This article has been published on Computer Networks. Please cite it as L. Bonati, M. Polese, S. D'Oro, S. Basagni, and T. Melodia, "Open, Programmable, and Virtualized 5G Networks: State-of-the-Art and the Road Ahead," Computer Networks, vol. 182, December 2020, doi: 10.1016/j.comnet.2020.107516.

Open, Programmable, and Virtualized 5G Networks: State-of-the-Art and the Road Ahead

Leonardo Bonati, Michele Polese, Salvatore D'Oro, Stefano Basagni, Tommaso Melodia

Institute for the Wireless Internet of Things, Northeastern University, Boston, MA 02115, USA

Email: {l.bonati, m.polese, s.doro, s.basagni, t.melodia}@northeastern.edu

***Abstract*—Fifth generation (5G) cellular networks will serve a wide variety of heterogeneous use cases, including mobile broadband users, ultra-low latency services and massively dense connectivity scenarios. The resulting diverse communication requirements will demand networking with unprecedented flexibility, not currently provided by the monolithic black-box approach of 4G cellular networks. The research community and an increasing number of standardization bodies and industry coalitions have recognized softwarization, virtualization, and disaggregation of networking functionalities as the key enablers of the needed shift to flexibility. Particularly, software-defined cellular networks are heralded as the prime technology to satisfy the new application-driven traffic requirements and to support the highly time-varying topology and interference dynamics, because of their *openness* through well-defined interfaces, and *programmability*, for swift and responsive network optimization. Leading the technological innovation in this direction, several 5G software-based projects and alliances have embraced the open source approach, making new libraries and frameworks available to the wireless community. This race to open source softwarization, however, has led to a deluge of solutions whose interoperability and interactions are often unclear. This article provides the first cohesive and exhaustive compendium of recent open source software and frameworks for 5G cellular networks, with a full stack and end-to-end perspective. We detail their capabilities and functionalities focusing on how their constituting elements fit the 5G ecosystem, and unravel the interactions among the surveyed solutions. Finally, we review hardware and testbeds on which these frameworks can run, and provide a critical perspective on the limitations of the state-of-the-art, as well as feasible directions toward fully open source, programmable 5G networks.**

***Index Terms*—Software-defined Networking, 5G, Open Source, Network Function Virtualization, O-RAN, ONAP.**

I. Introduction

The potential of 5th generation (5G) communications is being unleashed into the fabric of cellular networks, enabling unprecedented technological advancements in the networking hardware and software ecosystems [1]. Applications such as virtual reality, telesurgery, high-resolution video streaming, and private cellular networking—just to name a few—will be freed from the shadows of the spectrum crunch and resource-scarcity that have haunted 4th generation (4G) networks for years. By unbridling the sheer power of these applications, 5G will usher unparalleled business opportunities for infrastructure and service providers, and foster unrivaled cellular networking-based innovation [2].

This work was supported in part by the US National Science Foundation under Grant CNS-1618727 and in part by the US Office of Naval Research under Grants N00014-19-1-2409 and N00014-20-1-2132.

The journey to achieve the 5G vision, however, is still beset by many research and development challenges. Traditional cellular networks are characterized by an inflexible and monolithic infrastructure, incapable of meeting the heterogeneity and variability of 5G scenarios and the strict requirements of its applications [3]. Now more than ever, the limitations of the "black-box" approaches of current cellular deployments, where hardware and software are *plug-and-play* with little or no reconfiguration capabilities, are manifest. The lack of full control of the vast amount of available resources and network parameters makes it hard to adapt network operations to real-time traffic conditions and requirements, resulting in ineffective resource management, sub-optimal performance, and inability to implement Connectivity-as-a-Service (CaaS) technologies such as private cellular networking [4]. The inflexibility of current approaches is even more harmful in 5G scenarios, where densification and the need for directional communications call for fine-grained network control [5–7], resources are scarce and spectrum availability and energy consumption are strictly regulated [8].

Both industry and academia now agree that the practical realization of 5G systems needs a radical overhaul of all plug-and-play approaches in favor of new, agile and open paradigms for network deployment, control and management. In this context, revolutionary and innovative networking solutions based upon *programmability*, *openness*, *resource sharing* and *edgefication* are welcome to the cellular arena [9, 10]. New networking principles such as Software-defined Networking (SDN) [11], network virtualization [12], and Multi-access Edge Computing (MEC) [13] have demonstrated that dynamic network control and agile management (e.g., frequency planning, user scheduling, mobility management, among others) is possible. Similarly, the emergence of network slicing and cloud Radio Access Network (RAN) technologies have made it clear that infrastructure sharing not only maximizes resource utilization, but also opens new market opportunities (e.g., differentiated services, infrastructure leasing, CaaS), *thus representing a desirable solution for network operators and infrastructure providers alike* [14, 15].

Following the growing interest in softwarization and virtualization technologies, the 5G ecosystem has witnessed the exponential growth of dedicated solutions for 5G applications [16]. These solutions include software and hardware tailored to specific tasks [17] and full-fledged multitasking frameworks spanning the whole infrastructure [18]. Despite their diversity in structure and purpose, the majority of these solutions has two important aspects in common: They are *open source* and fully *programmable*. *These two aspects together are bringing unprecedented flexibility to 5G systems, making them accessible to a much broader community of research and developers.*

Just a few years ago, the majority of researchers had access to actual cellular networks. When they did, access v limited to individual network components or functionaliti Today, the software-defined paradigm as made popular by GNU Radio libraries [19], has been easily adopted by softw bundles such as OpenAirInterface (OAI) [17] and srsLTE [1 for swift instantiation of fully-functional cellular networks commercial Software-defined Radio (SDR) devices. Softw frameworks such as O-RAN [21, 22], which run on "wh box" servers, allow reconfiguration and optimization of r work and transceiver functionalities. These new software a hardware components have radically changed the way research community and the telecom industry plan, deploy, and interact with cellular systems. Prototyping, testing, and deploying new algorithms and protocols for cellular networks enjoys now unprecedented ease and time to market. The advantage of this revolutionary approach is twofold: (i) *Openness* allows researchers to evaluate and analyze their solutions on a real-world setup [23], and enables telecom operators to directly interact and control networking equipment [4]. Also, (ii) *programmability* fosters the design of novel and advanced algorithms that optimize network performance by efficiently and dynamically allocating network resources and controlling software and hardware functionalities, even in real time, if appropriate. For instance, telecom operators such as Rakuten are leveraging microservices to separate the user and control planes in their network deployments, thus endowing them with unprecedented flexibility [24]. Programs like Platforms for Advanced Wireless Research (PAWR) by the U.S. National Science Foundation [25], are bringing programmable wireless testing infrastructure at scale to broad communities of researchers—thus creating a fertile ground for software-based open innovation.

The race to the open source and programmable Holy Grail has generated a plethora of heterogeneous software and hardware components and frameworks, whose functionality, scope, and interoperability with other solutions are often obscure and hard to assess. *This article organizes the multiplicity of solutions into the appropriate building blocks of the open source and programmable 5G ecosystem. We detail how each components fits into a 5G network, highlight the interactions among solutions, and unfold their capabilities and functionalities, highlighting strengths and limitations. Our survey provides the first cohesive and exhaustive recount and taxonomy of open, programmable, and virtualized solutions for 5G networks.* As most frameworks and devices serve specific purposes in the 5G architectures, we also provide usage directives and how-to guidelines to combine different components into full-fledged open source 5G systems.

With respect to previous survey efforts [26, 27] we provide extensive details and commentary on the architecture of softwarized 5G networks, their building blocks, the software frameworks developed so far, and their interactions.

Figure 1 provides a visual guide to how the topics surveyed in our work relate to one another, as well as to the structure of the remainder of this article.

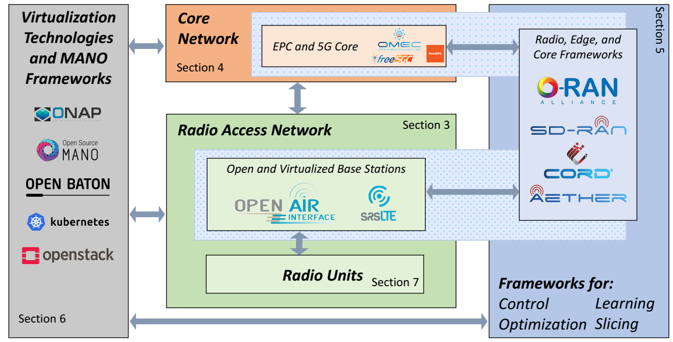

Fig. 1: The main building blocks of open source, programmable and virtualized 5G networks with their components and technologies.

Section II provides a bird's-eye view of the architecture of 5G systems, describing its components and technologies. Sections III and IV introduce and describe open source solutions for the RAN and Core Network (CN) portions of the infrastructure, respectively. General open source frameworks inclusive of both RAN and CN functionalities are discussed in Section V. Virtualization and management frameworks are provided in details in Section VI. Section VII describes software-defined hardware platforms for open source radio units, highlighting their features and their suitability for 5G applications. Section VIII presents a variety of experimental testbeds allowing instantiation of softwarized 5G networks and testing of new solutions. Finally, in Section IX we conclude this article by identifying limitations of the current 5G open source ecosystem and discuss the road ahead, with its unanswered research questions. A list of acronyms used throughout the article is provided in Appendix A.

II. Architectural Enablers of 5G Cellular Networks

Mobile networks are transitioning from monolithic architectures, based on dedicated "*black-box*" hardware with proprietary firmware and software, to disaggregated deployments based on open source software that runs on generic SDR or "agnostic" computing devices [28–30]. This trend is not new to cellular networking, as it has been part of the general discussion around 4G cellular networks. However, while software-based design represents a relatively recent evolution in the context of 4G networks, 5G specifications have foreseen the flexible deployment of agile, softwarized services already in their early stages, with their application to key infrastructure components such as the core, the RAN and the edge cloud [31]. This "flexibility-by-design" puts 5G networks in the privileged position to meet the requirements of heterogeneous traffic classes, mobility and advanced applications through design that is unified, open and dynamically changeable.

In this section we provide an overview of 4G and 5G cellular network architectures, as well as their main components and building blocks (Figure 1). We start by describing radio access and core network elements and general deployment paradigms. We then discuss the architectural and technology enablers such as Software-defined Networking (SDN), Network Function Virtualization (NFV), network slicing, MEC, and intelligent networks. Our aim is to provide a reference architecture to map the different open source software libraries and frameworks surveyed in this article to specific network functionalities.

A. Architecture of 4G and 5G Cellular Networks

Figure 2 provides a high-level overview of the 4G and 5G cellular architectures, along with some of the open source software frameworks envisioned as their components.

Cellular networks consist of a Radio Access Network (RAN) and a Core Network (CN). Even though this separation remains unaltered in 4G and 5G deployments, the actual implementation and configuration of these core components differ greatly. Particularly, they comply with the 3rd Generation Partnership Project (3GPP) Long Term Evolution (LTE) and NR[1] specifications for the RAN, and the Evolved Packet Core (EPC) and 5G Core (5GC) for the CN, respectively.[2] Figure 2 highlights the differences, in terms of flexibility, between the deployments of 4G (in the yellow boxes) and 5G (in the orange boxes) networks. For the CN, the 4G EPC has multiple components that have been traditionally executed on dedicated hardware, and only recently have transitioned to software-based deployments. The 5GC, instead, has been designed according to a service-based approach from the get-go. The EPC servers are split into multiple virtual network functions providing specific functionalities. They are connected to each other through open and standardized interfaces. A similar separation principle has been considered for the 5G RAN, now designed to provide a functional split among heterogeneous parts of the base stations (e.g., control, computing and radio units), with different layers of the protocol stack instantiated on different elements located in different parts of the network.

LTE and EPC: The LTE RAN is composed of evolved Node Bases (eNBs), i.e., the LTE base stations, which provide wireless connectivity to the mobile User Equipments (UEs). The eNBs are generally deployed as a single piece of equipment on dedicated hardware components, and are networked together and to the core network. LTE operates on a frame structure with 10 subframes of $1\ \mathrm{ms}$ per frame, and 12 to 14 OFDM symbols for each subframe. The maximum carrier bandwidth is $20\ \mathrm{MHz}$. Up to 5 carriers can be aggregated for a total of $100\ \mathrm{MHz}$ [34].

The LTE protocol stack for the user plane (also known as Evolved Universal Terrestrial Access Network (E-UTRAN), bottom right corner of the RAN box in Figure 2) consists of:

- The Packet Data Convergence Protocol (PDCP) layer, which implements security functionalities (e.g., ciphering of packets), performs header compression, and takes care of the end-to-end packet delivery between the eNB and the UE [35].
- The Radio Link Control (RLC) layer, which provides data link layer services (e.g., error correction, packet fragmentation and reconstruction). It supports three different configurations: The Transparent Mode (TM), to simply relay packets between the Medium Access Control (MAC) and PDCP layers; the Unacknowledged Mode (UM), for buffering, segmentation, concatenation and reordering, and the Acknowledged Mode (AM), for retransmitting packets via a ACK/NACK feedback loop [36].
- The MAC layer, which performs scheduling, interacts with RLC to signal transmissions, forwards the transport blocks to the physical layer, and performs retransmissions via Hybrid Automatic Repeat reQuest (HARQ) [37].
- The Physical (PHY) layer, which takes care of channel coding, modulates the signal, and performs transmissions in an OFDM-based frame structure [38].

These layers also perform control plane functionalities, which concern measurement collection and channel quality estimation. Additionally, the Radio Resource Control (RRC) layer manages the life cycle of the eNB to UE connection, and it is a point of contact with the core network for control functionalities.

The main components of the EPC (in the top right corner of Figure 2) are: (i) The Packet Gateway (PGW) and Service Gateway (SGW), which are packet gateways to and from the Internet; (ii) the Mobility Management Entity (MME), which handles handovers and the UE connection life cycle from the core network point of view, and (iii) the Home Subscription Server (HSS), which manages subscriptions and billing [39].

NR: The 3GPP NR RAN represents quite the evolution of the 4G LTE, especially in terms of protocol stack, functionalities and capabilities. First, it supports a wider range of carrier frequencies, which include part of the millimeter wave (mmWave) spectrum [40]. Second, the frame structure, while still OFDM-based, is more flexible, with a variable number of symbols per subframe, the option to use much larger bandwidths than LTE (up to $400\ \mathrm{MHz}$ per carrier), and the integration of signals and procedures to manage directional transmissions at mmWaves [41]. Third, the 5G RAN can be connected either to the 4G EPC (*non-standalone configuration*) or to the new 5GC (*standalone configuration*). Finally, the NR base stations (Next Generation Node Bases (gNBs)) allows distributed deployment, with different parts of the protocol stack in different hardware components.

The NR protocol stack (bottom left corner of Figure 2) features a new layer on top of the PDCP, i.e., the Service Data Adaptation Protocol (SDAP) layer [42], which manages the Quality of Service (QoS) of end-to-end flows, and maps them to local resources in the gNB-UE link. The design of the remaining layers has been updated to support the aforementioned NR features [43–47].

CU/DU Split and the Virtualized RAN Architecture: The main innovation introduced by NR comes from the possibility of splitting the higher layers of the 3GPP stack (PDCP, SDAP, and RRC) and the lower layers (RLC, MAC, and PHY) into two different logical units, called *Central Unit*

[1]Although initially introduced as “New Radio” in [32], NR has lost its original meaning in the latest 3GPP specifications [31] where it now refers to the 5G RAN.

[2]Notice that, while LTE has been originally associated with 4G networks, its evolution (e.g., LTE-A) will be part of the air interface of 5G networks, together with NR [33].

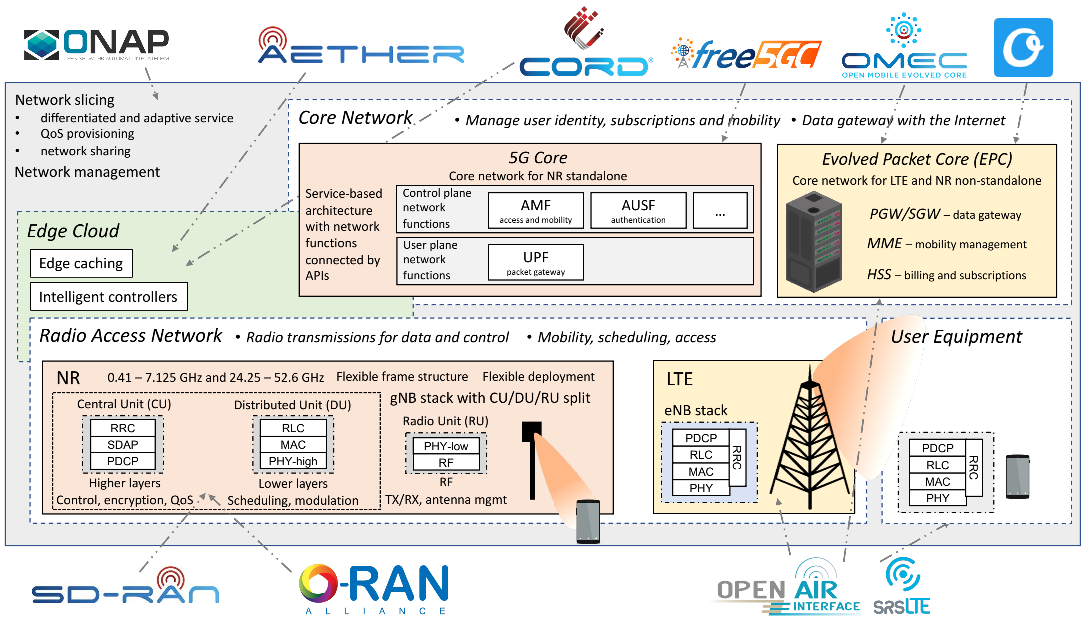

Fig. 2: Cellular network architecture.

(CU) and the *Distributed Unit (DU)*, which can be deployed at separate locations. Moreover, the lower part of the physical layer can be separated from the DU in a standalone *Radio Unit (RU)*. The CU, DU and RU are connected through well-defined interfaces operating at different data rates and latency (with tighter constraints between the DU and RU).

This architecture, proposed by 3GPP in [48], enables the Virtualized RAN (vRAN) paradigm. Specifically, the antenna elements (in the RU) are separated from the baseband and signal processing units (in the DU and CU), which are hosted on generic, even multi-vendor, hardware. If the interfaces between the different RAN components are open, the 5G deployment follows the Open RAN model, which defines open and standardized interfaces among the elements of the disaggregated RAN [49]. A notable example of Open RAN is currently being promoted by the O-RAN Alliance [4]. This consortium has defined a set of interfaces between CU, DU, RU, and a RAN Intelligent Controller (RIC) that can be deployed at the edge of the network (see also Section V-A).

The 5G Core: Openness and flexibility have guides the design of the 5GC, now realized according to a service-based approach [50]. Control and user plane core functionalities have been split into multiple network functions [51]. The 3GPP has also defined interfaces and Application Programming Interfaces (APIs) among the network functions, which can be instantiated on the fly, enabling elastic network deployments and network slicing (Section II-C). The User Plane Function (UPF) is a user plane gateway to the public Internet that acts as mobility anchor and QoS classifier for the incoming flows. On the control plane side, most of the MME functions (e.g., mobility management) are assigned to the Access and Mobility Management Function (AMF). The Session Management Function (SMF) allocates IP addresses to the UEs, and orchestrates user plane services, including the selection of which UPF a UE should use. For a detailed overview of all 5G core functions the reader is referred to [50, 52].

B. Enabling technologies for softwarized 5G cellular networks

5G networks will embody heterogeneous network components and technologies to provide unprecedented performance levels and a unique experience to subscribers. Managing the integration of such a menagerie of technologies, controlling such variegate infrastructure and orchestrating network services and functionalities is clearly no trivial feat. To solve this management and control problem, 5G networks have borrowed widespread and well-established processes and architectures from the cloud-computing ecosystem, where softwarization and virtualization are merged together to abstract services and functionalities from the hardware where they are executed. In the following, we introduce two of these technologies and how they integrate with future 5G systems.

Softwarization and Software-defined Networking: In order to integrate hardware components produced by multiple vendors with different functionalities and configuration parameters, 5G systems rely on *softwarization*. This technology concept grew in popularity in the second decade of the 21st century thanks to the Software-defined Networking (SDN) architectural paradigm and the widespread adoption of the now well-established OpenFlow protocol. As shown in Figure 3, SDN leverages softwarization to decouple network control from the forwarding (or data) plane, thus separating routing and control procedures from specialized hardware-based forwarding operations. By decoupling the functions of these two planes, network control dynamics can be directly programmed in software with an abstract view of the physical infrastructure. Then, a centralized network controller runs the network intelligence, retains a global view of the network, and makes decisions on policies regarding automated network optimization and management, among others.

The fundamental principle of SDN, namely, the separation of data and control, has been adopted by 5G networks to

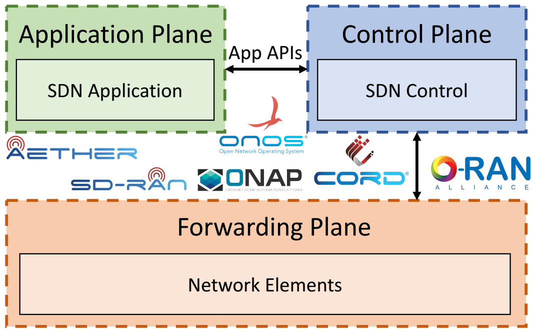

Fig. 3: High-level overview of the SDN architecture.

detach RAN and edge hardware components from their networking and service capabilities. In fact, 5G systems takes softwarization to a broader and comprehensive application range, where it is leveraged to effectively put into practice RAN disaggregation, where RUs operate as basic transceivers and all control and processing operations are performed in software through open interfaces and APIs. This is witnessed by the plethora of open source SDN solutions for mobile networks, also shown in Figure 3, which include Open Networking Operating System (ONOS) [10], Central Office Re-architected as a Datacenter (CORD) [53], O-RAN [4], Open Network Automation Platform (ONAP) [18], Aether [54], and SD-RAN [55].

Network Function Virtualization: NFV brings scalable and flexible management and orchestration to softwarized networks. This is achieved by virtualizing network services and functionalities and decoupling them from the hardware where they are executed. Each functionality is implemented in software via Virtual Network Functions (VNFs), which are executed on Virtual Machines (VMs) instantiated on general-purpose hardware. One of the main advantages of NFV is that each VNF provides atomic functionalities. Therefore, multiple VNFs can be combined together to create more complex and customized network services. Figure 4 depicts the main components of the NFV architecture.

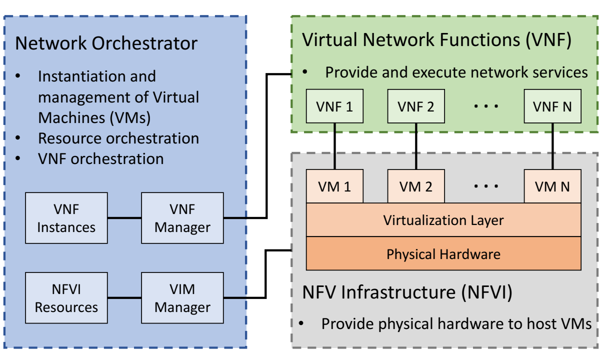

Fig. 4: High-level overview of the NFV architecture.

