No limits – smart cellular edges for cross-border continuity of automotive services

Reference:

Yilma Girma M., Slamnik-Krijestorac Nina, Liebsch Marco, Francescon Antonio, Marquez-Barja Johann.- No limits – smart cellular edges for cross-border continuity of automotive services
2022 IEEE Future Networks World Forum (FNWF), 10-14 October, 2022, Montreal, QC, Canada - ISSN 2770-7679 - IEEE, 2023, p. 468-473
Full text (Publisher's DOI): https://doi.org/10.1109/FNWF55208.2022.00089
To cite this reference: https://hdl.handle.net/10067/1966420151162165141

No Limits – Smart Cellular Edges for Cross-Border Continuity of Automotive Services

Girma M. Yilma*, Nina Slamnik-Kriještorac†, Marco Liebsch*, Antonio Francescon‡, Johann M. Marquez-Barja†
* NEC Laboratories Europe GmbH, Germany
† University of Antwerp - imec, IDLab - Faculty of Applied Engineering, Belgium
‡ Smart Networks and Services, Fondazione Bruno Kessler (FBK), Italy

***Abstract*—One of the major challenges in 5G-based Cooperative Connected and Automated Mobility is to ensure continuity of a service that is deployed on the network edge and used by a moving vehicle. We propose enablers for smart cellular edges, which support service continuity in cross-border scenarios by the timely preparation of a service instance in an anticipated topologically closer target edge, and by connecting the vehicle to such service instance before the cellular handover occurs. In this paper, we use the edge data centers of a German and Austrian mobile operator to showcase two main enabling pillars for edge service continuity, i.e., i) transparent edge bridging by means of a programmable data plane to serve a vehicle from the target edge before the vehicle performs handover to a different operator, and ii) smart applications, which apply data analytics to boost orchestration decisions for target edge preparation.**

***Index Terms*—MEC, 5G, NFV orchestration, service continuity, edge computing, vehicular communications**

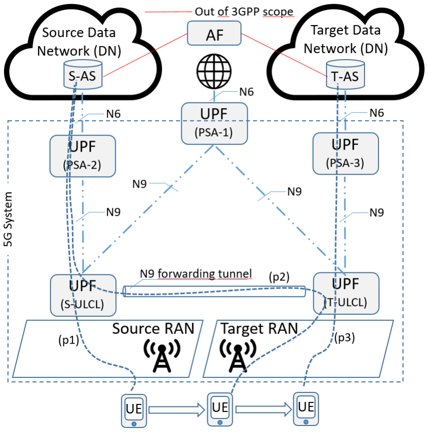

Fig. 1: Forwarding between ULCLs during network relocation [3].

I. INTRODUCTION

The 5^{th} Generation of a mobile communication system (5G), Network Function Virtualization (NFV), and edge computing, are promising enablers for connected cars and automated driving use cases. A key challenge in automotive industry is to keep a moving vehicle continuously connected to a service, which is deployed on a topologically close Multi-Access Edge Computing (MEC) site. This requires a session context relocation of the service to which a vehicle is connected, as well as the relocation of the network connection from the source edge service instance to the target one. In this case, the target service instance is being hosted by a topologically closer edge that may be provided by the same or different Mobile Network Operators (MNOs).

Data Network (DN) relocation has been addressed in the 3^{rd} Generation Partnership Project (3GPP) for the 5G System. Key mechanisms are solely based on the 5G control plane and User Plane Functions (UPF), such as the Packet Data Unit Session Anchor (PSA) and the Uplink Classifiers (ULCL), as well as on operations on the relevant architecture reference point in between them (N9), as depicted in Fig. 1. A moving User Equipment (UE) may be relocated from a source DN to a target DN to continue a service, whereas the 5G System can stepwise support the transition by i) assigning a new PSA that is closer to the target DN, and ii) configuring the other UPFs on the data path to serve as source- and target ULCL that forward the data plane between the UE's location and the connected DN. While the specification focuses on the role of the UPFs in such relocation procedure, the network segments and functions in between the PSA and the Application Service in the source and target DN (S-AS, T-AS), as well as the 3GPP Application Function (AF), are not considered.

