AI-DRIVEN PREDICTIVE AND SCALABLE MANAGEMENT AND ORCHESTRATION OF NETWORK SLICES

Sławomir Kukliński[1,2], Lechosław Tomaszewski[1], Robert Kołakowski[1,2], Anne-Marie Bosneag[3], Ashima Chawla[3], Adlen Ksentini[4], Sabra Ben Saad[4], Xu Zhao[5], Luis A. Garrido[6], Anestis Dalgkitsis[6], Bahador Bakhshi[7], Engin Zeydan[7]
[1]Orange Polska, Orange Innovation Poland, ul. Obrzeżna 7, 02-691 Warszawa, Poland, [2]Warsaw University of Technology, Faculty of Electronics and Information Technology, ul. Nowowiejska 15/19, 00-665 Warszawa, Poland, [3]Ericsson Ireland, Network Management Lab, Athlone, Co. Westmeath, N37PV44, Ireland, [4]Eurecom, Campus SophiaTech, 450 Route des Chappes, 06410 Biot, France, [5]NEC Laboratories Europe, Kurfürsten-Anlage 36, 69115 Heidelberg, Germany, [6]Iquadrat Informatica, S.L, Carrer Doctor Rizal, 10, 08006, Barcelona, Spain, [7]Centre Tecnològic de Telecomunicacions de Catalunya, Carrer Doctor Rizal, 10, 08006, Barcelona, Spain

NOTE: Corresponding author: Sławomir Kukliński, slawomir.kuklinski@orange.com

Abstract – *The future network slicing enabled mobile ecosystem is expected to support a wide set of heterogenous vertical services over a common infrastructure. The service robustness and their intrinsic requirements, together with the heterogeneity of mobile infrastructure and resources in both the technological and the spatial domain, significantly increase the complexity and create new challenges regarding network management and orchestration. High degree of automation, flexibility and programmability are becoming the fundamental architectural features to enable seamless support for the modern telco-based services. In this paper, we present a novel management and orchestration platform for network slices, which has been devised by the Horizon 2020 MonB5G project. The proposed framework is a highly scalable solution for network slicing management and orchestration that implements a distributed and programmable AI-driven management architecture. The cognitive capabilities are provided at different levels of management hierarchy by adopting necessary data abstractions. Moreover, the framework leverages intent-based operations to improve its modularity and genericity. The mentioned features enhance the management automation, making the architecture a significant step towards self-managed network slices.*

Keywords – 5G, 6G, AI, management, ML, network slicing, orchestration, ZSM

1. INTRODUCTION

Network slicing is a relatively new approach associated typically with a 5G System (5GS), but this technology can be used in a generic, virtualised networking environment. The main value of network slicing is the ability of dynamic creation of multiple service or network-oriented, isolated solutions. The concept requires slice lifecycle or runtime management and orchestration procedures. They have been defined in detail by the 3rd Generation Partnership Project (3GPP) for 5G networks, but with several limitations and so far not fully implemented in commercial 5GS deployments.

One of the main drawbacks of the 3GPP and European Telecommunications Standards Institute (ETSI) Network Function Virtualisation (NFV) approaches is centralization and the high complexity of management. In both cases, a single central Operations Support System (OSS)/Business Support System (BSS) solution is proposed to cope with all deployed Network Slice Instances (NSIs). Moreover, the monitoring data from all nodes or functions of a slice have to feed the central OSS/BSS. The overall management complexity depends on the number of functions and intricacy of NSIs, and the number of NSIs themselves, which is expected to be high. Furthermore, the management and orchestration mechanisms have to provide additional common and resource-oriented operations for all network slices. In fact, a single network slice may provide functionality similar to a classical network, therefore its management complexity can be compared to the management of a single network or a network combined with service(s). Such comparison shows well the complexity of management of multiple, potentially hundreds, of slices. As the number of running NSIs is changing dynamically, it is hard to predict the required performance of the management system. The management performance can be increased by the use of automation, typically based on the Monitor-Analyse-Plan-Execute (MAPE) paradigm introduced over 20 years ago in the context of autonomic computing [1]. The main disadvantage of MAPE is the need for fast monitoring and analysis of the monitoring data in real time. MAPE is a preferred solution for network slicing management and orchestration, however.

In the paper, the MonB5G project concept that addresses the mentioned issues is presented. Its main goal is to provide a highly scalable and performant management plane for network slices to enable quick reactions to events. To that end, a distributed architecture with operations driven by Artificial Intelligence (AI) has been designed. The AI algorithms are used for efficient monitoring, anomaly detection and network reconfiguration. To increase the management scalability, each NSI has an embedded management plane and all management components are programmable.

In order to emphasize the value of the concept, in Section 2 of the paper, an overview of the existing approaches is discussed. Section 3 describes the design assumptions and the architecture of the MonB5G system, whose components are presented in Section 4. Section 5 is devoted to the implementation approach of the MonB5G system and used technologies. Section 6 summarizes and concludes the paper.

2. RELATED WORK

The topic of network slices management and orchestration has attracted immense efforts of both academic and standardization bodies. The fundamentals of network slicing have been defined by the Next Generation Mobile Networks (NGMN) Alliance [2]. According to NGMN, the NSI is defined as a logical network that can be customized for the specific needs of some service or group of services. Each NSI is created on the basis of a pre-defined template and is built over a shared infrastructure composed of fully or partially isolated physical or logical computation, storage and transport resources. End-to-End (E2E) NSI can be composed of a single slice or multiple concatenated network sub-slices. Moreover, there exists a clear separation between NSI and end-users' services; i.e., the interactions occur via Application Programming Interfaces (APIs) and services are treated as external to NSI. The NGMN vision has been followed by both the ETSI NFV Framework as well as 3GPP in 5GS.

The network slices management and orchestration defined in the 3GPP Release 17 adopts theETSI NFV Management and Orchestration (MANO) framework [3]. The runtime management and Lifecycle Management (LCM) processes are handled by a single, centralized OSS/BSS and a single NFV MANO orchestrator, being responsible for the analysis of the abstracted description of a slice, creating an optimal placement strategy for the slice virtual functions and resource-scaling during the slice runtime. 3GPP defines four management levels of network slicing-based networks: network function, slice subnet, E2E slice, and communication service management.

Other ETSI standardization groups, such as Experiential Networked Intelligence (ENI) [4] and Zero-touch network and Service Management (ZSM) [5], aim to further extend the management and orchestration by applying network-specific AI/Machine Learning (ML) techniques facilitating automation. Application of these centralized concepts for network slicing is difficult.

Some efforts are also conducted to implement AI/ML in Radio Access Network (RAN) resources management. In [6], the O-RAN Alliance introduces network performance improvements using a collection of RAN-related metrics. However, the provisioning of network slicing support is still unclear.

On the basis of standardization, several open-source community projects have been launched to provide the implementation of the aforementioned concepts. In particular, the most noteworthy initiatives include Open Network Automation Platform (ONAP) [7] and Open-Source MANO (OSM) [8]. ONAP is one of the main solutions to implement a highly centralized framework accelerated by AI/ML, which could satisfy the network needs in terms of management and resource orchestration automation. In spite of considerable efforts that resulted in several published releases, there are no real-life commercial ONAP deployments, yet. OSM is an ETSI NFV-compliant MANO orchestrator supporting slicing, developed by the Linux Foundation. Moreover, it is noteworthy that both ONAP and OSM have already started the work on AI/ML for autonomous management provisioning.

Numerous projects under 5G Infrastructure Public Private Partnership (5G-PPP) have notably progressed in terms of system architectures and facilitation towards effective application of AI/ML for management and orchestration. Their achievements are described in [9].

The essential aspect for supporting AI/ML in management and orchestration is provision of the required granularity and distribution of the system. The 5G!Pagoda [10] project was the first step towards management and orchestration of a high number of parallel NSIs. The introduced reference architecture put a large emphasis on the scalability of a management plane, proposing the In-Slice Management (ISM) concept [11], which enables the distribution of management functions and their embedding inside the NSI. However, the support for AI/ML-driven management has not been included.

Several advancements have been done in terms of multi-domain operation support. 5G-VICTORI [12] proposed the architecture to facilitate the management of slices, resources and orchestration of services across different facilities and technological or administrative domains. The platform also supports seamless integration with the NFV MANO platform by reflecting necessary extensions in each facility. Additional innovations have been introduced by 5G-MoNArch [13]. Apart from the support of coordinated cross-domain management across slices and domains, the flexibility of Virtual Network Functions (VNFs) orchestration has also been increased by employing cloud-enabled protocols. The support for highly performing algorithms has also been considered by adopting experiment-driven optimization and ETSI ENI [4]. Moreover, some exploration of AI/ML for network slices management has been conducted. However, the solution might not scale well with the vast number of NSIs, due to its high centralization. 5G-CLARITY [14], apart from proposing the AI-based network management system, extended the concept by providing the intent-based interfaces enabling network configuration.

Significant progress has been also achieved regarding the improvement of cognitive capabilities of network slice management and orchestration. SELFNET [15] and SLICENET [16] projects proposed mechanisms to target self-organization of the network built atop NFV/Software-Defined Networking (SDN) paradigms and AI/ML technologies to simplify adoption of network slices for verticals. Autonomic slice management has been also approached by 5G-ZORRO [17], and 5G-Ensure [18] projects that proposed zero-touch management architectures adopted for a multi-stakeholder scenario while putting emphasis on aspects of network security and trust maintenance.

Major achievements can also be observed in terms of network slice management and orchestration optimization algorithms. The autonomicity of these processes is a crucial part of the vision in 5G/B5G networks. For this reason, both academia and industry have generated significant state-of-the-art algorithms and complete frameworks, generally speaking, as contributions to orchestration problems in these scenarios. Some problems include Service Function Chaining (SFC) embedding, VNF placement and orchestration [19], and admission control of both VNFs and user service requests [20].

Along these lines of research, authors in [21] developed a heuristic algorithm to optimize the network bandwidth consumption by taking advantage of the SFC placement. On the other hand, it is also possible to study the placement or embedding problem using tabular methods in Reinforcement Learning (RL) and Deep Reinforcement Learning (DRL). Developing heuristics that generalize well across different system scenarios has long been proven to be a difficult task, due to the customized nature of the heuristics design. In the process of further advancing the sophistication of algorithms for SFC placement, the authors in [22] proposed DRL to perform the placement of virtual network function forwarding graphs considering the constraints of the underlying infrastructure. The authors of [23] used DRL as in [22] for building an orchestration solution consisting of multiple DRL agents with the objectives of minimizing latency across SFCs and minimizing energy consumption. This approach also has the capability of orchestrating VNFs across multiple domains in a network.

The state-of-the-art literature on Slice Admission Control (SAC) is fairly extensive. Many algorithms have been proposed as solutions and many approaches formulate the SAC problem as a combinatorial optimization problem [24, 25], with sometimes heuristic solutions [26, 27], analytic solutions for very specific scenarios [28], or more advanced AI-based algorithms, specifically using RL [29, 30] or genetic algorithms [29, 31].

3. MONB5G SYSTEM ARCHITECTURE

The MonB5G framework has been designed for AI-driven management and orchestration of massive number of NSIs. It supports operations of fault and security management, self-healing, self-configuration, and performance optimization (including energy-saving). The framework follows the MAPE paradigm [1] as well as the International Telecommunication Union – Telecommunication Standardisation Sector (ITU-T) management system decomposition [32] and is founded on the key features, which are inline with key ETSI ZSM [5] requirements.