These are: (i) The network orchestrator, which instantiates and manages the VMs on the physical infrastructure and the services they run; (ii) the VNFs, which are executed on the VMs and implement the network services, and (iii) the Network Function Virtualization Infrastructure (NFVI), which consists of the general purpose physical hardware hosting the VMs deployed by the network orchestrator.

An example of open source network virtualization project is Open Platform for NFV (OPNFV), which facilitates the adoption and development of a common NFVI [56]. OPNFV also provides testing tools, compliance and verification programs to accelerate the transition of enterprise and service provider networks to the NFV paradigm.

C. RAN and Core Network Slicing

The whole 5G network design is rooted in softwarization, virtualization and sharing principles. This strategic design choice paved the way toward a new generation of more efficient, dynamic and profitable networks. Such a revolution has also been made possible by the concepts of *network slicing*.

Network slicing is a multi-tenancy virtualization technique where network functionalities are abstracted from hardware and software components, and are provided to the so-called *tenants* as *slices* [6]. The physical infrastructure (e.g., base stations, optical cables, processing units, routers, etc.) is shared across multiple tenants, each of which may receive one or more slices. Each slice is allocated a specific amount of physical resources and operates as an independent virtual network instantiated on top of the physical one. Although tenants have full control over their slices and the resources therein, they cannot interact with other slices, a concept known as *slice isolation* or *orthogonality* [5].

Each slice provides specific functionalities covering the RAN and the core portions of the network. For example, tenants can be granted RAN slices instantiated on selected base stations providing CaaS (e.g., for private cellular networking) to mobile users [57]. They can also instantiate network slices dedicated to specific services, users and applications. Such a flexible approach makes it possible to instantiate slices dedicated to resource-hungry applications, such as virtual and augmented reality, while simultaneously controlling another slice carrying low-priority traffic generated by browsing activities. An example of practical interest is shown in Figure 5, depicting how slicing technologies enable infrastructure sharing and support the instantiation of multiple slices embedding different infrastructure components.

The figure also lists relevant and well-established open source software projects for effective instantiation, control and configuration of network slices in different portions of the infrastructure (see also Sections V and VI).

The benefits of network slicing include: (i) Each slice can be reserved to handle specific traffic classes with diverse security requirements and is allocated with a different amount of resources, thus enabling *service differentiation* at the infrastructure level; (ii) slicing is controlled by software components, which enable real-time and on-demand instantiation, reconfiguration, and revocation of network slices to adapt to time-varying traffic demand and/or fulfill Service Level Agreements (SLAs), and (iii) underutilized resources can be leased to Mobile Virtual Network Operators (MVNOs) in the form of network slices, thus maximizing resource utilization and generating new profit opportunities for infrastructure providers [58].

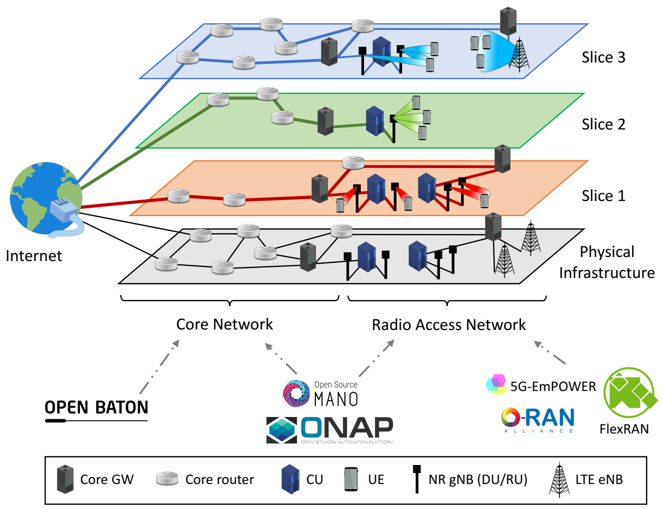

Fig. 5: An example of RAN and CN slicing.

In this context, it is worth mentioning how Operations Support Systems (OSSs) and Business Support Systems (BSSs) will play a vital role in the 5G ecosystem and will determine the success of network slicing. Network slices must be orchestrated, instantiated and revoked dynamically, must satisfy SLA agreements and should be robust against failures and outages. OSSs will serve as a tool to guarantee the fulfillment of services by facilitating closed-loop control and management of network slices. At the same time, operators and infrastructure owners providing slicing services to verticals must generate diversified offers with slices dedicated to services that reflect verticals' needs. BSSs will be necessary to control such a diversified environment and to implement dynamic billing and pricing mechanisms for each slice. Examples of initiatives confirming this trend are Open APIs and Open Digital Architecture led by the industry association TM Forum [59], ONAP [60] and the open source project OpenSlice [61]. Since these benefits affect both business and performance aspects of 5G networks, slicing has become pivotal to 5G systems. In this context, the open source community has led to the development of a variety of solutions to integrate slicing algorithms into the 5G ecosystem [15].

Section V will survey the most relevant and well-established open source projects enabling the delivery and handling of network slicing technologies for network RAN and core.

D. Multi-access Edge Computing

5G systems will leverage advanced and high-performance signal processing and transmission techniques for the highest data rates and QoS possible. However, these technologies alone are not enough to meet the stringent throughput and latency requirements of many 5G applications. For instance, tactile applications as those for virtual and augmented reality, rely upon near real-time processing and interaction with the environment. To be effective these technologies require sub-millisecond transmission times [62]. Furthermore, they involve a significant amount of computation (e.g., augmented reality processors hinge on GPU acceleration), which would result in excessive delay and poor user experience if performed in data centers from a distant cloud. With network technologies of the past meeting the strict constraints of these practical applications was almost considered an utopian task. 5G networks, on the other hand, leverage a simple, yet effective, approach to network architecture design based on softwarization principles to bring services and functionalities to the edge of the network [13].

In this context, Multi-access Edge Computing (MEC) (the technology formerly known as Mobile Edge Computing) has been identified as the solution of the needed *edgification* process. MEC introduces an innovative design shift where essential components of the architecture (both hardware and software) are moved closer to users. By building on edge computing, content caching, NFV and SDN, MEC provides an effective solution to the latency and throughput demands of 5G applications [63]. MEC (i) moves content and functionalities to the edge, meaning that data only sporadically needs to traverse the CN, thus resulting in low latency and in an offloaded core, and (ii) enables localized service provisioning such as private cellular networking, Internet of Things (IoT) data collection and processing at the edge for health and environmental monitoring, and augmented reality for education, telesurgery and advanced industrial applications [64]. There have been several proposals on how to enable MEC in 5G networks. Solutions include those from European Telecommunications Standards Institute (ETSI), which defines the term MEC in [65], and from the 3GPP, which has introduced open interfaces (in Releases 15 and 16) to integrate MEC apps with the softwarized 5G core [66]. The interested readers are referred to [67, 68] for analysis and reviews of different MEC architectures. Open source MEC frameworks and their enablers will be discussed in Section V.

E. Intelligence in the Network

Another key component of the 5G ecosystem is the application of machine learning and artificial intelligence-based technologies to network optimization [69]. The scale of 5G deployments makes traditional optimization and manual configuration of the network impossible. Therefore, automated, data-driven solutions are fundamental for self-organizing 5G networks. Additionally, the heterogeneity of use cases calls for a tight integration of the learning process to the communication stack, which is needed to swiftly adapt to quickly changing scenarios.

Learning techniques for 5G networks have been proposed for different applications. Use cases range from forecasting traffic demands to scale CN resources [73], to predicting HARQ feedback [74] to reduce latency in Ultra Reliable and Low Latency Communication (URLLC) flows, to beam adaptation in mmWave vehicular networks [75]. A summary of results from applying deep learning techniques can be found in [76, 77].

Notably, telecom operators have embraced the deployment of machine learning techniques for self-managed and self-optimized networks. The integration of machine learning in real deployments, however, faces several architectural and

TABLE I: Open source RAN software.

RAN Software	eNB	gNB	SDR UE	COTS UE Support	License	Main Contributor(s)	Community Support
OpenAirInterface [70]	yes	under development	yes (unstable)	yes	OAI Public License v1.1	OAI Software Alliance, EURECOM	mailing list
srsLTE [71]	yes	under development	yes	yes	GNU AGPLv3	Software Radio Systems	mailing list
Radisys [21, 72]	no	yes, (O-RAN)	no	N/A	Apache v2.0, O-RAN Software License v1.0	Radisys	no

procedural challenges [78]. This is primarily because real-time network telemetry and data need to be collected and aggregated to allow the data intensive learning operations of training and inference. The previously discussed MEC paradigm has been proposed as an architectural enabler for applying machine learning to networking, with intelligent controllers deployed at the edge of the network and integrated to the RAN. A software-based framework that implements this paradigm is O-RAN [4], which envisions a RAN Intelligent Controller (RIC) interfaced with gNBs and eNBs, for monitoring, learning, and performing closed-loop actuation. We discuss the O-RAN architecture in detail in Section V-A.

III. The Radio Access Network

This section describes the open source libraries and frameworks for 4G and 5G cellular networks to deploy a software-defined RAN. The most relevant of these open source software frameworks and their features are listed in Table I.

A. *OpenAirInterface*

The OpenAirInterface Radio Access Network (OAI-RAN) [17, 79] provides software-based implementations of LTE base stations (eNBs), UEs and EPC (OAI-CN; see Section IV) compliant with LTE Release 8.6 (with an additional subset of features from LTE Release 10).

The OAI-RAN source code is written in C to guarantee real-time performance, and is distributed under the OAI Public License [80], a modified version of the Apache License v2.0 that allows patent-owning individuals and companies to contribute to the OAI source code while keeping their patent rights. Both the eNB and UE implementations are compatible with Intel x86 architectures running the Ubuntu Linux operating system. (An experimental version for the CentOS 7 is under development.) Several kernel- and BIOS-level modifications are required for these implementations to achieve real-time performance, including installing a low-latency kernel, and disabling power management and CPU frequency scaling functionalities.

eNB Implementation: At the physical layer, the eNB can operate in Frequency Division Duplexing (FDD) and Time Division Duplexing (TDD) configurations with 5, 10, and 20 MHz channel bandwidths, corresponding to 25, 50, and 100 Physical Resource Blocks (PRBs). As for the transmission modes, it supports Single Input, Single Output (SISO), transmit diversity, closed-loop spatial multiplexing, Multi-user Multiple Input, Multiple Output (MIMO) (MU-MIMO), and 2×2 MIMO. Channel quality reports are sent through standard-compliant Channel Quality Informations (CQIs) and Precoding Matrix Indicatorss (PMIs). Finally, OAI-RAN also supports HARQ at the MAC layer.

In Downlink (DL), OAI-RAN implements synchronization signals used by UEs to acquire symbol and frequency synchronization (Primary Synchronization Signal (PSS) and Secondary Synchronization Signal (SSS)), and channels that carry information on the DL configuration used by the eNB (Physical Broadcast Channel (PBCH)) and on the DL control channel (Physical Control Format Indicator Channel (PCFICH)). The OAI-RAN eNB also implements the Physical Downlink Control Channel (PDCCH), which carries scheduling assignments of the UEs and DL control information, and the Physical Downlink Shared Channel (PDSCH), which transports data intended for specific UEs. Finally, ACKs/NACKs for the data received in uplink from the UEs are sent through the Physical Hybrid ARQ Indicator Channel (PHICH), while broadcast and multicast services are provided through the Physical Multicast Channel (PMCH).

In Uplink (UL), it supports the Physical Random Access Channel (PRACH), which is used by UEs to request an UL allocation to the base station, as well as channels carrying reference signals from the UE to the eNB (Sounding Reference Signal (SRS) and Discovery Reference Signal (DRS)). Data from the UEs to the eNB is carried by the Physical Uplink Shared Channel (PUSCH), while the Physical Uplink Control Channel (PUCCH) is used to transmit UL control information. Modulations up to 64 Quadrature Amplitude Modulation (QAM) and 16 QAM are supported in DL and UL, respectively.

The E-UTRAN stack of the eNB implements the MAC, RLC, PDCP, and RRC layers and provides interfaces to the core network with support for IPv4 and IPv6 connectivity (see Section II-A for a detailed description of these layers). As for the MAC layer scheduling, OAI-RAN implements a channel-aware proportional fairness algorithm commonly used in commercial cellular networks, as well as greedy and fair round-robin scheduling algorithms.

The eNB can be interfaced with both commercial and open source EPCs (e.g., OAI-CN and Open5GS; Section IV), and with a number of SDRs, including Ettus Research [81] B-series Universal Software Radio Peripherals (USRPs), e.g., USRP B210, and X-series, e.g., USRP X310 (see Section VII for a comprehensive description of compatible hardware platforms). However, to the best of our knowledge, at the time of this writing the eNB application executed over USRPs X-series appears to be less than fully stable. Both Commercial Off-the-Shelfs (COTSs) smartphones and SDRs can be used as UEs. However, OAI privileged the development of the eNB application rather than the UE one, which may result in connectivity issues between the two.

OAI-RAN also includes a simulation environment implementing layer-2 and layer-3 functionalities only, without the need to interface with any external SDR device. Being transparent to layer-1 procedures, the simulation environment provides a useful tool to evaluate the performance of algorithms and protocols at the upper-layers. Finally, NR-compliant applications for base stations (gNB), UEs and core network (5GC) are currently being developed [17, 82]. At the time of this writing, a major NR release has not been announced yet.

Sample Use Cases: OpenAirInterface has witnessed recent widespread adoption by both academia and industry. For instance, Kaltenberger et al. leveraged OAI to build a Cloud-RAN Massive MIMO testbed with Remote Radio Units (RRUs) built from commodity hardware [83]. Foukas et al. proposed Orion [84] and FlexRAN [85], two RAN-oriented centralized network virtualization solutions based on OAI. Liu et al. implemented a learning-assisted network slicing solution for cyber-physical systems on OAI [86], while D'Alterio et al. leveraged OAI to prototype and experimentally evaluate the performance of a Unmanned Aerial Vehicle (UAV)-based eNB [87].

Fujitsu is integrating and testing OAI in commercial units of its proprietary infrastructure [88], while WindyCitySDR is leveraging OAI to create low-bandwidth mobile phone data networks [89]. InterDigital and SYRTEM are developing mmWave software solutions and devices based on the OAI implementation [90, 91].

The full potential of an open and softwarized approach to cellular networking is demonstrated by Bonati et al. [92]. Specifically, a softwarized automatic optimization framework with RIC functionalities, called CellOS, is instantiated on a network with eNBs featuring an enriched version of OAI. In the experimental setting 3 eNBs serve a total of 9 UEs (COTS smartphones). Figure 6 compares the throughput achieved by the CellOS-driven automatic user scheduling optimization to that achieved by the OAI-RAN proportional-fairness and greedy scheduling algorithms.

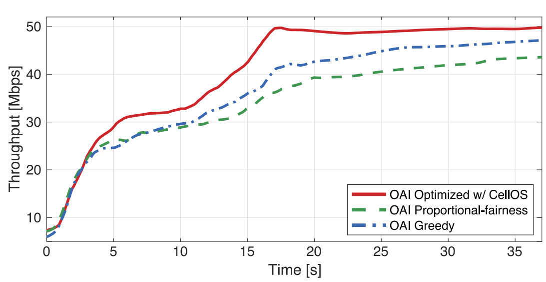

Fig. 6: Software-defined optimization with OpenAirInterface and CellOS [92].

The CellOS optimized approach increases the network throughput significantly, and reduces the convergence time to stable high throughput with respect to the other schedulers. This simple yet effective experiment shows the importance of gaining access and reconfiguring via software network parameters and protocols (the scheduling algorithm, in this case). Without open and programmable software such as OAI, it would have been unfeasible to change scheduling policies baked into hardware components and improve network performance swiftly and automatically.

B. *srsLTE*

Similarly to OAI-RAN, srsLTE [20, 93] provides software implementations of LTE eNB, UE, and EPC (discussed in Section IV) compliant with LTE Release 10 (with some features from higher versions, e.g., NR Release 15). The software suite is written in the C and C++ programming languages and it is distributed under the GNU AGPLv3 license [94]. srsLTE is compatible with the Ubuntu and Fedora Linux distributions. It does not require any kernel- or BIOS-level modifications to achieve real-time performance (unlike OAI; Section III-A). (Disabling CPU frequency scaling is recommended.)

eNB Implementation: At the physical layer, the eNB implementation supports FDD configurations with channel bandwidths of 1.4, 3, 5, 10, 15, and 20 MHz, corresponding to configurations from 6 to 100 PRBs. The available transmission modes are single antenna, transmit diversity, Cyclic Delay Diversity (CDD), and closed-loop spatial multiplexing. The channels supported in DL and UL are the same of OAI (Section III-A), with modulations up to 256 QAM.

Similar to OAI, the E-UTRAN stack of srsLTE eNB implements the MAC, RLC (TM, AM, and UM modes are supported), PDCP, and RRC layers (Section II-A). The eNB interfaces with the CN through the S1 Application Protocol (S1AP) and GPRS Tunneling Protocols (GTPs) interfaces, and supports IPv4 connectivity. It can be used to serve both COTS and SDR UEs, which can be implemented through srsLTE UE application.

UE Implementation: The UE implementation features PHY, MAC, RLC, PDCP, and RRC layers as the ones of the eNB. Additionally, srsUE also includes a Non-Access Stratum (NAS) layer that manages control plane communication between UE and CN, and a Gateway (GW) layer. The latter supports IPv4 and IPv6 connectivity, and is used to create a virtual interface on the machine that runs the user application to tunnel IP packets from/to the RF front-end.

To authenticate users and CN, the UE application supports both soft and hard Universal Subscriber Identity Module (USIM) cards. These are meant to contain values to uniquely identify the UE in the network, such as the International Mobile Subscriber Identity (IMSI), the authentication key (K), the operator code (OP), and the phone number. The soft USIM can work with both the XOR [95] and Milenage [96] authentication algorithms, and the previously mentioned UE authentication values are stored in a configuration file. The hard USIM, instead, requires a physical SIM card that needs to be programmed with the user parameters discussed above through a smart card reader/programmer. This SIM card, then, needs to be connected to the host computer that runs the UE application, for instance, through the same smart card reader/programmer.

As RF front-end, both eNB and UE applications are compatible with several of the boards that will be described in

Section VII, including USRP B- and X-series (i.e., USRP B210, B205mini-i, and X310), as well as limeSDR [97], and bladeRF [98]. To analyze some of the eNB and UE capabilities in a controlled environment, srsLTE provides utilities to simulate dynamics such as uncorrelated fading channels, propagation delays, and Radio-Link failures between eNBs and UEs. Finally, at the time of this writing, srsLTE is working toward NR compatibility. An initial support of NR at the MAC, RLC, RRC, and PDCP layers is included in the latest releases of the code.

Sample Use Cases: Several recent works have been using srsLTE to investigate the security of LTE networks. Bui and Widmer proposed OWL, an srsLTE-based framework to capture and decode control channel of LTE devices [99]. OWL is then leveraged by Meneghello et al. in [100], and Trinh et al. in [101] to fingerprint LTE devices through machine-learning approaches. Kim et al. designed LTEFuzz, a tool for semi-automated testing of the security of LTE control plane procedures [102], while Rupprecht et al. carried out a security analysis of LTE layer 2 [103]. Yang et al. proposed and evaluated SigOver, an injection attack that performs signal overshadowing of the LTE broadcast channel [104]. Singla et al. designed an enhanced paging protocol for cellular networks robust to privacy and security attacks [105]. The National Institute of Standards and Technology (NIST) built OpenFirst on top of srsLTE, a platform for first responders to test and validate LTE technologies focused on public safety communications [106], while Ferranti et al. experimentally evaluated the performance of a UAV-based eNB leveraging srsLTE [107]. Finally, D'Oro et al. proposed Sl-EDGE, an optimization-based MEC slicing framework to instantiate slice services on heterogeneous devices at the edge of the network [108].

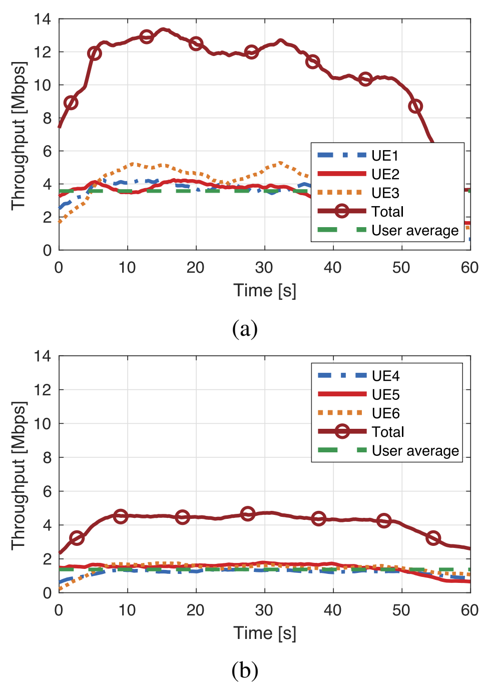

Fig. 7: Softwarized per-slice QoS differentiation on srsLTE using Sl-EDGE [108] and CellOS [92]: (a) Slice 1, premium service, and (b) slice 2, regular service.

One use case of particular interest is that of RAN slicing. The authors of [92] designed CellOS, a RIC which controls resources of the eNBs of the network by interfacing with different RAN software, such as OAI and srsLTE. In this case, the source code of srsLTE is extended to achieve differentiated service through 5G slicing technologies. Figure 7, shows experimental results in which 2 eNBs running srsLTE are serving 6 COTS UEs. First, the resource allocation for two slices, serving premium and regular users of the cellular network, is computed through Sl-EDGE [108], then this is applied by CellOS [92]. The network eNBs allocate 80% of spectrum resources to slice 1 (premium service in Figure 7a), and the remaining 20% to slice 2 (regular service in Figure 7b). As expected, the throughput of the premium UEs of the slice 1 outperforms that of the regular users of slice 2, with average gains of more than 2.5x.

C. *Radisys Open Source RAN Contributions*

Radisys is a 4G/5G vendor that contributes to a number of open source software consortia, including O-RAN and several Open Networking Foundation (ONF) initiatives [72].

As part of O-RAN (Section V-A), Radisys provides an open source implementation of the 3GPP NR stack for the gNB DU [109]. At the time of this writing, this does not represent a complete solution that can be deployed to run real-world experiments (as with OAI and srsLTE), as it lacks integration with open source CU and RU implementations. However, this represents a key first step toward the availability of an open source 5G gNB based on the CU/DU split principle described in Section II-A.

The currently available open source code, licensed according to the Apache License v2.0, provides a complete implementation of the MAC and RLC layers. The Radisys release also provides a layer that manages the operations of the DU and interfaces it with the CU, the RU and external controllers, when available. The codebase is aligned with Release 15 of the 3GPP NR specifications. The NR MAC uses the Functional Application Platform Interface (FAPI) to interact with a scheduler, adapted from an LTE implementation. The RLC layer supports the TM, UM and AM modes (see Section II-A for details on these modes of operation). Additionally, Radisys has open sourced a full implementation for 4G eNBs, licensed with AGPLv3 [110]. However, this implementation concerns the firmware of a specific Qualcomm chipset FSM9955, thus representing a solution for 4G small cell hardware vendors rather than an alternative to srsLTE and OAI.

IV. Core Network

In this section, we describe the main open source solutions for the 4G and 5G core networks, i.e., EPC and 5G Core, respectively. A summary of the solutions discussed in this section is shown in Table II.