For automotive services, we focus on Edge Applications (EdgeApps) as Application Services, that are deployed on MEC platforms in operators' distributed Data Networks. While the platform for such MEC is specified in the European Telecommunications Standard Institute (ETSI), a variety of implementations are being tested in production and experimental environments. Within the 5G-CARMEN project[1], we developed an orchestrated MEC platform based on the ETSI specifications [1], while providing extensions for network slicing [2], the integration and inter-working with edge orchestration layers for Life-cycle Management (LCM) of virtualized edge service instances, i.e., EdgeApps, as well as with the 5G cellular system and a programmable data plane leveraging Software Defined Networking (SDN). In this paper we describe the key enablers of the orchestrated and federated edge platform for supporting service continuity during cross-border mobility, and we present a proof-of-concept (PoC) development that has been deployed and tested in two MNOs' production network. These enablers are i) programmable edge data plane for the *transparent indirection* and associated steering of data plane traffic between a vehicle and a new edge service instance associated with the target MNO, denoted as Transparent Edge Bridging (TEB), just before the vehicle crosses the border and connects to a new MNO (detailed in Sec. III-A) , and ii) *smart EdgeApps*, which boost orchestration decisions by providing event notifications for service relocation to the edge orchestration layers (detailed in Sec. III-B). Such solution for edge relocation and service continuity applies to the network in between the 5G System and the edge services, hence complementing any DN relocation enablers from the 5G System without being dependent on those (Fig. 2).

We apply the strategy to transfer session states and bridge data plane packets associated with the service connection of a vehicle, from the currently used to the new edge and service instance at the target MNO even before the inter-MNO cellular

---

[1] H2020 5G-CARMEN, a cross-border automotive project funded by the European Commission: https://5gcarmen.eu

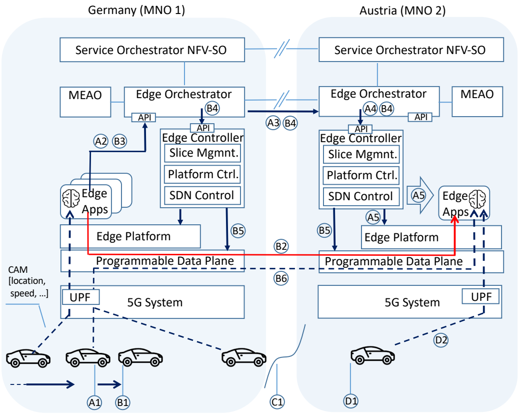

Fig. 2: Smart EdgeApps and TEB for service continuity.

handover happens. The experimental evaluation leverages the deployment of orchestrated and collaborating edge platforms at Deutsche Telekom and at Magenta Telekom (T-Mobile Austria GmbH) for the highway coverage at the German-Austrian border. To prove the concept without dependency on cellular network handover, the demo utilizes a virtual vehicle, which simulates vehicle mobility while connecting to the edge systems through the public Internet. In the remaining parts of the paper we summarize recent related work, describe the operational steps of the above-mentioned enablers and how the automotive use cases benefit from them, and provide insights into the experimental setup and associated results.

II. RELATED WORKS

Session and Service continuity, especially when edge computing is applied, has its own unique challenges to be addressed. According to [4], these challenges need to be addressed at MNO level but they even become more complex when multiple MNOs are involved, because of the inter-Public Land Mobile Network (PLMN) handover. In this regard the three main cross corridor projects 5GCroCo[2], 5G-CARMEN[3] and 5G-MOBIX[4] are heavily involved to analyze the challenges from various aspects. In [4], authors discussed that today's solutions, such as home routing, are applied where the public gateway between mobile radio network and other networks, such as Edge computing facilities and the public internet are located within the home network of a mobile user. Resulting in a long communication delay, which could break Service Level Agreements (SLAs) for latency critical use-cases.

Authors also discussed in detail the roaming impact to service and session continuity in the view of the Roaming regulations EU 2015/2120. The impact of roaming on service continuity is discussed from three distinct aspects of 5G roaming:

- Roaming between MNOs with 5G Non-Standalone (NSA), which is a scenario most likely to happen at the first phase of 5G deployments. This solution is expected to leverage existing 4G-LTE roaming performance, which has an end-to-end latency of at least 100 ms, which is far from the requirements of uRLLC use-cases.
- Roaming between MNOs with 5G Standalone (SA), where specialized SA features such as end-to-end slicing, and session and service continuity mode 3, can be leveraged to achieve the potential of improving performance & latency.
- Roaming between a 5G NSA network & a 5G SA network, where inter-working functionalities need to be supported as roaming extension interfaces will be required.

In [5], Kousaridas et al. discussed standardization challenges & enablers such as Edge computing, local breakout, MEC-5G-NFV Management & orchestration, for cross-border service continuity. The different technological enablers are solving part of the puzzle and are at different technological readiness level, which also requires significant standardization effort at The 3rd Generation Partnership Project (3GPP) RAN and SA groups, ETSI, and Internet Engineering Task Force (IETF).

