# Mosaic5G: Agile and Flexible Service Platforms for 5G Research

Navid Nikaein, Chia-Yu Chang, Konstantinos Alexandris
Communication Systems Department, EURECOM, France
firstname.lastname@eurecom.fr

## ABSTRACT

Network slicing is one of the key enablers to provide the required flexibility and to realize the service-oriented vision toward fifth generation (5G) mobile networks. In that sense, virtualization, softwarization, and disaggregation are core concepts to accommodate the requirements of an end-to-end (E2E) service to be either isolated, shared, or customized. They lay the foundation for a multi-service and multi-tenant architecture, and are realized by applying the principles of software-defined networking (SDN), network function virtualization (NFV), and cloud computing to the mobile networks. Research on these principles requires agile and flexible platforms that offer a wide range of real-world experimentations over different domains to open up innovations in 5G. To this end, we present Mosaic5G, a community-led consortium for sharing platforms, providing a number of software components, namely FlexRAN, LL-MEC, JOX and Store, spanning application, management, control and user plane on top of OpenAirInterface (OAI) platform. Finally, we show several use cases of Mosaic5G corresponding to widely-mentioned 5G research directions.

## CCS CONCEPTS

• **Networks** → *Mobile networks*; *Programmable networks*; *Network management*; *Network monitoring*;

## KEYWORDS

5G, Network Slicing, Open-source platforms, Service-orientation

## 1 INTRODUCTION

Fifth generation (5G) mobile networks represent a paradigm shift beyond the new radio and wider spectrum with the objective to improve the efficiency and flexibility of mobile networks. It aims also to evolve the computing for wireless networks and to enable the service-oriented vision to deliver networks on an *as-a-service* basis [16]. The idea to support multiple services and/or virtual networks on a single physical network with different service requirements is in terms of service level agreement (SLA), the control and management functions (e.g., either dedicated or shared), and also the performance (e.g., throughput and latency). To accommodate the requirements of an end-to-end (E2E) service, it is required to flexibly customize a slice service, automate its life-cycle management, and ease the development of network functions and applications.

Through the service-oriented 5G vision, naturally the network infrastructure providers (e.g., operators and data center owners), service providers (e.g., over-the-top and verticals), and network
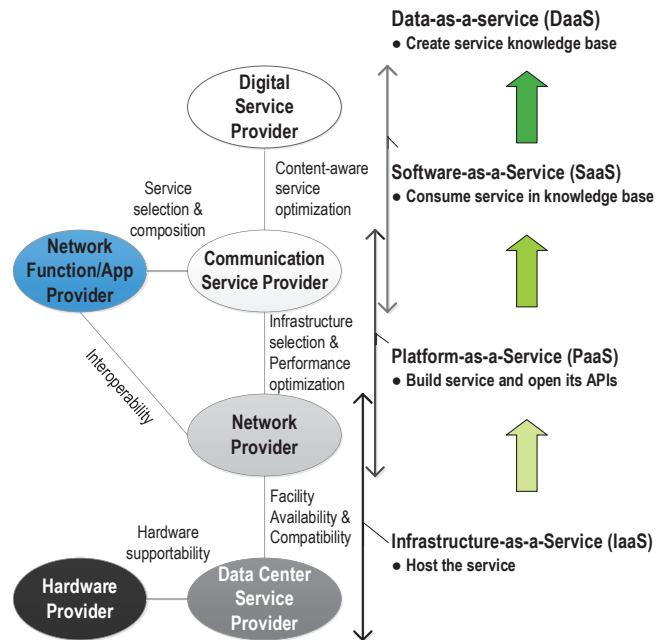


