

Review

5G-MEC Testbeds for V2X Applications

Prachi V. Wadatkar ^{1,2,*} , Rosario G. Garroppo ²  and Gianfranco Nencioni ¹ 

¹ Department of Electrical Engineering and Computer Science, University of Stavanger, Kjell Arholms Gate 41, 4021 Stavanger, Norway; gianfranco.nencioni@uis.no

² Department of Information Engineering, University of Pisa, Via Girolamo Caruso, 16, 56122 Pisa, Italy; rosario.garroppo@unipi.it

* Correspondence: prachi.v.wadatkar@uis.no

Abstract: Fifth-generation (5G) mobile networks fulfill the demands of critical applications, such as Ultra-Reliable Low-Latency Communication (URLLC), particularly in the automotive industry. Vehicular communication requires low latency and high computational capabilities at the network's edge. To meet these requirements, ETSI standardized Multi-access Edge Computing (MEC), which provides cloud computing capabilities and addresses the need for low latency. This paper presents a generalized overview for implementing a 5G-MEC testbed for Vehicle-to-Everything (V2X) applications, as well as the analysis of some important testbeds and state-of-the-art implementations based on their deployment scenario, 5G use cases, and open source accessibility. The complexity of using the testbeds is also discussed, and the challenges researchers may face while replicating and deploying them are highlighted. Finally, the paper summarizes the tools used to build the testbeds and addresses open issues related to implementing the testbeds.

Keywords: 5G; multi-access edge computing; V2X communication; MEC orchestrator

1. Introduction

Fifth-generation (5G) mobile networks define three usage scenarios: Enhanced Mobile Broadband (eMBB), Ultra-Reliable Low-Latency Communication (URLLC), and Massive Machine-Type Communication (mMTC) [1]. Among these, URLLC represents an enabler of advanced automotive applications. The stringent latency requirements of these applications triggered research activities aimed at moving the network's core computational resources to the network's edge. Due to the shorter physical distance, edge computing resources can minimize latency, allowing for real-time services. Information or real-time data can be processed at the network's edge using computational resources and a large amount of data are sent to a data center.

The delivery of Machine Learning (ML) and Artificial Intelligence (AI) applications from the cloud to edge devices, or even to the sensors themselves, is one of the challenges of edge computing [2]. The difficulty is handling data reliably and effectively in contexts with limited computational and storage resources. There have been several discussions and developments regarding standards in edge computing. The European Telecommunications Standards Institute (ETSI) is one notable organization in this regard. The ETSI defines and standardizes Multi-Access Edge Computing (MEC) technology.

According to the 5G Automotive Association (5GAA), MEC is one of the essential enabling technologies for many of the anticipated services and applications for connected cars and self-driving [3]. The automobile sector needs MEC to deal with the exponential explosion of data in autonomous vehicles. As automobiles create a high amount of data daily, ways to efficiently process all the sensor data in the car and upload some of that data to the cloud are becoming significant difficulties. Furthermore, safety-related features must be available and we cannot rely on wireless connectivity. MEC can meet the stringent requirements of URLLC applications and ensure Quality of Service (QoS) by delivering



Citation: Wadatkar, P.V.; Garroppo, R.G.; Nencioni, G. 5G-MEC Testbeds for V2X Applications. *Future Internet* **2023**, *15*, 175. <https://doi.org/10.3390/fi15050175>

Academic Editor: Paolo Bellavista

Received: 7 April 2023

Revised: 5 May 2023

Accepted: 6 May 2023

Published: 9 May 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

MEC services to application developers and content suppliers. MEC allows operators to access the cloud infrastructure, regardless of the underlying Radio Access Network (RAN) (e.g., Third-Generation Partnership Project (3GPP) [4]). MEC's agnostic nature makes deployments effortless.

Vehicle-to-Everything (V2X) applications benefit from both 5G and MEC [5]. Vehicles can connect with the Road-Side Unit (RSU) infrastructure and other vehicles around them using this technology. The vehicle communicates with compatible equipment via a short-range wireless signal that is immune to interference and bad weather. V2X technologies are primarily employed to improve safety and avoid collisions. Vehicle-to-Infrastructure (V2I), Vehicle-to-Network (V2N), Vehicle-to-Pedestrian (V2P), Vehicle-to-Device (V2D), Vehicle-to-Grid (V2G), and Vehicle-to-Vehicle (V2V) are the primary types of V2X technology [6]. V2I technology gives real-time infrastructure data to drivers, including road conditions, traffic congestion, parking availability, and accidents. V2V communication allows cars to transmit messages to each other via a wireless network about what they are doing, including their speed, position, braking, direction of motion, and loss of stability. Dedicated Short-Range Communications (DSRC) technology is enabled, which is equivalent to Wi-Fi [7]. Almost all road safety services and most traffic management and efficiency services require the periodic transmission of V2V and V2I messages with extremely low latency and high reliability [8].

The performance of V2X applications depends on many parameters at different layers of the protocol stack, from the physical problems related to the 5G radio link conditions to networking issues related to the QoS guarantee of the exchanged data and application problems related to the storage and computational resources available for completing application tasks. To account for the large set of parameters, testbeds have been deployed to experiment with V2X applications without the limitations of theoretical or simulation models.

1.1. Contributions of this Paper

This paper is inspired and motivated by the development and deployment strategies conducted for 5G and MEC systems to support V2X applications. The survey in [9] demonstrates MEC integration with new technologies, including 5G and beyond, and presents open source activities, testbeds, and implementations. MEC deployments for vehicular network applications and services are provided in [10]. Some 5G use case scenarios, along with corresponding traffic models, that align with the requirements of the Standard Development Organizations (SDOs) and stakeholders are presented in [11]. In [12], the authors discuss the MEC deployment of resources, its migration abilities, and use cases for industrial verticals, whereas in [13], the authors present the capabilities of MEC orchestration, deployment, and open source tools. In [14], the authors demonstrate 5G testbeds for network slicing applications.

In summary, the contributions of the paper are as follows:

1. To develop a 5G-MEC testbed for V2X application experimentation, we investigate the literature on the 5G-MEC system and study its architecture. Based on the generalized idea proposed by standardization bodies and commercial vendors, as well as the requirements of Mobile Network Operators (MNOs), we suggest an overall system view, where the hardware and software components are considered. The software modules, such as the MEC platform (MEP), orchestration, and network flow, are placed as per the industry standards and their responsibilities.
2. This paper presents some existing state-of-the-art testbeds, shows how the 5G-MEC system is implemented, and demonstrates some use case scenarios. In the case of the V2X application testbeds, we present a summary of the Vehicle User Equipment (VUE) and Road-Side Unit (RSU) types of equipment.
3. The core part of this paper is the study of the usage complexity of the surveyed testbeds and whether the testbeds are open source and accessible. We investigate the difficulties of using the testbed resources, replicating the testbeds, and their limitations.

4. This paper summarizes the available tools used in the surveyed testbeds. In addition, we present other open source tools that are used to deploy the functionalities of the testbed. Since most functionalities can be deployed using open source tools, we discuss their compatibility issues and ways to integrate them with the support of the open source community.

1.2. Organization of this Paper

The paper is organized as follows. Section 2 presents some background information on 5G aspects and features, as well as the fundamentals of MEC, essential 5G use cases such as URLLC-classified V2X applications, and a global view of the 5G-MEC testbed for vehicular communication. Section 3 discusses the existing testbeds and implementations, where the surveyed testbeds are evaluated and classified into 5G-MEC, 5G-V2X, and 5G-MEC for V2X testbeds. Section 4 describes the usage complexity of the surveyed testbeds. Section 5 provides a summary of the tools used to build the testbed. Finally, Section 6 concludes the paper and addresses the unresolved issues. Figure 1 shows a map and layout of this paper. We present the most commonly used acronyms and their definitions at the end of this paper.

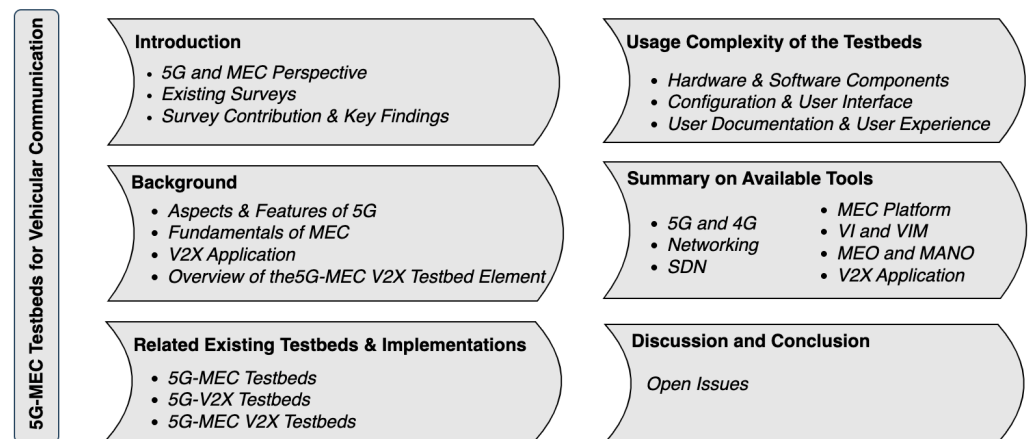


Figure 1. Map and layout of this paper.

2. Background

An MEC system provides the backend support for 5G to implement critical latency applications such as URLLC in which latency can be required to be a maximum of 1 ms. The 5G-MEC paradigm can provide latency and other critical requirements. The MEC system helps to analyze real-time scenarios in the case of V2X applications and can perform the required computation tasks. The MEC system provides computational capabilities using virtualization and softwarization technologies such as Software-Defined Networking (SDN) and Network Function Virtualization (NFV).

This section provides detailed insights into 5G-MEC and V2X applications and is divided into four subsections that discuss 5G features and capabilities, the fundamentals of MEC, V2X applications, and the general elements of the 5G-MEC V2X testbed. The MEC fundamentals include integrating MEC deployment into 5G and softwarization technologies such as SDN and NFV with MEC. The 5G-MEC V2X testbed’s general elements are divided into two parts: hardware and software components.

2.1. Aspects and Features of 5G

5G is growing as the new reference architecture in the telecommunication industry. The 5G network introduces modular Network Functions (NFs) for the control plane and user plane in both the Access Network (AN) and Core Network (CN) [15]. NFs provide specific network capabilities according to the application requirements. 5G can support different vertical sectors and industries with machine- and human-type communication

and is expected to provide native support for MEC with its low latency and high network efficiency.

The main usage scenarios of 5G are divided into three categories [16]. The first is eMBB, which provides a data rate higher than 4G accompanied by moderate latency improvements. eMBB supports the development of use cases such as 360-degree streaming video, Augmented Reality/Virtual Reality (AR/VR) media, and UltraHD applications. mMTC provides connectivity to an enormous number of devices. 5G will replace the previous technologies such as LTE-M, Low-Power Wide Area Network (LPWAN) technologies, and NarrowBand Internet of Things (NB-IoT). URLLC requires the 5G core deployment to reduce end-to-end (E2E) latency, accommodates emerging services and applications with high-reliability requirements, and enables the support of applications such as remote control and autonomous driving.

ITU-R M.2410 [17] presents the 5G requirements for eMBB, mMTC, and URLLC applications. The user plane latency required by eMBB and URLLC is 4 ms and 1 ms, respectively. In addition, eMBB requires a 20 Gbps downlink data rate and mMTC requires a connection density of 10^6 devices per km^2 .

The core of the 5G system is the service-based architecture (SBA), which defines the interactions between NFs responsible for the control plane and data plane procedures. Figure 2 depicts the non-roaming 5G system reference architecture. The bottom layer of the architectural elements is involved in transporting the user data, whereas the upper layer includes the essential NFs within the control plane. The NFs within the 5G reference architecture are Access and Mobility Management Function (AMF), Session Management Function (SMF), Network Slice Selection Function (NSSF), Network Repository Function (NRF), Unified Data Management (UDM), Policy Control Function (PCF), Network Exposure Function (NEF), Authentication Server Function (AUSF), and User Plane Function (UPF). 3GPP TS 23.501 [18] presents the functionalities and responsibilities in detail. In the 5G system, the separation of user plane functions and control plane functions leads to flexible deployment and scalability. The 5G system model is defined to enable technologies such as SDN and NFV and to provide data connectivity.

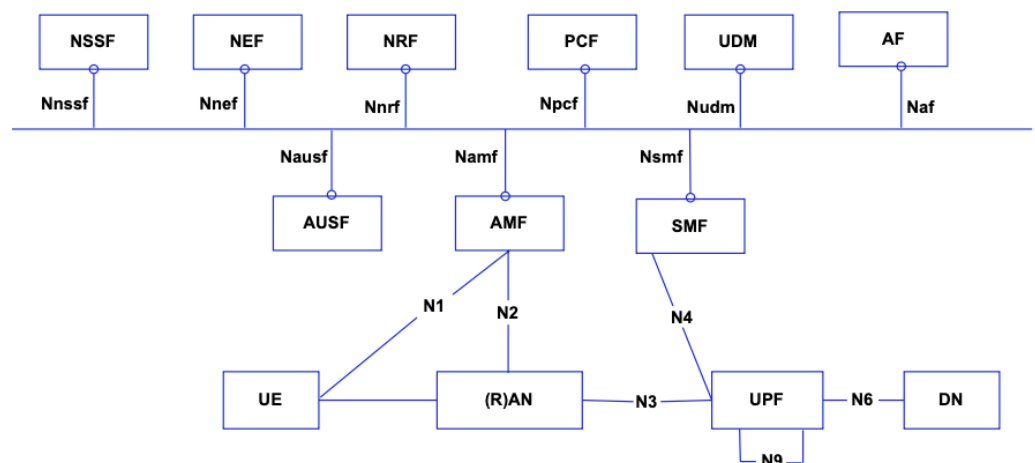


Figure 2. 5G system reference architecture.

