V-Edge: Virtual Edge Computing as an Enabler for Novel Microservices and Cooperative Computing

V-Edge: Virtual Edge Computing as an Enabler for Novel Microservices and Cooperative Computing

Falko Dressler, Carla Fabiana Chiasserini, Frank H.P. Fitzek, Holger Karl, Renato Lo Cigno, Antonio Capone, Claudio Casetti, Francesco Malandrino, Vincenzo Mancuso, Florian Klingler, Gianluca Rizzo

***Abstract*—As we move from 5G to 6G, edge computing is one of the concepts that needs revisiting. Its core idea is still intriguing: Instead of sending all data and tasks from an end user's device to the cloud, possibly covering thousands of kilometers and introducing delays lower-bounded by propagation speed, edge servers deployed in close proximity to the user, e.g., at some base station, serve as proxy for the cloud. This is particularly interesting for upcoming machine learning (ML)-based intelligent services, which require substantial computational and networking performance for continuous model training. Yet this promising idea is hampered by the limited number of such edge servers. In this position paper, we discuss a way forward, namely the virtual edge computing (V-Edge) concept. V-Edge helps bridging the gap between cloud, edge, and fog by virtualizing all available resources including the end users' devices and making these resources widely available. Thus, V-Edge acts as an enabler for novel microservices as well as cooperative computing solutions in next-generation networks. We introduce the general V-Edge architecture and we characterize some of the key research challenges to overcome in order to enable wide-spread and intelligent edge services.**

Figure 1. The V-Edge concept: Abstraction of the physical resources of the cloud back end, the 5G/6G core and RAN, as well as users and machines. All components provide (and may use) resources for edge computing and they may participate in a local, distributed orchestration on short time scales, while global optimization and other non-real-time operations can be performed in the cloud back end.

I. Introduction

Next-generation mobile networks are envisioned to provide the computational, memory, and storage resources needed to run services required by diverse third parties (referred to as vertical industries or verticals). Each service is associated with specific requirements, quantified as key performance indicators (KPIs). To this end, networks will require a high degree of flexibility and fully automated operations, with a drastically reduced service deployment time. Essential components to achieve these goals are softwarization of both networking and services using network function virtualization (NFV) [1], [2] and the ability to store and process data close to the end user leveraging so-called edge computing. Edge computing goes beyond classic multi-access edge computing (MEC) as standardized for 5G networks [3], [4]; rather, the network edge has become the convergence point of data processing, caching, and communication [5], which makes service provisioning at the edge one of the key challenges in future networks. This holds in particular for upcoming intelligent machine learning (ML)-based services that require substantial computational resources for model training and also need to continuously exchange (parts of) the trained models.

Network virtualization, significantly supported by the current 5G/6G standardization and research beyond it, pushes NFV to merge with the concept of microservices to improve practicality, universality, and automation. Service ubiquity and resilience are emerging as the ultimate goals – following recent work in the context of Tactile Internet [6], [7]. The degree of resilience describes how well a system can deal with changing environments including unforeseen events. To achieve these goals, networks are progressively integrating *machine learning* [8] in two main ways. First, an increasing number of user applications include ML models for a smarter application behavior, higher ability to adapt to users' preferences, and more effective interaction between users and machines. Second,

This work has been supported in part by the German Research Foundation (DFG) under grant no. DR 639/25-1.

F. Dressler is with the School of Electrical Engineering and Computer Science, TU Berlin, Berlin, Germany, e-mail: dressler@tkn.tu-berlin.de.

C. F. Chiasserini is with the Dept. of Electronics and Telecommunications, Politecnico di Torino, Torino, Italy, e-mail: carla.chiasserini@polito.it.

F. H.P. Fitzek is with the School of Engineering Sciences as well as the excellence cluster CeTI, TU Dresden, Dresden, Germany, e-mail: frank.fitzek@tu-dresden.de.

H. Karl is with the Hasso-Plattner-Institute, Potsdam, Germany, e-mail: holger.karl@hpi.de.

R. Lo Cigno is with the Dept. of Information Engineering, University of Brescia, Brescia, Italy, e-mail: renato.locigno@unibs.it.

A. Capone is with the Dept. of Electronics, Information and Bioengineering, Politecnico di Milano, Milano, Italy, e-mail: antonio.capone@polimi.it.

C. Casetti is with the Dept. of Control and Computer Engineering, Politecnico di Torino, Torino, Italy, e-mail: claudio.casetti@polito.it.

