Open RAN xApps Design and Evaluation: Lessons Learnt and Identified Challenges

Marcin Hoffmann, *Graduate Student Member, IEEE*, Salim Janji, *Graduate Student Member, IEEE*, Adam Samorzewski, *Graduate Student Member, IEEE*, Łukasz Kułacz, *Member, IEEE*, Cezary Adamczyk, Marcin Dryjański, *Senior Member, IEEE*, Pawel Kryszkiewicz, *Senior Member, IEEE*, Adrian Kliks, *Senior Member, IEEE*, and Hanna Bogucka, *Senior Member, IEEE*

***Abstract*— The concept of open radio access networks (RAN) creates numerous opportunities for developing new technology and economy branches. At the same time, a flexible and modular approach in the disaggregated RAN entails the need for careful design of the overall RAN architecture and the implementation and deployment process of new applications. It is assumed that dedicated and specialized software companies may deliver the latter. A joint effort must be guaranteed among different sectors (industry, academia, and standardization bodies) to make the whole process efficient, safe, and reliable. Here, one of the critical driving forces origins from the open-source community that often stimulates the development of a specific technology. In this paper, we address the challenges that have to be faced by third-party application developers in the context of Open RAN. Based on many implemented applications (called xApps or rApps), we compare various available solutions. We pose the most critical issues that must be tackled in the near future to stimulate the progress in open RAN development further. In particular, we compare available open platforms for xApp development and testing. We present the details of implementing four selected applications describing the problems encountered. The paper is split into two logical parts - first, we identify the key ambiguities related to the development of new xApps, which address more complicated use cases like beam management. In the second part, we present the challenges associated with detailed software implementation in existing open platforms. In the first case, we show that dedicated beam mobility management xApp can reduce beam switches and keep beam failures low. However, it requires access to detailed localization information. Similarly, the signaling storm detection xApp provides expected performance under the assumption that there is access to detailed information on, e.g., time advance resolution parameter. We conclude here that several aspects still need to be well-defined to allow smooth software implementation; these include the rules for data reporting in time, parameters available in service models, and localization features. Concerning the second logical part, related to low-level implementation, we compare the numerical results of the traffic steering and quality-of-service-based resource allocation xApps and draw conclusions related to implementation and testing. In particular, we point out problems associated with the simulator, the software, and conflicts inside. Finally, we identify the key challenges which should be treated as incentives for joint academia-industry cooperation in the field of Open RAN. Thus, the paper presents the lesson learned during the first years of xApp development.**

***Index Terms*— Open RAN, 5G, 6G, xApp, ML.**

I. Introduction

DISAGGREGATION, openness, flexibility, and modality are the new paradigms attributed to the next generation of wireless communication networks. Contrary to the traditional and prevalent approach to the radio access network (RAN) design, where most of the RAN elements are provided by one vendor and are hidden in the *black-box*, the concept of the Open RAN assumes that potentially multiple players provide dedicated RAN modules. Such a modular approach allows operators to modify and improve only selected network functionalities instead of completely replacing the black-boxed software. It is the network operator who decides what functions in the network should be activated or deactivated, which should be improved, kept unchanged, or uninstalled. These modifications can be done by adequately manipulating the installed software modules. This, in turn, opens the possibility for incremental system modifications following the concept of continuous integration and continuous development (CI/CD). The standardization activities related to Open RAN, which are led by the O-RAN ALLIANCE [1] emphasize the trend towards open and modular RAN. The set of standards released by this organization specifies the overall Open RAN architecture, requirements, and functionalities. In particular, new and open interfaces have been proposed to incentivize xApp/rApp providers to implement and deliver new algorithms dealing with specific aspects of wireless communications. However, along with the numerous and evident benefits of opening and disaggregating the RAN, significant challenges are related to the practical implementation of such a vibrant concept. First, the way for implementation and deployment of new xApps/rApps has to be unified and automated so

Manuscript received 22 December 2022; revised 2 August 2023; accepted 7 September 2023. Date of publication 28 November 2023; date of current version 17 January 2024. This work was in part by the National Centre for Research and Development in Poland within the 5GStar Project on "Advanced Methods and Techniques for Identification and Counteracting Cyberattacks on 5G Access Network and Applications under Grant CYBERSECIDENT/487845/IV/NCBR/2021. *(Corresponding author: Marcin Hoffmann.)*

Marcin Hoffmann, Salim Janji, Adam Samorzewski, Łukasz Kułacz, Pawel Kryszkiewicz, Adrian Kliks, and Hanna Bogucka are with the Rimedo Labs, 61-131 Poznań, Poland, and also with the Institute of Radiocommunications, Poznan University of Technology, 60-965 Poznań, Poland (e-mail: marcin.hoffmann@put.poznan.pl).