2.2. Fundamentals of ETSI-MEC

URLLC and eMBB applications require computational capabilities, which becomes an inherent concern due to the limited storage and computational abilities of the end-users. Due to these minimum requirements, the 5G network will face unusual traffic volumes. At the same time, MEC is an emerging concept in the 5G network that has the capacity to process a lot of data inside the RAN. MEC provides support for applications where real-time processing is needed. Fog computing and cloudlet are alternative architectures that, like MEC, offer cloud capabilities at the network’s edge. The core idea of MEC is to deploy the cloud computing capabilities within the RAN, close to the end-user [19]. The

ETSI is a standardization organization for MEC that provides a reference architecture for deploying network functionalities. The ETSI divides the MEC system into different MEC building blocks and discusses the contribution of each building block.

2.2.1. MEC Architecture

The MEC architecture provides an extensive high-level overview of the MEC system, enabling MEC applications to operate uninterrupted in a multi-access network. The MEC system reference architecture comprises an MEC management entity and one or more MEC hosts (MEHs) [20]. Figure 3 depicts the MEC system reference architecture defined by the ETSI. Below, we present the different architectural elements (referred to in this paper as MEC building blocks).

MEC Host

An MEH is an element that contains the MEC Platform (MEP); the Virtualization Infrastructure (VI), which is a set of computing and storage resources; and the network for the MEC applications (MEC apps). The data plane in the VI performs the traffic rules received by the MEP. The MEP routes the traffic among other services, applications, and access networks. Usually, these functions are run on Virtual Machines (VMs) that provide specific network services, with isolation supported by virtualization.

MEC Platform

The MEP is an integral component that delivers a set of tasks necessary for MEC apps to run on a specific VI, configures the DNS proxy and server, and provides an environment that both offers and leverages MEC services. It is up to the MEP to grant access to storage and time-of-day data.

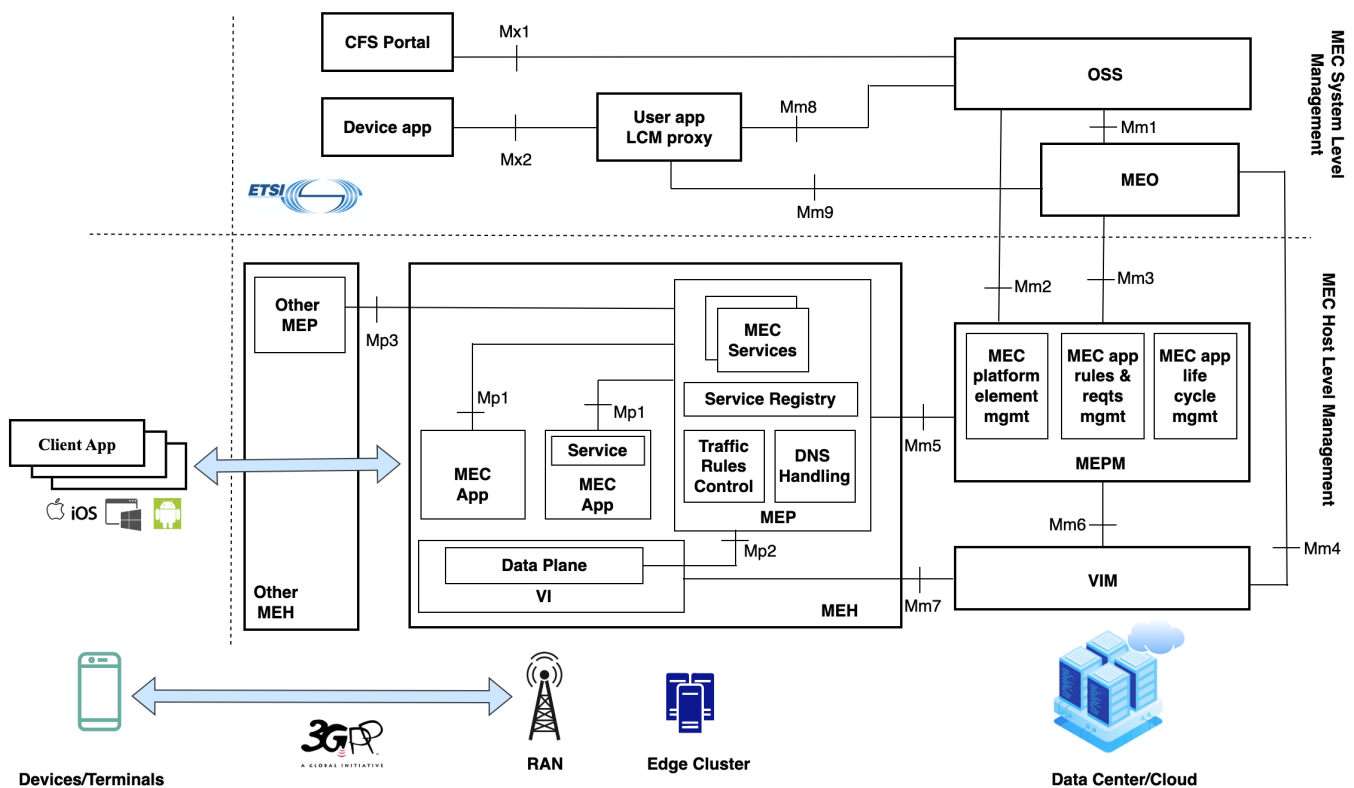


Figure 3. ETSI MEC reference architecture.

MEC Management

The MEC management entity consists of two components, MEC system-level management and MEC host-level management. MEC system-level management interacts with

MEC host-level management and the access network. MEC system-level management consists of the following elements:

- *MEC Orchestrator (MEO)*, which is the brains of the MEC as it contains the complete global overview of the MEHs, services, resources, and topology. It determines the most appropriate MEH based on the user requirements. An MEO can perform system-level tasks with the Operating Support System (OSS) and the user application life cycle management proxy. It can trigger application termination, relocation, and instantiation as needed.
- *OSS*, which receives requests via the Customer Facing Service (CFS) portal and makes decisions based on these requests that enable or terminate applications. For further processing, an MEO handles the requests. Where supported, the OSS can receive requests to relocate applications between the MEC system and the cloud.
- *User Application Life Cycle Management Proxy*, which is an MEC application used for the life cycle management of UE applications. It enables device applications to trigger instantiation, onboarding, and application termination for users, where supported. It is a function that can be accessed from the mobile network and can inform the state of the user application to the device application. For further processing of the requests, it interacts with the MEO and the OSS.

MEC host-level management consists of the following elements:

- *MEC Platform Manager (MEPM)*, which is responsible for a set of applications that provide critical information to the MEO concerning the MEP, provide information about the life cycle management of MEC apps, configure the DNS, manage rules, and authorize services.
- *Virtualization Infrastructure Manager (VIM)*, which allocates and manages the virtualized resources of the VI. It provides a virtual environment to host a VM and can rapidly deploy applications, where supported. It gathers and reports on the state of the performance and fault information of the virtualized resources.

2.2.2. 5G and MEC Integration

The MEC system interacts with the 5G SBA as shown in Figure 4. The MEP is treated as an application function (AF) to access the 5G core network [21]. The UPF configuration and MEC deployment scenario depend on the network and MEC provider. UPF integration into the MEC system is the primary component, which can have a distributed and customizable data plane from the perspective of the MEC system. The traffic configuration of the data plane is managed through the NEF-PCF-SMF route. The MEH elements can be deployed in the data network of the 5G system, whereas the NEF is deployed in the MEO [22].

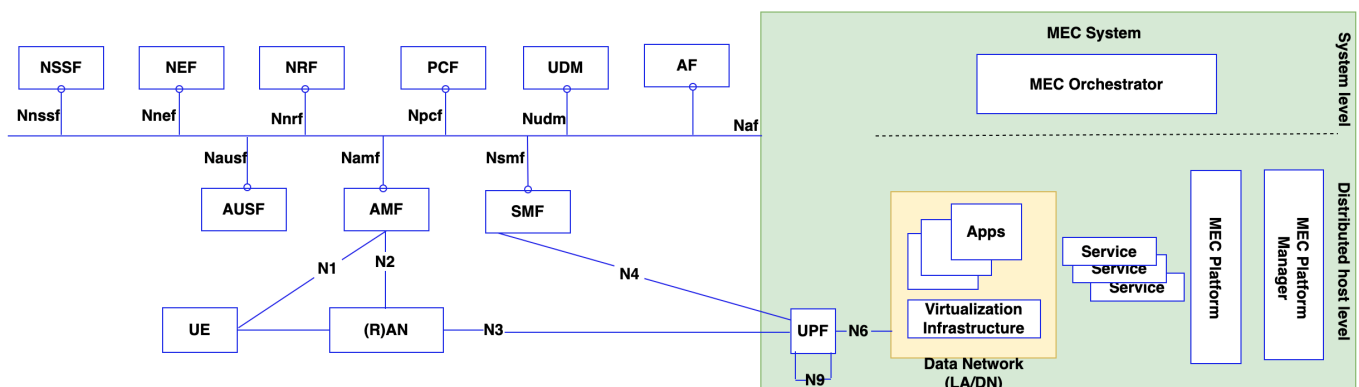


Figure 4. MEC deployment in 5G reference architecture.

2.2.3. Softwarization Technologies

Softwarization technologies are rapidly growing as are the network paradigms for network and service providers. SDN and NFV are the key enabling technologies in the network paradigm that can support the 5G-MEC system.

SDN aims to provide network management, programmability, and flexibility of networks. By separating the control plane from the data plane, SDN allows seamless network access. The decoupling is performed via a logically centralized network intelligence (SDN controller), and the data plane consists of network devices that forward the packets. Google and Facebook have already introduced SDN into their respective data centers [23].

NFV allows network operators to provide network services using virtual infrastructure. This allows NFV to deploy software network functions on off-the-shelf hardware rather than deploy hardware-based proprietary network functions. NFV provides cost efficiency and the timely benefits of cloud computing. The authors of [24] present an NFV reference architecture in which Virtual Network Functions (VNFs), NFV Infrastructure (NFVI), and Management and Network Orchestration (MANO) are the key elements.

2.2.4. MEC in the NFV Environment

In the future, MNOs will deliver the network as a service (NaaS) using NFV. To achieve this, MNOs use the VI to integrate the network elements, MEC apps, and components into the infrastructure [25]. The data plane in the VI routes traffic between applications, various services, access networks, and the MEP. The data plane configuration is managed via the Mp2 reference [26]. The VI is a fundamental component of the MEC and NFV architecture. The MEPM shown in the MEC system reference architecture is transformed into a “Multi-Access Edge Platform Manager-NFV” (MEPM-V) that transfers the Life Cycle Management (LCM) section to one or more VNF Managers (VNFMs), which is smart of the NFV MANO. The MEO in the MEC reference architecture is transformed into an MEC Application Orchestrator (MEAO) that uses an NFV Orchestrator (NFVO) for the orchestration of resources, along with a set of MEC application VNFs as one or more NFV Network Services (NSs) [27]. By deploying the MEC in the NFV environment, low latency and scalability are the main benefits. The NFV delivers scalability and the MEC offers low latency.

2.3. V2X Applications

For vehicular communication, the automotive and telecommunication industries have invested in the connected car market as it supports digital transformation by exploiting communication technologies [28]. V2X communications include V2I, V2P, and V2N communications, which help to enhance vehicular traffic and safe driving, as well as provide other services. The automotive industry supports acquiring MEC-based vertical segments in the 5G ecosystem. V2I communications require a low-latency environment (less than 10 ms) [29]. MEC has been proposed as a possible solution for these stringent performance requirements, and it facilitates direct communication between nodes about V2X-related information via an underlying network. MEC technologies provide an ideal setting for use cases, allowing them to be linked, communicate, and share information about traffic and maps. For V2X applications, which can also stream videos and AR, the MEC system provides cloud computing capabilities at the network’s edge and its access-agnostic feature ensures the reliability of the deployment via an underlying communication network.

2.4. Overview of the 5G-MEC V2X Testbed Elements

To develop and deploy a 5G-MEC testbed for V2X applications, the above-mentioned elements need to be considered. The 5G-MEC system delivers the support needed for V2X applications. Figure 5 shows a real-time scenario that involves V2X communication and backend activities within 5G and MEC that satisfies the performance requirements. In addition, we investigated and identified the generalized elements involved in building these testbeds. This section provides an overview of the testbed and is divided into

two parts. The first discusses the hardware components and the second discusses the software components.

Hardware elements

The hardware components are fundamental aspects of the testbed. In terms of vehicular communication applications, the testbed consists of multiple vehicles equipped with On-Board Units (OBUs) with RSU connectivity. The vehicles can communicate with each other and the infrastructure via standard technologies. The RSUs are connected with the rest of the network using Ethernet links or other industry standards. The end-user application for the VUE is referred to as the client app and runs from within the OBU. The OBU establishes a communication link with the RAN infrastructure via eNB, AP, or gNB. Each RAN infrastructure is equipped with an MEH connected to the AP, eNB, and gNB. The MEH may or may not be present at the RAN infrastructure. The RAN infrastructure can be provided by the MNOs or can be built using a Software-Defined Radio (SDR) such as a USRP B210. The MEH is a compute node that provides resources for running MEC applications and delivering services related to the MEC. The MEH is a high-performance computing unit. The transport network is the overlay network that provides connectivity between the RAN, MEC systems, and the data center/cloud. The data center/cloud can be private or public. In the case of a private cloud, high-performance servers are needed, with or without a GPU, depending upon the orchestration responsibilities.