Current state of the art is focusing mainly on discussing the role of UPFs in the relocation procedure, however other critical enablers for service continuity such as MEC, Orchestration, and 3GPP's Application Function are not considered. In this work, we leverage cloud native orchestration at the edge and novel concepts, such as transparent edge bridging, and edge-aware smart applications, to achieve service continuity for automotive use-case in a multi-MNO scenario.

III. CONNECTED EDGE SYSTEMS AND OPERATIONS

This section describes the principles and operation of Transparent Edge Bridging and of using SmartApps at cellular operators' network edges for cross-border continuity of automotive services. Furthermore, exemplary use cases are described, which can benefit from the proposed smart cellular edges.

A. Transparent edge bridging

The key functional components of the edge system are depicted in Fig. 2. Each edge system receives policies for EdgeApp LCM and orchestration from the domain's Service Orchestrator (NFV-SO). The Edge Platform is built on top of Kubernetes (K8s) and virtualized EdgeApps are instantiated as containers. The Edge Controller provides i) northbound REST APIs towards the edge level orchestration elements, the MEC Application Orchestrator (MEAO) and the local Edge orchestrator (EO), whereas ii) the southbound one applies platform-specific control for EdgeApp LCM, slice management, and data plane programming (SDN Controller). Two edge systems of different domains are connected at a federated management layer between the two NFV-SOs, as well as between the two EOs [6].

Since EdgeApps process the data plane packets of connected vehicles, e.g., Cooperative Intelligent Transport System (C-ITS) messages such as Cooperative Awareness Messages (CAMs), they can analyze those packets to support taking actions for service optimizations. In this evaluation, we leverage C-ITS message attributes at a smart EdgeApp that reveal the connected vehicle's geo-position, speed, and heading, to initiate the preparation of an EdgeApp on a different edge system, to which the vehicle will connect soon. The EO and MEAO expose an API dedicated for receiving such event notifications from EdgeApps.

Fig. 2 illustrates the system operations. While a vehicle exchanges data plane packets with an EdgeApp at MNO_1, this EdgeApp anticipates a cross-border movement and an associated handover to a new EdgeApp instance hosted by a different MNO (A1), which it notifies to the EO layer (A2). Through the management federation interfaces, the two edge systems ensure the provisioning of the target EdgeApp at MNO_2 (A3)-(A5). To connect the vehicle to the prepared EdgeApp before the cellular handover happens (B1), any existing application session state is transferred to the target

---

[2] https://5gcroco.eu/
[3] https://5gcarmen.eu/
[4] https://www.5g-mobix.com/

EdgeApp (B2). From that moment, the vehicle should be served by the target application, which is realized by the programmable data plane and initiated by the smart EdgeApp (B3)-(B5). Through a GRE tunnel between the two systems' programmable data plane, TEB is realized and packets are relayed between the vehicle and the target EdgeApp without the vehicle noticing it (B6). After handover to MNO_2 (C1), the vehicle can enforce a direct connection to the local EdgeApp (D1)(D2).

B. Smart edge applications

The *smartness* and *edge-awareness* of cloud-native Cooperative Connected Automated Mobility (CCAM) application services, or EdgeApps, is a particular design feature that we define and leverage in this paper as an orchestration booster. It enhances the decisions made by orchestration layers by allowing them to use the notifications generated by orchestrated EdgeApps themselves, and thus enabling them to retrieve some application-specific insights (e.g., change in route of vehicles connected to the service application, current positions of relevant vehicles, proximity to the country borders, detection of obstacles on the road, and network re-selection) that are usually not known by orchestrators [6]. In particular, the application can be considered as smart and edge-aware in the 5G CCAM context if its design allows it to be aware of:

- the *edge environment*, such as the elements of an orchestrated MEC system, e.g., MEAO and EO,
- the *other edge applications* that are relevant for their operation, such as other peering CCAM applications, or MEC Value-Added Services (VASs), as well as
- the *clients running in vehicles* that are connected to them and use their service, thus, providing them with real-time information about the movement of vehicles.

As mentioned above, such smart EdgeApps are capable of generating various important notifications that could improve and boost their own life-cycle management by enhancing the decision-making process performed by orchestration layers. The notifications derived from application-specific operations can be generated using either data analytics and/or different Machine Learning (ML) models that are executed by applications, and they refer to the processes that are specific for the application operation such as mobility of the vehicle, proximity from the border between two edge domains, or the border between two countries.