Cezary Adamczyk is with the Institute of Radiocommunications, Poznan University of Technology, 60-965 Poznań, Poland.

Marcin Dryjański is with the Rimedo Labs, 61-131 Poznań, Poland.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JSAC.2023.3336190.

Digital Object Identifier 10.1109/JSAC.2023.3336190

that every interested software provider may deliver valuable contributions to the community. Next, opening the RAN part to numerous, often external providers causes various security issues which must be tackled carefully. Also, the coexistence of applications from different xApp/rApp providers may lead to potential consistency and confluence problems and prospective conflicts. These topics are now the subject of both academic and industrial debate. However, despite all the efforts put into the foundation of an open and disaggregated RAN environment, the technology is still in its early stage of development. Although precisely specified in O-RAN ALLIANCE standards, the architecture is still modified and being adjusted to address new challenges and reply to the recent findings. Moreover, practical implementations also need more trusted simulation environments, commonly agreed ways for providing new software modules, methods of testing, and performance benchmarking.

These problems are particularly important from the perspective of the above-mentioned xApp/rApp providers, who still have limited possibilities of delivering new applications. When new xApp/rApp is being implemented, it has to be first simulated reliably and comprehensively, it has to be tested against numerous threats and risks, and the whole process has to be automated. Nowadays, it is not the case. This paper addresses this niche by presenting the observations gained in the years of xApp/rApp design and implementation. By implementing some xApps/rApps of different kinds, types, and scopes of functionalities, we are able to discuss the current state of the development art from the perspective of the xApp/rApp provider. We present our lessons learned and gained experience to identify key challenges, standardization, and research directions. To avoid the promotion of any commercial solutions and to promote open science, we concentrate on the Open RAN applications prepared with the openly available software and mutually compare the achieved results.

The paper is structured as follows - it contains four logical sections. First, a concise review of what O-RAN is is provided; next, the existing implementation frameworks are discussed and compared; third, we present four original xApp implementation results, discussing their performance and drawing conclusions about the whole design process. Its novelty, both comparative and scientific, can be summarized as follows:

- we present in detail four xApps, illustrating the message exchange between the particular O-RAN blocks; the proposed new methods solve particular research challenges related to wireless networks while preserving full compliance with the O-RAN standard requirements;
- we provide a detailed comparison of currently available software platforms and discuss their pros and cons; the comparison shall be the basis for the selection of open platforms for performing new research in the O-RAN domain,
- we provide an analysis of the identified O-RAN architectural ambiguities based on the challenges that have been faced during the implementation of the xApps,
- analogously, we share our synthetic observations in the context of current limitations related to xApp implementation; both ambiguities and limitations constitute the true research challenges for future O-RAN solutions.

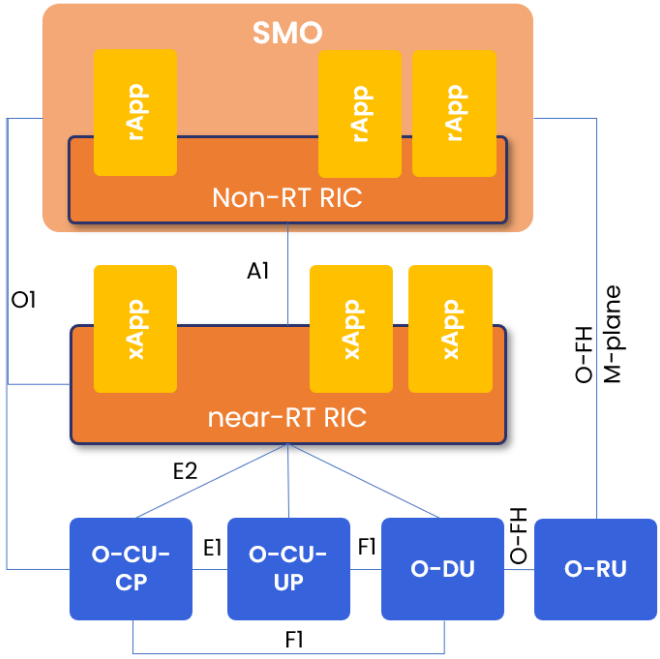


Fig. 1. O-RAN architecture, as defined by O-RAN ALLIANCE.