Software elements

The software elements represent the main virtualized components of the data plane. The application server/controller administrates and monitors the end-user application of the VUE. An application server can be present in the MEH or a separate cluster. The 4G and 5G CNs represent the core network functionalities. The MEC system is deployed using different MEC building blocks: the MEP, MEPM, VIM, and MEO. An MEP needs to be located within each MEH, whereas the rest of the MEC elements have no constraints on their location. In addition, the testbed can include an SDN controller to perform network management and control the traffic flow. As presented by the ETSI, an MEC in an NFV environment offers an NFVO for orchestrating the VNFs and VNFMs. The NFVO is located alongside the MEO. Based on our studies, the orchestration of MEC systems and VNF needs high-performance servers, and their allocation would be most appropriate and suitable within the data center/cloud infrastructure.

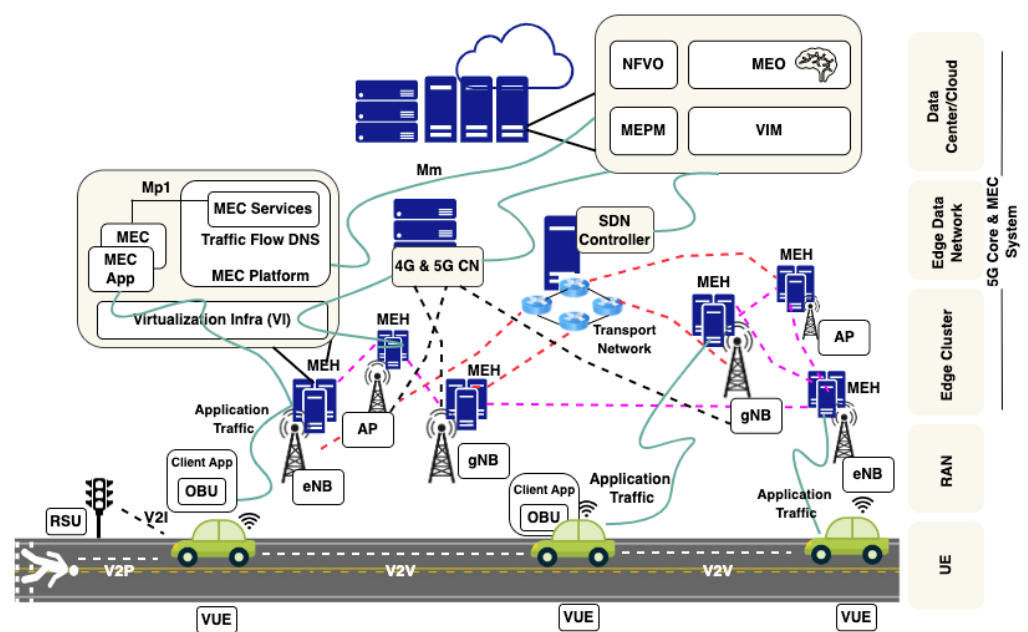


Figure 5. Overview of the 5G-MEC V2X testbed elements.

3. Existing Testbeds and Implementations

In this section, we focus on existing and open source activities on 5G-MEC testbeds for vehicular communications. This section is organized into three subsections: Section 3.1 describes a set of important 5G-MEC testbeds, whereas Section 3.2 focuses on 5G-V2X testbeds. Section 3.3 presents the features of the selected 5G-MEC V2X testbeds. We describe the deployment strategies and their respective use cases for each of the considered testbeds.

3.1. 5G-MEC Testbeds

Below, we present the selected 5G-MEC testbeds.

3.1.1. 5GCity

The 5GCity testbed involves different platforms and services and can support applications such as video streaming and smart cities. The 5GCity testbed provides an MEC-neutral host platform that focuses on UltraHD video streaming [30]. The testbed was deployed in three European cities, Bristol, Lucca, and Barcelona. The project developed a 5GCity scalable orchestrator using OpenStack as the VIM and an SDN RAN controller. The MEAO is deployed using Open Source MANO (OSM) [31] and implements the Mm1 interface in a proprietary way [32]. This allows for the creation and deletion of MEC applications and services. Furthermore, the functionalities of the MEPM are performed. The MEC platform is deployed with an Mp1 interface that interacts with the MEC applications. The 5GCity Software Development Kit (SDK) supports the resource placement and network slice application using the slice manager element. The 5GCity SDK includes a monitoring manager for the VNFs and MEC applications [33].

3.1.2. Mosaic5G

The Mosaic5G platform [34] provides flexible and scalable service deployment. The platform includes five software elements such as Open-Air Interface (OAI) [35] for the RAN and CN functionalities of the LTE network. FlexRAN [36] is an open source implementation of Software-Defined RAN (SD-RAN) and is deployed to control tasks within different base stations in a centralized manner or perform scattered control policies. Reconfiguring the control functionalities with a RAN helps to manage those tasks. Low-Latency MEC (LL-MEC) is the MEC platform that separates the data plane and controls the plane traffic at the network's edge. The MEC functionalities and services are enabled using the LL-MEC platform. Mosaic5G deploys the NFV MANO platform using JOX to provide E2E network slices. Store includes modules for monitoring and controlling applications.

3.1.3. 5GINFIRE University of Bristol Testbed

The testbed of the University of Bristol, developed within the 5GINFIRE project, was deployed for smart city use cases [37]. The testbed was developed using open source and licensed solutions. In this testbed, the MEP and MEPM are deployed using the CloudBand of Nokia, which implements the MEC functions and the VIM. The SDN architecture is based on the NetOS controller, whereas OSM is deployed to orchestrate the VNFs. Inter-Digital provides media services, including content distribution, which are utilized for the 5G smart tourism initiative. The media services are managed using OpenStack. The testbed involves mini-scale sensor nodes and equipment such as a Raspberry Pi. The necessary data are acquired and processed at the data center.

3.1.4. EdgeX Foundry

EdgeX Foundry [38] is an open source platform that is placed at the network's edge to build an MEC framework for IIoT applications. EdgeX was developed under the Linux Foundation Edge (LF Edge) project. At the code level, the golang version, EdgeX-Go, of EdgeX Foundry has seven subdirectories: cmd, Api, Docker, Internal, pkg, snap, and bin. The cmd folder contains the project program entry main.go and describes key information such as the IP and port of the called EdgeX microservice; the Api folder contains the

Application Programming Interface (API) in the form of HTTP for each microservice in the entire framework; the Docker folder contains instructions required to create the image; and the Internal folder is responsible for the initialization of the device microservice driver and the processing of northbound command requests. Snap contains all EdgeX-Go-based microservices and the bin contains the executable files. To sum up, EdgeX Foundry provides a dual conversion engine: one for uniformly converting data from different sensors and devices using various communication protocols and formats into common EdgeX data structures, and the other for specifying customer-specific data using a TCP/IP-based protocol format and providing these data to applications, enterprises, and cloud systems. EdgeX can run on small-scale equipment such as a Raspberry Pi [39].

3.2. 5G V2X Testbeds

The 5G-V2X testbeds are essential for testing V2X services without using the ETSI MEC architecture. For instance, these testbeds can be used to gauge the effectiveness of various edge and fog computing architectures that are alternatives to the ETSI MEC architecture. Additionally, these testbeds enable the execution of experiments to assess the effectiveness of well-known communication systems in a real-world setting (for instance, the network effectiveness of vehicle communications at real car speeds). The testbeds enable the testing of novel network protocols, for example, reducing the power consumption on the radio access network or increasing the exchanged data reliability and/or security. Finally, the testbeds can also be utilized to test new technologies on radio interfaces such as mmWave and evaluate the performance achieved by the cooperation among different Radio Access Technologies (RAT) in a multi-RAT scenario. The main challenges of using 5G-V2X testbeds are as follows:

- The testbed/experiment complexity for troubleshooting and programming.
- Interoperability between MNOs and vendors for validating the end-to-end solutions.
- Real-world scenarios, such as network conditions and environmental factors, can impact testbed conditions.
- Security and safety issues of the connected vehicle and controlled environment to test the experimental analysis.
- Regulatory compliance with operating a 5G V2X testbed.

The selected 5G-V2X testbeds are discussed below.

3.2.1. 5GVINNI Munich Experimentation Facility Site

The 5GVINNI project [40] includes the 5GVINNI Munich testbed as the facility's design solution. The testbed supports the eHealth use case named "Remote interventional support for emergency care application—mobile ultrasound". The testbed infrastructure supports MEC functionalities, 5G RAN, and the 5G core of Huawei. The SDN component is based on the floodlight controller, and Huawei deploys the MANO and NFVI. For the NFVI deployment, Docker container and Mininet are used. Mininet enables SDN capabilities that allow NFs inter-connectivity running on Docker through OpenvSwitch. The 5G site is connected to the core network using the Ethernet link for a high data rate. An ambulance is the UE where the 5G modem is mounted, which moves within the coverage area. The MEC functionalities are not fully deployed; Docker is used for the deployment of the MEC services and applications. The MANO includes an NFVO that enables VNF instantiation and termination. The VNFM compiles a software module to initiate and manage the UPF Docker container [41].

3.2.2. 5GINFIRE IT-Av Automotive Testbed

The 5GINFIRE project includes various testbeds, one of which is the "IT-AV automotive" environment, which was developed for experimenting with V2X applications in 5G systems. MEC-based solutions have allowed the automotive sector to facilitate V2X communications and exchange vehicle-related information between nodes via the underlying communication network. In [42,43], the authors present a novel V2X algorithm named

Vulnerable Road User Safety (VRU-safe) that runs on the considered architecture. VRU-safe provides a low-complexity system, efficiency, and scope for identifying and predicting the hazardous events between vehicles and Vulnerable Road Users (VRUs). The performance and results have been evaluated in the real-time 5G testbed of “IT-AV automotive”.

3.2.3. 5G V2X Testbed for Cooperative Automated Driving

This testbed is a 5G V2X wireless testbed built on an adjustable and reconfigured SDR that is designed for cooperative autonomous driving [44]. The key novelty of this testbed is the Radio-Frequency (RF) front end, which is reconfigurable. In particular, the testbed allows testing new OFDM waveforms built on parameters such as pulse-shaping, a dynamic and self-contained structure layout, GNSS-assisted hybrid synchronization, and scheduled multiple access for reducing latency. The radio interface, which includes modulation, coding, a frame structure, and multi-antenna systems, is freely designed and modified during the tests. The radio subsystem is simple to connect with the OBU inside the vehicle, which is a general-purpose computer-based system. The radio module can perform real-time analysis that enables live transmission, testing, data collection, and offline analysis for channel and automotive prediction. The radio module has a small form factor and power dissipation, making it ideal for integration into automobiles with limited performance and power supply. For the RF Unit (RFU) setup, the radio subsystem construction components are NI/Ettus-based USRP X310 SDR platforms. A high-performance Intel hardware-based computer running a lightweight Linux operating system is utilized for the Baseband Unit (BBU). The RF external (RFE) subsystem is self-built and integrated with the power amplifier and switches based on Time Division Duplex (TDD) to increase transmitting power and reception sensitivity.

Two types of use cases are considered in the testbed: Type 1 use cases exhibit predictable behavior and are limited to three phases: planning, initialization, and execution; and Type 2 usage scenarios are based on emergency braking and cooperative emergency maneuvers. This testbed describes short-distance platooning and lane merging as the main Type 1 usage scenarios. Conversely, Type 2 usage scenarios are distinguished by quick response times and the inability to plan ahead of time, which typically occur in emergency circumstances. The lab measurements and field tests carried out using this testbed show that the emergency brake triggered with a latency of less than a millisecond [45].

3.3. 5G-MEC V2X Testbeds

For the 5G-MEC V2X scenario, we present the following testbeds.

3.3.1. 5G-VINNI Moving Experimentation Facility Site

The 5G-VINNI project developed a moving experimentation facility to support public safety communications and emergency response and disaster relief applications. The moving experimentation facility [46] accommodates the SATis5 testbed and uses a Rapid Response Vehicle (RRV). The SATis5 testbed [47] includes an edge node and a central core network node for the satellite backhaul connection. At the same time, the UE connects to the edge node and central core network nodes via the satellite backhaul network. The backhaul is the transport layer used to exchange messages between the edge node and the central core network site. The Fraunhofer FOKUS Open5GCore toolkit was deployed for the implementation of the 3GPP 5G core network. The Open5GCore toolkit provides the 3GPP Release 15 core network functionality and 5G New Radio integration. The testbed considers the NSA architecture where the traffic is split between the 4G and 5G at eNodeB. The testbed uses the 4G eNB for the RAN instead of the 5G gNB for the base stations because the 5G gNB was not commercially available at that time. The Open5G core integrates with the iDirect satellite hub platform to operate the satellite network. The iDirect hub offers integration with the overall 5G architecture and SDN functionalities with the help of the Satellite Convergence (SatConv) layer and includes an MNO for managing the modems

and satellite hub network. The iDirect Velocity Intelligent Gateway system introduces the 3GPP core network (SatCore) into the satellite hub platform.

The core network within the central node differs from that of SatCore and supports the satellite backhaul network. The satellite network operations are virtualized by moving their execution environment from a dedicated server to a VM with the help of the OpenStack Pike VIM. Satellite VNFs incorporate the SatRAN software element, the satellite 3GPP Core Network function (SatCore), the satellite Network Management System (NMS), and the auxiliary VNFs implemented on the same system that utilizes the OpenStack VIM. The iDirect Velocity Intelligent Gateway system is connected to OSM with regard to MANO. The SATis5 testbed [48] presents the SATCOM integration with the 3GPP 5G architecture for an over-the-air demonstration.