If we take a look at the features that make an EdgeApp smart (listed above), being aware of the edge environment allows EdgeApp:

- to retrieve the topological and service coverage of the orchestrators (e.g., coverage of one edge domain), which is a relevant input for determining the boundaries of the service regions covered by MEAO and EO, thereby used by smart edge applications to timely trigger their relocation e.g., if the vehicle is approaching the border between two edge domains, and
- to pass the notifications to the orchestration entities, which these entities can further use to optimize their orchestration decisions that trigger operations such as EdgeApp instantiation, scaling, migration, or termination, and thus, maintain Quality of Service (QoS) at a required level.

Further, awareness of the other edge applications, such as peering EdgeApps, enables extending the application service operation beyond one edge domain, as peering EdgeApps can use service-based interfaces to exchange application metadata, such as location/speed/heading of vehicles connected to them (mobility of the vehicles). The EdgeApps can get more insights into the radio network related information from a VAS such as Radio Network Information Service (RNIS), where the granularity can be adjusted per radio cell, or even User Equipment (UE), which helps EdgeApps to retrieve network connectivity information about a particular vehicle that is about to cross the border and re-select the network. Given such information, a smart EdgeApp can apply a suitable ML model to predict network re-selection that usually breaks the service connectivity, and then based on such prediction, it can proactively trigger EdgeApp instantiation/migration in the target domain. To retrieve a real-time update on the location/speed/heading of the vehicles connected to it, a smart EdgeApp leverages either a VAS such as MEC location service, or it is aware of the clients in the vehicles that are connected and share their location data within C-ITS messages.

C. Use cases

Given their sensitivity on latency and/or throughput, and significantly high mobility of their users (vehicles on the public roads and highways), in this paper we focus on vehicular services and use cases, and here we briefly show how different classes of such use cases can benefit from the concept of smart cellular edges towards achieving (cross-border) service continuity.

a) Emergency services: Given the high mobility of vehicles in agile environments such as highways, it is essential to extend the awareness about the emergency vehicles on the roads, as well as to distribute the corresponding notifications about various emergency events. Vehicular edge application services can support such emergency situations by extending the range of the necessary notifications to the zones that are kilometers away from the emergency event, thus providing the means to improve safety and to optimize the traffic. If emergency situations stretch over multiple cellular edges that belong to different MNOs, the optimal network connectivity of emergency/civilian vehicles to edge services need to be ensured, so that the notifications are not delayed. Thus, the concept of transparent edge bridging that we propose in this paper could be leveraged to proactively deploy service instances in the edge domains (relevant for the emergency event), based on the triggers coming from a smart EdgeApp, and to steer user's traffic from one edge service to another thereby always ensuring that the vehicles are connected to the most suitable EdgeApp (e.g., the ones that deliver packets with the lowest latency).

b) Maneuvering services: Through a dedicated service endpoint, car clients share information about car status and surroundings plus driver intentions; thus, a service running at MEC can provide suggestions to each client for the next car move. The number of car clients in the service-covered area affects the performance of Maneuvering services. The chance for an on-demand creation/deletion of service instances and dynamic management of the coverage area assignment to each running instance allows the Maneuvering services to rapidly adapt to the evolving situations and thus constantly meet the strict service requirements. E.g., EdgeApps may detect when the number of associated cars reaches a critical threshold and notify the Orchestration layer of such a situation to have a new instance of the service added. The original covered area is split between the original and the new instance. When the new instance is deployed, car information is automatically redirected to one or the other instance according to the coverage assignment.

c) Infotainment services: Unlike the previous examples of vehicular EdgeApps, infotainment application services such as video streaming also contain the application state, which needs to be transferred from one application instance to another when the vehicle is moving between cellular edge domains. To ensure the required levels of latency, jitter, and throughput, a moving vehicle should be connected to the optimal EdgeApp. Thanks to the smartness of EdgeApps, explained in Section III-B, the corresponding instances of video streaming service can be proactively deployed on adequate cellular edges knowing the movement of vehicle. Taking into account the location and radio network-related information, smart EdgeApp can trigger the state to be transferred in that case, so that a vehicle (e.g., passengers inside a car) can have an uninterrupted and a high quality video streaming experience even when traversing between edge domains.

IV. DESIGN PRINCIPLES

A. Design principles of the Orchestrated edge platform for CCAM

The design of the Orchestrated edge platform for CCAM follows the cloud native principles, hence functional elements are implemented as container-based pieces of software making a highly modular design. The modularity enables a mix and match of different open-source software solutions. For instance, the Service Orchestrator (NFV-SO) is based on existing ETSI Open-Source MANO (OSM)[5], for centralized (high level) orchestration tasks, and local orchestration tasks are handled by Kubernetes based Edge Orchestrator (NFV-LO). The Edge Controller is also built on top of the popular open-source container orchestration platform i.e., Kubernetes.

