



HAL
open science

A Novel Architectural Approach for the Provision of Scalable and Automated Network Slice Management, in 5G and Beyond

Slawomir Kukliński, Lechoslaw Tomaszewski, Ioannis P. Chochliouros, Christos Verikoukis, Robert Kolakowski, Anastasia S. Spiliopoulou, Alexandros Kostopoulos

► To cite this version:

Slawomir Kukliński, Lechoslaw Tomaszewski, Ioannis P. Chochliouros, Christos Verikoukis, Robert Kolakowski, et al.. A Novel Architectural Approach for the Provision of Scalable and Automated Network Slice Management, in 5G and Beyond. 17th IFIP International Conference on Artificial Intelligence Applications and Innovations (AIAI), Jun 2021, Hersonissos, Crete, Greece. pp.39-51, 10.1007/978-3-030-79157-5_4 . hal-03789012

HAL Id: hal-03789012

<https://inria.hal.science/hal-03789012v1>

Submitted on 27 Sep 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

A Novel Architectural Approach for the Provision of Scalable and Automated Network Slice Management, in 5G and Beyond

Sławomir Kukliński^{1,2}, Lechosław Tomaszewski¹, Ioannis P. Chochliouros^{3*} [0000-0002-4208-1676], Christos Verikoukis⁴, Robert Kołakowski^{1,2}, Anastasia S. Spiliopoulou³ and Alexandros Kostopoulos³

¹ Orange Polska, ul. Obrzeźna 7, 02-691 Warsaw, Poland

² Warsaw University of Technology, pl. Politechniki 1, 00-661 Warsaw, Poland

^{3*} Hellenic Telecommunications Organization (OTE) S.A.,
99 Kifissias Avenue, 15124 Maroussi-Athens, Greece
ichochochliouros@oteresearch.gr

⁴ Centre Tecnologic de Telecomunicacions de Catalunya (CTTC),
Avinguda Carl Friedrich Gauss 7, Castelldefels Barcelona 08860, Spain

Abstract. The paper discusses a novel architecture proposed within the scope of the ongoing MonB5G EU-funded project. Considering a multiplicity of challenges towards realizing an effective network slice management in modern 5G networks, our work considers explicitly the context promoted by the Zero-touch network and Service Management (ZSM) framework, assessed as an indispensable part of next-generation management systems. MonB5G focuses upon the provision of scalable and automated management and orchestration of high numbers of parallel network slices, as envisioned in 5G and beyond. Within this scope, we propose a detailed architecture composed of static and dynamically deployed components. Altogether, they support operations related to slicing orchestration, fault management (self-healing), self-configuration, performance optimization (including energy saving), and security-related operations of slices. In the paper, we identified each separate architectural layer and explained all involved modules and interfaces. The proposed framework is able to support the deployment of a massive number of slices in different administrative and technological domains. Furthermore, the potential extensions and/or enhancements of the architecture are also proposed and assessed.

Keywords: 5G, Artificial Intelligence (AI), Machine Learning (ML), MANO, network management, network slicing, Orchestration, Virtual Network Function (VNF), Zero-touch network and Service Management (ZSM).

1 Introduction

As 5G technology is growing and very rapidly expanding in a great variety of applications and/or related sectors [1-3], this radically transforms the underlying communications market and creates a multiplicity of advantages for the participating actors [4] in view of new pervasive mobile services of different vertical industries.

5G introduces the use of virtualization technology as a means to offer customized communication service capabilities over the same infrastructure by partitioning it into individualized slices [5]. In this way, it is possible to satisfy the service requirements of different vertical industries. The slices consist of a set of Virtual Network Functions (VNFs) that encapsulate specific sub-services that the slice needs to provide the service functionalities it was designed for [6]. VNFs are mapped to physical nodes of the infrastructure, while the virtual links of the slice are mapped to physical links. The future 5G networks are projected to support massive numbers of network slices with different performance requirements, functionality and timespans [7-9] working concurrently, which together with the already high complexity of the network slicing solution, makes the tasks related to management and orchestration problematic.

The network slice management differs from classical network management schemes as it requires administrating not a single but multiple network domains and the complexity of architectures and the number of managed objects scale up rapidly. The slice-based network management concept generates a set of challenges related to scalability, security, automation in the management of heterogeneous resources (e.g. communication, computational and storage), as well as to energy efficiency without sacrificing performance. The elevated requirements for coverage, bandwidth and latency, as well as inter-domain operation, further exacerbate the complexity of network management [10], making already devised, standard, human-centric managing solutions insufficient and ineffective. The currently widespread centralized approach to network management also negatively impacts the separation and security of network slices as well as the complexity of the central managing entity. Moreover, centralisation also increases the overhead related to slice management data that has to be sent to the management system during the slice operation.

