ORCHESTRATING DISTRIBUTED 5G EDGES FOR AUTOMOTIVE CROSS-BORDER TRIALS: VALIDATION OF AN EXPERIMENTAL PROTOTYPE

Nina Slamnik-Krijestorac[1], Mauro Femminella[2], Girma M. Yilma[3], Marco Liebsch[3], Gianluca Reali[2], Johann M. Marquez-Barja[1]

[1]University of Antwerp - imec, IDLab - Faculty of Applied Engineering, [2]Dept of Engineering, University of Perugia, Italy, Consorzio Nazionale Interuniversitario per le Telecomunicazioni (CNIT), Parma, Italy, [1]NEC Laboratories Europe GmbH, Germany

NOTE: Corresponding author: Nina Slamnik-Krije\u0161torac, nina.slamnik-krijestorac@imec.be

Abstract – *The automotive industry requires ultra-reliable low-latency connectivity for its vehicles, and as such, it is one of the promising customers of 5G ecosystems and their orchestrated network infrastructure. In particular, Multi-Access Edge Computing (MEC) provides moving vehicles with localized low-latency access to service instances. However, given the mobility of vehicles, and various resource demand patterns at the distributed MEC nodes, challenges such as fast reconfiguration of the distributed deployment according to mobility pattern and associated service and resource demand need to be mitigated. In this paper, we present the orchestrated edges platform, which is a solution for orchestrating distributed edges in complex cross-border network environments, tailored to Connected, Cooperative, and Automated Mobility (CCAM) use cases within a 5G ecosystem. The proposed solution enables collaboration between orchestrators that belong to different tiers, and various federated edge domains, with the goal to enable service continuity for vehicles traversing cross-border corridors. The paper presents the prototype that we built for the H2020 5G-CARMEN trials, including the validation of the orchestration design choices, followed by the promising results that span both orchestration (orchestration latency) and application performance-related metrics (client-to-edge and edge-to-edge service data plane latencies).*

Keywords – 5G ecosystem, collaborative multi-tier orchestration, cross-border trials, distributed edges, edge computing

1. INTRODUCTION

The 5^{th} generation of the cellular mobile communication system (5G) is being deployed stepwise in the mobile operators' infrastructures, thereby promising reduced latency and high bandwidth communication services to not only mobile devices but also vertical industries. The 5G users and vertical industries have diverse service requirements, and access to services, needs to be provided in a resource and energy-efficient manner. Advanced releases of the 5G standard add features to the initial Release 15 that was frozen in 2019. The advanced features serve as a toolbox and can be selected and enabled in support of a tailored and customized network deployment, that has, for example, strong requirements on ultra-Reliable Low Latency Communication (uRLLC) with low or even zero tolerance. The Network Function Virtualization (NFV) as one of the main 5G technology enablers affords the 5G core network architecture to follow a clear separation between the control and data planes. This separation enables automated and agile deployment and Lifecycle Management (LCM) of the associated Virtualized Network Functions (VNFs), constituting to delivery of customized network services catering to a variety of use cases over the same 5G network infrastructure. Furthermore, Multi-Access Edge Computing (MEC) systems are being widely deployed in distributed cloud networks to support low-latency and localized access to virtualized services by deploying them

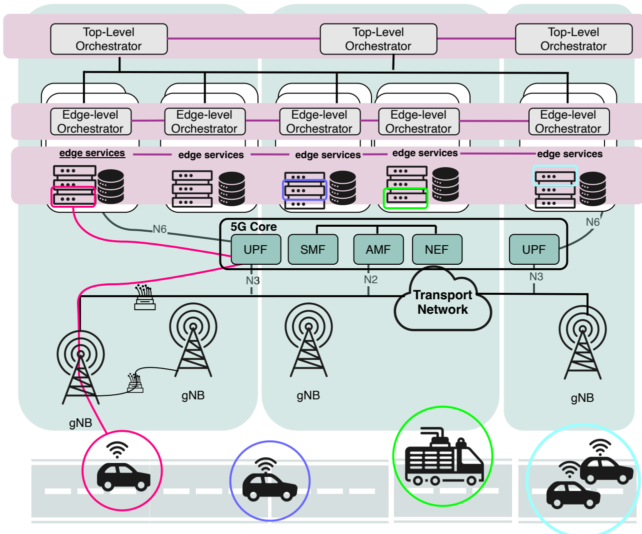

Fig. 1 – High-level overview of cross-domain orchestration of distributed 5G edges.

topologically close to the end users. However, due to the fact that 5G core, NFV, and MEC technologies are being developed by different standardization bodies, the deployment, integration, and interplay of these solutions in 5G are not coordinated.

In particular, MEC complements the 5G ecosystem end-to-end view, including deployment of and access to network and service functions from the Application Service (AS) as a service producer to the mobile User Equipment (UE) as a service consumer, with options to provide services topologically closer to the service consumer in distributed edge network data centers. Such distributed deployment of services on MEC platforms, denoted as edge services, enables the localization of services and potentially increases the experienced service quality. Furthermore, the collection and analytics of data points associated with edge services, such as data plane performance, or resource consumption, are localized as well. The provisioning of edge services on top of the virtualized infrastructure enables dynamic deployment and scaling of services. Continuous monitoring of service and platform-related data creates means for automating the deployment and LCM of distributed edge services. Hence, NFV and MEC potentially enable the optimization of a variety of Key Performance Indicators (KPIs) in the context of functional (Quality of Experience (QoE)), as well as non-functional, requirements. However, using localized services in an agile environment, such as the automotive industry with moving vehicles being connected to infrastructure services through various Mobile Network Operators (MNOs) and where UEs follow diverse and changing mobility patterns, is highly challenging. In that case, the dynamic reconfiguration of edge service deployment and maintaining connectivity of UEs to the topologically closest edge service is required, and as such, it demands edge-to-edge optimizations.

In this paper, we present the orchestrated edges platform as an achievement of the automotive EU project 5G-CARMEN, which enables federated and automated Management and Orchestration (MANO) of distributed MEC resources and corresponding edge services for Connected, Cooperative and Automated Mobility (CCAM). Key objectives of the project include the provisioning and continuity of CCAM services to connected vehicles while they travel across country borders and perform handover to a different MNO. In particular, Fig. 1 depicts a high-level view of the implemented reference architecture, where vehicles connect to edge services through the cellular 5G system of an MNO. The 5G system architecture comprises control and data plane functions, such as Access and Mobility Management Function (AMF), Session Management Function (SMF), Policy Control Function (PCF), and User Plane Function (UPF). The MNOs can extend their 5G system with MEC facilities and associated resources, where one or multiple instances of virtualized edge service can be deployed, monitored, reconfigured, and scaled, in order to meet and maintain all service level objectives and performance expectations. The described solution for orchestrating distributed 5G Edges includes key enabling elements for the localization of management and orchestration tasks, computational and protocol-related load distribution, data analytics, and mobile data plane programmability, for the enforcement of traffic steering policies on the 5G N6 reference point.

The first benefit of the proposed orchestrated edges platform is measured in terms of delegating the orchestration role to edge-level orchestrators and allowing them to communicate orchestration-related decisions directly from edge to edge. This is measured via metrics such as *orchestration latency* (duration of performing a particular orchestration operation such as CCAM service instantiation), and a *number of signaling messages* that represent a communication overhead imposed by orchestrators that work jointly towards performing service reconfigurations in cross-border setups. The second benefit is measured by studying the improvements in *data plane latency* experienced by end users (vehicles) in the case of upgrading Kubernetes native Container Networking Interface (CNI) with Fast Data Input Output (FDIO) capabilities. This metric refers to the data plane communication between the client and the CCAM service running on the edge, i.e., client-to-edge communication. Finally, we evaluate the third benefit, which is advancing the edge-to-edge communication between peering CCAM service instances, which is a service-to-service communication, thereby improving the *data plane latency* between MEC application instances.

In the view of continuous availability, quality, low latency, and scale of CCAM services in a such challenging automotive cross-border environment, this paper provides a brief insight into related work in Section 2 and a comprehensive description of the key architectural and functional components of the specified, developed and deployed enablers for federated and orchestrated 5G edges in Section 3. The methodology for testing and validation is described in Section 4 and is based on selected KPIs for the developed prototype, which has been integrated with the production network for mobile communication of three European MNOs, i.e. Deutsche Telekom AG (DTAG), Magenta Telecom Austria (MTA), and Telecom Italia Mobile (TIM). Results of the accomplished experimental analysis and validation are captured in Section 5, while Section 6 concludes this article with an assessment of the evaluated and presented KPIs as well as advice for future developments.

2. RELATED WORK

Over the course of the last decade, NFV has been an essential network and service management enabler. However, with the requirements coming from the vertical industries for real-time deployments of VNFs and physical network resources, reliable and fast lifecycle management of those functions has become extremely complex [1]. Given the high mobility of vehicles and their strict requirements for service quality in different vehicular scenarios, MEC has been studied as a prominent technology for enabling low-latency close-to-user access to the cloud-native services that enhance and extend awareness in cooperative and connected scenarios for vehicle maneuvering operations [2]. Nevertheless, as opposed to resources in the cloud, edge resources are: i) constrained, i.e., the amount of computing resources is limited due to the smaller processors and a limited power budget [3], ii) heterogeneous, as resources might belong to different vendors, and iii) distributed and dynamic, i.e., nature of edge resources is fluctuating due to the changes in workload, traffic demand, and users'

mobility [3, 4]. Therefore, proper management of such resources needs to be ensured, in order to use the computing and network resources in an optimized manner.
Given their well-known NFV MEC framework [5], ETSI is the global leader in standardizing orchestration-based frameworks, thus, NFV MANO tools should be designed and developed with reference to the ETSI NFV MANO framework to increase their applicability in real-life systems, as well as their compatibility with other orchestration systems. Our orchestrated edges platform follows the principles of the *ETSI NFV MEC framework* while extending current standards by defining reference points between mobile edge network orchestration functions.
As the focus of our paper is placed on the federation and multi-domain aspects for orchestrating highly challenging automotive services, the *multi-domain* capabilities represent a strong contributing factor to filtering the orchestration solutions. The '*multi-domain*' refers to the ability to establish a connection with MEC platforms from different edge domains using Representational State Transfer (REST) and to enable communication between different orchestration entities in multiple domains. Thus, let us start from an earlier study of challenges in multi-domain orchestration for NFV, conducted by Katsalis et al. [6], in which they proposed an architecture with distributed orchestration functions per domain, including the multi-domain orchestration entity on top of the distributed units. The main challenges that Katsalis et al. [6] noted are the distinction of boundaries between domains and the definition of the domain per se, lack of proper VNF and Network Service (NS) descriptors that can be used for multi-domain environments, the role of higher layer orchestrator (whether it should be centralized or distributed), lack of standardized interfaces, and challenges related to dynamicity in service provisioning. In this paper, and in particular Section 3, we address those challenges and present the architecture of the orchestrated edges platform that has been prototyped in the 5G-CARMEN cross-border trial environment.
There are several important pieces of research that have recently put the spotlight on federated and multi-domain orchestration in complex network environments [7, 8, 9]. In particular, Taleb et al. [7] propose a multi-domain management and orchestration framework for utilizing network slices that are deployed on top of the federated resources. Their solution is based on a fully-fledged network slice orchestration stratum, which interacts with a cross-domain slice coordinator in order to allocate single-domain resources for deploying the required network slice instances. To address the same issue, the 5GROUTES project [8] leverages European Telecommunications Standards Institute (ETSI) Zero-touch Service Management (ZSM) specification for achieving cross-domain management of services deployed in a CCAM context. However, Efthymiopoulou et al. [8] present only the idea and overview of the proposed architecture with "day 1" and "day 2" cross-domain operations (provisioning, and runtime management, respectively), with no details about the practical implementation and performance evaluation.
One practical approach to address multi-domain orchestration is presented by Baggio et al. [10], where their X-MANO solution introduces the federation over multiple domains through the following core components: i) Federation Agent (FA), which sits on top of a particular domain, and interacts with the domain orchestrators, and other lifecycle managers, ii) Federation Manager (FM), which interacts with one or more FAs, and iii) OpenVPN as a cross-domain link. Another orchestration solution that is widely used nowadays is Open Network Automation Platform (ONAP), consisting of modular and layered architecture that improves interoperability and simplifies integration with multiple VNF environments. In such an architecture, the service orchestrator performs orchestration at a high level, with an end-to-end view of the infrastructure, network, and applications, while the multi-site state coordination module enables scaling to multi-site environments to support global scale infrastructure requirements.
An important work on studying the benefits of evolving from single-domain networks to managing more challenging multi-domain scenarios is presented by Sciancalepore et al. [11]. Their solution is implemented leveraging the aforementioned ONAP concepts, and in the work [11], they focus on a 5G management and orchestration architecture that overcomes limitations imposed by state-of-the-art orchestration frameworks that are stretching single operational NFV domains. In particular, Sciancalepore et al. [11] built a proof-of-concept based on ONAP, where they collected important results on communication overhead as well as the end-to-end service delay while deploying three relevant multi-domain network slices and comparing their solution with the legacy system (a single orchestrator). Their results are promising, as they show improvements in both latencies of performing orchestration operations and signaling overhead. Our insights and results go beyond this study as we show the impact of localization gain (benefits of delegating orchestration operations to edge-level orchestrators) in a real cross-border setup built within the 5G-CARMEN project, focusing on the horizontal interactions between federated edges, once the vertical agreements between edge-level and top-level orchestrators are made (signaling overhead minimized).
We noticed that the definition of 'multi-domain' aspects varies across research works, where some of them refer to the network segments (radio, edge, transport, and core) as domains that should be orchestrated and harmonized. As a wide variety of vertical services need to operate in an end-to-end manner, i.e., stretching over network segments from fixed or mobile access to the edge and the core, or accessing the telco cloud and multiple hyper-scale clouds [1], Amdocs has developed a network orchestration solution that covers lifecycle management of services in different network segments. Their architecture consists of orchestrators that are carefully designed and located at different network segments, however, it does not include federation aspects where multiple edge or administrative domains are affected by users' maneuvering operations.
Finally, in their work on cross-domain orchestration for