3.3.2. Smart Highway V2X Testbed

The Smart Highway V2X testbed [49] is a section of the CityLab testbed [50]. The Smart Highway V2X testbed, located in Antwerp (Belgium), is built on top of an open highway and comprises a highway strip with RSUs installed. These RSUs can handle V2X radio connections, such as short-range based on ITS-G5 and C-V2X with a PC5 interface operating at 5.9 GHz and long-range C-V2X Uu, allowing them to communicate with vehicles equipped with these radio access technologies. 4G is used for long-distance connectivity. Experimenters can choose which communication modules to test on the RSUs, which are equipped with commercial and SDR modules. Each RSU is fiber-connected to the virtualized cloud-based backbone. The Smart Highway V2X testbed [51] includes two mobile OBUs, enabling V2X communications. The OBUs are also supported with long-range communications using 4G connectivity from the Mobile Network Operator (MNO). The OBUs are equipped with GNSS modules for localization.

The Smart Highway V2X testbed provides MEC collocation with the RSU and tests the benefit of the intelligent application in a real-time environment [52]. For the MEC platform deployment and RSU, Kubernetes cluster deployment is performed to monitor services based on Prometheus [53]. The real-time monitoring supports MEC applications and generates the statistics for the containerized MEAO (cMEAO). The cMEAO makes decisions about the MEC host availability and location of the vehicles.

3.3.3. 5G Carmen

The 5G Carmen testbed [54] involves the integration of different MEC systems and was developed in three countries: Italy, Germany, and Austria. The MEC platform in the 5G Carmen testbed consists of three interfaces: a connection to a Packet Gateway (PGW) via an SGi-interface, a public IP address that is exposed to other MEC platforms, and VPN interfaces for platform management functions and applications that are deployed on the platform by third parties. The setup in Germany includes the Deutsche Telekom integration and operation of the LTE/5G MEC system. The Nokia infrastructure provides the MEC platform, local edge processing capabilities, and the latency-optimized access path for LTE latency and reliability-sensitive applications. The MEC Edge cloud infrastructure is based upon the Nokia Airframe hardware, Ubuntu Linux and KVM virtualization, and the Application Life Cycle Management [55]. The setup in Germany deploys the BMW Cooperative Maneuvering Application. TIM proposed a setup in Italy, where Nokia provides the MEC system solution and the AMQP broker platform is provided by TIM. Some MEC platforms and application entities are deployed in virtual machines, whereas the remainder are realized in containers. A container-based service mechanism is used for life cycle management services. The application orchestration is provided via Helm charts. The VM orchestration in the MEC platform is performed using Nokia CBAM. The setup in Austria is managed and operated by Magenta Telekom Austria (MTA) with the MEC environment located within their network infrastructure. The setup is based on the OpenStack Red Hat Release 13 that can host multiple independent tenants. Currently, the setup is hosting the onboarding of MobileEdgeX, which provides a low-latency computing platform.

3.3.4. Akraino Connected Vehicle

The Akraino connected vehicle blueprint [56] focuses on the development of the MEC platform, specifically for V2X applications. Akraino Edge Stack [57] is an open source software stack that was created by the Linux Foundation in February 2018. It supports access methods such as 5G, LTE, Wireline, and Wi-Fi and provides high availability that is optimized for edge computing systems and application cloud services. It is designed to improve the state of the edge cloud infrastructure for enterprise edge, Over-The-Top (OTT) edge, and carrier edge networks, giving users a new level of flexibility to rapidly scale edge cloud services, maximizing the number of applications or users supported, and helping to ensure the reliability of the system at runtime. The Akraino open source project is organized around blueprints, using them to implement diverse edge computing-based use cases. Each blueprint contains the declarative configuration of the three-layer stacks described above, such as cloud platforms, APIs, and applications. Specifically, it includes the configuration statement, hardware, software, tools for managing the entire stack, and delivery point (PoD, pod of delivery) of edge computing applications or architectures in a specific scenario. Each published blueprint enables automated edge infrastructure builds, edge stack-site deployment, upstream and downstream project integration, automated orchestration, etc.

3.3.5. 5G-DRIVE

The 5G-DRIVE testbed [58] focuses on developing and evaluating eMBB and V2X scenarios. The network infrastructure built into the 5G-DRIVE testbed is based on Nokia's NetLeap LTE test network. In the 5G-DRIVE testbed, a virtual mobile network is enabled with its own EPC. The eNodeBs are interconnected with the OpenStack cloud and SDN-enabled backhaul. For the experimental tests, a public site in Finland was used. Some of the research activities focused on testing the coexistence of ITS-G5/LTE-V2X [59]. The 5G-DRIVE is equipped with a C-V2X box, a traffic light equipped with a C-V2X RSU, and a tablet in the car. The field test involved the LTE network (2.6 GHz), a 5.9 GHz RSU, and a 5.9 GHz NEBULA OBU. The testbed has been used to experimentally evaluate the latency and packet loss rate between the RSU and OBU [60].

Table 1 presents a categorization of the considered testbeds according to their type and open source accessibility. Furthermore, the equipment of the VUE and RSU are mentioned in Table 1 for the available 5G-V2X and 5G-MEC V2X testbeds. In the case of the 5G-MEC testbeds, the VUE and RSU are not applicable. In addition, Table 1 names the sensors used in the surveyed testbeds for collecting the necessary data.

Table 2 presents the surveyed 5G-MEC and 5G-V2X testbeds based on the categorization of 5G usage scenarios such as eMBB, URLLC, and mMTC. Furthermore, the surveyed testbed use case studies are briefly described. The same information is summarized in Table 3 for the 5G-MEC-V2X testbeds.

Table 1. Summary of the considered testbeds, classified by type and open source accessibility.

Project/Title of the Testbed	Type of Testbed	Open Source	VUE	RSU	Sensors	References
5GCity	5G-MEC testbed	Open source	Not applicable	Not applicable	Smartphone camera, CCTV camera, and microphones	[32]
Mosaic5G	5G-MEC testbed	Open source	Not applicable	Not applicable	Wearable sensors for monitoring patient health-related data (specifications not provided) and LiDAR sensor	[61]
5GINFIRE University of Bristol testbed	5G-MEC testbed	Partially open source	Bike rider helmet, Raspberry PI, 360-degree camera, and audio	Not applicable	Camera, microphone, air quality, and noise sensors	[37]
EdgeX Foundry	MEC testbed	Open source	Not applicable	Not applicable	Video cameras, motion sensors, and audio sensors	[38]
5GVINNI Munich experimentation facility	5G V2X testbed	No	Ambulance	Huawei 5G RAN C-Band 3.41 GHz, Remote hospital integration	Ambulance, equipped with HUAWEI 5G experimental UE, a high-resolution camera, and an ultrasound system. Hospital is equipped with a display monitor, Camera, PC, and remote medical expert.	[40]
5GINFIRE IT-Av automotive testbed	5G V2X testbed	Partially open source	Bike and commercial car	Single-board computer (SBC), DSRC wireless interface (IEEE 802.11p), WiFi interface (IEEE 802.11b/g/n), 4G Interface, GPS receiver, antennas for each technology	OBU internal sensors available (accelerometers, heading, speed, link-quality connection, GPS, compass, RSSI, neighboring car's density, etc.)	[62]
5G V2X testbed for cooperative automated driving	5G V2X testbed	No	Commercial autonomous car	NI/Ettus USRP X310 SDR with Intel-based PC	LiDAR, Radar, GPS/IMU, and CAN odometry	[44]
5GVINNI moving experimentation facility	5G-MEC V2X testbed	No	SES's Rapid Response Vehicle (RRV)	Not applicable	Phone, a wearable, a car, and a fixed IoT sensor aggregator	[63]
Smart highway V2X testbed	5G-MEC V2X testbed	Partially open source	BMW X5 xDrive25d LO (2014) provided by the University of Antwerp	Intel Xeon Broadwell E5-2620V4 SDR: Ettus USRP N310 ITS-G5: Cohda Wireless MK05 RSU LTE-V: Cohda Wireless MK6c EVK Mikrotik wAP LTE kit	Lidar sensor: Velodyne VLP-16; Camera and CAN sensor box: Raspberry Pi 3 B+ with can shield	[64]
5GCarmen	5G-MEC V2X testbed	No	BMW and CRF cars	Hardware not available	Vehicle and RSU sensors such as LiDAR, Camera, Radar, GPS	[54]
Akraino connected vehicle	5G-MEC V2X testbed	Partially open source	Tencent V2X business	Tencent V2X business	Camera, Radar, LiDAR, GPS, Accelerometers, and Gyroscopes	[56]
5GDRIVE	5G-MEC V2X testbed	No	Automated passenger car called "Marilyn", equipped with sensing systems and ECUs needed for automated driving functions. Finland's VTT Technical Research Centre provided the car equipment used in the 5G-DRIVE testbed.	Cohda Wireless, Dual ITS-G5 DSRC radio GPS, ITS-G5, DSRC, and WiFi (802.11abgn)	3 different 905 nm LiDARs, 24 GHz radar, Stereo camera, Inertia unit	[65]

Table 2. Surveyed testbed use cases—5G-MEC and 5G-V2X.

Project/Title of the Testbed	eMBB	URLLC	mMTC	Short Description of Case Study	References
5GCity	Yes	No	Yes	The 5GCity project focuses on different use cases (UC). UC1 focuses on unauthorized waste dumping prevention. UC2 examines the platform's capabilities using a single infrastructure to provide services to multiple MNOs. UC3 analyzes and edits the real-time video acquisition of audience smartphones during an event.	[66]
Mosaic5G	Yes	Yes	Yes	The Mosaic5G platform has been used for a few use cases, such as critical e-Health, V2N communication for intelligent transportation systems, and multi-service management and orchestration for smart cities.	[67]
5GINFIRE University of Bristol testbed	Yes	No	Yes	The 5GINFIRE University of Bristol testbed provides a smart city safety use case with capabilities that address daily needs ranging from parking and water treatment to city security.	[68]
EdgeX Foundry	Yes	No	Yes	The EdgeX platform has been used to build automation, video sensor security, and IoT control system use cases.	[39]
5GVINNI Munich experimentation facility	Yes	Yes	No	The 5GVINNI Munich facility has been used to perform the eHealth use case named "Remote interventional support for emergency care application—mobile ultrasound."	[69]
5GINFIRE IT-Av automotive testbed	Yes	No	Yes	The 5GINFIRE IT-Av automotive testbed includes three use cases: 5GCAGE, 5G driving trainer (5G-DT), and VRU-SAFE. The 5G-CAGE use case aims to implement a city safety solution, the 5G-DT system determines safe and ecological driving behavior, and the VRU-Safe experiment concerns the safety of VRUs and the prevention of accidents.	[43]
5G V2X testbed for cooperative automated driving	No	No	No	Type 1 use cases exhibit predictable behavior and are limited to three phases: planning, initialization, and execution; Type 2 usage scenarios are based on emergency braking and cooperative emergency maneuvering.	[44]

Table 3. Surveyed testbed use cases—5G-MEC-V2X.

Project/Title of the Testbed	eMBB	URLLC	mMTC	Short Description of Case Study	References
5GVINNI moving experimentation facility	Yes	No	Yes	A Rapid Response Vehicle (RRV) is typically used for government and Public Protection and Disaster Relief (PPDR) use cases. 5GVINNI integrates the SATis5 testbed to present the usage scenario such as a connected vehicle.	[70]
Smart highway V2X testbed	No	Yes	No	The smart highway V2X testbed tests and validates ultra-fast connectivity between V2V and V2I communications. The testbed supports the data-intensive applications in the vehicle and at the RSU using MEC support.	[49]
5GCarmen	Yes	No	Yes	5GCarmen use cases include cooperative maneuvering, situation awareness, video streaming in the vehicle, and green driving.	[71]
Akraino connected vehicle	Yes	Yes	No	The Akraino connected vehicle proposed use cases include identifying and tracking bad cars, smart navigation, improving safe driving, and reducing traffic rule violations.	[56]
5GDRIVE	Yes	No	Yes	The 5GDRIVE testbed use cases focus on the V2I. UC1 involves the green light optimal speed advisory (GLOSA), in which the automated vehicle communicates with the infrastructure. UC2 involves intersection safety, which emulates a pedestrian crossing an intersection as a vehicle approaches the location.	[65]

4. Usage Complexity of the Testbeds

The complexity of using the 5G-MEC V2X testbeds depends on the level of complexity required to set up, configure, and conduct the experiments. Below, we discuss some common factors that can influence the complexity of using the testbed.

Hardware components: The complexity of using a testbed can be influenced by factors such as the complexity of its hardware components and the number of required features. For the 5G-MEC V2X testbed, the hardware components mainly consist of servers and computing units that are used to deploy the 5G-MEC and V2X application servers. In some cases, open source tools can be used to deploy the 5G-MEC system, and telecommunication companies can provide the necessary infrastructure.

Software components: The complexity of using software elements in a testbed is influenced by several factors, such as the development of the libraries, APIs, and SDKs used for deployment. These factors can significantly impact the usage of testbeds. For the 5G-MEC deployment, both open source and proprietary/licensed SDKs can be used. Additionally, simulators can be employed to implement V2X applications and scenarios.

Configuration: The complexity of using testbeds can also be affected by the amount of configuration needed to set up the hardware components, including servers and computing units, according to the requirements of the experiments, and the SDKs to ensure that the testbed works correctly. In the case of open source tools, the deployment's complexity can be influenced by the availability of user documentation and the user's expertise.

User interface: A user-friendly interface is essential for managing the resources of the 5G-MEC or 5G-MEC V2X open source testbed. Several testbeds are available and accessible for conducting research and running experiments.