The main features of the framework are the following:

- *Hierarchical E2E slice orchestration.* The proposed approach uses one master orchestrator and multiple domain-level orchestrators. Such distributed, multilevel orchestration improves its scalability.

- *Hierarchical distribution of management operations.* These operations are AI-driven, distributed and pursue different goals. The hierarchy of the embedded management concerns node (or function), (sub-)slice, orchestration domain and the E2E slice. It enables the processing of management information at each of the mentioned levels, thus reducing the overall management data exchange. The scope of operations of OSS/BSS of each orchestration domain is limited to the NSIs' lifecycle management and to management and orchestration of resources of this domain only. It is also agnostic to the deployed NSIs.

- *Slice runtime management and orchestration as a part of each NSI.* The runtime management of each NSI is not a part of OSS/BSS, but it is embedded within the NSI as a part of its template. That way, the isolation of management planes of individual NSIs is provided, which is absent in the approaches of ETSI NFV MANO or 3GPP. Moreover, the management plane is implemented as a set of VNFs that contributes to the slice runtime management performance via dynamic resource allocation to these functions and their proper placement. It enables direct management of NSIs by their tenants without the need to use the orchestration domain OSS/BSS. Each NSI may act as a service orchestrator, i.e., it may request its template modifications via the *Os-Ma-nfvo*-like interface [3] of the orchestrator.

- *E2E slice management as a part of slice template.* In order to provide the E2E management of multi-domain slices, a unique, inter-domain component is defined as a part of the E2E slice template.

- *Programmable infrastructure management.* The MonB5G mechanisms for slices deployment can be reused by the infrastructure operator for the deployment of infrastructure services.

- *Management as a Service (MaaS).* The MonB5G framework enables the creation of a “management slice” only, i.e. a set of management dedicated VNFs. It can be used for runtime management of multiple NSIs instantiated using the same template. Such a slice can be operated by an entity called Management Provider.

The hierarchical MAPE/AI-based Fault, Configuration, Accounting, Performance, Security (FCAPS) management implements control loops of different scopes, goals and time scales, at the different levels are:

- *Global level:* control loop-based E2E slice management and orchestration in the multi-domain environment, e.g., cross-slice and cross-domain optimizations.
- *Orchestration domain level:* control loop-based orchestration domain level FCAPS and resources management, e.g., slice admission control, allocation of resources to NSIs.
- *Slice level:* control loop-based slice FCAPS management (embedded in slice template, also in the case of multi-domain slices) with an optional, direct management interface to slice tenant.
- *Node/function level:* control loop-based management of FCAPS of a function or node. It can be considered as modified, intelligent Element Manager (EM), see MANO [3] and provide node or function plug-and-play functionality.

All the listed MAPE-based management subsystems cooperate towards an overall goal. The above hierarchy features the fastest control loops at the local level and increasingly slower ones as the scope widens (e.g., slice-level, tenant-level, etc.). The management information exchange is also decreased while advancing upwards in the hierarchy. Such hierarchization and timescale separation augment the overall system’s stability and provides the ability to implement fast-acting management operations that are fundamental for networks supporting time-critical applications.

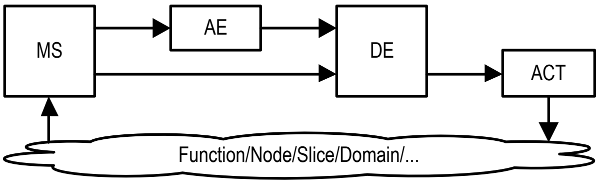


Fig. 1 – MonB5G operations pipeline (generic)

The AI-driven, multiple MAPE loops are used for level-specific, control loop-based optimization that is AI-driven in most cases. Each loop is implemented by the use of a pipeline composed of a Monitoring System (MS), Analytic Engine (AE), Decision Engine (DE) and Actuator (ACT), as presented in Fig. 1. The MS is involved in intelligent monitoring of a function, node, slice, etc. The AE looks into anomalies or specific interesting features in the monitored data from the MS. The DE is responsible for making the reconfiguration decisions based on the data from the MS and AE. The ACT decomposes high-level information from the DE into a set of atomic management operations. The AI algorithms can be used for the implementation of these components, but typically it is assumed that only the AE and DE are AI-driven. For each usage, the internal functions of the AE, DE, ACT can be different. However, it is expected that the MS is common and can be reused by different AEs and DEs.

Distributed AI has additional advantages. It simplifies inter-subsystem interaction and contributes to the minimization of data exchange between the management system components. To that end, the intent-based interfaces can be used for the interchange of high-level information, typically translated into multiple, low-level operations by an AI-driven engine. Moreover, the usage of Key Performance Indicators (KPIs) for the exchange of information between different subsystems of the architecture has been proposed to reduce the monitoring data exchange.

The MonB5G framework, described in the forthcoming sections, is composed of static and dynamic components. The dynamic components are NSIs created on demand, whereas the static components are the permanent entities of the framework, enabling its overall orchestration and management.

3.1 Static components of the framework

The static components of the framework, which constitute its skeleton, are presented in Fig. 2.

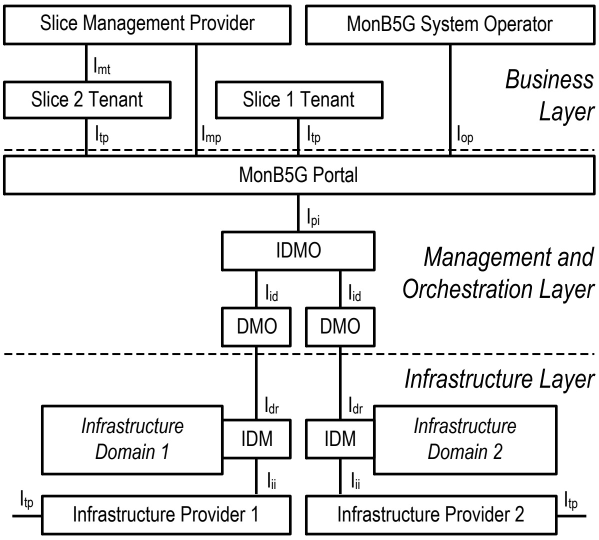


Fig. 2 – Static components and business actors of MonB5G architecture (no NSIs deployed)

Three layers of the framework have been defined:

- *Business Layer* with relevant business actors interacts with the Management and Orchestration Layer via a MonB5G Portal. The actors include the MonB5G System Operator, Slice Tenants and Slice Management Providers. The latter ones may operate or manage multiple slice instances based on the same template in the MaaS model if requested by Slice Tenants. The MonB5G System Operator or the Infrastructure Providers may play the role of Slice Tenants if they need to deploy slices that enhance the functionality of the Management and Orchestration Layer or the Infrastructure Layer respectively. The Business Layer contains no functional components.
- *Management and Orchestration Layer* is at the heart of the whole approach and includes the MonB5G System Portal as well as domain and intra-domain orchestrators. The layer is operated by the MonB5G System Operator.
- *Infrastructure Layer*, which consists of physical and virtual resources with a resource-focused management platform. This layer is operated by Infrastructure Providers.

The static components of the Management and Orchestration Layer are as follows:

- *MonB5G Portal.* The portal (see Fig. 3) is used for requesting NSI lifecycle operations (its deployment, modification, and termination) by Slice Tenants, Slice Management Providers, MonB5G System Operator and Infrastructure Providers. Infrastructure Providers may request the orchestration of infrastructure-oriented management functions, similarly to Slice Tenants requesting the deployment of NSIs. In both cases, the I_{tp} interface, typically web-based, is used. The MonB5G Portal exposes the capabilities of the MonB5G framework (available slice templates, etc.) and partakes in business negotiations during which it interacts with Inter-Domain Manager and Orchestrator (IDMO) via the I_{pi} interface. Afterwards, the I_{pi} interface is used for the LCM of negotiated slices. The MonB5G Portal also provides the I_{mp} interface to the Slice Management Provider to maintain its awareness of new NSIs and to the MonB5G System Operator via the I_{op} management interface.

  The MonB5G Portal is composed of: *Access Management* component, an entity responsible for users' access to MonB5G framework features policy enforcement/management, and access authorization, in cooperation with *MonB5G Subscribers Database* containing the relevant information; *System Health Monitoring*, an entity responsible for providing real-time high-level monitoring data showing the current state of the network for the MonB5G System Operator. In case of critical failures or instabilities of the system, the MonB5G System Operator, based on the accumulated monitoring data, can bring the framework back to stable conditions manually; *IDMO Connector* being responsible for communication with IDMO concerning, i.a., slice LCM-related requests, contract negotiation, and high-level system monitoring data; *Slice LCM API*, the interface enabling the Slice Tenant to make NSI LCM-related requests, including slice templates selection from the Templates Database (also part of the portal), NSI instantiation, NSI termination, etc.

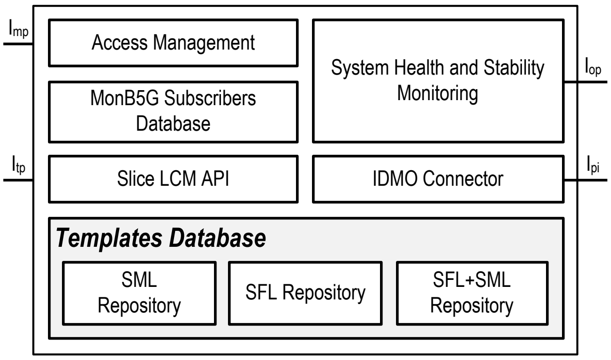


Fig. 3 – Internal components of MonB5G Portal (an example)

- *Inter-Domain Manager and Orchestrator (IDMO).* This entity interacts with instances of Domain Manager and Orchestrator (DMO) via the I_{id} interface (can be considered as an extension of ETSI NFV MANO *Os-Ma-nfvo* interface [3], providing LCM abstractions and exposing data and management capabilities of DMO to IDMO) to coordinate the E2E deployment of NSIs, based on the information from DMOs. Hence, it can be considered as an E2E orchestrator (or umbrella orchestrator [33]). Its role is crucial in slice preparation and deployment phases by negotiating deployment policy with slice requesters (Slice Tenants, Slice Management Providers or Infrastructure Providers). According to the negotiated contract, IDMO modifies the E2E slice template before its deployment. The modification includes slice stitching mechanisms for setting up the E2E NSI and proper adaptation of its management plane (correlation of events and KPIs from different domains in which the slice is deployed). If there are multiple Infrastructure Providers, IDMO may design the E2E slice template split, driven by various factors, e.g., price, performance or energy efficiency, into single-domain templates before the deployment. To that end, IDMO keeps its awareness of all involved infrastructure domains and their resources' status. IDMO also contains a permanent Accounting Database to store the accounting data, as the NSIs themselves and their management parts are temporary. The internal IDMO structure is decomposed into Functional and MonB5G Layers (see Fig. 4 for details of IDMO internal structure). Please note, that IDMO consists of *Template Partitioner* and *Template Configurator*. The Template Partitioner is responsible for partitioning a template if it is to be deployed in multiple domains of the same type. Such partitioning can be motivated by a lack of resources in a single domain for the deployment of the whole template. In the case where domains cover different geographical areas (or for economic reasons), for the sake of security, only part of a template is deployed in each do- main. The decision about partitioning is taken by the IDMO through cooperation with the *Resource Broker*. The Template Configurator is responsible for the initial configuration of NSI parameters and inter-actions between subnetwork NSIs (parts of the E2E NSI that are deployed in a single domain). The modified slice template includes mechanisms added by the IDMO for slice stitching to obtain the E2E slice and proper modification of the E2E slice management plane (correlation of events and KPIs from different domains that are used for slice deployments).