**Figure 1: Four *as-a-service* levels for different providers.**

function/application providers (e.g., vendors) are decoupled to allow a flexible and cost-effective network composition model. Fig. 1 illustrates the relationship between different providers and the transformation of the value-chain in telecommunication industry. Also, four *as-a-service* levels can be derived. The Infrastructure-as-a-Service (IaaS) provides programmable physical and/or virtual infrastructures (e.g., software-defined radio, x86-based infrastructure) and hosts the network services, either commercial or open-source. The Platform-as-a-Service (PaaS) extends IaaS in support of monitoring, control, orchestration, and network function virtualization (NFV), and provides application programming interfaces (APIs) and the slice-friendly development environment. The Software-as-a-Service (SaaS) consumes the programmable control applications such as radio resource management (RRM) and spectrum management application to provide the sophisticated service control logics. Finally, the Data-as-a-Service (DaaS) consumes the aggregated network information to produce a knowledge base and to support cognitive network management.
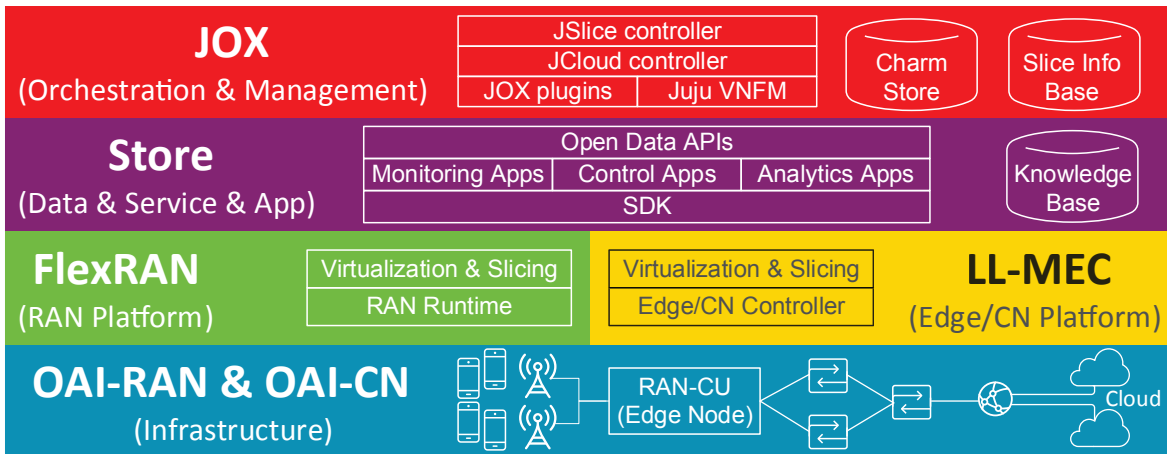
**Figure 2: Mosaic5G schematic architecture.**

To realize aforementioned service-orientation, Mosaic5G[1] initiative is formed to provide an open, flexible and agile 4G/5G experimentation platform. It aims to share an ecosystem of open-source platforms and use cases for 5G system research leveraging software-defined networking (SDN), network function virtualization (NFV), and multi-access edge computing (MEC) technology enablers. Mosaic5G spans five software components: (1) JOX as the service orchestrator, (2) Store as the repository of applications and datasets, (3) LL-MEC as core network (CN) and edge controller, (4) FlexRAN as the real-time radio access network (RAN) controller, and (5) OpenAirInterface (OAI) RAN and OAI CN as 3GPP-compliant implementation of LTE/LTE-A features. As a result, Mosaic5G strives to bring openness into 4G/5G for four directions among innovation, scalability, agility and flexibility. In the following sections, we will provide an overview on each components of Mosaic5G and give their example use cases.

## 2 MOSAIC5G OVERVIEW

As mentioned beforehand, there are five platforms in Mosaic5G schematic architecture depicted in Fig. 2 that are open access:

(1) **JOX** [13] is an event-driven Juju [3]-based service orchestrator core with several plugins to interact with different network domains, e.g., RAN and CN.
(2) **Store** [15] includes a constellation of platform packages, software development kits (SDKs), network control applications and datasets.
(3) **FlexRAN** [8] is a flexible and programmable platform to apply the SDN principle at the RAN domain that enables software-defined RAN (SD-RAN).
(4) **LL-MEC** [11] is an ETSI-aligned MEC platform that can act as a software-defined core network controller.
(5) **OAI-RAN** and **OAI-CN** [14] are 3GPP compatible implementations of a subset of RAN (Release 14) and CN (Release 12) features, respectively. The OAI-RAN and OAI-CN are correspondingly in support of FlexRAN and LL-MEC.

These five platforms can be mapped to different *as-a-service* levels as mentioned in Fig. 1: (a) IaaS level is related to both OAI-RAN and OAI-CN, (b) PaaS can be mapped from FlexRAN, LL-MEC and JOX platforms, and (c) both SaaS and DaaS are provided by the Store repository. Note that these platforms are not strongly coupled, for instance, FlexRAN, LL-MEC, and JoX can be used separately or together on top of OAI platform via the implemented extensible south-bound APIs and control protocols. The Store components can either be used together with other platform or with the offline datasets. Moreover, current software platforms can be deployed over common Intel-based x86-based infrastructures with a certain number of software dependencies managed through appropriate build scripts for each platform and for the top level. In the following, we elaborate on each platform in more details.

