

Advanced Edge Computing Architecture for AI-Driven Automation and Slicing in Beyond 5G

Elli Kartsakli¹, Oriol Sallent², Joan Pujol Roig³, Jordi Pérez-Romero², Irene Vilà², Nikolaos Bartzoudis⁴, Utku Gülen⁵, Semiha Kosu⁶, Oluwatayo Kolawole³, Xin Tao⁷, Swarup Kumar Mohalik⁷, Angelos Antonopoulos⁸, Charalampos Kalalas⁴, Eduardo Quiñones¹

¹Barcelona Supercomputing Center (BSC), Barcelona, Spain; ²Universitat Politècnica de Catalunya (UPC), Barcelona, Spain; ³Samsung R&D Institute, Staines, United Kingdom; ⁴Telecommunications Technological Center of Catalonia (CTTC), Barcelona, Spain; ⁵Ericsson Research, Istanbul, Turkey; ⁶Turkcell Technology, Istanbul, Turkey; ⁷Ericsson Research, Stockholm, Sweden; ⁸Nearby Computing S.L., Barcelona, Spain

Abstract—This paper proposes an open, modular, and distributed architecture for the edge-to-cloud compute continuum, in synergy with the beyond 5G and 6G architectural trends. The proposed design facilitates the lifecycle management of cloud-native services, as well as Artificial Intelligence (AI) pipelines developed for the optimization of complex and heterogeneous edge-enabled environments, while supporting a suite of novel mechanisms to ensure the trustworthiness of AI solutions. Finally, the proposed architecture introduces a potential extension of the network slicing paradigm for the joint provisioning of communication and computing resources for applications running at the edge.

Keywords— Edge computing architecture; AI/ML life cycle management; security and trustworthiness; B5G/6G evolution; edge-to-cloud compute continuum; closed-loop automation

I. INTRODUCTION

The edge computing paradigm is expected to become a key feature in emerging beyond 5G (B5G) and 6G networks, enabling novel services that require ultra-low latencies and extreme computation close to the data sources and end users. The synergy between 6G and edge computing is being studied in ongoing standardization efforts, e.g., by 3GPP and the Multi-access Edge Computing (MEC) initiative of the European Telecommunications Standards Institute (ETSI) [1]. However, the landscape is not yet consolidated [2], leaving open architectural challenges, including the coordination of compute and network resources that support edge applications and their communication needs.

In this context, the EU-funded research project VERGE [3] proposes an open, modular, and distributed edge architecture, designed to integrate seamlessly with the edge-to-cloud compute continuum. A fundamental aspect of this architecture is the lifecycle management (LCM) of Artificial Intelligence (AI) and Machine Learning (ML) processes. Such AI-driven services are used both in innovative next-generation applications, and as a means of adapting the system to the highly heterogeneous and rapidly changing B5G and eventually 6G environments. The overall VERGE concept has been introduced in [4], presenting the three conceptual pillars: i) “**Edge for AI**” (**Edge4AI**), responsible for enabling the unified compute continuum environment, encompassing the heterogeneous pool of computing resources from the edge to the cloud; ii) “**AI for edge**” (**AI4Edge**), a portfolio of AI methods to optimize and automate different aspects of the computing and network performance in diverse and dynamic B5G/6G environments; and iii) “**Security, Privacy and Trustworthiness for AI**” (**SPT4AI**), addressing relevant challenges on robustness, privacy, safety, and explainability of the developed AI models, targeting to enhance their overall trustworthiness. A preliminary version of the VERGE

architecture was presented in [5], also discussing its application to edge-enabled use cases, which include industrial applications that leverage eXtended Reality (XR) and autonomous tram services for smart cities.

Building on these earlier contributions, this paper presents the consolidated design of the proposed architecture for edge computing evolution towards 6G as considered by the VERGE project. The main contribution lies in detailing how the different architectural components are integrated to enable the compute continuum and facilitate closed-loop automation and continuous operation of AI/ML pipelines. Furthermore, the paper explores how this architecture can support the potential extension of the network slicing paradigm to jointly provision compute and network resources for applications running at the edge, by exploiting the new concept of MEC application slices recently introduced by ETSI MEC [6].