edge-based smart roads, Yuan et al. [12] present an interesting solution based on a multi-agent orchestration framework leveraging swarm intelligence. In their approach, each vehicle is associated with an edge cloud agent, which is in charge of not only provisioning and managing edge resources for this vehicle but also providing routing recommendations based on the behavior of vehicles on the road. Thus, the problem they tackle encompasses also the requirements for agent migration along with the movement of vehicles. Their multi-agent-based cross-domain resource orchestration framework is evaluated in a simulation environment, leveraging learning models trained on the datasets collected from the traffic conditions and routing patterns of taxicabs in peak and mid-peak hours during 2012. Although innovative and promising, such solutions still seem complex for real-life cross-domain environments, where associating each vehicle with an always-on exclusive agent on the edge cloud seems hardly feasible given the scarce resources and limited capabilities at the edges to run massive machine learning libraries.

From the overview of the state-of-the-art in this section, we can see that the existing orchestration solutions are tackling an end-to-end perspective in virtualized network infrastructure. However, when compared to our framework they are still lacking the support for automated edge-to-edge interactions toward low-latency service deployment that are tailored to highly challenging mobile scenarios while enabling fast orchestration operations across different network edges. In addition, another missing link is coupling with 3^{rd} Generation Partnership Project (3GPP) systems, such as 5G and beyond, as most state-of-the-art frameworks do not envision the design of the platform and its operations in accordance with the overall 5G ecosystem.

3. ORCHESTRATED EDGES PLATFORM

3.1 Platform overview and architecture

In this section, we present a brief overview of key architectural components of the orchestrated edges platform, which is illustrated in Fig. 2. The architecture shows a cross-border setup, i.e., a federated environment setting, with two administrative domains that can be associated with different countries and/or MNOs. The cooperation and interaction between platform components enable federated and secure cross-domain management and orchestration of 5G edge domains with MEC platforms that host distributed service deployments.

The standardization framework provided by ETSI MEC [5], ETSI NFV [13], and 3GPP [14, 15], is used as a baseline for the design of the overall platform. The resulting and presented solution for the orchestration of distributed 5G edges includes two orchestration tiers, i.e., i) top-level orchestrators that are responsible for the orchestration of the overall administrative domain (e.g., country or network provider domain), and ii) edge-level ones that perform lifecycle management of services deployed in their respective edge domains. This two-tier architectural design enforces the collaboration between 5G edges, via the Or-Or and Lo-Lo reference points shown in Fig. 2, which translates into the interaction between edge-level orchestrators that collaborate in order to proactively perform orchestration operations based on vehicles' mobility and resource demands on the distributed MEC nodes. In addition, to further support service continuity, the orchestrated edges platform also enables services and applications deployed on those edges to connect to peering service/application instances in different domains [2] on two more layers, denoted as service-based interfaces (Service Mesh) and a mobile data plane, as indicated with the blue and red dashed line in Fig. 2 for the service mesh and the data plane respectively.

The functional split between two orchestration tiers allows top-level orchestrators, i.e., NFV Service Orchestrators (NFV-SOs), to offload/delegate their tasks to the corresponding edge-level orchestrators, in order to decrease the processing load at the top-level tier while enabling more efficient MANO operations to be performed directly at the network edges. Here we briefly describe the role of each functional component in the architecture shown in Fig. 2.

- **NFV-SO**: The operational scope of the top-level orchestrator of the two-tier orchestrated edges platform includes the management of the entire virtualized infrastructure of one administrative domain (e.g., MNO's domain). The NFV-SO is responsible for the management and orchestration of application services from multiple tenants (e.g., car manufacturers). This type of orchestrator is responsible for enabling federation with the peering NFV-SOs that belong to the adjacent administrative domains (e.g., neighboring countries). In addition, it maintains a global repository of the application packages and software images that are usually used by the edge-level orchestrators for deploying services on suitable edge nodes.

- **NFV Local Orchestrator (NFV-LO)** and **MEC Application Orchestrator (MEAO)**: The combination of these two orchestrators belongs to the edge-level tier, whereas the operational scope includes the designated edge nodes (i.e., MEC hosts) that belong to a particular edge domain. In particular, there is a $1 : N$ relationship between the NFV-SO and the edge-level orchestrators. The pair of edge-level orchestrators decouples the management operations, i.e., MEAO is responsible for the lifecycle management of CCAM services deployed as MEC applications, while the NFV-LO performs the general management of the VNFs hosted on the NFV Infrastructure (NFVI).

- **Adaptation and Abstraction Module (AAM)**: The intermediate layer that enables the communication between top-level and edge-level orchestrators is defined as AAM. It is deployed on the top of the edge-level orchestration tier and is in charge of abstracting the features of this tier,

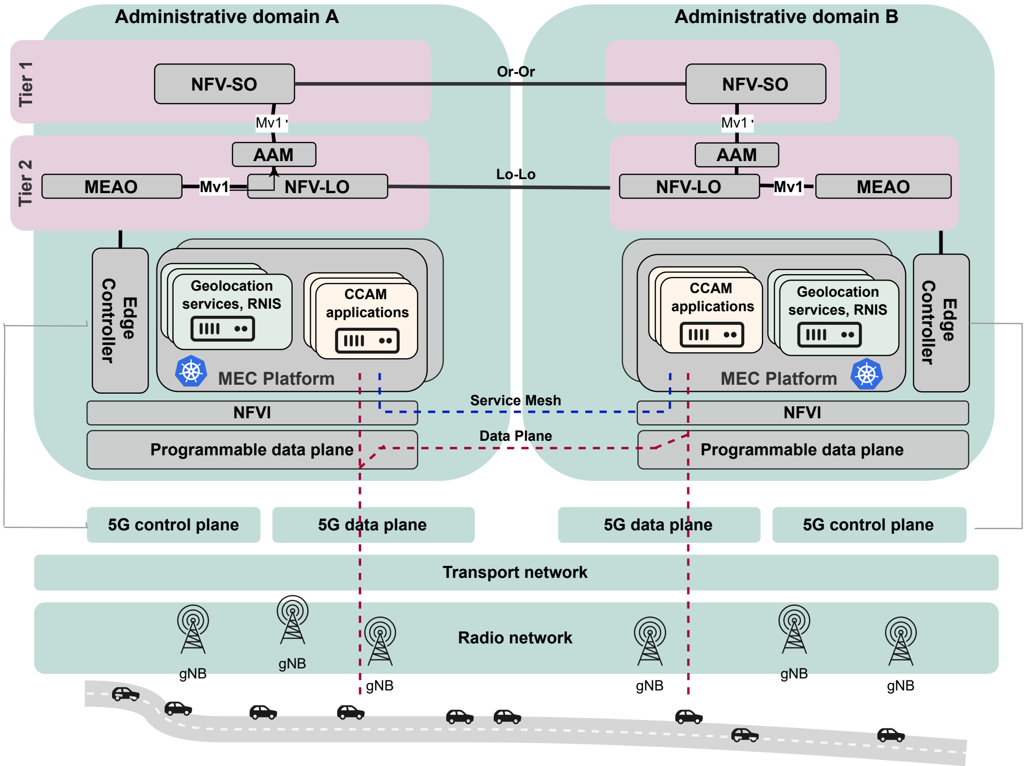

Fig. 2 – Architecture of the orchestrated edges platform, including all functional components that enable cross-domain orchestration operations and coupling with 5G system.

making it compliant with the standard de facto Open Source MANO (OSM) Application Programming Interfaces (APIs) of NFV-SO, towards the adaptation of both requests and response bodies for the messages exchanged between two tiers. Its northbound interface exposed to NFV-SO is denoted Mv1' [16], as in Fig. 2. As the MEAO takes an orchestration decision on CCAM services and communicates them to the NFV-LO via Mv1 (Fig. 2), in the same way, the NFV-SO takes a high-level orchestration decision on deployed services and communicates them via Mv1' to the NFV-LO. In turn, the NFV-SO can remain completely unaware/agnostic of the underlying NFV-LO proprietary APIs.

- **CCAM services**: Any service function or microservice instance that is deployed on the MEC nodes, thereby performing operations for CCAM-related use cases, is considered as a CCAM service. In the 5G-CARMEN project, we have made a distinction between *on-demand* services (e.g., situation-specific or dynamic mission-critical applications), and *persistent* services that need to be constantly running on the MEC nodes (e.g., assisted maneuvering services).

- **MEC Value-Added Services (VASs)**: VASs are MEC services that can be leveraged by any CCAM service as helper functions for obtaining e.g., network-related data, or UE locations. Some examples of VASs are defined by the ETSI ISG MEC [17], i.e., Radio Network Information Service (RNIS) and the geolocation service. Additional VASs in the form of message brokers can be deployed to support the dissemination of messages from the CCAM services to vehicles and vice versa.

- **MEC platform**: It represents a collection of essential functionalities required to run MEC applications (more specifically, CCAM applications) on top of the virtualized infrastructure, while these applications can deliver and consume various services, and connect to UEs (e.g., vehicles).

- **Edge Controller (EC)**: This is an abstraction layer between virtualization infrastructure management and edge-level orchestration that combines the following ETSI ISG MEC functions: MEC platform manager and VNF manager, but extending toward additional functions, such as VNF connectivity and service mesh control, data plane control, and enablers for coupling with 5G system components (as shown in Fig. 2). The EC enables edge network slicing with associated policies at the edge system level.

- **NFVI**: This virtualized infrastructure provides the necessary computing, storage, and network resources for the associated MEC applications running on top of the MEC platform.

- **5G data plane**, **5G control plane**, and **transport network** blocks represent the mobile core network abstractions, whereas the programmable data plane represents a data plane overlay of the 5G system's N6 reference point in support of policy routing and traffic steering [18].