The interfaces between orchestration components (i.e., Or-Or, Lo-Lo, Or-Lo, Mv1, and NFV-LO - Edge Controller) are implemented following the service-based architecture as micro-services running as Kubernetes Pods. These interfaces use REST based communication. Kubernetes is used as the main platform for container orchestration hosting both, the Orchestration components, and the Edge Controller. The MEAO/NFV-LO and Mv1 are implemented as separate containers and run as Kubernetes Pods, thereby managing the MEC applications and services via each component REST APIs (for CRUD operations management between orchestration components) and a message broker (for interaction with smart apps). Similarly, the MEC applications and services are implemented as container applications in different Kubernetes Pods within each MEC host.

Application packages, i.e., Docker images, are saved in private or public repositories and application package descriptors are managed using helm charts. Notification handling functions can be defined in the orchestration modules and then bond to specific notification via a dedicated descriptor. This allows high flexibility in binding a given notification type, the module managing such a notification and the specific handling function. To enable separation of control and data plane traffic each Pod with an instance of a CCAM service application can be equipped with one or multiple customized network interfaces, such as for service-based communication and data sharing with other application instances, or for fast data plane I/O and associated low-latency communication with other application instances or service clients.

B. Proof-of-Concept smart edge application

With reference to the rationale behind the concept of smart EdgeApps, which is described in Section III-B, now we provide more insights into one example of such smart edge-aware application. The Back Situation Awareness (BSA) application service, defined in our previous work [7], is a CCAM EdgeApp that supports emergency situations on the roads by proactively informing civilian vehicles about an Emergency Vehicle (EmV), thereby distributing notifications about EmV's arrival so that civilian vehicles can timely clear the corridor and let the EmV pass through unhindered. The BSA application service i) extracts the location and speed of the EmV from the real-time CAMs received via 5G network, ii) defines dissemination areas on the road, and iii) calculates Estimated Time of Arrival (ETA) values for each of the areas along the route path, thereby allowing civilian vehicles to receive ETA notifications that are specifically calculated for their geocasted area (i.e., dissemination area).

Given such real-time information about the EmV's movement, as well as the topological coverage of edge orchestrators (i.e., coverage area of one MEC platform in the project trials), BSA application service performs analysis and calculates the proximity from the border between the adjacent edge/administrative domains (source and target domain). When the proximity is determined, the rule-based algorithm compares it with the predefined thresholds and decides whether a peering application instance is needed in the target domain even before the vehicle reconnects to the other network. In such a way, a proactive deployment of the target instance to which vehicle can connect as soon as it reattaches from one MNO's User Plane Function (UPF) to the other is achieved. Applying this concept to a more generic use, applications can notify orchestration entities about the need for i) an on-demand instantiation of peering application instances in other edge and administrative domains, ii) a proactive application-context relocation to support service continuity, iii) an on-demand application termination when its services are not needed anymore by the users, so that resources can be released. The *smartness* of BSA EdgeApp can be further improved by leveraging e.g., RNIS, as the decision to trigger instantiation of peering EdgeApp can be further optimized knowing the usage of radio resources and attachment to a particular network and radio cell. Finally, leveraging different publish/subscribe mechanisms in the design of both applications and orchestration entities, orchestrators are capable to subscribe to the notifications published by applications and improve their internal decision-making process based on such application-specific insights.

V. EXPERIMENTAL SETUP AND EVALUATION

A. Experimental setup

The experimental setup that we leveraged on to evaluate the transparent edge bridging-based service continuity consists of i) two MEC platforms, one for each country, i.e., respective MNOs DTAG and MTA, ii) orchestrated edge platform installed in a K8s cluster on each of these MEC platforms, iii) smart EdgeApp for BSA application service described in Section IV-B, and iv) python-based client application running at NEC Laboratories Europe GmbH (NEC) lab premises emulating a virtual vehicle that sends CAM messages.

The CAM messages that compose the traffic sent from a vehicle to an EdgeApp (upstream CAM traffic) are encoding relevant information (e.g., speed, and location) over UDP, thereby emulating the movement of the vehicle on the corridor between Germany and Austria, and providing continuous location updates to the smart EdgeApp. In the context of BSA application service, these location updates are used for calculating the estimated time of arrival of an EmV, which is further disseminated as notifications to the other vehicles in the corridor. In this evaluation, we showcase i) the proactive service instantiation based on the triggers/notifications generated by smart EdgeApps (Phase 1 in Fig. 3), and ii) maintaining service continuity when vehicle is crossing the border via creating programmable data plane (Phase 2 in Fig. 3).

