A Novel Approach to Distributed Model Aggregation Using Apache Kafka

Saira Bano*
University of Pisa
Pisa, Italy
saira.bano@phd.unipi.it

Emanuele Carlini
ISTI-CNR
Pisa, Italy
emanuele.carlini@isti.cnr.it

Pietro Cassarà†
ISTI-CNR
Pisa, Italy
pietro.cassara@isti.cnr.it

Massimo Coppola
ISTI-CNR
Pisa, Italy
massimo.coppola@isti.cnr.it

Patrizio Dazzi
ISTI-CNR
Pisa, Italy
patrizio.dazzi@isti.cnr.it

Alberto Gotta‡
ISTI-CNR
Pisa, Italy
alberto.gotta@isti.cnr.it

ABSTRACT

Multi-Access Edge Computing (MEC) is attracting a lot of interest because it complements cloud-based approaches. Indeed, MEC is opening up in the direction of reducing both interaction delays and data sharing, called Cyber-Physical Systems (CPSs). In the near future, edge technologies will be a fundamental tool to better support time-dependent and data-intensive applications. In this context, this work explores existing and emerging platforms for MEC and human-centric applications, and proposes a suitable architecture that can be used in the context of autonomous vehicle systems.The proposed architecture will support scalable communication among sensing devices and edge/cloud computing platforms, as well as orchestrate services for computing, storage, and learning with the use of an Information-centric paradigm such as *Apache Kafka*

CCS CONCEPTS

• **Networks** → *Layering*; • **Human-centered computing** → *Mobile devices*; • **Computer systems organization** → *Architectures*.

KEYWORDS

Cloud/Edge Continuum, Edge Intelligence

ACM Reference Format:
Saira Bano, Emanuele Carlini, Pietro Cassarà, Massimo Coppola, Patrizio Dazzi, and Alberto Gotta. 2022. A Novel Approach to Distributed Model Aggregation Using Apache Kafka. In *Proceedings of the 2nd Workshop on Flexible Resource and Application Management on the Edge (FRAME '22), July 1, 2022, Minneapolis, MN, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3526059.3533621

*Also with ISTI-CNR.
†Also with CNIT.
‡Also with CNIT.

FRAME '22, July 1, 2022, Minneapolis, MN, USA.

ACM ISBN 978-1-4503-9310-2/22/07…$15.00
https://doi.org/10.1145/3526059.3533621

1 INTRODUCTION

The pervasive analysis of people's psychological state and health is currently proving to be one of the most profitable opportunities in the technological market for wearable sensor systems and autonomous driving. As pervasiveness is becoming a requirement, the interaction of wearable devices with the surrounding environment is of utmost importance in order to characterize the influence that the environment has on the human body and on the human perception. A complex physical environment whose features are targeted by a monitoring system is a smart environment, as defined in [13]. It is evident that modeling all the interactions within a large, complex smart environment may require both a significant amount of computation and the management of a comparably complex sensing and computing platform. Cloud platforms are nowadays seen as a candidate solution to solve any kind of computing problem, due to the low time to market, and the broad range of service providers within the cloud ecosystem. However, cloud computing has demonstrated to have some significant drawbacks when it comes to smart environments: (i) the communication between end-users and the cloud materializes as a latency in the application; (ii) remote processing discloses personal information on unspecified shared spaces, owned by cloud providers; Edge computing has started affirming as an alternative solution to classical cloud computing in order to overcome these drawbacks, bringing at least part of the computation as close as possible to the end-user, thus reducing communication delays and helping preserve the privacy of data. In addition, edge computing allows for distributing the power drain proportionally to the complexity and the requirements of the monitoring system of joint CPSs. There is an obvious convergence with the concept of CPSs, that are engineered systems built from, and depend upon, the seamless integration of computation and physical component with and without human intervention. In this paper, we will focus on a specific use case [1, 3] involving a smart-vehicle environment, where the principal investigation target is not the autopilot system, but the human perception of the driving style of the autopilot.