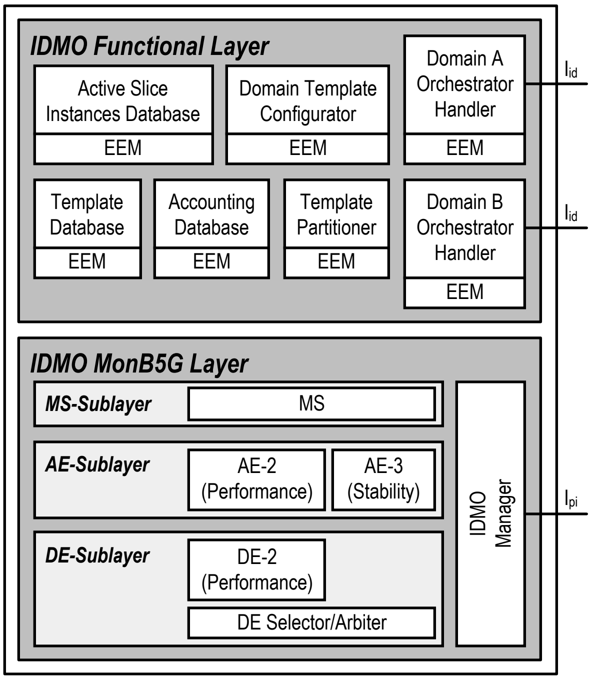


Fig. 4 – Internal components of IDMO (an example)

The AI-driven MonB5G Layer (management of IDMO) uses MS/AE/DE triplets and other components of the management architecture (see further).

- *Domain Manager and Orchestrator (DMO).* Within a Slice Orchestration Domain (SOD), it is responsible for the orchestration of NSIs and the management of resources. In the case of an NFV MANO-orchestrated domain, the DMO can be considered as a combined resource-oriented OSS/BSS and MANO orchestrator. In different technological domains, other orchestrators can be used. It has to be noted that IDMO does not interact directly with the SOD orchestrator, but with the SOD OSS/BSS. The I_{id} interface between the IDMO and DMOs of different orchestration technologies can therefore be defined similarly. The DMO is focused on SOD operations concerning resources allocation to NSIs and the lifecycle of NSIs. It is agnostic to slices, including initial NSI configuration, and it deals with the software dimension of slices only, whereas the runtime management is handled by the management functions embedded in slices. Similar to the IDMO, all DMO operations are AI-driven, so the internal structure of the DMO is also composed of Functional and MonB5G layers as shown in Fig. 5. These operations, as well as the exchange of infrastructure-related data (e.g., about energy consumption), are performed on the infrastructure via the I_{dr} interface and are mostly associated with slice admission, NSI lifecycle management and resource sharing.

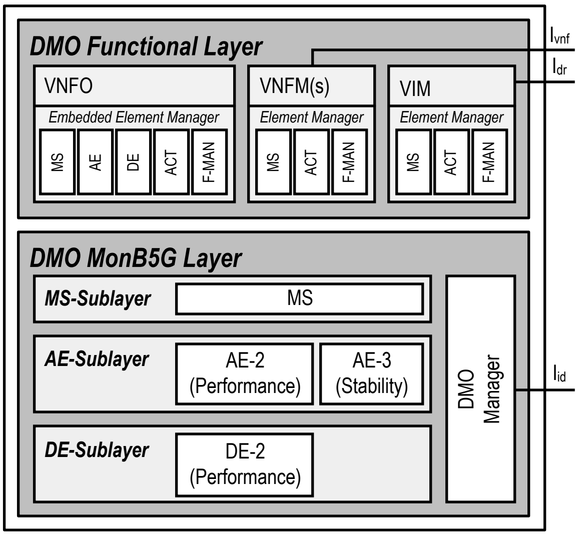


Fig. 5 – Internal components of DMO (an example)

- *Infrastructure Domain Manager (IDM)*. This entity is responsible for infrastructure management (see Fig. 6).

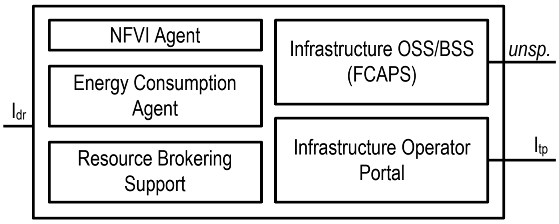

NFVI Agent
Energy Consumption Agent
Resource Brokering Support
Infrastructure OSS/BSS (FCAPS)
Infrastructure Operator Portal
I_{dr}
unsp.
I_{tp}

Fig. 6 – Generic structure of IDM

The MonB5G framework enables programmability of IDM, which can be enhanced by additional resource-oriented functions deployed in a way similar to NSI deployment. For this purpose, the Infrastructure Provider (IDM operator), can use the MonB5G portal to send LCM requests. Such functions are called Infrastructure Orchestrated Management Functionss (IOMFs) (see further).

3.2 Dynamic components of the framework

The dynamic components of the architecture are constituted by NSIs. In the MonB5G concept the term "slice" has been slightly redefined, as it describes not only "network slices" *per se*, i.e., E2E communication networks, but also any set of interconnected virtual functions serving a specific goal. The MonB5G generic slice is composed of the Slice MonB5G Layer (SML) and Slice Functional Layer (SFL) parts. The SFL part, built of virtual functions, provides slice "core" (e.g., communication service-related) functionality, whereas the SML part, also built using virtual functions, provides SFL management. Three options of SFL and SML integration are possible:

- In most cases, the SFL and SML create a single slice template, i.e, the ISM approach is implemented. It leads to self-managed slices, with reduced information exchange between NSIs and external management components of the architecture.
- A single SML can be used for the management of multiple SFL instances of the same type (SML is not SFL-agnostic in general). In fact, this is the implementation of MaaS, which is usually motivated by business requirements and reduction of the overall footprint of several NSIs of the same type.
- The SFL part of a slice or part of it can be shared among multiple SFLs. The shared functions available in SOD ("shared VNFs"), called Dynamically Shared Functions (DSFs), which are implemented within the SFL part as the Physical Network Function (PNF)/VNF or Cloud-native Network Function (CNF), may be used by all or some NSIs. This way, both the deployed NSIs footprint and deployment time are reduced. DSFs are grouped into a slice to ease their management by the DMO and implement in this manner the Platform as a Service (PaaS) model.

If the E2E NSI spans multiple orchestration domains, an entity responsible for the integration of SFLs of all domains is needed. To that end, an Inter-Domain Slice Manager (IDSM) entity is added. It interacts with SMLs using preferably intents and KPIs. Moreover, it provides an interface to the Slice Operator (Slice Tenant).

A generic, internal structure of a self-managed, single domain slice (SML/SFL) is presented in Fig. 7. The SFL part of the MonB5G slice template consists of AI-driven EMs, called Embedded Element Managers (EEMs). They include functions for monitoring (Monitoring System Function (MS-F)), anomaly detection (Analytic Engine Function (AE-F)), decision-making (Decision Engine Function (DE-F)) and actuation (Actuator Function (ACT-F)) at the node or function level. Legacy equipment with EMs [3] can also be used. To reduce the outside exchange of monitoring data, each EEM is involved in some local closed-loop management processes. The high-level DE output (e.g., the action space of a deep reinforcement learning algorithm) is to be translated by ACT-F into a set of low-level primitives compliant with specific API definitions of the controlled functions. EEMs and EMs indeed provide a link between the SFL and SML parts of a slice. Both the EM and EEM are required to contain a Management Function (MAN-F), which is responsible for the configuration of other components of the EEM and can be used for node/function manual management, if necessary.

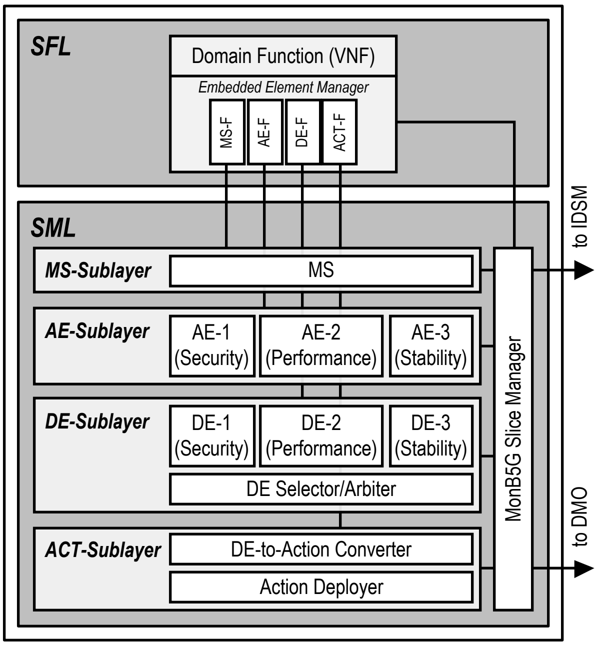


Fig. 7 – Generic structure of the SML/SFL slice

The SML part is logically split into MS, AE, DE, and ACT sublayers. An example processing of information of these sublayers is as follows:

- The Monitoring System Sublayer (MS-S) provides generic, reusable and programmable monitoring of certain granularities for all AEs.
- The monitoring data from the MS is analysed by multiple AEs of the Analytic Engine Sublayer (AE-S). Each AE has a specific goal, e.g., fault detection and its root-cause identification, performance or security attack identification.
- Each DE of the Decision Engine Sublayer (DE-S) is linked with an appropriate AE. The entity on the basis of analysed data and AE output may propose network reconfiguration to solve detected problems.
- The DEs' decisions are transferred to the ACT and the respective ACT converts the DE decision into a set of elementary actions. The ACT monitors their execution and provides feedback information to the DE.