F. Malandrino is with the Institute of Electronics, Computer and Communication Engineering of the National Research Council of Italy, Torino, Italy, e-mail: francesco.malandrino@ieiit.cnr.it.

V. Mancuso is with IMDEA Networks, Madrid, Spain, e-mail: vincenzo.mancuso@imdea.org.

F. Klingler is with the Dept. of Computer Science, Paderborn University, Paderborn, Germany, e-mail: florian.klingler@uni-paderborn.de.

G. Rizzo is with the Research Institute of Information Systems, HES-SO, Valais-Wallis, Switzerland, and with University of Foggia, Italy, e-mail: gianluca.rizzo@hevs.ch.

Table I
TYPICAL NODES PARTICIPATING IN V-EDGE CLUSTERS

Stationary / infrastructure-based systems				
Type	CPU	Storage	Network	Time available
ISP-operated MEC server	high-performance multi-core	$1 - 100\,\mathrm{TByte}$	$0.1 - 10\,\mathrm{Gbit/s}$	years
privately operated MEC server	multi-core	$1 - 10\,\mathrm{TByte}$	$0.1 - 1\,\mathrm{Gbit/s}$	weeks
Wi-Fi APs	single core	$0.1 - 100\,\mathrm{GByte}$	$0.1 - 1\,\mathrm{Gbit/s}$	months
Mobile / opportunistic systems				
Type	CPU	Storage	Network	Time available
Moving car	low-end multi-core	$0.1 - 1\,\mathrm{TByte}$	$0.1 - 1\,\mathrm{Gbit/s}$	$1 - 5\,\mathrm{min}$
Parked car	low-end multi-core	$0.1 - 1\,\mathrm{TByte}$	$0.1 - 1\,\mathrm{Gbit/s}$	$0.5 - 24\,\mathrm{h}$
Fully autonomous shuttle	high-performance multi-core	$1 - 10\,\mathrm{TByte}$	$0.1 - 1\,\mathrm{Gbit/s}$	$1 - 30\,\mathrm{min}$
User with smartphone	single to multi-core	$1 - 100\,\mathrm{GByte}$	$0.1 - 1\,\mathrm{Gbit/s}$	$5 - 15\,\mathrm{min}$

ML-based approaches have become common in research on automatic network management, resource orchestration, as well as in predicting a wide range of parameters (e.g., wireless channel properties, users' behavior, service demand). This allows to dynamically and proactively tune the operational parameters (and even switching between complete protocol stacks) if needed.

Given these emerging trends, *the network edge is turning into an enabler* between the cloud and a fully-distributed machine-to-machine (M2M) network, hosting virtualized network functions and user applications, to meet both service providers' and users' needs. Eventually, all current MEC applications are in scope but so are novel classes that are only enabled by new degrees of virtualization and inherent ML support.

The aim of this position paper is to analyze the research issues that arise when virtualization is applied to network services and user applications at the edge in a comprehensive manner. With the limited availability of 5G edge servers, we believe that the only way forward is to virtualize all edge resources. In particular, we introduce the *virtual edge computing (V-Edge)* concept. It takes advantage of the flexibility offered by network softwarization and NFV to integrate, opportunistically and dynamically, the highly heterogeneous set of resources available locally at the edge (e.g., computing, storage, and communication resources), while guaranteeing seamless and QoS-aware service provisioning to users in a variety of verticals.

A schematic representation of the V-Edge concept is depicted in Figure 1. Compared to 5G and traditional edge computing, the system comprises resources with varying availability/fluctuating availability: CPUs, connectivity, and storage capacity come and go as users do, carrying the corresponding devices. Thus, we have to move from allocating *static* resources to dynamic users and applications to allocating resources that are dynamic as well. An example is the integration of cars not only as service users but also as service providers, explored e.g. in [9], [10]. V-Edge goes well beyond initial activities towards distributed computing and data storage, realizing a full and harmonic integration between infrastructure-based communication networks and mobile edge systems at the resource level, as well as between user applications and network functions at the service layer.

In V-Edge, part of the orchestration of resources and tasks needs to be done at the edge on rather short time scales to cope with resource volatility and dynamics. The back-end cloud, instead, can be used for global optimization on longer time scales. Following current ML approaches to 5G and edge computing [8], V-Edge will also be inherently learning-based, supporting both user applications and network functions. Distributed learning concepts such as federated learning [11] serve as a blueprint but need to be integrated with reactive approaches such as reinforcement learning to deal with the dynamics of the overall system. Federated learning also helps to realize privacy-preserving, distributed approaches and to effectively transfer trained models where and when needed.

