IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 6, 2021, accepted January 14, 2022, date of publication January 18, 2022, date of current version January 27, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3144534

Service Based Virtual RAN Architecture for Next Generation Cellular Systems

ENGIN ZEYDAN[1], (Senior Member, IEEE), JOSEP MANGUES-BAFALLUY[1], JORGE BARANDA[1], (Senior Member, IEEE), MANUEL REQUENA[1], AND YEKTA TURK[2]

[1]Centre Tecnològic de Telecomunicacions de Catalunya, Castelldefels, 08860 Barcelona, Spain
[2]Aselsan Corporation, 34906 Istanbul, Turkey

Corresponding author: Engin Zeydan (engin.zeydan@cttc.cat)

This work was supported in part by the European Union (EU) H2020 5GROWTH Project under Grant 856709, in part by the Generalitat de Catalunya under Grant 2017 SGR 1195, and in part by the National Program on Equipment and Scientific and Technical Infrastructure under the European Regional Development Fund (FEDER) under Grant EQC2018-005257-P.

ABSTRACT Service based architecture (SBA) is a paradigm shift from Service-Oriented Architecture (SOA) to microservices, combining their principles. Network virtualization enables the application of SBA in cellular systems. To better guide the software design of this virtualized cellular system with SBA, this paper presents a software perspective and a positional approach to using fundamental development principles for adapting SBA in virtualized Radio Access Networks (vRANs). First, we present the motivation for using an SBA in cellular radio systems. Then, we explore the critical requirements, key principles, and components for the software to provide radio services in SBA. We also explore the potential of applying SBA-based Radio Access Network (RAN) by comparing the functional split requirements of 5G RAN with existing open-source software and accelerated hardware implementations of service bus, and discuss the limitations of SBA. Finally, we present some discussions, future directions, and a roadmap of applying such a high-level design perspective of SBA to next-generation RAN infrastructure.

INDEX TERMS Service-based architecture, network interfaces, radio access networks, software.

I. INTRODUCTION

The concept of Service Based Architecture (SBA) in cellular networks is mainly used in 5G core networks, where its features have been incorporated into the 5G core (e.g., via a set of interconnected Network Functions (NFs) thanks to the role of Network Repository Function (NRF) and Network Exposure Function (NEF)) [1]. SBA in 5G core networks involves the transition from traditional telecommunication style protocol interfaces to Service Bus Interfaces (SBIs) where NFs communicate with Hypertext Transfer Protocol (HTTP) Version 2 via web-based Application Programming Interfaces (APIs). However, the concept of SBA has hardly been used in the context of mobile radio access networks (RANs). The main reason for this is that traditional RAN architectures have long tended towards more monolithic structures. This trend has pushed back Mobile Network Operators (MNOs) to provide flexibility to service-based structures in RANs. However, especially with technological developments such as virtualization and cloud, RAN architectures have started to adapt to structures such as Virtual Radio Access Network (vRAN), cloud RAN, etc. The vRAN concept aims to split Base Stations (BSs) into a Central Unit (CU) hosting the highest layers of the stack, a Distributed Unit (DU), hosting the physical layer (PHY) and a Radio Units (RU) hosting basic radio functions such as amplification or sampling [2]. vRAN implements the RAN functions using a generic computing platform and manages the RAN application virtualization using cloud-native principles. [3] Indeed, vRAN concepts to replace legacy base stations and software stacks are already in the open source development stage e.g. srsLTE* and OpenAirInterface (OAI)†. In addition to the development of such BS software stacks, there are other industry efforts to design fully open RAN architectures [4], and even conduct extensive field trials [5]. As commercial products using vRAN and cloud RAN are gradually being implemented in the real world [6], [7], the need for a transition

The associate editor coordinating the review of this manuscript and approving it for publication was Derek Abbott.

*https://www.srslte.com/
†https://openairinterface.org/

to SBA has also arisen in RANs to operate mobile networks in a cloud-native manner.

To provide services in the context of mobile network environments, software development traditionally uses Service Oriented Architecture (SOA) methods to connect various Information Technology (IT) application components with other components. Moreover, SOA is mature enough for large-scale deployments. Since NFs (including RAN software) are becoming more and more similar to IT applications (thanks to virtualization layers, programmable integration, etc.), similar tools and methods used in software engineering can now be applied to the networking domain. In recent years, there has been an increase in interest in the use and popularity of microservice-based architectures and container frameworks. This is due to several advantages, such as service registration, deployment, or the ability to scale application components independently and easily [8]. SBA aims to combine the advantages of both SOA and microservices-based architectures. When SBA is deployed in the network domain, MNOs can efficiently and easily control and deploy its own network services. SBA can also help MNOs to adopt new applications in IT and cloud across the network supply chain and service life-cycle management. For these reasons, the ability to apply SBA concepts to RAN development will be a key enabler to achieve significant improvements in network performance and monitoring, more flexibility in application and service development, sustainability and cost efficiency. It is expected that service-based RAN using SBA principles will maximize the provisioning and deployment of radio-specific network services and enable MNOs to create fast and efficient service provisioning pipelines. The development of service based on SBA principles may also entail the development of a variety of open source tools, libraries and components that can help accelerate the integration, deployment and use of service-based RAN.

Traditional RAN systems are built on monolithic building blocks and communication takes place between nodes within RANs. In traditional cellular RAN, the baseband units and multiple Remote Radio Units (RRUs) reside at the same integrated cellular site. BaseBand Unit (BBU) is responsible for NFs for the layers of the RAN protocol stack, and RRU for the transmission. However, this approach can be costly since each BBU must be deployed at each integrated site. MIn addition, the RAN protocol stack is implemented with black box software in proprietary hardware. Therefore, there is no standard logic split and no interoperability capabilities between different vendors. In disaggregated cellular RAN with CU-DU split (e.g., in 5G RAN and New Radio (NR) gNodeBs), a logical BBU architecture with logical split is introduced after Release 15 of The 3rd Generation Partnership Project (3GPP). BBU is split into CU and DU [9]. CU further splits into the control plane (consisting of Radio Link Control (RLC) and PDCP-C layers) and the data plane (consisting of Service Data Adaptation Protocol (SDAP) and Packet Data Convergence Protocol (PDCP) layers). DU (consisting of RLC, Medium Access Control (MAC) and PHY-Upper layers) is connected to a radio unit (consisting of PHY-Lower layer) via fronthaul (Common Public Radio Interface (CPRI)). This split architecture enables flexibility, scalability, cost efficiency in hardware and software implementations and deployments, coordination for better load management/adaptation and performance optimization [10].

3GPP has developed the 5G radio access network called Next Generation Radio Access Network (NG-RAN), which is developed independently from the 5G core with a SBA, but is interoperable. For this reason, in the initial phase of the definition of the 5G architecture (also taking into account the transition between 4G and 5G), a number of options were defined to identify the different variants resulting from the integration between the access network and the core (e.g. from option 1 to option 7). Release 15 of 3GPP introduced the Non-Stand Alone (NSA) architecture for Option #3, and StandAlone (SA) architecture for Option #2. A good summary of them is also provided in [11]. However, those architecture refer to deployment scenarios proposed for MNOs which are different from an SBA-based vRAN architecture as discussed in this paper. In the SBA-based vRAN architecture, all components of RAN are compatible with the microservice architecture, exposes a set of software functionalities (applied to the signaling context) and are fully softwarizable allowing the use of a number of Virtual Network Functions (VNFs) that support other functions in the architecture through the producer/consumer model. Additionally, SBA-based vRAN allows VNFs to perform many actions such as service registration, authentication, authorization, discovery of and connection to specialized services, and so on.

On the other hand, O-RAN alliance* aims to open up the RAN by disaggregating hardware and software and creating open interfaces between them. O-RAN provides a disaggregated strategy in line with Control and User Plane Separation (CUPS), where DU and CU stacks are actually separated [4]. Like SBA, O-RAN is cloud-native where typical network functions can be implemented as containerized microservices. However, the entities studied in O-RAN are not scalable in number of services by design. Therefore, in principle only limited services can be created in O-RAN (based on either CU and DU). Our SBA-based RAN architecture proposal in this paper is compatible with both centralized and disaggregated structures and also consists of many services as part of a microservice architecture. More comparisons of O-RAN and the proposed SBA-based vRAN are given in Table 2.

In this paper, we review and summarize SBA developments from a cellular RAN perspective, which, to our knowledge, has not been addressed in any other work to date. We present an overview of SBA-based solutions to enhance the capabilities of cellular RANs in terms of software architecture and application design principles, and focus on how they can be

*https://www.o-ran.org/

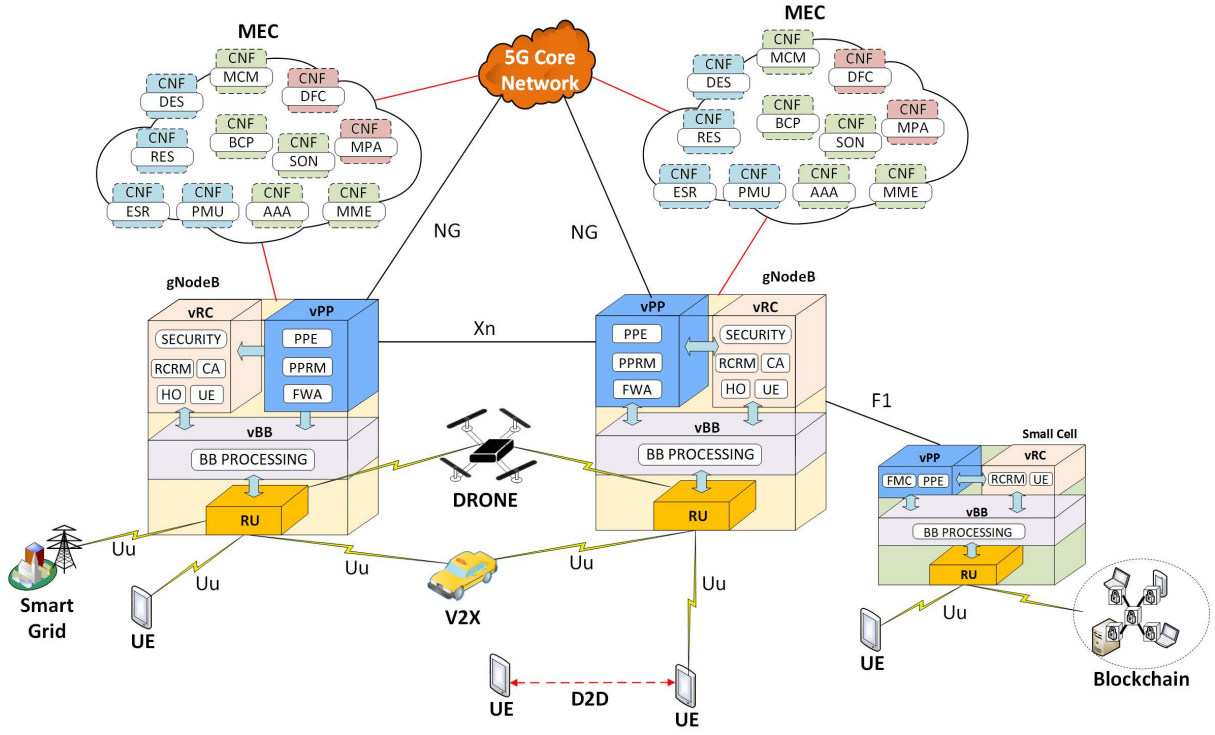


FIGURE 1. The high-level view of a cellular system with various services placed in the mobile nodes.