The control loop-based management system may generally have multiple goals which justifies multiple AEs and

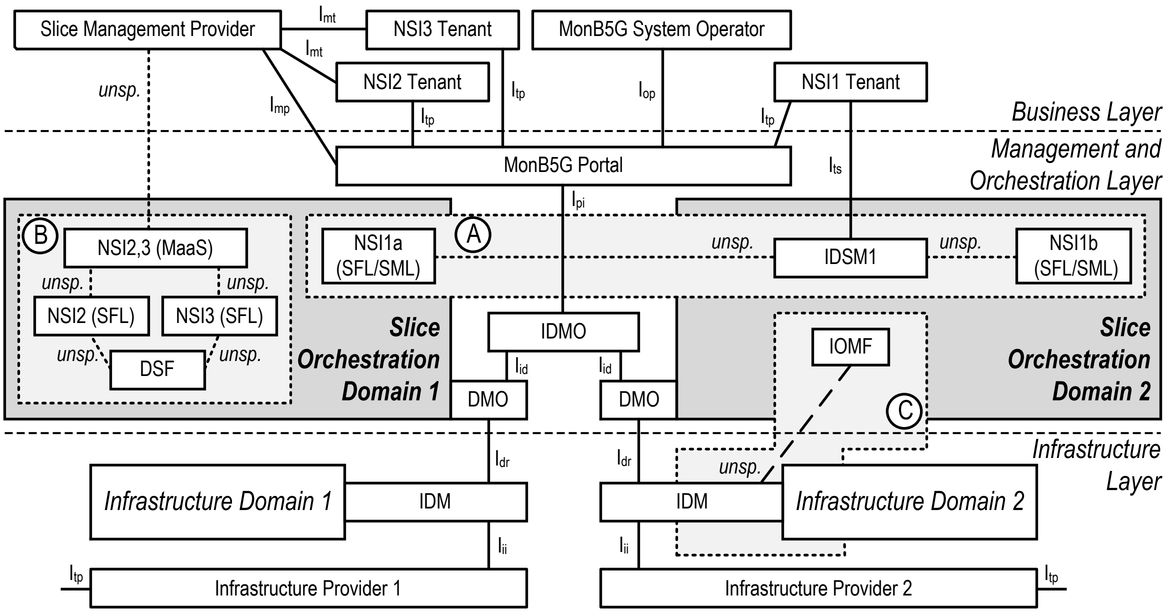


Fig. 8 – MonB5G system with different options of slice deployment: A – multi-domain slice; B – slices using MaaS and shared functions, C – orchestrated management functions of the Infrastructure Provider

DEs as a part of the SML. This inevitably leads to conflicts between DEs' outputs. In order to solve this problem, the DE Selector/Arbiter component is present in the DE-S. It is involved in the observation of the system's stability and counteracting the ping-pong effect or its chaotic behaviour (random-like reconfigurations).

The DEs of the SML have both management and orchestration capabilities. They may change SFL configuration parameters and also send new function (VNF) orchestration requests using a direct connection between each SML and DMO as well as decide about SML and SFL functionality updates. In contrast to the reactive resource scaling only by the NFV MANO orchestrator, the SML may also proactively request an update of resource allocation or scaling, based on the number of slice users' Quality of Experience (QoE) changes or spatial distribution of slice usage.

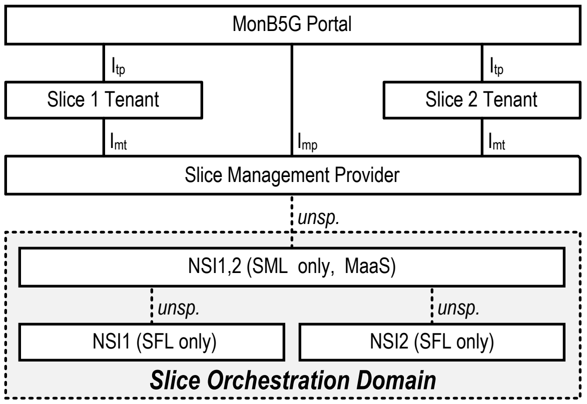


Fig. 9 – MaaS usage in MonB5G framework (example)

The already discussed options of slices' deployment are shown in Fig. 8:

- Option A – deployment of a self-managed multi-domain slice supported by IDSM; a single domain variant of such an option is not shown as trivial.
- Option B – deployment of slices with shared SFL functions (i.e., DSF) and the use of MaaS operated by Slice Management Provider (see Fig. 9).
- Option C – deployment of slice oriented to infrastructure management, i.e., the IOMF slice.

The dynamic enhancement of the IDM functionalities supporting the infrastructure management is provided by the MonB5G framework by orchestration of additional management services, called IOMFs. They can be orchestrated by the DMO upon the request of the Infrastructure Provider via the MonB5G Portal. The IOMFs role is to cooperate with the IDM to achieve its goal (prediction of resource consumption, energy saving, etc.).

4. MONB5G SYSTEM AUTOMATION

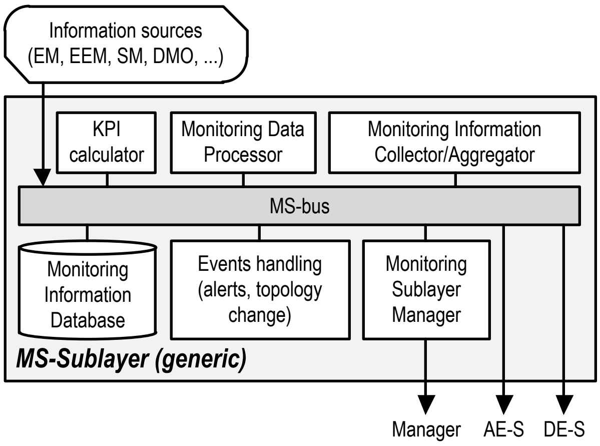
The MonB5G framework provides several features to facilitate the incorporation of NSIs management automation. Hereby, the main components supporting the cognitive capabilities of the system including monitoring data acquisition, KPI calculation, analytics and decision-making are described. The adopted approaches are presented in the context of FCAPS and Service Level Agreement (SLA) enforcement.

4.1 Monitoring data collection and processing

The MS-S is responsible for the provisioning of generic and reusable monitoring information to AEs, DEs, and other MonB5G framework entities via the MonB5G Slice Manager (see figures 7 and 10). The monitoring information from the SFL may be consumed by each SML entity.

The MS of the SML is responsible for the collection, aggregation, filtering and interpolation of the monitoring data related to the NSI, including resources monitoring. It is also in charge of collecting information about faults, topology changes, etc. as well as the calculation of NSI KPIs on the basis of acquired metrics. Time granularity and degrees of data aggregation can be tuned depending on the definition of the data collection and preprocessing strategy to enable meeting the heterogeneous requirements of each SML entity. The MS of the SML collects the monitoring information from the EM and EEM and, in a generic case, is composed of the following blocks:

- Monitoring Information Collector/Aggregator: an entity interacting with sources of information (for example EMs/EEMs of the SFL).
- Monitoring Information Database: storage of monitoring data in raw and/or preprocessed format.
- Monitoring Information Processor: an entity responsible for filtering, interpolation and prediction of the monitoring data.
- Monitoring Sublayer Manager: an entity allowing remote configuration of MS operations.



Information sources
(EM, EEM, SM, DMO, ...)
KPI calculator
Monitoring Data Processor
Monitoring Information Collector/Aggregator
MS-bus
Monitoring Information Database
Events handling (alerts, topology change)
Monitoring Sublayer Manager
MS-Sublayer (generic)
Manager
AE-S
DE-S

Fig. 10 – Generic structure of MS Sublayer (example)

The output of the MS is accessible to other components of the SML via a dedicated message bus using a publish/subscribe paradigm. In general, the MS has to interact with VNF-specific EEMs of different technological domains. However, several MS operations and queries are generic, e.g., those related to computing resource availability and consumption. Therefore, many of the internal components of the MS can be reused for multiple slice templates. The definition of protocols for efficient communication between the SML and MS is out of this paper's scope.

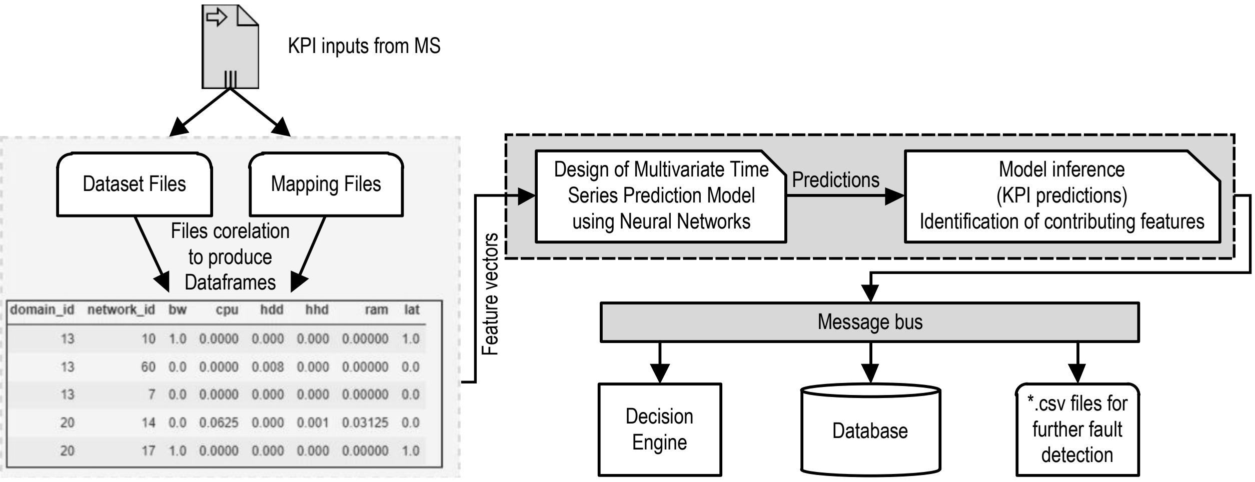
4.2 Analytic Engines

For the MonB5G AE entities, we investigate and extend diverse distributed ML and representation learning algorithms to fulfil the requirements of 5G networks and beyond and enable the decomposition of a traditionally centralized AE into interconnected entities that can be deployed in a distributed manner in RAN, edge and cloud domains. This allows highly intelligent, accurate, and scalable reactions to non-stationary network conditions, new traffic patterns and evolving slice characteristics. The distribution of the MonB5G AE includes multiple levels, such as learning concise representations of local data to reduce the amount of information exchanged for management purposes, as well as boosting slice-level KPI prediction and the corresponding AI models with different native data.

4.2.1 *KPI calculation and prediction*

To support automated and proactive decisions at the slice level, AE components can provide predictions of NSI KPIs. E2E slice-level KPIs have been defined [34] and include, i.a., upstream/downstream throughput for NSI, average E2E uplink/downlink delay, virtualised resource utilization per NSI, etc. The collection of KPIs is supported by the MS, as explained in Section 4.1. Despite being implemented in the legacy systems, slice-level KPI prediction in a 5G system requires a specific approach as the number of NSIs, as well as the amount of data collected, can be large, implying the need for scalability. To address this, we designed a Tensorflow-based multivariate time series prediction algorithm to investigate a variant of the Recurrent Neural Network (RNN), i.e., Gated Recurrent Units (GRUs). GRUs were shown to perform well with a substantial reduction in training times when compared to Long Short-Term Memory (LSTM) models due to faster convergence in training. The model takes the multivariate input values (multiple features at the same time) as feature vectors and the target NSI KPI to predict as the output.