User documentation: The project deliverable that provides the hardware infrastructure used for the required use cases and the GitHub README files used to set up the projects/required testbed repositories are examples of user documentation that can significantly impact the testbed replication or usage complexity.

User experience: The user experience is rated based on the complexity of all components and the technicalities of the testbed. In general, the 5G-MEC testbed also requires user expertise.

The usage complexity of a testbed can directly impact its accessibility and usability for researchers. Reducing the complexity of the testbed can increase the efficiency and effectiveness of further experiments/research.

The usage complexities of the available 5G-MEC, 5G-V2X, and 5G-MEC-V2X testbeds are shown in Table 4, Table 5, and Table 6, respectively, along with the aforementioned common factors.

Table 4. Main features of the 5G-MEC testbeds.

Project/Title of the Testbed	Hardware Components	Software Components	Configuration	User Interface	User Documentation	User Experience
5GCity	Hardware supports seven VMs with specifications including 2 vCPU, 4GB RAM, and 20GB VHDD. In addition, each use case has different VM requirements.	OpenStack, Docker, Kubernetes, OSM, i2CAT SDN RAN Controller, and 5GCity SDKs	The configuration required to set up the software components and the 5GCity SDKs is complex. Individual element deployment is possible but compiling and running experiments is difficult.	None	Project deliverable, GitHub repositories, installation documents, and release notes.	5GCity details the testbed and use case deployments but replicating the testbed is complex. The information for configuring the testbed is not provided in a particular iteration.
Mosaic5G	Computers: Gigabyte GB-BRi7-8550; SDRs Ettus: USRP B210, USRP B200-mini; Mobile phones: Google/Huawei Nexus 6P, Google Pixel 2.	JOX juju orchestrator, Low-Latency MEC (LL-MEC) platform, FlexRAN master controller, OAI-RAN, OAI-CN	The open source platforms and customized solutions are used to configure the Mosaic5G ecosystem. Subscribing to the Mosaic5G ecosystem is possible through GitLab.	None	GitLab repositories	The Mosaic5G ecosystem provides detailed information on the configuration of the open source platforms and the hardware requirements, making it easy to replicate the testbed for conducting experiments.
5GINFIRE University of Bristol testbed	Nokia RAN and CN, Corsa DP2100, Edgcore AS4610, Dell Power Edge T630 Compute Servers (1TB RAM and 100TB HDD Storage)	Nokia CN, OpenDaylight, Nokia and CloudBand MEC, OpenStack and Nokia VIM, OSM	The 5GINFIRE University of Bristol testbed does not provide configuration details.	5GINFIRE provides a wiki to access the testbed resources for conducting experiments using a portal. Researchers can obtain information from the sites of the available testbed providers and conduct necessary experimental analyses. However, access to the testbed resources is limited and the 5GINFIRE portal is no longer operational. The testbed operates under different policies and is part of the smart internet lab at the University of Bristol.	For 5GINFIRE, user documentation is provided on both GitHub and the 5GINFIRE wiki portal. Additionally, the project deliverable describes the use cases based on the testbed.	The 5GINFIRE University of Bristol testbed does not provide configuration and setup information. Some elements of the testbed are deployed using open source platforms, whereas others are proprietary. Replicating the testbed is challenging and accessing resources is also difficult.
EdgeX Foundry	VM based on x86 or ARM architecture	Docker	EdgeX is an open source platform developed by the Linux foundation. The EdgeX Foundry ecosystem wiki provides information on configuration and installation.	None	GitHub README files and GitHub repositories	The EdgeX ecosystem is an open source platform that offers an MEC solution for IoT and automation applications. It is built with Docker and used in various industrial applications.

Table 5. Main features of the 5G-V2X testbeds.

Project/Title of the Testbed	Hardware Components	Software Components	Configuration	User Interface	User Documentation	User Experience
5G-VINNI Munich experimentation facility	5G RAN (Huawei) 3.41 GHz, 5G Core (Huawei), MANO and NFVI (Huawei), SDN (Floodlight), Intel(R) Xeon(R) E5-2697 v2 @ 2.70GHz CPUs, 80GB RAM	Docker and Mininet	The configuration for operating the NFVI is limited to Docker and Mininet. Configuration of the 5G RAN and core is carried out by Huawei and the ETSI-specified MEC architecture is not implemented. Docker virtualization technologies are used to support the edge site.	None	Project deliverable	The 5GVINNI Munich experimentation facility testbed provides minimal information on the setup and experiment, making it difficult to replicate either.
5GINFIRE IT-Av automotive testbed	72 vCPUs, 704 GB of memory, 2 TB of storage, NIC Passthrough, DPDK and SR-IOV capabilities, 1x 24-port 1Gbps switch, interconnecting the infrastructure.	SUMO Simulator, OpenStack, OSM, OpenDaylight	The 5GINFIRE IT-Av automotive testbed does not provide configuration information. Along with utilizing open source platforms, the testbed provides details on the SUMO simulator scenario.	Same as the 5GINFIRE University of Bristol testbed; however, the IT-Av automotive testbed runs under the 5GASP project with similar objectives and is part of the Aveiro Tech City Living Lab (AT-CLL).	Same as the 5GINFIRE University of Bristol testbed	The 5GINFIRE IT-Av automotive testbed does not include setup and configuration details. However, the testbed utilizes open source platforms for experimentation, making replication possible.
5G V2X testbed for cooperative automated driving	NI/Ettus USRP X310 SDR with Intel-based PC	C/C++ with efficiency optimization based on Intel's x86 single instruction	None	None	None	The 5G V2X testbed for cooperative automated driving does not provide details on the software components and configuration.

Table 6. Main Features of the 5G-MEC-V2X testbeds.

Project/Title of the Testbed	Hardware Components	Software Components	Configuration	User Interface	User Documentation	User Experience
5G-VINNI moving experimentation facility	Lenovo™ ThinkCentre M920q Tiny Server Edge Node Fraunhofer FOKUS', SES's satellites, SES's RRV, Dell R630, –CISCO switch CATALYST 2960XR, MEC Server Super Server 5019A-FTN4	Open5GCore, Open Baton, OSM OpenStack, OpenNESS	Information regarding the hardware and software components is provided for the testbed setup. Although most software components are open source platforms, the configuration of the components for each experiment is not provided. For end-to-end (E2E) orchestration, Fraunhofer FOKUS provides a solution using Open Baton and OSM.	None	Project deliverable	The 5GVINNI moving experimentation facility was partially developed using open source platforms. However, detailed deployment information regarding the platforms regarding the experimental needs is not provided. Due to the lack of information, replicating the testbed is challenging.
Smart highway V2X testbed	General-purpose computing unit (GPCU) Intel Xeon Broadwell E5-2620V4 @2.1GHz (8-core) 32GB DDR4-2666 2Rx4 LP ECC	Docker, Kubernetes, Customized cMEAO	The smart highway V2X testbed is part of the Citylab testbed and offers comprehensive information on the software and hardware components.	Testbed resources are available instantly to conduct experiments. The FED4fire portal allows for registration and access to the resources, including the option to reserve them. Detailed information about accessing the testbed is available on the FED4fire portal.	The user manual is divided into various sections with detailed instructions. The jFed tool was created by FED4fire to access the testbed resources. By using the guidance provided by the Citylab testbed, the hardware components can be registered and reserved.	The Smart Highway V2X testbed is an easily accessible open source testbed, together with a thorough explanation of the procedures, that enables researchers to use the hardware and SDKs to conduct their experiments. The testbed allows for creating and deploying scenarios, as well as access to all involved nodes, making it possible to repeat experiments or utilize testbed resources.
5G Carmen	Intel NUC7i7DNBE, x86 server	OAI, KVM hypervisor, Docker, Kubernetes, Customized MEC platform and LightEDGE, OSM, and LightMANO	The configuration of the testbed is not provided in detail, although the elements are deployed using open source platforms and customized solutions.	None	Project deliverable	The 5GCarmen testbed provides details about the testbed and use cases for conducting experiments, as described in the project deliverable that outlines the testbed connectivity and use cases. However, the testbed is currently unavailable and difficult to replicate.

Table 6. Cont.

Project/Title of the Testbed	Hardware Components	Software Components	Configuration	User Interface	User Documentation	User Experience
Akraino connected vehicle	4 Arm/X86 servers, MEC Platform (1 Server) + 1 App Server (1 Server) + 2 Simulators (2 Servers)	OpenStack, Kubernetes, Docker, DPDK, OpenNESS, OVS	The Akraino connected vehicle provides a comprehensive blueprint of the testbed architecture and offers detailed documentation to access the testbed resources. This makes it easier for researchers to set up and replicate experiments in the connected-vehicle domain.	The Akraino connected vehicle provides access to servers using SSH. To run an experiment, the required details are provided using Gerrit.	Gerrit Documentation, Akraino wiki	The Akraino connected vehicle provides a list of the hardware and software elements in the blueprint but does not offer support for the setup and configuration of the testbed. The testbed resources can be accessed through a portal but replication may be challenging.
5G-DRIVE	Intel® Core™ i7-8700 CPU@ 3.2 GHz 16 Gb RAM 256 Gb HDD Intel Xeon Silver 42,162.1 GHz, 16C/32T, 9.6 GT/s32 GB RDIMM, 1 TB	OAI and Amarisoft, OpenStack, OSM, magma EPC	The setup and configuration of the 5GDRIVE testbed are not disclosed.	None	Project deliverable	The configuration information is unavailable, the 5GDRIVE testbed cannot be replicated, and the MEC deployment is inadequately explained. The testbed is not publicly accessible or open source.

5. Summary of Available Tools

The implementation of a 5G-MEC testbed needs SDKs and open source tools and has specific hardware requirements. Hardware components are crucial due to the computing power required by SDKs and applications. This section broadly discusses the open source tools used to implement the 5G-MEC system. In the context of the V2X applications, the available simulators are mentioned. The open source tools and SDKs are classified based on their role in the 5G-MEC V2X testbed.

5.1. 5G and 4G Systems

Available open source tools can help to implement and deploy the 5G and 4G core functionalities. However, most analyzed testbeds used commercial infrastructure, where the product included the SDKs. Commercial infrastructure SDKs are proprietary and licensed. Some open source tools and SDKs are still in development. Here, we present the most commonly used open source solutions and tools in the analyzed testbeds.

OAI [35] provides the RAN and CN implementation that can be used to deploy the 4G and, partially, 5G functionalities. In the future, OAI will be able to deploy full standalone 3GPP-compliant 5G CN. The OAI CN implementation has been tested with commercial RAN solutions such as Amarisoft gNB. Furthermore, in order to avoid the transmission of radio signals over the licensed spectrum, OAI can be connected to the open source state-of-the-art 5G UE and RAN (gNodeB) simulator, UERANSIM [72]. OAI CN supports primary functionalities such as UE registration, service requests, and PDU session management. OAI can support multiple UE deployments.

Open5GCore [73] was developed by the Fraunhofer FOKUS to provide 3GPP Release 15 and 16 core network functionalities. Open5GCore runs on typical hardware components such as a Raspberry Pi and computing servers using containers, pods, or virtualized environments. Open5GCore is licensed.

MagmaEPC [74] was developed within the Linux foundation projects. MagmaEPC is an open source tool that supports extendable mobile core network functionalities. It can be deployed both by using the Docker and on bare metal. MagmaEPC has three significant elements: an access gateway, an orchestrator, and a federation gateway.

NextEPC [75] is an open source tool for implementing the 4G/5G CN and is not fully compliant with 3GPP. NextEPC includes CN functionalities such as a Mobility Management Entity (MME), Home Subscriber Server (HSS), Serving Gateway (SGW), Packet Data Network Gateway (PGW), and Policy and Charging Rules Functions (PCRF).

Open5GS [76] is an open source implementation for 4G, 5G NSA CN, and 5G SA CN. Open5GS can be deployed using Docker and virtual machines. This CN implementation can be integrated with the open source RAN simulator UERANSIM. Using UERANSIM, multiple UEs can be deployed and supported by the Open5GS CN.

Free5GC [77] was mainly developed by National Chiao Tung University (NCTU). This implementation was based on the NextEPC platform, and the current version only supports the basic 5G CN, although it is planned to develop 5G CN functionalities defined in 3GPP Release 15 and beyond. Free5GC is deployed using virtual machines.

Simu5G [78] is an open source simulator for 5G new radio (NR) that is based on SimuLTE libraries. Simu5G was developed using the OMNET++ simulation framework. Simu5G's features can allow testing resource allocation and management in a 5G network in addition to the targeted UE, depending on the modulation schemes. Simu5G has also been tested with the ETSI MEC system model.

5.2. Networking

In the 5G-MEC system, networking is crucial for connecting the UE to the MEHs and the MEC applications. Additionally, networking within the 5G CN to the MEC system needs to be established. The MEC orchestrator or MANO can have control over the 5G-MEC system via networking.

The **Data Plane Development Kit (DPDK)** [79] was developed by the Linux foundation. The DPDK includes a collection of libraries for performing efficient packet tracing. The DPDK performs microservices that support the Container Network Interface (CNI) and is managed using a standard SDN controller.

Mininet [80] is an open source network emulator used to create virtual networks that run on a kernel, switch, and application code. Mininet provides support for developing applications based on OpenFlow. Mininet hosts run standard Linux software and can provide efficient routing along with SDN.

Open vSwitch (OVS) [81] is an open source tool for a multi-layer virtual switch that can redirect networking traffic within virtual machines. Furthermore, OVS operates as a soft switch running in the hypervisor. The Akraino connected vehicle implements OVS with OpenNESS to control the network traffic between containerized applications.