Figure 1 shows the core CPS, which will be our reference, and is made up of data producers, i.e.: (*i*) one or more passengers' heterogeneous wearable devices, such as heart rate monitors, galvanic skin response sensors, face-tracking cameras; (*ii*) the vehicle, whose data streams include logs exposed from the car CAN bus[1] as well

[1]Details on the CAN bus can be read at: can.bosch.com

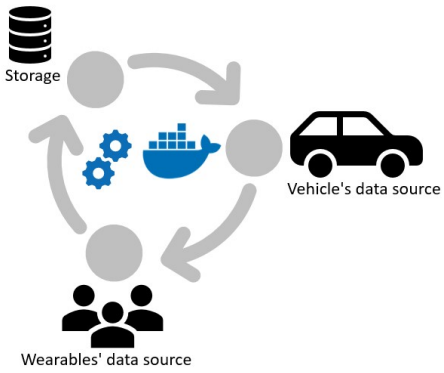

Figure 1: Use case under analysis: autonomous vehicles and human passengers' Quality of Experience (QoE).

Figure 2: Edge and Cloud interaction.

as readable input from embedded sensors like lidars, dash cameras, obstacle detection units. Yet (*iii*) a data storage, to preserve, locally, the privacy of data. Finally, (*iv*) a local computing platform (in blue) is deputed at collecting and storing data as well as to training local models of the emotional state of the passengers.

In these settings privacy control is implemented by design, as no raw personal data coming from wearable sensors or from the CAN bus can be disclosed outside the vehicle.

Starting from the use case model proposed in this paper and shown in Figure 1, the rest of this paper is organized as follows: Section 2 presents the challenges in the design of the architecture of the proposed use case, by taking into account the QoE and the privacy requirements. Section 3 focuses on the distributed computing architecture, whose our use case is made up of, that are: federated learning, distributed inter-process communication using Pub/sub mode of communication and an overview of MEC architectures based on European Telecommunications Standards Institute (ETSI) definitions. The conclusions can be read in Section 4.

2 PERSONALIZED QUALITY OF EXPERIENCE CHALLENGES FROM A PRIVACY VIEWPOINT

Data collection is the basis of the analysis processes that aim to obtain a complete and accurate picture of a phenomenon. In this work, we address the data collection of both an autonomous vehicle and its human passengers in order to improve the driver (or passenger) experience. The collected data, once analysed, can be used to obtain information to achieve the desired goal by using the collected insights. Improving the QoE for human passengers in autonomous vehicles is the very objective to be achieved in our scenario, taking into account the limitations posed by the current regulatory systems about privacy. We fully agree with [11]'s perspective that intelligent and emotion-aware systems are the next step to provide smart services in a vehicle and improve user interaction. Two conflicting factors are at play here: the need to monitor and potentially improve the QoE of people in the vehicle by using collected sensitive data such as their emotions or physiological parameters, and the need to enforce privacy when sharing data with a remote cloud-based system, as in Figure 1. This is one of the biggest challenges we face in this activity. Privacy-by-design is one of the most important approaches today, which can be considered as a philosophy of system design [6].

The emotional and physiological state of a human passenger is monitored in our environment with the help of cameras and wearables that collect a whole range of data that are merged into a model. The model of perceived QoE will strongly and directly depend on the number of observable parameters, on the possibility to collect real-time data with high spatial resolution, thus also on the quality of the installed sensors, and on the performance and efficiency of the algorithms used for data analysis. For this model to be as precise as possible, some tradeoffs need to be accepted concerning the privacy preservation (e.g. data analysis separation and anonymizing by aggregation may not always be practical).

Edge computing may ease privacy protection because data are stored closer to the user and not in a remote storage facility. Thus, it is more likely that users have more control over sensitive data; Furthermore, edge cloud is considered to be a key enabler of low-latency applications [11], as in our automotive scenario.

3 DISTRIBUTED COMPUTING INFRASTRUCTURE

This section provides a review of the existing technologies that can be integrated to define a service infrastructure meeting the criteria discussed in Section 2. In fact, the ambition of this paper is not to provide a full-fledged state-of-the-art analysis but to highlight a set of key contributions, along with the challenges they face, to give a reasonable scientific support to our perspective.