As shown in Fig. 11, the input data sets are generated for each NSI and include the Dataset Files (DSs) used by the AE and Mapping Files (MFs) that link the DS files with the network elements and their functions. These files link the items of the State 2D array of the data source files to the network elements and their function and are intended to be used as a dataset decoder for further analysis or exploitation by the AE. As shown in Fig. 11, a useful addition to the multivariate time series KPI prediction is a model-agnostic explainability method such as DeepLIFT or SHAP that can give more insight (interpretable models) into which input KPIs affects the target NSI KPI the most (this information can be later exploited by the DE). The slice-level KPI predictions results from the AI-driven AE are then synchronized and sent via a message bus for further processing (storage in a database, used by another AE or DE, etc.). Moreover, to further scale the AEs towards managing a massive number of coexisting NSIs, we also explore the federated learning techniques in KPI prediction for downstream tasks, such as decreasing


KPI inputs from MS
Dataset Files
Mapping Files
Files corelation to produce Dataframes
Feature vectors
Design of Multivariate Time Series Prediction Model using Neural Networks
Predictions
Model inference (KPI predictions) Identification of contributing features
Message bus
Decision Engine
Database
*.csv files for further fault detection

domain_id	network_id	bw	cpu	hdd	hhd	ram	lat
13	10	1.0	0.0000	0.000	0.000	0.00000	1.0
13	60	0.0	0.0000	0.008	0.000	0.00000	0.0
13	7	0.0	0.0000	0.000	0.000	0.00000	0.0
20	14	0.0	0.0625	0.000	0.001	0.03125	0.0
20	17	1.0	0.0000	0.000	0.000	0.00000	1.0

Fig. 11 – Analytics Engine flow

SLA violation. This AE function enhances the federated learning method and introduces a set of well-designed statistical constraints with focus on distributed network management. The novel function facilitates network slicing decentralized resource allocation while guaranteeing very low SLA violations. In addition, novel Distributed Deep Neural Network (DDNN) methods are tailored to implement a decentralized AE towards connected intelligence. The AE enables distributed inference across different technical domains, which fulfils local analysis for simple cases, and pushes the extracted low dimensional data to a higher layer of the management hierarchy when the local low-resource AE is less confident of its predictions. The DDNN-based AE also accelerates the feedback processes.

Techniques for representation learning can also be enhanced in order for the KPI predictors to become aware not only of the specific temporal behavioural patterns of the features it predicts, but also to increase their awareness of the *context* of the performed predictions. A way to make a KPI predictor to be context-aware is to embed knowledge, which can be expressed through analytical formulations of the context at hand, into the predictor's training process. This context could be defined as knowledge related to resource orchestration, admission control, fault management, security, etc. Recapping Fig. 11, the output from the AE (calculated/predicted KPIs) is expected to be used by DEs, which perform state-altering decisions on the network. By enhancing the AEs to be context-aware and exploiting the distributed characteristic of the network through federated techniques, the DEs are able to make their decisions using *predicted* future states. This also enhances the DEs' operation by enabling their decisions to be statistically more proactive, rather than reactive.

4.2.2 AI-driven detection of anomalies including security use cases

Fault detection and root cause analysis have always been an integral part of telecom network management. The rising volume of 5G network data [35] has contributed to increased momentum in AI/ML-based network analysis and troubleshooting research. Available network resources and services are impacted by changing patterns of use and deployment of NSIs. To ensure the infrastructures provide a high level of robustness to customers, rule-based systems, based on domain knowledge, have traditionally been implemented to analyse and detect anomalies in the network. While these systems have their place in the network management area, they can be insufficient in a highly dynamic environment, where NSIs are automatically deployed and changed to maintain SLAs.

The importance of intelligent monitoring and root cause analysis is imperative for network operators and the area of intensified research. The sophistication of Deep Neural Network (DNN) algorithms (often used for this purpose) has recently increased with the ability to process massive network management data sets while extracting useful features to quickly identify anomalies (e.g. for different NSI KPIs).

Furthermore, the explainability part of our AE will provide operators with actionable insights, enabling a deeper investigation of the influencing factors. This will improve automated troubleshooting of anomalies across multiple components of the network, thus reducing operational costs and resources usage. Hereby, an interesting exploration for the anomaly detection AEs is conducted, which aims to use explainable time series forecasting for cause analysis of NSI performance degradation. We extract high order features from telemetry time series monitoring the running of the related network components, learn dynamic patterns between the E2E NSI KPI and the telemetries, and finally craft the impact indicators of the telemetries based on the averaged gradients (also known as saliency map [36]) of the sampled time series models to identify the causes of the performance degradation perceptively. We also consider the influence of the model uncertainty to make the inferred explanations more robust against the randomness introduced by the prediction model itself. The AE function facilitates detection of the root causes of NSI performance degradation for proactive mitigation of SLA violations.

4.3 Distributed AI for network slices and infrastructure optimization

An important 5G task that has been recently addressed with data-driven methods is NSI resource allocation with DNNs. For example, the authors of DeepCog [37] have used a DNN to directly predict the amount of NSI resources needed at a data centre/cloud serving assigned Base Stations (BSs) to avoid both underprovisioning (resulting in a costly SLA violation) and overprovisioning (wasting valuable resources to be used by another NSI/tenant). Similarly, the authors of Context-Aware Traffic Prediction (CATP) [38] have also used a DNN, but their approach predicts resource utilization considering the costs and utilities related to resource orchestration, provisioning and reconfiguration in a more generalized way. As a result, this control/decision problem can be tackled with popular DNN architectures (e.g., convolutional networks or multilayer perceptrons), by training an objective with appropriately tuned under and overprovision components. CATP considers resource reconfiguration costs; DeepCog has been extended to include them.

While these approaches are promising, the standard assumption of a *centralized* implementation of the DNN architecture faces challenges, when used to control key network functions. *First,* unlike the use of DNNs for some application layer tasks (e.g., image classification on a phone) that can be "lazily" offloaded to a central computational cloud, the use of DNNs for controlling 5G edge resources (e.g., allocation of RAN resource blocks among tenants, CPU allocation for Cloud-RAN processing) requires significantly lower latency; data transfer to a central DNN, centralized decision-making, and actuation of the desired edge components, might violate these requirements. *Second,* constantly sending raw monitored data over possibly already congested links towards a DNN architecture lying deep in the Core Network (CN), or even outside, has a prohibitive network footprint.

To that end, a key focus of MonB5G is to propose, train and study a *distributed* DNN architectures for data-driven resource allocation problem. Some key contributions of our work include:

- Proposing an appropriate distribution of the layers of a 3D Convolutional Neural Network (CNN) architecture, between an edge cloud and a core/remote cloud, and investigation on joint training to provide accurate local and remote NSI resource scaling decisions.
- At runtime, the local layers will communicate with the remote layers and delegate the decision there, only if there is limited confidence in the local decision. We propose a novel way to evaluate local *confidence*, based on Bayesian methods, using dropout during the forward pass. This is in contrast with the standard dropout methods for regularization or the entropy-based uncertainty of recent distributed DNN works in the field of machine learning.
- Using real data, our results suggest the distributed architectures are able to resolve up to 80% of decisions locally, while the uncertainty measure is able to pick out the correct remaining decisions that would benefit from a forward pass through the additional remote layers; the layer distribution and offload mechanisms, in conjunction, can always achieve this large overhead/latency reduction with minimal objective degradation, and sometimes even improve the objective, compared to a fully centralized architecture with all layers involved in all decisions.

4.4 AI for network slicing security

The MonB5G architecture focuses on security inside a single domain and inter-domain security of the domains that compose NSIs, employing the three key components: MS, AE, and DE. These components offer an automatic and intelligent closed-loop to detect and mitigate security threats and attacks in real time. The real time data collected from the VNF is filtered by the MS and exposed to the AE that analyses and detects abnormal behaviour of the NSI components in the heat of the moment, which makes DE intervention faster and in a timely manner before the threat spreads.

Fig. 12 illustrates a simplified view of the system architecture representing the MonB5G elements and their interactions with the components of a Massive Machine Type Communications (mMTC) NSI. In this scenario, we assume an mMTC NSI with a high number of devices connected through different next generation NodeBs (gNBs). The composite NSI consists of a dedicated NSI running tenant applications and a shared Network Slice Subnet Instance (NSSI) managed by the Infrastructure Provider. The NSSI is permanent and composed of gNB and CN running on top of a virtualised infrastructure. The gNBs are considered as a shared PNF, while the CN elements are VNFs shared among the NSIs. The DMO is informed about each enforced DE decision, which in this case covers the CN and RAN.

In the following paragraphs, we will detail the role of each element in the system.

MS: We assume that the mMTC devices are controlled by an attacker that launches Distributed Denial of Service (DDoS) attacks on the Access and Mobility Management Function (AMF) [39] of the CN by generating a high number of attach and detach requests. Each request generates

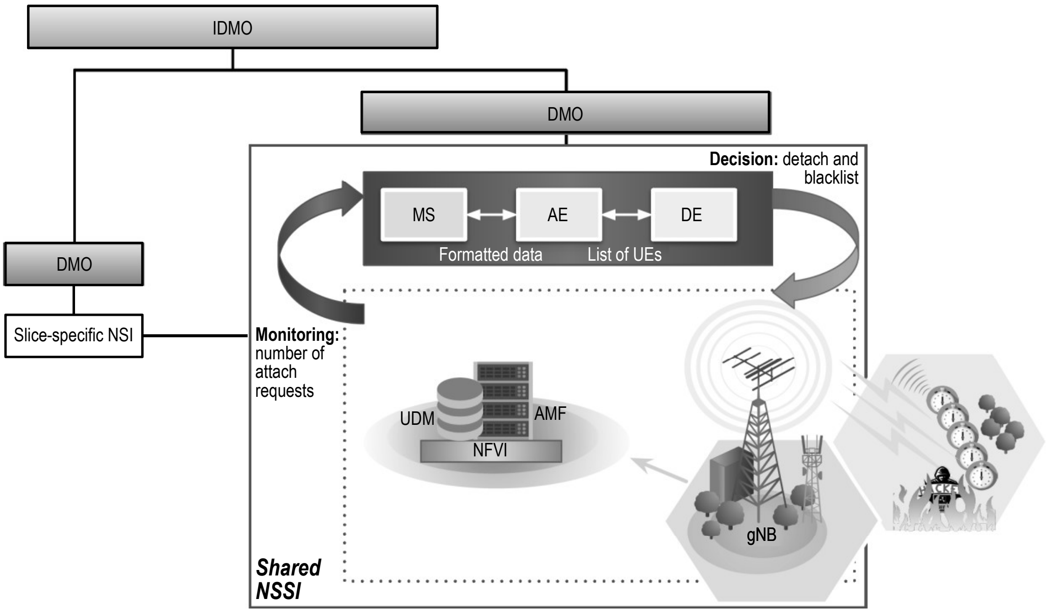


Fig. 12 – Intelligent closed loop for detection and mitigation of in-slice attacks

several control plane messages, and if the number of requests is very high, the AMF may be overloaded, and the provided service is disturbed. Therefore, the number of attach and detach requests and their rate should be continuously monitored and collected by the MS. Such information is provided by the EM of the AMF.

AE: To detect attacks in an mMTC NSI, it is important to know the model of the mMTC traffic, which is typically dependent on the application type. In that context three traffic types can be identified. The first one is "event-based", where the mMTC devices connect if an event happens (fire-detection, earthquake, etc.) and send a small amount of traffic (few packets). The second one is "trigger-based", where a remote server triggers a connection to the device to gather data (temperature, humidity measurements, etc.). The third one is "periodic", where the mMTC devices connect periodically to send data (temperature, humidity, etc.). While types two and three can be easily predicted using information from the NSI tenant, the first type is very difficult to anticipate. Knowing the mMTC traffic model, different options can be considered for the attack detection by the AE. One of the solutions is the usage of a neural network (RNN, LSTM, or an association of both), which can be trained using the normal traffic generated by mMTC devices of the NSI (either synthetic one, generated through one of the models specified by 3GPP or a real one). Then, the trained algorithm can run as an Intrusion Detection System (IDS), and by analysing the attach requests classify the traffic as normal or abnormal.