The top-level and edge-level orchestrators interface with each other in single domains and federate with their peer orchestrators from the same tier in cross-domain operations. For the purpose of the cross-domain federation and cross-edge collaboration, both orchestrator tiers are leveraging different reference points, i.e. i) the Or-Or reference point, which is based on the ETSI NFV standard [13], and is responsible for federating between the NFV-SOs in different administrative domains, and ii) the Lo-Lo reference point, which is derived from the Or-Or reference point, and enables the cross-edge coordination between the NFV-LOs for supporting cross-domain migration orchestration operations, thereby enabling state migration, service continuity, and low-latency service orchestration requirements [2].

3.2 Requirements, features, and KPIs

The design we presented in Section 3.1 is entirely based on the features that are defined to fulfill a set of requirements for orchestrating cross-domain service deployments in highly challenging and agile environments such as CCAM deployments on distributed 5G edges. Thus, in this section, we focus on the main features of the orchestrated edges platform presented in Fig. 3, and we emphasize the ones that we focus on in this paper, defining the KPIs that we measure to evaluate the benefits of the designed features.

Two-tier orchestration: Given the mobility of UEs as the target customers of 5G-enhanced CCAM services that support them in maneuvering operations, proper coordination of services and infrastructure resources in heterogeneous 5G environments is a must for providing required levels of service quality. This becomes even more challenging if vehicles are traversing from one domain to another, changing the operator that provides 5G connectivity, where orchestrators need to take into account UE mobility, and resource demands, and based on that provide the optimal service deployment for them. Thus, two-tier orchestration is an enhancement of the state of the art orchestration mechanisms, which enables orchestrated edges platform to support decentralized multi-domain MEC environments, where services are deployed in a distributed fashion, thereby following the mobility of the UEs, i.e., vehicles. The orchestrated edges platform is capable of performing a flexible and agile service orchestration in a hierarchical and distributed manner, by deploying the top-level orchestrators in different administrative domains, and the edge-level orchestrators in multiple edge domains per each administrative domain. With such a setup, services and their respective resources can be managed locally where they are deployed (i.e., in edge domains), but different orchestration layers collaborate to optimize the outcome of the orchestration operations. The benefits of such features are evaluated through orchestration KPIs such as single-domain/cross-domain NS creation latency, and in-domain/cross-domain runtime orchestration latency (scaling, termination, and migration), including the study on component-associated contribution to the orchestration latencies. The details about those KPIs are further presented in Table 1, and in Section 5, we present the measurements for NS instantiation latency in case of single and cross-domain scenarios, identifying the impact communication and processing latencies in the overall orchestration operation.

Edge and inter-edge level autonomy per Management Level Agreement (MLA): As first introduced by Yousaf et al. in [19], the concept of MLA enables delegation of orchestration tasks between orchestrators in NFV-based orchestration platforms. In the case of the orchestrated edges platform, MLA is essential for: i) establishing federated environments by creating an Or-Or interface between NFV-SOs, ii) allowing top-level orchestrators, i.e., NFV-SOs to delegate their tasks to the edge-level orchestrators, i.e., NFV-LOs, in order to balance the orchestration load, and iii) enabling edge-level orchestrators to bypass Tier 1, and cooperate directly with their peering edge orchestrators from the other MEC domains via the Lo-Lo reference point. The MLA is negotiated between top-level NFV-SOs of different administrative domains via the Or-Or reference point and enforced in each domain at the NFV-LO via the AAM, which is introduced in Section 3.1. Furthermore, to evaluate the improvements that direct cooperation between edge-level orchestrators brings, we measure cross-domain orchestration latency and localization gain (Fig. 3), which are defined in Table 1. In particular, localization gain is measured in terms of the number of signaling messages that are exchanged between orchestrators during single-domain or cross-domain orchestration, as it might increase the traffic load and delay the overall orchestration process. Thus, in Section 5, we present the gain achieved in terms of faster cross-domain orchestration operations and decreased number of signaling messages, in the case of direct edge-to-edge collaboration.

Edge control: Complementary to the delegation of MANO operations in a federated environment, the edge control feature leverages the MEC system's awareness of a mobile subscriber's data plane policy to enforce aligned traffic treatment rules in between the UPF and the MEC service, e.g., in terms of Quality of Service (QoS), metering, or traffic steering. This feature enhances the existing Service and Session Continuity (SSC) mode 3 mechanism in 5G systems, which enables mid-session relocation of a mobile subscriber's UPF without breaking the Packet Data Network (PDN) session. The enhancement refers to the proactive deployment of CCAM service instances at target edges, which are proactively selected following the relocated UPF of a mobile subscriber that is connected to the CCAM service instance. This feature is essential for maintaining an optimized routing path between the UE and CCAM

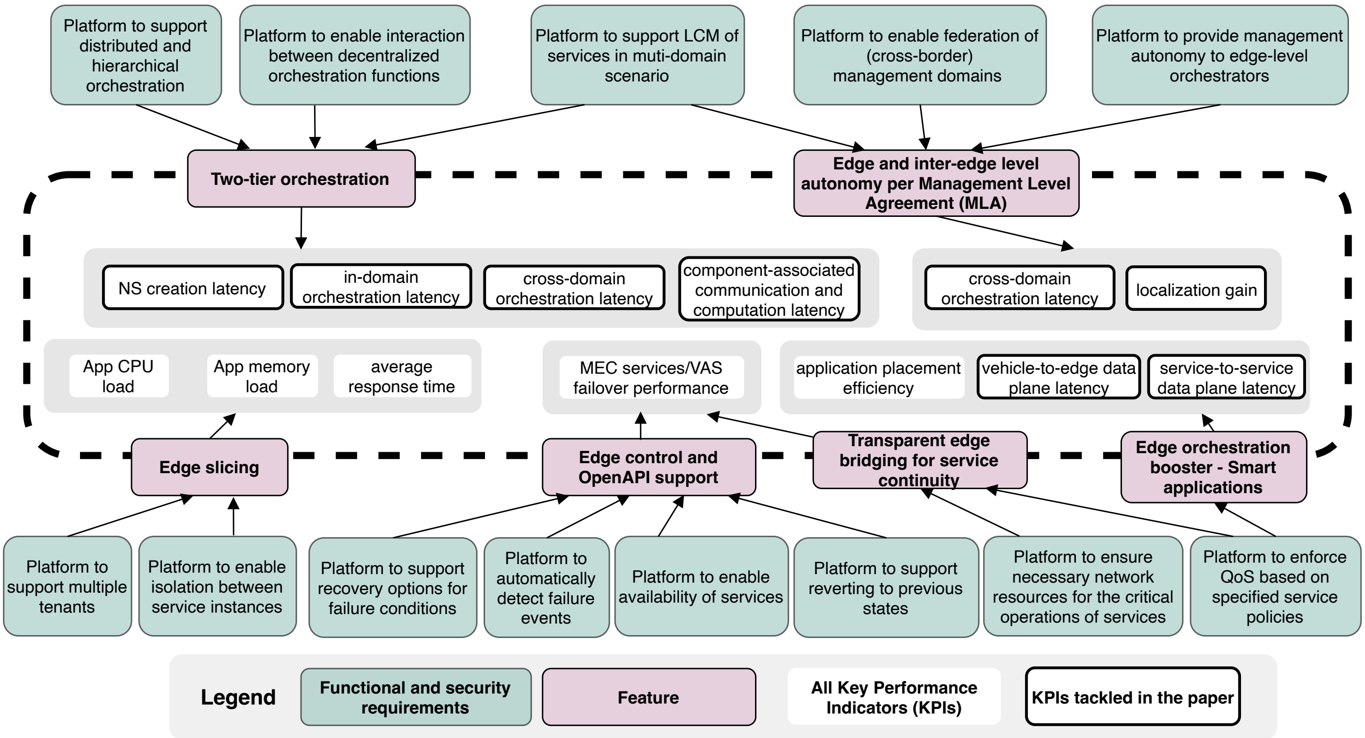

Fig. 3 – Mapping KPIs to functional requirements.

Table 1 – Definition of KPIs listed in Fig. 3.

KPI name	KPI type	Description
NS creation latency	Orchestration	Latency of creating a service instance based on the onboarded application package (e.g., descriptor, and Docker container image) in all MEC hosts within required domains.
NS instantiation latency	Orchestration	The time needed for the platform to process instantiation requests, and to instantiate CCAM applications on the selected edges.
Runtime orchestration latency	Orchestration	The time needed for the platform to perform any LCM operations, i.e., while the application instance is running (e.g., scaling, termination, heal, state migration, etc.).
Cross-domain instantiation latency	Orchestration	The overall time needed for the platform: i) to simultaneously instantiate the same application in multiple domains, and ii) to properly notify NFV-SOs about instantiation in all domains. In particular, a more detailed definition of: i) is the time needed to instantiate the peering application instance in domain 2 from the moment when an instantiation request is generated within the platform, to the moment when the peering instance is up and running in domain 2. On the other hand, ii) means an additional notification delay to notify operations to remote NFV-SO, in a SO-SO communication via Or-Or.
Component-associated communication and computation latency	Orchestration	Average communication and computation latency between specific platform components: NFV-LO & NFV-SO (inside one domain, via AAM), NFV-LO & MEAO NFV-LO & edge controller, NFV-LO & AAM, and AAM & NFV-SO (via notification)
MEC service/VAS failover performance	Orchestration	Since MEC applications realized as Kubernetes PoDs are mortal, the platform needs to react quickly to failures and bring the failed service or VAS back, including network connectivity, applying policies, number, and type of interfaces, connecting to external volume to access session states, etc.
Localization gain	Orchestration	The balance between NFV-SO/Or-Or and NFV-LO/Lo-Lo operations. Too much edge delegation can be counterproductive (NFV-LO/Lo-Lo load) and may result in latency for LCM operations.
Service-to-service data plane latency	Application	Delay in cross-domain communication between two MEC application instances.
Client-to-edge data plane latency	Application	Delay in communication between the client (e.g., vehicle) and MEC application instance.
Application placement efficiency	Application	Impact of application placement on the average response time, depending on the decision made by orchestrated edges platform (e.g., MEAO/NFV-LO).
Application CPU load	Application	Average CPU usage during application runtime.
Application memory load	Application	Average memory usage during application runtime.

service running at the MEC while achieving service continuity with short communication paths that contribute to low-latency requirements. In this paper, we measure the impact of edge control on the application performance by measuring data plane latency both between the client and the edge (i.e., consumed CCAM application), and between two peering application instances that are running on adjacent edges, while exchanging performance-specific data (vehicle location/speed, security tokens) in order to mitigate the cross-domain orchestration operations. The strategies for testing the aforementioned KPIs are described in sections 4.3 and 4.4, followed by results in sections 5.2 and 5.3, respectively.

Edge slicing: Network slicing is already a well-known concept in 5G ecosystems, as it enables MNOs to customize their virtualized network deployments to different tenants, depending on their network and service performance requirements. This concept can be applied in an end-to-end manner, covering Radio Access Network (RAN), backhaul, transport, and core network, whereas in the case of the orchestrated edges platform, we provide support for slicing at the edge computation level. In particular, top-level and edge-level orchestrators are preparing service and slice descriptors, which are further sent to EC that translates these descriptors into MEC platform-specific configurations. In particular, the use of Kubernetes namespaces and storage volumes, as well as the control of resource quota helps the EC to assign, monitor, and isolate resources dedicated to the CCAM and VAS types of edge services. This concept of edge slicing is out of the scope of this paper, but we refer the reader to our previous work [20] for more details on the edge slicing concept and performance evaluation.

Transparent edge bridging for service continuity: As described in our previous work [21], this concept is leveraging the programmable data plane at orchestrated edge level to enable transparent bridging of the data plane between vehicles and CCAM edge services within different administrative domains (MNO domains), in case vehicles are just about to reconnect to the new network when approaching a country border. Thus, this feature can be considered as a cross-edge service-level *make-before-break* handover solution, as it allows vehicles to connect to target edge services before the cellular network handover happens, thereby reducing service interruption and data loss. The transparent edge bridging procedure is triggered by edge-level orchestrators (based on their decisions, or the ones enhanced by smart CCAM applications), and it is performed by EC, which configures the programmable data plane to relay data plane packets between the two orchestrated edges, transparently to the vehicle.