---

[5] https://osm.etsi.org/

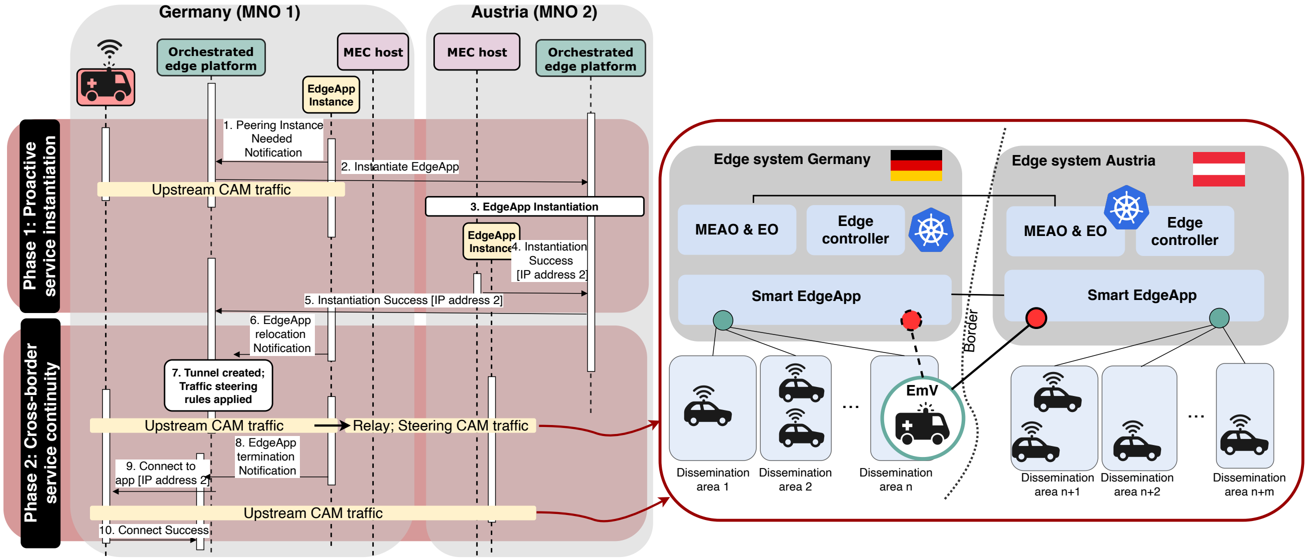

Fig. 3: Operations included in Proactive service instantiation and Cross-border service continuity: Proof-of-Concept.

a) Phase 1: Proactive service instantiation: To perform Phase 1 illustrated in Fig. 3, we use a BSA type of smart EdgeApp, which is containerized and deployed as K8s POD on the edge system in Germany. This EdgeApp notifies MEAO and EO (i.e., Orchestrated edge platform in Fig. 3); to instantiate peering EdgeApp in Austria (step 1, Fig. 3), using the data-analytics algorithm that determines the corresponding moment for proactive instantiation, so that all required resources are allocated on the target edge before EmV crosses the border. Afterwards, EO in Germany is sending the instantiation request to EO in Austria (step 2, Fig. 3), which then further uses Edge controller to instantiate peering EdgeApp (step 3). Once EdgeApp is up and running on the edge system in Austria (step 4), it connects to its peering instance from the source domain (step 5), and receives metadata based on which it can create notifications for dissemination areas in Austria. Upon instantiation, vehicle can re-attach from German to Austrian EdgeApp at any moment, which is in our case determined again, by the smart EdgeApp.

b) Phase 2: Cross-border service continuity: In Phase 2, the TEB procedure is applied, i.e., the procedure that enables a smooth re-attachment of vehicle from a smart EdgeApp in Germany to its peering instance in Austria. This is possible because of the programmable data plane of our orchestrated edge platform, which relays the packets sent from a vehicle to an EdgeApp while vehicle is connected to Austrian network (MNO 2) but still on the German side of the border. Once smart EdgeApp decides that relocation should happen (e.g., EmV close to the border between two countries), it sends notification to MEAO (step 5, Fig. 3), after which the tunnel is dynamically created and traffic steering rules applied (steps 6 and 7), as described in Section III-A. This way, EdgeApp that receives packets from EmV is a German one, and once EmV crosses the border, source EdgeApp (Germany) triggers EdgeApp termination (step 8), which breaks the tunnel, and terminates the source EdgeApp. Finally, the CAM traffic coming from the EmV is being sent directly to Austrian instance (steps 9 and 10). In Phase 2 shown in Fig. 3, we measure the latency at the client side, which in our case is a python-based UDP client application[6] running in the lab in Heidelberg, Germany. This application can be easily on-boarded to the vehicles, and used for sending CAMs from vehicle to EdgeApp, thereby measuring the real-time latency as a round-trip time.