2 Zero-Touch Management in the Scope of the MonB5G Project

The Zero-touch network and Service Management (ZSM) framework [11] is envisaged as a next-generation management system that aims to have all operational processes and tasks executed automatically, ideally fully autonomous. It is based on a variety of distinct principles, including modularity, extensibility, scalability, model-driven and open interfaces, closed-loop management automation, support for stateless management functions, resilience, separation of concerns in management, service composability, intent-based interfaces, functional abstraction, simplicity and automation. In this scope, Artificial Intelligence (AI) is envisioned as a “key enabler” of self-managing capabilities, resulting in lower operational costs, accelerated time-to-value and reduced risk of human error [12].

Zero-touch management is perceived as one of the “key concepts” that can contribute to and significantly simplify network slice management and orchestration tasks. With the extensive usage of AI-driven mechanisms, its goal is to provide self-managed networks with little to no human interaction [13]. With the aim of facing this challenge and within the scope of the ongoing EU-funded MonB5G project [14], innovative architecture is proposed, capable of “addressing” the aforementioned issues by enabling distribution of functions and provision of strong separation of management of network slices’ runtime and orchestration domains.

More specifically, MonB5G aims to provide a new model for the management and orchestration of high numbers of parallel network slices as envisioned in 5G and beyond. One of its core pillars is about promoting data-driven mechanisms, based on novel distributed machine learning algorithms, to enable self-management and self-configuration of network slices, towards reaching the principle of scalable zero-touch network management, where federated Machine Learning- (ML-) aided algorithms will enable a proactive, energy-efficient and secure resource management and slice creation [15]. The main goal of the MonB5G approach is to achieve scalable and automated management of multiple network slices.

The proposed architectural concept intends to facilitate self-managed slices composed of self-managed functions, further extended to slices that are created in multiple orchestration domains. In the related scope, the issue of management complexity is addressed by using AI at multiple levels to achieve specific management goals and to minimize interactions between architectural entities, e.g. by means of hierarchical closed-loop controls and aggregated Key Performance Indicators (KPIs). A heavy emphasis is also on security [16], management programmability and energy efficiency [17] aspects of network slicing. Furthermore, the proposed concept is deeply rooted in the already devised network slicing management and orchestration solutions that have been developed by other EU-funded projects (in the 5G-PPP framework) or research and standardization bodies.

The MonB5G framework uses the management system decomposition that follows the ITU-T [18] and the MAPE (Monitor-Analyse-Plan-Execute) paradigm [19] as the basis. In our case, the MAPE concept is implemented in a distributed way by means of multiple AI-driven operations. Moreover, the runtime management of slices is distributed and programmable. Additionally, the MANO approach has been slightly enhanced by distributing some orchestration functions [20, 21].

3 Architectural Framework

The key features of the proposed MonB5G framework are the following: (i) a strong separation of concerns; (ii) distribution of management operations; (iii) hierarchical, end-to-end (E2E) slice orchestration; (iv) In-Slice Management (ISM) capability of driving the orchestrator; (v) scalable and programmable slice management; (vi) enhanced security of slices; (vii) support for Management as a Service (MaaS), and (viii) programmable, energy-aware infrastructure management. The abovementioned features are in line with several already established requirements of ETSI ZSM [23].

The MonB5G architecture is composed of static and dynamically deployed components. Altogether, they support operations related to slicing orchestration, fault management (self-healing), self-configuration, performance optimization (including energy-saving), and security-related operations of slices. The overall MonB5G framework is presented in Fig.1. The MonB5G framework is composed of three distinct layers:

- **Business Layer**, consisting of the business entities operating the framework, provides slice management services to slice tenants or own a slice (slice tenants).
- **Management and Orchestration Layer**, composed of the core functions of the framework responsible for management and orchestration of Network Slices (NSs), slice Life Cycle Management (LCM), and management interfaces' exposure to specific business entities.
- **Infrastructure Layer**, consisting of the infrastructure, infrastructure providers and functions enabling communication with Management and Orchestration Layer and enabling optimization of usage of infrastructural resources.