The contributions of our position paper can be summarized as follows:

- we characterize the move from classic MEC to harmonized virtual edge computing for improved scalability, resilience, and flexibility;
- we introduce the conceptual architecture of V-Edge making consequent use of virtualization to deal with the high degree of dynamics in the network; and
- we summarize and discuss relevant research questions to be solved to make V-Edge reality.

II. THE V-EDGE ECOSYSTEM

Before outlining our conceptual virtual edge computing architecture, we introduce the underlying basic components of the V-Edge eco-system, including the major services it can support.

Users: As in conventional systems, users still contribute to the traffic demand while using edge-based applications. In V-Edge, users may have a dual representation in the system as edge users but also as resource providers. A big challenge, as for other resource-sharing systems, is incentivation, as users may not be willing to share their resources (at least energy, possibly costly communication) with others. Incentivation concepts need to be explored based on prosumer models in other fields (users producing and consuming at the same time).

Resources: Required resources to satisfy the user demand, network-wise and application-wise, are now provided by an increasing variety of devices ranging from the cloud over internet service provider (ISP) to community-operated edge servers, and even to small internet of things (IoT) systems. V-Edge goes well beyond classic MEC, by dropping the differentiation between cloud and edge and fog, and opportunistically recruiting local, already existing – yet possibly unused – resources. In V-Edge, even small "fog" devices are conceptually turned into "edge servers" to provide functions to third parties. This way, edge computing coordinated by an edge server and, in most cases, by the ISP and/or the cloud, merges conceptually with fog computing, which is by nature fully distributed and yields a heterogeneous, yet fully integrated, virtualized ecosystem. A list of typical V-Edge nodes with their computational, storage, and networking resources is provided in Table I, also indicating the average time such nodes will be available in a given location. We are, obviously, talking about very dynamic scenarios. Classic MEC assumes dynamics in terms of users and their tasks coming and going. Now, also the available edge computing resources come and go, which can be seen as constituting a *virtualized edge server* with time-varying resource availability. A fundamental part is therefore user management and resource discovery. Solutions similar to, for example, in vehicular micro-clouds [9], may be considered.

Services and Functions: Network services, and often user applications, need to be deployed within the V-Edge. The classes of user applications that can benefit most from a virtual edge implementation include:

- services with tight latency constraints or whose support with dedicated static infrastructure would have entailed too high a CAPEX, e.g., cooperative (automated) driving and UAV control, in need of local edge support even out of cities;
- services that may exhibit bursts of demand of computing tasks, e.g., due to "flash crowds";
- IoT applications like monitoring tasks, where local data have to be pre-processed for immediate use, or transferring large amounts of data to the cloud that would require too much data rate;
- augmented reality (AR), and in general extended reality (XR), applications, as well as any six degrees of freedom (6DoF) immersive technology that requires both low latency and large data rate;
- ML applications making use of ML as a service [12], which has emerged as a new paradigm whereby trained or pre-trained models are provided for making decisions in different contexts.

Orchestrator: To complete the above functions, resource and service orchestration is needed. Resource orchestration can be both reactive (which may sometimes be too late) or proactive, so that resources, and the functions mapped thereon, can naturally follow demand in space and time. We remark that the orchestration itself becomes one of the tasks (likely ML-supported) to be distributed and executed within the virtual edge, in this respect similar to user applications.

The orchestrator (Figure 2) has to observe and monitor nodes and their computing and communication resources and schedule them for individual functions or entire microservices. Machine learning will help to make such decisions with little and often impaired information about the available edge components. From an architectural perspective, the orchestrator can be centralized at a (physical) edge server (or even in the cloud, with the risk of additional problems due to the communication delay), or decentralized through nodes participating in the V-Edge. In realistic deployments, an at least partially distributed solution may be preferred for better resilience (no single point of failure) and responsiveness of the overall system (optimized solutions on a global scale but updated locally).