5.3. SDN

The ETSI refers to virtualization technologies such as SDN as key enablers for network slicing applications and recommends considering them within its MEC architecture. SDN differentiates the control plane and user plane functionalities. An SDN controller increases network traffic efficiency by providing management and a global view of the network topology. A survey of SDN-based MEC solutions for network slicing and V2X URLLC applications is presented in [82], and Zhu et al. [83] investigated SDN controllers and classified their work based on application needs.

OpenDaylight [84] is one of the most popular open source SDN controllers developed by the Linux foundation. The OpenDaylight SDN controller supports the OpenFlow southbound interface protocol. To test this controller, a switch is required or an alternative solution such as Mininet can be used. OpenDaylight is based on a Java model controller that employs YANG as its modeling language. The configuration and usage of OpenDaylight are thoroughly documented.

Open Network Operating System (ONOS) [85] is an open source tool developed using Bazel of Google. ONOS supports different southbound protocols of which OpenFlow is the primary one. An ONOS SDN controller can be hosted on a VM and integrated with Mininet. ONOS provides comprehensive documentation and prerequisites for its usage.

Secci et al. present a performance comparison of OpenDaylight and ONOS SDN controllers [86].

5.4. MEP

The ETSI architecture can be customized to build an MEC system in which the MEP plays a pivotal role in delivering MEC applications hosted on an MEH. The MEP provides the necessary MEC services for MEC applications. The MEC ecosystem comprises a list of ETSI-compliant MEC development tools, many of which are still in development. The implementations are classified based on the ETSI MEC-compliant architecture. The MEH encompasses the MEP and MEC applications running on the VI.

The **5GCity SDK** [32] MEC system builds a customized MEP on virtualized infrastructure compliant with the ETSI architecture. To deploy the 5GCity MEC system, VMs are required, and the specifications are listed in Table 4. The 5GCity MEP facilitates MEC application notifications and enables the creation, deletion, and discovery of new services required by the MEC system. Furthermore, the 5GCity MEP manages the DNS platform and DNS rules within the MEH. The 5GCity MEP can also communicate with third-party applications and manage the life cycle of the MEC applications. The 5GCity MEC system implements the Mp1 and Mm5 interfaces in a proprietary manner.

The Low-Latency MEC (**LL-MEC**) [87] platform is partially compliant with the ETSI MEC architecture. The LL-MEP aligns the Mp1 interface with the MEC application and the Mp2 interface with the virtual infrastructure to support low-latency MEC applications. The Mp1 interface uses a REST API to enable MEC services and the Mp2 interface focuses on DNS rules and traffic routing. The LL-MEP also allows for the separation of the user

plane functions from the control plane functions using SDN principles and the OpenFlow protocol. Additionally, LL-MEP includes the RNIS MEC service standardized by the ETSI for deriving RAN information. LL-MEP is mainly implemented using C++ on an x86 Linux system.

The **EdgeX foundry** [38] MEP is an industrialized solution developed by the Linux foundation. EdgeX is an open source edge platform that is non-compliant with the ETSI architecture. It uses Docker for containerized MEC applications, and to manage these applications, EdgeX uses docker-compose running on an x86 Linux system. EdgeX supports virtual device services that simulate user devices, allowing for device and data management using the MQTT broker.

OpenNESS [88] is an Intel smart-edge open source MEC SDK that is compliant with the ETSI architecture. It has an MEP and an MEP manager that allow for building, deploying, and managing MEC applications. OpenNESS includes different SDKs depending on the requirement, such as on-premises, access edge, and near-the-edge experience kits. It provides support for high-performance data planes and the management of virtual infrastructure. However, OpenNESS has two dissemination kits: one is open source and the other is an Intel-licensed distribution. OpenNESS has Intel-based hardware requirements to optimize the solution and run the experiments. In addition, it provides an interface and the possibility to integrate with the orchestration VIM tools such as OSM, OpenBaton, OpenStack, and K8s. OpenNESS allows for exploring resource allocation and management using edge multi-cluster orchestration (EMCO). It has an easy way of handling new application onboarding [89].

LightEDGE [90] is an open source lightweight MEP solution that is compliant with the ETSI architecture. It includes an MEP manager that is non-compliant with the ETSI architecture. LightEDGE utilizes a microservice architecture and operates on a K8s cluster. The LightEDGE MEP includes several MEC services specified by the ETSI, including the RNIS service. It also includes a UPF microservice, and 5G-EmPOWER is used to incorporate the RNIS services to extract RAN information. LightEDGE [91] supports 4G and 5G environments and can integrate with the Open5GS and srsRAN solutions. LightEDGE was developed to integrate with OSM to ensure seamless orchestration.

AdvantEDGE [92] is a mobile edge emulation platform (MEEP) that was developed by Inter-digital. AdvantEDGE is compliant with the ETSI architecture and runs on Docker and K8s. It allows the sandbox creation to deploy the 5G-MEC scenario. AdvantEDGE provides ETSI-defined MEC services such as location services, RNIS, WAIS, application enablement, application mobility, and V2X services. It also includes AdvantEDGE-specific edge services and network models to run the experiments. Furthermore, this platform allows the network traffic between elements to be controlled.

5.5. VI and VIM

The VI runs the MEP and applications by providing computation, memory, and storage capabilities. The VI enhances the MEC system deployment, and as a result, all open source tools use virtualization. The VI manager is responsible for allocating, managing, and releasing the virtualized storage, memory, and computation based on real-time needs.

Docker [93] is a lightweight container-based virtualization tool that is mainly used for hosting MEC applications. However, some MEC tools use Docker to deploy and run an MEP. Docker accelerates MEC application deployment and management with the help of Kubernetes (K8s). Early versions of the K8s cluster were based on Docker.

KVM hypervisor [94] is a kernel-based virtual machine solution for Linux systems that runs on top x86 hardware. The KVM hypervisor allows the kernel to function as a hypervisor. The 5GCarmen testbed uses a KVM hypervisor to host the MEH and run the MEC applications.

LXC [95] is an acronym for Linux container. The LXC runs and hosts a single Linux kernel system. LXC has been experimented with and tested for having a low-level container runtime.

Kubernetes (K8s) [96] is an open source tool used to deploy, scale, and manage containerized applications. K8s helps to automate processes by creating a cluster to host multiple MEHs and allowing the initiation of containerized applications. It performs the role of a manager (VIM) for the container-based solution. K8s is used by many MEP tools and in the 5G-MEC testbed. Some of the 5G-MEC testbed solutions extend K8s usage for orchestration.

OpenStack [97] offers an IaaS cloud infrastructure solution to deploy and manage a VM. It is mainly used for larger-scale deployment models and controls large computing, storage, and networking resources. OpenStack allows the configuration, creation, and deployment of VMs based on specific requirements. However, in some cases, OpenStack is used to manage containers. Many MEC solutions use K8s rather than OpenStack. The ETSI refers to OpenStack for the VIM in the MEC system.

5.6. MEO and MANO

The MEO is responsible for various functionalities in the MEC system such as tracking the overall view of the system depending on the MEH deployment of computing, networking, and storage resources and MEC services. The MEO needs to ensure appropriate MEH selection and MEC application relocation based on the user requirements. It also handles the onboarding of applications, packages, and SDKs. The ETSI specifies the deployment model of the MEC system in an NFV environment in which the MEC application and the NFV VNFs use the same VI. To fulfill the management and orchestration functionalities of the MEC system, the ETSI defined the NFV MANO architecture. Most of the surveyed testbeds deploy MANO functionalities using open source tools and platforms.

OSM [31] is an open source tool developed by the ETSI community. It delivers the MANO stack for an NFV and is capable of handling the VNFs independent of the VIM. However, OSM is compatible with open source VIM such as OpenStack. OSM has minimum hardware resource requirement specifications, and can deploy MEC applications as VNFs in the NFV environment. In a paper by Pablo Fondo-Ferreiro et al., OSM was used to deploy the MEC orchestrator functionalities [98].

JOX [99] is a Juju-based service orchestrator that can interact with CN functionalities using the orchestrator plugins. JOX supports the orchestration of the virtualized network to deploy the network slices. JOX is compatible with the ETSI MANO NFV environment and allocates network slices based on the VNF and resource configuration specifications. JOX runs on juju VNF and provides a plugin to the LL-MEP and FlexRAN SD-RAN controller.

OpenBaton [100] is an open source platform providing the ETSI MANO functionalities. OpenBaton VNF packages are adaptable with Juju VNF or generic VNF. Generic VIMs can be integrated with the system via Juju VNF. OpenBaton provides an NFV orchestration framework and manages MEC applications as VNFs. OpenBaton is widely used for the deployment of network slice applications.

cMEO [52] is a centralized MEC application orchestrator in the smart highway V2X testbed. cMEO is compatible with the K8s VIM and its API. cMEO allows the resources at the network's edge to be allocated by applying a smart mechanism. The smart mechanism that carries out the application placement is influenced by factors such as the MEH availability, RSUs, latency requirements, and the user's geolocation. In addition, cMEO supports the exchange of messages between the vehicles and the edge.

LightMANO [101] is a lightweight open source orchestrator platform that is compatible with the ETSI MANO architecture. LightMANO is built using Docker and runs within the K8s environment. The LightMANO framework orchestrates the NFV services in a distributed system. LightMEP [102] adapts LightMANO as the orchestrator for its MEC system.

5.7. V2X

V2X applications can benefit from the 5G-MEC testbeds and solutions to achieve URLLC. However, V2X application deployment and integration with the MEC system still need to be completed. The ETSI defines V2X application services and its MEC deployment scenarios. The V2X infrastructure is too complex to run real-time testbed experiments. However, some works have achieved V2I communication, and others have proposed simulation tools to test and run the experiments based on the V2X application needs.

Simulation of urban mobility (SUMO) [103] is an open source tool used to create and experiment with simulated traffic scenarios. SUMO presents real-world graphs of road networks, where each vehicle has a unique identifier and includes information such as the departure time and route. SUMO [104] includes a time-discrete solution, where the default step length is 1 sec but it can be reduced to 1 ms. Traffic and communication factors are essential to obtain and run the V2X simulated scenarios, and SUMO needs to be coupled with ns3. ns3 is an open source simulator for external communication [105].

6. Discussion and Conclusions

5G-MEC solutions are proving to be fundamental to delivering high-latency applications with strict requirements and improving the user experience. MEC technology is a value-added service for MNOs and user application developers and MEC solutions are increasing their commercial importance. However, there are research and practical development gaps between the ETSI-defined deployment model and the end solutions. The commercial or community solutions for the MEC system building blocks are incomplete and some critical issues remain:

1. How can open source 5G-MEC tools/platforms be integrated to create 5G-MEC testbeds or run experiments?
2. Are there any platforms that can be used to run 5G-MEC V2X simulations/emulations?
3. Are the different MEC building block tools compatible with each other?
4. What APIs are available and how can interfaces be developed between MEC elements?
5. How can existing open source 5G-MEC testbeds be used to experiment with V2X applications?

In this paper, we examined existing 5G-MEC V2X testbeds and categorized them. Most 5G-MEC testbeds are built using open source platforms and tools with specific hardware requirements. However, communities and commercial solutions have been developed based on the various 5G and MEC building blocks, as described in Section 5. Open source communities are actively working to offer support and develop API interfaces for integrating MEC elements. Standard VIMs can be integrated with most open source MANO tools; however, MANO tools lack interface development for MEC platforms and MEC platform managers.

To facilitate the development and testing of 5G-MEC scenarios, the ETSI created an MEC sandbox solution that enables user application developers to experiment with the existing 5G-MEC environment. This sandbox solution simulates the 5G-MEC environment and allows developers to gain a better understanding of the ETSI-defined MEC services.

The MEC system is divided into building blocks according to their functionalities. Scientific communities are working on each building block separately, which may shed light on issues such as compatibility. The ETSI defined the interfaces between the building blocks of the MEC system and presented their definitions and the REST API model. However, the development of the interfaces by the communities is still ongoing and needs to be completed.

In this paper, we examined and presented the existing open source 5G-MEC testbeds, some of which can be replicated and some of which can be used for running experiments. For replicating a 5G-MEC testbed for V2X application integration, one can use a SUMO simulator or access a smart highway V2X testbed.

The architectural approaches presented in [106–108] include programmable edge-to-cloud virtualization, cloud-to-edge meta-operating systems, and ML-assisted applications, which aim to minimize the amount of data transferred between the cloud–edge–IoT continuum. This is crucial for developing efficient and scalable systems. However, the surveyed testbeds face several challenges in supporting and testing future data-intensive applications and IoT-based systems that run at the far edge. These testbeds offer limited storage capacities and finite processing capabilities for end-users and cannot easily integrate novel 5G-V2X wireless access solutions, such as NR-V2X, which are able to satisfy the requirements of automotive applications. In general, the 5G-MEC and 5G-MEC-V2X testbeds can provide a flexible platform and integration to potentially enhance the solution, which involves complexity and compatibility issues due to their adaptability and the need for more standardization. Finally, these testbeds do not natively integrate AI/ML, which are now mature enough to provide efficient solutions for complex optimization and prediction problems necessary to improve the orchestration and network automation functions of beyond-5G systems.