The paper is structured as follows. Section II provides a high-level overview of the main architectural blocks, which are discussed in more detail in Section III. Section IV presents the joint management of the network slices and MEC application slices for an efficient support of edge computing services, whereas Section V is devoted to conclusions.

II. HIGH-LEVEL ARCHITECTURAL OVERVIEW

VERGE proposes an evolved edge architecture integrated with the B5G network fabric. The proposed design enables the seamless execution of cloud-native services, including disaggregated Radio Access Network (RAN) functions, core network functions, AI-powered applications, and AI-based edge and network optimization functions. The VERGE architecture is modular and scalable, facilitating the levels of overall trustworthiness needed by the AI-based solutions employed to enable the platform to adapt to the diverse and ultra-demanding requirements of 6G applications.

A high-level overview is shown in Figure 1, depicting the main interactions between the three VERGE pillars: i) the **Edge4AI**, with its three key components, i.e., the Orchestration, Management, and Control (OMC) layer, the cognitive framework, and the programming models; ii) the **AI4Edge**; and iii) the **SPT4AI**. The edge-to-cloud compute continuum infrastructure is depicted at the bottom of the figure, abstracting all the available compute resources where the different applications, services, and intelligent algorithms are executed and orchestrated. 5G/6G network components, including RAN, core, and transport network, are represented below the compute continuum infrastructure. In reality, these two domains are converging and cannot be strictly separated, as we will discuss later. Finally, the VERGE Open Dataspace represents the access to an open repository where relevant datasets for the training of AI/ML models are stored.

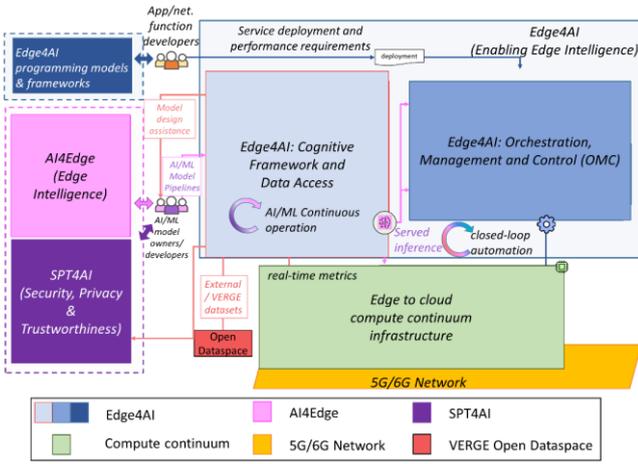


Figure 1 – High level overview of the VERGE architecture

Multiple stakeholders are involved in this complex ecosystem, including telecom operators, edge and cloud infrastructure providers, equipment manufacturers, and end users (e.g., vertical industries). All these stakeholders will benefit from this modular design, even though their interactions and relevant business models extend beyond the proposed architecture. A notable group among these stakeholders is software developers, who are expected to play a key role in the evolution of edge computing by designing applications, networks, and AI-enabled services that run on this heterogeneous infrastructure. As shown in Figure 1, these stakeholders are grouped into two categories:

- **Application and network function developers:** These are the software developers who design the vertical applications and virtualized network functions (VNFs), to be deployed and executed as cloud-native services on the compute continuum infrastructure. Cloud-native design is generally adopted, splitting monolithic structures into microservices for flexible and scalable deployment and execution. Edge4AI additionally provides support for specialized programming models and frameworks that fully exploit the capabilities of the underlying infrastructure, e.g., by supporting distributed/split computing execution, parallelization over hardware (HW)-accelerated platforms, etc.
- **AI/ML model owners and developers:** These are the software developers of the secure and trustworthy AI-based solutions aiming to optimize and automate the network and the compute continuum. The AI4Edge pillar provides multiple AI/ML models and services for networking and compute resource optimization, and includes support for advanced distributed strategies such as federated and split learning [8]. Furthermore, the SPT4AI layer introduces specific techniques that aim to increase the trustworthiness of the AI models [9]. It is worth noting that AI/ML model development can be conducted by multiple stakeholders (e.g., network equipment manufacturers, specialized third parties, etc.).