Edge orchestration booster - Smart applications: This feature enhances the orchestration decisions made by orchestration tiers by leveraging the dynamic notifications generated by CCAM applications themselves. These enhancements are particularly important for service continuity, as for orchestration decisions to deploy application instances on the target edge nodes and relocate them from the source one, it is important to be on time in order to avoid or minimize service downtime. In particular, the applications that are designed to be smart and edge-aware are actually aware of the edge environment (e.g., elements of the orchestrated edge platform), other applications that are relevant for their operation (i.e., other peering CCAM applications, or MEC VASs), as well as vehicles that are connected to them. With such an increased awareness, CCAM applications are capable of generating various important notifications that could improve and boost their lifecycle management, i.e., enhance the decision-making process performed by orchestration entities. The notifications are generated based on data analytics and/or Machine Learning (ML) running in the applications and refer to the processes that are specific for the application operation (e.g., mobility of the vehicle, proximity from the border between two edge domains, or proximity from the border between two countries). Although we mention it as an important feature of the orchestrated edges platform, which ultimately enhances service continuity, this feature has been studied in detail in our previous work [21], thus, we do not elaborate further on it in this paper.

4. TESTING AND VALIDATION METHODOLOGY

In this section, we describe the testing environments, as well as the validation methodologies, which have been employed to assess the performance of orchestrated edge components and services in mobile network operator edge clouds. This methodology includes a combination of testing within the MNOs' edge clouds, on-road testing using connected vehicles, and lab testing tools for increased versatility. Additionally, various enablers and their components are being evaluated in lab settings, such as the performance of the Service-Based Interface (SBI) communication via Kubernetes NodePort compared with the CNI extension, which is expected to further decrease latency in communication between users and CCAM services deployed at the edge by enhancing the network resource utilization. Furthermore, advanced simulation tools and models are utilized to verify and create realistic data sets, such as simulating the movement patterns of a large number of vehicles on a highway, for the purpose of testing and validating enabling technologies in terms of scalability.

4.1 Testing environments

This section briefly describes a testing environment that includes both an in-lab setting (illustrated in Fig. 7), and a pilot environment deployed within the network of two MNOs and their edge cloud data centers (see Fig. 8). The in-lab test is used to validate the impact of the improvements on the client-to-edge interface to the data plane latency, i.e., to compare Kubernetes native networking with additional interfaces such as FDIO, which has been developed for speeding up the communication between the client (i.e., vehicle) and CCAM application running on the MEC system. The pilot environment, as shown in Fig. 8, is used to evaluate the data plane latency between two CCAM application instances running on adjacent edges.
In the case of pilot setting, Table 2 depicts the character-

istics of the Orchestrated edge platform deployments on top of the resources provided by different MNOs, i.e., DTAG and MTA, thereby detailing on the virtualization techniques, components deployed, and computing capabilities. As shown in the table, two different virtualization techniques are used, Kernel-based Virtual Machine (KVM) at DTAG and OpenStack in MTA. The resources allocated to the different MNOs also vary depending on the MNO network, where for example the MTA network has been assigned a total of 56 dedicated vCPUs compared to 36 in the case of DTAG. Additionally, the memory and storage used for MTA and DTAG are also different, which allowed us to test how orchestrators perform in resource-constrained environments, and how the computing capabilities of underlying NFVI affect orchestration performance.

4.2 Testing strategies for orchestration KPIs

The testing strategies for collecting orchestration KPIs focus on: i) the time required for basic LCM operations to be performed by the platform on the testbed illustrated in Section 4.1, and ii) the associated overhead. First, the LCM operation latency is evaluated by summing the processing time at each component involved and the transmission time (communication time) required for communication among the components. As for the overhead for LCM operations, it is evaluated by considering the overall number of messages necessary to complete an operation. The tests include both local, i.e., single-domain instantiation, which is an operation performed within the same domain (e.g., a domain of one edge orchestrator), and cross-domain instantiation, which is an operation requested in one domain but performed on a different federated domain. In this evaluation, we focus on the service instantiation operation, which is more complex than the deletion operation because it requires interaction with the MEAO module and a higher computation effort at the NFV-LO side. In our tests, we assume that onboarding of service descriptors and service images in edge nodes has been already carried out in a previous step.

To provide a better understanding of the orchestration KPIs, we present a simplified overview of the orchestration operations in figures 4, 5, and 6, detailing the flow of the main orchestration messages. These figures do not report standard UML sequence diagrams, since they would require more details, which would hinder figures' readability. In addition, in these figures, it is important to be able to appreciate the latency contribution associated with message propagation time. In fact, especially in the communication between different orchestration tiers or between remote peers of the same tier, it may result not negligible, whereas sequence diagrams usually neglect it. In particular, Fig. 4 illustrates the flow of messages required to instantiate a MEC, or in our context CCAM, application (which is also called network service, or NS, in the ETSI jargon [13]) within a single domain. The local NS instantiation is triggered by the domain NFV-SO towards the edge where the application will be executed. The orchestration latency can be measured as the time needed by the NFV-SOs to know that the application is up and running in the edge node identified by the local MEAO. In our proposal, before NS instantiation, it is necessary to create the inter-domain federation on the Or-Or reference point, exchanging the MLA descriptors between peer NFV-SOs. In turn, each NFV-SO needs to enforce the MLA towards the edge nodes. It is important to note that this procedure happens only once, e.g., during the bootstrap phase of the orchestrated edges platform when the interfaces between orchestration components are getting established.

From this time, an NS can be instantiated without further exchange of MLA descriptors. However, upon a change in policies between top-level and edge-level orchestrators, or a change triggered by adding new involved peers to other domains, it could be necessary to update the local MLA before service instantiation. In fact, the MLA descriptor includes information about peer NFV-LOs that can be directly contacted by the local NFV-LO in case of cross-border instantiation (see Fig. 5).

For this reason, in numerical results, we will also show the contribution to the overall orchestration latency due to the MLA update, to identify the worst case for the latency of NS instantiation.

To sum up, in order to quantify the instantiation latency KPI and the associated components (as described in Table 2), we refer to Fig. 4. In this diagram, it is easy to recognize three main contributors to this latency:

- The latency associated with the upload of the updated MLA descriptor from NFV-SO to the NFV-LO. This process is optional, depending on whether the updates in MLA are required or not.
- The latency associated with NS creation from NFV-SO, which is a step needed to maintain compatibility with ETSI NFV standards.
- The actual NS instantiation delay, triggered by the NFV-SO towards the NFV-LO and MEAO via the AAM.

The NS creation operation ("Create NS identifier" LCM operation [22]) has been introduced to keep backward compatibility with the ETSI NFV standard on NFV-SOL 005 interface [22]. This latency contribution is common to both the baseline and proposed approach and, as for the MLA descriptor instantiation, it counts only for the first time a service is *created* and a proper Id is returned to the NFV-SO. In fact, once the resource is present on the AAM, the NS can be instantiated and terminated multiple times, without the need for NS creation after each termination [16]. It is worth noting that it could be possible to incorporate the MLA descriptor update as an optional field of the NS creation operation, by extending the relevant API. In this way, the additional latency due to MLA descriptor upload from NFV-SO could be limited to the additional data carried in the create NS request, and the only newly added latency would be the enforcement of the MLA update between the AAM and NFV-LO, which is intra-edge and thus of almost negligible impact.

Table 2 – Deployment details in the cross-border trial sites.

MNO	Virtualization	Components	vCPU	RAM [GB]	Storage [GB]	Dedicated Fast data I/O	Remark
MTA	OpenStack	edge controller NFV_LO, MEAO, AAM	56	52	400	YES	vCPU dedicated
DTAG	KVM	edge controller NFV_LO, MEAO, AAM	24	36	320	YES	vCPU shared

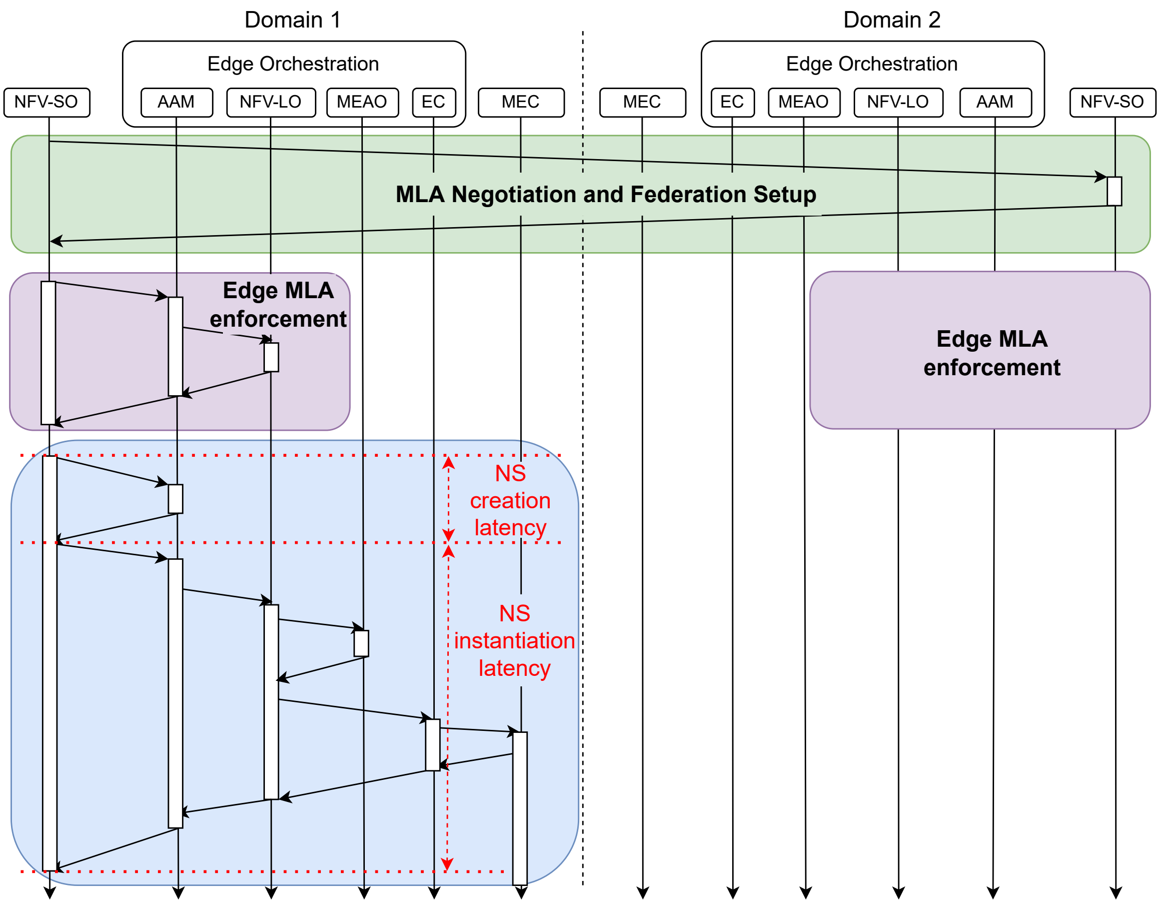

Fig. 4 – Single-domain NS instantiation triggered by the NFV-SO.