A. *Evolved Packet Core*

Implementations of the 4G EPC, discussed in details in Section II-A, typically include components for the Mobility

TABLE II: Open source CN software.

CN Software	EPC	5G Core	License	Main Contributor	Community Support
OpenAirInterface [70]	yes	under development	Apache v2.0	OpenAirInterface Software Alliance, EURECOM	mailing list
srsLTE [71]	yes	no	GNU AGPLv3	Software Radio Systems	mailing list
Open5GS [111]	yes	under development	GNU AGPLv3	Open5GS	mailing list / forum
OMEC [112]	yes	compatible	Apache v2.0	ONF, Intel, Deutsche Telekom, Sprint, AT&T	mailing list
free5GC [113]	no	yes	Apache v2.0	free5GC	forum

Management Entity (MME), the Home Subscription Server (HSS), the Service Gateway (SGW), and the Packet Gateway (PGW).

The MME is responsible for control messages to establish connection with the UEs, paging and mobility procedures. It includes the NAS signaling and security features, as well as tracking area list management, PGW/SGW selection, UE authentication, and reachability procedures. It also takes care of bearer management, i.e., a tunnel between UE and PGW in the case of EPC, and between UE and UPF in the case of 5GC [50, 114]. Moreover, it supports protocols for control plane signaling between EPC and E-UTRAN, reliable message-level transport service. Tunneling protocols for User Datagram Protocol (UDP) control messaging are also provided, as well as protocols for authentication, authorization and charging of UEs.

The HSS implements the user database, and stores information on the subscribers, e.g., identity and key. It is also responsible for user authentication. It provides interfaces for user provisioning in the HSS database, as well as interfaces to connect to the MME.

The SGW and PGW components carry packets through the GTP for both user and control planes, i.e., through the GPRS Tunneling Protocol User Plane (GTP-U) and GPRS Tunneling Protocol Control Plane (GTP-C), which use UDP as transport protocol. Packet routing and forwarding, IP address allocation to UEs, and paging are also supported. Open source implementations of the LTE EPC are provided by OpenAirInterface (with OAI Core Network (OAI-CN)), srsLTE (with srsEPC), Open5GS, and Open Mobile Evolved Core (OMEC).

TABLE III: Implemented EPC interfaces.

Interface	OpenAirInterface [70]	srsLTE [71]	Open5GS [111]	OMEC [112]
MME / HSS				
S1-MME	x	x	x	x
S6a	x	x	x	x
S10	x	-	-	-
S11	x	x	x	x
SGW / PGW				
S1-U	x	x	x	x
S5 / S8	-	x	x	x
S11	x	x	x	x
SGi	x	x	x	x
Gx	-	-	x	x

A summary of the most relevant 3GPP interfaces implemented by each of these EPC softwares is shown in Table III:

- *S1-MME:* It enables the flow of the S1-AP control application protocol between E-UTRAN and MME.
- S6a: It connects MME and HSS, it is used for user authentication and authorization and to transfer user subscriptions.
- *S10:* Control interface among different MMEs.
- *S11:* Control plane interface between MME and SGW used to manage the Evolved Packet System (EPS).
- *S1-U:* It enables the flow of user plane data between E-UTRAN and SGW.
- *S5/S8:* It provides user plane tunneling management, and control services between SGW and PGW.
- *SGi:* It connects the PGW to the Internet.
- *Gx:* It allows the transfer of QoS policies and charging rules from the Policy and Charging Rules Function (PCRF) to the Policy and Charging Enforcement Function (PCEF) in the PGW.

OAI-CN: OAI-CN is written in the C and C++ programming languages, and it is distributed under the Apache License v2.0 [17]. It is compatible with Intel x86 architectures running the Ubuntu Linux operating system. Kernel modifications similar to those discussed in Section III-A for OAI-RAN are required to guarantee real-time capabilities. Dynamic QoS with establishment, modification and removal of multiple dedicated bearers, and policy-based QoS update are also features implemented by the OAI-CN MME. Traffic Flow Template (TFT) operations, such as fault detection, filter rules, and IP-filters are also provided. Finally, implicit (e.g., service request failures) and explicit (e.g., bearer resource and delete commands) congestion indicators are supported, along with multi-Access Point Name (APN), paging, and restoration procedures.

srsEPC: The EPC implementation included in the srsLTE software suite, namely, srsEPC, is written in C++ and distributed under the GNU AGPLv3 license [20]. It is compatible with the Ubuntu and Fedora Linux operating systems. The HSS supports the configuration of UE-related parameters in the form of a simple textual csv file. UE authentication is supported by XOR and Milenage authentication algorithms. srsEPC enables per-user QoS Class Identifier (QCI) and dynamic or static IP configurations.

Open5GS: This EPC is written in C and distributed under the AGPLv3 license [111].[3] It is compatible with a variety of Linux distributions, such as Debian, Ubuntu, Fedora, and CentOS, as well as FreeBSD and macOS. Differently from other EPCs, Open5GS supports the delivery of voice calls and text messages through the LTE network instead of relying on traditional circuit switching networks. This is achieved by leveraging Voice over LTE (VoLTE) and SG-SMS solutions, respectively. Moreover, Open5GS includes a PCRF module, through which operators can specify network policies in real-time, including prioritizing a certain type of traffic. The implementation of 5GC functionalities is currently under development.

OMEC: This is a high performance open source implementation of LTE Release 13 EPC developed by the ONF together with telecom operators and industry partners, such as Intel, Deutsche Telekom, Sprint, and AT&T [112]. OMEC is built using a NFV architecture to sustain scalability in large-scale scenarios such as those of 5G and IoT applications. It is distributed under the Apache License v2.0, and offers connectivity, billing and charging features. OMEC can be used as a standalone EPC, or integrated in larger frameworks, such as Converged Multi-Access and Core (COMAC) (see Section V-C).

Sample Use Cases: Sevilla et al. developed CoLTE, a Community LTE project to bring cellular connectivity in rural areas that are not covered by traditional cellular service [116]. CoLTE interfaces commercial eNBs (BaiCellsNova-233) with OAI-CN, modified to include features such as user billing and accounting. CoLTE is currently porting its implementation from OAI-CN to Open5GS. Core network functions for 5G NR, instead, are being developed by bcom starting from the OAI-CN implementation [117]. Moreover, OAI-CN is used as EPC inside the Magma framework (see Section V-C). Haavisto et al. use NextEPC (now Open5GS) to deploy a 5G open source RAN [9]; Lee et al. interface it with srsLTE to study security concerns of modern cellular networks [118].

B. 5G Core

An open source implementation of the 5G core is offered by free5GC and distributed under the Apache License v2.0 [113]. It is written in the Go programming language, and it is compatible with machines running the Ubuntu Linux operating system. This implementation, which was initially based on NextEPC (now Open5GS), supports the management of user access, mobility, and sessions (AMF and SMF), and the discovery of the services offered by other network functions (Network Repository Function (NRF)). It also includes network functions to select which network slices to allocate to UEs (Network Slice Selection Function (NSSF)), to manage, store and retrieve user data (Unified Data Management (UDM) and Unified Data Repository (UDR)), to perform UEs authentication within the network (Authentication Server Function (AUSF)). Functions for the operation, administration and management of the core network (Operations, Administration and Maintenance (OAM)), and to perform network orchestration, among others, are also included.

[3] Open5GS was previously known as NextEPC [115]. The renaming happened in 2019.

The 3GPP interfaces implemented by free5GC are:

- *N1/N2:* Connect the AMF to the UE and RAN, respectively. They are used for session and mobility management.
- *N3/N4/N6:* Connect the UPF to the RAN, SMF, and data network, respectively. They support user plane functions.
- *N8:* Connects the UDM and the AMF. It enables user authorization procedures.
- *N10/N11:* Connect the SMF to the UDM and AMF, respectively. They handle subscription and session management requests.
- *N12/N13:* Connect the AUSF to the AMF and UDM, respectively. They enable authentication services.

In addition to the EPCs that are evolving toward the 5G architecture (Section IV-A), an open source implementation of the 5GC is currently being developed by Hewlett Packard Enterprise (HPE) [119].

Software implementations of the data plane of the 5GC can also benefit from the acceleration introduced by domain-specific, platform-independent packet-processing languages such as P4 [120], which is a protocol-independent programming language that instructs the networking hardware on how to process packets. Several P4-based UPF implementations have been proposed commercially and in the literature. Most of their software is not open source [121–123].

V. RAN AND CORE FRAMEWORKS

This section describes several open source frameworks that operate both in the RAN and CN domains. While the software described in Sections III and IV performs specific functions (e.g., eNB, UE, or CN), the frameworks that will be introduced in the following paragraphs are more general and broad in scope, and interact with individual components in the RAN and CN for control, management, and coordination.

Table IV compares the features, license and availability of the different frameworks and projects that will be presented throughout this section.

A. O-RAN

The O-RAN Alliance is an industry consortium that promotes the definition of an open standard for the vRAN, with two goals [4]. The first is the integration of machine learning and artificial intelligence techniques in the RAN, thanks to intelligent controllers deployed at the edge [78]. The second is the definition of an agile and open architecture, enabled by well-defined interfaces between the different elements of the RAN. Since all O-RAN components must expose the same APIs, it is easy to substitute components with others offering alternative implementations of the same functionalities. This allows O-RAN-based 5G deployments to integrate elements from multiple vendors, thus opening the RAN market to third-party entities providing new functionalities and diversified services. Moreover, it makes it possible to adopt COTS hardware, in an effort to promote flexibility and reduce costs. Eventually, following the trend started with cloud-native infrastructures,

TABLE IV: Open frameworks and projects.

Framework	Main Focus	Status	License	Main Members	Community Support
Mobile					
O-RAN [21]	Virtualized, intelligent RAN	available	Apache v2.0, O-RAN Software License v1.0	O-RAN Alliance w/ telecom operators	no
COMAC [124]	Agile service delivery at the edge	available	Apache v2.0	ONF	mailing list
SD-RAN [125]	CU/DU control and user planes	under development		ONF	N/A
Aether [126]	5G/LTE, Edge-Cloud-as-a-Service (ECaaS)	under development		ONF	N/A
Magma [127]	CN Orchestration	available	BSD	Facebook	mailing list / forum
OpenRAN [128]	Programmable, disaggregated RAN w/ open interfaces	closed source		TIP	no
Radio Edge Cloud [129]	O-RAN RIC automated configuration / integration testing blueprint	available	Apache v2.0	Akraino	no
Aerial [130]	SDK for GPU-accelerated 5G vRAN	early access	proprietary	NVIDIA	N/A
Slicing					
5G-EmPOWER [131]	Centralized controlled for heterogeneous RAN	available	Apache v2.0	FBK (in the framework of multiple EU projects)	no
FlexRAN [132]	Real-time controller for software-defined RAN	available	MIT License	Mosaic5G Consortium	mailing list
Edge					
CORD [133]	Data center for network edge	available	Apache v2.0	ONF, AT&T, Google, Telefonica	mailing list
LL-MEC [134]	Low-latency MEC and network slicing	available	Apache v2.0	Mosaic5G Consortium	mailing list
LightEdge [135]	MEC services	available	Apache v2.0	FBK (in the framework of multiple EU projects)	N/A

the O-RAN Alliance also aims at promoting open source software as part of the consortium effort.

The O-RAN Architecture: Figure 8 illustrates the high-level architecture and the interfaces of O-RAN [4].

With respect to Figure 2, O-RAN concerns the edge and the 4G and 5G RANs. It is composed by a non-real-time and a near-real-time RAN Intelligent Controller (RIC), and by the eNBs and gNBs. The service management and orchestration framework (top of Figure 8) operates a non-real-time RIC, which performs control decisions with a granularity higher than one second. For example, it can provision the different functions of O-RAN, and train learning algorithms over data provided by the RAN, among others. A near-real-time RIC, instead, performs a control loop with a much tighter timing requirement (with a decision interval as short as $10\mathrm{ms}$), relying on different start, stop, override, or control primitives in the RAN, e.g., for radio resource management. These APIs can be used by different applications installed on the near-real-time RIC (named xApps), which can be developed by third-party entities and pulled from a common marketplace. For example, through the near-real-time RIC and its xApps, an operator can control user mobility processes (e.g., handovers), allocate networking resources according to predicted paths for connected vehicles and UAVs, perform load balancing and traffic steering, and optimize scheduling policies [136]. The near-real-time RIC can also leverage machine learning algorithms trained in the non-real-time RIC. The remaining components of the O-RAN architecture concern the CU/DU/RU into which 5G gNBs are split [137], and the 4G eNBs [34] (bottom of Figure 8). The CU is further split into a control plane CU and a user plane CU. Among the different options investigated by the 3GPP, O-RAN has selected split 7-2x for the DU/RU split [138], in which coding, modulation and mapping to resource elements are performed in the DU, and the inverse FFT, the cyclic prefix addition and digital to analog conversion are carried out in the RU. Precoding can be done in either of the two.

O-RAN Interfaces: O-RAN is in the process of standardizing the interfaces between each of the components in Figure 8. The two RICs interact using the A1 interface, while the non-real-time RIC uses the O1 interface to interact with the RUs and legacy 4G eNBs. The A1 interface [139] allows the non-real-time RIC to provide (i) policy-based guidance to the near-real-time RIC, in case it senses that its actions are not fulfilling the RAN performance goals; (ii) manage machine learning models, and (iii) provide enrichment information to the near-real-time RIC, for example from RAN-external sources, to further refine the RAN optimization. The O1 interface, instead, has operation and management functions, and strives at being compatible with existing standards to permit a seamless integration with existing management frameworks (Section VI) [140]. For example, it relies on the IETF Network Configuration Protocol (NETCONF) [141] and on several 3GPP-defined APIs. The non-real-time RIC uses O1 to (i) provision management, fault supervision, and performance assurance services; (ii) perform traces collection; (iii) start up, register, and update physical equipment, and (iv) manage communication surveillance services.

The near-real-time RIC exposes the E2 interface [142] to multiple elements, i.e., the CU, the DU and the eNB. This interface only concerns control functionalities, related to the deployment of near-real-time RIC control actions to the nodes terminating the E2 interface, and to the management of the interaction of the RIC and these nodes [143].

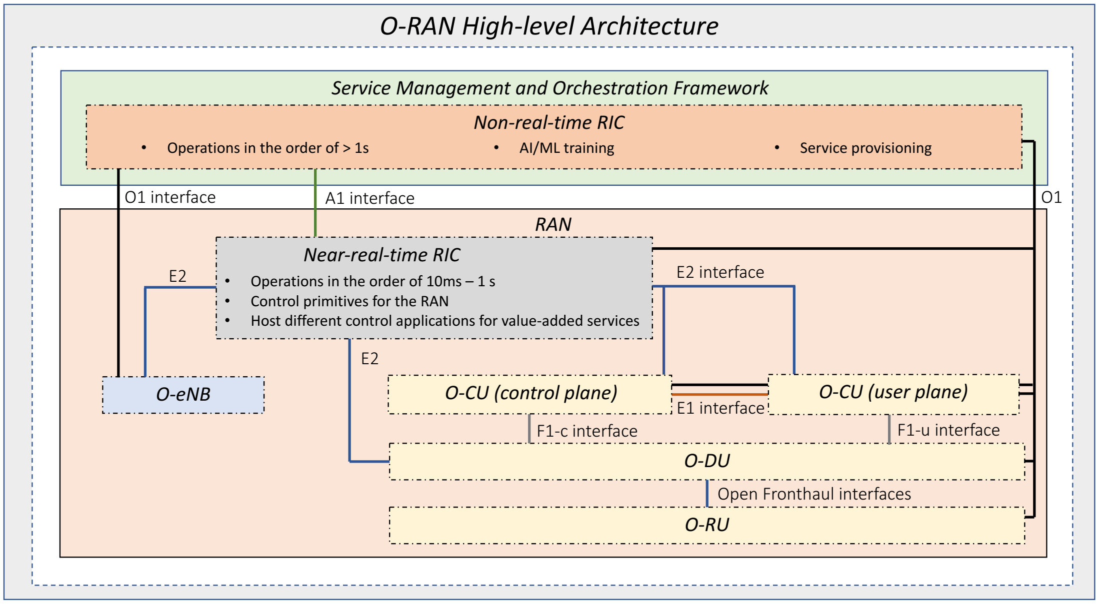

Fig. 8: O-RAN high-level blocks and interfaces.

The E1 and F1 interfaces comply to the specifications from 3GPP. The E1 interface runs between the control and user plane CUs, and its main functions concern trace collection for specific UEs, and bearer setup and management [144]. The F1 interface operates between the CUs and DUs [145]. It has two different versions, one for the control plane and one for the user plane. F1 transports signaling and data between CUs and DUs, to carry out RRC procedures and PDCP-RLC packet exchange. Finally, the interface toward the RU is developed by the Open Fronthaul initiative inside O-RAN [138]. This interface carries compressed IQ samples for the data plane, and control messages for beamforming, synchronization, and other physical layer procedures.

O-RAN Deployment Options: O-RAN envisions different strategies for the deployment of its architecture in *regional* and *edge* cloud locations and at operator-owned cell sites [136]. Each facility could either run O-Cloud, i.e., a set of containers and virtual machines executing the O-RAN code with open interfaces, or be a proprietary site, which still uses the O-RAN open APIs, but could run closed-source code. Both cases are illustrated in Figure 9, which depicts the six different O-RAN deployment combinations indicated in [136].

In Scenario A all the components except the RUs are deployed at the edge of the network, co-located in the same data center that terminates the fronthaul fiber connectivity. Other alternatives foresee the RICs and CUs co-located at a regional cloud facility, with DUs and RUs at the edge or on cell sites. The preferred deployment solution, however, is Scenario B, which deploys the RIC in the regional cloud, CUs and DUs at the edge, and only the RUs in the operator cell sites [136].

The O-RAN Software Community: Besides concerning standardization activities, the O-RAN Alliance has established a *Software Community* in collaboration with the Linux Foundation for contributing open source 5G software that is compliant with the O-RAN specifications [4]. The first two O-

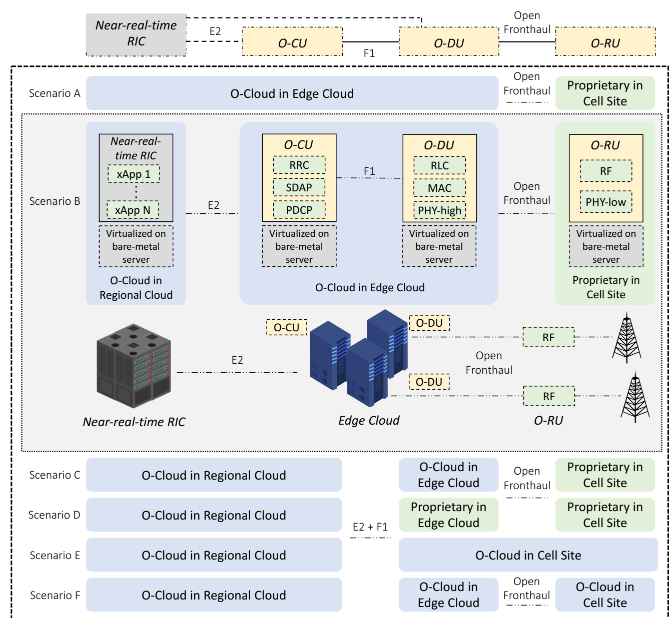

Fig. 9: Logical (top) and physical (bottom) deployment options for O-RAN.

RAN releases date November 2019 (Amber release [146]) and June 2020 (Bronze release [147]), and feature contributions from major vendors and operators, including AT&T, Nokia, Ericsson, Radisys and Intel. These releases include Docker containers (which will be discussed in Section VI-A) and the source code for multiple O-RAN components:

- The non-real-time RIC, with the A1 interface controller, and the possibility of managing machine learning and artificial models in the RAN.
- The platform for the near-real-time RIC, with multiple applications, such as admission control, UE manager, performance and measurement monitor, which talk to the DU through the E2 interface.

TABLE V: ONF frameworks interactions.

	Aether [126]	COMAC [124]	CORD [133]	SD-RAN [125]	OMEC [112]	ONOS [10]
Aether [126]	-	x	x	x	x	x
COMAC [124]	x	-	x	x	x	x
CORD [133]	x	x	-	-	-	-
SD-RAN [125]	x	x	-	-	-	x
OMEC [112]	x	x	-	-	-	-
ONOS [10]	x	x	-	x	-	-

- The DU, as previously discussed in Section III-C, and an initial version of the fronthaul library.
- A framework for operation, administration, and maintenance, and the virtualization infrastructure.

Moreover, a simulator has been developed to test the functionalities of the different interfaces.

Expected Use Cases: The full realization of the O-RAN vision will revolutionize not only the modus operandi and business of telecom operators [136], but also the world of those scientists, researchers and practitioners that will be able to run a modern, open source, full-fledged 5G control infrastructure in their lab and investigate, test and eventually deploy all sorts of algorithms (e.g., AI-inspired) for cellular networks at scale. Researchers will be able to deploy and use the open source software provided by the O-RAN community to develop, test, and evaluate real-time RAN control applications. The O-RAN open source suite will enable 5G networking in a standardized environment, thus allowing reproducibility and easing future extensions. Moreover, deploying third-party xApps in the RIC (in collaboration with telecom operators) could enable experimentation running directly on O-RAN-compliant cellular networks. These tests could start in labs to be then they scaled up to larger deployments in the operator network, using the RICs of both scenarios as a *trait d'union*.

At the time of this writing, however, O-RAN is not at production-level. Therefore, future releases (e.g., the Cherry release expected in December 2020) will attempt to complete integration of the different RAN components with the RICs. A parallel development effort is also being led by the SD-RAN project [125], aiming at an open source, 3GPP-compliant RAN integrated with the O-RAN RIC and interfaces. According to [55], the reference code will include the DU and CU, interoperable with third-party RUs, and a near-real-time RIC based on ONOS [10].

B. Open Networking Foundation Frameworks

The Open Networking Foundation (ONF) is a consortium of several telecom operators that contribute open source code and frameworks used for the deployment of their networks. Specific examples include OMEC (Section IV-A) and the aforementioned SD-RAN [55] and ONOS [10]. The ONF generally distinguishes between *Component Projects*, which are frameworks and/or software that serve a specific purpose, and *Exemplar Platforms*, which combine several Component Projects in a deployable, proof-of-concept reference design. The different projects developed by the ONF are characterized by modular design, facilitating the integration of component projects, and providing the means to incorporate new open source projects. Additionally, some of them (e.g., ONOS and Trellis) integrate P4 packet processing pipelines. Table V summarizes the dependencies across different Component Projects and Exemplar Platforms.

Component Projects: The ONF currently overlooks the development of 10 open source component projects, concerning SDN, transport networks, programmable networking hardware, and mobile networks [148]. In the last category, besides OMEC and SD-RAN, notable efforts include CORD [53], which is an open source project (also part of the Linux Foundation portfolio) for deploying and managing edge cloud facilities for the MEC (Section II-D). The CORD framework is based on multiple software solutions that, together with reference hardware design, realize a reference MEC architecture based on SDN, NFV and cloud-native solutions. CORD aims at (i) reducing deployment costs by using commodity hardware, and (ii) enabling innovative services, thanks to well-defined APIs for accessing edge computing facilities and multi-domain security. Moreover, CORD can be easily extended to address the heterogeneous requirements of different markets. In particular, two CORD architectures specific for mobile and residential services have been spawned off into two Exemplar Platforms (SDN-Enabled Broadband Access (SEBA) [149] and COMAC [124]). CORD is one of the ONF projects with the largest number of contributions by the open source community. It includes detailed installation, operation and development guides [150], and a set of repositories with its source code [151].