Figure 1 illustrates how these stakeholders interact with the proposed architecture. On the one hand, the programming models and frameworks supported in the Edge4AI pillar (top left) can be employed by the app/network function developers to create and optimize services that exploit the compute

continuum. The OMC layer (top right) is responsible for deploying and orchestrating these cloud-native services on the most appropriate resources of the compute continuum. On the other hand, the AI/ML model owners and developers can access and/or develop the portfolio of AI/ML models and services within the AI4Edge and SPT4AI pillars (middle and bottom left, respectively). These models can be registered and accessed through the cognitive framework, which facilitates the AI model developers to manage and automate the AI/ML pipelines, handling all aspects of the AI LCM and continuous operation. This includes managing data coming from the compute continuum and 5G/6G network infrastructure (real-time metrics) and/or other datasets from the Open Dataspace. Finally, the results of the inference of the AI/ML models are consumed by the OMC layer, providing the necessary insights for intelligent decision-making, which are then enforced by the different components of the OMC layer, closing the loop of the management and orchestration plane.

III. THE VERGE ARCHITECTURE FOR EDGE COMPUTING EVOLUTION

All the above features are depicted in more detail in the consolidated VERGE architecture (Figure 2). The details are discussed in this section, starting with the compute continuum infrastructure and the cloud-native services (Section III.A), continuing with the LCM of the cloud-native applications and network services (Section III.B), tackled by the Edge4AI programming models and OMC components. Finally, the LCM of trustworthy AI services is discussed in Section III.C, detailing the available assets provided by the AI4Edge and SPT4AI pillars and the AI/ML continuous operation capabilities enabled by the Edge4AI cognitive framework.

A. Compute continuum and cloud-native services

The bottom part of Figure 2 depicts the compute continuum infrastructure (in green). Computational resources are distributed across multiple decentralized edge sites, each potentially consisting of multiple edge computing nodes, e.g., edge servers, micro-data centres, or smaller-footprint AI embedded platforms, that feature heterogeneous processing elements such as Graphics Processing Units (GPUs) and Field-Programmable Gate Arrays (FPGAs). These sites may be collocated with different RAN nodes, such as gNBs or relays, hosting the relevant VNFs. They may also be situated at network aggregation points, forming regional data centers or extend to public or private cloud domains.

This flexible infrastructure forms an edge-to-cloud compute continuum that enables service components to be placed according to their specific needs and current network status. In addition, compute resources may also be available on-device/on-premise, as in the case of UEs with high computational capacities (e.g., high-end smartphones or connected devices at vehicles equipped with onboard embedded processors).

Three broad categories of services are running on this infrastructure, described below:

- **Application functions** implement the edge-enabled vertical services, such as XR-enabled services (e.g., rendering XR content for industrial collaborative design) or autonomous tram and smart city services (e.g., AI-aided hazard detection). These services run across the compute continuum, possibly spanning multiple edge sites and even on-device. To tackle the