### 2.1 FlexRAN

The FlexRAN platform is the first open-source SD-RAN platform that is designed to flexibly separate control and user plane operations, as discussed in several related works [10, 18]. Moreover, it can either centralize RAN domain control logics among multiple base stations (either monolithic or disaggregated RAN) or delegate control decisions in a distributed manner. Hence, FlexRAN provides customized control functions, a hierarchical control framework with a well-defined API allowing "on-the-fly" monitoring, control delegation and reconfiguration in the RAN domain.

Two key elements can be found in FlexRAN as shown in Fig. 3: (a) Real-time controller (RTC) that enables coordinated control over multiple RANs, reveals high/low-level primitives and provision SDKs for control application, and (b) RAN runtime [6] that acts as a local agent controlled by RTC, virtualizes the underlying RAN radio resources, pipelines the RAN service function chain, and provides SDKs enabling distributed control applications. Further, the RAN runtime can support various slice requirements (e.g., isolation) and also improve multiplexing benefits (e.g., sharing) in terms of radio resource abstractions and modularized/customized RAN compositions for RAN slicing purpose. The FlexRAN protocol used between the RAN runtime and the RTC can provide several characteristics: provide statistics, enable reconfiguration, trigger events and delegate control.
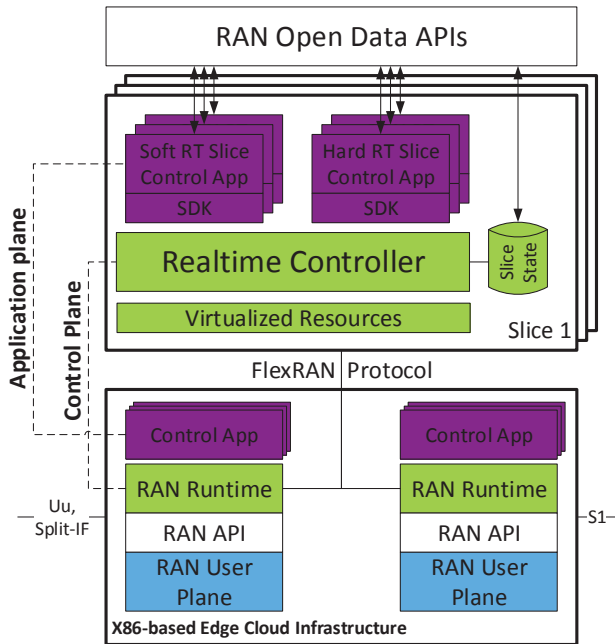
---

[1]http://mosaic-5g.io/
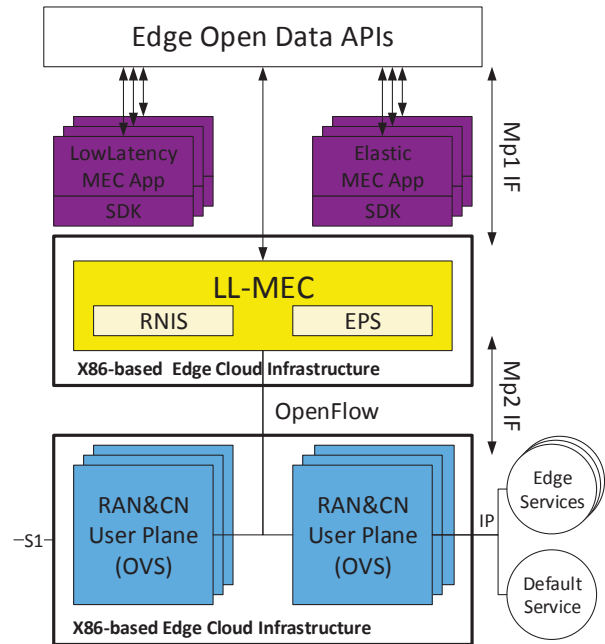
Figure 3: FlexRAN platform.



Figure 4: LL-MEC platform.

## 2.2 LL-MEC

The LL-MEC platform, as shown in Fig. 4, leverages the SDN principle in order to separate user plane processing from its control logics at the edge and core networks and to enable the MEC principle [17]. By applying OpenFlow traffic rules from the LL-MEC to the underlying GTP-enabled Open Virtual Switch (OVS), the user plane can be abstracted for monitoring and analysis as well as be programmed for customizing the control. Further, SDKs are provided to enable a flexible MEC application development environment.