Another project related to software-defined mobile networks is ONOS [10], an open source operating system for networking projects. While it has mostly been used for SDN deployments in wired networks, ONOS will provide a common substratum for SD-RAN and several Exemplar Platforms, such as Aether and COMAC, described next.

Exemplar Platforms: An Exemplar Platform is given by extending a Component Project to implement a specific target or by combining and integrating multiple projects that can be deployed as a proof of concept. Among those currently available the following ones concern cellular and mobile networks [152].

- COMAC, which extends CORD into a platform that targets the integration of multiple access and CN technologies, including 4G and 5G cellular networks, broadband, fiber and cable networks, and Wi-Fi deployments [124]. The framework provides a common data plane in the core, which aggregates user data to and from different access technologies, and the possibility of managing users' subscriptions and identities with a single manage-

ment platform. COMAC is based on the SEBA platform (a lightweight multi-access technology platform, which provides high-speed links from the edge of the network to the backbone of the infrastructure), and on multiple Component Projects, such as OMEC, for the mobile core and edge, and CORD for the broadband subscriber management. Moreover, it will exploit O-RAN (with the SD-RAN implementation) for the control plane of the mobile cellular access. The first COMAC release [153] provides instructions on how to configure the different software components to actually set up the overall platform (except for the SD-RAN portion that will be made available in future releases). Additionally, a self-contained COMAC-in-a-Box can be used to install the whole platform on a single server or virtual machine, to run end-to-end tests through an emulated data-plane (based on the OAI simulator, introduced in Section III-A) and the virtualized core and management environments [154].
- Aether, for streamlined deployment of private enterprise cellular networks [126]. It combines three main elements, namely, a control and orchestration interface to the RAN, an edge cloud platform (the Aether edge), with support for cloud computing APIs, and a central cloud (the Aether core), for orchestration and management [54]. The Aether project will build and integrate several ONF efforts, including SD-RAN, ONOS, CORD and OMEC. At the time of this writing, the source code and the deployment pipeline are not publicly available. When the code will be released, besides providing an opportunity for private 5G networks, Aether could be effectively used to deploy and manage integrated RAN-edge testbeds for 5G research and innovation.

C. Other Frameworks and Projects

Along with O-RAN and the ONF solutions, several open source communities (e.g., from 5G European projects) and companies have released frameworks and projects targeting connectivity, slicing and core-related functionalities. A few noteworthy examples are presented next.

5G-EmPOWER: 5G-EmPOWER [155] is an operating system for heterogeneous RAN architectures. It consists of an open source and reprogrammable software platform abstracting the physical RAN infrastructure and providing high-level APIs to control RAN functionalities. The code of the platform is released under the Apache License v2.0 [156].

5G-EmPOWER embraces the SDN philosophy to decouple control and data planes. This separation is obtained in practice via two main components, i.e., a centralized controller and a set of agents. The centralized controller (i) acts as an operating system with complete visibility of the physical infrastructure and its functionalities, and (ii) orchestrates the agents' actions via control directives sent through the OpenEmpower protocol [155]. In turn, agents (i) run on each network element; (ii) abstract the underlying RAN-specific protocol implementations (e.g., LTE, Wi-Fi) to the controller, and (iii) modify the underlying protocol parameters according to the controller's directives.

5G-EmPOWER currently supports several mobile Radio Access Technologies (RATs) such as LTE via srsLTE, Wi-Fi, and LoRa. The 5G NR is not supported yet. Integration of diverse RATs is obtained through agents embedding specialized *wrappers*, one for each RAT. While the general architecture of the agent is RAT-independent, the wrapper is RAT-specific. For instance, new RATs (e.g., 5G NR) can be integrated by implementing new wrappers.

Despite the current lack of support for 5G NR, 5G-EmPOWER already integrates relevant 5G-related technologies such as RAN slicing. Specifically, it provides software components that allow the instantiation of customized and isolated RAN slices on top of a shared physical infrastructure. Each RAN slice is created from a Slice Descriptor specifying SLAs and users belonging to each slice. A slice resource manager and a hypervisor are, then, in charge of admitting/revoking RAN slices and provisioning them with a certain amount of resources necessary to meet the corresponding SLA.

FlexRAN: FlexRAN leverages abstraction and softwarization technologies to develop a RAT-independent RAN management platform [85]. FlexRAN embraces SDN principles to decouple control and data planes. The control plane is orchestrated by a real-time centralized controller, which controls a set of agents, one for each network element. FlexRAN implements a set of REST APIs in JSON format describing the northbound interface of FlexRAN. These APIs are used by the agents to interface with base stations, thus enabling control of the protocol stack and functionalities of the base stations (i.e., MAC, RRC, PDCP).

FlexRAN directly interfaces with OAI. As such, it does not support 5G NR communications yet. However, the northbound REST APIs can be used to specify and reconfigure slicing policies and requirements, providing support for 5G technologies such as RAN slicing. The FlexRAN code is available upon request and released as part of the Mosaic5G project under MIT license [157].

Magma: Magma is a framework developed by the Facebook Connectivity initiative for simplifying the deployment of cellular networks in rural markets [127]. Notably, its goal is to avoid dependence on a specific access technology (i.e., cellular or Wi-Fi) or on a generation of 3GPP core networks. Moreover, it avoids vendor lock-in for telecom operators, while offering advanced automation and federation capabilities. The latter is particularly relevant in rural and underdeveloped scenarios, as it allows the pooling of resources from multiple network operators. Magma is made up of three main components:

- The access gateway, which interfaces the access network to the CN. The current Magma release supports an LTE EPC, and has been tested as termination point for the S1 interface of some commercial LTE base stations (see sections III and IV for more details).
- A cloud-based orchestrator, which monitors the operations of the wireless network and securely applies configuration changes. It exposes an analytics interface providing control and traffic flow information.
- A federation gateway, which is a proxy between the Magma core running in the access gateway and the net-

work operator 3GPP-compliant core. This proxy exposes the 3GPP interfaces to the different CN components, thus bridging the local mobile deployment with the operator backbone.

LL-MEC: LL-MEC is an open source MEC framework for cellular systems compliant with 3GPP and ETSI specifications [158]. This framework merges SDN, edge computing and abstraction principles to provide an end-to-end platform where services requested by mobile users are executed on edge nodes of the network. LL-MEC consists of two main components: The Edge Packet Service controlling core network elements (e.g., routers and gateways) via OpenFlow APIs; and the Radio Network Information Service interfacing the data plane and physical RAN elements (e.g., eNBs) via the FlexRAN protocol [85]. Aside from MEC capabilities, LL-MEC supports network slicing for differentiated services applications with diverse latency and throughput requirements. The LL-MEC code is available upon request and released as part of the Mosaic5G project under Apache License v2.0 [157].

LightEdge: LightEdge is a MEC platform for 4G and 5G applications compliant with ETSI MEC specifications [159]. LightEdge allows network operators to provide MEC services to mobile users through cloud-based applications. The framework provides a Service Registry summarizing services and applications registered to the MEC platform. LightEdge also includes modules and libraries for real-time information exchange across applications and services, and to perform traffic steering to and from the cellular network. LightEdge supports multiple eNBs and is compatible with several open source projects such as srsLTE, Open5GS, and srsEPC. The LightEdge code is available from the project repository under the Apache License v2.0 [160].

OpenRAN: The Telecom Infra Project (TIP) OpenRAN project aims at developing fully programmable RAN solutions that leverage general purpose hardware platforms and disaggregated software [161, 162]. Albeit being closed source, OpenRAN implements open interfaces among the various elements of an NR-enabled cellular network, such as CU, DU, and RU (see Figure 2). One of the projects spawned from OpenRAN is OpenRAN 5G NR, for defining a white-box gNB platform [128, 163]. This platform allows equipment manufacturers to build flexible 5G RAN solutions for seamless multi-vendor support.

Akraino REC: Akraino Radio Edge Cloud (REC) is a blueprint to support and meet the requirements of the O-RAN RIC (Section V-A) [129, 164, 165]. It is part of the Telco Appliance blueprint family [166]. Its features include automated configuration and integration testing to facilitate the management and orchestration of the virtualized RAN. The blueprint is made up of modular building blocks and provides an abstraction of the underlying hardware infrastructure, allowing O-RAN RIC to run on top of it, and to seamlessly interface with the provided APIs.

NVIDIA Aerial: NVIDIA Aerial is a set of Software Development Kits (SDKs) that allows to build Graphics Processing Unit (GPU)-accelerated software-defined, cloud-native applications for the 5G vRAN [130]. At the time of this writing, Aerial provides two main SDKs: cuBB and cuVNF.

- The cuBB SDK provides a highly-efficient 5G signal processing pipeline that runs PHY-layer operations directly on a GPU to deliver high throughput.
- The cuVNF SDK provides optimized input/output and packet placement in which 5G packets are directly sent to the GPU from compatible Network Interface Cards (NICs). Packets can be read and written directly from the NIC to the GPU memory bypassing the Central Processing Unit (CPU), thereby reducing latency. This SDK also allows developers to implement additional VNFs, e.g., deep packet inspection, firewall and vRAN.

VI. Open Virtualization and Management Frameworks for Networking

Besides RAN and CN software, virtualization and management frameworks have an important role in the management and deployment of end-to-end, carrier-grade networks. Several communities and consortia have led the development of open source frameworks that have been deployed at scale by major telecom operators for the management of their physical and virtual infrastructure [22, 167, 168].

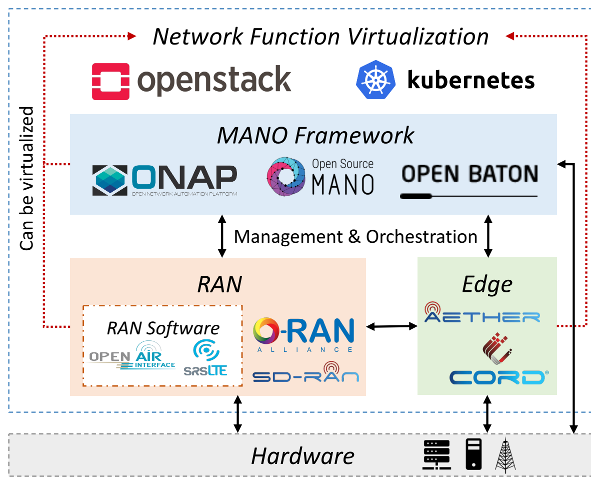

Fig. 10: High-level relationship among MANO, RAN, and edge frameworks, and virtualization components.

ETSI has defined a set of common features that an NFV Management and Orchestration (MANO) framework should have, mainly for orchestrating network functions [169, 170]. Figure 10 depicts where these NFV components fit in the 5G ecosystem. Table VI summarizes the main differences between the frameworks that will be described later in this section, i.e., ONAP, Open Source NFV Management and Orchestration (OSM), and Open Baton. NFV orchestrators are in charge of provisioning network services, i.e., combinations of physical and virtual network functions that can be chained together with a specific topology, managing their creation and life-cycle [173]. Notably, during the initialization of a network service, a basic configuration (0-day) is applied by default. Then, the MANO framework advertises the actual configuration for the function or service (1-day). Finally, updates (2-day configurations) can be deployed at a later stage. These operations are performed in concert by the different components of the NFV orchestrator. Following the ETSI

TABLE VI: Comparison among different VNF orchestrators.

	ONAP [18]	OSM [171]	Open Baton [172]
Community	Linux Foundation w/ telecom operators	ETSI w/ telecom operators	Fraunhofer FOKUS TU Berlin
Compliance w/ ETSI MANO	in progress	yes	yes
External APIs	REST APIs (for external controllers, OSS/BSS, etc.)		Java SDK
Network Services	VNFs and PNFs		VNFs
Infrastructure	Virtual Machines		
	Containers w/ Kubernetes and Docker	Containers w/ Kubernetes	Containers w/ Docker

architecture [169, 170], an NFV orchestrator is composed (i) of a subsystem that manages the virtualization infrastructure (e.g., Virtualization Infrastructure Manager (VIM) frameworks, such as OpenStack, Kubernetes, and Docker) and the connections to the physical hardware; (ii) of the actual MANO framework, and (iii) of the collection of VNFs that it manages. These frameworks are equipped with southbound and northbound APIs to interact with other cellular infrastructure components, as shown in Figure 10, such as edge frameworks for governing the RAN environment (e.g., Aether and CORD), and RAN frameworks (see Section V for more details). The latter include O-RAN and SD-RAN (described in Sections V-A and V-B, respectively), which execute functions such as bringing intelligence to the network (Section II-E) and interacting with open source software, such as OAI and srsLTE (Section III). These, in turn, focus on the CU/DU split introduced by 5G NR, and act as cellular base stations.

In the remainder of this section we discuss virtualization techniques and VIMs (Section VI-A) and we describe popular MANO frameworks, such as ONAP, OSM, and Open Baton (Sections VI-B, VI-C, and VI-D, respectively).

A. Virtualization Techniques

The NFV paradigm decouples the services deployed in a network from the hardware infrastructure on which they run. Applications are packaged into hardware-independent virtual machines, which can be instantiated on different physical machines. This way, NFV eliminates the need for hardware dedicated to each network function and enables scalability of network services.

NFV, whose high-level architecture is depicted in Figure 11, provides many different ways to decouple applications and services, also known as VNFs, from the general-purpose infrastructure on which they run, thus improving scalability and portability. The most common approaches are: (i) Traditional VMs; (ii) bare-metal hypervisors; (iii) containers, and (iv) unikernels.[4] In the NFV paradigm, a hypervisor can be used to create/run VMs, containers, and unikernels, as well as to manage their resource allocation over the physical hardware. Finally, VIMs are leveraged to control the NFV infrastructure at a higher level.

Traditional Virtual Machines: A traditional VM emulates a computer operating system through a guest operating system and kernel (Figure 11). To provide machine-level isolation, the VM requires the virtualization of the hardware of the physical machine on which it runs (called "host machine"). This task is taken care of by the hypervisor, which is *hosted* by the operating system of the physical machine and coordinates resource allocation between host and virtual machines. In general, traditional VMs are considered a resource-heavy approach because of the many hardware virtualization requirements (e.g., virtual disk, CPU, network interfaces, etc.) that are needed to run the VNFs.

Bare-metal Hypervisors: The approach of a bare-metal hypervisor VM is similar to that of traditional VMs, although the hypervisor *directly runs* on the *bare-metal hardware* of the host, without requiring a host operating system (Figure 11). Additionally, a bare-metal hypervisor can be used to run and manage a *container or unikernel* (described next) instead of a full-fledged VM.

Containers: Containers are virtual environments that package a specific code and its dependencies to run applications and services in a virtualized way. They are isolated from each other (through *namespace isolation*) and share access to the operating system and kernel of the physical machine on which they run. They only require a minimal guest operating system instead of the heavy and resource-wasteful hardware virtualization required by VMs (Figure 11). Containers can be maintained both by a container manager interfaced with the operating system of the host machine, or by a hypervisor directly running on a bare-metal host machine. The most widespread open source container virtualization systems are Linux Containers (LXC) [174] and Docker [175] (Figure 11):

- *LXC* was the first major implementation of the modern containers. It leverages control groups and namespace isolation to create virtual environments with separated networking and process space.
- *Docker* enables the creation of containers, and uses virtualization at the operating system level to deploy them on the physical machine. Differently from LXC, on which it was initially based upon, Docker breaks applications, services, and dependencies into modular units and layers inside each container. Additionally, these layers can be shared among multiple containers, increasing the efficiency of Docker container images. Compared to LXC, Docker containers lack some UNIX functionalities and subsystems.

Unikernels: Unikernels are minimal, lightweight, specialized images built with the sole purpose of running specific applications (Figure 11). They compile application services and dependencies into the executable virtual images, without including unnecessary components that would be, instead, included by a generic operating system. This way, unikernels achieve better performance than traditional containers and virtual machines. Since unikernels only include the software

[4]Another virtualization solution that has recently been proposed is that of *serverless computing*, or *Function-as-a-Service (FaaS)*, in which the virtualization platform only spawns computing resources for specifying functions invoked by complex applications, without the need to manage a container or VM. In this work, however, we focus primarily on the most common approaches in the NFV domain.

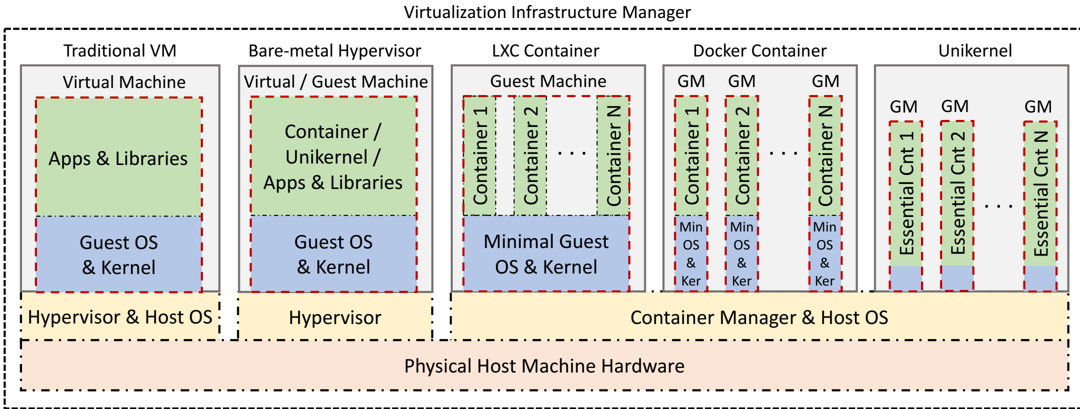

Fig. 11: High-level NFV architecture.

components that are needed to run the application of interest, they also improve the security of the system by exposing fewer functionalities that can be attacked by malicious entities.

Examples of unikernel systems are: (i) ClickOS [176], IncludeOS [177] and OSv [178], which focus on high-performance, low-latency, and secure applications; (ii) MirageOS [179], which includes several libraries that are then converted to kernel routines upon image compilation, and (iii) UniK [180], which deals with compilation and orchestration of unikernel images.

Unikernels applications for cellular networks include the following. Wu et al. that integrates Android system libraries into OSv to offload mobile computation for Mobile Cloud Computing (MCC) and Mobile Fog Computing (MFC) [181]. Valsamas et al. propose a content distribution platform for 5G networks that is based on unikernels such as ClickOS, OSv, and MirageOS [182]. A performance comparison of the IncludeOS unikernel and Docker containers instantiated as VNF for 5G applications is carried out in [183].

***Hypervisors**:* A hypervisor is software that creates and runs virtual machines, the *guest machines*, on a physical computer, called the *host machine*. Key tasks of an hypervisor include (i) providing isolation between virtual/guest machine and the host machine; (ii) managing allocation/reallocation of resources, such as CPU, memory and storage, to the guest machines, and (iii) scheduling of resources among host and guest machines.

There are two types of hypervisors: Type 1 and type 2 hypervisors. The former are referred to as *bare-metal hypervisors* and manage the guest operating system by running them directly on the host hardware, thus acting as an operating system for the host machine. Examples of hypervisors of type 1 are Xen [184] and VMware ESXi [185]. Type 2, instead, are referred to as *hosted hypervisors* and run on top of the host operating system as a software layer or application. Examples of hypervisors of type 2 are Linux Kernel-based Virtual Machine (KVM) [186], BSD bhyve [187], and Oracle VirtualBox [188].

***Virtualization Infrastructure Managers**:* A VIM is in charge of control and management of the NFV infrastructure and its resources, such as storage, computation, and networking resources, and coordinates the instantiation of virtual guest machines on the hardware of the physical host machines. The VIM is part of MANO frameworks, such as those described in the remaining of this section. Examples of VIMs are OpenStack [189] and Kubernetes [190]:

- *OpenStack* is a cloud computing software platform capable of controlling a plethora of heterogeneous resources, such as compute, storage and networking resources. Among its very many features, it can act as a VIM, managing the network infrastructure, virtual machines, containers, unikernels, VNF services and applications.
- *Kubernetes* provides automatic deployment, scaling, and management of virtual machines, containers, unikernels, and their applications through a set of primitives. Kubernetes abstracts and represents the status of the system through a series of objects. These are persistent entities that describe the VNF or applications that are running on the Kubernetes-managed cluster, their available resources, and the policies on their expected behavior. In the past few years, a number of projects able to interact with Kubernetes have been launched to solve complex problems at the layers 2 and 3 of the protocol stack. These projects include Istio [191] and Network Service Mesh (NSM) [192]. Istio mesh services carry out traffic management, policy enforcement and telemetry collection tasks. NSM interfaces with Kubernetes APIs to support advanced use cases and facilitates the adoption of new cloud native paradigms. Specifically, it allows network managers to seamlessly perform tasks such as adding radio services, requesting network interfaces, or bridging multiple layer 2 services.

B. The Open Network Automation Platform

ONAP is an NFV framework developed as a project of the Linux Foundation, with AT&T, China Mobile, Vodafone, China Telecom, Orange, Verizon and Deutsche Telekom as main mobile operator supporters. ONAP is deployed in several commercial cellular networks, and vendors like Ericsson, Nokia, Huawei and ZTE, among others, provide ONAP support and integration in their products [193, 194]. Therefore, ONAP represents one of the most advanced software-based solutions for commercial cellular networks, actively maintained

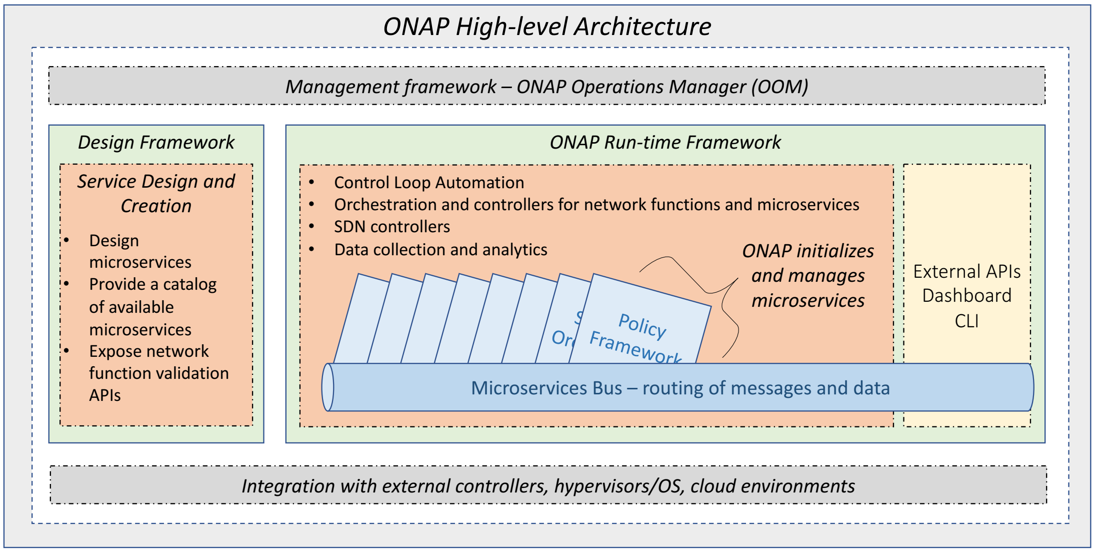

Fig. 12: High-level architecture of the ONAP framework.

and developed to meet production-level quality standards and satisfy new emerging requirements [195].