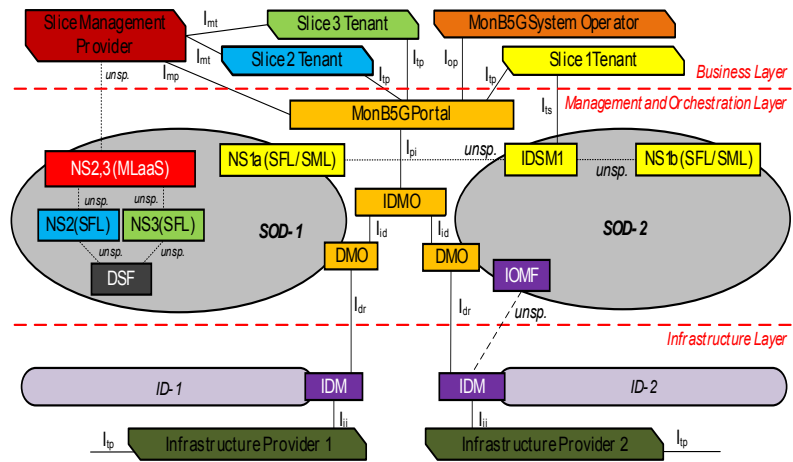


Fig. 1. MonB5G architectural framework.

According to the architecture presented in Fig.1, we can distinguish *static* and *dynamic components*. The *static components* are described and discussed in the following paragraphs. Slice Tenants use the MonB5G portal, Slice Management Providers and Infrastructure Providers to request operations regarding slice LCM (i.e. slice deployment, slice modification and slice termination). It also exposes the capabilities offered by the MonB5G framework (available slice templates, etc.) and partakes in negotiations related to the business dimension of the contract. The portal is also used to pass all the accounting and billing-related information and it implements three northbound web interfaces that expose the MonB5G framework capabilities to: MonB5G System Operator (I_{op} , allowing the operator to manage the whole MonB5G system), Slice Tenants (Slice Tenants and Infrastructure Providers use I_{ip} – the interface for slice LCM) and Slice Management Providers (I_{mp} – the interface that the Slice Management provider uses for the communication with Slice Tenants for runtime slice management). Slice Management Providers may use the MaaS platform,

called MonB5G Layer as a Service (MLaaS), to offer management of multiple instances of slices based on the same template (as the slice runtime management is slice-specific). LCM of MLaaS is done via the I_{mp} interface. I_{mt} is a web interface used by Slice Management Provider for the runtime slice management communication with Slice Tenants

The *Inter-Domain Manager and Orchestrator (IDMO)* is at the heart of the system. This entity plays a crucial role in slice preparation and deployment phases by negotiation of deployment policy with a slice requester (Slice Tenants, Slice Management Providers or Infrastructure Providers). MonB5G Portal interacts with IDMO via the southbound I_{pi} interface to perform negotiations related to the business dimension of the contracts. The exchanged information concerns aspects like availability of resources, existing policies, the resource demand and other data that enables allocation of a certain amount of resources to the requester. After the successful establishment of the contract, the I_{pi} interface is used for LCM of negotiated slices. If the infrastructure has multiple owners, IDMO may decide how to split the end-to-end slice template dynamically to a new one, which supports inter-domain interaction of slice components located in different orchestration domains. The split may be shaped by various factors, e.g. price, performance or energy efficiency.

The *Domain Manager and Orchestrator (DMO)* is responsible for the orchestration and management of each of the Slice Orchestration Domain (SOD) slices. DMO can be seen as a combination of resource-oriented Operations Support Systems (OSS)/ Business Support Systems (BSS) and an orchestrator (it can be either a MANO orchestrator or other). In a similar way to IDMO, all DMO operations are AI-driven. Therefore, the internal structure of DMO is also composed of Functional and MonB5G Layers. The operations related to resource management as well as the exchange of infrastructure-related data (e.g. about energy consumption) are done via the I_{dr} interface.

IDMO interacts with DMOs via the I_{id} interface by using domain handlers to deploy the end-to-end-slices based on the information obtained from DMOs. This interface can be seen as an extended MANO *Os-Ma-Nfvo* interface, and it may provide LCM abstractions and provides IDMO data and management capabilities of DMO. It is responsible for modification of the end-to-end slice template before its deployment according to the negotiated contract, and it can be seen as an E2E orchestrator. IDMO may also interact with IDM (via DMO) in order to decide how to deploy slice instance, considering the price, performance or other important factors such as energy efficiency.