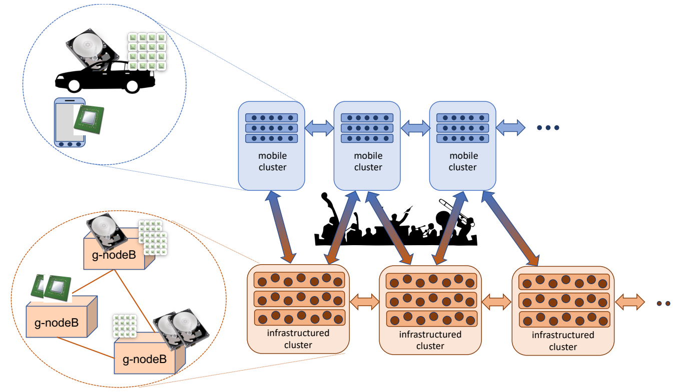

Figure 2. V-Edge architecture: Logical resources from mobile users and infrastructure-based systems (e.g., edge servers co-located with a gNB) are aggregated into clusters. Multiple clusters are appropriately coordinated and microservices can migrate from one cluster to another to optimize the service location. Resource management is done by an orchestrator, which may interwork with others, controlling neighboring clusters, to migrate services.

Architecture: The architecture of a V-Edge system enables the interaction between the components described above (Figures 1 and 2). A key feature of the V-Edge architecture is that users are grouped together so as to virtually cluster their resources and to provide these resources qualitatively equivalent to the ones provided by the cloud or edge infrastructure. This cluster-based organization facilitates and optimizes resource management while providing resilience and flexibility, like done, e.g., in the context of vehicular micro-clouds [9]. Services and network functions can be instantiated in one cluster and then migrated to another one dynamically, under the coordination of the orchestrator, and driven by the learning process that underpins its operations. Figure 2 zooms into the architecture outlining interconnected mobile and infrastructure clusters that are orchestrated together. The distributed nature of all resources additionally requires novel concepts and interfaces for distributed orchestration and for the cooperation between orchestrators and even between multiple such clusters, edge components, and the back-end cloud servers.

III. Key Technologies and Research Challenges

Existing work on edge computing has predominantly focused on resource allocation on edge servers that may experience dynamic load but whose deployment is static or only changes on long time scales. V-Edge goes well beyond this limitation by allowing also resources to be mobile, thus, computational, storage, and communication resources may come and go at any time. In this section, we identify most relevant key technologies and the related research challenges that will make V-Edge a reality.

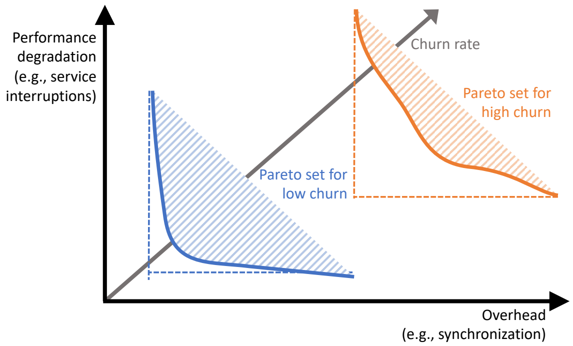

Figure 3. Services KPIs in the context of the V-Edge concept.

A. Performance Aspects

Similar to non-virtual edge clouds, a V-Edge system needs to optimize classic KPIs such as throughput, latency, service rejection rate, utilization, combined with high dependability and easy management, as well as to maximize the number of satisfied users and expected revenue compared to capital or operational expenditure (CAPEX, OPEX). There exists, however, a differentiating factor between V-Edge and non-virtual edge computing: the node *churn rate*, i.e., the rate at which nodes join and leave the V-Edge system and the evolution of the network topology in space and time due to device mobility. Mobility needs to be tracked to estimate the time to connection loss. ML algorithms to can help to detect patterns of user behavior, to track gatherings, or other mobility patterns. The churn rate is not a performance metric but rather a system characteristic with a twofold impact. On one hand, it may degrade the V-Edge KPIs, which could be characterized as the *price of virtualization*; on the other hand, the use of mobile devices to the V-Edge allows significant CAPEX and OPEX savings and improving overall system resilience.

While this is a fair perspective from an end user's or investor's perspective, it can fall short when comparing different V-Edge realizations against each other. First, more fine-grained metrics would be needed in this case to characterize the performance of services as well as management and orchestration systems (e.g., packet latency vs. service initiation time, or traffic throughput vs. number of service deployments per second). Second, suitable metrics should be selected to highlight the existing trade-offs in performance. A typical example is the overhead introduced by state synchronization to ward off service interruptions, compared against the degradation in the users' quality of experience caused by those same service interruptions.