ONAP handles the design, creation, and life cycle management of a variety of network services. Network operators can use ONAP to orchestrate the physical and virtual infrastructure deployed in their networks, in a vendor-agnostic way [18]. In addition to common NFV orchestrator functionalities (e.g., automated policy-driven management of the virtualization infrastructure and of the network services), ONAP provides a design framework to model network applications and services as well as a framework for data analytics to monitor the services for healing and scaling. Additionally, ONAP provides a number of reference designs, i.e., *blueprints*. These can be used to deploy the ONAP architecture, depicted at a high-level in Figure 12, in specific markets or for specialized use cases (i.e., 5G networks or Voice over LTE deployments). They have been tested in combination with their typical hardware configurations.

The main components of the ONAP architecture [18], depicted in Figure 12, are: (i) The Management Framework; (ii) the Design Framework, and (iii) the Run-time Framework. The management framework, called ONAP Operations Manager (OOM), orchestrates and monitors the lifecycle of the ONAP components. The OOM leverages Kubernetes (Section VI-A), and Consul [196], which enables service control, discovery, configuration, and segmentation. Among its functionalities the most noteworthy are: (i) Component deployment, dependency manager, and configuration; (ii) real-time health monitoring; (iii) service clustering and scaling, and (iv) component upgrade, restart, and deletion.

The design framework allows to create network services with a declarative modeling language, which makes it possible to specify requirements and functionalities of each service. It allows to model resources, services, products and their management and control functions, through a set of common specifications and policies. Additionally, it includes service design and creation modules for the definition, simulation, and certification of systems assets, processes and policies. Finally, this module provides a database of existing services, and APIs for the validation of network functions.

ONAP run-time framework is made up of several software frameworks for most of its management and orchestration functionalities. In the run-time domain, a microservices bus allows communication and routing of messages and data among the different network functions initialized and managed by ONAP. The run-time framework dispatches and terminates microservices, using an automated control loop, and collects data and analytics from the platform. The run-time component exposes APIs, a dashboard and a command-line tools with a unified interface to control the network infrastructure. Finally, a southbound layer (gray box at the bottom of Figure 12) can be used for the integration with external controllers, operating systems and cloud environments, while northbound APIs are offered to OSS/BSS, big data, and other relevant services.

Integration with 5G networks*:* Besides representing a general framework for the management and orchestration of mobile networks, ONAP offers some key features that are relevant to 5G deployments. The main requirements that operators have identified are the need to support a hybrid infrastructure, with both physical and virtual appliances, edge automation, with the cloud geographically distributed in different edge locations, and real time analytics, which would enable closed-loop automation.

The Dublin release (June 2019) has introduced a first iteration on the implementation of a 5G blueprint. The most noteworthy feature is the joint discovery of virtual and physical network functions. As previously mentioned (sections II-A and III) the 5G RAN will not only deal with software components, but will also be in charge of real time signal and RF processing through physical hardware equipment. As such, ONAP also includes discovery and integration procedures to gain awareness of both these *virtual* and *physical network functions*, and to properly manage them. Additionally, the Dublin release includes preliminary support for network slicing. Preliminary evaluation studies for the feasibility of network slicing in ONAP are discussed in [197, 198]. Finally,

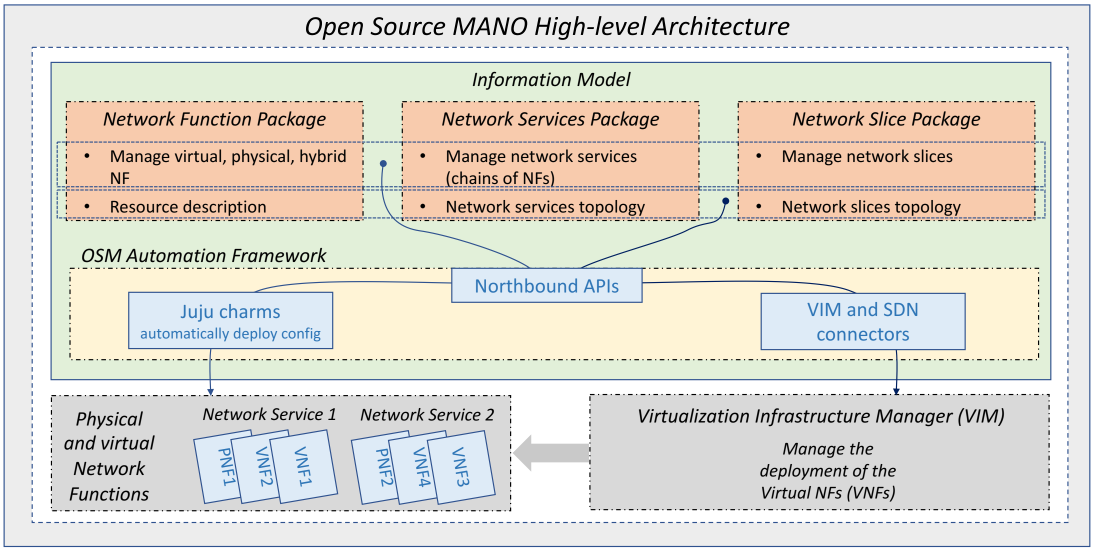

Fig. 13: High-level architecture of the OSM framework.

the ONAP 5G blueprint currently supports a dynamic configuration and optimization of 5G network parameters. This relies on a data collection platform, which transfers in a matter of minutes relevant Key Performance Indicators (KPIs) from the network edge to central processing facilities, and analyzes these data to automatically apply optimizations, or scale resources as needed.

The Frankfurt release (June 2020) builds upon the Dublin release by adding support for end-to-end network slicing and service orchestration. Additionally, it allows network designers to define control loops without having to wait for the next official ONAP release. One of the most noteworthy features introduced by the Frankfurt release is the harmonization effort toward O-RAN compatibility through the O1 and A1 interfaces. This effectively aims at defining the specifications for managing the elements of the 5G RAN, such as CU, DU, RU (Section V-A and Figure 8). Future releases will further integrate ONAP and O-RAN, making it easier for telecom operators to deploy an integrated O-RAN/ONAP solution [199].

C. Open Source NFV Management and Orchestration

Open Source NFV Management and Orchestration (OSM) is a MANO framework developed by a set of network providers, including Telefonica, BT and Telenor. The community also counts cloud and open source entities, such as Amazon and Canonical. Similar to ONAP, the OSM framework is well developed and deployed in major cellular networks.

The goals and general architecture of OSM are introduced in [171], and shown at a glance in Figure 4. Overall, the framework is an end-to-end network service orchestrator, tailored for deployment in mobile networks. Figure 13 describes the OSM architecture and its interactions with the network functions and VIM it manages, following the typical NFV orchestrator structure described in the introduction to this section. The main logical components of OSM are:

- *The Information Model:* It performs the modeling of network functions, services, and slices into templates called packages. This is enabled by a well-defined information model provided by the ETSI MANO framework [169]. Similarly to the design component of ONAP, this allows telecom operators to analyze the requirements of the network and model the resources that need to be deployed for functions, services, and slices.
- *The OSM Automation Framework:* It automates the life cycle of network services, from instantiation to scaling and, eventually, deletion. This is done by applying the information model to the actual deployed infrastructure, as shown in Figure 13, through a northbound interface that the automation framework exposes to the different modeling components. The 0-day, 1-day and 2-day configurations of the actual services and functions are done through Juju Charms [200] (i.e., tools to define, configure, and deploy services on cloud and bare-metal systems).

Similarly to ONAP, OSM has southbound and northbound APIs that can be exploited by other external services, such as other orchestrators and OSS/BSS, respectively [201].

Integration with 5G networks: OSM has published in December 2018 a 5G-ready release, which added the possibility of managing both virtual and physical network functions, network slicing, and a policy-based closed control loop, and extended the analytics and interface frameworks. Several 5G European projects have used and/or contributed to OSM. Metro-Haul focused on the design of an SDN-based optical transport infrastructure for 5G networks, and developed an OSM component for the management of the infrastructure in a distributed wide area network [202]. Similarly, 5G Tango has developed multiple components for OSM, including an emulator for the virtual infrastructure manager, and network slicing capabilities, while discussing and proposing possible extensions of the MANO concept for 5G into more advanced frameworks [168, 203]. The 5GCity and 5G-MEDIA projects have used OSM as NFV orchestrator for management frame-

TABLE VII: Capabilities of SDRs and their integration with RAN software.

SDR	TX/RX Channels	Frequency Range	Instantaneous Bandwidth (up to)	RAN Software	Target
bladeRF	1	[300 MHz, 3.8 GHz]	28 MHz	OAI, srsLTE	DAS node, small cell
bladeRF 2.0 micro	2	[47 MHz, 6 GHz]	56 MHz	OAI, srsLTE	DAS node, small cell
Iris	2	[50 MHz, 3.8 GHz]	56 MHz	OAI	DAS node, small cell, cell tower
LimeSDR	4 TX, 6 RX	[100 kHz, 3.8 GHz]	61.44 MHz	OAI, srsLTE	DAS node, small cell
USRP B205mini-i	1	[70 MHz, 6 GHz]	56 MHz	srsLTE	DAS node
USRP B210	2	[70 MHz, 6 GHz]	56 MHz	OAI, srsLTE	DAS node, small cell
USRP N310	4	[10 MHz, 6 GHz]	100 MHz	OAI	DAS node, small cell, cell tower, rooftop
USRP X310	up to 2 (daughterboards)	[DC, 6 GHz] (daughterboards)	160 MHz (daughterboards)	OAI, srsLTE	DAS node, small cell, cell tower

works of networks for smart cities and media distribution over a Content Distribution Network (CDN), respectively [204, 205]. Finally, 5G-TRANSFORMER has integrated OSM in its network slicing framework for the management of computing resources [206].

The author of [207] investigates how to integrate OSM and OAI-CN, to facilitate the deployment of fully-software-based solutions in testbeds and edge locations. Finally, [208] uses OSM to experiment with dynamic virtual network function placing in a 5G vehicular scenario.

D. Open Baton

Open Baton [209] is an open source project jointly developed by Fraunhofer FOKUS and TU Berlin aimed at providing a modular and reconfigurable framework for the orchestration of network services. The framework focuses on NFV management and is fully-compliant with the ETSI NFV MANO specification. Its source code is available online under Apache License v2.0 [210].

Open Baton provides a full-fledged ecosystem to instantiate and handle atomic VNFs, as well as to compose them to create more complex network services. The framework has been designed to operate over a virtualized infrastructure. For this reason, Open Baton features drivers to directly interface with most VIMs, with specific support for OpenStack [189] (see Section VI-A).

Besides VNF orchestration, Open Baton also provides support for multi-tenancy applications through network slicing and MEC [172]. Specifically, Open Baton features a Network Slicing Engine (NSE), a Java-based external software component that interacts with Open Baton via dedicated SDKs. The NSE allows network operators to specify QoS requirements for each network slice (e.g., minimum bandwidth for a target traffic class) in a clean and simple way via minimal JSON or YAML configuration files. Through Open Baton's VIM drivers, these configuration files are, then, dispatched to the VIM, which ultimately guarantees that each slice meets the set QoS requirements.

VII. Software-defined Radio Support for Open Source Radio Units

The open source software described throughout this article can be mostly executed on commodity hardware, except for the signal processing related to the physical layer, which generally runs on the Field Programmable Gate Arrays (FPGAs) of the SDRs. In this section, we discuss the main SDR solutions compatible with the software suites described in Section III. These platforms are pivotal in enabling researchers to deploy and experiment with end-to-end networks, even though they may not have access to carrier-grade hardware deployed by the major telecom operators.

A summary of the capabilities of each SDR is shown in Table VII. There we can find powerful SDRs that can act as rooftop base stations, such as the USRP N310, and cell towers, such as the USRP X310 or arrays of Iris SDRs. Smaller SDR models, such as USRPs B210, bladeRF/2.0 micro, and LimeSDR, instead, are powerful enough to operate as small cells, while, the ultra compact and lightweight USRP B205mini-i can act as a Distributed Antenna System (DAS) node. A description of the capabilities of each of these SDRs will be given in the following paragraphs.

USRP: The Universal Software Radio Peripheral (USRP) are SDR solutions produced by National Instruments/Ettus Research for designing, prototyping and testing wireless protocols and systems [81]:

- *USRP B210:* It is a full-duplex SDR with two transmit receive channels. It covers a frequency range from 70 MHz to 6 GHz with a real-time bandwidth of up to 56 MHz. It connects to the host computer through a USB 3.0 interface, and is compatible with OAI and srsLTE discussed in Section III.
- *USRP B205mini-i:* It is a full-duplex SDR with a frequency range from 70 MHz to 6 GHz, and an instantaneous bandwidth of up to 56 MHz. Similar to USRP B210, it connects to the host computer through a USB 3.0 interface. It is compatible with srsLTE.
- *USRP X310:* It is an SDR with two daughterboard slots, which enable up to two full-duplex transmit/receive chains. The covered frequency range and instantaneous bandwidth vary according to the specific daughterboard

TABLE VIII: 5G Testbeds.

Testbed	Technology available	5G Open Source Software	Framework	Scenario
AERPAW	5G and CR for UASs	under development		City-scale outdoor
Arena	5G, CR, massive MIMO	RAN & Core	N/A	Large-scale office
Colosseum	5G, CR	RAN & Core	O-RAN RIC	Large-scale network emulator
CORNET	5G, CR	RAN & Core	N/A	Large-scale indoor
COSMOS	5G, mmWave, CR, optical switching	RAN & Core	O-RAN components	Indoor, city-scale outdoor
Drexel Grid	5G, CR	RAN & Core	N/A	Large-scale indoor
FIT testbeds	5G, CR, IoT, NFV	RAN & Core	OSM	Large-scale indoor
IRIS	5G, CR, Wi-Fi, WiMAX, cloud-RAN, NFV, S-band	RAN & Core	N/A	Indoor
NITOS	5G, CR, Wi-Fi, WiMAX	RAN & Core	N/A	Large-scale indoor and outdoor, office
POWDER-RENEW	5G, CR, massive MIMO, Network Orchestration	RAN & Core	O-RAN RIC	Indoor, city-scale outdoor
5TONIC	5G NFV, network orchestration	N/A	OSM	Data center

model (from DC to 6 GHz, and up to 160 MHz). This USRP can connect to a host computer through a range of interfaces such as 1 Gigabit Ethernet, dual 10 Gigabit Ethernet, PCIe Express, and ExpressCard. It is compatible with both OAI and srsLTE;

- *USRP N310:* It is a full-duplex networked SDR with four transmit/receive chains. It covers the $[10\text{ MHz}, 6\text{ GHz}]$ frequency range with an instantaneous bandwidth of up to $100\,\text{MHz}$. It can be connected to a host computer through 1 Gigabit Ethernet, 10 Gigabit, or Xilinx Aurora over two SFP+ ports, and is compatible with the OAI software.

***bladeRF**:* The bladeRF denotes a series of full-duplex SDR devices produced by Nuand [98]. They are available with different form factors and FPGA chips, and are compatible with both OAI and srsLTE. They and connect to the host computer through a USB 3.0 interface.

- *bladeRF:* This SDRs comes in two different configurations, i.e., bladeRF x40 with a 40KLE Cyclone IV FPGA, and bladeRF x115 with a 115KLE Cyclone IV FPGA. Regardless of the specific FPGA chip, the bladeRF SDR has a single transmit/receive chain, which covers the $[300\,\text{MHz}, 3.8\,\text{GHz}]$ frequency range with up to $28\,\text{MHz}$ of instantaneous bandwidth.
- *bladeRF 2.0 micro:* This model is equipped with two transmit/receive chains and supports 2×2 MIMO operations. It is available with different FPGA chips, i.e., 49KLE Cyclone V FPGA chip (bladeRF xA4), and 301KLE Cyclone V FPGA chip (bladeRF xA9). It covers the $[47\text{ MHz}, 6\text{ GHz}]$ frequency range with an instantaneous bandwidth of 56 MHz.

***LimeSDR**:* This SDR is produced by Lime Microsystems [97], has four transmit and six receive chains, and supports 2×2 MIMO operations. It covers the $[100\,\text{kHz}, 3.8\,\text{GHz}]$ frequency range with an instantaneous bandwidth of up to 61.44 MHz. The LimeSDR is compatible with both OAI and srsLTE. Other versions of this SDR include the LimeSDR Mini, which has two channels instead of four, and the LimeSDR QPCIe, which enables 4×4 MIMO configurations instead of the 2×2 of the standard model. Support for these LimeSDR models has not been explicitly reported by either OAI or srsLTE.

***Iris**:* This is a networked SDR device with two transmit/receive chains, produced by Skylark Wireless [211]. It works in the $[50\,\text{MHz}, 3.8\,\text{GHz}]$ frequency range and supports an instantaneous bandwidth of up to 56 MHz. This SDR connects to the host computer through a 1 Gigabit Ethernet interface, and is compatible with OAI. It can be combined with additional hardware platforms provided by Skylark to boost its performance, while multiple Iris SDRs can be grouped in arrays to enable massive MIMO operations (see Argos [212], and the POWDER-RENEW PAWR testbed [213, 214] described in Section VIII).

VIII. Testbeds

In this section we describe a number of testbeds that can be used to instantiate softwarized 5G networks by leveraging the open source utilities, frameworks and hardware components described in this article. An overview of the capabilities of each testbed is given in Table VIII.

***Platforms for Advanced Wireless Research**:* The objective of the NSF-funded Platforms for Advanced Wireless Research (PAWR) program is to enable experimental investigation of new wireless devices, communication techniques, networks, systems, and services in real wireless environments through several heterogeneous city-scale testbeds [25]. The following are the PAWR platforms awarded so far and that are being built.

- *POWDER-RENEW:* The combination of Platform for Open Wireless Data-driven Experimental Research [213, 215] and Reconfigurable Eco-system for Next-generation End-to-end Wireless [214, 216] provides a testbed, namely POWDER-RENEW, which covers an area of 6 km^2 of the University of Utah campus in Salt Lake City, UT. Its objective is to foster experimental research

for a range of heterogeneous wireless technologies, including 5G, RAN, network orchestration, and massive MIMO technologies. POWDER-RENEW is equipped with cutting-edge compute, storage, and cloud resources, as well as state-of-the-art SDRs.
Besides allowing users to install their own software suites, the testbed offers a series of ready-to-use "profiles," which are instantiated on its bare-metal machines. These include open source RAN software, e.g., OAI and srsLTE, coupled with different EPC solutions, as well as components of the frameworks previously discussed, including the O-RAN RIC. Finally, the POWDER-RENEW platform has been used to demonstrate automated optimization of 5G networks in [92].

- *COSMOS:* The Cloud Enhanced Open Software Defined Mobile Wireless Testbed for City-Scale Deployment (COSMOS) [217–219] is being deployed in the densely-populated neighborhood of West Harlem, New York City, NY. Upon completion, it will cover an area of $2.59\,\mathrm{km}^2$. This testbed focuses on providing ultra-high-bandwidth and low-latency wireless communications, and it will have edge-computing capabilities. Among others, COSMOS will allow researchers to experiment with mmWave and optical switching technologies.
COSMOS includes the Open-Access Research Testbed for Next-Generation Wireless Networks (ORBIT) [220], a platform with a number of USRPs X310, compatible with the open source RAN solutions described in Section III. The platform served as Open Test and Integration Center during the O-RAN plugfest, a proof of concept for demonstrating the potentials of multi-vendor interoperability of cellular networks [221].
- *AERPAW:* The Aerial Experimentation and Research Platform for Advanced Wireless (AERPAW) [222–224] will be the first-ever wireless platform to allow large-scale Unmanned Aerial System (UAS) experimentation for 5G technologies and beyond. AERPAW will be deployed in the North Carolina Research Triangle and its features will include flying aerial base stations to provide cellular connectivity to ground users.

Additional PAWR platforms are scheduled to be selected in 2021, possibly providing testbeds for experimental research on rural broadband communication and networking. As publicly funded testbeds, the PAWR platforms will be accessible to the wireless research community for experimental use. As such, each platform implements the fundamental requirements that will ensure reproducibility of experiments, interoperability with other platforms, programmability, open access to the research community, and sustainability.

Colosseum is the world's most powerful wireless network emulator [225]. It is housed at Northeastern University Innovation Campus in Burlington, MA. It is composed of 21 server racks with 256 USRP X310 SDRs, half of which are controllable by experimenters, while the other half is allocated to Colosseum Massive Channel Emulator (MCHEM). Through MCHEM scenarios, researchers can seamlessly emulate entire virtual worlds with up to 65,536 concurrent wireless channels with realistic characteristics, such as path loss and fading. This way, Colosseum assures the ultimate reproducibility of experiments in the sub-6 GHz spectrum band.

At its core, Colosseum is a full-fledged data-center with state-of-the-art hardware, including more than 900 TB of networked storage, over 170 high-performance servers, 320 FPGAs, and full mesh high-speed connections. Colosseum allows users to install and instantiate the open source RAN and core network solutions discussed in sections III and IV, as well as components of the frameworks detailed in Section V (e.g., the O-RAN RIC). After having being used in the DARPA Spectrum Collaboration Challenge (SC2), Colosseum will be available to the research community shortly.

Arena is an indoor testbed composed of 24 SDR USRPs (16 USRPs N210 and 8 USRPs X310) stacked up in a radio rack, and controlled by 12 Dell PowerEdge R340 high-performance machines in a server rack [23]. Servers and SDRs are connected through dual 10 Gigabit Ethernet connections to guarantee rapid and low-latency radio control and communication. The USRPs are connected to a grid of 8×8 antennas hung off the ceiling of a $208.1\,\mathrm{m}^2$ dynamic indoor office space through same-length cables that guarantee equal signal delays across the whole testbed. Moreover, Arena SDRs are fully synchronized through 4 Octoclock clock distributors to enable massive MIMO applications, among others.

The Arena testbed allows researchers to experimentally evaluate wireless protocols and solutions for indoor 5G deployments in an office-like environment. For instance, Arena can be used to evaluate Wi-Fi and Cognitive Radio (CR) solutions, to generate communication traces and data sets, and to evaluate the performance of standard compliant cellular networks through the OAI and srsLTE protocol stacks (Section III). Arena has been used to demonstrate future cellular networks capabilities [226], 5G RAN optimization [92] RAN slicing [6, 108].

5TONIC includes data center and equipment for 5G virtual network experimentation [227]. It is composed of a NFV infrastructure with high-performance servers and workstations running network orchestration and virtualization functions and a number of SDR platforms and devices. 5TONIC allows users to run complex NFV and orchestrations frameworks such as OSM. 5TONIC has been used for NFV MANO [228] and mmWave applications [229].

FED4FIRE+ is a Horizon 2020 [230] project to foster experimentally-driven research in the future Internet ecosystem [231]. Among others, it includes a number of *federated* testbeds for wireless, 5G, IoT, cloud, and big data applications. Below, we describe the testbeds of the FED4FIRE+ project targeting open source cellular networks research and compatible with the OAI and srsLTE RAN software tools (Section III).