3.1 Federated Learning

Federated Learning (FL) is a promising distributed learning technique that deals with large-scale learning problems and satisfies the privacy issues presented in Section 2. This means that the learning issues depend on the collaboration between the endpoints and the edge nodes, as well as the collaboration between the edge nodes and the central cloud. Since the learning problems involve a large amount of users' sensitive data, two main problems need to be addressed, namely (i) optimizing the computational overhead between simpler devices, edge nodes, and the central cloud, and (ii) processing the data while limiting the issues and risks related to users' privacy. In the FL context, locally collected data on end-user devices are used to perform lightweight feature extraction operations. These features can be used to define local training models that can be updated at the edge nodes. Finally, the aggregated global

model is sent back to the end-user devices that execute the learning algorithms locally. Therefore, only the processed information or the models can be transmitted to both the edge nodes and the central cloud, while the data containing the users' sensitive information remains in the device. Moreover, FL can also overcome other problems typical of distributed learning [5], such as:

- the training data set at the single end-device captures limited information, so it is not necessarily representative of the whole population distribution of the end devices (as the amount and distribution over time of local training data is different for each end-device) while federated models can gather universal information and provide more accurate answers;
- the computational capacity of the federated resources is significantly greater than the resources at the single end device. For a learning system feeding on data by many end-devices, the scalability of FL increases with the number of end-devices;

In [15], the authors propose an interesting analysis of how the limited computational and communication resources on endpoints can be efficiently used together with edge resources to achieve the optimal global learning performance in a scenario like the one shown in Figure 1. The author proposes a formulation of the optimization problem and a convergence analysis to determine the frequency of global aggregation such that the available resources are used as efficiently as possible. The problem of efficiency in terms of network-resource overhead and system scalability is also addressed in [10], where the authors investigate the introduction of quantization techniques in FL to improve the efficiency of data exchange between federated nodes and an AI model aggregation service (AIaaS) on the Edge. In [18], the author propose an application of FL to the vehicular scenario called Vehicular Edge Computing (VEC), where the end devices connected with the in-proximity edge-nodes are Intelligent Connected Vehicles (ICV). The broad scenario shown in Figure 1 is common to this and other cases. The ICVs are endowed with computational and storage resources, so that a preprocessing of the raw data can be performed at the vehicle, preserving sensitive information about the passengers or other vehicles. The authors in [5, 9] also propose an infrastructure where the edge resources can be assumed located partly at the end devices and partly at the edge-node. In [18], the authors propose a selective model aggregation approach to reduce the influence of the information asymmetry, differently from the approach proposed in [15] where the local models are merged by averaging them. The information asymmetry is due to the difference among the local training data models, the availability of subset of end device to provide train data. In [18] the authors state the model selection procedure as a two-dimensional contract theory problem. In this contract theory scenario all the contract items are broadcast to vehicles periodically. The contract items are signed if they are accepted by the vehicles. Each item includes the amount of information available, the amount of computation resources and the reward. After confirming the contract items the vehicles download the global learning model from the central server and performs local model training by using local information and computation resources, according to the accepted contract items.

3.2 MEC Architectures Based on ETSI Definitions

The effective and efficient exploitation of resources at the edge is a key enabler to satisfy QoE requirements stated in Section 2, relying on resources located nearby the end-user.

MEC technology is the emerging solution to address the challenges stated in Section 2 for the sensing systems of the vehicular scenario, such as that taken into account in this work. MEC allows to design infrastructures where the data acquisition-preprocessing of data and training of local models can be performed at the end device and at the edge, respectively. Then, the cloud can perform the optimal aggregation of these models. Performing at the edge some of the computing operations, MEC allows to reduce significantly both the computational load for the core network and the latency of the services for the end user, optimizing so the user experience. The authors in [8, 9] provide an overview about existing MEC-based infrastructure, analysing communication and computational issues. The core of MEC infrastructure is based on a virtualized platform that leverages recent advancements in Information-Centric Networks (ICN), and Software-Defined Networks (SDNs). ICN provides an end-to-end service recognition paradigm for MEC, shifting from a end device-centric to an information-centric one for implementing context-aware computing. SDN allows MEC infrastructure administrators to manage services via function abstraction, achieving scalable and dynamic computing.

In [12, 19], the authors propose a resource allocation method for mobile multi-end user scenario, where computation tasks can be split simultaneously in local computing and offloading. The authors provide an interesting analysis about the performance obtained by the proposed method where parameters such as latency, computation time and limited hardware resources are not negligible parameters. In [7, 14, 17], the authors propose examples of MEC-based infrastructures for the vehicular scenario. Different solutions are explored for the scheduling of the service operation started by the vehicles, using both approaches offloading, or by the cooperation of vehicles. The infrastructure can rely on edge servers on vehicles and at Road Side Units (RSU). In the first case, the proposed infrastructure allocates the resources to address the operations locally at the edge-nodes in case of light computing operations, in case of complex operations these are offloaded to the central cloud or among a cluster of vehicles.