The proposed framework assumes that the Infrastructure may also need management. To that end, it is proposed a separate management entity called *Infrastructure Domain Manager (IDM)*. This provides the overall management of the Infrastructure. Its interface to DMO allows for allocating resources (Network Functions Virtualisation Infrastructure (NFVI) agent), exchanging the information related to the energy consumption of resources, and exchanging the information related to the cost of resources that can be used by IDMO for resource brokering. The framework enables programmable infrastructure management. DMO can dynamically deploy management functions that cooperate with IDM to achieve infrastructure management. IDM has an interface to the Infrastructure Provider, who can use the MonB5G portal to ask for the deployment of additional infrastructure management functions, called

Infrastructure Orchestrated Management Functions (IOMFs), which are specific for the virtualization technology used in the infrastructure and tools. IDMO can orchestrate the functions upon request of an Infrastructure Provider via the MonB5G Portal.

The *architecture's dynamic components* are slices that are defined in a different way than NGMN has defined them. In MonB5G, a slice is a set of functions that implement a specific goal (not necessarily a network), for example, network management, implementation of services or accelerators that support certain operations of multiple slices.

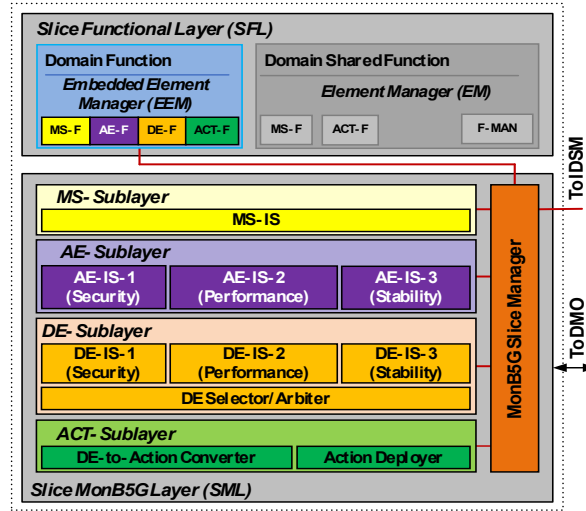


Fig. 2. Generic structure of MonB5G slice.

In MonB5G slice structure (cf. Fig.2), two separate layers can be distinguished: the slice management part called *Slice MonB5G Layer (SML)* and slice main part called the *Slice Functional Layer (SFL)*. The former performs FCAPS (Fault, Configuration, Accounting, Performance, Security) at the slice-level and can be considered as an embedded slice-level OSS/BSS, with interfaces to the Element Managers (EMs) of the slice's Virtual Network Functions (VNFs)/Physical Network Functions (PNFs) or Cloud-Native Functions (CNFs) and to DMO. SML can be a part of the slice template or be deployed independently.

SFL contains a set of virtual functions that are dedicated solely to a slice and managed via the modified *Element Manager (EM)*, called *Embedded Element Manager (EEM)*, which contains components responsible for its VNF monitoring (MS-F), anomaly detection (AE-F), decision engine (DE-F) and actuating component (ACT-F). SFL can also use functions that are shared functions available in SOD. Such functions may be used by all or some slices. The functions are called *Domain Shared Functions (DSFs)*, which can be implemented as PNFs/VNFs or CNFs and can be reused by SFLs of multiple slices. The use of DSFs provides a reduced footprint of the deployed slices improving that way also slices deployment time. DSFs are grouped for their management and are managed by the DMO.

SML is an implementation of the ISM [23] concept having in mind the AI-based MAPE management and it provides direct, intent-based management to the Slice

Tenant. The SML-based slice management approach can also be used for end-to-end slice management [23] when slices are implemented across multiple domains (SODs). In such a case, the IDSM entity is responsible for the end-to-end slice management. It interacts with SMLs of all domain slices that compose the end-to-end slice. IDSM is a part of the slice template (a set of VNFs), and in some cases, it can be generated automatically by IDMO (if IDMO is responsible for slice template split between multiple SODs). When IDSM is in use, it provides to the Slice Tenant the management interface. IDSM is also responsible for the calculation of slice-related KPIs. I_{ts} is a web interface used by Slice Tenant for runtime slice management and interaction with IDSM or SML.