To precisely reflect the above topics, this paper is split into seven chapters, where the following section recaps the O-RAN architecture, RAN Intelligent Controller, and proposed use cases. Chapter 3 overviews the open-source platform for xApp development and testing. Next, Chapters 4 and 5 present the ways of four xApp implementations and draw conclusions related to architecture ambiguities and practical implementation, respectively. Chapter 6 discusses the key research challenges. The whole work is summarized in Chapter 7.

II. O-RAN Architecture, RIC and Use Cases

O-RAN ALLIANCE [1] is the main standardization body specifying the O-RAN reference architecture, interfaces, deployment scenarios, use cases, etc. In addition to this, it also leads official plugfests, provides an open-source implementation of the O-RAN stack, and interoperability and testing of the O-RAN solutions. This chapter provides an overview of the O-RAN architecture defined by O-RAN ALLIANCE, focusing on the **RAN Intelligent Controller (RIC)** along with xApps.

A. O-RAN Architecture

The O-RAN architecture is defined in [2] and builds upon 3GPP RAN standards towards openness and intelligence by adopting RAN splits, new interfaces, RICs, and Service Management and Orchestration (SMO) (see Figure 1). It adopts split 2 (also referred to as higher-layer split, HLS) between PDCP and RLC protocols within the New Radio (NR) air interface stack; and split 7.2x (also referred to as lower-layer split, LLS) within the PHY layer. The corresponding elements

of O-RAN are called **O-RAN Central Unit (O-CU)**, **O-RAN Distributed Unit (O-DU)**, and **O-RAN Radio Unit (O-RU)**.

O-CU is further split into the control plane (O-CU-CP), which covers Radio Resource Control (RRC) with Packet Data Convergence Protocol-Control Plane (PDCP-C) protocols, and the user plane (O-CU-UP) covering and Service Data Adaptation Protocol (SDAP) with PDCP-User Plane (PDCP-U). O-DU, in turn, encompasses Radio Link Control (RLC), Medium Access Control (MAC), and a high-physical layer, including the MAC scheduler. Finally, O-RU includes low-physical layer functionality like Orthogonal Frequency Division Multiple Access (OFDMA) processing, beamforming, and Radio Frequency (RF) front end.

An essential element introduced in O-RAN is the RAN Intelligent Controller (RIC), a separated-out entity from the processing units that allow access to RRM functions. RIC is split onto **Non-Real-Time RIC (Non-RT RIC)** and **Near-Real-Time RIC (Near-RT RIC)**. The former works in the timescale of above 1 s, is used for non-real-time radio resource management, higher layer procedure optimization, and policy optimization in RAN, and enables the artificial intelligence (AI) and machine learning (ML) workflow for RAN components. In addition, it provides policy-based guidance for the applications in Near-RT RIC and delivers Enrichment Information (EI) for the Near-RT RIC's applications. Near-RT RIC, on the contrary, is part of the RAN to enable control and optimization of algorithms for radio resource management, and it works with the control loop in a timescale of longer than 10 ms and shorter than 1 s utilizing the use-case specific applications called xApps.

O-RAN ALLIANCE also specifies new interfaces, including Open Fronthaul (OFH), which connects O-DU to O-RU, E2, and A1 serving as control loop connections, and O1, O2, OFH M-plane - i.e. management interfaces. O-CU-CP, O-CU-UP, and O-DU are called "E2 Nodes" in the O-RAN architecture. This is because they are connected via the E2 interface to the Near-RT RIC, by which their functionality can be controlled through external applications, i.e., the abovementioned xApps.

Among the mentioned interfaces, E2 and A1 are considered important in this paper, namely:

- **E2 interface**, which creates a closed loop within the RAN domain, is used to send the RIC control and policy toward E2 Nodes and to obtain the feedback from E2 Nodes to the Near-RT RIC.
- **A1 interface**, which provides policies, EI, and ML models towards Near-RT RIC and gets the policy feedback back to the Non-RT RIC.

B. O-RAN Near-RT RIC, xApps and Use Cases

Near-RT RIC is a software platform allowing the xApps to control the RAN. This is supported by the RAN and UE databases storing the network state, along with xApp management, security, and conflict mitigation functions. It enables near real-time control optimization of the E2 Nodes via actions sent over the E2 interface, including CONTROL, INSERT, POLICY, and REPORT services [2]. The detailed description of Near-Real-Time RIC is defined in [3].