Practically, the LL-MEC platform is aligned with the ETSI MEC Mp1 and Mp2 reference interfaces defined in ETSI GS MEC 003. The Mp1 interface enables low-latency or elastic MEC applications through Core API, REST API and message bus, while the Mp2 interface can instruct user plane in terms of how to route traffic among applications, networks, services, etc. Within the LL-MEC platform, two main services are provided [4]: (a) Edge packet service (EPS) (equivalent to traffic rule control) that manages the static and dynamic traffic rules and handles multiple OpenFlow libraries and OVS, and (b) Radio network information service (RNIS) that extracts real-time RAN information (e.g., user and radio bearer statistics) and delegates the control decision over the user plane.

## 2.3 Store

The Store is in form of a distribution repository that contains a constellation of platform packages, SDKs, control applications, datasets and models as depicted in Fig. 5. It aims to develop and bundle plug-and-play (P&P) network applications tailored to a particular use case, and also to compose and customize a network service delivery platform across reusable applications. Each control application has its control purpose, and it relies on different granularities of network status information from the platform SDK and may further provide APIs to other control applications. Note that these Store

control applications can operate either on real-time structured data, i.e., JSON, that is being produced or on the previously recorded datasets as the offline mode. Datasets are the platform-specific aggregated network information that can be processed to identify possible anomalies and to forecast future patterns. It can be utilized to generate the knowledge base and be analyzed to either find appropriate network control actions or validate some hypotheses through the decision making algorithms. Via the open data APIs, an application can publish its knowledge and capabilities to other applications, or subscribe to the knowledge and capabilities from other applications. Finally, the Store also includes snaps[2] for different platforms (e.g., OAI-RAN, OAI-CN, FlexRAN, LL-MEC, JOX) and Charms[3] templates that can be bundled for different use cases. For instance, several Charms and service bundles can be found at Juju charm store at https://jujucharms.com/q/oai.

## 2.4 JOX

JOX is a Juju-based orchestrator for the virtualized network that natively supports network slicing in order to deploy network slices with different isolated E2E logical networks as shown in Fig. 6. Using JOX, each network slice can be independently optimized with specific configurations on its resources, virtual network functions (VNFs) and service chains. JOX operates on top of the Juju virtual network function management (VNFM) with a plugin architecture to interface with FlexRAN, LL-MEC and virtual infrastructure management (VIM). Last but not least, JOX is compatible with the ETSI MANO architectural framework.

The JOX architecture includes two main components: (a) JOX core comprises both JSlice (representing each slice as a set of models with a policy) and JCloud (host and control the underlying cloud
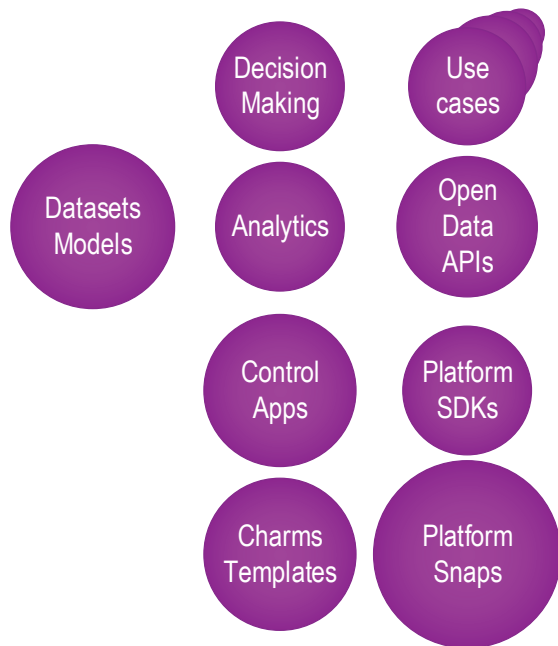
[2]https://snapcraft.io/
[3]https://jujucharms.com/

Figure 5: Store repository.



Figure 6: JOX platform.

resources) controllers to control slice and cloud resources respectively, and (b) the JOX plugin framework that enables different plugins for RAN, CN, MEC, and VIM to enable fast reactions like event handling and monitoring. Furthermore, it exposes the northbound REST API to enable several basic operations such as create, (re-) configuration, on each JSlice, connected to a JCloud.