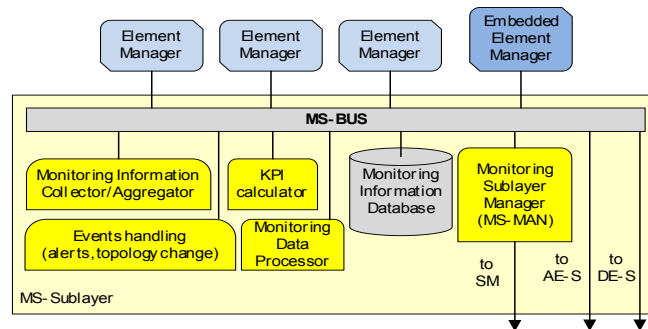


Fig. 3. Monitoring System Sublayer internal components.

It is assumed that Monitoring System Sublayer provides generic, reusable monitoring that is consumed by AEs, DEs and other entities of SML, as presented in Fig.3. MS should contain:

- *Monitoring Information Collector/Aggregator* – an entity, which interacts with the *Embedded Element Managers (EEMs)*, i.e. MAPE-based Element Managers (presented in Fig.2) of SML;
- *Monitoring Information Database* – a database in which collected monitoring data are stored in a raw and pre-processed format;
- *Monitoring Information Processor* – an entity that is responsible for filtering, interpolation and prediction of the monitoring data;
- *Slice KPI calculator* – an entity that is used for the calculation and prediction of slice specific KPIs
- *Monitoring Sublayer Manager* – an entity that allows remote configuration of MS sublayer operations.

The MS sublayer's output is accessible to other components of SML via a message bus (publish/subscribe paradigm in use). MS has to interact with EEMs that are VNF-specific, but most of the MS operations are generic. Therefore, many of the internal components of MS can be reused for multiple slice templates. The protocols for efficient communication between EEMs and MS and the adaptability of monitoring (adaptive sample rate or resolution, using gossiping protocols, etc.) are out of the scope of the paper, but such mechanisms are allowed by the MonB5G framework.

The *Analytic Engines Sublayer* includes a set of AEs and the *AE Sublayer Manager* that is used to configure AEs remotely, as depicted in Fig.4.

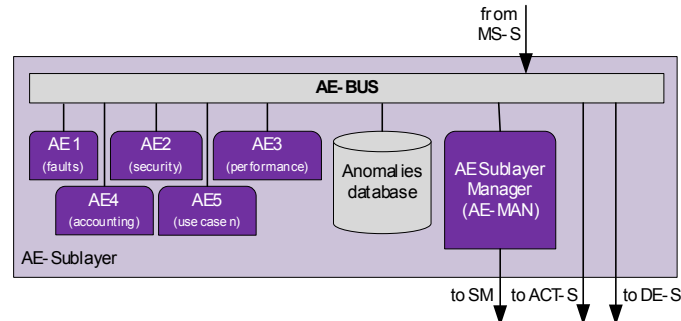


Fig. 4. Analytic Engine Sublayer internal components.

Each of AEs has a specific goal, i.e. it may analyse the monitoring traffic for a specific security threat, fault or performance degradation. The internal specification of AE is algorithm-dependent and cannot be provided with *a priori*; it is, however, possible to create a library of AEs that, with a relatively small adaptation, can be used for different slice templates. In general, it is assumed that between AE and DE, there is a one-to-one mapping, but the architecture allows to use of multiple AEs for the same DE. It is worth mentioning the MS sublayer provides some kind of abstraction of the monitored data that positively contributes to the reusability of AEs.

The *DE Sublayer* consists of the entities that are responsible for the reconfiguration of SFL or SML. It is assumed that the DE sublayer is composed of multiple DEs, as presented in Fig.5. Each of them is trying to reach a specific goal regarding performance optimization according to KPIs, fault handling, security, or enforcing the slice's energy-efficient operations. The input to the DE sublayer is the output of AE and ME sublayers. Each of the FCAPS functions may require multiple DEs. The existence of multiple "selfish" DEs implies the need for their decisions arbitrage. For this problem, the *DE Selector/Arbiter* component is implemented in the DE sublayer. The component can be AI-driven as one of the implementation options is implementing several DEs that use different algorithms for the same goal. In such a case, the *Coordinator/Arbiter* is creating a ranking of DEs. As the feedback-loop-based management's stability can be an issue, a special entity called *Stability Observer* is introduced in the DE sublayer. It is used in order to avoid the chaotic behaviour of the system or the ping-pong effect. The *Stability Observer* uses *Reconfiguration History Database* to restore to the last stable configuration. This database stores recent reconfiguration decisions together with the input values that were used by DEs to take the decision reconfiguration. As other sublayers of SML, the DE sublayer has a *DE Sublayer Manager* that can be used for the change of the configuration of its components or their policies.