- *NITOS* is a Future Internet Facility composed of an outdoor, an indoor, and an office testbed with both SDRs and commercial nodes [232]. The outdoor testbed comprises nodes with Wi-Fi, WiMAX, and LTE capabilities, while the indoor and the office testbeds are made up of of Icarus Wi-Fi nodes [233] deployed in an isolated environment. NITOS has been used for MANO [234], 5G distributed spectral awareness [235], and MEC applications [236, 237], among others.

- The *IRIS* testbed focuses on Cloud-RAN, NFV, and SDN experimental research [238]. The testbed includes a number of ceiling-mounted SDR devices supporting Wi-Fi, WiMAX, and 4G/5G technologies, as well as S-band transceivers.

COgnitive Radio NETwork (CORNET) is a testbed of 48 SDR nodes deployed in a four-story building in the Virginia Tech campus (Blacksburg, VA) that enables experiments on dynamic spectrum access and CR research [239]. CORNET allows users to perform 5G experimental research by leveraging open source software, such as OAI and srsLTE, or by emulating cellular signals through COTS equipment. Among other applications, CORNET has been used to evaluate link adaptation in cellular systems [240].

Future Internet of Things (FIT) is a French project for large-scale testbeds for wireless communications [241]. It includes: (i) FIT Wireless, which targets indoor Wi-Fi, CR, and 5G applications (through OAI, for instance); (ii) FIT IoT-Lab, which concerns IoT-related experimentation, and (iii) FIT Cloud, which supports the other two by enabling SDN and NFV research through OSM, among other frameworks.

Drexel Grid is a testbed made up of 24 SDR devices (20 USRPs N210 and 4 USRPs X310) hung off the ceiling of a dedicated indoor room to evaluate diverse 5G and CR wireless technologies [242, 243]. The USRPs X310 of the testbed can be used with open source RAN software such as OAI and srsLTE (Section III). Additionally, this testbed includes a channel emulator with simulated nodes to evaluate wireless systems in a controlled and repeatable environment.

IX. Softwarized 5G: Limitations and Road Ahead

Our work aimed at investigating how the "softwarization of everything"—that is pervasive to current trends in computing, communication and networking—got its way into cellular networks, and in particular how it has not only revolutionized their fourth generation, but has also established a radically new way, both technical and commercial, to usher in the 5G era successfully.

Our overview has focused on the most recent advances in the open source and reprogrammable 5G ecosystem. We have listed and discussed a variety of heterogeneous, yet modular software and hardware components. In particular, we have illustrated how their expected evolution is key to transitioning from the traditional black box approach of cellular network management to those white box principles that will bring both research and industry communities to swift innovation, shorter time-to-market and overall higher customer satisfaction.

By way of conclusion, we finally intend to point out that despite operators, vendors and scientists are paying considerable attention to the new software-defined technologies described in this article, these solutions are not ready for prime time on commercial 5G networks just yet. Indeed, the road to celebrate this marriage needs overcoming a few show stoppers, which we describe below.

- **Keep pace with the standards.** The cellular network community faces constant pressure to keep up with the specifications/technologies being introduced by new communication, networking and even programming standards. A notable example are the NR and mmWave communication technologies introduced as 5G enablers by 3GPP and are currently being deployed in *closed source* commercial networks. The RAN software libraries described in Section III have not yet completed the development of the support for NR, as such task requires a considerable effort in terms of coding and testing. In this domain, open source network simulators have, so far, provided a more controlled development environment for the development and assessment of 5G solutions [244–247]. The testing of real-world 5G software, especially for mmWaves, is indeed extremely complex due to the lack of accessible open hardware for the software to run, which precludes testing important components, such as beam management. The platforms described in Section VII are optimized for carriers below 6 GHz, even though early prototypes of open mmWaves boards are currently being developed [248–251]. Similarly, most of the testbeds described in Section VIII focus on sub-6 GHz deployments, with only a few (e.g., COSMOS) considering an extension to mmWaves. The road ahead for the development of high-band 5G software-defined solutions, therefore, lies in a more concerted, joint software development effort, and in hardware platforms that can keep up with the requirements of the software community. Furthermore, the current lack of a mature code base for 3GPP NR RAN software libraries hinders the development of more advanced features even in the sub-6 GHz spectrum, such as URLLC support (e.g., with mini slots [44]) and multi-connectivity [31].
- **Latency and scalability issues.** The scalability of an SDR-based system, both in terms of processing and computing requirements, depends on the number of signal processing operations, which are generally proportional to the available bandwidth [252]. As the next generation wireless systems will deal with larger and larger bandwidths for higher data rates [31], the implementation of the radio stack and its software processing chains will have to be extremely efficient, robust, and count on powerful and reliable hardware. Moreover, considering the tight latency and throughput requirements of many 5G use cases, the integration of software and hardware needs to be seamless, to deliver the best possible performance. In this regard, although virtualized solutions add unprecedented flexibility to the network, they also come with new challenges. Specifically, these solutions rely upon resource sharing (which limits and regulates resource utilization among different processes [253]), and virtualization requires additional interactions between the virtualized and bare-metal environments [62]. Together, these aspects introduce additional latency which might not be tolerable for many 5G application and services.
- **Limited contributions for RAN open source software.** The same large telecom operators and vendors driving the

development of open source CN and MANO frameworks are not showing the same level of attention to RAN-related projects. RAN efforts have indeed mostly come from academia or from smaller companies, with limited manpower and resources. As some sophisticated digital signal processing and implementations of the lower layers of the RAN stack are often source of intellectual property and product-bearing revenues for telecom businesses, major vendors and operators are not encouraged to release their solutions as open source. Recognizing this limitation, the OpenAirInterface Software Alliance has licensed the OAI RAN implementation with a permissive license, which allows contributors to retain intellectual property claims (see Section III-A). Additionally, the O-RAN Alliance is moving encouraging first steps toward an openly softwarized RAN (see Section V-A), even though the current development efforts do not include also an open source software for the radio front-ends. Therefore, the wireless community should aim at increasing the support toward the development of complete and open RAN and radio software libraries, increasing the number of active contributors to the currently available open source RAN projects.

- **Lack of robust, deployable, and well-documented software.** As of now, most of the frameworks and libraries described in Sections III-VI cannot be used in actual networks, as their open source component is either incomplete, requires additional integration and development for actual deployment, or lacks robustness. Moreover, to reach the quality of commercial solutions, the open source community should aim at delivering well-documented, easy-to-deploy, and robust software, specifying all dependencies and additional software components that guarantee the correct and efficient functioning of the system. For example, the container-driven development model as used in cloud-native computing could be adopted to simplify and expedite the software deployment process.
- **Need for secure open source software.** Heightened attention to software development following best practices for robustness and security is sorely required [254], to guarantee privacy, integrity, and security to the end users of softwarized networks. Openness already facilitates useful scrutiny of the code. Audits and reviews from the open source community can help prevent bugs and/or security holes, whose existence needs to be responsibly disclosed to the project maintainers [255]. Appropriate security, especially "by design," however, is still lacking. The exposure of APIs to third party vendors (e.g., for the RIC apps), for instance, could introduce new vulnerabilities in the network, in case the APIs are not properly securely designed, and contain weaknesses that can be exploited by attackers. It is clear that the security of the open source software that will be eventually deployed in 5G and beyond systems must be a key concern for the developers and telecom ecosystem. The wireless community, thus, should follow the best practices developed over the years by other open source communities (e.g., the Linux kernel), that constantly make it possible to tighten the security of open source products [256].

All these road blocks are currently preventing, or considerably slowing down, the widespread and painless application of several of the softwarized solutions presented in this article. It is now the task of the wireless research and development community to transform these challenges into the opportunity to innovate further in the direction of truly realizing open, programmable, and virtualized cellular networks.

References

[1] S. Parkvall, E. Dahlman, A. Furuskar, and M. Frenne, "NR: The New 5G Radio Access Technology," *IEEE Communications Standards Magazine*, vol. 1, no. 4, pp. 24–30, December 2017.

[2] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, "Five Disruptive Technology Directions for 5G," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 74–80, May 2014.

[3] D. Martín-Sacristán, J. F. Monserrat, J. Cabrejas-Penuelas, D. Calabuig, S. Garrigas, and N. Cardona, "On the Way Towards Fourth-generation Mobile: 3GPP LTE and LTE-Advanced," *Springer EURASIP Journal on Wireless Communications and Networking*, vol. 2009, no. 1, p. 354089, August 2009.

[4] O-RAN Alliance White Paper, "O-RAN: Towards an Open and Smart RAN," https://www.o-ran.org/resources, 2018.

[5] S. D'Oro, F. Restuccia, and T. Melodia, "The Slice Is Served: Enforcing Radio Access Network Slicing in Virtualized 5G Systems," in *Proc. of IEEE Intl. Conf. on Computer Communications (INFOCOM)*, Paris, France, May 2019.

[6] S. D'Oro, F. Restuccia, and T. Melodia, "Toward Operator-to-Waveform 5G Radio Access Network Slicing," *IEEE Communications Magazine*, vol. 58, no. 4, pp. 18–23, April 2020.

[7] M. Zambianco and G. Verticale, "Interference Minimization in 5G Physical-Layer Network Slicing," *IEEE Transactions on Communications*, March 2020.

[8] N. Bhushan, J. Li, D. Malladi, R. Gilmore, D. Brenner, A. Damnjanovic, R. T. Sukhavasi, C. Patel, and S. Geirhofer, "Network Densification: The Dominant Theme for Wireless Evolution into 5G," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 82–89, 2014.

[9] J. Haavisto, M. Arif, L. Lovén, T. Leppänen, and J. Riekki, "Open-source RANs in Practice: an Over-The-Air Deployment for 5G MEC," in *Proc. of IEEE European Conf. on Networks and Communications (EuCNC)*, Valencia, Spain, June 2019.

[10] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O'Connor, P. Radoslavov, W. Snow, and G. Parulkar, "ONOS: Towards an Open, Distributed SDN OS," in *Proc. of ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN)*, Chicago, Illinois, USA, 2014.

[11] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 38, no. 2, pp. 69—-74, March 2008.

[12] N. M. M. K. Chowdhury and R. Boutaba, "A Survey of Network Virtualization," *Computer Networks*, vol. 54, no. 5, pp. 862–876, April 2010.

[13] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, November 2017.

[14] L. Zhang, M. Xiao, G. Wu, M. Alam, Y. Liang, and S. Li, "A Survey of Advanced Techniques for Spectrum Sharing in 5G Networks," *IEEE Wireless Communications*, vol. 24, no. 5, pp. 44–51, October 2017.

[15] A. A. Barakabitze, A. Ahmad, R. Mijumbi, and A. Hines, "5G Network Slicing Using SDN and NFV: A Survey of Taxonomy, Architectures and Future Challenges," *Computer Networks*, vol. 167, p. 106984, February 2020.

[16] S. Sun, M. Kadoch, L. Gong, and B. Rong, "Integrating Network Function Virtualization with SDR and SDN for 4G/5G Networks," *IEEE Network*, vol. 29, no. 3, pp. 54–59, June 2015.

[17] F. Kaltenberger, A. P. Silva, A. Gosain, L. Wang, and T.-T. Nguyen, "OpenAirInterface: Democratizing Innovation in the 5G Era," *Computer Networks*, no. 107284, May 2020.

[18] ONAP, "Architecture Overview," https://www.onap.org/wp-content/uploads/sites/20/2019/07/ONAP_CaseSolution_Architecture_062519.pdf, 2019.

[19] GNU Radio Project. GNU Radio. http://www.gnuradio.org. Accessed July 2020.
[20] I. Gomez-Miguelez, A. Garcia-Saavedra, P. Sutton, P. Serrano, C. Cano, and D. Leith, "srsLTE: An Open-source Platform for LTE Evolution and Experimentation," in *Proc. of ACM Intl. Workshop on Wireless Network Testbeds, Experimental evaluation & CHaracterization (WiNTECH)*, New York City, NY, USA, October 2016.
[21] O-RAN. https://o-ran.org. Accessed July 2020.
[22] Open Networking Foundation. Open Network Automation Platform (ONAP). https://onap.org. Accessed July 2020.
[23] L. Bertizzolo, L. Bonati, E. Demirors, A. Al-Shawabka, S. D'Oro, F. Restuccia, and T. Melodia, "Arena: A 64-antenna SDR-based Ceiling Grid Testing Platform for Sub-6 GHz 5G-and-Beyond Radio Spectrum Research," *Computer Networks (COMNET)*, 2020.
[24] Rakuten, "How Elegant Software Can Make 5G Networks More Resilient," https://rakuten.today/blog/5g-network-reliability-lightreading.html, 2020, accessed July 2020.
[25] Platforms for Advanced Wireless Research (PAWR). https://www.advancedwireless.org. Accessed July 2020.
[26] 5G Americas Whitepaper, "The Status of Open Source for 5G," https://www.5gamericas.org/wp-content/uploads/2019/07/5G_Americas_White_Paper_The_Status_of_Open_Source_for_5G_Feb_2019.pdf, 2019, accessed August 2020.
[27] T. Chen, M. Matinmikko, X. Chen, X. Zhou, and P. Ahokangas, "Software Defined Mobile Networks: Concept, Survey, and Research Directions," *IEEE Communications Magazine*, vol. 53, no. 11, pp. 126–133, November 2015.
[28] M. Condoluci and T. Mahmoodi, "Softwarization and Virtualization in 5G Mobile Networks: Benefits, Trends and Challenges," *Computer Networks*, vol. 146, pp. 65–84, December 2018.
[29] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies, and Solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2429–2453, August 2018.
[30] D. Thembelihle, M. Rossi, and D. Munaretto, "Softwarization of Mobile Network Functions Towards Agile and Energy Efficient 5G Architectures: A Survey," *Wireless Communications and Mobile Computing*, vol. 2017, pp. 1–21, November 2017.
[31] 3GPP, "NR and NG-RAN Overall Description," TS 38.300, V15.0.0, 2018.
[32] ——, "Study on Scenarios and Requirements for Next Generation Access Technologies," TS 38.913, V14.1.0, 2018.
[33] ATIS - on behalf of 3GPP, "3GPP 5G Candidate For Inclusion In IMT-2020: Submission 1 (SRIT)," ITU Document 5D/1216-E, June 2019.
[34] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 36.300, 04 2017, version 14.2.0. [Online]. Available: http://www.3gpp.org/DynaReport/36300.htm
[35] ——, "Evolved Universal Terrestrial Radio Access (E-UTRA); Packet Data Convergence Protocol (PDCP) Specification," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 36.323, 03 2017, version 14.2.0. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2439
[36] ——, "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Link Control (RLC) Protocol Specification," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 36.322, 03 2017, version 14.0.0. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2438
[37] ——, "Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) Protocol Specification," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 36.321, 04 2017, version 14.2.1. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2437
[38] ——, "Evolved Universal Terrestrial Radio Access (E-UTRA); LTE Physical Layer; General Description," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 36.201, 03 2017, version 14.1.0. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2424
[39] ——, "Network Architecture," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 23.002, 3 2018, version 15.0.0. [Online]. Available: http://www.3gpp.org/DynaReport/23002.htm
[40] S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter-Wave Cellular Wireless Networks: Potentials and Challenges," *Proceedings of the IEEE*, vol. 102, no. 3, pp. 366–385, March 2014.
[41] M. Giordani, M. Polese, A. Roy, D. Castor, and M. Zorzi, "A Tutorial on Beam Management for 3GPP NR at mmWave Frequencies," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 173–196, February 2019.
[42] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA) and NR; Service Data Adaptation Protocol (SDAP) Specification," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 37.324, 03 2018, version 15.1.0. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3282
[43] ——, "NR; Physical Layer; General Description," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.201, 01 2018, version 15.0.0. [Online]. Available: http://www.3gpp.org/DynaReport/38201.htm
[44] ——, "NR; Medium Access Control (MAC) Protocol Specification," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.321, 01 2018, version 15.0.0. [Online]. Available: http://www.3gpp.org/DynaReport/38321.htm
[45] ——, "NR; Radio Link Control (RLC) Protocol Specification," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.322, 01 2018, version 15.0.0. [Online]. Available: http://www.3gpp.org/DynaReport/38322.htm
[46] ——, "NR; Packet Data Convergence Protocol (PDCP) Specification," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.323, 01 2018, version 15.0.0. [Online]. Available: http://www.3gpp.org/DynaReport/38323.htm
[47] ——, "NR; Radio Resource Control (RRC); Protocol Specification," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.331, 01 2018, version 15.0.0. [Online]. Available: http://www.3gpp.org/DynaReport/38331.htm
[48] ——, "Study on CU-DU Lower Layer Split for NR," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 38.816, 01 2018, version 15.0.0. [Online]. Available: http://www.3gpp.org/DynaReport/38816.htm
[49] D. Brake, "A U.S. National Strategy for 5G and Future Wireless Innovation," https://itif.org/publications/2020/04/27/us-national-strategy-5g-and-future-wireless-innovation, April 2020.
[50] 3GPP, "System Architecture for the 5G System (5GS)," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 23.501, 3 2020, version 16.4.0. [Online]. Available: http://www.3gpp.org/DynaReport/23501.htm
[51] ——, "Study on Enhancements to the Service-Based Architecture," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 23.742, 12 2018, version 16.0.0. [Online]. Available: http://www.3gpp.org/DynaReport/23742.htm
[52] J. Kim, D. Kim, and S. Choi, "3GPP SA2 Architecture and Functions for 5G Mobile Communication System," *ICT Express*, vol. 3, no. 1, pp. 1–8, April 2017.
[53] L. Peterson, A. Al-Shabibi, T. Anshutz, S. Baker, A. Bavier, S. Das, J. Hart, G. Palukar, and W. Snow, "Central Office Re-architected as a Data Center," *IEEE Communications Magazine*, vol. 54, no. 10, pp. 96–101, October 2016.
[54] ONF White Paper, "Aether - Enterprise-5G/LTE-Edge-Cloud-as-a-Service," https://www.opennetworking.org/wp-content/uploads/2020/02/Aether-white-paper.pdf, February 2020.
[55] ——, "ONF's Software-Defined RAN Platform Consistent with the O-RAN Architecture," https://www.opennetworking.org/wp-content/uploads/2020/03/SD-RAN-White-Paper.pdf, February 2020.
[56] Open Platform for NFV (OPNFV). https://www.opnfv.org. Accessed July 2020.
[57] S. D'Oro, F. Restuccia, T. Melodia, and S. Palazzo, "Low-Complexity Distributed Radio Access Network Slicing: Algorithms and Experimental Results," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2815–2828, December 2018.
[58] P. Rost, C. Mannweiler, D. S. Michalopoulos, C. Sartori, V. Sciancalepore, N. Sastry, O. Holland, S. Tayade, B. Han, D. Bega, D. Aziz, and H. Bakker, "Network Slicing to Enable Scalability and Flexibility in 5G Mobile Networks," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 72–79, May 2017.
[59] TM Forum. https://www.tmforum.org. Accessed July 2020.
[60] ONAP, "ONAP 5G Blueprint Overview," 2019. [Online]. Available: https://www.onap.org/wp-content/uploads/sites/20/2019/07/ONAP_CaseSolution_5G_062519.pdf
[61] OpenSlice Repositories. https://github.com/openslice. Accessed July 2020.
[62] Z. Xiang, F. Gabriel, E. Urbano, G. T. Nguyen, M. Reisslein, and

F. H. P. Fitzek, "Reducing Latency in Virtual Machines: Enabling Tactile Internet for Human-Machine Co-Working," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 5, pp. 1098–1116, May 2019.

[63] F. Giust, V. Sciancalepore, D. Sabella, M. C. Filippou, S. Mangiante, W. Featherstone, and D. Munaretto, "Multi-Access Edge Computing: The Driver Behind the Wheel of 5G-Connected Cars," *IEEE Communications Standards Magazine*, vol. 2, no. 3, pp. 66–73, March 2018.

[64] A. Kropp, R. Schmoll, G. T. Nguyen, and F. H. P. Fitzek, "Demonstration of a 5G Multi-access Edge Cloud Enabled Smart Sorting Machine for Industry 4.0," in *Proc. of IEEE Annual Consumer Communications Networking Conf. (CCNC)*, Las Vegas, NV, USA, 2019.

[65] ETSI, "Multi-access Edge Computing (MEC); Framework and Reference Architecture," ETSI GS MEC 003 V2.1.1, https://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/02.01.01_60/gs_MEC003v020101p.pdf, 2019, accessed July 2020.

[66] S. Kekki, W. Featherstone, Y. Fang, P. Kuure, A. Li, A. Ranjan, D. Purkayastha, F. Jiangping, D. Frydman, G. Verin, K. W. Wen, K. Kim, R. Arora, A. Odgers, L. M. Contreras, and S. Scarpina, "MEC in 5G networks," ETSI White Paper No. 28, https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp28_mec_in_5G_FINAL.pdf, June 2018, accessed July 2020.

[67] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1657–1681, Third quarter 2017.

[68] R. Bruschi, R. Bolla, F. Davoli, A. Zafeiropoulos, and P. Gouvas, "Mobile Edge Vertical Computing over 5G Network Sliced Infrastructures: An Insight into Integration Approaches," *IEEE Communications Magazine*, vol. 57, no. 7, pp. 78–84, July 2019.

[69] C. Jiang, H. Zhang, Y. Ren, Z. Han, K. Chen, and L. Hanzo, "Machine Learning Paradigms for Next-Generation Wireless Networks," *IEEE Wireless Communications*, vol. 24, no. 2, pp. 98–105, April 2017.

[70] OpenAirInterface Software Alliance. OpenAirInterface (OAI). https://openairinterface.org. Accessed July 2020.

[71] Software Radio Systems. srsLTE. https://srslte.com. Accessed July 2020.

[72] Radisys. https://radisys.com. Accessed July 2020.

[73] I. Alawe, A. Ksentini, Y. Hadjadj-Aoul, and P. Bertin, "Improving Traffic Forecasting for 5G Core Network Scalability: A Machine Learning Approach," *IEEE Network*, vol. 32, no. 6, pp. 42–49, November 2018.

[74] N. Strodthoff, B. Göktepe, T. Schierl, C. Hellge, and W. Samek, "Enhanced Machine Learning Techniques for Early HARQ Feedback Prediction in 5G," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 11, pp. 2573–2587, November 2019.

[75] A. Asadi, S. Müller, G. H. Sim, A. Klein, and M. Hollick, "FML: Fast Machine Learning for 5G mmWave Vehicular Communications," in *Proc. of IEEE Intl. Conf. on Computer Communications (INFOCOM)*, Honolulu, HI, USA, April 2018.

[76] C. Zhang, P. Patras, and H. Haddadi, "Deep Learning in Mobile and Wireless Networking: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2224–2287, March 2019.

[77] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y. Liang, and D. I. Kim, "Applications of Deep Reinforcement Learning in Communications and Networking: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133–3174, May 2019.

[78] M. Polese, R. Jana, V. Kounev, K. Zhang, S. Deb, and M. Zorzi, "Machine Learning at the Edge: A Data-Driven Architecture with Applications to 5G Cellular Networks," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2020.

[79] N. Nikaein, M. K. Marina, S. Manickam, A. Dawson, R. Knopp, and C. Bonnet, "OpenAirInterface: A Flexible Platform for 5G Research," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 33–38, October 2014.

[80] OpenAirInterface (OAI) Public License. https://www.openairinterface.org/?page_id=698. Accessed July 2020.

[81] Ettus Research. Universal Software Radio Peripheral (USRP). https://www.ettus.com/products. Accessed July 2020.