used in evolving RAN systems that can be hosted in either cloud or data center environments. The main contributions of this paper can be summarized as follows:

- We provide a clear definition of the concept of SBA, its general description and components, and discuss the state-of-the-art technologies that can be used for cellular RANs,
- We explore the adaptation of SBA in vRANs which is a novel application of SBA principles to RANs from a software perspective. We also map the RAN implementation aspects with main SBA components, interfaces and services.
- The potential of SBA-based vRAN is explored by mapping between the requirements of 5G RAN functional split options and the existing Service Bus (SB) open source software and accelerated hardware implementation results currently available in the literature to assess the feasibility in meeting the requirements of each split.
- The current limitations of various RAN options, their advantages and disadvantages and future research directions for realizing SBA with RAN services are provided.

The remainder of the paper is organized as follows: Section II provides the background SBA and its development in RAN. Section III describes the general framework and components of SBA. Section IV discusses some of the features for managing services during the network service lifecycle. Section V-C compares existing SBA-based solutions with the requirements of 5G RAN functional split and discuss potential benefits, challenges and future research directions and finally Section VI provides the conclusions of the paper.

II. SBA & SERVICE-BASED RAN SYSTEMS

Three main components of RANs are available in the literature: (i) vRAN, (ii) Open RAN and (iii) Open Radio Access Network (O-RAN). *(i) In vRAN,* RAN functions are virtualized (especially higher and lower layers of BBU) so that NFs can move from proprietary hardware to Commercial off-the-shelf (COTS) cloud platforms. Software implementation of virtual functions can now be done in COTS servers. Together with vRAN, virtualized BBUs can be connected to RRUs using CPRI or enhanced CPRI (eCPRI) protocols. However, in vRAN the interfaces are still proprietary, i.e. not necessarily open, and the radio unit hardware is still proprietary. *(ii) Open RAN* is a general term for an open RAN architecture. *(iii) O-RAN* refers to Open RAN, which is standardized by the O-RAN Alliance. The architecture of O-RAN includes split gNodeB architecture defined by 3GPP, so these two architectures complement each other. O-RAN aims at virtualization, open interfaces, interoperability and intelligence. In O-RAN RUs and DUs are connected via the eCPRI protocol with a new low-layer-split called Option 7-2x.

A general reference network where SBA and its corresponding services can be used is shown in Figure 1. The general Cloud Native Function (CNF) functionalities can be divided into two main categories: core- and radio-specific CNFs and application-specific CNFs. *In core- and*

radio-specific CNFs, RAN-related functionalities are available in BS within Virtual Radio Controller (vRC), Virtual Packet Processor (vPP) and Virtual BaseBand (vBB) and can be either at user equipment (UE) level or at network level. Mobility and interference level management is done at the UE level. The radio resource situation (conditions, usage, availability), backhaul conditions/availability (e.g. BS neighbourhood change), network quality-of-service (QoS), automatic/on-demand scale-in/out, RAN part of network slicing and network support (for Vehicle-to-Everything (V2X)) are at the network level [12]. vAAA (virtual Authentication, Authorization, Accounting), vMME (virtual Mobility Management Entity), vSON (virtual Self Organized Networks) are also in the *core and radio CNF* category. *In the application CNFs* some sample functionalities are vBCP (virtual BlockChain Processing), vRES (virtual Renewable Energy Sources), vDFC (virtual Drone Flight Control), vDES (virtual Distributed Energy Storage), vESR (virtual Electricity Substation & Rerouting), vMPA (virtual Media Processing & Analysis) [13].

RAN functionality can be used in different places depending on service needs and deployment characteristics, as shown in Figure 1. For example, NFs that are parts of User Plane (UP) services can run on servers optimized for packet processing in gNodeB, whereas cross-country network services are intended to run on edge cloud infrastructure, e.g., Multi-Access Edge Computing (MEC).

A. MOVING FROM NODES TO SERVICES IN RAN

In long term evolution (LTE), some initial steps have been taken towards SBA, where service discovery mechanisms have been introduced to set up intra-system communications [14]. 5G NR (especially in the development of vRC and vPP) also takes into account the decomposition of the software into smaller units depending on the characteristics of the use case and the development requirements [15], [16].

Figure 2 shows the high-level structure of the traditional and the vRAN system implemented with SBA principles and outlines the main differences. In traditional embedded platforms of RANs, as shown in Figure 2a, the development and deployment units are "nodes", which consist of platform and application software, and hardware. They are all managed by a common Operation, Administration and Management (OAM) system and described in a node-specific managed object model [17]. The coupling is quite strong in this traditional system, as all executables share the same operating system (OS) and platform layers. The design is also monolithic, meaning that all RAN application-related components and services are encapsulated in one package. At the same time, the embedded nature of the system also means that assumptions can be made about the environment, such as exactly what capacity is available, what other software is running on the same processor, and so on. The result is a rigid and rather inflexible product, but once integrated it delivers the promised performance with high reliability. The communication methods and principles

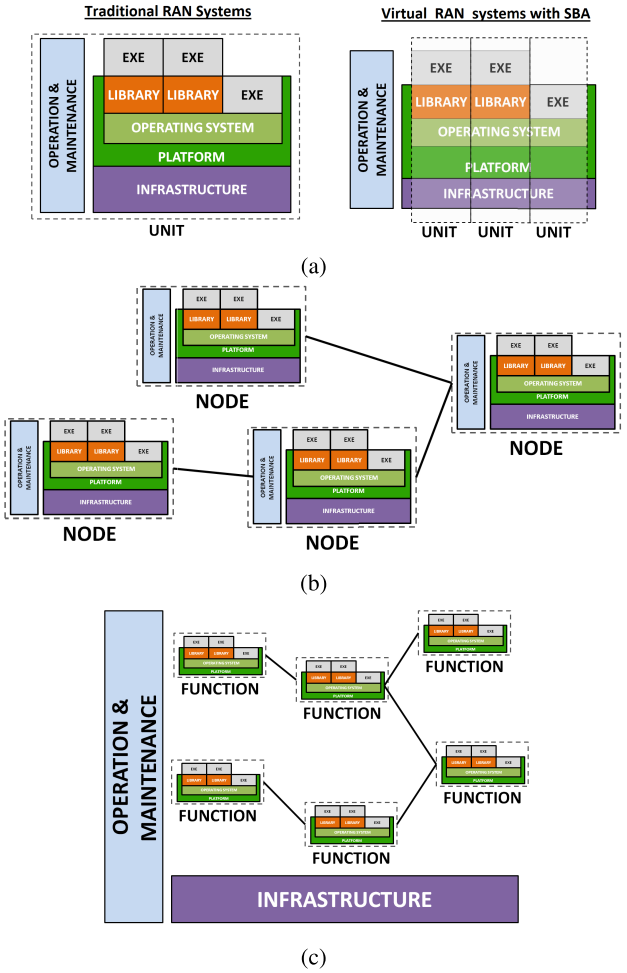


FIGURE 2. High-level node and functional level structures (a) Traditional RAN systems and vRAN systems implemented with SBA (b) Traditional RAN systems are node-centric (c) vRAN systems implemented with SBA are service-centric.

are often very different within the node and between the nodes.

Figure 2b and Figure 2c show the high-level view comparisons for the node-level and service-level structures and the interconnection principals of the traditional and SBA-based vRANs respectively. Figure 2b shows that traditional or reference-based RAN systems that are *node-centric*. Figure 2c shows that virtual RAN systems implemented with SBA principles are *service-centric*. Each defined function in Figure 2c are interacting and independently customizable software components or services that can invoke many other functions or services. In Figure 2c, services are contained in CNFs and the most likely connectivity options between them are *service-centric*. Note that the service blocks are simplified and subject to change, and that other deployments may be of interest, such as deploying L1/L2 on cloud hardware. In addition, packaging and deployment flexibility and reusability the key aspects of this new BS software in SBA [18], [19].

The software of the future RAN is expected to support different deployment scenarios and product variants

in heterogeneous embedded and cloud environments only through configuration changes. Container-based systems (e.g., Kubernetes) run software workloads on hardware to optimize deployment costs and can simplify the Life Cycle Management (LCM) of the CNFs. Higher quality software can reduce costs and speed up development time. This can help meet MNOs' expectations for a flexible, containerized and service-based 5G network infrastructure [20]. SBA consists of the provisioning/deployment aspects such as container-based infrastructure and Development and Operations (DevOps) [21]. Together with microservices, DevOps can minimize the coordination between development and operations teams [22]. On the other hand, from a DevOps perspective, SBA performance in future mobile networks requires well-defined performance metrics for microservices and data center resources with real-time monitoring capabilities [23].