The DE decision can be used for the reconfiguration of SFL or SML. In both cases, there are three possible reconfiguration operation types:

- Reconfiguration of functions/nodes of SFL/SML;
- Change of resource allocation to SFL/SML components (including transport). Dependent on the implementation, it can be done directly or by the interaction with DMO;

- Modification of SFL/SML by the upgrade of the slice template. In this case, the SML will interact with DMO requesting deployment or removing a specific function or a node.

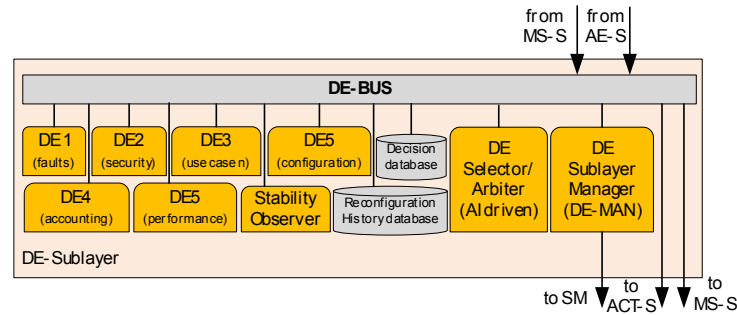


Fig. 5. DE Sublayer internal components.

It is noteworthy that the modification of resource allocation to SFL or SML by SML can be proactive instead of the MANO orchestration's reactive approach. Moreover, in the case of SFL, resource allocation can be driven by slice Quality of Experience. The modification of the SFL template, driven by SML, may be used for cloning some slice functions to optimize slice traffic or add additional components like DPIs or firewalls. The same mechanism can be used for the programmability of SML, providing that way programmability of the slice management plane. Using the mechanism during SML runtime, new components like AEs or DEs can be added. This feature is vital for the evaluation of different AE and DE algorithms, but the programmability is also important in real implementations.

The *Actuating Functions Sublayer* role is to convert high-level (intent) reconfiguration commands obtained from the DE sublayer into a set of atomic reconfiguration commands, as shown in Fig.6.

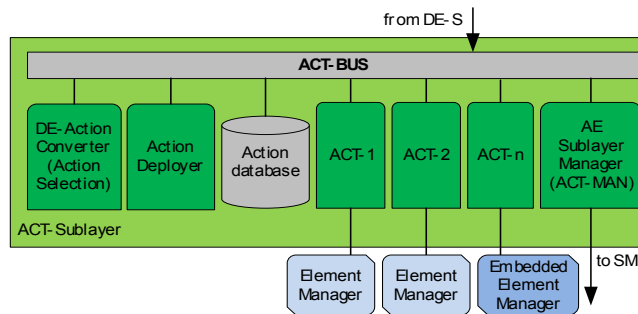


Fig. 6. ACT-Sublayer internal components.

Due to the existence of the ACT sublayer, the DEs do not have to deal with reconfiguration details. Therefore, they can be designed in a more generic way. The ACT sublayer can be seen as a set of device-specific (i.e. node/functions) drivers. ACT typically interacts with EMs/EEMs of SFL, but they may also interact with DMO requesting orchestration related action (adding or removing a VNF).

Slice Manager (SM), cf. Fig.7, is an entity of SML that provides interactions with DMO and IDSM; it can also be used for the manual management of SFL or to implement Policy-Based Management. It interacts with EEMs, MS, AEs and DEs. It is responsible for sending to DMO and, if applicable, to IDSM, slice-related synthetic information (KPIs). SML provides direct, intent-based management to the Slice Tenant. This is a perfect way of providing slices management isolation. For that purpose, the Slice Manager has a tenant portal and a set of tools that enable simple and comfortable slice management by slice tenant. A *conditio sine qua non* for such management is the management's embedded intelligence that is in our case provided by AI algorithms.

