1–6.

[9] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," Journal of Systems Architecture, vol. 98, pp. 289–330, 2019.

[10] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5g wireless networks: A comprehensive survey," IEEE Communications Surveys Tutorials, vol. 18, no. 3, pp. 1617–1655, thirdquarter 2016.

[11] F. S. Dantas Silva, E. Neto, C. Santos, T. Almeida, I. Silva, and A. V. Neto, "Evolving fast innovation in next-generation networking through flexible and customized softwarization and slicing capabilities," in 2020 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), 2020, pp. 188–193.

[12] D.-H. Luong, H.-T. Thieu, A. Outtagarts, and Y. Ghamri-Doudane, "Cloudification and autoscaling orchestration for container-based mobile networks toward 5g: Experimentation, challenges and perspectives," in 2018 IEEE 87th Vehicular Technology Conference (VTC Spring), 2018, pp. 1–7.

[13] E. P. Neto, F. S. D. Silva, L. M. Schneider, A. V. Neto, and R. Immich, "Seamless mano of multi-vendor sdn controllers across federated multi-domains," Computer Networks, vol. 186, p. 107752, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128620313311

[14] M. Carmo, F. S. D. Silva, A. Neto, D. Corujo, and R. Aguiar, "Network-cloud slicing definitions for wifi sharing systems to enhance 5g ultra dense network capabilities," Wireless Communications and Mobile Computing, vol. 2019, p. 17, Feb 2019.

[15] D. M. Gutierrez-Estevez, M. Gramaglia, A. D. Domenico, G. Dandachi, S. Khatibi, D. Tsolkas, I. Balan, A. Garcia-Saavedra, U. Elzur, and Y. Wang, "Artificial intelligence for elastic management and orchestration of 5g networks," IEEE Wireless Communications, vol. 26, no. 5, pp. 134–141, 2019.

[16] Y.-J. Liu, G. Feng, Y. Sun, S. Qin, and Y.-C. Liang, "Device association for ran slicing based on hybrid federated deep reinforcement learning," IEEE Transactions on Vehicular Technology, vol. 69, no. 12, pp. 15 731–15 745, 2020.

[17] C. H. F. dos Santos, M. P. S. de Lima, F. S. Dantas Silva, and A. Neto, "Performance evaluation of multiple attribute mobility decision models: A qoe-efficiency perspective," in 2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), 2017, pp. 159–166.

[18] G.-P. S. N. W. Group, "From webscale to telco, the cloud native journey," 5G PPP, Tech. Rep., 07 2018.

[19] M. Steinke, I. Adam, and W. Hommel, "Multi-tenancy-capable correlation of security events in 5g networks," in 2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), 2018, pp. 1–6.

[20] M. A. Faruque, "A review study on "5g nr slicing enhancing iot amp; smart grid communication"," in 2021 12th International Renewable Engineering Conference (IREC), 2021, pp. 1–4.

[21] P. Valsamas, P. Papadimitriou, I. Sakellariou, S. Petridou, L. Mamatas, S. Clayman, F. Tusa, and A. Galis, "Multi-pop network slice deployment: A feasibility study," in 2019 IEEE 8th International Conference on Cloud Networking (CloudNet), 2019, pp. 1–6.

[22] B. M. Nguyen, H. Phan, D. Q. Ha, and G. Nguyen, "An information-centric approach for slice monitoring from edge devices to clouds," Procedia computer science, vol. 130, pp. 326–335, 2018.

[23] M. B. de Carvalho, R. P. Esteves, G. da Cunha Rodrigues, C. C. Marquezan, L. Z. Granville, and L. M. R. Tarouco, "Efficient configuration of monitoring slices for cloud platform administrators," in 2014 IEEE Symposium on Computers and Communications (ISCC). IEEE, 2014, pp. 1–7.

[24] A. Beltrami, P. D. Maciel, F. Tusa, C. Cesila, C. Rothenberg, R. Pasquini, and F. L. Verdi, "Design and implementation of an elastic monitoring architecture for cloud network slices," in NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium. IEEE, 2020, pp. 1–7.

[25] F. S. D. Silva, M. O. Lemos, A. Medeiros, A. V. Neto, R. Pasquini, D. Moura, C. Rothenberg, L. Mamatas, S. L. Correa, K. V. Cardoso et al., "Necos project: Towards lightweight slicing of cloud federated infrastructures," in 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft). IEEE, 2018, pp. 406–414.