B. SBA-BASED RAN FUNCTIONS

In a typical SBA-enabled RAN system there are three main components: (i) RAN service producers, (ii) RAN service consumers and (iii) RAN service brokers (including registries and repositories). These components help to provide RAN as a service in SBA design. As a result, RAN-related network services or functions may be provided as a service to external parties. Note that the producer or consumer of a RAN service can be any component of a service, a completely different service, third party applications, the transport or core network of a single domain, or multiple administrative domains (e.g. different MNOs interacting with each other in a federated network [24]).

One of the key ideas of a SBA is the principle of providing (authorized) consumers with highly customized network services that are leveraged through virtualization and software-defined networking techniques. For this reason, SBA-based vRAN should also be generic enough to enable a wide range of applications (e.g., for vertical markets automotive, e-health, etc.) while supporting new generation use cases and sophisticated service requirements. In addition, it is of great importance that the principles and solutions are aligned with other Cloud Native Software initiatives.

Of course there are certain basic functions that must initially be in place to enable the development of services, such as service discovery and service communication functions. However, the decomposition of the system into services will be gradual and based on requirements of the application, such as scaling and performance characteristics will not be dictated by the architecture from the beginning. Another important aspect is that initially the system software will most likely be based on a few larger services covering a wider range of functionality. As the system matures and the application needs certain features, the service will become more specialized and smaller in scope, while the individual services will become more fine-grained and therefore increase in number [25].

C. SERVICES IN VIRTUALIZED RAN

A high-level view of the target system with SBA is shown in Figure 3. During the implementation phase of this architectural design, the leading figures of SBA (e.g. MNOs, Service Providers (SPs), telecommunication vendors, etc.), the desired features, capacity, deployment scenario, and characteristics driving the decomposition for system development must be clearly identified [26]. In addition, for security in a virtualized environment, the service level agreements for MNOs need to be specified to understand, asses and determine the level of security of the services provided [27].

In Figure 3, the services within the vBB, vRC, vPP are shown. From the RAN perspective, RAN resources and features (e.g. RAN data processing services, MEC, V2X, optimization and performance improvement services, network deployment, etc.) are controlled by the Radio and Cloud Resource Management (RCRM) service. The procedures for UE such as attachment and bearer establishment, etc. are provided by the UE service within the vRC. The encryption and integrity operations are provided by the Security service. Handover (HO), UE (services comprising all RAN protocol layers PDCP/RLC/MAC/physical layer (PHY)) and Carrier Aggregation (CA) are the other services within the vRC. In vBB, there is only one service, which is the BaseBand Processing (BBP) service. In vPP, the Packet Processing Engine (PPE) service is responsible for all packet processing operations. The resources of PPE are controlled by the Packet Processing Resource Management (PPRM) service. Fixed Wireless Access (FWA) is an optional service that can be used when fixed wireless access is required for connectivity. It is a way of providing wireless connectivity through radio links to provide wireless internet access where the cost of laying fiber is costly.

Figure 3 also illustrates that in addition to functionalities vRC, vBB and vPP mentioned in Section II, there are also functionalities such as Artificial Intelligence (AI)/Machine Learning (ML) (e.g. for intelligent Self Organizing Networks (SONs) and network automation), interfaces as defined Section III-C and complementary middleware services as defined in Section III-D. The target characteristics of the described SBA-based system are summarized in Table 1.

III. SBA FRAMEWORK COMPONENTS

A. SERVICE UNIT

The Service Unit (SU) can be described as the smallest unit in the overall SBA. A network service may consist of several SUs. The design of the SBA using SUs ensures that the RAN services are loosely connected and flexible within an independent lifecycle framework. SU has standardized interfaces that allow external entities to communicate and interact with it. By standardizing SU, it is possible to build a supporting ecosystem ranging from development environments and tools to deployments. SU can also be packaged in various ways. An example is a Virtual Machine (VM) image in a Network Functions Virtualization (NFV)

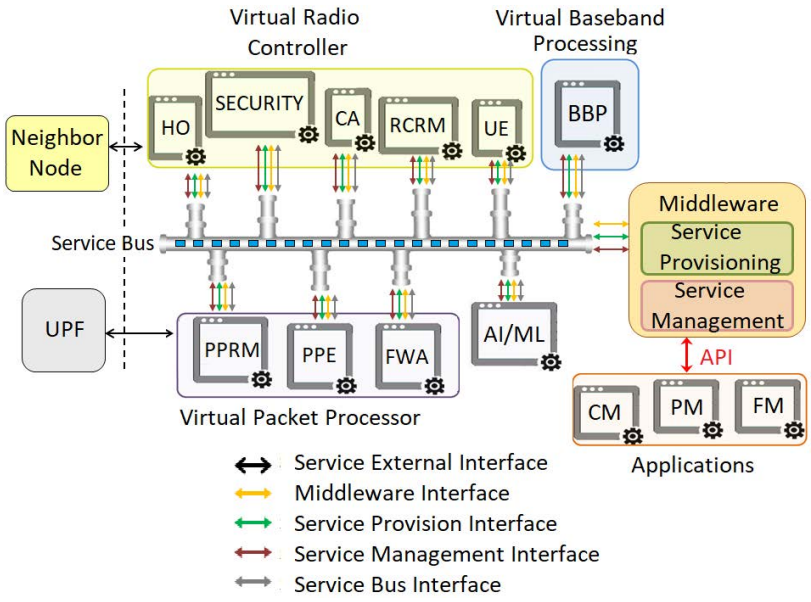


FIGURE 3. Software functions as services in service-based virtual RAN environment.

TABLE 1. Characteristics of the target system with SBA.

Definition	Characteristics	Related Works
Development	— Lower development and integration costs by reusing software in different nodes as source code. — Decoupled development lifecycles for different parts of the software. — Modularization of software which in turn increases parallelism in development.	[4], [8] [9], [12] [15]
Deployment	— Supports distributed deployment across multiple environments & sites. — High quality to support flexible product packaging. — Supports continuous integration and deployment.	[2], [10] [6], [15] [7]
Infrastructure	— Customizable, expandable and scalable for large and small environments with low overhead. — Small footprint with reusable software in the cloud and embedded systems. — Increase organizational flexibility by adopting trusted common solutions across the software base.	[5], [14], [16] [13] [11], [17]

based infrastructure or a Linux container image in a container-based environment.

The goal of using SU to package services during network startup on a physical server is to exploit the lightweight nature of container technology. The container can then be loaded and launched on a physical or virtual host, depending on the various deployment scenarios. This approach provides maximum flexibility, streamlines development and release workflows, and simplifies product handling. In addition to the container itself, SU includes a metadata file that describes dependencies on other RAN services and infrastructure. SU templates are used to describe information related to the RAN service. This includes the version, base image, maintainer and dependencies on other RAN services or the overall infrastructure. An example may be a Virtual Network Function Component (VNFC) template derived for SU templates [28].

B. SERVICE BUS

For certain use cases, the amount of information and high update frequency may overwhelm the Representational State Transfer (RESTful) interface between connection points. Therefore, for redundancy and delay minimization reasons, a peer-to-peer messaging broker referred to SB in SBA, is required to connect different RAN service producers to their corresponding consumers through a unified interface. A RAN functionality can be accessed directly from another function without going through another node. The reason for this is that SB works asynchronously and provides abstractions of service and infrastructure.

Abstraction is critical to separate infrastructure complexity from features that provide RAN flexibility without impacting individual network service consumer. SB is also responsible for Service Interface (SI) selection by applying the desired intelligence and features to the routing and load balancing algorithms, so that those consumers using the SBA service without permissions cannot directly access a particular SI through the SB. The features such as auditing, monitoring, security, standardized logging, session tracking, and bindings can all be embedded in the API of the SB as a plug-in. The SB uses a service discovery service (e.g. Namerd) to help consumers find and connect to producers.

C. INTERFACES

We defined five tasks in the architecture (external communication, middleware interactions, provisioning, management,

and data communication as shown in Figure 3) and designed the corresponding interfaces for the SUs interactions in RAN.

The interfaces are designed to ensure strict separation between tasks for the RAN depending on the requirements. Using a single interface for all the tasks can be another alternative architecture, but for ease of implementation, software development, and fault analysis, it is better to have an optimal number of interfaces that meet the requirements of the tasks. These defined interfaces for SBA in RANs, as shown in Figure 3, are described below.

i) Service Management Interface (SMI): A mandatory interface for the web-based LCM and runs according to the Ve-Vnfm reference point of European Telecommunications Standards Institute (ETSI) [29]. This interface provides service consumers with a resource model in which the capabilities of SU are represented and controlled. The service consumers of this interface are middleware services. Interfaces designed to be managed by a Management and Orchestration (MANO) CNF Manager, for time synchronization, etc., are interfaces for managing services.

ii) Service External Interface (SEI): Specific interfaces such as a UP interface or a HTTP interface exposed by the SU are SEIs. Interfaces such as NG, N2, etc. are external interfaces for services.

iii) Middleware Interface (MI): These are the interfaces for communicating with the middleware services, i.e. this interface connects the middleware services via a SB or HTTP/Representational State Transfer (REST). The MIs are abstracted from the application via an API and an consumer side proxy. The abstraction is required to enable the standardized service implementation and does not change without affecting the consumers service.

iv) Service Bus Interface (SBI): Not a mandatory, but the most commonly used interface associated with the standard asynchronous peer-to-peer messaging mechanism (namely, the SB).

v) Provision Interface (PI): The PI is mandatory and is used to provision SU. It can be a Docker environment variable if the SU is container-based, or cloud-init if the SU is VM based. Examples of required parameters are Uniform Resource Locator (URL) or the Internet Protocol (IP) address of the Service Registry and Discovery (SRD) function [30]. PI can be used for tasks like slice management.

D. COMPLEMENTARY MIDDLEWARE SERVICES

Within the development of SBA, the need for a complementary middleware arises. In particular, a new middleware platform is needed to support multi-tenant cloud environments as well as embedded environments. The main purpose of the middleware is to provide the foundation for SUs to be implemented for SBA services. Middleware is responsible for loading SUs on hosts including virtual hosts in the container-based environment and implementing functionality for non-SBA services such as backend databases, log servers, message queues, configuration management (CM), performance management (PM), and fault management(FM), etc. In addition, middleware must support LCM activities of SBA services on embedded and cloud hosts as shown in Figure 3.