E2 Nodes mentioned above expose parameters and functionalities towards RIC through the E2 interface, which xApps and rApps can use to tune the behavior of the radio network. Examples of xApps are mobility, interference or beamforming management, traffic steering, load balancing, slice control, admission control, signaling anomaly detection, etc.

In this paper, we focus on xApps which are applications run at the Near-RT RIC. An xApp provides information to the Near-RT RIC about the data types it consumes and about outputs it produces. Such an application is independent of the Near-RT RIC and may be developed by a third-party provider. It controls a specific RAN functionality exposed by the E2 Node using the E2 service models (E2SM). The current service models include KPM (Key Performance Measurements), RC (RAN Control), NI (Network Interface), and CCC (Cell Configuration and Control) [3].

To summarize, Near-RT RIC is one of the critical elements in the O-RAN architecture, which allows feeding intelligence into the operations of the RAN. It creates a platform on which the software providers could build per-use case RRM algorithms to allow the optimization of radio resources for specific scenarios, also known as use cases, which are covered in the following subsection. The use cases, based on which xApps and rApps are developed, are defined in [4] and are based on the requirements of O-RAN ALLIANCE members. Those requirements also come as input to the O-RAN ALLIANCE's standardization in the form of priorities from Telecom Infra Project, an organization that brings together several operators. TIP's OpenRAN program supports the development of disaggregated and interoperable RAN solutions based on service provider requirements [5]. Specifically, within the RRM part, TIP defines the RAN Intelligence and Automation subgroup (RIA), aiming to develop and deploy AI-based xApps for use cases like RRM, SON, Massive MIMO, etc.

The current set of O-RAN ALLIANCE use cases, as specified in [4] covers 23 items and includes, among others: V2X HO management, UAV radio resource allocation, QoE optimization, traffic steering, Massive MIMO BF optimization, RAN sharing, QoS-based resource optimization, RAN slice SLA assurance, Dynamic Spectrum Sharing, indoor positioning, signaling storm protection, congestion protection, energy saving, etc.

Based on the use case definition and description defined by O-RAN WG1, other working groups define parameters and procedures to create a normative way for interoperable interfaces to allow interworking between vendors. Examples include parameters and new service models at the E2 interface or policy definitions for those use cases at the A1 interface.

III. O-RAN Deployments—Open-Source Platforms Comparisons

To date, several open-source projects are used to implement Open RAN systems. Such platforms may provide the entire stack, including RAN software, RICs, SMO, or a subset of those components. This section presents several platforms and briefly describes the modules they provide. Since a complete end-to-end deployment or simulation of 5G systems requires implementing both the RAN and Core Network (CN) domains,

we also mention the 5G CN implementation that each project leverages in its platform while discarding any LTE Evolved Packet Core (EPC) implementations. We also highlight a few differences between them and conclude with an evaluation of each platform based on the documentation they provide and the hands-on experience gained while testing some of these projects.

A. OpenAirInterface (OAI) [6]

OpenAirInterface (OAI) Software ALLIANCE (OSA) was established in 2014 by the non-profit organization EURECOM. Among others, OAI provides the following projects.

1) OAI 5G CN and EPC CN: these projects provide 5G standalone (SA) CN and 5G Non-standalone (NSA) CN network functions (NFs) implementations, respectively.

2) OAI 5G RAN: this OAI project implements software for NSA and SA gNB, eNB, 5G NSA and SA UE, and LTE UE.

3) OAI's MOSAIC5G: this project develops control and orchestration frameworks on top of OAI's RAN and CN modules, allowing for monitoring and controlling of the network. It includes Trirematrics and FlexCN platforms in its roadmap that provide SMO and CN control modules and a FlexRIC software we introduce below.

a) E2 agent and FlexRIC: FlexRIC provides an SDK that can implement a multi-vendor O-RAN compliant RT RIC that is specialized for a particular service (e.g., slice control, traffic control, etc.) with built-in service models (SMs) and support for the creation of further SMs [7]. OAI's FlexRIC design is meant to be extensible and compact with minimum overhead. Furthermore, unlike RICs provided by other projects, it follows an event-driven rather than poll-driven approach. The main modules contain an agent library that deploys E2-compatible agents in a base station and a server library that manages agents' connections, stores network information in the radio network information base (RNIB), and handles E2SM subscriptions. These subscriptions can be established by iApps, which are controller internal applications that either implement a specific control logic or expose E2SM subscriptions to xApps deployed on external controllers through different interfaces.