B. Results and Discussion

To evaluate the performance of proactive EdgeApp service instantiation, we have tested i) the average processing delay of a smart EdgeApp algorithm, which is the time needed for the algorithm running in the BSA EdgeApp to make decisions about the peering EdgeApp instantiation, and EdgeApp relocation/termination, ii) the average instantiation delay calculated at the MEAO side as a time needed for MEAO to process the notifications coming from a smart EdgeApp and instruct EO and Edge controller to deploy the application service, iii) the average delay of updating the tunnel, which is the time that MEAO takes to process the relocation notification received from the EdgeApp and to trigger the updates on the tunnel that results in traffic steering from one domain to another, and iv) the average termination delay, which is calculated at the MEAO side as a time needed for MEAO to process the notifications coming from a smart EdgeApp and instruct EO and Edge controller to terminate the source application service, i.e., EdgeApp, break the tunnel, and allow vehicle to directly connect to the target EdgeApp.

The result shown in Fig. 4 provides an insight into the latency budget of processing notifications and deriving decisions at both EdgeApp and orchestration layers, showing how essential it is to perform these operations proactively, i.e., before they start affecting the user experience by increasing perceived latency or decreasing service reliability. For instance, taking into account the current location and speed of an EmV, algorithm running in the EdgeApp takes 190ms on average to derive a decision (e.g., whether a peering instance in the other domain needs to be instantiated, or a service needs to be relocated/terminated), and MEAO takes 10ms on average to process the notification received from the smart EdgeApp and to proceed with the peering EdgeApp instantiation. These 200ms do not affect the overall operation as both the decision-making and the EdgeApp instantiation are performed proactively, i.e., while the vehicle is still connected to the source EdgeApp. However, not applying such a proactive mechanism,

---

[6] Python-based UDP client configured to produce a response to every received packet, enabling measurements of a round trip time.

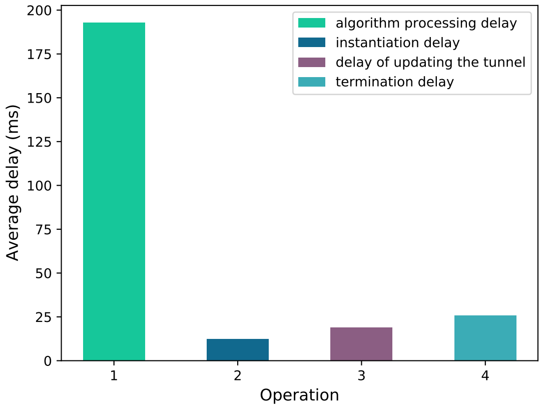

Fig. 4: Performance of i) the smart algorithm within the BSA application service, and ii) MEAO

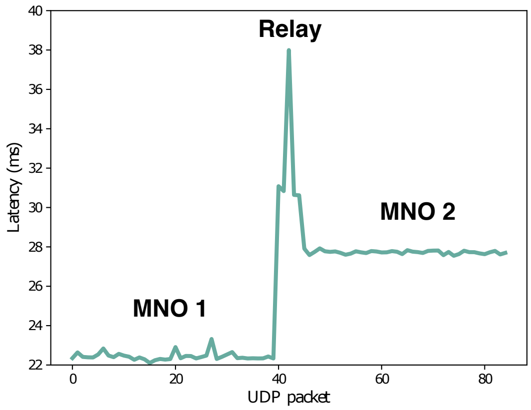

Fig. 5: Cross-border service continuity enhancements due to proactive traffic steering initiated by the transparent edge bridging procedure.

i.e., proactive service instantiation as proposed in this paper, will have an adverse impact on the service operation, as the vehicle will lose connectivity to the service right after re-attaching from the MNO 1 network to the MNO 2, thus, significantly disrupting the reliability of the EdgeApp service. On the other side, if EdgeApp relocation is triggered too early (e.g., vehicle far away from the border), the service will suffer from an increased latency due to the relaying procedure (further explained in Fig. 5). Such a result helps to understand how proactive procedures should be triggered to prevent service downtime and disruptions for the highly susceptible services such as those designed for vehicular use cases.