## 3 MOSAIC5G EXAMPLE USE CASES

Based on the described Mosaic5G platforms, we hereby show their versatility for 5G community to rapidly prototype system and to test genuine ideas in four use cases: (1) e-health, (2) intelligent transportation system (ITS), (3) augmented and virtual reality (AR/VR), and (4) smart cities. These use cases can be applied by different providers (cf. Fig. 1) to compose the customized mobile network.

### 3.1 Critical Communications for E-health

5G aims to provide a flexible system in which different services with diverging requirements can be satisfied. Among these services, the critical communications take a prominent place to provide reliable voice and short text message. Moreover, new applications can provide large-bandwidth data and video communication services with on-demand high priority, such as delivering the live video contents from paramedic to the doctors that are remote or in the vicinity for e-health [9]. To this end, two issues are raised in terms of dynamic resource partitioning and slice prioritization. In the following, we conduct two corresponding experiments via applying the FlexRAN and RRM application in the Store repository.

For the first issue, we instantiate two slices, i.e., normal and public safety (PS) slices, and attach one commercial-of-the-shelf (COTS) user equipment (UE) to each slice for comparing their user plane performances. In the beginning, the RRM applies a 50/50 policy for these two slices, i.e., 50 percent of radio resources are
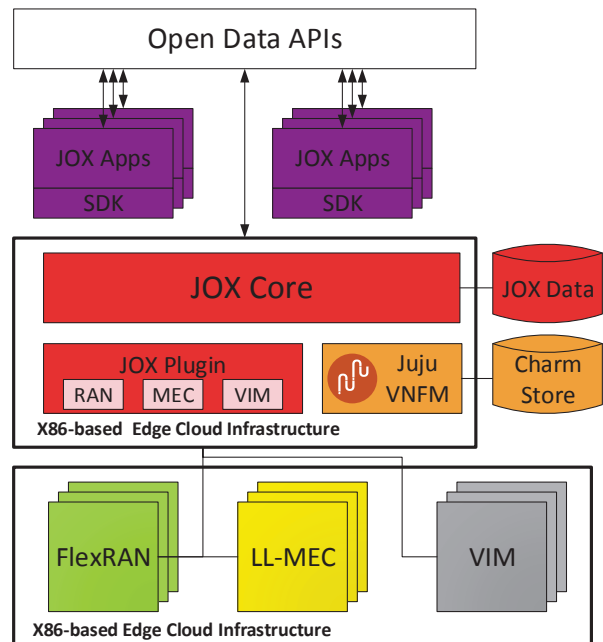
dedicated for normal slice and another 50 percent of radio resources are dedicated for PS slice as shown in Fig. 7a. We can observe that both UEs experience the same level of goodput (at the top) and delay jitter (at the bottom) during the first 40 seconds. However, the RRM policy is changed into 20/80 policy in the time period of 40 to 45 s, in which only 20 percent of radio resources are dedicated to normal slice, while 80 percent of radio resources are dedicated to PS slice. Hence, the goodput is significantly increased and delay jitter is largely reduced for the PS user.

Secondly, we show the impact of slice prioritization in Fig. 7b, in which there are three slices with respective priorities: (a) PS slice can preempt resources of other slices when the instant traffic load exceeds the supported dedicated radio resources, (b) best effort (BE) slice can increase its multiplexing gain by utilizing the unallocated resources, and (c) the wearable sensor (WS) slice sustains its dedicated data rate as it can neither preempt nor multiplex resources. We can see that the PS slice can adapt its data rate (left part) as a function of workload, i.e., from 3 Mbps to 6 Mbps, via preempting the resources from other slices, i.e., BE slice experiences a goodput drop from 10 Mbps to 8 Mbps. The same trend is seen in the delay jitter (right part), where PS slice experiences the minimum jitter as it has the highest priority (even its workload is increased). However, the WS slice suffers from the largest delay jitter due to its lowest priority.