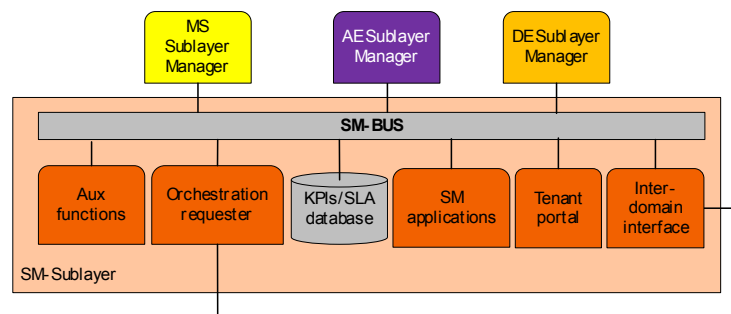


Fig. 7. Slice Manager internal components.

The management interface is created after slice deployment, and the slice tenant can use it for the lifetime of a slice. For accounting and historical reasons, the accounting data combined with slice resource consumptions and KPIs are transferred to the IDMO Accounting database before termination of the slice.

4 Assessment and Further Expectations

According to our best knowledge, the proposed preliminary concept is the first one that addresses the scalability and robustness of network slicing management and orchestration by using a distributed, AI-driven and programmable management architecture. These features make the proposed approach effective, scalable and flexible, on a dynamic basis.

The ETSI ZSM requirements list contains over 170 different topics that makes impossible to recall all of them in the context of the current MonB5G approach. More specifically, some of the requirements concern procedures like testing or software upgrade; therefore, they are not related directly to the actual MonB5G architecture. The list is “flat” and without any grouping. For the purpose of our current approach, it has been decided to split them into several categories related to the specific aspects of autonomic service management. These categories all fully satisfied by the MonB5G context and include: (i) Monitoring and data analytics (which is about the requirements associated with collection of the performance data, their aggregation and ways of data usage to fuel analytic engines); (ii) management actions (which consist of requirements in terms of network maintenance, coordination of management, reco-

very actions, etc.); (iii) management operations (which relate to access to network slicing management services, LCM, management data policies, etc.); (iv) control loops (their operation, disabling in terms of faulty operation, etc.); and (v) several other important requirements not belonging to previous categories.

The use of distributed components with embedded intelligence has made it possible to use intent-based interfaces that also reduce the information exchange between management functions and subsystems. Moreover, we have used a multi-domain orchestration and separation of each slice's management from domain resource management. The use of the ISM concept can reduce the number of external slice interfaces and provides a perfect separation of the slice management plane that cannot be achieved in the 3GPP approach to network slicing management. The implementation of slice management as a part of a slice (i.e. a set of VNFs) provides higher scalability of slice performance and allows for the programmability of slice management services on-the-fly.

In addition, AI-enabled management operations can be adopted at different levels of the management hierarchy. In this scope, AI-driven slice management functionalities can be embedded as a part of a slice, providing, in that way, the higher elasticity in the creation and the deployment of diverse slice types. This framework also provides a strong separation of concerns which contributes significantly to complexity reduction and easier administration of slices, especially in multi-domain slices deployed over different infrastructure domains belonging to several owners. Altogether the abovementioned features enable making a significant step towards self-managed network slices.

Acknowledgments. This work has been performed in the scope of the *Mon5G* European Research Project and has been supported by the Commission of the European Communities (*H2020, Grant Agreement No. 871780*).

References

1. Global System for Mobile Communications Alliance (GSMA): An Introduction to 5G Network Slicing. GSMA (2017, November). <https://www.gsma.com/futurenetworks/wp-content/uploads/2017/11/GSMA-An-Introduction-to-Network-Slicing.pdf>
2. Rost, P., Banchs, A., Berberana, I., Reitbach, M., Doll, M. et al.: Mobile network architecture evolution toward 5G. *IEEE Communications Magazine* **54**(5), 84-91 2016.
3. Andrews, J.G., Buzzi, S., Choi, W., Hanly, S.V., et al.: What will 5G be. *IEEE JSAC, Special Issue on 5G Wireless Communications Systems* **32**(6), 1065-1082 (2014)
4. Chochliouros, I.P., Kostopoulos, A., Spiliopoulou, A.S., Dardamanis, A., Neokosmidis, I., Rokkas, T., and Goratti, L.: Business and market perspectives in 5G Networks. In: *Proceedings of the Joint 13th CTTE and 10th CMI Conference 2017*, pp. 1--6. IEEE (2017)
5. Bosneag, A.M., and Wang, M.X.: Intelligent network management mechanisms as a step towards 5G. In: *Proceedings of the NOF-2017 Conference*, pp. 52-57 IEEE (2017)
6. European Telecommunications Standards Institute: ETSI GS NFV 002 V1.2.1 (2014-12): "Network Functions Virtualisation (NFV); Architectural Framework", ETSI. https://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.02.01_60/gs_nfv002v010201p.pdf
7. Sharma, S., Miller, R., and Francini, A.: A cloud-native approach to 5G network slicing. *IEEE Communications Magazine* **55**(8), 120-127 (2017)