Consequently, V-Edge will need to find compromises between multiple metrics that cannot be traded off against each other. In such a case, configurations that are worse in all metrics than some other known configuration are clearly not interesting; we need to find the set of configurations that are better than all other configurations in at least one metric (often called the "Pareto front" as the set of all Pareto-optimal solutions). While this is a well-known problem from optimization theory, it is aggravated in V-Edge as these Pareto fronts likely will depend on extrinsic, non-controllable parameters. As an example, Figure 3 illustrates two Pareto fronts for the two metrics "Overhead" and "Performance degradation" (thick blue or orange lines), depending on "Churn rate" as an extrinsic parameter. These complex dependencies and entanglement between diverse aspects of the V-Edge are not well captured by existing performance metrics; they rather call for a novel approach to choosing the most suitable system configuration.

B. Orchestration of Microservices

Network softwarization is taking over the data, control, and management planes, as well as different protocol layers. Examples of data plane virtualization include virtual routers and user applications, while a relevant control plane example is the new "Open and Smart Radio Access Network" (O-RAN) architecture developed by the O-RAN Alliance to transform the radio access networks industry towards open, intelligent, virtualized, and fully interoperable RANs.[1] As mentioned in Section II, virtual network functions (VNFs) stemming from such softwarization can be seen as (components of) microservices, which need to be properly and jointly orchestrated, whenever they compete for the same physical resources. Further, depending on their type and logic, microservices can be executed in different execution environments with varying trade-offs in terms of capabilities and performance.

Thus, an orchestrator for a V-Edge system needs to provide the same functionality as any of the orchestrators proposed for an ordinary edge infrastructure. Namely, it has to map VNFs used to compose microservices to the available resources, taking into account not only their requirements but also the computing and communication capabilities of the device on which they are mapped and the performance impact of the services that leverage such microservice instances. This is, however, not the only issue a V-Edge orchestrator faces. Indeed, it has to cope with the network and node churn: quickly changing network conditions and node availability. A V-Edge orchestrator has to be aware of this churn as well as of the services' ability to deal or not to deal with it (e.g., stateless vs. stateful services) and their temporal and spatial availability requirements – aspects that are exacerbated in V-Edge with respect to conventional scenarios. This fact invalidates any conventional, long-term approach and demands a more agile, adaptive solution.

We address this challenge by leveraging machine learning techniques, conceiving a multi-faceted framework that can effectively deal with the multitude of necessary observations and actions. Specifically, the proposed V-Edge orchestration framework includes:

1) a network model, partially based on explicit information (e.g., battery or computing capacity of a device) and partially learned information (e.g., movement patterns and sojourn time), to account for individual devices' capabilities and behavior;
2) a service model, partially provided by the VNF graph composing the service and the VNFs' specifications, partially learned (e.g., how disruption-tolerant is a service, how does a disruption affect the users' quality of

[1] O-RAN Alliance: https://www.o-ran.org

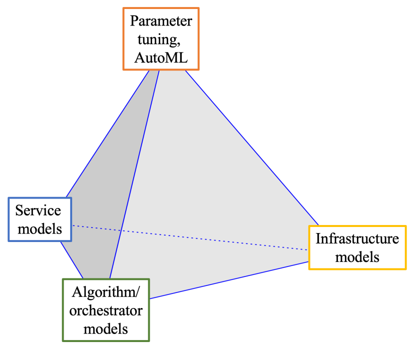

Figure 4. The V-Edge tetrahedral orchestrator (VETO).

experience); we underline that some information that could be provided by the service developer might actually need to be learned in practice and that a continuous update of the service understanding is necessary;
3) the orchestrator as such, i.e., learning scaling, placement, routing, migration, and other actions based on the network and service models;
4) an Auto-ML component, which is necessary to train hyperparameters, since the above three models need to be continuously trained in the field and since properly parameterizing training is hard.

The four components of the framework are connected in a tetrahedron as depicted in Figure 4, as they all depend on each other; we dub such a framework V-Edge tetrahedral orchestrator (VETO). VETO provides a functional separation of a learning-based orchestrator. Some important challenges, however, remain to be addressed for a detailed framework design. In particular, it is critical to: (i) learn correlation between network and service models, e.g., between user spatial distribution and service demand dynamics, (ii) identify the hyperparameters to be learned by the Auto-ML component, (iii) define the time scale over which the different components should operate, (iv) understand with which granularity instances of VETO should be deployed to deal with different geographical areas to make the system scalable.