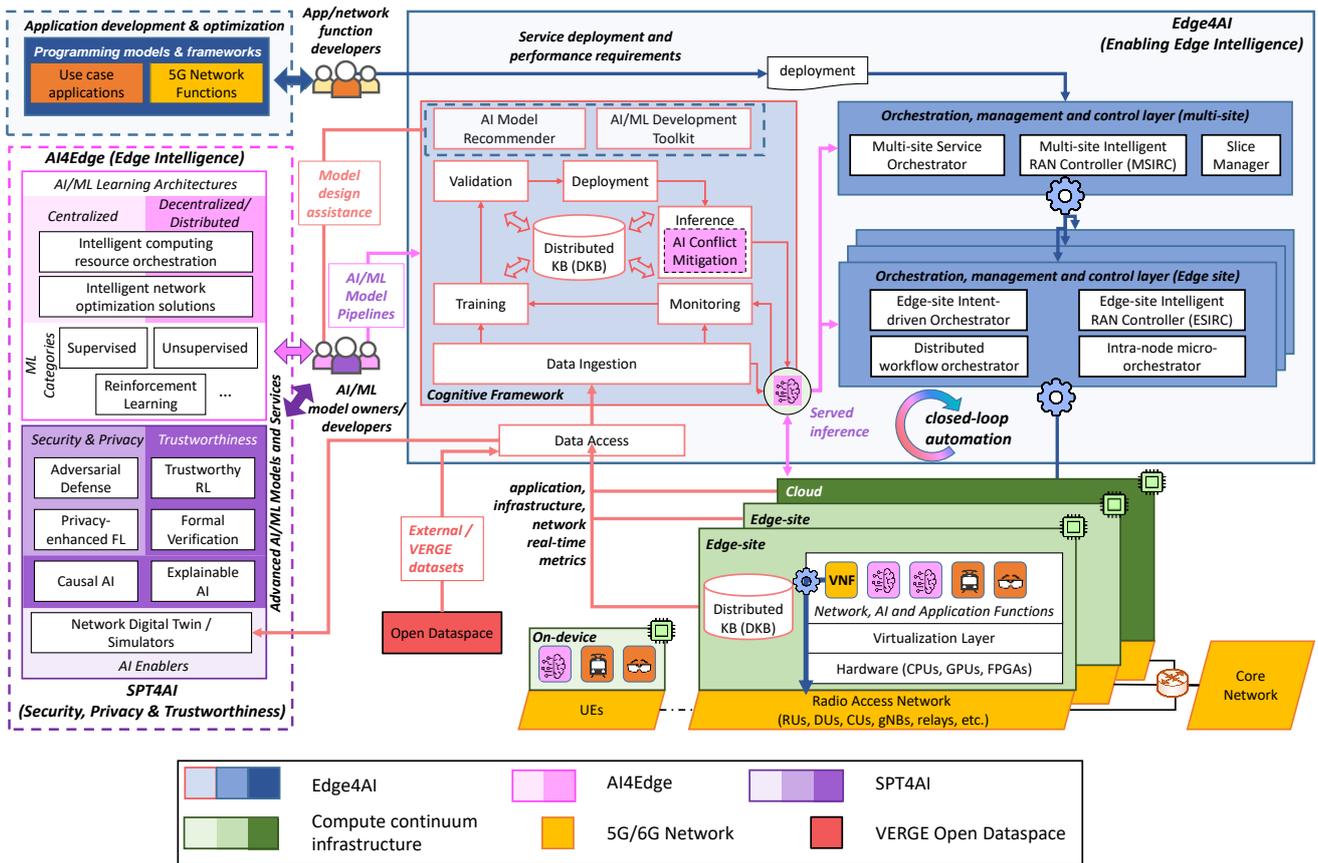


Figure 2 – Proposed architecture for the edge computing evolution in B5G.

infrastructure heterogeneity, new paradigms of HW abstraction and flexible virtualization are needed for computational offloading, splitting, or scaling.

- **Network functions** mainly represent cloud-native VNFs for the RAN and the core network executed at the edge. Specific examples include virtualized RAN (vRAN) functions, e.g., Distributed Unit (DU) or Central Unit (CU) functionalities of a gNB or virtualized core network functions like a local User Plane Function (UPF). These VNFs will coexist with other functions implemented on dedicated HW, such as the Radio Unit (RUs) of the gNBs, relay nodes, etc., as shown at the bottom of the figure.
- AI/ML functions represent different stages of the execution workflow of the AI/ML models in the AI4Edge intelligence layer. This includes training or the inference that may run at the edge, as well as distributed learning methods, such as Federated Learning (FL), that may also be executed on-device.