Middleware can be a great support, especially when it comes to security services, e.g., when a node needs to connect securely over Internet Protocol Security (IPSec) to core network (CN) or containers need to communicate with each other over Transport Layer Security (TLS). In addition, vRAN nodes must have Hardware Security Module (HSM) modules within themselves. This HSM can be used to perform some security operations such as digital signature to authenticate the communication or to store the private keys. Then, the hardware module within the vRAN architecture can run over this middleware.

IV. SERVICE MANAGEMENT & LIFECYLE OPERATIONS

A. SERVICE REGISTRATION

Figure 4 shows the roles of SI and RAN service consumer in the RAN service consumer pool for an exemplary radio resource SI. Note that a SI can also be used as both a RAN service producer and as consumer. The first step in Figure 4 indicates that the SI program must register the service with the SB to be published in a SRD via the Service Registry (SR) agent. SR in the message broker allows RAN service producers to register and track their RAN services. Since RAN services consist of one or more SIs, the SR must also be updated when SIs are created, updated, or removed.

During registration, the vRAN or one of its components being instantiated first provides information about the RAN services. This information includes the capacity of the RAN service, the load-balancing support of the RAN service, the endpoints to reach at a given NF, the service quality metrics, etc. When a RAN service consumer intends to initiate an attachment request to a RAN service that uses one of the SB APIs in a connection-oriented interaction, the RAN service consumer must first specify some parameters such as the service name, preferred API versions, visibility domains, etc. At an earlier or later time, one or more SIs (depending on the architecture chosen and the type of network deployment at the site) registers to the RAN service along with the requested parameters. The RAN service consumer then receives a notification when at least one of the SIs becomes available. Later, SB requests a connection request with parameters describing the session requirements. After this step, SB selects the appropriate SI depending on the input parameters, location, traffic load conditions, etc. and forwards the confirmation message to the RAN service consumer. After acknowledgment by SB, the RAN service consumer can now establish a session to the RAN service producers.

An example of this procedure is the requests to establish a radio bearer session from users to BS. In this case, users may register with the UE label of vRAN. After UE service registration, users will also need to register to the SECURITY service for the encryption operations.

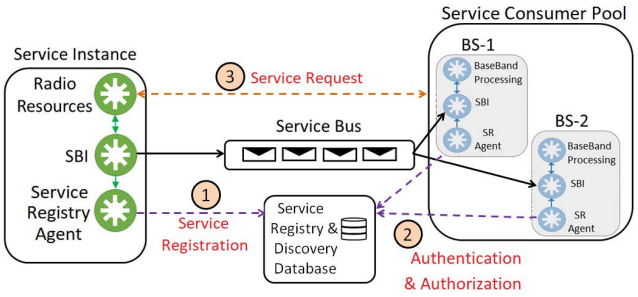


FIGURE 4. Baseband processing services (belonging to different base stations) in the service consumer pool discover and subscribe to the radio resource SI provided at the antenna site.

B. SERVICE AUTHORIZATION, AUTHENTICATION & PRODUCER DISCOVERY

After RAN service registration is completed, RAN consumer service authorization & authentication and RAN service discovery steps follow as given in step 2 of Figure 4. These steps allow tracking which RAN service consumer accesses which RAN services to verify their authentication credentials, privileges, and corresponding network service addresses/endpoints. During the service discovery process, multiple SIs publish their IP addresses/URLs (depending on the implementation options) and this published data is then written to a SRD for further use. Subsequently, RAN service consumers read from this database to access the desired RAN service. Also, any RAN service consumer in the network can reach the desired SIs.

In step 3 of Figure 4, a session connection request is sent with parameters describing the session requirements. After this request, the SB searches for an appropriate SI that can accommodate the request based on the properties such as input parameters, location of the request, traffic load, etc., and forwards the connection request to the appropriate SI. The SI acknowledges the connection request and the session is established. Later, any number of sessions can be established between a RAN service consumer and a SI and one of the peers can close the session. The information is exchanged between the SI and the RAN service consumer pool via the SB.

The SB provides an API that includes session multiplexing and session isolation to have an overlay structure over the transport infrastructure. SB mainly uses two structures for the transmission of application-defined message types: the first is the connection-oriented mode and the second is the connectionless mode. Each interface connected to the SB is a separate entity from the perspective of the service discovery mechanism. Thus, with this capability, the current legacy programs can publish different RAN services with different capabilities through RAN interfaces.

Some typical radio network information that is ingested by the radio resource SI and can be consumed by RAN service consumers are: Measurement data at UP based on 3GPP specifications, UE context and associated radio access bearers collected at the appropriate granularity, e.g. per cell, per UE, etc. Messages between the RAN service consumer and the SI are as mentioned above, user-level messages and are in binary format according to SB. In addition to session-based communication, the SB can also support connectionless Remote Procedure Call (RPC)-style messaging initiated by the RAN service consumer.

SRD database given in Figure 4 stores SR agent data in a hierarchical namespace similar to a file system or tree data structure (practical implementations include Apache ZooKeeper or BookKeeper). It acts as a metastore library and is used by all NFs within RAN to retrieve consumer data and context. Each time a new label (a category or feed name under which messages can be stored and published) is created, a NF record should first be created in the SRD database. Essentially, the SRD database helps to dynamically create, stop or scale network services. The RAN service consumer is later connected to a service via the SB. The SB will use the SR agent to retrieve relevant SI. The SB will later connect to all relevant endpoints, execute the SI selection and manage the connections between RAN service producers and consumers.

C. RAN SERVICE LABELLING

RAN service labels can identify the NF naming, i.e., the identities of the physical or virtual host running the SI, its supported features, the API version, the container version and so on. All of these identifiers can be used to represent the SI. Some of these labels can be specified by the application, others by the middleware.

For example, if a RAN service consists of sub-services that may be instantiated in domains controlled by different MNOs (e.g., small cells and macro cells may belong to different MNOs), the proper interconnection of all components in the target domains should be configured consistently over the F1 interface. If a small cell wants to pass information to a macro cell via F1 interface, as shown in Figure 1, then it must select the label corresponding to F1 interface and send a request containing commands and instructions about what it wants to do via SB. In this way, the message formulated by small cell addressed to a macro cell is published in SB to be consumed later by the corresponding macro cell.

D. FILTERING MECHANISM

Filtering is used to exclude some SIs for a RAN service consumer to reduce the total number of peer-to-peer connections during runtime. Therefore, static domains are defined when deploying the RAN. Both RAN services and their consumers may be associated with multiple domains. These static domains are used to separate services in the networks such as data centers and CNFs, before the service discovery process. The geographical location, QoS parameters, latency, hops between the RAN service consumer and the service within the domains can be used as reference points to reduce the number of available SIs.

An example RAN SI that can be filtered out by these reference points is the Automatic Neighbour Relationship (ANR) feature. Labels can also be used in the filtering mechanism of service discovery. For example, the filtering

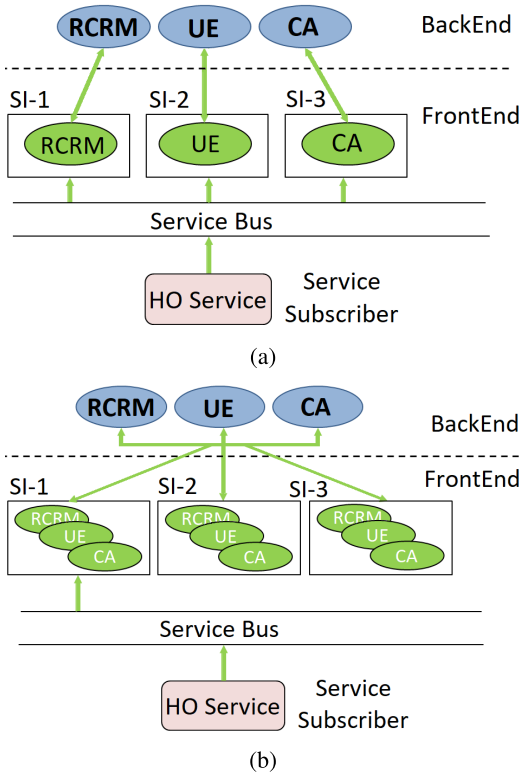


FIGURE 5. Different methods of providing HO service to consumers in vRAN (a) A single SI provides the resource over SB. (b) Load balancing is supported by holding multiple resources in different SIs of the front-end.

mechanism can help filter out the SI that performs the UE functions that may have many interactions with other services to reduce traffic and system complexity. Labels provided by applications can also be used to filter out SIs that rarely change. Another example is filtering out SIs that control a cell of a BS or a SI that communicates with other RAN services via RPC or session calls.

There can also be different QoS patterns in a vRAN node. Two examples can be given for this scenario such as priority assignment and dedicated bandwidth provisioning, but the QoS patterns can be much more numerous. Thus, if a UE is registered to a priority SI label, then it may not be registered to a dedicated bandwidth SI. In this case, a filtering mechanism may be used to filter out the QoS patterns for that specific UE or UE group.

E. LOAD MANAGEMENT

In order to provide load balancing for SB, load-aware algorithms (e.g., max-min, proportional fairness, etc. as opposed to simple round-robin) must be used before traffic arrives at SB. The traffic load at each SI can be measured as Central Processing Unit (CPU) consumption rate, relative number of initiated Input/Output (IO) sessions, etc. Depending on the incoming traffic load or the level of RAN service request rate, load feedback can be sent from SIs to RAN service producers at variable intervals [31].

Figure 5 shows two variants of using labels to consume a particular RAN service, e.g., an HO service for UEs between gNodeBs. In Figure 5a, there are three SIs and each SI controls a different resource with labels of *RCRM*, *UE* and *CA* respectively. When a HO process is needed for a UE, the HO service registers to the resources of *SI-1*, *SI-2*, *SI-3* and the SB selects the appropriate SIs. The scenarios in Figure 5a are ideal for rural deployment cases where mobility of users is not frequent and there are fewer HOs between gNodeBs.