### 3.2 V2X Communications for ITS

To attain the intelligent transportation vision, one key pillar is to enable vehicle-to-everything (V2X) communications to serve several vehicular services, such as autonomous driving, vehicle infotainment, and remote diagnostics and management (D&M) [2]. In that sense, the network shall be sliced in an E2E manner across different domains to provide real-time vehicular services. We hereby utilize
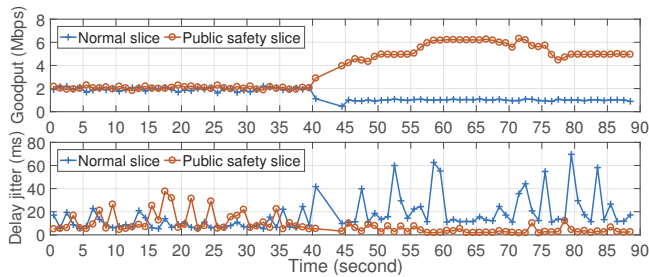
**(a) Dynamic radio resource partitioning**
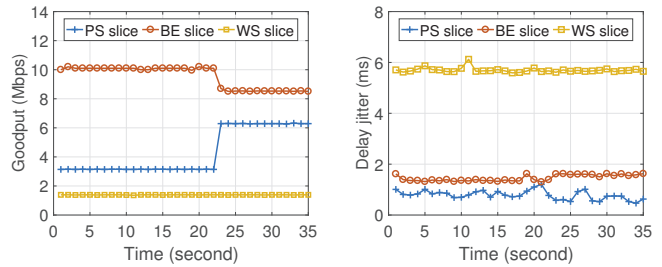


**(b) Slice prioritization**

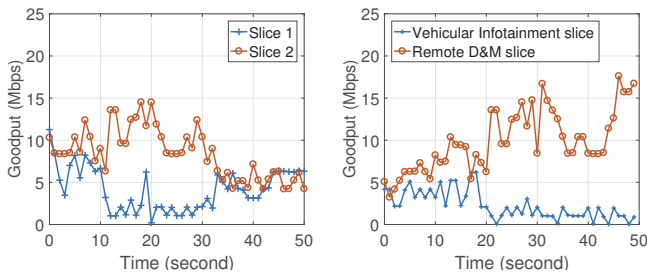**Figure 7: Impacts of dynamic radio resource partitioning and slice prioritization.**



**Figure 8: Uncoordinated (left) and coordinated (right) E2E network slicing.**



**Figure 9: Radio-aware video streaming with varying CQI.**

both the FlexRAN and the LL-MEC platforms to show how the coordinated programmability across RAN and CN domain can facilitate the V2X requirements. Specifically, two slices are initiated and the number of dedicated radio resources and switching bandwidth for each slice are allocated according to the applied RAN and CN policies, respectively. Further, we compare the user plane goodput in two policy enforcement manners in Fig. 8: (1) uncoordinated (left part) and (2) coordinated (right part) programmability.

For the uncoordinated case, three different sets of policies are applied at different time instances. The first policy is applied at $t_1$=10 s to dedicate 1 Mbps for slice 1 and 15 Mbps for slice 2 in both RAN and CN domain. At $t_2$=20 s, the second policy is enforced only over the RAN domain to dedicate 8 Mbps for both slices (i.e., equivalent to the aforementioned 50/50 policy), while the CN domain is kept the same as at $t_1$. We can see that there is no impact on slice 1 as its bottleneck is at the CN domain, while the goodput drops significantly for slice 2. Then, the third policy is enforced only over the CN domain at $t_3$=30 s to change the dedicated switching bandwidth to 6 Mbps for both slices while the RAN domain is kept the same as at $t_2$. These two slices have the same policy across the RAN and the CN domains, and thus their performances can not differentiate their services. In contrast, the coordinated case applies one policy at $t_4$=20 s over both RAN and CN domains to craft a vehicular infotainment slice with 1 Mbps and a remote D&M slice with 15 Mbps. The result shows that these two E2E slices can serve their individual services.

## 3.3 Radio-aware Video Streaming for AR/VR

Following the trend to bring the cloud computing capability toward the network edge, one interesting case is to provide different video qualities according to up-to-date radio information. Such concept is crucial to provide the AR/VR service with sufficient quality of
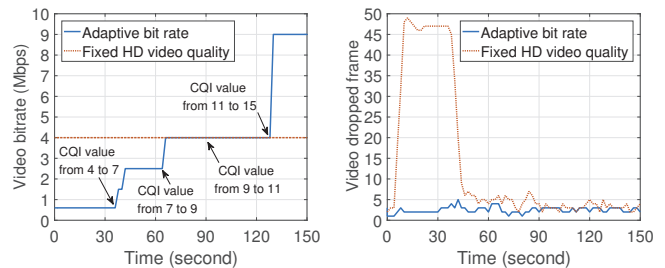
experience (QoE) for different end users [17]. Here, we use the FlexRAN platform with an adaptive bit rate (ABR) application to dynamically adjust the video quality based on monitoring RAN information, i.e., channel quality indicator (CQI). Such CQI values can reflect the maximum achievable goodput of each end user, that will be utilized by ABR application to provide video segments of different qualities, e.g., VGA, HD and FHD.