Furthermore, Fig. 5 illustrates the cross-border instantiation of an NS, or CCAM application, triggered by the MEAO via the Lo-Lo reference point. The goal of this operation is to address the movement of the vehicle across the border between two edge or administrative domains, thereby scaling the MEC application that is already running on the MEC platform of Domain 1 to another one running on the MEC of Domain 2, as described in our previous work [21]. The blue box in Fig. 5 indicates the procedure described in detail in Fig. 4. It is considered that MLA enforcement for intra- and inter-domain operations has already taken place. According to the information flow, the presence of the MLA delegates the NFV-LO complete autonomy to contact the peer NFV-LO for remote NS instantiation (e.g., cross-domain NS scaling operation). This implies a local instantiation in the peer domain and relevant notification to the peer NFV-SO that, in turn, will update the NFV-SO of Domain 1 to keep it informed about ongoing inter-domain scaling. Although the signaling necessary to keep the information about running NSs consistent across the domain has to include a final exchange on the Or-Or reference point, the measurement of orchestration latency does not include these last steps. In fact, the latency can be considered as the time needed by the running MEC or CCAM application in Domain 1 to be informed about another peer application in Domain 2, to which is possible to send packets on the data plane. This is highlighted by red arrows in Fig. 5 and Fig. 6.

Finally, Fig. 6 illustrates the baseline system, built according to the ETSI specifications for multi-domain operations [23]. In this case, no autonomy delegation is enforced via MLA, thus, upon the trigger received from the MEAO, the NFV-LO has to assume the role of a *nested* NFV orchestrator, which asks the orchestrator with the *composite* role, i.e. its NFV-SO, to trigger the scaling of the instantiated NS in the peer domain. This happens on the Or-Or reference point between the NFV-SO of Domain 1, acting again as *composite* NFV Orchestrator (NFVO), towards the NFV-SO of Domain 2, acting as

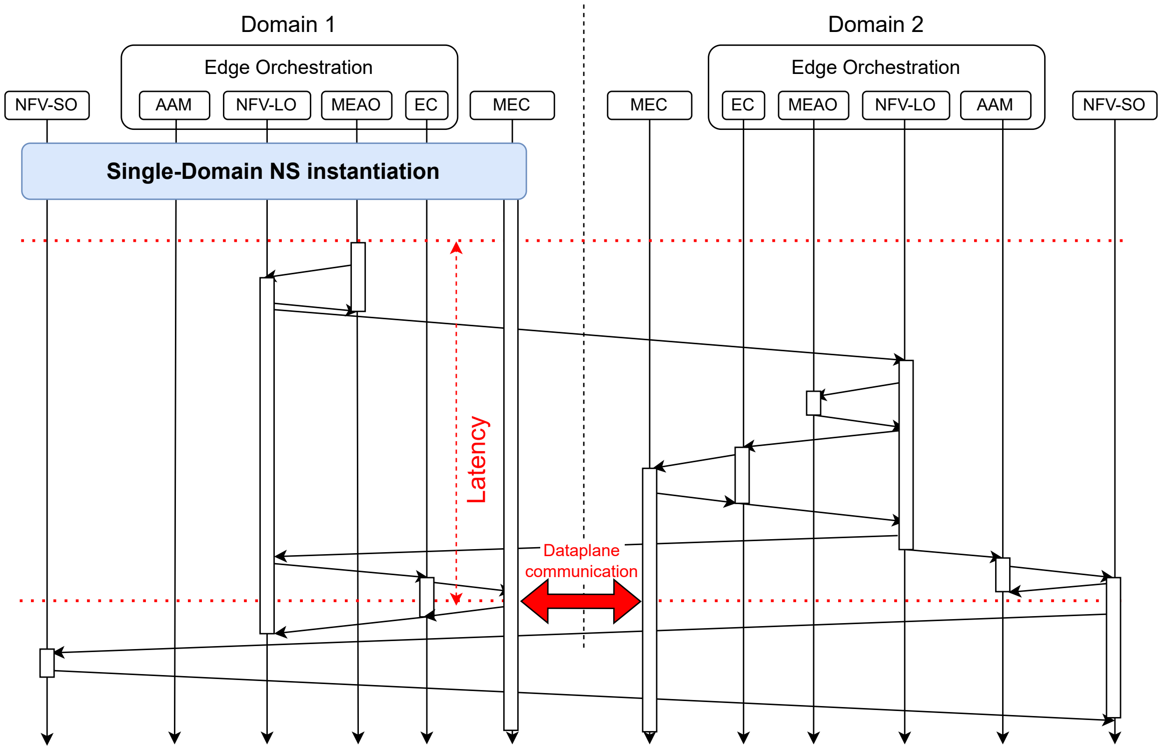

Fig. 5 – Cross-domain NS instantiation by the MEAO via Lo-Lo reference point.

nested NFVO. In turn, this last one, acting this time as *composite* NFVO towards the orchestration of controlled edge, i.e. NFV-LO of Domain 2, acting as *nested* NFVO, triggers the instantiation of the target NS. This info propagates back via Or-Or and then intra-domain to the NFV-LO that initiated the LCM procedure, which informs the local EC of the data plane reference point on which the local MEC application can send data plane packets. This concludes the evaluation of the LCM latency, even if some additional messages have to be exchanged to properly close the signaling exchanges.

4.3 Testing strategies for client-to-edge communication

In Kubernetes, a POD is the smallest unit that can be deployed and managed by the platform. By default, each POD only has a single network interface, which may not be suitable for telco services that require multiple interfaces for different types of traffic, such as management, data plane, and control. However, there are ways to work around this limitation by using network policy, network segmentation, or by using multi-NIC PODs. These methods can be used to create multiple virtual network interfaces within a POD and route traffic to them as needed. Additionally, Kubernetes also allows the use of CNI plugins that can provide more advanced networking features such as layer-2 segmentation and SR-IOV support. In this regard, the edge controller implements a data plane acceleration by defining additional interfaces based on Open-VSwitch (OVS) by extending the Kubernetes CNI for low latency communication. Fig. 7 depicts the evaluation setup for the two types of interfaces, i.e. the native flannel interface and the OVS-based additional interface managed by the edge controller.

4.4 Testing strategies for edge-to-edge service communication

The evaluation setup of Round Trip Time (RTT) measurements presented in this section is based on Fig. 8, and it is used to compare the delay differences between the service-based communication path (via NodePort) and the data plane communication path (via Fast Data I/O) for inter-MNO edge services communication. This measurement is important because it provides insight into the performance of these two different communication paths, which are key enablers for Edge/MEC applications. The service-based communication path is typically used for control and management traffic, while the data plane communication path is used for high-speed data traffic. The RTT measurements can be used to determine which path is more suitable for a given application or service, based on the specific requirements for delay, throughput, and reliability.

5. VALIDATION

5.1 Evaluation of orchestration KPIs

In this section, we present the results for latency measurements associated with the LCM operations. In order to highlight the benefits introduced by the proposed solution, which includes not only orchestration hierarchy but also delegation for autonomous operation at lower

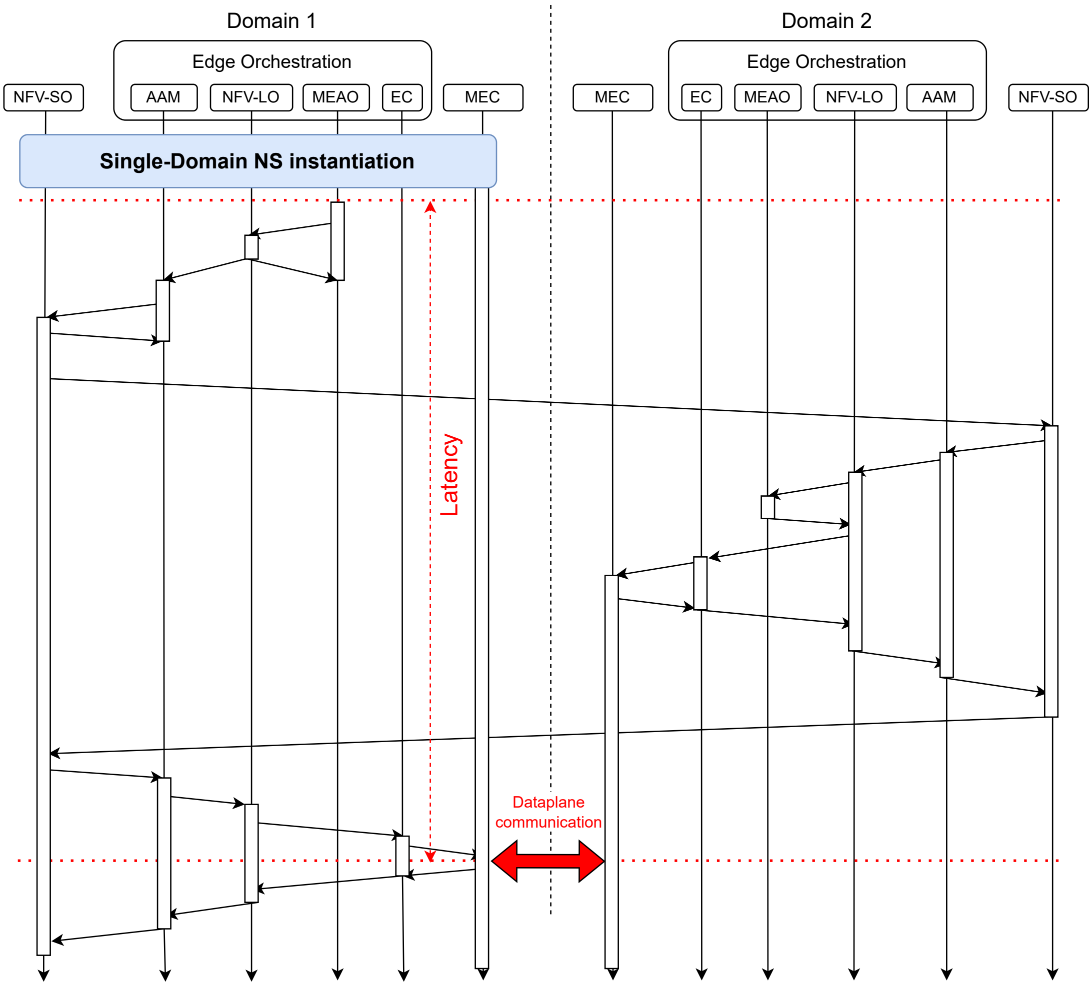

Fig. 6 – Cross-domain NS instantiation by the MEAO via Or-Or reference point.

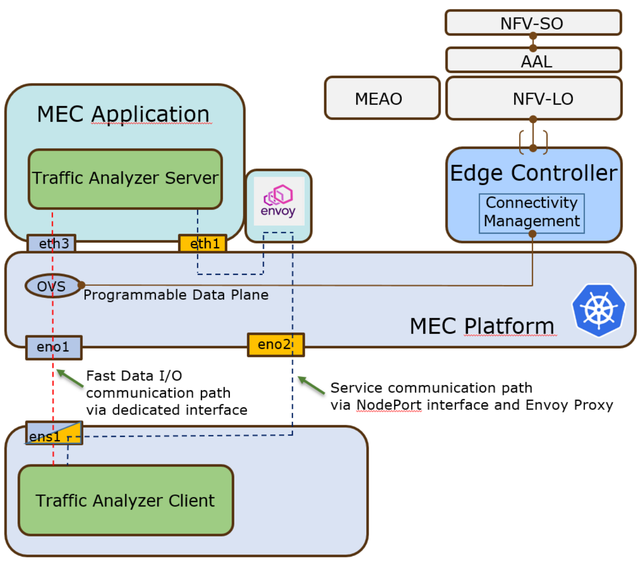

Fig. 7 – CNI extension evaluation setup for the client-to-edge *data plane* communication.

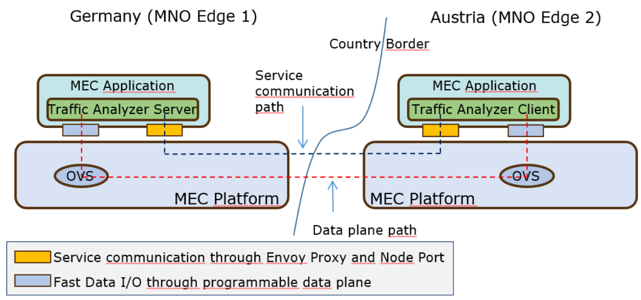

Fig. 8 – The measurement setup between DTAG and MTA MNOs in the edge-to-edge service case.

levels, we compare the performance of the proposed solution (labeled *w/ MLA*) with those of a baseline version (labeled *w/o MLA*). This includes single-domain NS instantiation (as described in Fig. 4) as well as cross-border LCM operations associated with *w/ MLA*, presented in Fig. 5, and those carried out according to the baseline version *w/o MLA*, presented in Fig. 6. Evaluation of performance is carried out on the testbed described in Section 4.1.