Figure 5b on the other hand, shows the load balancing scenario when there are resource states in one back-end and three front-ends SIs. Since each front-end SI can handle resource requests with labels *RCRM*, *UE*, and *CA*, this scenario is ideal for urban and congested locations with high mobility of users that also require load balancing. In this scenario, the demand for the HO service is high and all three resources at the back-end should be immediately available at the SIs front-end in case RAN service needs them. The SB then load balances the traffic across all front-end SIs. Considering these two scenarios, the scenario with only one SI is easy to implement but inefficient for the high mobility areas, while the scenario with load balancing is hard to implement in software but efficient for high mobility scenarios.

F. SI SELECTION

A SI can be assigned more than one label via SB. Some examples of labels are cell IDs or UE contexts, for which SI is responsible, the flavor of SI (whether it is optimized for latency-sensitive or compute intensive applications), etc. At the same time, consumers of SRD can register to a filtered label list on each SI, so that interconnection network can be further simplified. The selection of SI is one of the main tasks performed at runtime by SB, based on the needs of the RAN service consumers and the current state of the system as a whole.

During the SI selection process, the SB first performs filtering to simplify the large number of SIs to a small subset that meets the RAN service requirements for consumers. The filter selection may be based on the maximum latency of the transport network and the available bandwidth of the connected BS to another SI, depending on the maximum geographical distance or the congestion status the SI [32]. After filtering, a load balancing algorithm is applied to the remaining SIs to stabilize the traffic load over all available SIs. The load balancing algorithm shall minimize the number of SIs used while avoiding overloading the SIs as described in Section IV-E.

G. EVENT MANAGEMENT

SB must be based on event-driven architectural principles, the service producer has more responsibility for maintaining state and communicating updates to the service consumer. In case of failures, the SB informs the application on the service consumer side about failed sessions by using RPC calls. It also informs the application that the service is

no longer available if all SIs in the environment fail [33]. Event ordering, message logging, call path tracing and other types of service interactions in RAN (e.g. UE mobility, session or connectivity management, radio channel quality degradation, modulation coding index monitoring, radio resource management, etc.) can also be managed through the SB. An information message mechanism implemented at the SB level indicates the procedures of the service interactions. At the same time, the consumer should be able to ensure that an appropriate message decoding process is enabled on its side to track events (e.g., service provisioning and resource sharing via a blockchain-enabled RAN [34], [35] or Unmanned Aerial Vehicle (UAV)/drone-assisted RAN area coverage optimization [36]) that can be managed by SB.

H. SERVICE VISIBILITY, UPDATE & DEREGISTRATION

Basically, the services of RAN can be published in either global or local domains. The global domain contains services that have consumers in other CNFs. As a best design practice, common RAN services are deployed in multiple CNFs so that they are accessible by RAN service consumers in different domains. Note that as next-generation networks are deployed, there may be a large number of SIs in RAN deployment areas with an excessive number of antennas. Due to the geographical location of these antennas, it is not necessary to publish most SIs for consumers within the global domain, as there will be no interaction between most of them. Moreover, the increased number of SIs in the global domain is difficult for the service discovery system to manage and creates a large overhead. Therefore, to reduce the number of peer-to-peer connections using SB, the number of SIs visible to each consumer in the global domain must be either limited or in the local domain [37].

From the point of view of SBA, all RAN services are treated equally as long as they are reachable over the network. For example, a RAN service consumer may need to use a SI that manages the 5G NG connection to a BS. This can be any particular RAN service consumer, either component of an entire service, e.g. ehealth monitoring, Industry 4.0, automotive or media or a completely different network service requiring 5G NG interface connection in inter- or intra-Point of Presence (PoP) network through a SI. However, in cases where this particular RAN service is published in the wrong domain, the NG interface cannot be reached by the SI and the connection is rejected. Another example is the SI that manages the ANR service. To connect to a ANR service, the SI of the baseband service belonging to that BS must be reachable from all SIs of the other baseband services in the neighboring domains. The same applies to the SI of a HO service.

There may also be RAN services positioned at the boundaries of local domains that are within the global domain. These services are visible outside the deployment entity in the local domain. They are published in the global domain but can connect to local services, that is they can have services for RAN service consumers in more than one separate local domain. For example, customer-facing RAN services are the border services that that have open interfaces to O-RAN, 3GPP, etc. architectures and their corresponding consumers. As local services are deployed together with the corresponding cross-border RAN services, API compatibility rules could be relaxed, but this may increase complexity. For this reason, strict API compatibility is critical for cross-border services. In addition, updates to RAN services may impact and require coordination with other CNFs. When there are updates to service features, the NF profile must be updated in the SRD database shown in Figure 4. If the RAN service is no longer available, the deregistration of the service is also performed and this unavailable RAN service must be deregistered from the SRD repository.

V. VALIDATIONS, ADVANTAGES, CHALLENGES & FUTURE DIRECTIONS

A. SBA-BASED RAN SUITABILITY FOR 5G RAN

In this section, we would like to investigate the state-of-the-art in terms of upper performance bounds and the behaviour of existing service bus performances. For this reason, we would like to know how well the service bus can support microservices over inter-service communication and how their performance behaves when managing a large number of requests in complex and heavily loaded network scenarios. The main benchmark Key Performance Indicators (KPIs) are system throughput and system latency, which are primary performance metrics in event streaming systems in production environments*. System throughput is defined as the average producer throughput up to which consumers can keep up without increased backlog. Similarly, system latency is defined as the end-to-end latency for a message to pass from producer to consumer.

Microservices themselves can bring some disadvantages such as a significant risk of failure in the communication between services or even some complexity in managing a large number of these services. Therefore, concerns about the application of SBAs as an essential technology for 5G infrastructures need to be confirmed. For this reason, issues such as system latency, throughput, load balancing, etc. need to be addressed through rigorous experimentation and comparative performance evaluations that take these complexities into account. To validate how well the proposed architecture can behave in 5G vRAN and potentially improve the performance of 5G infrastructures, we first investigate the functional split requirements of 5G RAN [38] and then attempt to map the existing service bus benchmark tests and statistical results from the literature [39]. The technical report in [38] shows an overview of the functional split and the corresponding transport latency and throughput requirements for each split option in RAN.

5G RAN functional split and their corresponding required latency and throughput values are shown in Figure 6. The

*https://www.confluent.io/blog/kafka-fastest-messaging-system/, accessed November-2021

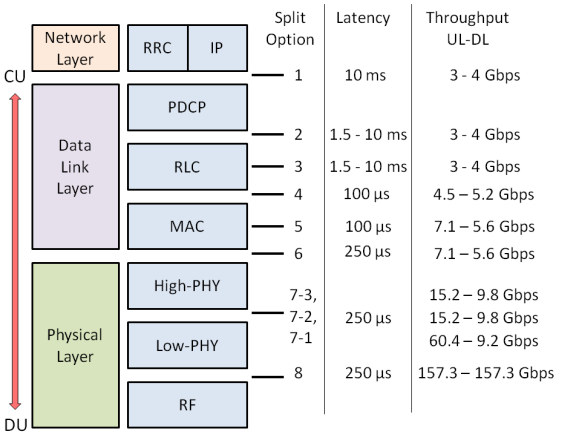


FIGURE 6. 5G RAN protocol stack split options with required latency and throughput values in 3GPP.

split option was proposed by 3GPP as a solution to the high capacity requirements of fronthaul networks (between CU and DU) in a fully centralized Cloud Radio Access Network (C-RAN) [40]. These different split options allow for different reductions in bandwidth and latency requirements. For example, in this split architecture, split option #8 is a fully centralized C-RAN, where all the baseband functions are centralized in CU pools, while split #1 corresponds to a traditional architecture where all baseband functions are allocated at DU location.

Figure 7 shows the latency versus throughput values for the 5G functional split option requirements, Field-Programmable Gate Array (FPGA) Inline Acceleration, Apache Kafka (on COTS)[†], and Apache Pulsar (on COTS)[‡]. For the performance of Apache Kafka and Apache Pulsar on COTS, we use the existing state-of-the-art results listed in the detailed tests in [39]. These reference tests are designed to compute the lowest latency any configured system can achieve when processing workloads consisting of publish and tailing reads. For a given number of subscriptions and number of partitions, the impact on publish and end-to-end latency is observed. In the test strategy of [39], each message was replicated three times to ensure fault tolerance. A message size of 1KB is used and the producer sent messages at a fixed rate of 200 MB/s and the tailing-read consumers processed the messages while the producer kept sending them.

In Figure 7, gray colored areas denote the regions where the requirements are not satisfied while the light green colored areas denote the regions where the requirements are satisfied by the corresponding framework. At this time, existing open-source service bus solutions such as Apache Kafka and Apache Pulsar have been benchmarked and shown to provide an average end-to-end system latency of 2.11 msec. and 2.86 msec. respectively and a system throughput of 300 Mbps. This comes from extensive test results with 10 subscriptions, 2 consumers per subscription,

[†]https://kafka.apache.org/, accessed November-2021
[‡]https://pulsar.apache.org/, accessed November-2021

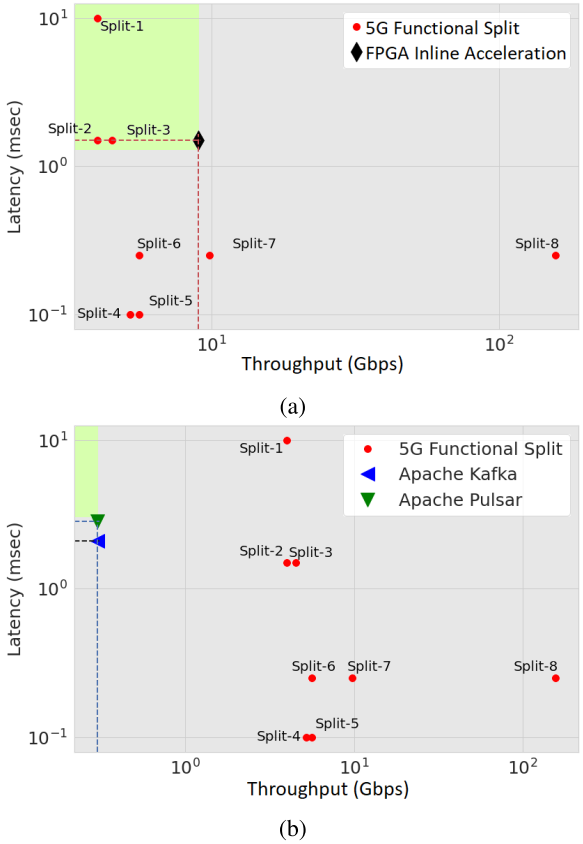


FIGURE 7. Values for the latency versus throughput requirements for the 5G RAN functional split options and the provided values for (a) FPGA inline acceleration [41] and (b) Apache kafka and apache pulsar [39].

and 100 partitions on Pulsar and Kafka (without data synchronization and ack-1) [39]. On the other hand, with hardware acceleration techniques on FPGA, FPGA inline acceleration can deliver throughput of 9 Gbps and latency of about 1.5 msec (for 500 FFT frames) [41], [42][§] end-to-end latency by 22x while maintaining a data rate of 9 Gbps compared to a system without hardware accelerator. In these references, the latency is measured from the time the UDP packets are sent out by the traffic generator to the time the traffic is received back on the host. The test setup demonstrates low latency data ingestion using the FPGA. Traffic from the hosts is generated over a 10GbE interface and the UDP protocol. The traffic arrives over an optical link and is received by the FPGA. The FPGA accelerator is implemented using the Intel FPGA Software Development Kit (SDK) for OpenCL framework.