While the result in Fig. 4 shows the efficiency of proactive orchestration operations, Fig. 5 illustrates the latency perceived by the end user, i.e., latency measured on the client side. In Fig. 5, we present the result that shows the service continuity enhancements due to proactive steering of CAM traffic (i.e., traffic generated by the vehicle, and received by EdgeApps) from the edge system in Germany to the edge system in Austria. In this scenario, the result shows that the first 40 packets (i.e., CAM messages encoded into UDP packets) are sent directly to smart EdgeApp in Germany. The relay procedure illustrates the scenario in which vehicle connects to Austrian network while being on the German side of the border, which requires programmable data plane to relay packets from Austria to Germany. Although there is an increase in latency during the relay, as soon as EmV crosses the border, i.e., starts sending packets directly to Austrian instance, the latency decreases and becomes more stable. The difference between the average latency in Germany (22.438ms), and Austria (27.728ms), is expected as the location of the client is in Germany (lab). Thus, the service continuity enhancements achieved by proactive EdgeApp instantiation and traffic steering is evident, as vehicle does not lose the connectivity to the service, thereby being always connected to the optimal EdgeApp.

VI. CONCLUSION

Continuity of automotive services within and across mobile operator boundaries is challenging. In this paper we propose and proof enablers for orchestrated and federated smart cellular edges, that deploy automotive services on MEC platforms in operators' distributed Data Networks. By means of data analytics in smart edge applications and the cooperation with the edges' orchestration layers for proactive relocation of a connected vehicle's service and associated data session to a new MEC, we achieve a high degree of service continuity at the cost of a short period of higher communication latency during a transition phase. As proven in a prototyped system, that has been deployed in two MNOs' production network, well designed SmartApp algorithm and efficient inter-play with federated and orchestrated edges help to minimize the transient phase of higher latency, while the session continues, and possible interruptions depend solely on the resulting cellular handover performance. While this paper focuses on the gain in using smart cellular edges, continued evaluation with real vehicles using the 5G cellular network are planned in a next step.

VII. 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.

REFERENCES

[1] ETSI, "Multi-Access Edge Computing (MEC); Framework and Reference Architecture," *ETSI ISG MEC, ETSI GS MEC 003 V2.1.1*, 2019. Online [Available]: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/02.01.01_60/gs_MEC003v020101p.pdf.

[2] G. M. Yilma, U. Fattore, M. Liebsch, N. Slamnik-Kriještorac, A. Heider-Aviet, and J. M. Marquez-Barja, "5G AutoMEC - Boosting edge-to-edge service continuity for CAM in a sliced network," in *IEEE 5G for CAM - 5G Virtual Summit*, 2021. Online [Available]: https://www.researchgate.net/publication/351424586_5G_AutoMEC_-Boosting_edge-to-edge_service_continuity_for_CAM_in_a_sliced_network.

[3] 3GPP, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Study on enhancement of Ultra-Reliable Low-Latency Communication (URLLC) support in the 5G Core network (5GC) (Release 16)," *3GPP TS 23.725 V16.2.0*, 2019.

[4] 5G-Corridor Projects, *5G Trials for Cooperative, Connected and Automated Mobility along European 5G Cross-Border Corridors - Challenges and Opportunities*. Accessed on: 2022-06-13, https://5g-ppp.eu/wp-content/uploads/2020/10/5G-for-CCAM-in-Cross-Border-Corridors_5G-PPP-White-Paper-Final2.pdf.

[5] A. Kousaridas, M. Fallgren, E. Fischer, F. Moscatelli, R. Vilalta, M. Mühleisen, S. Barmpounakis, X. Vilajosana, S. Euler, B. Tossou, and J. Alonso-Zarate, "5g vehicle-to-everything services in cross-border environments: Standardization and challenges," *IEEE Communications Standards Magazine*, vol. 5, no. 1, pp. 22–30, 2021.

[6] N. Slamnik-Krijestorac, G. M. Yilma, M. Liebsch, F. Z. Yousaf, and J. Marquez-Barja, "Collaborative orchestration of multi-domain edges from a Connected, Cooperative and Automated Mobility (CCAM) perspective," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2021. doi: https://doi.org/10.1109/TMC.2021.3118058.

[7] R. Halili, F. Z. Yousaf, N. Slamnik-Kriještorac, G. M. Yilma, M. Liebsch, E. de Britto e Silva, S. A. Hadiwardoyo, R. Berkvens, and M. Weyn, "Leveraging MEC in a 5G System for Enhanced Back Situation Awareness," in *2020 IEEE 45th Conference on Local Computer Networks (LCN)*, pp. 309–320, 2020. doi: https://doi.org/10.1109/LCN48667.2020.9314838.