Practically, in Fig. 9, two experiments are conducted to show its impacts when varying CQI values from 4 to 15 in time. First of all, when fixed HD video is streamed, there are a number of dropped frames when CQI is low (first 60 seconds). Via applying the ABR application, the video quality can be correspondingly adapted from quarter VGA (CQI=4), super VGA (CQI=7), HD (CQI=9 and 11), to FHD (CQI=15) with fewer dropped video frames. We can observe that the ABR application can enhance video QoE in terms of lower dropped frames (low CQI) and higher video quality (high CQI).

## 3.4 Multi-service management and orchestration for Smart Cities

Smart cities is envisioned to culminate the Internet of Things (IoT) concept to connect various kinds of public utilities and infrastructures for responding everyday public services such as energy supply, waste management and traffic control [7]. To enable a wide-range of services in smart cities, 5G targets to bring flexible multi-service management and orchestration into the picture across several domains. In the following paragraphs, we show two particular cases leveraging Mosaic5G: (1) Dynamic spectrum management [1], and (2) Multi-domain service orchestration [5].

First of all, dynamic spectrum management is crucial to enable a multi-service architecture in order to utilize all available frequency bands according to the traffic workload, radio propagation environment, cell size and service requirements. Here, we can apply the
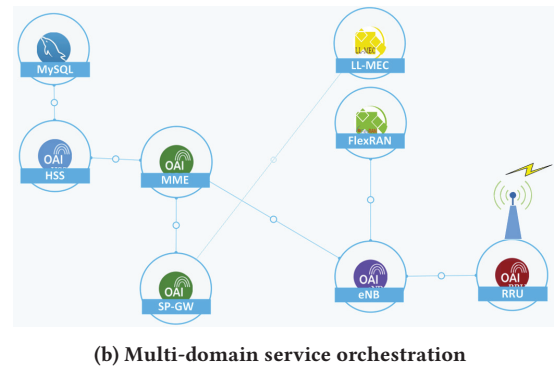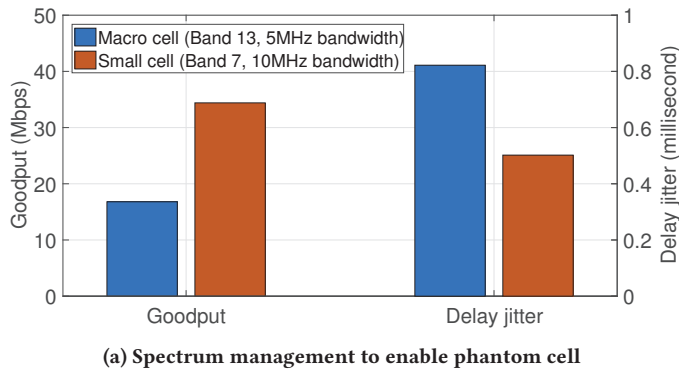
(a) Spectrum management to enable phantom cell



(b) Multi-domain service orchestration

**Figure 10: Multi-service management and orchestration.**

spectrum management application (SMA) in the Store repository to manage and process different spectrum management policies and rules pre-defined by various stakeholders (e.g., national regulator authorities, license owner). Moreover, SMA will interpret these policies and make decisions on the applied spectrum according to the sensing data provided by the FlexRAN platform (e.g., detected neighboring base stations information) as well as a set of rules identified by the service providers (e.g., smart city services). Finally, FlexRAN will take actions to deploy new spectrum over the underlying RAN. An example is shown in Fig. 10a that is originated from the phantom cell concept [12], in which the small cell is deployed at a higher carrier frequency with a larger bandwidth to boost the user plane performance (i.e., higher goodput and lower delay jitter), compared to the macro cell that serves in a wider area with a lower carrier frequency and smaller bandwidth.

Secondly, we apply the JOX platform to orchestrate the E2E service deployment. More specifically, the deployment of standard LTE service chain is orchestrated across multiple domains in Fig. 10b, i.e., eNB, MME, S/P-GW, HSS, MySQL for a new JSlice, in different environments ranging from either physical machine, container or virtual machine. Moreover, a monolithic LTE eNB can be split into a remote radio unit (RRU) that only contains cell common processing (e.g., radio frequency front-end), while rest processing are centralized for coordinated RAN processing. Further, both FlexRAN and LL-MEC are orchestrated to enable the network programmability over RAN and CN respectively. Note that service dependencies may exist when deploying chains, e.g., the relationship between MySQL and HSS cannot be built until the HSS is installed and configured. JOX can automatically handle such dependencies and conflicts exploiting Juju VNFM.