Author Contributions: Conceptualization, P.V.W., R.G.G., and G.N.; investigation, P.V.W.; resources, P.V.W.; writing—original draft preparation, P.V.W.; writing—review and editing, R.G.G. and G.N.; supervision, R.G.G. and G.N.; funding acquisition, G.N. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the Norwegian Research Council through the 5G-MODaNel project (no. 308909) and the Italian Ministry of Education and Research (MIUR) in the framework of the FoReLab project (Departments of Excellence).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

ETSI	European Telecommunication Standards Institute
MEC	Multi-Access Edge Computing
5G	Fifth-Generation
URLLC	Ultra-Reliable Low-Latency Communication
V2X	Vehicle-to-Everything
SDN	Software-Defined Network
NFV	Network Function Virtualization
MEO	MEC Orchestrator
MEH	MEC Host
VNF	Virtual Network Function
VIM	Virtualization Infrastructure Manager
MEPM	MEC Platform Manager
MEP	MEC Platform
VI	Virtualization Infrastructure
AN	Access Network
CN	Core Network
eMBB	Enhanced Mobile Broadband
mMTC	Massive Machine-Type Communication
UPF	User Plane Function
RAN	Radio Access Network
OSS	Operating Support System
LCM	Life Cycle Management

MNO	Mobile Network Operator
NS	Network Service
NFVO	NFV Orchestrator
V2I	Vehicle-to-Infrastructure
V2P	Vehicle-to-Pedestrian
V2N	Vehicle-to-Network
V2V	Vehicle-to-Vehicle
AP	Access Point
NUC	Next Unit of Computing
VUE	Vehicle User Equipment
RSU	Road-Side Unit
OBU	On-Board Unit
SDKs	Software Development Kits
VM	Virtual Machines
K8s	Kubernetes
OAI	Open-Air Interface
OSM	Open Source MANO
MANO	Management and Orchestration

References

1. Witkowski, D. *Bridging the Gap—21st Century Wireless Telecommunications Handbook*, 2nd ed.; Joint Venture Silicon Valley Report; Joint Venture Silicon Valley: San Jose, CA, USA, 2019; ISBN 978-1675085080.
2. Zhu, G.; Liu, D.; Du, Y.; You, C.; Zhang, J.; Huang, K. Toward an Intelligent Edge: Wireless Communication Meets Machine Learning. *IEEE Commun. Mag.* **2020**, *58*, 19–25. [[CrossRef](#)]
3. Sabella, D. Toward Fully Connected Vehicles: Edge Computing for Advanced Automotive Communications, Dec. 2017 [Online]. Available online: <https://5gaa.org/toward-fully-connected-vehicles-edge-computing-for-advanced-automotive-communications/> (accessed on 20 March 2023).
4. Americas, G. *Wireless Technology Evolution towards 5G*; 3GPP Release 13 and Release 15 to Beyond; 5GAmericas: Bellevue, WA, USA, 2017.
5. ETSI. GS MEC 030 V2.2.1: Multi-Access Edge Computing (MEC); V2X Information Service API, 2022. [Online]. Available online: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/030/02.02.01_60/ (accessed on 20 March 2023).
6. Cao, H.; Gangakhedkar, S.; Ali, A.R.; Gharba, M.; Eichinger, J. A testbed for experimenting 5G-V2X requiring ultra reliability and low-latency. In Proceedings of the WSA 2017; 21th International ITG Workshop on Smart Antennas, Berlin, Germany, 15–17 March 2017; pp. 1–4.
7. Abboud, K.; Omar, H.A.; Zhuang, W. Interworking of DSRC and cellular network technologies for V2X communications: A survey. *IEEE Trans. Veh. Technol.* **2016**, *65*, 9457–9470. [[CrossRef](#)]
8. Giust, F.; Sciancalepore, V.; Sabella, D.; Filippou, M.C.; Mangiante, S.; Featherstone, W.; Munaretto, D. Multi-access edge computing: The driver behind the wheel of 5G-connected cars. *IEEE Commun. Stand. Mag.* **2018**, *2*, 66–73. [[CrossRef](#)]
9. Pham, Q.V.; Fang, F.; Ha, V.N.; Piran, M.J.; Le, M.; Le, L.B.; Hwang, W.J.; Ding, Z. A survey of multi-access edge computing in 5G and beyond: Fundamentals, technology integration, and state-of-the-art. *IEEE Access* **2020**, *8*, 116974–117017. [[CrossRef](#)]
10. Hou, L.; Gregory, M.A.; Li, S. A Survey of Multi-Access Edge Computing and Vehicular Networking. *IEEE Access* **2022**, *10*, 123436–123451. [[CrossRef](#)]
11. Navarro-Ortiz, J.; Romero-Diaz, P.; Sendra, S.; Ameigeiras, P.; Ramos-Munoz, J.J.; Lopez-Soler, J.M. A survey on 5G usage scenarios and traffic models. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 905–929. [[CrossRef](#)]
12. Spinelli, F.; Mancuso, V. Toward enabled industrial verticals in 5G: A survey on MEC-based approaches to provisioning and flexibility. *IEEE Commun. Surv. Tutor.* **2020**, *23*, 596–630. [[CrossRef](#)]
13. Taleb, T.; Samdanis, K.; Mada, B.; Flinck, H.; Dutta, S.; Sabella, D. On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1657–1681. [[CrossRef](#)]
14. Esmaeily, A.; Kravetska, K. Small-scale 5g testbeds for network slicing deployment: A systematic review. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 6655216. [[CrossRef](#)]
15. 3GPP. 5G System Overview. Available online: <https://www.3gpp.org/technologies/5g-system-overview> (accessed on 20 March 2023).
16. Series, M. IMT Vision—Framework and overall objectives of the future development of IMT for 2020 and beyond. *Recomm. ITU* **2015**, *2083*. Available online: https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.2083-0-201509-1!!PDF-E.pdf (accessed on 20 March 2023).
17. Series, M. Minimum requirements related to technical performance for IMT-2020 radio interface (s). *Report* **2017**, *2410*. Available online: <https://www.itu.int/pub/R-REP-M.2410-2017> (accessed on 20 March 2023).

18. ETSI. TS 123 501 V16.6.0: 5G; System Architecture for the 5G System (5GS) (3GPP TS 23.501 version 16.6.0 Release 16), 2020. [Online]. Available online: https://www.etsi.org/deliver/etsi_ts/123500_123599/123501/16.06.00_60/ts_123501v160600p.pdf (accessed on 20 March 2023).
19. Storck, C.R.; Duarte-Figueiredo, F. A Survey of 5G Technology Evolution, Standards, and Infrastructure Associated with Vehicle-to-Everything Communications by Internet of Vehicles. *IEEE Access* **2020**, *8*, 117593–117614. [CrossRef]
20. ETSI. GS MEC 003 V2.2.1: Multi-Access Edge Computing (MEC); Framework and Reference Architecture, 2020. [Online]. Available online: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/02.02.01_60/gs_MEC003v020201p.pdf (accessed on 20 March 2023).
21. Filali, A.; Abouaomar, A.; Cherkaoui, S.; Kobbane, A.; Guizani, M. Multi-access edge computing: A survey. *IEEE Access* **2020**, *8*, 197017–197046. [CrossRef]
22. Kekki, S.; Featherstone, W.; Fang, Y.; Kuure, P.; Li, A.; Ranjan, A.; Purkayastha, D.; Jiangping, F.; Frydman, D.; Verin, G.; et al. MEC in 5G networks. *ETSI White Pap.* **2018**, *28*, 1–28.
23. Nencioni, G.; Garroppo, R.G.; Gonzalez, A.J.; Helvik, B.E.; Procissi, G. Orchestration and Control in Software-Defined 5G Networks: Research Challenges. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 6923867. [CrossRef]
24. Ersue, M. ETSI NFV management and orchestration—An overview. *Present. IETF* **2013**, *88*. [Online]. Available online: <https://www.ietf.org/proceedings/88/slides/slides-88-opsawg-6.pdf> (accessed on 20 March 2023).
25. Sabella, D. MEC in Virtualized Environments. In *Multi-Access Edge Computing: Software Development at the Network Edge*; Springer: Cham, Switzerland, 2021; pp. 159–199. [CrossRef]
26. Antevski, K.; Bernardos, C.J.; Cominardi, L.; de la Oliva, A.; Mourad, A. On the integration of NFV and MEC technologies: Architecture analysis and benefits for edge robotics. *Comput. Netw.* **2020**, *175*, 107274. [CrossRef]
27. ETSI. GR MEC 017 V1.1.1: Mobile Edge Computing (MEC); Deployment of Mobile Edge Computing in an NFV Environment, 2018. [Online]. Available online: https://www.etsi.org/deliver/etsi_gr/mec/001_099/017/01.01.01_60/gr_mec017v010101p.pdf (accessed on 20 March 2023).
28. Chen, S.; Hu, J.; Shi, Y.; Peng, Y.; Fang, J.; Zhao, R.; Zhao, L. Vehicle-to-everything (V2X) services supported by LTE-based systems and 5G. *IEEE Commun. Stand. Mag.* **2017**, *1*, 70–76. [CrossRef]
29. Aissioui, A.; Ksentini, A.; Gueroui, A.M.; Taleb, T. On enabling 5G automotive systems using follow me edge-cloud concept. *IEEE Trans. Veh. Technol.* **2018**, *67*, 5302–5316. [CrossRef]
30. Colman-Meixner, C.; Khalili, H.; Antoniou, K.; Siddiqui, M.S.; Papageorgiou, A.; Albanese, A.; Cruschelli, P.; Carozzo, G.; Vignaroli, L.; Ulisses, A.; et al. Deploying a novel 5G-enabled architecture on city infrastructure for ultra-high definition and immersive media production and broadcasting. *IEEE Trans. Broadcast.* **2019**, *65*, 392–403. [CrossRef]
31. ETSI OSM. Open Source MANO (OSM). Available online: <https://osm.etsi.org/> (accessed on 20 March 2023).
32. 5GCity. SDK. Available online: <https://github.com/5GCity/5GCity-SDK> (accessed on 20 March 2023).
33. 5GCity. 5GCity Neural Hosting Platform-Installation Instructions. Available online: <https://www.5gcity.eu/installation-instructions/> (accessed on 20 March 2023).
34. Nikaein, N.; Chang, C.Y.; Alexandris, K. Mosaic5G: Agile and flexible service platforms for 5G research. *ACM SIGCOMM Comput. Commun. Rev.* **2018**, *48*, 29–34. [CrossRef]
35. Nikaein, N.; Marina, M.K.; Manickam, S.; Dawson, A.; Knopp, R.; Bonnet, C. OpenAirInterface: A flexible platform for 5G research. *ACM SIGCOMM Comput. Commun. Rev.* **2014**, *44*, 33–38. [CrossRef]
36. Foukas, X.; Nikaein, N.; Kassem, M.M.; Marina, M.K.; Kontovasilis, K. FlexRAN: A flexible and programmable platform for software-defined radio access networks. In *Proceedings of the 12th International Conference on Emerging Networking Experiments and Technologies*, Irvine, CA, USA, 12–15 December 2016; pp. 427–441.
37. 5GINFIRE. 5GINFIRE-University of Bristol. Available online: <https://5ginfire.eu/wp-content/uploads/2018/10/university-of-bristol-testbed-short-description-5ginfire.pdf?x18504> (accessed on 20 March 2023).
38. EdgeX. EdgeX Foundry. Available online: <https://www.edgexfoundry.org/get-started/> (accessed on 20 March 2023).
39. EdgeX. EdgeX Foundry Use Cases. Available online: <https://www.edgexfoundry.org/why-edgex/> (accessed on 20 March 2023).
40. 5GVINNI. 5GVINNI Munich Experimentation Facility Site. Available online: <https://www.5g-vinni.eu/munich-experimentation-facility-site/> (accessed on 20 March 2023).
41. 5GVINNI. 5G-VINNI Solution Facility Sites High Level Design (HLD)—v1 Munich Facility Site Annex. Available online: https://www.5g-vinni.eu/wp-content/uploads/2019/02/5g-vinni_d2.1_annex_a7_munich.pdf (accessed on 20 March 2023).
42. Alex Mavromatis, UNIVIBRIS Deliverable D5.2—Enabling 5G Automotive Vertical for Experimentation and Other EVIs 2020 pp. 104 + 62 (Annex). Available online: <https://ec.europa.eu/research/participants/documents/downloadPublic?documentId=080166e5ccda9f5&appId=PPGMS> (accessed on 20 March 2023).
43. Barmponakis, S.; Tsiatsios, G.; Papadakis, M.; Mitsianis, E.; Koursiompas, N.; Alonistioti, N. Collision avoidance in 5G using MEC and NFV: The vulnerable road user safety use case. *Comput. Netw.* **2020**, *172*, 107150. [CrossRef]
44. Cao, H.; Gangakhedkar, S.; Ali, A.R.; Gharba, M.; Eichinger, J. A 5G V2X testbed for cooperative automated driving. In *Proceedings of the 2016 IEEE Vehicular Networking Conference (VNC)*, Columbus, OH, USA, 8–10 December 2016; pp. 1–4. [CrossRef]