C. Cooperative Computing

Cloud computing introduced dynamic resource allocation and flexible costs in computing, enabling many novel services. Edge computing places services in close proximity of the user enabling new kind of services, focusing on low latency, e.g., services used in Tactile Internet [6], [7]. Cooperative computing, central to the V-Edge concept, adds resilience and flexibility by distributing computing tasks dynamically based on their requirements to achieve the needed level of responsiveness (e.g., ensured termination within time bounds).

Latency is caused by propagation and computing delay, both of which need to be taken into account to find placement and distribution of a function to return its result in time at minimum cost [13]. Figure 5 illustrates the problem in the application scenario of connected vehicles: V-Edge clusters running computing functions on the vehicles themselves can be hierarchically extended using 5G/6G infrastructure, e.g., following the vehicular micro cloud architecture [9]. In this

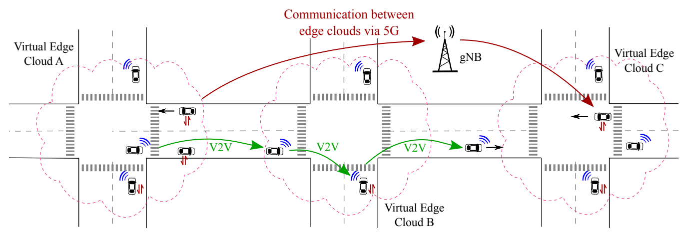

Figure 5. Placement of mobile edge computing for connected vehicles.

example, vehicle to vehicle (V2V) communication is used between the nodes within a V-Edge and between adjacent V-Edges, while distant V-Edges communicate via the 5G network. The V-Edge architecture provides vertically and horizontally distributed placement of computing functions, assuming distribution and coordination (Figure 2).

Recent advances in the field of coding, in particular network coding and coded caching, provide efficiency, resilience, and low latency for information distribution and storage. *Coded computing* may provide solutions for cooperative computing [14] but requires novel results and innovations that progressively solve the following challenges to be enabled in V-Edge:

- *Challenge 1.* Interconnect Edge nodes in V-Edge clusters and distribute tasks among themselves; the back-end data centers are used as fallback solution as described already for vehicular micro-clouds [9]. The critical issue here is the distribution and synchronization of all relevant data to perform computing and meta data describing tasks. Hierarchical organization simplifies a solution but does not necessarily yield the optimum.
- *Challenge 2.* A better solution can be based on coding. Similar to network coding for storage and communication, also computing tasks can be coded to avoid outages if physical nodes leave the virtual edge [14]. Coded computing is normally used between neighboring nodes such as mobile robots or cars, but it can be extended to cooperation among multiple edge clouds, adding resilience and performance.
- *Challenge 3.* Integrate such coding-based distributed computing with new paradigms for resource management. Current management solutions focus separately on communication and computational resources, with some initial, simplistic attempts to joint allocation, normally tackled as a static optimization problem. However, the problem is highly non-linear and time-correlated (future allocations depend on past ones in non-trivial ways), thus management based on advanced ML techniques is compelling, yet far from trivial.

Thus, for cooperative computing, the open research questions can be summarized as follows:

- Characterization of the computing requirements beforehand to select the appropriate subset of computing resources that can perform the task with the required dependability;
- identification of the subset of data to be distributed, in particular what data goes to which computing node;

- dynamic and distributed allocation of communication and computing resources in an environment subject to continuous change and sudden failures of both computing nodes and communication links;
- data fusion and compressed sensing in distributed environments, where computational decisions influence further computing tasks and future decision (think, as an example, of distributed inference on roads, including vulnerable users as pedestrians and bikers);
- sparsification of models to reduce the overhead resulting from model distribution in distributed learning.

Such challenges and research questions naturally rely on distributed learning concepts. Federated learning in particular will play a dominant role because it realizes distributed training and then merging the models generated in a privacy-preserving manner [11], [15].

IV. Discussion and Conclusion

Revisiting the motivation for our virtual edge computing (V-Edge) approach, in the following, we discuss the benefits of this novel architecture and what cannot be done with traditional cloud, MEC, and fog computing. From a paradigmatic point of view, there are many reasons to make use of V-Edge even though some fundamental problems need to be addressed before implementation. Making the V-Edge concept reality, however, requires a tremendous effort for algorithms, for a conceptual view on scalability and resilience, as well as for technologically discovering nodes quickly and fast communication between V-Edge nodes.