## 4 CONCLUSIONS

In this article, we introduce Mosaic5G as a community-led consortium for sharing platforms. Currently, it provides a number of software components including FlexRAN, LL-MEC, JOX and Store, spanning application, management, control and user plane in order to offer an open-source ecosystem for 5G research. Several Mosaic5G use cases are highlighted that can be further extended in the context of future 5G research directions. In the context of the evolutionary path toward 5G, Mosaic5G provides the R&D and prototyping framework for rapid proof-of-concept designs by presenting researchers and developers with an agile and flexible

prototyping environment with which genuine innovations can be achieved.

## REFERENCES

[1] Auon Muhammad Akhtar et al. 2016. Synergistic Spectrum Sharing in 5G HetNets: A Harmonized SDN-enabled Approach. *IEEE Communications Magazine* 54, 1 (Jan. 2016), 40–47.

[2] Claudia Campolo et al. 2017. 5G Network Slicing for Vehicle-to-Everything Services. *IEEE Wireless Communications* 24, 6 (Dec. 2017), 38–45.

[3] Canonical. 2018. Juju. https://www.ubuntu.com/cloud/juju Accessed: Jul. 5, 2018.

[4] Chia-Yu Chang et al. 2016. MEC architectural implications for LTE/LTE-A networks. In *Proceedings of the Workshop on Mobility in the Evolving Internet Architecture (MobiArch '16)*. ACM, 13–18.

[5] Chia-Yu Chang et al. 2018. Slice Orchestration for Multi-Service Disaggregated Ultra Dense RANs. *IEEE Communications Magazine* (2018).

[6] Chia-Yu Chang and Navid Nikaein. 2018. RAN Runtime Slicing System for Flexible and Dynamic Service Execution Environment. *IEEE Access* (2018).

[7] Panagiotis Demestichas et al. 2015. Intelligent 5G Networks: Managing 5G Wireless/Mobile Broadband. *IEEE Vehicular Technology Magazine* 10, 3 (Sept. 2015), 41–50.

[8] Xenofon Foukas et al. 2016. FlexRAN: A Flexible and Programmable Platform for Software-Defined Radio Access Networks. In *Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies (CoNEXT '16)*. ACM, 427–441.

[9] Cesar Garcia-Perez et al. 2017. Improving the Efficiency and Reliability of Wearable Based Mobile eHealth Applications. *Pervasive and Mobile Computing* 40 (2017), 674–691.

[10] Aditya Gudipati et al. 2013. SoftRAN: Software defined radio access network. In *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN '13)*. ACM, 25–30.

[11] Anta Huang et al. 2017. Low Latency MEC Framework for SDN-based LTE/LTE-A Networks. In *2017 IEEE International Conference on Communications (ICC)*. 1–6.

[12] Hiroyuki Ishii et al. 2012. A Novel Architecture for LTE-B: C-plane/U-plane Split and Phantom Cell Concept. In *2012 IEEE Globecom Workshops*. 624–630.

[13] Kostas Katsalis et al. 2018. JOX: An Event-driven Orchestrator for 5G Network Slicing. In *2018 IEEE/IFIP Network Operations and Management Symposium (NOMS)*. 1–9.

[14] Navid Nikaein et al. 2014. OpenAirInterface: A Flexible Platform for 5G Research. *SIGCOMM Comput. Commun. Rev.* 44, 5 (Oct. 2014), 33–38.

[15] Navid Nikaein et al. 2015. Network store: Exploring slicing in future 5G networks. In *Proceedings of the 10th International Workshop on Mobility in the Evolving Internet Architecture (MobiArch '15)*. ACM, 8–13.

[16] Peter Rost et al. 2016. Mobile Network Architecture Evolution toward 5G. *IEEE Communications Magazine* 54, 5 (May 2016), 84–91.

[17] Tarik Taleb et al. 2017. On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration. *IEEE Communications Surveys Tutorials* 19, 3 (thirdquarter 2017), 1657–1681.

[18] Mao Yang et al. 2013. OpenRAN: A Software-defined RAN Architecture via Virtualization. *SIGCOMM Comput. Commun. Rev.* 43, 4 (Aug. 2013), 549–550.