B. LCM of cloud-native applications and network services

The application and network developers (top left) determine the cloud-native vertical and network services that will run on the compute continuum. During the development of the services, they can optionally leverage the supported suite of programming models and frameworks, fully described in [7]. The resulting specialized workflows aim to boost computational performance, targeting heterogeneous and accelerated resources (e.g., GPUs and FPGAs), providing additional capabilities for splitting the computation vertically between UEs and edge resources, and supporting distributed execution. These models are described as follows:

- **ReproAccel** [10], a programming framework for FPGA-based System-on-Chip devices, which enables the seamless runtime reconfiguration of the software or FPGA-accelerated part of a function in multi-tenancy scenarios.
- **Split computing** framework [7], enabling the optimal and dynamic split of heavy AI inference workloads, thus distributing computation vertically between constrained UEs and more powerful edge nodes.
- **COMP Superscalar** (COMPSs) [11], a distributed computation programming model and runtime framework, to facilitate the development of complex AI workflows and their distributed execution over heterogeneous edge computing resources (e.g., CPUs, GPUs, etc.).
- A framework to enable and optimize the execution of **FL workflows** over heterogeneous edge environments [7], supporting time-triggered aggregation methods leveraging the advantages of synchronous and asynchronous aggregation.
- **oneAPI** [7], an open, cross-industry, multi-architecture programming model that aims to optimize AI/ML applications over heterogeneous hardware platforms (CPUs, GPUs, etc.).

The developed applications and network services are deployed and orchestrated by the OMC layer, which is modular and hierarchical, containing the necessary components to manage both the cloud-native services and compute resources and the networking components. The

following OMC components are considered:

- The **multi-site service orchestrator**, which has a global view of the compute continuum spanning across multiple edge (and potentially cloud) sites, e.g., organized as Kubernetes clusters. The multi-site orchestrator is the entry point of the cloud-native services and is responsible for their LCM across the compute continuum, supporting AI-driven orchestration decisions such as workload placement or migration actions.
- The **edge-site Intent-Driven Orchestrator (IDO)**, handling the LCM of the applications/services and resources on each local edge site (cluster). The edge-site orchestrator leverages an IDO planner component to enable intent-driven resource management. The developers can define the desired performance requirements (e.g., latency constraints) for the deployed services in the form of intents and the edge-site IDO will perform the necessary actions (e.g., resource allocation or scaling) to meet them. The closed-loop automation enables the IDO to learn the system capabilities and adapt to changes of the compute continuum.
- The **distributed workflow orchestrator**, handling the fine-grained management of distributed applications developed with the COMPSs programming model. This orchestrator determines the order in which the tasks of a distributed workflow are executed (e.g., prioritization of real-time tasks) and maps them to the most appropriate resources (e.g., specific tasks may require GPU resources or may have locality constraints) based on customizable and possibly intelligent policies.
- The **slice manager**, which handles the LCM of end-to-end slices that encompass both the network slice and the new MEC application slice concept recently introduced by ETSI MEC [6], as it will be further elaborated in Section IV.
- The **Multi-site and Edge-site Intelligent RAN Controllers (MSIRC and ESIRC, respectively)**, responsible for the management of the RAN elements. In VERGE, intelligent RAN control decisions are spatially separated into a multi-site perspective at the MSIRC that makes decisions regarding the optimization of multiple RAN sites, and an edge-site perspective at the ESIRC that applies intelligent control decisions with a local RAN site scope. The RAN controllers implement several intelligent solutions, e.g., for distributing radio resources among network slices, for relay management or for intelligent reconfiguration and scaling of accelerated RAN functions.
- The **intra-node micro-orchestrator**, which is an optional component that enables the joint micro-orchestration of the radio and computing resources on FPGA-based accelerator platforms. Leveraging the agile function reconfigurability provided by the ReproAccel programming framework, this component supports intelligent decision-making to handle actions such as function reconfiguration, replacement, scaling, or migration (i.e., to another

on-chip processing element).

C. LCM trustworthy AI services

A fundamental part of the proposed architecture is supporting trustworthy intelligent solutions for the automation and optimization of this heterogeneous compute continuum environment. It is expected that a wide range of AI-enabled solutions may be needed to target various aspects of the system (e.g., optimization of compute or RAN resources). Such solutions might operate on different time scales (e.g., near real-time or non real-time), have a local or global scope (e.g., optimization of a single edge cluster or RAN site versus multi-site optimization), or consider specific requirements driven by the use cases (e.g., slicing solutions for smart mobility scenarios or specific to XR-generated traffic).