From a *policy perspective*, the V-Edge concept addresses many problems that have hampered (mobile) global communications in the past decades. As discussed above, V-Edge requires open solutions at different architectural levels. Openness in telecommunications and computing has proven to be one of the key enablers for innovation and economic growth. Thus, the V-Edge vision naturally becomes the melting pot for novel services, solutions, start-ups, and technological evolution. This consideration alone should be enough for all actors, and standardization bodies in particular, to embrace V-Edge and mold future business based on this equitable architecture.

Scalability is going hand in hand with efficient resource discovery and utilization to make advanced services more accessible and, thus, affordable by a wider sector of the society. If, on one hand, a distributed architecture is the only solution to scalability, on the other hand, it is well known that uncoordinated distribution uses resources inefficiently, from storage and computing power down to communication and energy. V-Edge proposes an advanced, ML-oriented orchestration that enables the efficient and dynamic use of resources, especially leveraging those that go unused for a large part of the time. An example for all: processing power on autonomous vehicles when they are parked. The safety requirements of Society of Automotive Engineers (SAE) Level-5 autonomous driving require a processing power (CPU and GPU) that is comparable to several nodes of high-performance computing systems, and this extreme capacity is right there, at the edge, but with traditional architectures it is impossible to tap it.

Resilience is one of the most pressing reasons to foster V-Edge. 5G/6G architectures, together with computation (think about GPUs) and local access (WiFi 6 and the upcoming WiFi 7), have shown that only extreme distribution and densification of resources can meet the increasing requirements on communications and services. V-Edge is bringing this evidence from subliminal awareness to architectural design, highlighting and formalizing the interdependence between communications, computing, management (resource allocation and scheduling), and service KPIs. So doing, V-Edge clarifies the technical challenges that need to be addressed for success, first of all in the realm of ML, acknowledging that traditional models cannot be applied to a system whose evolution is not predictable *a priori*. Resilience, to some extent, requires some level of adaptability and flexibility that empowers the autonomous evolution of functions, services, and management models through autonomous learning and self-development.

Finally, *security, privacy, and trust* need to be considered, which goes well beyond the scope of this paper but aligns well with resilience. Lessons learned in computer science indicates that distributed systems are in general safer, more secure, and most of all naturally support the implementation of “privacy by design” principles. V-Edge clearly matches this indication, with its extreme distribution and the orchestration of resources coming from different actors and entities. We are, however, also well aware that *practical* systems often fail to meet theoretical results, in particular concerning security where the complexity of distributed systems may lead to design failures, with severe consequences. Trust will also help incentivizing users to participate. This is a further topic for research and design towards the V-Edge realization.

References

[1] F.-Y. Wang, L. Yang, X. Cheng, S. Han, and J. Yang, “Network softwarization and parallel networks: beyond software-defined networks,” *IEEE Network*, vol. 30, no. 4, pp. 60–65, Jul. 2016.

[2] I. Alam, K. Sharif, F. Li, et al., “A Survey of Network Virtualization Techniques for Internet of Things Using SDN and NFV,” *ACM Computing Surveys*, vol. 53, no. 2, pp. 1–40, Jul. 2020.

[3] C. Jiang, X. Cheng, H. Gao, X. Zhou, and J. Wan, “Toward Computation Offloading in Edge Computing: A Survey,” *IEEE Access*, vol. 7, pp. 131 543–131 558, Jan. 2019.

[4] P. Mach and Z. Becvar, “Mobile Edge Computing: A Survey on Architecture and Computation Offloading,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, Mar. 2017.

[5] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, “A Survey on Mobile Edge Networks: Convergence of Computing, Caching and Communications,” *IEEE Access*, vol. 5, pp. 6757–6779, Jun. 2017.

[6] F. Dressler, F. Klingler, M. Segata, and R. Lo Cigno, “Cooperative Driving and the Tactile Internet,” *Proceedings of the IEEE*, vol. 107, no. 2, pp. 436–446, Feb. 2019.

[7] F. H. P. Fitzek, S.-C. Li, S. Speidel, T. Strufe, M. Simsek, and M. Reisslein, Eds., *Tactile Internet with Human-in-the-Loop*. Academic Press, 2021.