Fig. 9 shows the impact of the three components (optional MLA update, one-time NS creation, and NS instantiation) for the two domains considered in this work, reported on the abscissa. There are a number of comments associated with this figure. The first is that the actual NS instantiation latency is the dominant contribution to the overall latency with respect to NS creation and optional MLA update, which set up the worst-case latency for the proposed *w/ MLA* approach. Thus, the legacy procedure associated with ETSI NFV [13, 22] NS creation and novel addition as a by-product of this proposal including MLA descriptor update have a less important impact on the overall latency. In summary, the additional delays associated with the introduction of the delegation are not only optional but also quite limited. However, although Fig. 9 provides information about the different contributions to latency, it does not justify the significant difference between the delay measured in the DTAG and MTA trial sites. In this regard, Fig. 10 provides additional insights. In fact, it shows that while the communication delays (Round-Trip Time (RTT) between NFV-SO and NFV-LO plus small latencies due to intra-edge communications) associated with the two testbeds are nearly the same, the processing time for the DTAG trial site is significantly higher than those measured in the MTA one. This is due to the amount of computing resources reserved in MTA for these operations, which are significantly higher than those in DTAG, as shown in Table 2. In addition, computing resources in the DTAG node are shared with other services running in the same node, whereas those reported for MTA are exclusively reserved for the considered trialing purposes. The consequence is that in a well-provisioned node (i.e., the MTA one), the communication delay dominates the processing one, whereas, in a setting with strong resource overbooking (the DTAG one), processing time can be even twice the contribution associated with intra-domain delays.

To complete our evaluation, we consider the number of signaling messages as a metric defined as the localization gain (Table 1). As expected, during NS instantiation, we do not achieve any gain, since MLA upload implies additional messages, as shown in Fig. 11. However, the additional messages are only two for intra-domain signaling (NFV-SO - AAM) and the additional two for intra-edge signaling (AAM - NFV-LO). As discussed before, since MLA upload can be implemented as an optional field of the "Create NS identifier" LCM operation, the final difference would be just two intra-edge messages, which can be considered negligible. Thus, the localization loss due to the delegation procedure is practically negligible for both latency and overhead.

Once an NS is instantiated and a MEC application is up and running, an interesting operation for evaluating the benefit introduced by our solution is the cross-border scaling operation, illustrated in Fig. 5 for *w/ MLA* approach and in Fig. 6 for the baseline one, respectively. Similar to what is done with the NS instantiation, we evaluate the localization gain in terms of both orchestration latency and the number of exchanged messages. The impact of our proposal on orchestration latency is reported in Fig. 12. As in the previous analysis, we provide results for both approaches (*w/ MLA*, labeled "Lo-Lo" in the figure vs. *w/o MLA*, labeled "Or-Or") and for both directions, in order to also analyze the impact of the different computing capabilities of involved platforms.

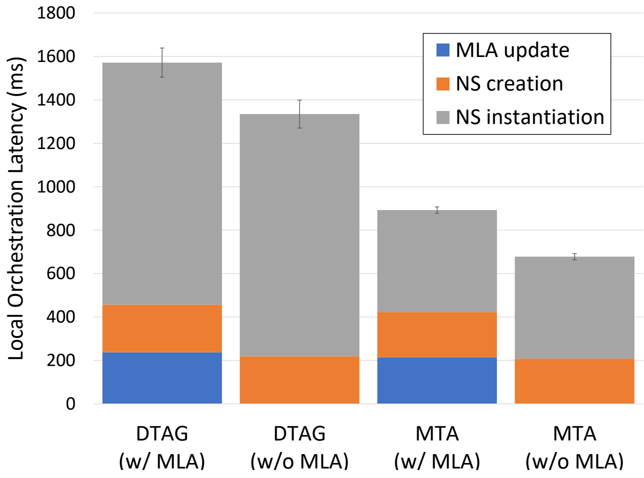

Fig. 9 – Contributions associated with MLA upload, NS creation, and actual NS instantiation in the latency for local instantiation of an NS from the NFV-SO in different clusters; 95% confidence intervals are provided.

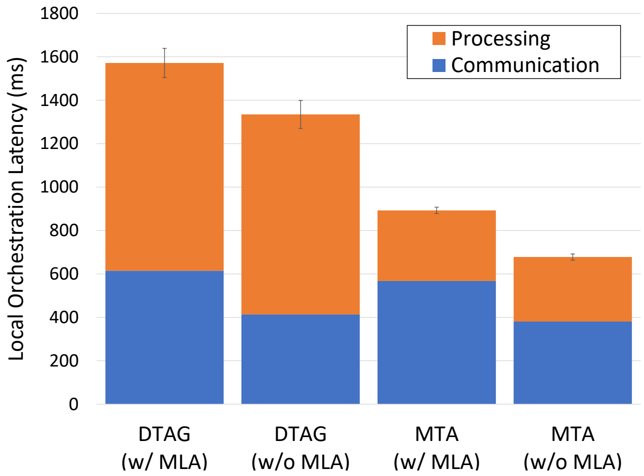

Fig. 10 – Contributions associated with communications and processing in the latency for local instantiation of an NS from the NFV-SO in different clusters; 95% confidence intervals are provided.

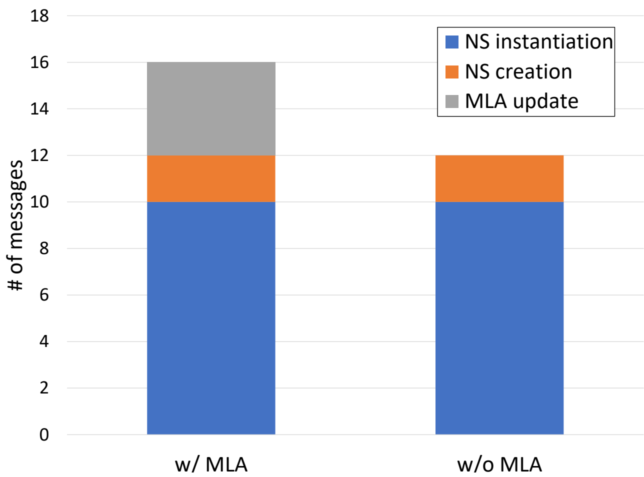

Fig. 11 – Amount of signaling messages associated with local instantiation of an NS from the NFV-SO.

Furthermore, Fig. 12 shows that orchestration latency reduction due to the usage of the Lo-Lo reference point is significant. It implies a localization gain (latency reduction) of about 2x in the direction MTA→DTAG, and even 7x in the reverse direction. From the analysis of the figure, it is evident that the gain associated with a reduction of long RTT communication is really significant, but that further important contribution is associated also with the savings in processing time. In fact, the communication delay decreases from ≈500-600 ms to ≈70 ms, due to the absence of intra-domain communications (RTT slightly less than 200 ms) and inter-domain ones (RTT of ≈75 ms). But the large difference between baseline and *w/ MLA* strategies is mainly due to the significantly fewer operations to carry out before the MEC application in the peer domain is up and running and able to communicate with that already instantiated in the other domain, thus with a definitely lower processing time. However, again the amount and type of computing resources allocated in the two testbeds play an important role. In fact, in the MTA→DTAG direction with "Lo-Lo" approach, the only time-consuming operation is the local NS instantiation in the peer domain (i.e. DTAG), which is higher than the MTA counterpart, as can be appreciated by the figure. Assuming that a well-provisioned node is the default choice in operation, the overall orchestration latency is well below 0.5 s, which is a good result, considering an RTT on the Lo-Lo in the order of 60 ms from real measurements.

However, when considering the baseline strategy using the Or-Or reference point to carry out this LCM operation, things become more complex. In fact, while it is true that the MTA node is computing-wise more powerful, its impact emerges in different ways, since the time-consuming operations concurring to the overall orchestration latency are mainly two: i) notification of AAM to the NFV-SO of the need to scale the current MEC application on the MEC node of the peer domain, including the "creation" (with ETSI meaning) of a new NS instance, and ii) instantiation of the MEC application on the peer edge node. In the MTA→DTAG direction, the notification happens in MTA, thus its impact on latency is not so large, and the dominating contribution is that of NS instantiation in DTAG. However, in the DTAG→MTA direction, it is exactly the contrary, and the resulting overall contribution to the latency of AAM notification in DTAG makes the two orange bars quite close (about 1.7 s for DTAG→MTA and 1.4 s for MTA→DTAG). In summary, the benefit of using a delegation strategy proves to be important in terms of latency savings.

Concerning the impact of operation being directly performed via the Lo-Lo reference point, on the signaling overhead, we can notice another gain due to the localization of operations brought by MLA adoption. In particular, Fig. 13 shows that our solution implies significantly fewer messages on the intra-domain interface and only two extra messages on the Or-Or, in addition to fewer messages intra-edge. In total, the localization gain for the overhead is about 24%, which is noticeable.

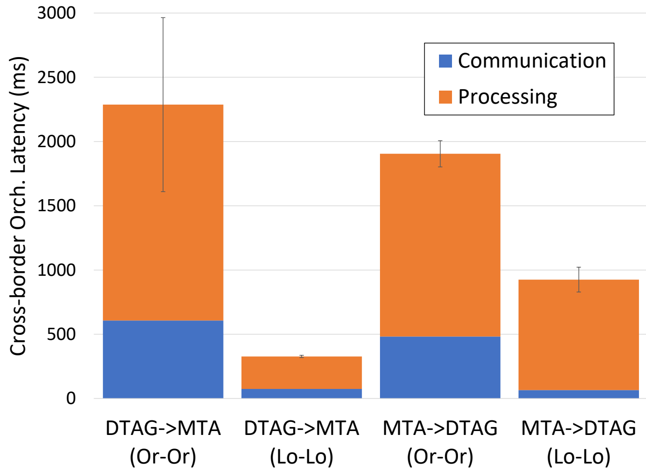

Fig. 12 – Latency for cross-border instantiation of an NS triggered by the MEAO; 95% confidence intervals are provided.

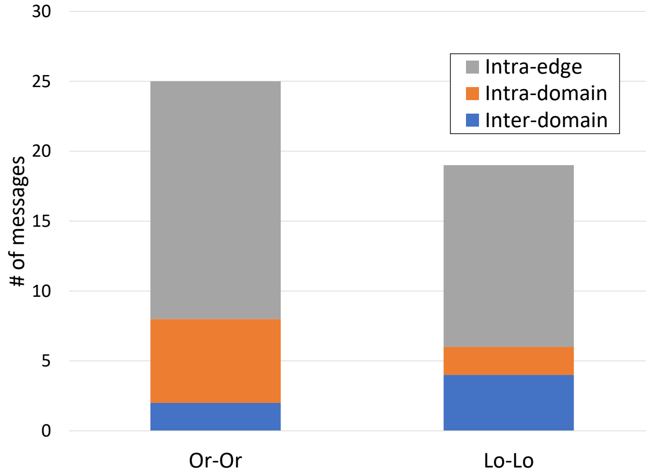

Fig. 13 – Amount of signaling messages associated with cross-border instantiation of an NS triggered by the MEAO.

5.2 Evaluation of client-to-edge data plane latency

The evaluation of the two types of networks is performed by running a Kubernetes cluster and an external client. The edge controller's Northbound Interface (NBI) deploys a service POD with an additional interface, eth3, for the data plane traffic, and the default CNI flannel deploys and manages the service-based interface, eth1, for the service-based communication. The service POD runs an Iperf service for TCP, and the external client creates an Iperf connection using different paths: via the NodePort and via the Fast Data I/O. This allows us to compare the RTT delay differences between the two paths and to evaluate the performance of the NodePort managed by the CNI flannel (for Service-Based communication) with an Envoy sidecar and Open-VSwitch-based fast data I/O (for data plane) managed by edge controller connectivity manager. The results of this evaluation can be used to determine which path is more suitable for a given application or service, based on the specific requirements for delay, throughput, and reliability.