The results in Figure 7a show that FPGA inline acceleration providing accelerated hardware can potentially meet the requirements of 5G RAN functional split options 1, 2, and 3 (as indicated by light green colored areas). On the other hand, 5G RAN functional split options 4 to 8 are not met by the currently studied solutions (as indicated by gray colored areas) in Figure 7a. Therefore, some of the functions of RAN such as FWA (which is a highly centralized

[§]The FPGA inline accelerator can reduce

application that does not require cell-site coordination and has relatively relaxed latency and bandwidth requirements on the transport network) inside the vPP service of Figure 3 can run as microservices since it complies with split option 2. On the other hand, some of the functions of RAN such as BBP within the vBB or CA within the vRC service of Figure 3 cannot currently be operated as a microservice because they are designed either for networks with high capacity and reliability requirements supporting network densification in urban areas and enabling sharing among multiple MNOs or coordination between cell sites depending on the data-rate requirements on the fronthaul network. This can be achieved with low-level splits such as level 7 splits. At the same time, existing open source software versions such as Apache Kafka and Apache Pulsar may not currently meet the 5G functional split requirements (as indicated by the gray colored areas) in Figure 7b.

B. ADVANTAGES

SBA offers several advantages such as extensibility (by simply adding NFs), updatability (by loosely coupling between microservices) and reusability. In SOA, each service consists of the code and integration that have the goal of performing a specific function. On the other hand, SBA's microservice architecture, running on a container-based environment, is a pure cloud-native approach with a scalable and portable solution for building RAN services. This advantage of SBA allows RAN greater vendor independence. For example, each microservices within the SBA platform can be provided by different and multiple parties (vendors, SPs, MNOs, Cloud Service Providers (CSPs), etc.). Moreover, those companies can develop their own services tailored to their consumers. For example, the current monolithic architectures from different vendors have different features for mobile networks which makes it difficult to deploy them in a different architecture. However, the SBA-based approach provides more freedom to deploy many more applications and services in a RAN environment.

For the use cases of next-generation RAN services and from the perspective of different mobile user profiles of MNOs, SBA also brings great flexibility in terms of managing the RAN services. In traditional monolithic architectures, capacity expansion is usually achieved by deploying additional hardware and purchasing the appropriate software licenses. As an example consider a particular region where UEs have many different QoS characteristics. In these regions, network services depend on heavy radio resource usage. If this high usage adversely affects the performance of the RAN in this region, the MNO will need to invest in additional hardware and software in a monolithic architecture when the currently deployed hardware is fully utilized.

On the other hand, if SBA is used, scaling operations up/down or out/in can be easily performed for the heavily loaded services. Although these operations can also be performed in a cloudified but proprietary system, and scaling operations in SBA on a cloudified system also requires additional COTS hardware investments, the operational expenses are significantly reduced due to ease of service deployment and replications. Because of the reduced interdependence, RAN services will also be much more fault tolerant. In addition, a vRAN with SBA can allow non-telecom specialized entities (e.g., CSPs) to offer RAN services as well as CN services over the virtualized and containerized environment.

C. CHALLENGES

i) Service Interoperability: In cases where SUs are developed by different initiatives or even by different mobile network vendors, interoperability issues may arise. Standardized SU design, development, and deployment strategies must be followed by all service providers to resolve interoperability issues of services from different service providers.

ii) Resource allocation to the services: Another important point is the placement and design for hierarchical resource usage of RAN services. For example, SUs responsible for cell and shared channel processing must be the part of baseband processing and visible to many SIs since they will have many interactions. Moreover, common RAN services can also be deployed as part of different NFs or split into multiple CNFs to meet the use case requirements. With the proposed SBA, RAN functions can be placed anywhere in the infrastructure. However, most services may have connections to at least some of them (due to location constraints, spectrum usage, etc.). For this reason, the service discovery process must be implemented in such a way that it can be used by all services [43]. As a possible solution, for example, Single Frequency Network (SFN) based approaches can reduce the load on these services by simplifying the number of frequencies used [44].

iii) Partitioning of services: Partitioning with domains affects the visibility of services, which helps reduce the number of possible paths between SIs. Partitioning in a mobile network environment with heterogeneous network (HetNet) deployment is challenging. Another issue that may arise in the context of partitioning is the deployment of concurrent services (e.g., 5G services such as enhanced Mobile Broadband (eMBB), machine Type Communication (MTC), Ultra-Reliable Low-Latency Communication (URLLC)) in mobile networks. When domains are created based on criteria such as nodes, some services are overused while others may not be used at all [45].

iv) Network connectivity: In the service producer discovery process of SBA, there is no mechanism to check whether the exposed SIs are routable in the network domain. Therefore, RAN services may be corrupted if all SIs are published in a common domain but there is no valid route between the subnets, or if SIs are published in different domains. Another problem can occur when it is decided to separate the traffic of UP and Connection Point (CP) by using different network paths (a common solution in RAN), during the design phase. During this period, network connectivity needs to be rechecked when the service discovery process is

running. As an example, consider the case where there are four SIs in RAN. Three SIs are connected to both CP and UP networks while only one SI is connected to CP. If the SB connection of this one SI that has only the CP connection is configured to use the CP for a GPRS Tunneling Protocol (GTP)-UP connection in a packet processing service and with the UP for a forwarding service, the UP connection cannot be established.

A UE may request support for multiple network services or slices of RAN simultaneously. This can occur when opening multiple Packet Data Unit (PDU) sessions if each network service is assigned at least one PDU session. Each PDU session supporting each of these RAN services with different connectivity requirements may also require different IP addresses depending on the scenarios considered, which may add complexity. Consequently, the service discovery process must have a logic that controls the connectivity at the time of service discovery. This mechanism must verify that all SIs and RAN service consumers have routable IP addresses to ensure full visibility between SIs and RAN service consumers.

v) Service interactions: In a mobile network, UE controllers such as Mobile Device Management (MDM) can be used to manage the allocation of other network services. However, the problem arises in managing the interaction of this UEs controller with other RAN services such as baseband processing or packet processing services. This can be partially solved by settings labels and partitioning so that only selected SIs in the datacenter with desired properties (e.g. lower latency and higher bandwidth) can be supported [46].

vi) Security aspects: In a SBA, application complexity is distributed across deployable computational units. Therefore, SUs and SB must be implemented securely. Moreover, they should allow independent security functions for the user and control planes and should not allow unauthorized access, since the complexity mentioned above can cause serious security vulnerabilities. This can be achieved by using encryption techniques such as TLS, Datagram Transport Layer Security (DTLS), depending on the implementation and performance requirements of CPU, memory, I/O, etc. In addition, secure communication between microservices via TLS should be ensured for the availability of RAN services/microservices. In this case, key storage of TLS can be ensured by using an external hardware security module.

D. ENABLING TECHNOLOGIES

In Section V-C we discussed the challenges of SBA based vRAN. In this section, we will discuss more about the feasibility and the possible enabling technologies that can provide a clear roadmap for the development of the proposed SBA-based vRAN.

Although SBA components and technologies specific to all RAN use cases are yet available, the available technologies in the ecosystem can still be used to solve some of the limited use scenarios described above. For microservices orchestration, there are several solutions such as `Kubernetes`, `Docker Swarm`*, `Apache Mesos`† as well as cloud computing options such as Amazon Elastic Container Service (ECS), Microsoft's Azure Container Service. For message/streaming buses that can be used for microservice architectures, in addition to Apache Kafka and Apache Pulsar, related technologies include `Spark Streaming`‡, `Flink Data Streams`§ for streaming, `RabbitMQ`¶, `RocketMQ`‖ for messaging services are other available open-source options. For vendor- and cloud-based solutions, Amazon's `Kinesis`** streams (similar to Kafka), Facebook's `Puma`, `Swift`, and `Stylus` stream processing systems, Google's Cloud `Pub/Sub` (data ingestion and messaging for event-driven systems as well as streaming analytics) and Azure's `Event Hubs`, `IoT Hub`, `Stream Analytics` are the corresponding data ingestion services. As a messaging system, `Oracle Enterprise Messaging Service` and `IBM Websphere MQ` are other examples of event buses for processing asynchronous data flows. However, most of the existing cloud or on-promise technology solutions listed above are still at the early stages of meeting the stringent requirements for the RAN services they support. For this reason, they are not currently available in the market to address all the challenges of the existing wide range of RAN use cases.

SIs must run in execution environments (containing the base image and components of SU) that have small-footprint Linux OS distributions with container runtime features. Otherwise, the size of the container image can become a bottleneck for startup time (unless the image is copied to internal storage as in the Google Cloud Run platform). SomeOS distros currently available in the market are VMware PhotonOS, CoreOS, RancherOS, Fedora CoreOS, Alpine Linux, RedHat Project Atomic, etc. These distros are smaller in terms of occupied capacity, but they contain a footprint of hundreds of applications. For example, the size of Alpine Linux image can occupy 5.24 MB, while the size of Fedora OS size is 214.95 MB. It is also necessary to select a suitable OS that can support the technologies such as Data Plane Development Kit (DPDK) in real-time containers.