DE: The DE analyses the alerts, the attach requests obtained from the AE, makes their classification and decides which User Equipments (UEs) to disconnect or ban UEs from network access. We have designed two versions of the DE. The first one performs blacklisting a device if its predicted value is higher than a configurable threshold. The second one analyses the whole event and checks if a significant set of devices had higher than usual detection rates. This is to achieve that the UEs are classified into three groups based on their score. If the number of devices in the largest group is higher than the two other groups, the DE will add their Subscription Permanent Identifier (SUPI) values to a table of blacklisted devices and disconnect them from the network. Otherwise, the DE will add the SUPI values to a table of non-trusted devices. The DE can request from the AMF to detach the concerned list of International Mobile Subscriber Identities (IMSIs) involved in the DDoS attacks communicated by the AE.

4.5 AI for solving multi-domain issues

The inter-slice DEs have a view of the network operations and its performance, across multiple NSIs and domains. At large, their role is to coordinate the VNF processes across all technological domains inside the physical network of the operator. The problem is highly complex, as the decisions taken in one domain can affect the effects of decisions taken in the other domains. For example, overprovisioning, can waste resources in the domain, if the rest of the domains that this NSI spans are not also provisioned in harmony to ensure that the E2E KPIs are satisfactory.

In the MonB5G project, we heavily leverage the dynamic, AI-based control methods to orchestrate resource allocation and VNF placement across domains. Reinforcement learning solutions can gradually learn to optimally place the various VNFs comprising an NSI's VNF chain across domains, without requiring any *a priori* knowledge regarding their traffic demands. Nevertheless, the problem of

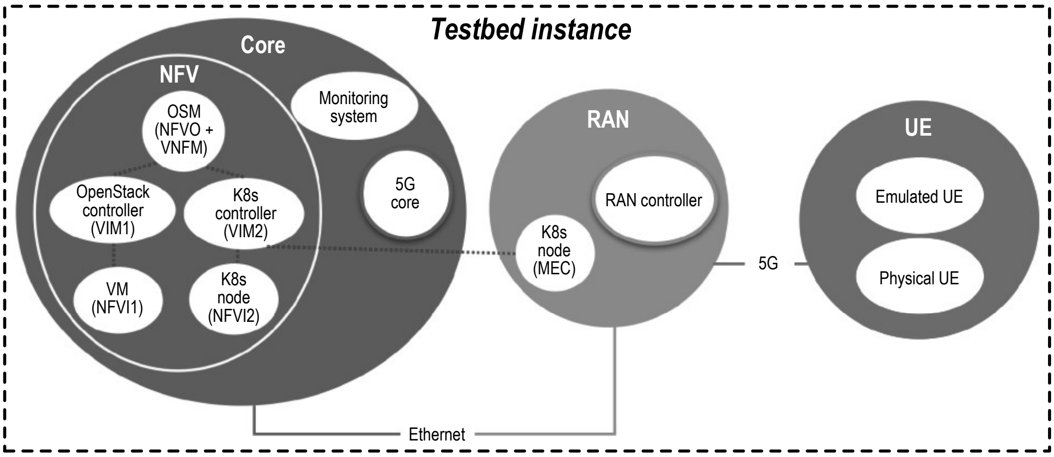


Fig. 13 – Emulated virtualised E2E 5G network used in the second phase of the implementation.

placing multiple VNFs, across multiple nodes, and in different domains, has restrictive complexity when it comes to vanilla RL. To that end, in MonB5G, we have two main paths to tackle issues arising in complex cross-domain scenarios: state-of-the-art DRL methods and the multi-agent RL solutions. Both approaches can efficiently cope with the large state spaces, however, the latter also radically improves the solutions' convergence. Furthermore, they are a natural match for cross-domain orchestration, where each domain might have its own agent controlling its own actuators, nodes, and local VNFs, yet the agents communicate (possibly in a privacy-preserving manner) to satisfy E2E KPIs and related SLA depending on the performance of the action in each domain.

5. IMPLEMENTATION REMARKS

The implementation roadmap, framework, and technologies of the MonB5G system are explained below.

5.1 Implementation roadmap

The MonB5G architecture described in Section 3 is being implemented according to the roadmap consisting of three phases described below.

In the first development phase, the functional components of the architecture are implemented in simulated environments. Referring back to figures 4 and 7, the design and implementation of the MS are quite generic, distributed MS capable of collecting monitoring data from different sources, while the AE and DE need to be tailored to specific requirements.

In the second implementation phase, the integration of different components and performance evaluation of the emulated and virtualised E2E 5G networks, depicted in Fig. 13, is done. The environment, not only emulates an E2E 5G network including the UE, RAN and CN but also provides an NFV ecosystem composed of:

- an NFV Orchestrator (NFVO) and a Virtual Network Function Manager (VNFM), both jointly implemented by the OSM;
- two Virtualised Infrastructure Managers (VIMs) for Virtual Machine (VM)-based and container-based virtual infrastructure implemented by OpenStack [40] and Kubernetes [41] respectively.

The instances of the AE and DE for each technological domain are deployed in this environment as VNFs/CNFs of the SML and integrated with the MS to provide a functional instance of the MonB5G architecture. The performance of the system is then evaluated using emulated traffic.

The final phase of the roadmap lies on the integration of the components and their deployment in the real network. The deployment of MonB5G in the real network is gradual. Hence, before the full migration to MonB5G architecture, the management of the network will be hybrid, which requires some kind of collaboration and cooperation between the traditional centralized OSS/BSS and MonB5G. This can be implemented via the I_{op} management interface shown in Fig. 2 through which the OSS/BSS can request management functions from MonB5G.

5.2 Implementation framework

The previous sections explained that the MonB5G architectural framework is designed for automated, scalable, and AI-driven orchestration of MAPE-based NSIs.

The implementation framework of MonB5G, mainly the SML, is based on the concepts proposed in [42], in particular the definition of PaaS. Using PaaS brings benefits such as fast reconfigurability, proximity to SFL components, reduced resource footprint, etc. To isolate and secure NSIs, the 3GPP recommendations [43] can be reused. For this purpose, Network Function (NF) discovery and registration must be authorized and should support confidentiality, integrity, and replay protection of data.

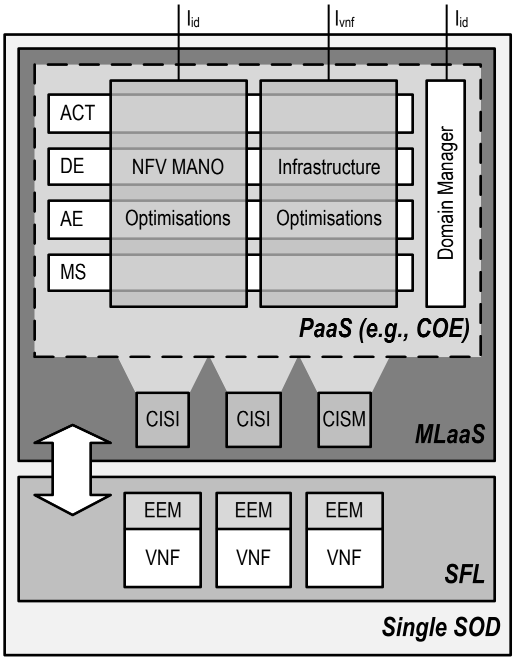


Fig. 14 – Implementation of the MonB5G slice (example)

By applying this concept to the SML of a slice, in cases where a set of various SFL need to be managed, the SML will act as the MonB5G Layer as a Service (MLaaS) and contribute to the footprint in terms of resource consumption and cost. In addition, the PaaS instances offered to MonB5G SFLs (as VNF Common/Dedicated Service) are referred to as a DSF. Both concepts use the same implementation mechanisms and can be used concurrently for the same slice instance, although the DSF and MaaS are SFL template-specific in most cases. Since DSFs may not be operated exclusively by a single Slice Tenant or DMO operator, it is necessary to use a Slice Management Provider (SMP) as an operator. From the management point of view, the DSF and MLaaS are considered as slices. Therefore, they can be considered as vertical stitching of slices, where instances of MLaaS and DSF service can be deployed by the IDMO.

The ETSI NFV framework [42] provides support for container-based service deployment and orchestration. In general, most of the mechanisms described there are applicable to the aforementioned PaaS use cases.

A simplified version of a slice according to MonB5G architecture from the implementation point of view is shown in Fig. 14, which includes an SFL and MaaS. The former is composed of tenants' VNF and corresponding EEMs/EMs that allow MaaS components to gather the required metrics. On the other hand, MaaS is composed of a set of VNFs providing a Container Infrastructure System (CIS), which is Container Infrastructure Service Instance (CISI) and Container Infrastructure Service Managers (CISM). Moreover, it includes PaaS itself as a Container Orchestration Engine (COE) to effectively act as a runtime environment for MonB5G administrative elements and components. MaaS management services are therefore conceived as cloud-native applications build-out of interaction among several MonB5G components of SML, which is exemplified by NFV MANO and infrastructure optimizations.

As mentioned earlier, a generic distributed MS has been designed and implemented in MonB5G, as shown in Fig. 15. MS is capable of collecting data from the SFL via interactions with the EEM, providing an API for the AE to control and consume monitoring data, transforming data by adding semantic and contextual information, e.g., timestamp, data source, etc., and storing the monitored data in persistent storage for interpolation and extrapolation of future requests.

5.3 Implementation technologies

Various state-of-the-art open-source technologies are used in different domains to implement the MonB5G architecture. The infrastructure virtualisation solutions are OpenStack [40], Docker [44], and Kubernetes [41] for both local resources and remote resources in the public cloud. For the Management and Orchestration Layer in the CN and Multi-access Edge Computing (MEC), ETSI OSM is used for VNF management and orchestration. In addition, OpenDaylight, an open-source SDN controller, controls the virtual links between different nodes using Open vSwitches. The components of MonB5G (SML), e.g., the MS, AE and DE, are deployed as CNFs on Kubernetes, which acts as a PaaS, aligned with the ETSI NFV approach [42]. FlexRAN [45] over OpenAirInterface (OAI) [46] is used to control and slice the RAN domain. The main components of the Business Layer are the web interfaces and APIs with a database to store authentication credentials. There are many open source tools to implement these functionalities, e.g., Django [47]/Angular [48] and Swagger [49] used in the MonB5G are among the most popular.

As mentioned earlier, the MonB5G framework relies heavily on the data-driven AI, especially the AE and DE components. The NetData [50] is used to implement the MS component to collect and expose resource-related and network telemetry data from deployed VNF instances and PNFs. Moreover, Prometheus [51] is used for automatic scraping of custom metrics from SML components, leveraging information from multiple custom agents to derive slice-specific KPIs. The implementations of AI-based components, i.e., AEs and DEs utilize ML frameworks such as Python TensorFlow [52], PyTorch [53] and OpenAI Gym [54]. The communication between MS, AE and DE sublayers is based on Kafka [55] as the publish/subscribe platform.