[8] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J. A. Zhang, “The Roadmap to 6G: AI Empowered Wireless Networks,” *IEEE Communications Magazine*, vol. 57, no. 8, pp. 84–90, Aug. 2019.

[9] F. Dressler, G. S. Pannu, F. Hagenauer, M. Gerla, T. Higuchi, and O. Altintas, “Virtual Edge Computing Using Vehicular Micro Clouds,” in *IEEE International Conference on Computing, Networking and Communications (ICNC 2019)*, Honolulu, HI: IEEE, Feb. 2019.

[10] N. Cha, C. Wu, T. Yoshinaga, Y. Ji, and K.-L. A. Yau, “Virtual Edge: Exploring Computation Offloading in Collaborative Vehicular Edge Computing,” *IEEE Access*, vol. 9, pp. 37 739–37 751, Jan. 2021.

[11] S. Wang, T. Tuor, T. Salonidis, et al., “Adaptive Federated Learning in Resource Constrained Edge Computing Systems,” *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.
[12] D. Bacciu, S. Chessa, C. Gallicchio, and A. Micheli, “On the need of machine learning as a service for the internet of things,” in *1st ACM International Conference on Internet of Things and Machine Learning (IML 2017)*, Liverpool, United Kingdom: ACM, Oct. 2017.
[13] M. Keller and H. Karl, “Response-Time-Optimized Service Deployment: MILP Formulations of Piece-Wise Linear Functions Approximating Bivariate Mixed-Integer Functions,” *IEEE Transactions on Network and Service Management*, vol. 14, no. 1, pp. 121–135, Mar. 2017.
[14] K. T. Kim, C. Joe-Wong, and M. Chiang, “Coded Edge Computing,” in *39th IEEE International Conference on Computer Communications (INFOCOM 2020)*, Virtual Conference: IEEE, Jul. 2020.
[15] F. Sattler, S. Wiedemann, K.-R. Muller, and W. Samek, “Robust and Communication-Efficient Federated Learning From Non-i.i.d. Data,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 9, pp. 3400–3413, Sep. 2020.

Falko Dressler (F’17) is full professor and Chair for Telecommunication Networks at TU Berlin. His research objectives include adaptive wireless networking (sub-6GHz, mmWave, visible light, molecular communication) and wireless-based sensing.

Carla Fabiana Chiasserini (F’18) is full Professor with Politecnico di Torino and a Research Associate with CNR-IEIIT. Her research interests include architectures, protocols, and performance analysis of wireless networks.

Frank H. P. Fitzek is leading the Deutsche Telekom Chair of Communication Networks. Current research directions are Tactile Internet, Quantum Communication, and Post Shannon with a general interest in SDN, NFV, ICN, network coding, cooperative networks, and robotics.

Holger Karl currently holds the chair of IT-Technology and Systems at the Hasso-Plattner-Institute, Potsdam. Before, from 2004 to 2021, he was chair of Computer Networks at Paderborn University. His interests are network softwarization, data centres, and mobile/wireless networks.

Renato Lo Cigno (SM) is Full Professor at the University of Brescia, Italy, where he leads the Advanced Networking Systems group. His interests are in wireless and vehicular networks, performance evaluation, and novel communication paradigms and architectures.

Antonio Capone (F’19) is full Professor at Politecnico di Milano and Dean of the School of Industrial and Information Engineering. His research areas include network softwarization and data plane programmability, resource optimization and planning of wireless networks.

Claudio Casetti (SM) is a full professor with Politecnico di Torino, Turin, Italy. His research interests include vehicular networks, 5G networks, IoT and has published over 200 papers on these topics. He is Senior Editor of IEEE Vehicular Technology Magazine.

Francesco Malandrino (SM’19) is a researcher at the National Research Council of Italy (CNR-IEIIT). His research interests include the architecture and management of wireless, cellular, and vehicular networks.

Vincenzo Mancuso (SM’21) is Research Associate Professor at IMDEA Networks Institute, Spain. His interests include analysis, design, and experimental evaluation of opportunistic architectures for intelligent wireless networks and edge computing.

Florian Klingler received his Ph.D. degree (Dr. rer. nat., summa cum laude) in Computer Science from Paderborn University, Germany, in 2018. His research focuses on protocols for adaptive wireless networks in highly dynamic scenarios.

Gianluca Rizzo is Associate Professor at the University of Foggia, Italy, and Senior Research Associate at HES SO Valais, Switzerland. His research interests are in performance evaluation of Computer Networks.