E. DISCUSSIONS & FURTHER DIRECTIONS

Realizing SB with RAN services with strict low latency and high throughput requirements is a challenging problem. If a SB is to be implemented in the current implementation landscape on regular $x86$ hardware and available open-source software, RAN services (which are sensitive to critical latency values and require high bandwidth) may suffer. In this case, the ideal solution would be to implement SB with

*https://docs.docker.com/engine/swarm, accessed November-2021
†http://mesos.apache.org/, accessed November-2021
‡https://spark.apache.org/, accessed November-2021
§https://flink.apache.org/, accessed November-2021
¶https://www.rabbitmq.com/, accessed November-2021
‖https://rocketmq.apache.org/, accessed November-2021
**https://aws.amazon.com/kinesis/, accessed November-2021

TABLE 2. Comparisons of different RAN modes to provide 5G services: Proprietary C-RAN, open RAN & SBA-based virtualized RAN.

RAN Strategy	Characteristics	Limitations	Advantages / Benefits
Proprietary C-RAN	— Consists of radio remote head (RRH), and based band processing unit (BBU) as separate entities. — Proprietary building blocks.	— Hard to add new RAN services — Requires fine-grained configurations before deployment. — High dependency to Vendors.	— Easy installation & configuration —Use of out-of-box services. — Simple to scale horizontally.
Open RAN	— Open interfaces between BBUs (open- F1/E1 interfaces), — Works with any BBU software. —Radio intelligence controller for NFs in BBU.	—Each service requires its own continuous delivery cycle. — Integration seems less generic. — Security vulnerabilities is not appropriately for designed open source software deployments. —The entities are not scalable in number of services by design.	— Creates openness, competition & novelty. — Easy third party application development. — Ongoing standardization efforts in a community. — Reduced inter-dependencies. — Improved modularity.
SBA-based Virtualized RAN	— Combines microservices & SOA. — Different services via service mesh topology. — Based on the design of the services. — Implementations are specific to environment. — Service are represented to utilize business descriptions with consistent context.	— Potential chance of failure during inter-service communication. — Some complexity in managing a large number of services (meta-data management). — Challenging testing environment. — Created massive amount of data and complicated computation. — High CAPEX investment to virtualize fully.	— Easy transition to multi-cluster environment. — Highly scalable. — Can provide low latency and high throughput in case scaled appropriately. — Vendor, product and technology independence. — Ability to adapt quickly to different external environments. —Self-adaptive characteristics via orchestration.

cloud-based FPGA solutions [47]. Note that this approach is different from the regular implementation of 5G products for the radio, baseband and massive MIMO antennas on FPGA, as it may have the downside of cost and power consumption*. FPGA-based acceleration in the cloud can be used in the queuing and computational load reduction operations of the SB [48]. At the same time, the fact that only options 1, 2, and 3 of the 5G RAN functional split are supported with the existing technology landscape may be a limiting factor for the deployment of future RAN services. However, the general trend in the industry is towards softwarization of all components in an end-to-end mobile architecture. In addition, general-purpose servers (which have enabled the realization of Software Defined Radios (SDRs)) are increasingly becoming more powerful [49].

Note that in Section V-A only the throughput and latency values of some SB frameworks are analyzed. Although these are important metrics, other broader results such as reliability can also be examined for benchmarking purposes. Since SBA was first proposed for the 5G core network, several metrics such as CPU/memory cost, requests success rate, and some HTTP-specific performance metrics have already been studied in [25]. These metrics can also be studied in the proposed SBA-based vRAN case in the future. In addition, services for ML/AI assistance can be further investigated to position them in the SBA-based vRAN in future studies. This may be particularly useful in the area of parallel and distributed computing to reduce computational load and increase scalability. Exploring high-performance SB technologies can also make the qualitative step from running SDR (which are now quite hardware-bound) to a more generic microservice that is easier to deploy. Another possible future work could focus specifically on how well different 5G RAN slicing services can be integrated into a SBA-based vRAN environment. One possible solution is to use container isolation technologies (e.g., gVisor [50], Nabla [51]) to provide separate RAN services. GVisor is a user-space kernel developed by Google and written in the Go programming language. Nabla containers are implemented by IBM as processes on Linux.

Finally, Table 2 provides summary comparison of the proprietary C-RAN, the Open RAN and the proposed SBA-based vRAN strategies in terms of their characteristics, limitations and advantages in providing RAN services.

VI. CONCLUSION

5G core network has already adapted to the evolving SBA and its principles. However, the application of SBA principles in RANs is not yet mature and has not been adopted for 5G RAN in 3GPP at this time. Description functions for vRAN are gradually maturing, so the RAN domain is moving rather slowly towards virtualization. In this paper, we have studied the application of SBA-based design to the vRAN domain for cellular networks. We have approached the application of SBA principles in RAN domain in terms of software, rather than high-level concept studies. We have listed and highlighted the factors driving SBA in general, described the features for implementing SBA-based vRAN, and outlined its design principles and implementation details.

We have also investigated the potential application of an SBA-based vRAN solution to 5G by comparing existing 5G RAN functional split requirements with available SB open-source software and corresponding FGPA-accelerated hardware implementations in terms of latency and throughput. The aggregated results indicate that the studied open-source

*https://www.fiercewireless.com/5g/nokia-made-a-bad-call-for-5g-chips-scrambles-to-rectify-situation, accessed November-2021

software versions (Apache Kafka and Apache Pulsar) may not currently meet the stringent 5G functional split requirements, while the accelerated hardware versions (with FPGA inline acceleration) can potentially meet the requirements of 5G RAN functional split options 1,2 and 3. At the end of the paper, we have also presented some discussions and future directions for the application of SBA-based vRAN in the next-generation mobile infrastructure wherein SBA-based vRAN could be a possible technical direction in 6G.

REFERENCES

[1] G. Brown. (2017). *Service-Based Architecture for 5G Core Networks*. Accessed: Jan. 2021. [Online]. Available: https://bit.ly/3o6i7nU

[2] *5G: NG-Ran: Architecture Description*, document TS 38.401 version 16.2.0 Release 16, 3GP, 2020.

[3] Ericsson. (2020). *Cloud Ran-Tech Unveiled*. Accessed: Sep. 2021. [Online]. Available: https://bit.ly/3rk64b8

[4] *O-RAN Architecture Description*, ORAN, Algeria, North Africa, 2021.

[5] *Reimagining the End-to-End Mobile Network in the 5G Era*, Cisco, San Jose, CA, USA, 2019.

[6] Ericsson. (2020). *Cloud RAN (Radio Access Network) by Ericsson*. Accessed: Nov. 2020. [Online]. Available: https://bit.ly/34RM2c2

[7] G. Brown. (2017). *Cloud RAN & The Next-Generation Mobile Network Architecture*. Accessed: Oct. 2020. [Online]. Available: https://bit.ly/3en5rVJ

[8] J. Wu, Z. Zhang, Y. Hong, and Y. Wen, "Cloud radio access network (C-RAN): A primer," *IEEE Netw.*, vol. 29, no. 1, pp. 35–41, Jan. 2015.

[9] L. M. P. Larsen, A. Checko, and H. L. Christiansen, "A survey of the functional splits proposed for 5G mobile crosshaul networks," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 146–172, 1st Quart., 2019.

[10] M. F. Hossain, A. U. Mahin, T. Debnath, F. B. Mosharrof, and K. Z. Islam, "Recent research in cloud radio access network (C-RAN) for 5G cellular systems—A survey," *J. Netw. Comput. Appl.*, vol. 139, pp. 31–48, Aug. 2019.

[11] K. Vieira Cardoso, C. Bonato Both, L. Rene Prade, C. J. A. Macedo, and V. Hugo L. Lopes, "A softwarized perspective of the 5G networks," 2020, *arXiv:2006.10409*.

[12] 5G PPP Architecture Working Group. (2019). *View on 5G Architecture*. Accessed: Nov. 2020. [Online]. Available: https://bit.ly/3cFXpHK

[13] NRG-5 Project. (2019). *Enabling Smart Energy as a Service Via 5G Mobile Network Advances*. Accessed: Nov. 2020. [Online]. Available: https://bit.ly/3iGKN6Q

[14] S. Doumiati, H. Artail, and D. M. Gutierrez-Estevez, "A framework for lte-a proximity-based device-to-device service registration and discovery," *Proc. Comput. Sci.*, vol. 34, pp. 87–94, Oct. 2014.

[15] *5G: NR: Radio Resource Control (RRC): Protocol Specification*, document TS 138 331 V15.2.1 (2018-06), ETSI, 2018.

[16] *NGMN Overview on 5G RAN Functional Decomposition*, NGMN Alliance, Frankfurt am Main, Germany, 2018.

[17] Q. Duan, Y. Yan, and A. V. Vasilakos, "A survey on service-oriented network virtualization toward convergence of networking and cloud computing," *IEEE Trans. Netw. Service Manage.*, vol. 9, no. 4, pp. 373–392, Dec. 2012.

[18] F. Oliveira, T. Eilam, P. Nagpurkar, and C. Isci, "Delivering software with agility and quality in a cloud environment," *IBM J. Res. Develop.*, vol. 60, nos. 2–3, pp. 10:1–10:11, 2016.

[19] T. G. J. Schepers, M. E. Iacob, and P. A. T. Van Eck, "A lifecycle approach to SOA governance," in *Proc. ACM Symp. Appl. Comput.*, 2008, pp. 1055–1061.

[20] R. Schmidt and N. Nikaein, "RAN engine: Service-oriented RAN through containerized micro-services," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 1, pp. 469–481, Mar. 2021.

[21] A. Balalaie, A. Heydarnoori, and P. Jamshidi, "Microservices architecture enables devops: Migration to a cloud-native architecture," *IEEE Softw.*, vol. 33, no. 3, pp. 42–52, May/Jun. 2016.

[22] S. Dutta, T. Taleb, and A. Ksentini, "QoE-aware elasticity support in cloud-native 5G systems," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Oct. 2016, pp. 1–6.