[82] F. Kaltenberger, G. de Souza, R. Knopp, and H. Wang, "The OpenAirInterface 5G New Radio Implementation: Current Status and Roadmap," in *Proc. of ITG Workshop on Smart Antennas (WSA)*, Vienna, Austria, April 2019.

[83] F. Kaltenberger, X. Jiang, and R. Knopp, "From Massive MIMO to C-RAN: The OpenAirInterface 5G Testbed," in *Asilomar Conf. on Signals, Systems, and Computers (ACSSC)*, Pacific Grove, CA, USA, October 2017.

[84] X. Foukas, M. Marina, and K. Kontovasilis, "Orion: RAN Slicing for a Flexible and Cost-Effective Multi-Service Mobile Network Architecture," in *Proc. of ACM Intl. Conf. on Mobile Computing and Networking (MobiCom)*, Snowbird, UT, USA, October 2017.

[85] X. Foukas, N. Nikaein, M. M. Kassem, M. K. Marina, and K. Kontovasilis, "FlexRAN: A Flexible and Programmable Platform for Software-Defined Radio Access Networks," in *Proc. of ACM SIGCOMM Conf. on emerging Networking EXperiments and Technologies (CoNEXT)*, Irvine, CA, USA, December 2016.

[86] Q. Liu, T. Han, and N. Ansari, "Learning-Assisted Secure End-to-End Network Slicing for Cyber-Physical Systems," *arXiv preprint arXiv:1910.13537 [cs.NI]*, October 2019.

[87] F. D'Alterio, L. Ferranti, L. Bonati, F. Cuomo, and T. Melodia, "Quality Aware Aerial-to-Ground 5G Cells through Open-Source Software," in *Proc. of IEEE Global Communications Conf. (GLOBECOM)*, Waikoloa, HI, USA, December 2019.

[88] Fujitsu demonstrates the power of OpenAirInterface. (2019, March) https://www.openairinterface.org/?news=fujitsu-demonstrates-the-power-of-openairinterface.

[89] WindyCitySDR. http://www.windycitysdr.com/home. Accessed July 2020.

[90] InterDigital MWC18: 5G Air Interface. (2018, February) https://www.interdigital.com/presentations/mwc18-5g-air-interface.

[91] SYRTEM. http://www.syrtem.com. Accessed July 2020.

[92] L. Bonati, S. D'Oro, L. Bertizzolo, E. Demirors, Z. Guan, S. Basagni, and T. Melodia, "CellOS: Zero-touch Softwarized Open Cellular Networks," *Computer Networks*, vol. 180, pp. 1–13, June 2020.

[93] A. Puschmann, P. Sutton, and I. Gomez, "Implementing NB-IoT in Software - Experiences Using the srsLTE Library," *arXiv preprint arXiv:1705.03529 [cs.NI]*, May 2017.

[94] GNU Affero General Public License Version 3. https://www.gnu.org/licenses/agpl-3.0.en.html. Accessed July 2020.

[95] 3GPP, "3G Security; Cryptographic Algorithm Requirements," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 33.105, 6 2018, version 15.0.0. [Online]. Available: http://www.3gpp.org/DynaReport/33105.htm

[96] ——, "3G Security; Specification of the MILENAGE Algorithm Set: An Example Algorithm Set for the 3GPP Authentication and Key Generation Functions f1, f1*, f2, f3, f4, f5 and f5*; Document 2: Algorithm Specification," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 35.206, 10 2018, version 15.0.0. [Online]. Available: http://www.3gpp.org/DynaReport/35206.htm

[97] Lime Microsystems. LimeSDR. https://limemicro.com/products/boards/. Accessed July 2020.

[98] Nuand. bladeRF. https://www.nuand.com. Accessed July 2020.

[99] N. Bui and J. Widmer, "OWL: A Reliable Online Watcher for LTE Control Channel Measurements," in *Proc. of ACM Workshop on All Things Cellular: Operations, Applications and Challenges (ATC)*, New York City, NY, USA, October 2016.

[100] F. Meneghello, M. Rossi, and N. Bui, "Smartphone Identification via Passive Traffic Fingerprinting: A Sequence-to-Sequence Learning Approach," *IEEE Network*, vol. 34, no. 2, pp. 112–120, February 2020.

[101] H. D. Trinh, A. F. Gambin, L. Giupponi, M. Rossi, and P. Dini, "Mobile Traffic Classification through Physical Channel Fingerprinting: A Deep Learning Approach," *arXiv preprint arXiv:1910.11617 [eess.SP]*, October 2019.

[102] H. Kim, J. Lee, E. Lee, and Y. Kim, "Touching the Untouchables: Dynamic Security Analysis of the LTE Control Plane," in *Proc. of IEEE Symp. on Security and Privacy (SP)*, San Francisco, CA, USA, May 2019.

[103] D. Rupprecht, K. Kohls, T. Holz, and C. Pöpper, "Breaking LTE on Layer Two," in *Proc. of IEEE Symp. on Security and Privacy (SP)*, San Francisco, CA, USA, May 2019.

[104] H. Yang, S. Bae, M. Son, H. Kim, S. M. Kim, and Y. Kim, "Hiding in Plain Signal: Physical Signal Overshadowing Attack on LTE," in *Proc. of USENIX Security Symp.*, Santa Clara, CA, USA, August 2019.

[105] A. Singla, S. R. Hussain, O. Chowdhury, E. Bertino, and N. Li, "Protecting the 4G and 5G Cellular Paging Protocols against Security and Privacy Attacks," in *Proc. of Sciendo Privacy Enhancing Technologies*, Montreal, QC, Canada, July 2020.

[106] NIST. OpenFirst. https://www.nist.gov/ctl/pscr/openfirst. Accessed July 2020.

[107] L. Ferranti, L. Bonati, S. D'Oro, and T. Melodia, "SkyCell: A Prototyping Platform for 5G Aerial Base Stations," in *Proc. of IEEE Workshop on Wireless Networking, Planning, and Computing for UAV Swarms (SwarmNet)*, Cork, Ireland, August 2020.

[108] S. D'Oro, L. Bonati, F. Restuccia, M. Polese, M. Zorzi, and T. Melodia,

"Sl-EDGE: Network Slicing at the Edge," in *Proc. of ACM Intl. Symp. on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (MobiHoc)*, Shanghai, China, October 2020.

[109] Radisys. Radysis O-RAN DU. https://gerrit.o-ran-sc.org/r/admin/repos/o-du/l2. Accessed July 2020.

[110] ——. Open-Source 4G RAN Software for Qualcomm FSM9955 Chipset. https://www.radisys.com/OpenRadisys-4G-RAN-Software. Accessed July 2020.

[111] Open5GS. https://open5gs.org. Accessed July 2020.

[112] Open Networking Foundation. Open Mobile Evolved Core (OMEC). https://opennetworking.org/omec. Accessed July 2020.

[113] free5GC. https://free5gc.org. Accessed July 2020.

[114] 3GPP, "General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 23.401, 3 2020, version 16.6.0. [Online]. Available: http://www.3gpp.org/DynaReport/23401.htm

[115] NextEPC. https://nextepc.org. Accessed July 2020.

[116] S. Sevilla, M. Johnson, P. Kosakanchit, J. Liang, and K. Heimerl, "Experiences: Design, Implementation, and Deployment of CoLTE, a Community LTE Solution," in *Proc. of ACM Intl. Conf. on Mobile Computing and Networking (MobiCom)*, Los Cabos, Mexico, October 2019.

[117] bcom. https://5g.labs.b-com.com/. Accessed July 2020.

[118] G. Lee, J. Lee, J. Lee, Y. Im, M. Hollingsworth, E. Wustrow, D. Grunwald, and S. Ha, "This is Your President Speaking: Spoofing Alerts in 4G LTE Networks," in *Proc. of ACM Intl. Conf. on Mobile Systems, Applications, and Services (MobiSys)*, Seoul, South Korea, June 2019.

[119] Hewlett Packard Enterprise, "HPE Speeds Up 5G Adoption with Cloud Native 5G Core Software Stack, Available as-a-Service," March 2020. [Online]. Available: https://www.hpe.com/us/en/newsroom/press-release/2020/03/hpe-speeds-up-5g-adoption-with-cloud-native-5g-core-software-stack-available-as-a-service.html

[120] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese, and D. Walker, "P4: Programming Protocol-Independent Packet Processors," *SIGCOMM Computer Communication Review*, vol. 44, no. 3, p. 87–95, July 2014.

[121] R. Ricart-Sanchez, P. Malagon, P. Salva-Garcia, E. C. Perez, Q. Wang, and J. M. Alcaraz Calero, "Towards an FPGA-Accelerated Programmable Data Path for Edge-to-core Communications in 5G Networks," *Journal of Network and Computer Applications*, vol. 124, pp. 80–93, December 2018.

[122] R. Ricart-Sanchez, P. Malagon, J. M. Alcaraz-Calero, and Q. Wang, "P4-NetFPGA-based Network Slicing Solution for 5G MEC Architectures," in *Proc. of ACM/IEEE Symp. on Architectures for Networking and Communications Systems (ANCS)*, Cambridge, United Kingdom, September 2019.

[123] Kaloom, "The Kaloom 5G User Plane Function (UPF)," https://www.mbuzzeurope.com/wp-content/uploads/2020/02/Product-Brief-Kaloom-5G-UPF-v1.0.pdf, accessed July 2020.

[124] Open Networking Foundation. Converged Multi-Access and Core (COMAC). https://opennetworking.org/comac. Accessed July 2020.

[125] ——. SD-RAN. https://opennetworking.org/sd-ran. Accessed July 2020.

[126] ——. Aether. https://opennetworking.org/aether. Accessed July 2020.

[127] Facebook Connectivity. Magma. https://connectivity.fb.com/magma. Accessed July 2020.

[128] OpenRAN 5G NR. https://telecominfraproject.com/5gnr. Accessed July 2020.

[129] Akraino Radio Edge Cloud. https://www.lfedge.org/projects-old/akraino/release-1/telco-appliance-radio-edge-cloud. Accessed July 2020.

[130] NVIDIA. Aerial SDK. https://developer.nvidia.com/aerial-sdk. Accessed July 2020.

[131] Fondazione Bruno Kessler (FBK). 5G-EmPOWER. https://5g-empower.io. Accessed July 2020.

[132] FlexRAN. http://mosaic-5g.io/flexran. Accessed July 2020.

[133] Open Networking Foundation. Central Office Re-architected as a Datacenter (CORD). https://opennetworking.org/cord. Accessed July 2020.

[134] LL-MEC. http://mosaic5g.io/ll-mec. Accessed July 2020.

[135] LightEdge. https://lightedge.io. Accessed July 2020.

[136] O-RAN Alliance White Paper, "O-RAN Use Cases and Deployment Scenarios," https://static1.squarespace.com/static/5ad774cce74940d7115044b0/t/5e95a0a306c6ab2d1cbca4d3/1586864301196/O-RAN+Use+Cases+and+Deployment+Scenarios+Whitepaper+February+2020.pdf, February 2020.

[137] 3GPP, "NG-RAN; Architecture Description," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.401, 01 2018, version 15.0.0. [Online]. Available: http://www.3gpp.org/DynaReport/38401.htm

[138] O-RAN Fronthaul Working Group, "Control, User and Synchronization Plane Specification - v2.00," ORAN-WG4.CUS.0-v02.00 Technical Specification, 2019.

[139] O-RAN Working Group 2, "A1 Interface: General Aspects and Principles - v1.00," ORAN-WG2.A1.GA&P-v01.00 Technical Specification, 2019.

[140] O-RAN Working Group 1, "O-RAN Operations and Maintenance Interface - v2.00," O-RAN-WG1.O1-Interface-v02.00 Technical Specification, 2019.

[141] R. Enns, M. Bjorklund, J. Schoenwaelder, and A. Bierman, "Network Configuration Protocol (NETCONF)," Internet Requests for Comments, RFC Editor, RFC 6241, June 2011. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6241.txt

[142] O-RAN Working Group 3, "O-RAN Near-Real-time RAN Intelligent Controller Architecture & E2 General Aspects and Principles - v1.00," ORAN-WG3.E2GAP-v01.00 Technical Specification, 2020.

[143] O-RAN Working Group 1, "O-RAN Architecture Description - v1.00," O-RAN-WG1-O-RAN Architecture Description - v01.00.00 Technical Specification, 2020.

[144] 3GPP, "NG-RAN; E1 General Aspects and Principles," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.460, 01 2019, version 16.0.0. [Online]. Available: http://www.3gpp.org/DynaReport/38460.htm

[145] ——, "NG-RAN; F1 General Aspects and Principles," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.470, 01 2018, version 15.0.0. [Online]. Available: http://www.3gpp.org/DynaReport/38470.htm

[146] O-RAN Software Community. Amber Release. https://wiki.o-ran-sc.org/pages/viewpage.action?pageId=14221337. Accessed July 2020.

[147] ——. Bronze Release. https://wiki.o-ran-sc.org/pages/viewpage.action?pageId=14221635. Accessed July 2020.

[148] Open Networking Foundation. Component Projects List. https://wiki.opennetworking.org/display/COM/Component+Projects. Accessed July 2020.

[149] ——. SDN-Enabled Broadband Access (SEBA). https://opennetworking.org/seba. Accessed July 2020.

[150] ——. OpenCORD Guide. https://guide.opencord.org. Accessed July 2020.

[151] ——. OpenCORD Repositories. https://github.com/opencord. Accessed July 2020.

[152] ——. Exemplar Platform List. https://wiki.opennetworking.org/display/COM/Exemplar+Platforms. Accessed July 2020.

[153] ——. COMAC Release. https://guide.opencord.org/profiles/comac/release-notes.html. Accessed July 2020.

[154] ——. COMAC in a Box. https://guide.opencord.org/profiles/comac/install/ciab.html. Accessed July 2020.

[155] E. Coronado, S. N. Khan, and R. Riggio, "5G-EmPOWER: A Software-Defined Networking Platform for 5G Radio Access Networks," *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 715–728, April 2019.

[156] Smart Networks and Services (SENSE) Research Unit at Fondazione Bruno Kessler (FBK). 5G-EmPOWER Repositories. https://github.com/5g-empower. Accessed July 2020.

[157] Mosaic5G Community. Mosaic5G and FlexRAN Projects Repositories. https://gitlab.eurecom.fr/mosaic5g. Accessed July 2020.

[158] N. Nikaein, X. Vasilakos, and A. Huang, "LL-MEC: Enabling Low Latency Edge Applications," in *Proc. of IEEE Intl. Conf. on Cloud Networking (CloudNet)*, Tokyo, Japan, October 2018.

[159] E. Coronado, Z. Yousaf, and R. Riggio, "LightEdge: Mapping the Evolution of Multi-Access Edge Computing in Cellular Networks," *IEEE Communications Magazine*, vol. 58, no. 4, pp. 24–30, April 2020.

[160] LightEdge. LightEdge Repositories. https://github.com/lightedge. Accessed July 2020.

[161] G. Brown, "TIP OpenRAN: Toward Disaggregated Mobile Networking," https://cdn.brandfolder.io/D8DI15S7/as/qc19tk-54bsw-305pae/TIP_OpenRAN_-Heavy_Reading_May_2020-_White_Paper.pdf, May 2020, accessed July 2020.

[162] OpenRAN. https://telecominfraproject.com/openran. Accessed July 2020.

[163] Telecom Infra Project, "OpenRAN 5G NR Base Station Platform Requirements Document," https://cdn.brandfolder.io/D8DI15S7/

as/q688zl-7ly8w8-4xfk0m/TIP_OpenRAN_5GNR_Requirements_Document.pdf, 2020, accessed July 2020.

[164] Akraino Radio Edge Cloud Blueprint. https://wiki.akraino.org/pages/viewpage.action?pageId=6128402. Accessed July 2020.

[165] Akraino Edge Stack Repositories. https://github.com/akraino-edge-stack. Accessed July 2020.

[166] Akraino Telco Appliance Blueprint Family. https://wiki.akraino.org/display/AK/Telco+Appliance+Blueprint+Family. Accessed July 2020.

[167] G. M. Yilma, F. Z. Yousaf, V. Sciancalepore, and X. Costa-Perez, "On the Challenges and KPIs for Benchmarking Open-Source NFV MANO Systems: OSM vs ONAP," *arXiv preprint arXiv:1904.10697 [cs.NI]*, April 2019.

[168] P. Trakadas, P. Karkazis, H. C. Leligou, T. Zahariadis, F. Vicens, A. Zurita, P. Alemany, T. Soenen, C. Parada, J. Bonnet, E. Fotopoulou, A. Zafeiropoulos, E. Kapassa, M. Touloupou, and D. Kyriazis, "Comparison of Management and Orchestration Solutions for the 5G Era," *Journal of Sensor and Actuator Networks*, vol. 9, no. 1, January 2020.

[169] C. Rotsos, D. King, A. Farshad, J. Bird, L. Fawcett, N. Georgalas, M. Gunkel, K. Shiomoto, A. Wang, A. Mauthe, N. Race, and D. Hutchison, "Network Service Orchestration Standardization: A Technology Survey," *Computer Standards & Interfaces*, vol. 54, pp. 203–215, November 2017.

[170] L. Mamushiane, A. A. Lysko, T. Mukute, J. Mwangama, and Z. D. Toit, "Overview of 9 Open-Source Resource Orchestrating ETSI MANO Compliant Implementations: A Brief Survey," in *Proc. of IEEE Wireless Africa Conference (WAC)*, Pretoria, South Africa, August 2019.

[171] Open Source MANO End User Advisory Group, "OSM Scope, Functionality, Operation and Integration Guidelines," https://osm.etsi.org/images/OSM_EUAG_White_Paper_OSM_Scope_and_Functionality.pdf, February 2019.

[172] G. A. Carella, M. Pauls, T. Magedanz, M. Cilloni, P. Bellavista, and L. Foschini, "Prototyping NFV-based Multi-access Edge Computing in 5G Ready Networks with Open Baton," in *Proc. of IEEE Conf. on Network Softwarization (NetSoft)*, Bologna, Italy, July 2017.

[173] A. J. Gonzalez, G. Nencioni, A. Kamisiński, B. E. Helvik, and P. E. Heegaard, "Dependability of the NFV Orchestrator: State of the Art and Research Challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3307–3329, Fourth quarter 2018.

[174] Canonical. Linux Containers (LXC). https://linuxcontainers.org. Accessed July 2020.

[175] Docker. https://www.docker.com. Accessed July 2020.

[176] J. Martins, M. Ahmed, C. Raiciu, and F. Huici, "Enabling Fast, Dynamic Network Processing with ClickOS," in *Proc. of ACM Workshop on Hot Topics in Software Defined Networking (HotSDN)*, Hong Kong, China, August 2013.

[177] A. Bratterud, A.-A. Walla, H. Haugerud, P. E. Engelstad, and K. Begnum, "IncludeOS: A Minimal, Resource Efficient Unikernel for Cloud Services," in *Proc. of IEEE Intl. Conf. on Cloud Computing Technology and Science (CloudCom)*, Vancouver, BC, Canada, December 2015.

[178] A. Kivity, D. Laor, G. Costa, P. Enberg, N. Har'El, D. Marti, and V. Zolotarov, "OSv: Optimizing the Operating System for Virtual Machines," in *Proc. of USENIX Annual Technical Conf.*, Philadelphia, PA, USA, June 2014.

[179] Xen and Linux Foundation. MirageOS. https://github.com/mirage/mirage. Accessed July 2020.

[180] UniK. https://github.com/solo-io/unik. Accessed July 2020.

[181] S. Wu, C. Mei, H. Jin, and D. Wang, "Android Unikernel: Gearing Mobile Code Offloading Towards Edge Computing," *Future Generation Computer Systems*, vol. 86, pp. 694–703, April 2018.

[182] P. Valsamas, S. Skaperas, and L. Mamatas, "Elastic Content Distribution Based on Unikernels and Change-Point Analysis," in *Proc. of European Wireless Conf. (EW)*, Catania, Italy, May 2018.

[183] J. B. Filipe, F. Meneses, A. U. Rehman, D. Corujo, and R. L. Aguiar, "A Performance Comparison of Containers and Unikernels for Reliable 5G Environments," in *Proc. of IEEE Intl. Conf. on the Design of Reliable Communication Networks (DRCN)*, Coimbra, Portugal, March 2019.

[184] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," *ACM SIGOPS Operating Systems Review*, vol. 37, no. 5, pp. 164–177, October 2003.

[185] VMware. ESXi. https://www.vmware.com/products/esxi-and-esx.html. Accessed July 2020.

[186] Linux Kernel-based Virtual Machine (KVM). https://www.linux-kvm.org. Accessed July 2020.

[187] BSD. bhyve. https://bhyve.org. Accessed July 2020.

[188] Oracle. VirtualBox. https://www.virtualbox.org. Accessed July 2020.

[189] OpenStack Project. OpenStack. https://www.openstack.org. Accessed July 2020.

[190] Linux Foundation. Kubernetes. https://kubernetes.io. Accessed July 2020.

[191] Istio. https://istio.io. Accessed July 2020.

[192] Network Service Mesh (NSM). https://networkservicemesh.io. Accessed July 2020.

[193] Linux Foundation, "ONAP Dublin Release," https://www.linuxfoundation.org/press-release/2019/07/onap-doubles-down-on-deployments-drives-commercial-activity-across-open-source-networking-stack-with-dublin-release, July 2019.

[194] ——, "ONAP Frankfurt Release," https://www.onap.org/announcement/2020/06/18/onaps-6th-release-frankfurt-available-now-most-comprehensive-secure-and-collaborative-software-to-accelerate-5g-deployments, June 2020.

[195] A. Kapadia, *ONAP Demystified: Automate Network Services with ONAP*. North Charleston, SC, USA: CreateSpace Independent Publishing Platform, 2018.

[196] Consul, "Service Mesh and Microservice Networking," https://www.datocms-assets.com/2885/1536681707-consulwhitepaperaug2018.pdf, August 2018.

[197] F. Slim, F. Guillemin, A. Gravey, and Y. Hadjadj-Aoul, "Towards a Synamic Adaptive Placement of Virtual Network Functions Under ONAP," in *Proc. of IEEE Conf. on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Berlin, Germany, 2017.

[198] V. Q. Rodriguez, F. Guillemin, and A. Boubendir, "5G E2E Network Slicing Management with ONAP," in *Proc. of IEEE Conf. on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, Paris, France, February 2020.

[199] ONAP. Mobility Standards Harmonization with ONAP. https://wiki.onap.org/display/DW/MOBILITY+STANDARDS+HARMONIZATION+WITH+ONAP. Accessed July 2020.

[200] Canonical. Juju Charms. https://jaas.ai. Accessed July 2020.

[201] Open Source MANO End User Advisory Group, "OSM Deployment and Integration," https://osm.etsi.org/images/OSM_EUAG_White_Paper_OSM_Deployment_and_Integration.pdf, February 2020.

[202] R. Casellas, R. Martínez, R. Vilalta, and R. Muñoz, "Metro-Haul: SDN Control and Orchestration of Disaggregated Optical Networks with Model-Driven Development," in *Proc. of IEEE Intl. Conf. on Transparent Optical Networks (ICTON)*, July 2018.

[203] T. Soenen, W. Tavernier, M. Peuster, F. Vicens, G. Xilouris, S. Kolometsos, M. Kourtis, and D. Colle, "Empowering Network Service Developers: Enhanced NFV DevOps and Programmable MANO," *IEEE Communications Magazine*, vol. 57, no. 5, pp. 89–95, May 2019.