In this context, the architecture serves a two-fold objective. First, it supports a portfolio of AI solutions (AI4Edge pillar) to tackle a broad spectrum of challenges in edge-enabled 6G scenarios, while integrating methodologies to ensure security, robustness, privacy, safety, explainability, and overall trustworthiness of AI models (SPT4AI pillar). Second, it provides future-proof mechanisms for the unified LCM of the AI/ML solutions. In this respect, the cognitive framework offers services and Application Programming Interfaces (APIs) that enable the continuous AI/ML operation and facilitate the onboarding of multiple and diverse AI/ML models.

The AI4Edge pillar consists of a suite of AI/ML models that target: i) the intelligent allocation and scaling of computing resources, enforced by the multi-site, edge-site, and distributed workflow orchestrators of the OMC layer; and ii) the intelligent optimization of the networking resources, providing AI-driven policies to optimize aspects of the B5G systems handled by the slice manager, the MSIRC and ESIRC, and the intra-node micro-orchestrator.

The AI4Edge is sustained on AI/ML models encompassing different strategies, such as supervised learning, unsupervised learning, and reinforcement learning (RL), as well as centralized and decentralized/distributed learning architectures. For example, the AI4Edge portfolio [8] includes solutions such as transformer models, deep RL (DRL), advanced FL paradigms for collaborative model training and inference, multi-agent frameworks, etc.

Furthermore, the SPT4AI pillar offers specific solutions to enhance the security, privacy, and trustworthiness of AI models. The proposed design supports the integration with enabling tools such as a Network Digital Twin (NDT) or other simulation environments that can be leveraged to train and test the AI/ML solutions in a safe and controlled environment. The envisioned SPT4AI models and services [9] can be grouped as follows:

- **Adversarial defense methods**, aiming to detect and mitigate AI threats in the training phase, affecting the quality of the training data.
- **Privacy-enhancement methods for FL**, such as homomorphic encryption, ensuring the privacy of sensitive information that might be disclosed during model updates.
- Solutions to maximize the **trustworthiness of RL** solutions, which leverage the NDT to ensure the rigorous testing, validation, and continuous

monitoring of the learned policies.

- Methods for the **formal verification of RL**, ensuring that the trained models satisfy the properties specified by the model user and/or developer.
- **Causal AI techniques**, adopting causal representation learning to improve the sample efficiency and generalizability of training data, thus contributing to the robustness and safety of the trained AI models.
- **Explainable AI solutions**, targeting to improve the interpretability and transparency of AI models without significantly compromising their scalability.

The cognitive framework, shown at the centre of the architecture in Figure 2, enables the development, deployment, and LCM of the aforementioned AI/ML models. Through the cognitive framework APIs, AI model developers can access the AI/ML development toolkit, that includes widely used software tools and libraries for model design. Additionally, a model recommender function implements a data-driven service that helps developers to discover existing models (previously introduced and tested on the infrastructure) that meet the desired performance requirements. A key component of the cognitive framework is the Distributed Knowledge Base (DKB), which stores all relevant knowledge generated and required by the AI/ML models, such as the datasets used for training, the trained and validated models, and other relevant metadata and platform metrics. The distributed nature of the DKB, with nodes both at a central location and at the edge sites, ensures the proximity of data, facilitating also the implementation of distributed training algorithms such as FL.

The main cognitive framework services that manage fundamental ML Operations (MLOps) are:

- **Data ingestion service:** Considers both the real-time data streams from the compute continuum infrastructure, the network domain, and the applications, as well as the existing datasets in the VERGE open dataspace. The data access layer is responsible for providing the necessary endpoints to make this data available to the ingestion service.
- **Training and validation services:** They enable AI developers to specify the learning model and validation methods, as well as the respective datasets.
- **Deployment service:** Facilitates the deployment of the trained models on the infrastructure at the optimal location based on latency and resource requirements, configuring the model serving environment, and updating the DKB with associated metadata. The deployment service may involve A/B testing, where multiple model versions are deployed to subsets of the infrastructure for performance comparison.
- **Inference service:** This service ensures the exposure of the AI inference to the corresponding components of the OMC layer. Depending on the specific requirements of each AI model, the AI inference may run as a cloud-native service at a given edge site (e.g., to enable near real-time edge-site optimizations) or at a centralized location (e.g., cloud) and be accessed as a service by the OMC components using the cognitive framework interfaces. Furthermore, an AI conflict

mitigation model, developed within the AI4Edge pillar, can be employed for the resolution of specific cases where different AI inference models generate conflicting decisions.