45. Gharba, M.; Cao, H.; Gangakhedkar, S.; Eichinger, J.; Ali, A.R.; Ganesan, K.; Jain, V.; Lapoehn, S.; Frankiewicz, T.; Hesse, T.; et al. 5G enabled cooperative collision avoidance: System design and field test. In Proceedings of the 2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), Macau, China, 12–15 June 2017; pp. 1–6.
46. Jan Pitter, Ericsson Deliverable D2.1 5G-VINNI Solution Facility-Sites High Level Design (HLD) Report (R) March 2019; p. 104. Available online: <https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5c2be8c35&appId=PPGMS> (accessed on 20 March 2023).
47. Corici, M.; Liolis, K.; Burkhardt, F.; Gheorghe-Pop, I.; Covaci, S.; Politis, C.; Geurtz, A.; Koernicke, J.; Völk, F.; Kapovits, A. SATis5: A 5G Testbed Integrating Satellite and Terrestrial Infrastructures. In Proceedings of the Advanced Satellite Multimedia Systems Conference (ASMS), Berlin, Germany, 10–12 September 2018.
48. Liolis, K.; Cahill, J.; Higgins, E.; Corici, M.; Troudt, E.; Sutton, P. Over-the-air demonstration of satellite integration with 5G core network and multi-access edge computing use case. In Proceedings of the 2019 IEEE 2nd 5G World Forum (5GWF), Dresden, Germany, 30 September–2 October 2019; pp. 1–5.
49. IDLab, U.o.A. Smart Highway. Available online: <https://www.uantwerpen.be/en/research-groups/idlab/infrastructure/smart-highway/> (accessed on 20 March 2023).
50. FED4FIRE. Citylab. Available online: <https://www.fed4fire.eu/testbeds/citylab/> (accessed on 20 March 2023).
51. Marquez-Barja, J.; Lannoo, B.; Braem, B.; Donato, C.; Maglogiannis, V.; Mercelis, S.; Berkvens, R.; Hellinckx, P.; Weyn, M.; Moerman, I.; et al. Smart Highway: ITS-G5 and C-V2X based testbed for vehicular communications in real environments enhanced by edge/cloud technologies. In Proceedings of the 2019 European Conference on Networks and Communications, Valencia, Spain, 18–21 June 2019.
52. Slamnik-Kriještorac, N.; Marquez-Barja, J.M. Unraveling Edge-based in-vehicle infotainment using the Smart Highway testbed. In Proceedings of the 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 9–11 January 2021; pp. 1–4.
53. The Linux Foundation. Prometheus. Available online: <https://prometheus.io/> (accessed on 20 March 2023).
54. 5GCarmen. 5G-CARMEN Connected and Automated Road Mobility in the European Union. Available online: <https://5g-ppp.eu/wp-content/uploads/2020/06/5G-Carmen.pdf> (accessed on 20 March 2023).
55. Slamnik-Kriještorac, N.; Yilma, G.M.; Liebsch, M.; Yousaf, F.Z.; Marquez-Barja, J. Collaborative orchestration of multi-domain edges from a Connected, Cooperative and Automated Mobility (CCAM) perspective. *IEEE Trans. Mob. Comput.* **2021**, *22*, 2001–2020. [[CrossRef](#)]
56. Robertqiu. Connected Vehicle Blueprint(Aka CVB). 2021. Available online: <https://wiki.akraino.org/pages/viewpage.action?pageId=9601601> (accessed on 20 March 2023).
57. Liang, J.; Liu, F.; Li, S.; Cai, Z. A comparative research on open source edge computing systems. In Proceedings of the Artificial Intelligence and Security: 5th International Conference, ICAIS 2019, New York, NY, USA, 26–28 July 2019; pp. 157–170.
58. Kutila, M.; Kauvo, K.; Pyykönen, P.; Zhang, X.; Martinez, V.G.; Zheng, Y.; Xu, S. A C-V2X/5G Field Study for Supporting Automated Driving. In Proceedings of the 32nd IEEE Intelligent Vehicles Symposium, IV'21, Nagoya, Japan, 11–16 September 2021; pp. 315–320.
59. Chochliouros, I.P.; Spiliopoulou, A.S.; Kostopoulos, A.; Vasilaki, E.; Herzog, U.; Dardamanis, A.; Chen, T.; Ladid, L.; Agapiou, M. Testbeds for the Implementation of 5G in the European Union: The Innovative Case of the 5G-DRIVE Project. In Proceedings of the IFIP International Conference on Artificial Intelligence Applications and Innovations, Hersonissos, Greece, 24–26 May 2019; pp. 78–92.
60. Kutila, M.; Nykänen, L.; Lankinen, M. 5g-drive: Eu china c-v2x collaboration. In Proceedings of the 8th Transport Research Arena, TRA 2020-Conference Cancelled, Liikenne-ja viestintävirasto Traficom, Helsinki, Finland, 27–30 April 2020; p. 147.
61. Mosaic5G. Available online: <https://gitlab.eurecom.fr/mosaic5g> (accessed on 20 March 2023).
62. 5GINFIRE. Available online: <https://5ginfire.eu/it-av-automotive-testbed/> (accessed on 20 March 2023).
63. 5GVINNI. 5GVINNI Moving Experimentation Facility Site. Available online: <https://www.5g-vinni.eu/moving-experimentation-facility-site/> (accessed on 20 March 2023).
64. FED4FIRE. Smart Highway V2X Testbed. Available online: <https://www.fed4fire.eu/testbeds/smart-highway/> (accessed on 20 March 2023).
65. 5G-DRIVE. D4.1: V2X Development and Test Plan. Available online: <https://5g-drive.eu/resources-and-results/project-deliverables/#1552568196209-3ef4b649-a787> (accessed on 20 March 2023).
66. 5GCity. Use Cases-5GCity. Available online: <https://www.5gcity.eu/use-cases/> (accessed on 20 March 2023).
67. Mosaic5G. Use Cases-Mosaic5G. Available online: <https://mosaic5g.io/#portfolioModal1> (accessed on 20 March 2023).
68. 5GINFIRE. Smart Internet Lab, University of Bristol. Available online: <https://5ginfire.eu/wp-content/uploads/2018/05/bristol-testbed.pdf> (accessed on 20 March 2023).
69. 5GVINNI. Deliverable D3.1 Specification of Services Delivered by Each of the 5G-VINNI Facilities. Available online: <https://www.5g-vinni.eu/deliverables/> (accessed on 20 March 2023).
70. Politis, C.; Liolis, K.; Corici, M.; Troudt, E.; Szabó, Z.; Cahill, J. Design of moving experimentation facility to showcase satellite integration into 5G. In Proceedings of the 2019 European Conference on Networks and Communications (EuCNC), Valencia, Spain, 18–21 June 2019; pp. 177–181.
71. 5GCarmen. Use Cases-5GCarmen. Available online: <https://5gcarmen.eu/use-cases/> (accessed on 20 March 2023).

72. UERANSIM. Available online: <https://github.com/aligungr/UERANSIM> (accessed on 20 March 2023).
73. Open5GCore. Available online: <https://www.open5gcore.org/> (accessed on 20 March 2023).
74. MagmaEPC. Available online: <https://github.com/magma/magma> (accessed on 20 March 2023).
75. NextEPC. Available online: <https://github.com/nextepc/nextepc> (accessed on 20 March 2023).
76. Open5GS. Available online: <https://open5gs.org/open5gs/docs/guide/01-quickstart/> (accessed on 20 March 2023).
77. Free5GC. Available online: <https://github.com/free5gc/free5gc> (accessed on 20 March 2023).
78. Nardini, G.; Sabella, D.; Stea, G.; Thakkar, P.; Virdis, A. Simu5G—An OMNeT++ library for end-to-end performance evaluation of 5G networks. *IEEE Access* **2020**, *8*, 181176–181191. [[CrossRef](#)]
79. Pongrácz, G.; Molnár, L.; Kis, Z.L. Removing roadblocks from SDN: OpenFlow software switch performance on Intel DPDK. In Proceedings of the 2013 Second European Workshop on Software Defined Networks, Berlin, Germany, 10–11 October 2013; pp. 62–67.
80. De Oliveira, R.L.S.; Schweitzer, C.M.; Shinoda, A.A.; Prete, L.R. Using mininet for emulation and prototyping software-defined networks. In Proceedings of the 2014 IEEE Colombian Conference on Communications and Computing (COLCOM), Bogota, CO, USA, 4–6 June 2014; pp. 1–6.
81. Linux Foundation Collaborative Project. OpenvSwitch. Available online: <https://www.openvswitch.org/> (accessed on 20 March 2023).
82. Shah, S.D.A.; Gregory, M.A.; Li, S. Cloud-native network slicing using software defined networking based multi-access edge computing: A survey. *IEEE Access* **2021**, *9*, 10903–10924. [[CrossRef](#)]
83. Zhu, L.; Karim, M.M.; Sharif, K.; Xu, C.; Li, F.; Du, X.; Guizani, M. SDN controllers: A comprehensive analysis and performance evaluation study. *ACM Comput. Surv. (CSUR)* **2020**, *53*, 1–40. [[CrossRef](#)]
84. OpenDaylight Project. OpenDaylight. Available online: <https://docs.opendaylight.org/projects/controller/en/latest/dev-guide.html> (accessed on 20 March 2023).
85. Open Networking Foundation. Open Network Operating System (ONOS). Available online: <https://wiki.onosproject.org/display/ONOS/ONOS+%3A+An+Overview> (accessed on 20 March 2023).
86. Secci, S.; Diamanti, A.; Vilchez, J.M.S.; Bah, M.T.; Vizzarreta, P.; Machuca, C.M.; Scott-Hayward, S.; Smith, D. Security and Performance Comparison of ONOS and ODL Controllers. Informational Report, Open Networking Foundation. 2019. Available online: <https://hal.science/hal-03188550> (accessed on 20 March 2023).
87. Nikaein, N.; Vasilakos, X.; Huang, A. LL-MEC: Enabling Low Latency Edge Applications. In Proceedings of the 2018 IEEE 7th International Conference on Cloud Networking (CloudNet), Tokyo, Japan, 22–24 October 2018; pp. 1–7. [[CrossRef](#)]
88. Intel. OpenNESS Edge Applications. Available online: <https://github.com/smart-edge-open/edgeapps> (accessed on 20 March 2023).
89. Intel. OpenNESS. Available online: <https://github.com/smart-edge-open> (accessed on 20 March 2023).
90. Roberto Riggio LightEDGE. Available online: <https://lightedge.io/> (accessed on 20 March 2023).
91. Roberto Riggio. LightEDGE. Available online: <https://github.com/lightedge/lightedge.github.io/wiki> (accessed on 20 March 2023).
92. Michel, R.; Lallo, K.D.; Robert, G. AdvantEDGE: A Mobile Edge Emulation Platform (MEEP). Available online: <https://github.com/InterDigitalInc/AdvantEDGE> (accessed on 20 March 2023).
93. Docker. Available online: <https://www.docker.com> (accessed on 20 March 2023).
94. Red Hat OpenShift. KVM Hypervisor. Available online: https://www.linux-kvm.org/page/Main_Page (accessed on 20 March 2023).
95. LXC—Linux Containers. Available online: <https://github.com/lxc/lxc> (accessed on 20 March 2023).
96. The Kubernetes Authors. Kubernetes. Available online: <https://kubernetes.io> (accessed on 20 March 2023).
97. OpenInfra Foundation Project. OpenStack. Available online: <https://www.openstack.org/software/> (accessed on 20 March 2023).
98. Fondo-Ferreiro, P.; Estévez-Caldas, A.; Pérez-Vaz, R.; Gil-Castiñeira, F.; González-Castaño, F.J.; Rodríguez-García, S.; Sousa-Vázquez, X.R.; López, D.; Guerrero, C. Seamless multi-access edge computing application handover experiments. In Proceedings of the 2021 IEEE 22nd International Conference on High Performance Switching and Routing (HPSR), Paris, France, 7–10 June 2021; pp. 1–6.
99. Katsalis, K.; Nikaein, N.; Huang, A. JOX: An event-driven orchestrator for 5G network slicing. In Proceedings of the NOMS 2018—2018 IEEE/IFIP Network Operations and Management Symposium, Taipei, Taiwan, 23–27 April 2018; pp. 1–9. [[CrossRef](#)]
100. Open Baton. OpenBaton Documentation. Available online: <https://openbaton-docs.readthedocs.io/en/3.0.0/> (accessed on 20 March 2023).
101. LightMANO. The LightMANO Platform. Available online: <https://github.com/lightmano> (accessed on 20 March 2023).
102. Subramanya, T.; Baggio, G.; Riggio, R. lightMEC: A Vendor-Agnostic Platform for Multi-access Edge Computing. In Proceedings of the 2018 14th International Conference on Network and Service Management (CNSM), Rome, Italy, 5–9 November 2018; pp. 198–204.
103. SUMO. SUMO V2X Communication Research (Platooning and CIM). Available online: <https://github.com/sraddon/SUMO-V2X-Communication-Research-Platooning-and-CIM> (accessed on 20 March 2023).
104. German Aerospace Center (DLR) and others. SUMO User Documentation. Available online: <https://sumo.dlr.de/docs/index.html> (accessed on 20 March 2023).
105. Krajzewicz, D.; Erdmann, J.; Behrisch, M.; Bieker, L. Recent development and applications of SUMO—Simulation of Urban MObility. *Int. J. Adv. Syst. Meas.* **2012**, *5*, 128–138.

106. Trakadas, P.; Masip-Bruin, X.; Facca, F.M.; Spantideas, S.T.; Giannopoulos, A.E.; Kapsalis, N.C.; Martins, R.; Bosani, E.; Ramon, J.; Prats, R.G.; et al. A Reference Architecture for Cloud–Edge Meta-Operating Systems Enabling Cross-Domain, Data-Intensive, ML-Assisted Applications: Architectural Overview and Key Concepts. *Sensors* **2022**, *22*, 9003. [[CrossRef](#)]
107. Alvarez, F.; Breitgand, D.; Griffin, D.; Andriani, P.; Rizou, S.; Zioulis, N.; Moscatelli, F.; Serrano, J.; Keltsch, M.; Trakadas, P.; et al. An edge-to-cloud virtualized multimedia service platform for 5G networks. *IEEE Trans. Broadcast.* **2019**, *65*, 369–380. [[CrossRef](#)]
108. Rizou, S.; Athanasoulis, P.; Andriani, P.; Iadanza, F.; Trakadas, P.; Griffin, D.; Kheirkhah, M.; Breitgand, D.; Weit, A.; Ustok, R.F.; et al. Programmable edge-to-cloud virtualization for 5G media industry: The 5G-media approach. In Proceedings of the Artificial Intelligence Applications and Innovations. AIAI 2020 IFIP WG 12.5 International Workshops: MHDW 2020 and 5G-PINE 2020, Neos Marmaras, Greece, 5–7 June 2020; pp. 95–104.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.