6. SUMMARY AND CONCLUSIONS

In this paper, the MonB5G framework for distributed, AI-driven orchestration and management of network

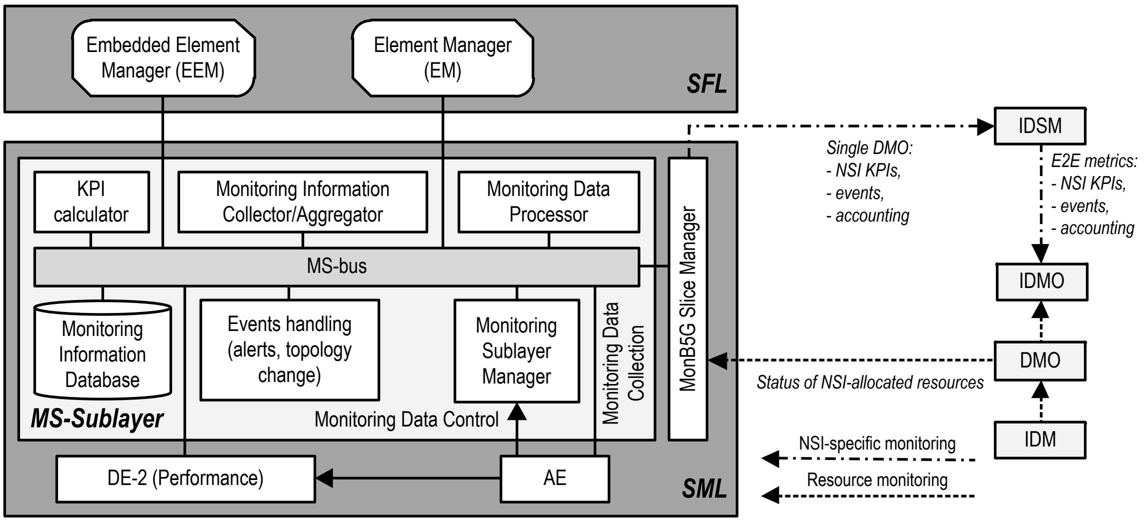


Fig. 15 – Monitoring roles and information flow in MonB5G architecture.

slices has been presented. The proposed solution is characterized by high scalability of management and orchestration that has been achieved by distribution of management operations and by the use of AI. The adoption of multilevel data processing allows for handling a large amount of data locally and transferring abstracted monitoring information to the components being higher in the system's hierarchy. Such an approach provides improved openness of the system and configurability of the management plane on a per slice basis. Altogether, the framework's characteristics contribute to its genericity, thus facilitating the deployment of any slice types and their thorough configuration. The high degree of distribution of the MonB5G framework also provides the flexibility in the decision-making process. Due to the implementation of slice management as a part of the slice, the slice reconfigurations can be handled in near real time thanks to local analysis and processing of the monitoring events. Moreover, minimizing the management information exchange between components in comparison to centralized approaches consequently increases the overall network scalability. The MonB5G solution strongly supports the adoption of AI techniques for automation of the management and orchestration processes. Nonetheless, the architecture itself is agnostic to AI and does not assume the implementation of the proposed methods, i.e. each component or algorithm can be adjusted to suit the requirements of the specific use case. The framework proposes to use predicted data (including KPIs), which allows for proactive management operations for SLA fulfilment or faults handling. Due to such a feature, the proposed approach can significantly contribute to high service continuity. In addition to the presentation of the framework itself, the paper also provided an example description of its implementation including open source tools that can be used.

7. ACKNOWLEDGMENT

This work has been supported by the EU Horizon 2020 project MonB5G (Grant Agreement No. 871780). The views and opinions are those of the authors and do not reflect the official position of their companies.

REFERENCES

[1] IBM. *An architectural blueprint for autonomic computing*. Autonomic Computing White Paper, 3rd edition. IBM Corporation, 2006. URL: https://www-03.ibm.com/autonomic/pdfs/AC%20Blueprint%20White%20Paper%20V7.pdf.

[2] NGMN Alliance. *NGMN 5G White Paper*. White Paper v1.0. Next Generation Mobile Networks, Jan. 2015.

[3] ETSI. *Network Functions Virtualisation (NFV); Management and Orchestration*. Standard ETSI GS NFV-MAN 001 V1.1.1. European Telecommunications Standards Institute, Dec. 2014.

[4] ETSI. *Experiential Networked Intelligence (ENI)*. [Website]. Accessed: Oct. 10, 2022. URL: https://www.etsi.org/committee/1423-eni.

[5] ETSI. *Zero touch network & Service Management (ZSM)*. [Website]. Accessed: Feb. 03, 2022. URL: https://www.etsi.org/committee/zsm.

[6] O-RAN Alliance. *O-RAN Working Group 2: AI/ML workflow description and requirements*. Report O-RAN.WG2.AIML-v01.01. Dec. 2020.

[7] The Linux Foundation. *Open Network Automation Platform*. [Website]. Accessed: Oct. 10, 2022. URL: https://www.onap.org/.

[8] ETSI. *Open Source MANO*. [Website]. Accessed: Oct. 10, 2022. URL: https://osm.etsi.org/.

[9] 5G-PPP. *View on 5G Architecture*. White Paper v4.0. The 5G Infrastructure Public Private Partnership, Oct. 2021. URL: https://5g-ppp.eu/wp-content/uploads/2021/11/Architecture-WP-V4.0-final.pdf.

[10] *5G!Pagoda: Federating Japanese and European 5G testbeds to explore relevant standards and align views on 5G mobile network infrastructure supporting dynamic creation and management of network slices for different mobile services*. [Website]. Accessed: Oct. 10, 2022. URL: https://5g-pagoda.aalto.fi/.

[11] Sławomir Kukliński and Lechosław Tomaszewski. "DASMO: A scalable approach to network slices management and orchestration". In: *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*. 2018, pp. 1–6. DOI: 10.1109/NOMS.2018.8406279.

[12] *5G-Victori: Vertical demos over common large scale field trials for rail, energy and media industries*. [Website]. Accessed: Oct. 10, 2022. URL: https://www.5g-victori-project.eu/.

[13] *5G-MoNArch: 5G Mobile Network Architecture for diverse services, use cases, and applications in 5G and beyond*. [Website]. Accessed: Aug. 31, 2022. URL: https://5g-monarch.eu/.

[14] *5G-CLARITY: Beyond 5G multi-tenant private networks integrating cellular, Wi-Fi, and LiFi, powered by artificial intelligence and intent based policy*. [Website]. Accessed: Oct. 10, 2022. URL: https://www.5gclarity.com.

[15] *SELFNET: A framework for self-organized network management in virtualized and software defined networks*. [Website]. Accessed: Oct. 10, 2022. URL: https://www.eurescom.eu/archive/SELFNET/about-selfnet/index.html.

[16] *SLICENET: End-to-end cognitive network slicing and slice management framework in virtualised multi-domain, multi-tenant 5G networks*. [Website]. Accessed: Oct. 10, 2022. URL: https://slicenet.eu/.

[17] *5GZORRO: Zero-touch service, network and security management in multi-stakeholder environments*. [Website]. Accessed: Oct. 10, 2022. URL: https://www.5gzorro.eu/.

[18] *5G-ENSURE: 5G Enablers for Network And System secUrity and ResiliencE*. [Website]. Accessed: Oct. 10, 2022. URL: https://www.5gensure.eu/.

[19] Spyridon Vassilaras, Lazaros Gkatzikis, Nikolaos Liakopoulos, Ioannis N. Stiakogiannakis, Meiyu Qi, Lei Shi, Liu Liu, Merouane Debbah, and Georgios S. Paschos. "The Algorithmic Aspects of Network Slicing". In: *IEEE Communications Magazine* 55.8 (2017), pp. 112–119. DOI: 10.1109/MCOM.2017.1600939.

[20] Mourice O. Ojijo and Olabisi E. Falowo. "A Survey on Slice Admission Control Strategies and Optimization Schemes in 5G Network". In: *IEEE Access* 8 (2020), pp. 14977–14990. DOI: 10.1109/ACCESS.2020.2967626.

[21] Zilong Ye, Xiaojun Cao, Jianping Wang, Hongfang Yu, and Chunming Qiao. "Joint topology design and mapping of service function chains for efficient, scalable, and reliable network functions virtualization". In: *IEEE Network* 30.3 (2016), pp. 81–87. DOI: 10.1109/MNET.2016.7474348.

[22] Pham Tran Anh Quang, Yassine Hadjadj-Aoul, and Abdelkader Outtagarts. "A Deep Reinforcement Learning Approach for VNF Forwarding Graph Embedding". In: *IEEE Transactions on Network and Service Management* 16.4 (2019), pp. 1318–1331. DOI: 10.1109/TNSM.2019.2947905.

[23] Anestis Dalgkitsis, Luis A. Garrido, Prodromos-Vasileios Mekikis, Kostas Ramantas, Luis Alonso, and Christos Verikoukis. "SCHEMA: Service Chain Elastic Management with Distributed Reinforcement Learning". In: *2021 IEEE Global Communications Conference (GLOBECOM)*. 2021, pp. 01–06. DOI: 10.1109/GLOBECOM46510.2021.9685162.

[24] Bin Han, Vincenzo Sciancalepore, Di Feng, Xavier Costa-Perez, and Hans D. Schotten. "A Utility-Driven Multi-Queue Admission Control Solution for Network Slicing". In: *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*. 2019, pp. 55–63. DOI: 10.1109/INFOCOM.2019.8737517.

[25] Nazih Salhab, Rana Rahim, Rami Langar, and Raouf Boutaba. "Machine Learning Based Resource Orchestration for 5G Network Slices". In: *2019 IEEE Global Communications Conference (GLOBECOM)*. 2019, pp. 1–6. DOI: 10.1109/GLOBECOM38437.2019.9013129.

[26] Amal Kammoun, Nabil Tabbane, Gladys Diaz, and Nadjib Achir. "Admission Control Algorithm for Network Slicing Management in SDN-NFV Environment". In: *2018 6th International Conference on Multimedia Computing and Systems (ICMCS)*. 2018, pp. 1–6. DOI: 10.1109/ICMCS.2018.8525945.

[27] Nipuni Uthpala Ginige, K. B. Shashika Manosha, Nandana Rajatheva, and Matti Latva-aho. "Admission Control in 5G Networks for the Coexistence of eMBB-URLLC Users". In: *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. 2020, pp. 1–6. DOI: 10.1109/VTC2020-Spring48590.2020.9129141.

[28] Fidan Mehmeti and Thomas F. La Porta. "Admission Control for URLLC Users in 5G Networks". In: *Proceedings of the 24th International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. New York, NY, USA: Association for Computing Machinery, 2021, pp. 199–

206. ISBN: 9781450390774. DOI: 10.1145/3479239.3485686.

[29] Bin Han, Antonio DeDomenico, Ghina Dandachi, Anastasios Drosou, Dimitrios Tzovaras, Roberto Querio, Fabrizio Moggio, Omer Bulakci, and Hans D. Schotten. "Admission and Congestion Control for 5G Network Slicing". In: *2018 IEEE Conference on Standards for Communications and Networking (CSCN)*. 2018, pp. 1–6. DOI: 10.1109/CSCN.2018.8581773.

[30] Rongpeng Li, Zhifeng Zhao, Qi Sun, Chih-Lin I, Chenyang Yang, Xianfu Chen, Minjian Zhao, and Honggang Zhang. "Deep Reinforcement Learning for Resource Management in Network Slicing". In: *IEEE Access* 6 (2018), pp. 74429–74441. DOI: 10.1109/ACCESS.2018.2881964.