8. Elliott, J., and Sharma, S.: Dynamic End-to-End Network Slicing unlocks 5G Possibilities. Nokia (2016). <https://www.nokia.com/blog/dynamic-end-end-network-slicing-unlocks-5g-possibilities/>
9. Chochliouros, I.P., Spiliopoulou, A.S., Lazaridis, P., Dardamanis, A., Zaharis, Z., and Kostopoulos, A.: Dynamic Network Slicing: Challenges and Opportunities. In: Proceedings of AIAI-2020, IFIP AICT 585, pp. 47--60. Springer, Cham (2020)
10. Chochliouros, I.P., Spiliopoulou, A.S., Kostopoulos, A., et al.: Putting intelligence in the network edge through NFV and cloud computing: The SESAME approach. In: Proceedings of EANN-2017, CCIS 744, pp. 704--715. Springer, Cham (2017)
11. European Telecommunications Standards Institute: ETSI GS ZSM 002 V1.1.1 (2019-08): "Zero-touch Network and Service Management (ZSM); Reference Architecture", ETSI. https://www.etsi.org/deliver/etsi_gs/ZSM/001_099/002/01.01.01_60/gs_ZSM002v010101p.pdf
12. Benzaid, C., and Taleb, T.: AI-Driven Zero Touch Network and Service Management in 5G and Beyond: Challenges and Research Directions. IEEE Network **34**(2), 186-194 (2020)
13. Carrozzo, G., Shuaib Siddiqui, M., Betzler, A., Bonnet, J., et al.: AI-driven Zero-touch Operations, Security and Trust in Multi-operator 5G Networks: a Conceptual Architecture. In: Proceedings of EuCNC 2020, pp. 254--258. IEEE (2020)
14. H2020/5G-PPP MonB5G project: "Distributed management of Network Slices in beyond 5G", Grant Agreement No.871780. <https://www.monb5g.eu/>
15. Jiang, C., Zhang, H., Ren, Y., Han, Z., et al.: Machine learning paradigms for next-generation wireless networks. IEEE Wireless Communications **24**(2), 98-105 (2017)
16. Benzaid, C., and Taleb, T.: ZSM Security: Threat Surface and Best Practices. IEEE Network Magazine **34**(3), 124-133 (2020)
17. Chen, X., Wu, J., Cai, Y., Zhang, H., et al.: Energy-efficiency oriented traffic offloading in wireless networks: A brief survey and a learning approach for heterogeneous cellular networks. IEEE Journal on Selected Areas in Communications **33**(4), 627-640 (2015)
18. International Telecommunication Union – Telecommunication Standardization Sector (ITU-T) (2000): "ITU-T Recommendation M.3000 (02/00): Overview of TMN Recommendations" (2000)
19. IBM Corporation: An architectural blueprint for autonomic computing. Autonomic Computing White Paper, 4th edition (2006)
20. European Telecommunications Standards Institute: ETSI GS NFV-MAN 001 V1.1.1 (2014-12): "Network Functions Virtualisation (NFV); Management and Orchestration", ETSI. https://www.etsi.org/deliver/etsi_gs/nfv-man/001_099/001/01.01.01_60/gs_nfv-man001v010101p.pdf
21. European Telecommunications Standards Institute: ETSI GR NFV-IFA 023 V3.1.1 (2017-07): "Network Functions Virtualisation (NFV); Management and Orchestration; Report on Policy Management in MANO; Release 3", ETSI. https://www.etsi.org/deliver/etsi_gr/NFV-IFA/001_099/023/03.01.01_60/gr_NFV-IFA023v030101p.pdf
22. European Telecommunications Standards Institute: ETSI GS ZSM 001 V1.1.1 (2019-10): "Zero touch network and Service Management (ZSM); Requirements based on documented scenarios", ETSI. https://www.etsi.org/deliver/etsi_gs/ZSM/001_099/001/01.01.01_60/gs_ZSM001v010101p.pdf
23. Kukliński, S., and Tomaszewski, L.: DASMO: A scalable approach to network slices management and orchestration. In: proceedings of NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium, pp. 1--6. IEEE.