- **Monitoring service:** This service allows tracking key performance indicators of the deployed models, implementing data drift detection, generating alerts for performance degradation that may trigger model retraining, and provides visualization tools for platform users (e.g., the mobile network operators).

IV. NETWORK AND MEC APPLICATION SLICING SUPPORT

Network slicing is a key feature of 5G systems, enabling the simultaneous support of multiple services with heterogeneous network requirements, such as data rates, latency, reliability, and availability. It allows the sharing of a common infrastructure by creating end-to-end logical networks (network slices) on top of it. Each network slice is optimized based on the specific requirements of the services it supports. While the network slice concept is already quite consolidated ([12] and references therein), a new dimension of slicing that encompasses the edge computing domain has been recently introduced by ETSI MEC [6].

This concept, termed **MEC Application (App) slicing**, arises from a customer-oriented perspective, where customers use a virtualized application (e.g., XR or AI application), referred to as MEC App, that needs to consume MEC resources. In this case, the network slicing cannot cover the additional requirements of MEC Apps, such as compute resource allocation, application-level isolation, or virtualization technologies (e.g., deployment as virtual machines, containers). Thus, the MEC App slice arises as an independent entity from the network slice, relying on similar concepts regarding isolation and QoS guarantees but adapted to the MEC environment [13].

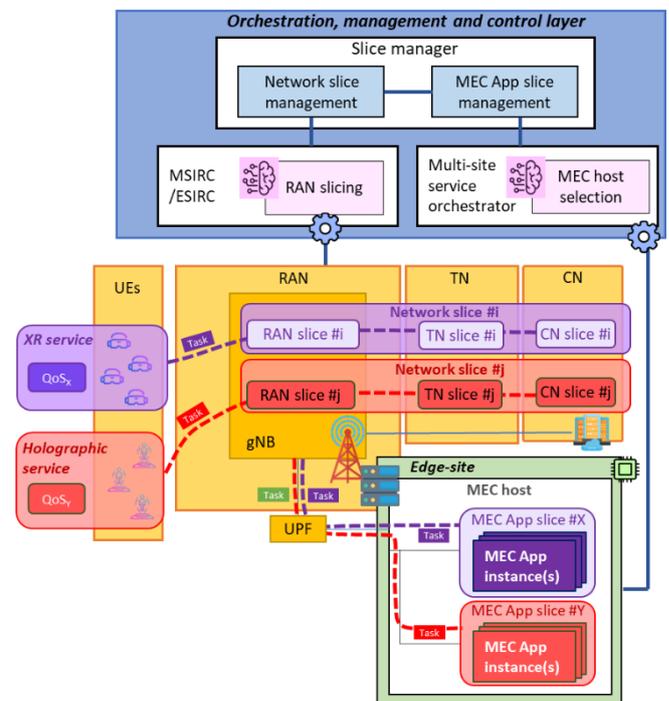


Figure 3 – Joint vision of network and MEC application slices

According to this vision, to provide a service that involves

the computation of tasks at the MEC system, an end-to-end slice is required. This slice comprises i) a network slice instance that provides the necessary network connectivity and ii) a MEC App slice instance that provides the computational power at the edge site. The MEC App slice instance is composed of the required MEC App instances according to the architecture of the software solution of the service [13]. For example, in case of a holographic service, the MEC App slice might include a MEC App instance for rendering, another for positioning and status update, etc. This joint vision of MEC App slices and network slices is illustrated in Figure 3, which shows an example with two services, each one supported by a different network slice. These slices consist of RAN, transport network (TN) and core network (CN) subnet slices, along with a MEC App slice available at the MEC host of an edge site, which in this example is collocated with the gNB. The gNB can forward IP-based traffic to the MEC host via the local UPF.