[31] Bin Han, Ji Lianghai, and Hans D. Schotten. "Slice as an Evolutionary Service: Genetic Optimization for Inter-Slice Resource Management in 5G Networks". In: *IEEE Access* 6 (2018), pp. 33137–33147. DOI: 10.1109/ACCESS.2018.2846543.

[32] ITU-T. *Overview of TMN Recommendations*. Recommendation ITU-T M.3000 (02/00). International Telecommunication Union – Telecommunication Standardization Sector, Feb. 2000. URL: https://www.itu.int/rec/T-REC-M.3000-200002-I.

[33] ETSI. *Network Functions Virtualisation (NFV); Management and Orchestration; Report on Architectural Options*. Standard ETSI GS NFV-IFA 009 V1.1.1. European Telecommunications Standards Institute, July 2016.

[34] 3GPP. *5G end to end Key Performance Indicators (KPI)*. Technical Standard TS 28.554, ver. 17.8.0. 3rd Generation Partnership Project, Sept. 2022. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3415.

[35] Yaser Al Mtawa, Anwar Haque, and Bassel Bitar. "The Mammoth Internet: Are We Ready?" In: *IEEE Access* 7 (2019). doi: 10.1109/ACCESS.2019.2941110. ISSN: 21693536.

[36] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. "Sanity Checks for Saliency Maps". In: *32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*. 2018.

[37] Dario Bega, Marco Gramaglia, Marco Fiore, Albert Banchs, and Xavier Costa-Perez. "DeepCog: Cognitive network management in sliced 5G networks with deep learning". In: *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE. 2019, pp. 280–288.

[38] Luis A. Garrido, Prodromos-Vasileios Mekikis, Anestis Dalgkitsis, and Christos Verikoukis. "Context-Aware Traffic Prediction: Loss Function Formulation for Predicting Traffic in 5G Networks". In: *ICC 2021 - IEEE International Conference on Communications*. 2021, pp. 1–6. DOI: 10.1109/ICC42927.2021.9500735.

[39] 3GPP. *System architecture for the 5G System (5GS)*. Technical Standard TS 23.501, ver. 17.6.0. 3rd Generation Partnership Project, Sept. 2022. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3144.

[40] *OpenStack*. [Website]. Accessed: Oct. 10, 2022. URL: https://www.openstack.org/.

[41] *Docker*. [Website]. Accessed: Oct. 10, 2022. URL: https://kubernetes.io/.

[42] ETSI. *Report on the Enhancements of the NFV architecture towards Cloud-native and PaaS*. Report ETSI GR NFV-IFA 029, V3.3.1. European Telecommunications Standards Institute, Nov. 2019.

[43] 3GPP. *Security architecture and procedures for 5G System*. Technical Standard , ver. 17.7.0. 3rd Generation Partnership Project, Sept. 2022. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3169.

[44] *Docker*. [Website]. Accessed: Oct. 10, 2022. URL: https://www.docker.com/.

[45] *FlexRAN*. [Website]. Accessed: Oct. 10, 2022. URL: https://mosaic5g.io/flexran/.

[46] *OpenAirInterface*. [Website]. Accessed: Oct. 10, 2022. URL: https://openairinterface.org/.

[47] *Django*. [Website]. Accessed: Oct. 10, 2022. URL: https://www.djangoproject.com/.

[48] *Angular*. [Website]. Accessed: Oct. 10, 2022. URL: https://angular.io/.

[49] *Swagger*. [Website]. Accessed: Oct. 10, 2022. URL: https://swagger.io/.

[50] *NetData*. [Website]. Accessed: Oct. 10, 2022. URL: https://www.netdata.cloud/.

[51] *PyTorch*. [Website]. Accessed: Oct. 10, 2022. URL: https://prometheus.io/.

[52] *TensorFlow*. [Website]. Accessed: Oct. 10, 2022. URL: https://www.tensorflow.org/.

[53] *Prometheus*. [Website]. Accessed: Oct. 10, 2022. URL: https://pytorch.org/.

[54] *OpenAI Gym*. [Website]. Accessed: Oct. 10, 2022. URL: https://github.com/openai/gym/.

[55] *Kafka*. [Website]. Accessed: Oct. 10, 2022. URL: https://kafka.apache.org/.

AUTHORS

Sławomir Kukliński received a Ph.D. with honours from the Warsaw University of Technology (1994), and since then, he is an Assistant Professor there, working on mobile networks security. Since 2003 he is also with Orange Polska working on self-managed solutions and network slicing. He has contributed to many projects, including EU projects FP6 MIDAS, FP7 EFIPSANS, FP7 4WARD, FP7 ProSense, Celtic COMMUNE, EU-Japanese project 5G!Pagoda and EU-China project 5G-DRIVE. Currently he is involved in EU projects, 5G!Drones, Hexa-X, and MonB5G. He is active in ITU-T standardization of network slicing. He has published nearly 100 conference and journal papers, given several invited keynotes, and received the Best Paper Award of IFIP CNSM 2015.

Lechosław Tomaszewski received an M.Sc. Eng. with honours (1996) and Ph.D. (2005) from the Silesian University of Technology in Gliwice, Poland. He has been with Orange Polska (formerly Telekomunikacja Polska) since 1996. His industrial experience covers transmission systems, mobile core and IMS, network and service management. He is interested in various aspects of 5G/6G, network slicing, AI-supported network management, and has been involved in EU H2020 5G!Pagoda, 5G-DRIVE, 5G!Drones, and MonB5G projects.

Robert Kołakowski received his B.Sc. (2018) and M.Sc. with honours (2019) in Telecommunications from the Warsaw University of Technology. He has been working for Orange Polska since 2019. His research interests cover various aspects of 5G and next-generation mobile networks, with emphasis on network management and network slicing technology. Currently, he is pursuing his Ph.D. thesis related to SDN and traffic engineering algorithms. He is involved in EU H2020 5G!Drones and MonB5G projects.

Anne-Marie Bosneag holds a Ph.D. degree in Computer Science from Wayne State University, U.S.A. (2005) and M.Sc. (1998) and B.Sc. (1997) degrees from University Politehnica, Romania. She is a Senior Research Engineer at Ericsson Ireland since 2005, where she has worked on network management research for several generations of telecommunication networks. Anne-Marie has authored seven international patents, has published and reviewed in peer-reviewed ACM, IEEE, Springer conferences and journals, as well as contributed to 5G-PPP papers in the area of 5G networks architecture and AI/ML-based network management. She has participated in several collaborative FP6, FP7 and H2020 European projects. Her current research interests are in telecommunications network evolution, network management, ML and cognitive methods.

Ashima Chawla is an Applied Research Engineer at Ericsson Ireland, where she has been working since 2019. Her current work focuses on ML-based 5G network management techniques, including techniques from Deep Neural Networks and model agnostic explainable methods. She is currently pursuing Ph.D. studies at the Technological University of the Shannon: Midlands Midwest, Ireland and she received a B.Eng. from Amity University (2013), India. She has authored multiple publications in IEEE ComSoc and Springer conferences and has been involved in one international patent. Her current research interests are in telecommunications network management, explainable AI and a mixture of experts.

Adlen Ksentini is a professor in the Communication Systems Department of EURECOM. He leads the Network Softwarization group activities related to Network Slicing and Edge Computing. He has been involved in several H2020 EU projects on 5G, such as 5G!Pagoda, 5GTransformer, 5G!Drones and MonB5G. His research interests are in Network Softwarization and Network Cloudification focusing on network virtualisation, SDN, Edge Computing, network slicing for 5G and beyond networks. He is interested in both system and architectural issues, but also in algorithms problems related to those topics, using Markov Chains, optimization algorithms and ML.

He has received the best paper award from IEEE IWCMC 2016, IEEE ICC 2012, and ACM MSWiM 2005 conferences, and has been awarded the 2017 IEEE Comsoc Fred W. Ellersick (best IEEE Communications Magazine's paper).

Sabra Ben Saad currently works at EURECOM on "Security architectures for network slice management for 5G and beyond" as a Ph.D. student, in the Communication Systems Department. She has authored multiple publications in IEEE and Wiley journals. From January to October 2019, she worked at OBS Consulting, as an SAP consultant. From August 2015 to October 2018, she studied at Telecommunications Engineering at SUP'COM, higher communication school of Tunis, in Tunisia, and from August 2013 to June 2015, as an undergraduate engineering student at IPEIM (Specialty: Physics-Chemistry), Preparatory Institute for Engineering Studies of Monastir, Tunisia. Her current research interests are in telecommunications network management, security, ML and explainable AI.

Zhao Xu is a senior researcher at NEC Laboratories Europe. She received her Ph.D. in Computer Science from Ludwig-Maximilians-University of Munich, with doctoral research at Siemens Corporate Research. Before joining NEC Labs, she was a research scientist at Fraunhofer IAIS and Fraunhofer SCAI. She has published in top conferences and served on the programme committee of prestigious conferences in ML and AI, such as ICML, IJCAI, AAAI, ICDM, ECML and UAI. She has extensive experience in the areas of AI, ML and network data analysis. Her current research interests are in the areas of explainable AI, uncertainty estimation of deep neural networks, and AI-driven network management.

Luis A. Garrido is a Senior Researcher at Iquadrat Informatica S.L since 2020. He holds an Electronics Engineering and Telecommunications degree (2011) and an M.Sc. in Electrical Engineering and Computer Science (2013) from Technological University of Panama and National Chiao Tung University, Taiwan, Province of China, respectively. He received a PhD from the Polytechnic University of Catalonia, Spain (2019). Currently, he is involved in EU-funded projects. His research interests include low-level virtualisation, cloud computing, AI/ML, and 5G/6G networks.

Anestis Dalgkitsis received an MSc in Informatics & Telecommunications Engineering from the University of Western Macedonia, Kozani, Greece in 2018. Since October 2018, he works as a Research Engineer at Iquadrat and is involved in EU-funded research projects, while pursuing his Ph.D. degree at the Technical University of Catalonia, Barcelona, Spain. His main research interests are NFV, Management and Orchestration and ML.

Bahador Bakhshi received his Ph.D. degree in Computer Engineering from Amirkabir University of Technology, Tehran, Iran in 2011. He has been with the Computer Engineering Department of Amirkabir University of Technology as an assistant professor since 2012, where he has managed several industrial projects, including IoT ones. Since 2020 he is a Researcher at CTTC in Barcelona, where he is the testbed technical manager in EU projects 5G-ROUTES, MonB5G, and 5G-MediaHUB. His research interests include 5G/6G, NFV, SDN, IoT, and Reinforcement Learning.

Engin Zeydan received his Ph.D. in February 2011 from the Department of Electrical and Computer Engineering at Stevens Institute of Technology, Hoboken, NJ, USA. Prior to that, he received his M.Sc. and B.Sc. degrees from the Department of Electrical and Electronics Engineering Middle East Technical University, Ankara, Turkey, in 2006 and 2004, respectively. Dr. Zeydan worked as a R&D engineer for Avea, a mobile operator in Turkey, between 2011 and 2016. He was with Turk Telekom Labs working as a senior R&D engineer between 2016 and 2018. He was also a part-time instructor at Electrical and Electronics Engineering department of Ozyegin University between 2015 and 2018. Currently, he works in the Services as networkS (SaS) research unit of the Centre Tecnològic de Telecomunicacions de Catalunya (CTTC) as a senior researcher and the project coordinator of the European H2020 MonB5G project. His research interests are in the areas of telecommunications and data engineering for networks.