3.3 Distributed inter-process communication

LinkedIn has developed a novel messaging system for log processing called Kafka[2], which is licenced as open source. While traditional messaging systems such as IBM Websphere MQ[3] offered highly reliable message delivery (as each message had to be confirmed), such a strict reliability constraint is often overkill. The cost of data loss is considered negligible with Kafka, as Kafka has fault tolerance and resilience capabilities by exploiting multiple instances of the broker.

In Apache Kafka, data is stored in the form of *topics*. A producer can publish messages on a topic. The published messages are stored on a server called *broker*. While consumer subscribed to these topics

[2]Kafka: https://kafka.apache.org/
[3]IBM Websphere MQ: https://www.ibm.com/support/knowledgecenter/

can consume the messages from these *topics* over the *broker*. Most of the messaging systems support a *push* policy for messages that the broker has to deliver messages from producers to consumers. Kafka, instead, applies a *pull* policy in message forwarding: each consumer can try to retrieve data at the maximum rate that it can handle, but is never flooded by messages, thus reducing the overhead over the *broker*. Another design choice that makes Kafka very scalable and suitable for distributed system is that it can exploit on multiple broker instances. To simplify the coordination of the instances, Kafka employs Zookeeper in order to (*i*) detect the addition and the removal of both brokers and consumers, (*ii*) maintain the relationship among brokers and consumers, (*iii*) trigger a rebalancing of the workloads when either brokers or consumers are added or removed. Several data collection processing solution on the edge employ Kafka. In [16], a Wireless Sensor Network (WSN) for monitoring CO_2 makes use of Kafka and the Impala database let a huge amount of data flow from the sensor nodes to the database. The authors in [4] have developed an efficient and low-latency distributed messaging system for Connected Vehicle (CV) applications that provides a data-oriented view of the entire CV ecosystem.

In [2], the authors propose a new two-tier federated learning communication architecture, namely *Kafkafed*, for vehicular applications. They use the Apache Kafka publish/subscribe platform for aggregating models. This architecture has two advantages: First, it decouples the data producers (client) and the data consumers (server) and provides scalability. This decoupling also provides an additional layer of privacy, as the server knows nothing about the clients. In addition, the clients, or in our case the vehicles, do not need to be connected to the server, as is the case with classic client-server communication, where the client remains constantly connected to the server to exchange information, which is not possible with mobile nodes, as they may be out of range of the server while driving. The second major advantage of *Kafkafed* is the reduction in communication cost in the FL scenario, as the server sends the aggregated model to some brokers (at the edge) that serve a large number of vehicles near the respective edge, so that the vehicles do not need to connect to the cloud, as the connection between the cloud and the edge nodes causes a large latency. For model aggregation, we will rely on the *Kafkafed* technique with a large number of broker instances to increase scalability in automotive scenarios.

4 CONCLUSIONS

Existing solutions and approaches in the literature typical for VEC to collect physiological data to provide emotion-awareness of the drivers in autonomous vehicles have been investigated in order to evaluate them with respect to the specific context that is being analysed. MEC can offer tremendous benefits in such a context due to its capabilities, starting from the possibility to increase the privacy of human passengers and to provide tight and real-time computing power to achieve the desired QoE. The FL approach is used to aggregate the models trained on different nodes using a pub/sub mode communication architecture. The authors are currently working on the design and development of such an edge-based software platform to support the use case and believe that it will open up innovative scenarios in the automotive sector.

ACKNOWLEDGMENT

This work was partially supported by EU-H2020 research and innovation program TEACHING, "A computing toolkit for building efficient autonomous applications leveraging humanistic intelligence" under grant agreement no. 871385.

REFERENCES

[1] Davide Bacciu, Siranush Akarmazyan, Eric Armengaud, Manlio Bacco, George Bravos, Calogero Calandra, Emanuele Carlini, Antonio Carta, Pietro Cassarà, Massimo Coppola, et al. 2021. TEACHING-Trustworthy autonomous cyber-physical applications through human-centred intelligence. In *2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS)*. IEEE, IEEE, NY, 1–6.

[2] Saira Bano, Nicola Tonellotto, Pietro Cassarà, and Alberto Gotta. 2022. KafkaFed: Two-Tier Federated Learning Communication Architecture for Internet of Vehicles. In *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. IEEE, NY, 515–520. https://doi.org/10.1109/PerComWorkshops53856.2022.9767510

[3] Valerio De Caro, Saira Bano, Achilles Machumilane, Alberto Gotta, Pietro Cassarà, Antonio Carta, Rudy Semola, Christos Sardianos, Christos Chronis, Iraklis Varlamis, et al. 2022. AI-as-a-Service Toolkit for Human-Centered Intelligence in Autonomous Driving. In *IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, IEEE, NY, 1–6.