[204] C. Colman-Meixner, P. Diogo, M. S. Siddiqui, A. Albanese, H. Khalili, A. Mavromatis, L. Luca, A. Ulisses, J. Colom, R. Nejabati, and D. Simeonidou, "5G City: A Novel 5G-Enabled Architecture for Ultra-High Definition and Immersive Media on City Infrastructure," in *Proc. of IEEE Intl. Symp. on Broadband Multimedia Systems and Broadcasting (BMSB)*, Valencia, Spain, June 2018.

[205] S. Rizou, P. Athanasoulis, P. Andriani, F. Iadanza, G. Carrozzo, D. Breitgand, A. Weit, D. Griffin, D. Jimenez, U. Acar, and O. P. Gordo, "A Service Platform Architecture Enabling Programmable Edge-To-Cloud Virtualization for the 5G Media Industry," in *Proc. of IEEE Intl. Symp. on Broadband Multimedia Systems and Broadcasting (BMSB)*, Valencia, Spain, June 2018.

[206] A. de la Oliva, X. Li, X. Costa-Perez, C. J. Bernardos, P. Bertin, P. Iovanna, T. Deiss, J. Mangues, A. Mourad, C. Casetti, J. E. Gonzalez, and A. Azcorra, "5G-TRANSFORMER: Slicing and Orchestrating Transport Networks for Industry Verticals," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 78–84, August 2018.

[207] T. Dreibholz, "Flexible 4G/5G Testbed Setup for Mobile Edge Computing Using OpenAirInterface and Open Source MANO," in *Springer Web, Artificial Intelligence and Network Applications (WAINA)*, March 2020.

[208] C. Tranoris, S. Denazis, L. Guardalben, J. Pereira, and S. Sargento, "Enabling Cyber-Physical Systems for 5G Networking: A Case Study on the Automotive Vertical Domain," in *Proc. of the IEEE/ACM Intl. Workshop on Software Engineering for Smart Cyber-Physical Systems (SEsCPS)*, Gothenburg, Sweden, 2018.

[209] Open Baton. Open Baton Project. https://openbaton.github.io. Accessed July 2020.

[210] ——. Open Baton Repositories. https://github.com/openbaton. Accessed July 2020.

[211] Skylark Wireless. https://www.skylarkwireless.com. Accessed July 2020.

[212] Argos. http://argos.rice.edu/. Accessed July 2020.

[213] Platform for Open Wireless Data-driven Experimental Research (POWDER). https://www.powderwireless.net. Accessed July 2020.

[214] Reconfigurable Eco-system for Next-generation End-to-end Wireless (RENEW). https://renew.rice.edu. Accessed July 2020.

[215] J. Breen, A. Buffmire, J. Duerig, K. Dutt, E. Eide, M. Hibler, D. Johnson, S. Kasera, E. Lewis, D. Maas, A. Orange, N. Patwari, D. Reading, R. Ricci, D. Schurig, L. Stoller, J. Van der Merwe, K. Webb, and G. Wong, "POWDER: Platform for Open Wireless Data-driven Experimental Research," in *Proc. of ACM Intl. Workshop on Wireless Network Testbeds, Experimental evaluation & CHaracterization (WiNTECH)*, London, United Kingdom, September 2020.

[216] R. Doost-Mohammady, O. Bejarano, and A. Sabharwal, "Good Times For Wireless Research," in *Proc. of ACM Intl. Workshop on Wireless Network Testbeds, Experimental evaluation & CHaracterization (WiNTECH)*, London, United Kingdom, September 2020.

[217] Cloud Enhanced Open Software Defined Mobile Wireless Testbed for City-Scale Deployment (COSMOS). http://www.cosmos-lab.org. Accessed July 2020.

[218] D. Raychaudhuri, I. Seskar, G. Zussman, T. Korakis, D. Kilper, T. Chen, J. Kolodziejski, M. Sherman, Z. Kostic, X. Gu, H. Krishnaswamy, S. Maheshwari, P. Skrimponis, and C. Gutterman, "Challenge: COSMOS: A City-Scale Programmable Testbed for Experimentation with Advanced Wireless," in *Proc. of ACM Intl. Conf. on Mobile Computing and Networking (MobiCom)*, London, United Kingdom, September 2020.

[219] M. Kohli, T. Chen, M. Dastjerdi, J. Welles, I. Seskar, H. Krishnaswamy, and G. Zussman, "Open-Access Full-Duplex Wireless in the ORBIT and COSMOS Testbeds," in *Proc. of ACM Intl. Workshop on Wireless Network Testbeds, Experimental evaluation & CHaracterization (WiNTECH)*, London, United Kingdom, September 2020.

[220] Open-Access Research Testbed for Next-Generation Wireless Networks (ORBIT). https://www.orbit-lab.org. Accessed July 2020.

[221] O-RAN Alliance Conducts First Global Plugfest to Foster Adoption of Open and Interoperable 5G Radio Access Networks. (2019, December) https://static1.squarespace.com/static/5ad774cce74940d7115044b0/t/5dfba8fb1326ae1bcf4a8b6f/1576773884092/O-RAN-2019.12.19-EC-C-PR-on-2019-Plugfest-v1.0.pdf.

[222] Aerial Experimentation and Research Platform for Advanced Wireless (AERPAW). https://www.aerpaw.org. Accessed July 2020.

[223] V. Marojevic, I. Guvenc, M. Sichitiu, and R. Dutta, "An Experimental Research Platform Architecture for UAS Communications and Networking," in *Proc. of IEEE Vehicular Technology Conference (VTC2019-Fall)*, Honolulu, HI, USA, September 2019.

[224] M. Sichitiu, I. Guvenc, R. Dutta, V. Marojevic, and B. Floyd, "AERPAW Emulation Overview," in *Proc. of ACM Intl. Workshop on Wireless Network Testbeds, Experimental evaluation & CHaracterization (WiNTECH)*, London, United Kingdom, September 2020.

[225] Colosseum. https://www.colosseum.net. Accessed July 2020.

[226] L. Bertizzolo, L. Bonati, E. Demirors, and T. Melodia, "Demo: Arena: A 64-antenna SDR-based Ceiling Grid Testbed for Sub-6 GHz Radio Spectrum Research," in *Proc. of ACM Intl. Workshop on Wireless Network Testbeds, Experimental evaluation & CHaracterization (WiNTECH)*, Los Cabos, Mexico, October 2019.

[227] 5TONIC. https://www.5tonic.org. Accessed July 2020.

[228] B. Nogales, I. Vidal, D. R. Lopez, J. Rodriguez, J. Garcia-Reinoso, and A. Azcorra, "Design and Deployment of an Open Management and Orchestration Platform for Multi-Site NFV Experimentation," *IEEE Communications Magazine*, vol. 57, no. 1, pp. 20–27, 2019.

[229] M. J. Roldan, P. Leithead, and J. Mack, "Experiments and Results of a mmW Transport Platform to Enable 5G Cloud RAN Lower Layer Splits," in *Proc. of IEEE Long Island Systems, Applications and Technology Conf. (LISAT)*, Farmingdale, NY, USA, 2018.

[230] Horizon 2020. https://ec.europa.eu/programmes/horizon2020. Accessed July 2020.

[231] Federation 4 Future Internet Research and Experimentation Plus (FED4FIRE+). https://www.fed4fire.eu. Accessed July 2020.

[232] NITOS. https://nitlab.inf.uth.gr/NITlab/nitos. Accessed July 2020.

[233] Icarus Node. https://nitlab.inf.uth.gr/NITlab/hardware/wireless-nodes/icarus-nodes. Accessed July 2020.

[234] P. Karamichailidis, K. Choumas, and T. Korakis, "Enabling Multi-Domain Orchestration using Open Source MANO, OpenStack and OpenDaylight," in *Proc. of IEEE Intl. Symp. on Local and Metropolitan Area Networks (LANMAN)*, Paris, France, July 2019.

[235] K. Chounos, N. Makris, and T. Korakis, "Enabling Distributed Spectral Awareness for Disaggregated 5G Ultra-Dense HetNets," in *Proc. of IEEE 5G World Forum (5GWF)*, Dresden, Germany, September 2019.

[236] N. Makris, V. Passas, C. Nanis, and T. Korakis, "On Minimizing Service Access Latency: Employing MEC on the Fronthaul of Heterogeneous 5G Architectures," in *Proc. of IEEE Intl. Symp. on Local and Metropolitan Area Networks (LANMAN)*, Paris, France, July 2019.

[237] V. Passas, N. Makris, V. Miliotis, and T. Korakis, "Pricing Based MEC Resource Allocation for 5G Heterogeneous Network Access," in *Proc. of IEEE Global Communications Conf. (GLOBECOM)*, Waikoloa, HI, USA, December 2019.

[238] IRIS. http://iristestbed.eu. Accessed July 2020.

[239] COgnitive Radio NETwork (CORNET). https://cornet.wireless.vt.edu. Accessed July 2020.

[240] R. M. Rao, V. Marojevic, and J. H. Reed, "Analysis of Non-Pilot Interference on Link Adaptation and Latency in Cellular Networks," in *Proc. of IEEE Vehicular Technology Conf. (VTC2019-Spring)*, Kuala Lumpur, Malaysia, April 2019.

[241] Future Internet of Things (FIT). https://fit-equipex.fr. Accessed July 2020.

[242] Drexel Grid SDR Testbed. https://research.coe.drexel.edu/ece/dwsl/research/drexel-grid-sdr-testbed. Accessed July 2020.

[243] K. R. Dandekar, S. Begashaw, M. Jacovic, A. Lackpour, I. Rasheed, X. R. Rey, C. Sahin, S. Shaher, and G. Mainland, "Grid Software Defined Radio Network Testbed for Hybrid Measurement and Emulation," in *Proc. of IEEE Intl. Conf. on Sensing, Communication, and Networking (SECON)*, Boston, MA, USA, June 2019.

[244] M. Mezzavilla, M. Zhang, M. Polese, R. Ford, S. Dutta, S. Rangan, and M. Zorzi, "End-to-End Simulation of 5G mmWave Networks," *IEEE Communications Surveys Tutorials*, vol. 20, no. 3, pp. 2237–2263, April 2018.

[245] N. Patriciello, S. Lagen, B. Bojovic, and L. Giupponi, "An E2E Simulator for 5G NR Networks," *Simulation Modelling Practice and Theory*, vol. 96, no. 101933, November 2019.

[246] S. Pratschner, B. Tahir, L. Marijanovic, M. Mussbah, K. Kirev, R. Nissel, S. Schwarz, and M. Rupp, "Versatile Mobile Communications Simulation: The Vienna 5G Link Level Simulator," *Springer EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, p. 226, September 2018.

[247] E. J. Oughton, K. Katsaros, F. Entezami, D. Kaleshi, and J. Crowcroft, "An Open-Source Techno-Economic Assessment Framework for 5G Deployment," *IEEE Access*, vol. 7, pp. 155 930–155 940, October 2019.

[248] Pi-Radio. Fully-digital mmWave Front-ends. https://www.pi-rad.io/home/product. Accessed July 2020.

[249] A. Dhananjay, K. Zheng, J. Haarla, L. Iotti, M. Mezzavilla, D. Shasha, and S. Rangan, "Calibrating a 4-channel Fully-Digital 60 GHz SDR," in *Proc. of ACM Intl. Workshop on Wireless Network Testbeds, Experimental evaluation & CHaracterization (WiNTECH)*, London, United Kingdom, September 2020.

[250] M. Polese, F. Restuccia, A. Gosain, J. Jornet, S. Bhardwaj, V. Ariyarathna, S. Mandal, K. Zheng, A. Dhananjay, M. Mezzavilla, J. Buckwalter, M. Rodwell, X. Wang, M. Zorzi, A. Madanayake, and T. Melodia, "MillimeTera: Toward A Large-Scale Open-Source mmWave and Terahertz Experimental Testbed," in *Proc. of ACM Workshop on Millimeter-Wave Networks and Sensing Systems (mmNets)*, Los Cabos, Mexico, 2019.

[251] J. Haarla, V. Semkin, K. Zheng, A. Dhananjay, M. Mezzavilla, J. Ala-Laurinaho, and V. Viikari, "Characterizing 60 GHz Patch Antenna Segments for Fully Digital Transceiver," in *Proc. of the 14th European Conference on Antennas and Propagation (EuCAP)*, Copenhagen, Denmark, 2020.

[252] R. Akeela and B. Dezfouli, "Software-defined Radios: Architecture, State-of-the-Art, and Challenges," *Computer Communications*, vol. 128, pp. 106–125, 2018.

[253] M. Savi, M. Tornatore, and G. Verticale, "Impact of Processing-Resource Sharing on the Placement of Chained Virtual Network Functions," *IEEE Transactions on Cloud Computing*, 2019.

[254] C. Cowan, "Software Security for Open-source Systems," *IEEE Security Privacy*, vol. 1, no. 1, pp. 38–45, January 2003.

[255] V. Piantadosi, S. Scalabrino, and R. Oliveto, "Fixing of Security Vulnerabilities in Open Source Projects: A Case Study of Apache HTTP Server and Apache Tomcat," in *Proc. of IEEE Conf. on Software Testing, Validation and Verification (ICST)*, Xi'an, China, 2019.

[256] P. Wang, J. Krinke, K. Lu, G. Li, and S. Dodier-Lazaro, "How Double-Fetch Situations turn into Double-Fetch Vulnerabilities: A Study of Double Fetches in the Linux Kernel," in *Proc. of USENIX Security Symp. (USENIX Security)*, Vancouver, BC, Canada, August 2017.

APPENDIX A
ACRONYMS

3GPP 3rd Generation Partnership Project
4G 4th generation
5G 5th generation
5GC 5G Core
AERPAW Aerial Experimentation and Research Platform for Advanced Wireless
AM Acknowledged Mode
AMF Access and Mobility Management Function
API Application Programming Interface
APN Access Point Name
AUSF Authentication Server Function
BSS Business Support System
CaaS Connectivity-as-a-Service
CDD Cyclic Delay Diversity
CDN Content Distribution Network
CN Core Network
COMAC Converged Multi-Access and Core
CORD Central Office Re-architected as a Datacenter
CORNET COgnitive Radio NETwork
COSMOS Cloud Enhanced Open Software Defined Mobile Wireless Testbed for City-Scale Deployment
COTS Commercial Off-the-Shelf
CPU Central Processing Unit
CQI Channel Quality Information
CR Cognitive Radio
CU Central Unit
DAS Distributed Antenna System
DL Downlink
DRS Discovery Reference Signal
DU Distributed Unit
E-UTRAN Evolved Universal Terrestrial Access Network
eNB evolved Node Base
EPC Evolved Packet Core
EPS Evolved Packet System
ETSI European Telecommunications Standards Institute
FaaS Function-as-a-Service
FAPI Functional Application Platform Interface
FDD Frequency Division Duplexing
FED4FIRE+ Federation 4 Future Internet Research and Experimentation Plus
FIT Future Internet of Things
FPGA Field Programmable Gate Array
gNB Next Generation Node Base
GPU Graphics Processing Unit
GTP GPRS Tunneling Protocol
GTP-C GPRS Tunneling Protocol Control Plane
GTP-U GPRS Tunneling Protocol User Plane
GW Gateway
HARQ Hybrid Automatic Repeat reQuest
HSS Home Subscription Server
IMSI International Mobile Subscriber Identity
IoT Internet of Things
KPI Key Performance Indicator
KVM Kernel-based Virtual Machine
LTE Long Term Evolution
LXC Linux Containers
MAC Medium Access Control
MANO Management and Orchestration
MCC Mobile Cloud Computing
MCHEM Massive Channel Emulator
MEC Multi-access Edge Computing
MFC Mobile Fog Computing
MIMO Multiple Input, Multiple Output
MME Mobility Management Entity
mmWave millimeter wave
MU-MIMO Multi-user MIMO
MVNO Mobile Virtual Network Operator
NAS Non-Access Stratum
NFV Network Function Virtualization
NFVI Network Function Virtualization Infrastructure
NIC Network Interface Card
NRF Network Repository Function
NSE Network Slicing Engine
NSM Network Service Mesh
NSSF Network Slice Selection Function
OAI OpenAirInterface
OAI-CN OAI Core Network
OAI-RAN OpenAirInterface Radio Access Network
OAM Operations, Administration and Maintenance
OMEC Open Mobile Evolved Core
ONAP Open Network Automation Platform
ONF Open Networking Foundation
ONOS Open Networking Operating System
OOM ONAP Operations Manager
OPNFV Open Platform for NFV
ORBIT Open-Access Research Testbed for Next-Generation Wireless Networks
OSM Open Source NFV Management and Orchestration
OSS Operations Support System
PAWR Platforms for Advanced Wireless Research
PBCH Physical Broadcast Channel
PCEF Policy and Charging Enforcement Function
PCFICH Physical Control Format Indicator Channel
PCRF Policy and Charging Rules Function
PDCCH Physical Downlink Control Channel
PDCP Packet Data Convergence Protocol
PDSCH Physical Downlink Shared Channel
PGW Packet Gateway
PHICH Physical Hybrid ARQ Indicator Channel
PHY Physical
PMCH Physical Multicast Channel
PMI Precoding Matrix Indicators
POWDER Platform for Open Wireless Data-driven Experimental Research
PRACH Physical Random Access Channel
PRB Physical Resource Block
PSS Primary Synchronization Signal
PUCCH Physical Uplink Control Channel
PUSCH Physical Uplink Shared Channel
QAM Quadrature Amplitude Modulation
QCI QoS Class Identifier
QoS Quality of Service
RAN Radio Access Network
RAT Radio Access Technology
REC Radio Edge Cloud
RENEW Reconfigurable Eco-system for Next-generation End-to-end Wireless
RIC RAN Intelligent Controller
RLC Radio Link Control
RRC Radio Resource Control
RRU Remote Radio Unit
RU Radio Unit
S1AP S1 Application Protocol
SC2 Spectrum Collaboration Challenge
SDAP Service Data Adaptation Protocol
SDK Software Development Kit
SDN Software-defined Networking
SDR Software-defined Radio
SEBA SDN-Enabled Broadband Access
SGW Service Gateway
SISO Single Input, Single Output
SLA Service Level Agreement
SMF Session Management Function
SRS Sounding Reference Signal
SSS Secondary Synchronization Signal
TDD Time Division Duplexing
TFT Traffic Flow Template
TIP Telecom Infra Project
TM Transparent Mode
UAS Unmanned Aerial System
UAV Unmanned Aerial Vehicle
UDM Unified Data Management
UDP User Datagram Protocol
UDR Unified Data Repository
UE User Equipment
UL Uplink
UM Unacknowledged Mode
UPF User Plane Function
URLLC Ultra Reliable and Low Latency Communication
USIM Universal Subscriber Identity Module
USRP Universal Software Radio Peripheral
VIM Virtualization Infrastructure Manager
VM Virtual Machine

VNF Virtual Network Function
VoLTE Voice over LTE
vRAN Virtualized RAN

Leonardo Bonati received his B.S. in Information Engineering and his M.S. in Telecommunication Engineering from University of Padova, Italy in 2014 and 2016, respectively. He is currently pursuing a Ph.D. degree in Computer Engineering at Northeastern University, MA, USA. His research interests focus on 5G and beyond cellular networks, network slicing, and software-defined networking for wireless networks.

Michele Polese is an Associate Research Scientist at Northeastern University, Boston, since March 2020, working with Tommaso Melodia. He received his Ph.D. at the Department of Information Engineering of the University of Padova in 2020 under the supervision of with Michele Zorzi. He also was an adjunct professor and postdoctoral researcher in 2019/2020 at the University of Padova. During his Ph.D., he visited New York University (NYU), AT&T Labs in Bedminster, NJ, and Northeastern University, Boston, MA. He collaborated with several academic and industrial research partners, including Intel, InterDigital, NYU, AT&T Labs, University of Aalborg, King's College and NIST. He was awarded with an Honorable Mention by the Human Inspired Technology Research Center (HIT) (2018), the Best Journal Paper Award of the IEEE ComSoc Technical Committee on Communications Systems Integration and Modeling (CSIM) 2019, and the Best Paper Award at WNS3 2019. His research interests are in the analysis and development of protocols and architectures for future generations of cellular networks (5G and beyond), in particular for millimeter-wave communication, and in the performance evaluation of complex networks. He is a Member of the IEEE.

Salvatore D'Oro received received his Ph.D. degree from the University of Catania in 2015. He is currently a Research Assistant Professor at Northeastern University. He serves on the Technical Program Committee (TPC) of Elsevier Computer Communications journal and the IEEE Conference on Standards for Communications and Networking (CSCN) and European Wireless. He also served on the TPC of Med-Hoc-Net 2018 and several workshops in conjunction with IEEE INFOCOM and IEEE ICC. In 2015, 2016 and 2017 he organized the 1st, 2nd and 3rd Workshops on COmpetitive and COoperative Approaches for 5G networks (COCOA). Dr. D'Oro is also a reviewer for major IEEE and ACM journals and conferences. Dr. D'Oro's research interests include game-theory, optimization, learning and their applications to 5G networks. He is a Member of the IEEE.

Stefano Basagni is with the Institute for the Wireless Internet of Things and an associate professor at the ECE Department at Northeastern University, in Boston, MA. He holds a Ph.D. in electrical engineering from the University of Texas at Dallas (December 2001) and a Ph.D. in computer science from the University of Milano, Italy (May 1998). Dr. Basagni's current interests concern research and implementation aspects of mobile networks and wireless communications systems, wireless sensor networking for IoT (underwater and terrestrial), definition and performance evaluation of network protocols and theoretical and practical aspects of distributed algorithms. Dr. Basagni has published over nine dozen of highly cited, refereed technical papers and book chapters. His h-index is currently 45 (August 2020). He is also co-editor of three books. Dr. Basagni served as a guest editor of multiple international ACM/IEEE, Wiley and Elsevier journals. He has been the TPC co-chair of international conferences. He is a distinguished scientist of the ACM, a senior member of the IEEE, and a member of CUR (Council for Undergraduate Education).

Tommaso Melodia is the William Lincoln Smith Chair Professor with the Department of Electrical and Computer Engineering at Northeastern University in Boston. He is also the Founding Director of the Institute for the Wireless Internet of Things and the Director of Research for the PAWR Project Office. He received his Ph.D. in Electrical and Computer Engineering from the Georgia Institute of Technology in 2007. He is a recipient of the National Science Foundation CAREER award. Prof. Melodia has served as Associate Editor of IEEE Transactions on Wireless Communications, IEEE Transactions on Mobile Computing, Elsevier Computer Networks, among others. He has served as Technical Program Committee Chair for IEEE Infocom 2018, General Chair for IEEE SECON 2019, ACM Nanocom 2019, and ACM WUWnet 2014. Prof. Melodia is the Director of Research for the Platforms for Advanced Wireless Research (PAWR) Project Office, a $100M public-private partnership to establish 4 city-scale platforms for wireless research to advance the US wireless ecosystem in years to come. Prof. Melodia's research on modeling, optimization, and experimental evaluation of Internet-of-Things and wireless networked systems has been funded by the National Science Foundation, the Air Force Research Laboratory the Office of Naval Research, DARPA, and the Army Research Laboratory. Prof. Melodia is a Fellow of the IEEE and a Senior Member of the ACM.