Although network and MEC App slices are distinct entities, from a management perspective, they must be jointly considered to properly serve the customers. Thus, as depicted in Figure 3, the slice manager in the OMC layer provides management functionalities for both network and MEC App slices. The network slice management function is in charge of the LCM of end-to-end network slices, including their creation, operation, and termination. To that end, it interacts with other OMC layer components, such as the MSIRC or the ESIRC, to enforce these decisions, e.g., supported by an AI/ML model to optimize the allocation of radio resources to the RAN slices of the different cells. Similarly, the MEC App slice management oversees the LCM of the MEC App slice instances based on their requirements, and interacts with the multi-site service orchestrator, e.g., to onboard application packages and select the MEC hosts for instantiating the MEC App instances associated with a MEC App slice. These decisions can also be supported by AI/ML models.

V. CONCLUSION

This paper presented an open, modular, and distributed architecture for the edge-to-cloud compute continuum in B5G systems, consolidated under the VERGE project. The proposed design specifically addresses the challenges and requirements of emerging applications and AI-enabled services in highly heterogeneous environments, introducing the necessary mechanisms for AI and service lifecycle management, closed-loop automation, and AI trustworthiness. Furthermore, the paper shares the vision of joint management of network and compute resources to meet the requirements of applications running at the edge, under the emerging concept of MEC application slicing.

ACKNOWLEDGEMENT

Project VERGE has received funding from the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101096034. Views and opinions expressed are, however, those of the authors only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them. UK participants in VERGE are supported by UKRI grant numbers 10071211 (Samsung Electronics (UK) Limited) and 10061781 (King's College London). The authors would like to thank the whole VERGE consortium for shaping the VERGE architectural concepts.

REFERENCES

- [1] ETSI GS MEC 003, v3.2.1, "Multi-access Edge Computing (MEC); Framework and Reference Architecture", April 2024.
- [2] G. Karagiannis, "StandICT.eu Landscape Analysis Report - Edge Computing Standardisation Gaps", June 2024. doi: 10.5281/zenodo.10891284.
- [3] <https://www.verge-project.eu/>
- [4] E. Kartsakli et al., "AI-Powered Edge Computing Evolution for Beyond 5G Communication Networks," 2023 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit), Gothenburg, Sweden, 2023, pp. 478-483.
- [5] E. Kartsakli et al., "An Evolutionary Edge Computing Architecture for the Beyond 5G Era," 2023 IEEE 28th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Edinburgh, United Kingdom, 2023, pp. 61-67.
- [6] ETSI GR MEC 044 v3.1.1, "Multi-access Edge Computing (MEC); Study on MEC Application Slices", April, 2024.
- [7] E. Kartsakli (editor), "First report on VERGE Edge4AI design", Deliverable D2.1 of VERGE project, December, 2023.
- [8] O. Y. Kolawole, J. S. Pujol Roig (editors), "First report on VERGE edge intelligence and network management", Deliverable D3.1 of VERGE project, December, 2023.
- [9] H. Alakoca, Ö. F. Tuna, U. Gülen, G. Kesik (editors), "Initial design of VERGE solutions for security, privacy, and trustworthiness for AI", Deliverable D4.1 of VERGE project, December, 2023.
- [10] N. Bartzoudis, J. Rubio Fernández, D. López-Bueno, A. Román Villarroel, A. Antonopoulos, "Agile FPGA Computing at the 5G Edge: Joint Management of Accelerated and Software Functions for Open Radio Access Technologies," *Electronics* 2024, vol. 13, no. 4:701.
- [11] E. Tejedor et al., "PyCOMPSS: Parallel computational workflows in Python," in *International Journal of High Performance Computing Applications*, Jan. 2017, doi: 10.1177/1094342015594678.
- [12] P. Wyszowski, et al. "Comprehensive Tutorial on the Organization of a Standards-Aligned Network Slice/Subnet Design Process and Opportunities for its Automation", *IEEE Comms. Surveys & Tutorials*, Vol. 26, No. 2, 2024.
- [13] S. Bolettieri, D. Thai, R. Bruno, "Towards end-to-end application slicing in Multi-access Edge Computing systems: Architecture discussion and proof-of-concept", in *Future Generation Computer Systems*, vol. 136, November 2022, pp. 110-127.