[23] N. Serrano, J. Hernantes, and G. Gallardo, "Service-oriented architecture and legacy systems," *Softw. Technol.*, vol. 31, no. 5, pp. 15–19, Sep./Oct. 2014.

[24] X. Li, C. Guimaraes, G. Landi, J. Brenes, J. Mangues-Bafalluy, J. Baranda, D. Corujo, V. Cunha, J. Fonseca, J. Alegria, A. Z. Orive, J. Ordonez-Lucena, P. Iovanna, C. J. Bernardos, A. Mourad, and X. Costa-Perez, "Multi-domain solutions for the deployment of private 5G networks," *IEEE Access*, vol. 9, pp. 106865–106884, 2021.

[25] J. B. Moreira, H. Mamede, V. Pereira, and B. Sousa, "Next generation of microservices for the 5G service-based architecture," *Int. J. Netw. Manage.*, vol. 30, no. 6, p. e2132, Nov. 2020.

[26] K. Samdanis and T. Taleb, "The road beyond 5G: A vision and insight of the key technologies," *IEEE Netw.*, vol. 34, no. 2, pp. 135–141, Mar./Apr. 2020.

[27] J. Luna, N. Suri, M. Iorga, and A. Karmel, "Leveraging the potential of cloud security service-level agreements through standards," *IEEE Cloud Comput.*, vol. 2, no. 3, pp. 32–40, May 2015.

[28] S. Polona and M. Cigale, "Switch workbench: A novel approach for the development and deployment of time-critical microservice-based cloud-native applications," *Future Gen. Comp. Syst.*, vol. 99, pp. 197–212, Oct. 2019.

[29] *Network Functions Virtualisation (NFV) Release 3: Management and Orchestration: VE-VNFM Reference Point—Interface and Information Model Specification*, document ETSI GS NFV-IFA 008 V3.1.1, 2018.

[30] J. F. Santos, M. Kist, J. Rochol, and L. A. DaSilva, "Virtual radios, real services: Enabling RANaaS through radio virtualisation," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 4, pp. 2610–2619, Dec. 2020.

[31] T. V. K. Buyakar, H. Agarwal, B. R. Tamma, and A. A. Franklin, "Prototyping and load balancing the service based architecture of 5G core using NFV," in *Proc. IEEE Conf. Netw. Softwarization (NetSoft)*, Oct. 2019, pp. 228–232.

[32] T.-X. Do and Y. Kim, "Latency-aware placement for state management functions in service-based 5G mobile core network," in *Proc. IEEE 7th Int. Conf. Commun. Electron. (ICCE)*, Jul. 2018, pp. 102–106.

[33] S. Ramanathan and K. Kondepu, "Performance evaluation of two service recovery strategies in cloud-native radio access networks," in *Proc. 21st Int. Conf. Transparent Opt. Netw. (ICTON)*, 2019, pp. 1–5.

[34] F. Wilhelmi and L. Giupponi, "On the performance of blockchain-enabled RAN-as-a-service in beyond 5G networks," 2021, *arXiv:2105.14221*.

[35] L. Giupponi and F. Wilhelmi, "Blockchain-enabled network sharing for O-RAN in 5G and beyond," 2021, *arXiv:2107.02005*.

[36] W. Shi, J. Li, W. Xu, H. Zhou, N. Zhang, and S. Zhang, "Multiple drone-cell deployment analyses and optimization in drone assisted radio access networks," *IEEE Access*, vol. 6, pp. 12518–12529, 2018.

[37] E. Pateromichelakis, J. Gebert, T. Mach, J. Belschner, W. Guo, and N. P. Kuruvatti, "Service-tailored user-plane design framework and architecture considerations in 5G radio access networks," *IEEE Access*, vol. 5, pp. 17089–17105, 2017.

[38] ITU-T G-series Recommendations. (2018). *5G Wireless Fronthaul Requirements in a Passive Optical Network Context*. Accessed: Apr. 2021. [Online]. Available: https://bit.ly/3wpcRzS

[39] P. Li and S. Guo. (2020). *Benchmarking Pulsar and Kafka—The Full Benchmark Report*. Accessed: Apr. 2021. [Online]. Available: https://bit.ly/35jMzTD

[40] *Study on New Radio Access Technology: Radio Access Architecture and Interfaces*, document TR 38.801 V14.0.0 (2017-03), 3GPP, 2020.

[41] M. relax Kainthand, D. Pritsker, and H. S. Neoh. (2020). *FPGA Inline Acceleration for Streaming Analytics*. Accessed: Apr. 2021. [Online]. Available: https://intel.ly/3wpcMMA

[42] Bittware. (2018). *Accelerate Kafka Producers With FPGAs (White Paper)*. Accessed: Sep. 2021. [Online]. Available: https://www.bittware.com/resources/kafka/

[43] G. Ortiz, J. Caravaca, and A. García-de-Prada, "Real-time context-aware microservice architecture for predictive analytics and smart decision-making," *IEEE Access*, vol. 7, pp. 183177–183194, 2019.

[44] Y. Turk, E. Zeydan, and C. A. Akbulut, "On performance analysis of single frequency network with c-ran," *IEEE Access*, vol. 7, pp. 1502–1519, 2019.

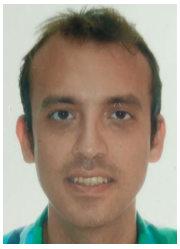
[45] Q. He, J. Yan, H. Jin, and Y. Yang, "Quality-aware service selection for service-based systems based on iterative multi-attribute combinatorial auction," *IEEE Trans. Softw. Eng.*, vol. 40, no. 2, pp. 192–215, Feb. 2014.

[46] Y. Wu, F. He, D. Zhang, and X. Li, "Service-oriented feature-based data exchange for cloud-based design and manufacturing," *IEEE Trans. Services Comput.*, vol. 11, no. 2, pp. 341–353, Mar. /Apr. 2018.

[47] A. A. Al-Aghbari and M. E. S. Elrabaa, "Cloud-based FPGA custom computing machines for streaming applications," *IEEE Access*, vol. 7, pp. 38009–38019, 2019.
[48] N. Tarafdar, N. Eskandari, V. Sharma, C. Lo, and P. Chow, "Galapagos: A full stack approach to FPGA integration in the cloud," *IEEE Micro*, vol. 38, no. 6, pp. 18–24, Nov./Dec. 2018.
[49] R. Akeela and B. Dezfouli, "Software-defined radios: Architecture, state-of-the-art, and challenges," *Comput. Commun.*, vol. 128, pp. 106–125, Sep. 2018.
[50] Gvisor. (2020). *Application Kernel for Containers*. Accessed: Jul. 2021. [Online]. Available: https://github.com/google/gvisor
[51] Nabla Containers. (2020). *A New Approach to Container Isolation*. Accessed: Jul. 2021. [Online]. Available: https://nabla-containers.github.io/

ENGIN ZEYDAN (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees from the Department of Electrical and Electronics Engineering, Middle East Technical University, Ankara, Turkey, in 2004 and 2006, respectively, and the Ph.D. degree from the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, USA, in February 2011. He has worked as a Research and Development Engineer at Avea, a mobile operator in Turkey, from 2011 to 2016. He was working as a Senior Research and Development Engineer with Turk Telekom Labs, from 2016 to 2018. He was also a part-time Instructor at the Electrical and Electronics Engineering Department, Ozyegin University, from 2015 to 2018. He is currently with the Communication Networks Division, Centre Tecnològic de Telecomunicacions de Catalunya (CTTC), working as a Senior Researcher. His research interests include telecommunications and data engineering. He received the Best Paper Award from the Network of Future Conference, in 2017.

JOSEP MANGUES-BAFALLUY received the degree and Ph.D. degrees in telecommunications engineering from UPC, in 1996 and 2003, respectively. He is currently a Senior Researcher and the Head of the Communication Networks Division, Centre Tecnològic Telecomunicacions Catalunya (CTTC), Barcelona. Previously, he was a Researcher and an Assistant Professor with UPC. He has participated in various roles (including leadership) in several public funded and industrial research projects, such as 5GPPP 5Growth, 5G-Transformer, or Spanish 5G-REFINE. His research interests include NFV applied to mobile networks and autonomous network management. He was the Vice-Chair of the IEEE WCNC, Barcelona, in 2018.



JORGE BARANDA (Senior Member, IEEE) received the M.S. degree in electrical engineering from the Technical University of Catalonia, in 2008. He is currently a Senior Researcher with the Department of Mobile Networks, Centre Tecnològic Telecomunicacions Catalunya (CTTC), Barcelona. At CTTC, he has participated in several European, national, and industrial projects related to management and orchestration of SDN/NFV mobile networks, efficient routing strategies for mobile network backhauling, and novel wireless communication systems. He has coauthored over 30 different peer-reviewed journals and conference papers. His current research interests include management and orchestration of SDN/NFV mobile networks, wireless communications, wireless backhaul, network routing protocols, and network optimization.

MANUEL REQUENA received the M.Sc. degree in computer science from the Technical University of Valencia (UPV), in 1998. He released his master's thesis at the Ecole Nationale Supérieure de l'Electronique et de ses Applications (ENSEA), France. He worked as a Software Engineer developing telecommunication software solutions with Atos Origin Integration, France, from 1998 to 2003. He is currently a Senior Research Engineer and the Coordinator of the EXTREME Testbed with the Centre Tecnològic Telecomunicacions Catalunya (CTTC), Barcelona, where he is responsible for the laboratory with the Mobile Networks Department. Moreover, he is one of the designers, developers, and maintainers of the LTE module of the NS-3 simulator. His current research interests include design and implementation of networking protocols for IP networks, self-organized wireless mesh networks, cellular networks, such as LTE, HSPA, and UMTS, ll-IP heterogeneous environments, network function virtualization (NFV), software-defined networking (SDN), and networks simulation.

YEKTA TURK received the B.Sc. degree in electrical and electronics engineering from Anadolu University, Turkey, in 2005, the M.Sc. degree in telecommunications and computer networks from George Washington University, DC, USA, in 2007, and the Ph.D. degree from the Department of Computer Engineering, Maltepe University, Istanbul, Turkey, in 2018. He is currently a Lead Systems Engineer at Aselsan Corporation, Istanbul. His research interests include mobile radio telecommunications and computer networks.