[4] Yuheng Du, Mashrur Chowdhury, Mizanur Rahman, Kakan Dey, Amy Apon, Andre Luckow, and Linh Bao Ngo. 2017. A distributed message delivery infrastructure for connected vehicle technology applications. *IEEE Transactions on Intelligent Transportation Systems* 19, 3 (2017), 787–801.

[5] R. Fantacci and B. Picano. 2020. Federated learning framework for mobile edge computing networks. *IET Transactions on Intelligence Technology* 5, 1 (January 2020), 15–21.

[6] Jaap-Henk Hoepman. 2014. Privacy design strategies. In *IFIP International Information Security Conference*. Springer, Springer, Switzerland, 446–459.

[7] G. Luo, Q. Yuan, H. Zhou, N. Cheng, Z. Liu, F. Yang, and X. S. Shen. 2018. Cooperative vehicular content distribution in edge computing assisted 5G-VANET. *China Communication* 15, 7 (April 2018), 1–17.

[8] P. Mach and Z. Becvar. 2017. Mobile edge computing: A survey on architecture and computation offloading. *IEEE Communication Surveys Tutorials* 19, 3 (March 2017), 1628–1656.

[9] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief. 2017. A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys Tutorials* 19, 4 (November 2017), 2322–2358.

[10] Nicola Tonellotto, Alberto Gotta, Franco Maria Nardini, Daniele Gadler, and Fabrizio Silvestri. 2021. Neural network quantization in federated learning at the edge. *Information Sciences* 575 (2021), 417–436. https://doi.org/10.1016/j.ins.2021.06.039

[11] Hans-Jörg Vögel, Christian Süß, Thomas Hubregtsen, Elisabeth André, Björn Schuller, Jérôme Härri, Jörg Conradt, Asaf Adi, Alexander Zadorojniy, Jacques Terken, et al. 2018. Emotion-awareness for intelligent vehicle assistants: a research agenda. In *2018 IEEE/ACM 1st International Workshop on Software Engineering for AI in Autonomous Systems (SEFAIAS)*. IEEE, NY, 11–15.

[12] C. Wang, C. Liang, F. R. Yu, Q. Chen, and L. Tang. 2017. Computation offloading and resource allocation in mobile edge computing. *IEEE Trans. on Wireless Communications* 16, 8 (August 2017), 4924–4938.

[13] Fei-Yue Wang, Daniel Zeng, and Liuqing Yang. 2006. Smart cars on smart roads: an IEEE intelligent transportation systems society update. *IEEE Pervasive Computing* 5, 4 (2006), 68–69.

[14] J. Wang, D. Feng, S. Zhang, J. Tang, and T. Q. Quek. 2019. Computation offloading for mobile Edge computing enabled vehicular networks. *IEEE Access* 7 (May 2019), 62624–62632.

[15] S Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan. 2019. Federated learning framework for mobile edge computing networks. *IEEE Journal of Selected Areas in Comsmunication* 37, 6 (June 2019), 1205–1221.

[16] R. Wiska, N. Habibie, A. Wibisono, W. S. Nugroho, and P. Mursanto. 2016. Big sensor-generated data streaming using Kafka and Impala for data storage in Wireless Sensor Network for CO2 monitoring. In *2016 International Workshop on Big Data and Information Security (IWBIS)*. IEEE, NY, 97–102.

[17] RENCHAO Xie, Qinqin Tang, Qinqin Wang, XU Liu, F. RICHARD Yu, and Tao Hu. 2019. Collaborative Vehicular Edge Computing Networks: Architecture Design and Research Challenges. *IEEE Access* 7 (December 2019), 178942–178952.

[18] D. Ye, R. Yu, and Z. PAN, M.and HAN. 2020. Federated Learning in Vehicular Edge Computing: A Selective Model Aggregation Approach. *IEEE Access* 8 (February 2020), 23920–23935.

[19] C. You, K. Huang, H. Chae, and B.-H. Kim. 2018. Energy-efficient resource allocation for mobile-edge computation offloading. *IEEE Trans. on Wireless Communications* 66, 4 (April 2018), 1594–1608.