Table 3 compares the RTT from the MEC host for two

Table 3 – MEC host to application service RTT using ping variation.

Measurement PING	Min RTT [ms]	Average RTT [ms]	Max RTT [ms]	Mdev_RTT [ms]
Native CNI for SBI	0.032	0.070	0.283	0.024
FDIO for data plane	0.025	0.050	0.209	0.031

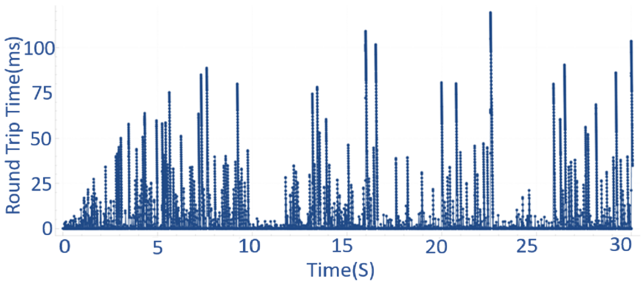

Fig. 14 – Round-trip time delay: NodePort with Envoy.

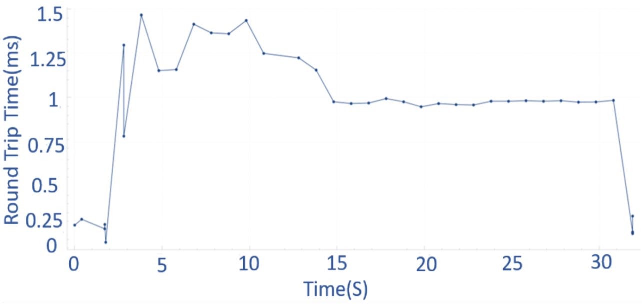

Fig. 15 – Round-trip time delay FDIO.

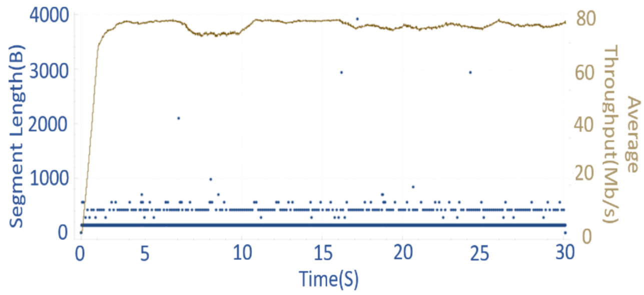

Fig. 16 – Throughput NodePort with Envoy.

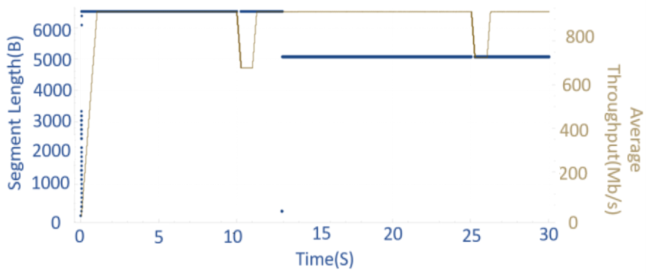

Fig. 17 – Throughput FDIO.

network paths: the native flannel-based CNI and the FDIO paths. The table shows that the average RTT for the service path is 0.07 ms, while it is 0.05 ms for the FDIO path. The FDIO path performs better in terms of both minimum and maximum RTT values, however, the mean deviation is higher than that of the native CNI interface.

As shown in Fig. 14 and Fig. 15 the maximum RTT for the FDIO interface is 1.5 ms with reasonable stability; on the other hand the maximum RTT for the NodePort could exceed 100 ms with quite some variations. Moreover, Fig. 16 and Fig. 17 depicts a throughput of 0.8 Gbps in the case of FDIO interface for TCP traffic over a 1 Gbps link and only 0.08 Gbps is achieved via the NodePort over the same link.

5.3 Evaluation of edge-to-edge data plane latency

Concerning the evaluation of edge-to-edge communication described in Section 4.4, Table 4 presents the results of measuring RTT using ping, between the Austrian (MTA) and German (DTAG) edges. The minimum RTT delay between the two MEC platforms is 11.255 ms for the dedicated data plane path and 26.309 ms for the SBA path. The average RTT for the SBI path is 26.384 ms and 11.604 ms for the data plane path. The maximum RTT for the SBI path is 26.459 ms and 16.802 ms for the data plane path. The mean deviation for the RTT on the SBI path is 0.173 ms, while it is 0.711 ms for the data plane path.

6. LESSONS LEARNED AND CONCLUDING REMARKS

In this section, we clarify how the validation results should be interpreted and how future developments can build on our project results. The experimental evaluation that we used for some parts of the validation is based on the developed enablers and the associated research prototype for the federated orchestration of distributed edges. The main objective is to prove the feasibility and value of the specified enablers to handle a challenging customer of a 5G ecosystem such as the automotive industry while focusing on mobility across a single administrative domain. Achieving absolute optimization in selected KPIs has been ruled out so far, as for the purpose of this article it is considered secondary and can, for example, leverage hardware acceleration for lower communication latency, and higher throughput as well as data analytics and machine learning. Summarized information on lessons learned during development, integration, and deployment, may help to find the right direction for further builds based on the specified enablers.

The availability of Open Source Software (OSS) for management and orchestration as well as for container virtualization was a starting point for the specifications and later development of the project. While popular orchestration frameworks such as OSM and ONAP are

Table 4 – RTT measurement results for data plane path and service-based communication path.

Measurement PING	Min RTT [ms]	Average RTT [ms]	Max RTT [ms]	Mdev_RTT [ms]
DTAG-MTA (SBI path)	26.039	26.284	26.459	0.173
DTAG-MTA (Data plane path)	11.255	11.604	16.802	0.711

well suited for the management of end-to-end systems, they are lacking support for automated edge-to-edge optimizations, which is particularly needed in an agile environment. The inter-connect and inter-play of different systems for mobility management (5G system), edge computing (MEC), edge services and data analytics, transport network, as well as MANO, is now better understood and a prerequisite for the automation of the overall ecosystem in the view of achieving the best match between resources utilization, performance, and experienced quality. Various enablers have been specified, developed, tested, integrated, and trialed, throughout various phases of the 5G-CARMEN project. Some enablers in the context of federation and orchestration of distributed edges in a 5G ecosystem have been briefly described in this article while a reference is being provided to an external source of previously published evaluation results. Recent results are described in more detail. This makes the article a comprehensive report on the targeted core concepts and enablers.

Distributed orchestration: The system performance gains from the specified two-tier edge orchestration architecture as tasks can be localized when appropriate and load can be distributed. The distribution of orchestration functions and the localization of edge orchestration tasks while enabling controlled orchestration interfaces between edge orchestration functions results in an improvement of the overall latency figures for the lifecycle management of edge services, as presented in this article. Further impact of the presented hierarchical orchestration system on distributed load and associated costs for orchestration at computational (CPU resources), networking and energy levels have been determined in an experimental system as well as analytically, and previously presented in [2].

Controlled edge autonomy and localization gain: In the view of maintaining ETSI NFV principles, the specified concept of MLAs keeps the top-level orchestrator, i.e., NFV-SO, as a root function to determine the level of autonomous operations at and in between edge orchestration functions. Negotiation and alignment of MLA policies for localized orchestration and cross-domain edge level orchestration (via Lo-Lo reference point) are performed between the two involved operators and their associated NFV-SO functions.

Separate treatment of network traffic for data plane and service mesh: While Kubernetes OSS includes support for service-based communication, e.g., through Envoy communication proxies, the benefit in adding enablers for separate treatment of network traffic for the communication between edges services (service mesh) and (mobile) data plane traffic has been recognized and realized in this project. The KPIs associated with network traffic can be different, hence different network traffic paths and technologies between Kubernetes PODs running edge services and the physical network infrastructure help to implement the required service-level objectives. In the presented research prototype, technology for the communication in a service mesh has been enriched with a fast data path to reduce the dependency on costly overlay solutions for networking in Kubernetes, i.e. node ports with IP address sharing and address re-writing, packet encapsulation, etc.

Cloud-native design and development: Following cloud-native principles when developing both the orchestration system components (MEC application orchestration, NFV local orchestrators, etc.) and the vehicular services (e.g., Back Situation Awareness (BSA) service [24]), and leveraging REST-based interfaces for inter-component interaction, significantly facilitated the deployment of those components on distributed edge platforms (either on bare metal or inside dedicated VMs) and their integration, thereby following the same procedure of deployment (e.g., the same YAML descriptors for Kubernetes component deployment). For instance, once deployed and tested on one of the associated MEC platforms (e.g., DTAG MEC), all aforementioned components were easily replicated to the remaining two platforms (e.g., MTA and TIM MEC platforms).

Resource selection and service placement algorithms: Taking multiple metrics into account when making decisions on placing vehicular edge service deployments on a particular MEC platform is essential for achieving QoS guarantees. One example of such a multi-criteria decision-making method could be the Technique of Order Preference Similarity (TOPSIS), which has been used in the design of the MEAO. The decisions on where to place the vehicular edge services (on which MEC platform), and where to relocate service due to mobility or changes in resource availability, should be made in a robust manner, thus taking into account: i) resource consumption of the underlying NFV resources, ii) mobility of connected vehicles (their current location, speed, heading), iii) coverage area of edge orchestrators (service and topological area of NFV-LOs and MEAOs where their orchestration decisions apply), iv) mobility events retrieved from 5G core (e.g., via NEF), among others.

Data analytics and the role of (edge) services: Embedding a certain level of intelligence into vehicular edge application services is important as it enables anticipation of different events on the roads (the route, speed, or heading of a connected vehicle), and as such, the service could help orchestration systems to make more efficient decisions on service scaling or relocation procedures towards improving service continuity (even in cross-border scenarios). In the scope of the project, we have demonstrated this capability of improving edge-to-edge service deployment for supporting service continuity, by allowing a developed and deployed edge service, that periodically exchanges ITS messages with moving vehicles, to issue notifications to edge orchestrators, informing them about a vehicle leaving their respective coverage area. Taking into account such a notification, edge orchestrators are able to proactively deploy the same instance of the service in the adjacent edges, and prepare vehicles for reconnecting from the source to a target edge service instance, which goes beyond standard handover procedures.

Mutual load awareness at system components: In line with what is stated above, it is important to also enable vehicular edge services to dynamically adjust their performance based on the computational load on the underlying NFV infrastructure where they are running (e.g., Kubernetes environment on the MEC platforms), i.e., to make them edge-aware. One pragmatic approach to enable such edge awareness is to leverage pub/sub mechanisms (e.g., ZeroMQ, ActiveMQ, RabbitMQ, Kafka), and to allow edge controllers and orchestrators to publish their monitoring data (CPU/memory load of the underlying platform), while edge services subscribe to various topics and collect those measurements in real time.

Micro-service-based software design and development: The design of vehicular edge services that are tailored to run on the edges of 5G ecosystems should follow the micro-service-based approach, where the overall service logic is split into several loosely-coupled micro-services that are deployed as e.g., containers and as such, orchestrated as Kubernetes PODs. This design enables orchestration decisions to be applied on a micro-service basis, resulting in less time needed for more fine-granular orchestration decisions and operations, which could have an impact on the service performance as well (less time needed to scale up one piece of service, which may result in unnoticeable downtime of the overall service, compared to the scenario of scaling up the whole service). Such design also enables debugging on per micro-service/software component level.

7. CONCLUSION

In this paper we presented the design and a developed experimental prototype for the orchestration of distributed 5G edges, which has been integrated with three MNOs' production networks for automotive cross-border trials in the EU project 5G-CARMEN. The automotive industry represents a challenging customer of a 5G ecosystem, with high demand on service quality and continuity even when moving across country borders and performing handovers to a different MNO. While the 5G ecosystem comprises many contributing sources that have an impact on the final service performance and experienced service quality, the two-tier orchestration system that we described along with the enablers, tackle the provisioning and lifecycle management of automotive services at MNOs' edge computing resources while improving latency and service continuity figures for cross-border CCAM. The article is concluded with a comprehensive summary of lessons learned during the specified system's development and experimentation while clarifying how the experimental results should be interpreted and treated in follow-up research and development that build on top of the presented results.

8. ACKNOWLEDGEMENT

This work has been performed in the framework of the European Union's Horizon 2020 project 5G-CARMEN co-funded by the EU under grant agreement No. 825012. The views expressed are those of the authors and do not necessarily represent the project.

REFERENCES

[1] Amdocs. *New Approaches Address End-to-End Network and Service Orchestration (E2ENSO) Challenges*. Accessed: 2023-January-19.

[2] Nina Slamnik-Krijestorac, Girma M. Yilma, Marco Liebsch, Faqir Zarrar Yousaf, and Johann Marquez-Barja. "Collaborative orchestration of multi-domain edges from a Connected, Cooperative and Automated Mobility (CCAM) perspective". In: *IEEE Transactions on Mobile Computing* (2021), pp. 1–1. DOI: 10.1109/TMC.2021.3118058.

[3] C. Hong and B. Varghese. "Resource Management in Fog/Edge Computing: A Survey on Architectures, Infrastructure, and Algorithms". In: *ACM Comput. Surv.* 52.5 (Sept. 2019). doi: 10.1145/3326066. ISSN: 0360-0300.

[4] S. Fu, F. Yang, and Y. Xiao. "AI Inspired Intelligent Resource Management in Future Wireless Network". In: *IEEE Access* 8 (2020). doi: 10.1109/ACCESS.2020.2968554, pp. 22425–22433.

[5] ETSI. "Multi-Access Edge Computing (MEC); Framework and Reference Architecture". In: *ETSI ISG MEC, ETSI GS MEC 003 V2.1.1* (2019).

[6] K. Katsalis, N. Nikaein, and A. Edmonds. "Multi-Domain Orchestration for NFV: Challenges and Research Directions". In: *2016 15th International Conference on Ubiquitous Computing and Communications and 2016 International Symposium on Cyberspace and Security (IUCC-CSS)*. doi: 10.1109/IUCC-CSS.2016.034. 2016, pp. 189–195.

[7] T. Taleb, I. Afolabi, K. Samdanis, and F. Z. Yousaf. "On Multi-Domain Network Slicing Orchestration Architecture and Federated Resource Control". In: *IEEE Network* 33.5 (2019). doi: 10.1109/MNET.2018.1800267, pp. 242–252.

[8] M. Efthymiopoulou, K. Ramantas, N. Vesselinova, M. M. Alam, L. Sanabria-Russo, and C. Verikoukis. "Multi-Domain Architecture for CAM Service Delivery Across Borders". In: *2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*. doi: 10.1109/MeditCom49071.2021.9647459. 2021, pp. 106–108.

[9] C. Mannweiler, L. C. Schmelz, S. Lohmüller, and B. Bauer. "Cross-domain 5G network management for seamless industrial communications". In: *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*. doi: 10.1109/NOMS.2016.7502917. 2016, pp. 868–872.

[10] G. Baggio, A. Francescon, and R. Fedrizzi. "Multi-domain service orchestration with X-MANO". In: *2017 IEEE Conference on Network Softwarization NetSoft)*. doi: 10.1109/NETSOFT.2017.8004259. 2017, pp. 1–2.

[11] V. Sciancalepore, C. Mannweiler, F. Z. Yousaf, P. Serrano, M. Gramaglia, J. Bradford, and I. Labrador Pavón. "A Future-Proof Architecture for Management and Orchestration of Multi-Domain NextGen Networks". In: *IEEE Access* 7 (2019). doi: 10.1109/ACCESS.2019.2923364, pp. 79216–79232.

[12] Q. Yuan, J. Li, H. Zhou, G. Luo, T. Lin, F. Yang, and X. S. Shen. "Cross-Domain Resource Orchestration for the Edge-Computing-Enabled Smart Road". In: *IEEE Network* 34.5 (2020), pp. 60–67.

[13] ETSI. "Network Functions Virtualisation (NFV); Management and Orchestration". In: *ETSI ISG NFV, ETSI GS NFV-MAN 001, V1.1.1* (2014).

[14] 3GPP. "Technical Specification Group Services and System Aspects; System Architecture for the 5G System (5GS) Stage 2". In: *3GPP TS 23.501 V16.6.0* (2020).

[15] 3GPP. "Technical Specification Group Services and System Aspects; Procedures for the 5G System (5GS) Stage 2". In: *3GPP TS 23.502 V16.6.0* (2020).

[16] M. Femminella and G. Reali. "An edge abstraction layer enabling federated and hierarchical orchestration of CCAM services in 5G and beyond networks". In: *ITU Journal on Future and Evolving Technologies* 3.1 (July 2022). doi: 10.52953/lnav1342, pp. 58–80.

[17] ETSI. "Mobile Edge Computing; Market Acceleration; MEC Metrics Best Practice and Guidelines". In: *ETSI GS MEC-IEG 006 V1.1.1* (2017).

[18] 5G-CARMEN. "Deliverable 4.2 - Advanced prototype for secure, cross-border, and multi-domain service orchestration". In: *H2020 5G-CARMEN Project Consortium* (2021).

[19] F. Z. Yousaf, V. Sciancalepore, M. Liebsch, and X. Costa-Perez. "MANOaaS: A Multi-Tenant NFV MANO for 5G Network Slices". In: *IEEE Communications Magazine* 57.5 (2019). doi: 10.1109/MCOM.2019.1800898, pp. 103–109.

[20] G. M. Yilma, U. Fattore, M. Liebsch, N. Slamnik-Kriještorac, A. Aviet, and J. M. Marquez-Barja. "5G AutoMEC -Boosting edge-to-edge service continuity for CAM in a sliced network". In: *IEEE 5G for Connected and Automated Mobility (CAM)* (2021).

[21] Girma M. Yilma, Nina Slamnik-Kriještorac, Marco Liebsch, Antonio Francescon, and Johann M. Marquez-Barja. "No Limits – Smart Cellular Edges for Cross-Border Continuity of Automotive Services". In: *IEEE Future Networks World Forum (FNWF)* (2021).

[22] ETSI. "Network Functions Virtualisation (NFV) Release 3; Protocols and Data Models; RESTful protocols specification for the Os-Ma-nfvo Reference Point". In: *ETSI ISG NFV, ETSI GS NFV-SOL 005 V3.3.1* (2020).

[23] ETSI. "Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Multiple Administrative Domain Aspect Interfaces Specification". In: *ETSI GS NFV-IFA 030 V3.6.1* (2022).

[24] Rreze Halili, Faqir Zarrar Yousaf, Nina Slamnik-Krijestorac, Girma M. Yilma, Marco Liebsch, Rafael Berkvens, and Maarten Weyn. "Self-correcting Algorithm for Estimated Time of Arrival of Emergency Responders on the Highway". In: *IEEE Transactions on Vehicular Technology* (2022). doi: 10.1109/TVT.2022.3209100, pp. 1–16.

9. ANNEX

ACRONYMS

3GPP 3^{rd} Generation Partnership Project
AAM Adaptation and Abstraction Module
AMF Access and Mobility Management Function
API Application Programming Interface
AS Application Service
CCAM Connected, Cooperative and Automated Mobility
CNI Container Networking Interface
DTAG Deutsche Telekom AG
EC Edge Controller
ETSI European Telecommunications Standards Institute
FA Federation Agent
FDIO Fast Data Input Output
FM Federation Manager
KPI Key Performance Indicator

KVM Kernel-based Virtual Machine
LCM Lifecycle Management
MANO Management and Orchestration
MEAO MEC Application Orchestrator
MEC Multi-Access Edge Computing
ML Machine Learning
MLA Management Level Agreement
MNO Mobile Network Operator
MTA Magenta Telecom Austria
NBI Northbound Interface
NFV Network Function Virtualization
NFVI NFV Infrastructure
NFV-LO NFV Local Orchestrator
NFVO NFV Orchestrator
NFV-SO NFV Service Orchestrator
NS Network Service
ONAP Open Network Automation Platform
OSM Open Source MANO
OSS Open Source Software
PCF Policy Control Function
PDN Packet Data Network
QoE Quality of Experience
QoS Quality of Service
RAN Radio Access Network
REST Representational State Transfer
RNIS Radio Network Information Service
RTT Round-Trip Time
SBI Service-Based Interface
SMF Session Management Function
SSC Service and Session Continuity
TIM Telecom Italia Mobile
TOPSIS Technique of Order Preference Similarity
UE User Equipment
UPF User Plane Function
uRLLC ultra-Reliable Low Latency Communication
VAS Value-Added Service
VNF Virtualized Network Function
ZSM Zero-touch Service Management

AUTHORS

Nina Slamnik-Kriještorac is a postdoctoral research fellow at the IMEC research center in Belgium and at the University of Antwerp. In July 2016, she obtained her master's degree at the Faculty of Electrical Engineering, University of Sarajevo. Nina obtained her Ph.D. degree in December 2022, at the University of Antwerp-imec, IDLab-Faculty of Applied Engineering. She authored or co-authored more than 30 publications in journals and international conferences. Her current research is mostly based on NFV/SDN-based network architectures with edge computing for vehicular systems, and the management and orchestration of the flexible and programmable next-generation end-to-end network resources and services, with a focus on edge applications.

Mauro Femminella received both a master's degree and a Ph. D. in electronic engineering from the University of Perugia in 1999 and 2003, respectively. Since November 2006, he is with the Department of Engineering, University of Perugia, where he is currently an associate professor. He is a co-author of more than 120 papers in international journals and refereed international conferences. His current research interests focus on artificial intelligence for network management, 6G systems, and molecular communications.

Girma M. Yilma is a senior research engineer at NEC Laboratories Europe in Heidelberg, Germany. He received his B.Sc. (2010) in electrical and computer engineering from Addis Ababa University, and his M.Sc. (2016) in telecommunications engineering from the University of Trento, Italy. His current research interest is focused on NFV MANO, orchestration, cloud-native networking, NFV/SDN-related technologies in the context of 5G networks and beyond.

Marco Liebsch is chief researcher at NEC Laboratories Europe GmbH and is working in the area of 5G mobility management, mobile edge computing, content distribution, mobile cloud networking, and software-defined networking. He received his Ph.D. degree from the University of Karlsruhe, Germany, in 2007. He has worked on different EU research projects and is contributing to standards in the IETF, ETSI, and 3GPP. He has a long record of IETF contributions as well as RFC, journal and conference publications.

Gianluca Reali has been an associate professor at the University of Perugia, Department of Engineering, Italy since January 2005. He received a Ph.D. degree in telecommunications from the University of Perugia in 1997. From 1997 to 2004 he was a researcher at the Department of Electronic and Information Engineering of the University of Perugia. In 1999 he visited the Computer Science Department at UCLA. His research activities include resource allocation over packet networks, wireless networking, network management, multimedia services, big data management, and nanoscale communications.

Johann M. Marquez-Barja is a professor at the University of Antwerp, as well as a professor at IMEC, Belgium. He is a member of ACM, and a senior member of the IEEE Communications Society, IEEE Vehicular Technology Society, and IEEE Education Society, where he participates on the board of the Standards Committee. His main research interests are 5G advanced architectures including edge computing, flexible and programmable future end-to-end networks, IoT communications and applications. He is also interested in vehicular communications, mobility, and smart city deployments. Prof. Marquez-Barja is co-leading the Citylab Smart City testbed, part of the City of Things program, and the SmartHighway testbed, both located in Antwerp, Belgium. Prof. Marquez-Barja has given several keynotes and invited talks at different major events, as well as having received 30 awards in his career so far, and co-authored more than 160 articles.