The agent library is radio access technology (RAT) and vendor-neutral, allowing multi-RAT and multi-vendor deployments. Agents can also connect to multiple controllers through the server library, which provides isolation between them. Furthermore, a virtualization layer with an agent can be implemented on top of a server deployment which allows recursive agent-server layers. This is beneficial in cases where we want to abstract out RAT heterogeneity or delegate control to multiple controllers per slice using different SMs.

B. O-RAN Software Community (OSC) [8]

OSC is founded by O-RAN and Linux Foundation, and it aims to provide software that is fully O-RAN compliant. The project generally encompasses all O-RAN-related components, RAN elements, and interfaces. We present some OSC projects below.

1) O-DU: This project is composed of two sub-projects. *O-DU* Low focuses on the baseband PHY reference design, including three interfaces: L1/Fronthaul;*O-DU Low*/*O-DU High*, and *O-DU Low*/accelerator. *O-DU High*, is responsible for implementing L2 blocks for 5G NR SA mode that include NR MAC, NR Scheduler, and NR RLC layers. *O-DU High* also provides *DU APP*, which configures and manages all O-DU operations, and interfaces with external entities (e.g., O-CU, RIC, etc.). Finally, it implements an O1 module to handle O1 communication.

2) O-CU: O-CU was supposed to provide O-CU UP. However, it seems the project was disbanded, and instead, OSC uses a binary test stub provided by Radisys for end-to-end testing.

3) Near-RT RIC: This project provides an initial RIC platform to support xApps with limited support for O1, A1, and E2 interfaces.

4) Non-RT RIC: In the context of Non-RT RIC, OSC provides a Non-RT RIC Control Panel, which provides administrative and operator functions through A1, like policy management and Near-RT RICs setup. Also, an A1 Simulator module is implemented, which terminates the A1 interface and allows testing of the Non-RT RIC without deploying Near-RT RICs. To support management functions, an SMO project implements O1 and O1/VES interfaces responsible for the configuration, management, and report handling of NFs. Finally, and OAM project provides administrative and operator functions for O-RAN components.

C. Open Networking Foundation (ONF) [9]

ONF was established as a project to develop software-defined networking (SDN) technologies, and currently, it is driven by operators and a community of developers. ONF developed its SD-RAN project, which provides a Near-RT RIC adapted to O-RAN specifications in its latest version at the time of writing this paper. Besides the Near-RT RIC, which is called *μONOS-RIC* due to its implementation being based on ONF's *ONOS* platform, SD-RAN provides open-source components for the control and user planes of CU and DU, a RAN simulator, and xApps development SDK. The CU/DU modules are derived from OAI's 5G RAN project (see III-A.2). SD-RAN leverages a microservice approach that is compatible with O-RAN specifications

D. Open AI Cellular (OAIC) [10]

Founded by USA National Science Foundation, OAIC uses OSC's Near-RT RIC (see III-B.3) on top of srsRAN [11], which provides components for implementing a complete end-to-end 4G and 5G NSA networks. For E2 implementation, OAIC leverages POWDER's E2 agents [12] in their architecture. Moreover, OAIC provides OAIC-T, an open-source AI cellular testing framework for testing xApps. It consists of a server that establishes the simulation environment according to input from configuration files and the actors that perform the test actions received from the server. Each actor contains an AI core component, and it can communicate with xApps or rApps under test and srsUEs to generate radio testing signals. Within

its framework, srsRAN provides srsUE to deploy 4G/5G UEs using ZeroMQ, srsENB as an eNB implementation with 5G NSA support, and srsEPC as a lightweight implementation of LTE EPC. At the same time, it lacks an implementation of 5G CN (they advertise using Open5GS [13] for 5G CN).

E. OpenRAN Gym [14], [15], [16]

Combining several software frameworks, OpenRAN Gym allows data acquisition of RAN performance indicators from emulators or testbeds and RAN control to test O-RAN-compliant solutions powered by AI/ML. The platform encompasses the following.

- Open experimental wireless platforms for acquiring RAN data and testing solutions (e.g., Colosseum, which is the world's largest wireless network emulator, Arena testbed, etc.),
- RAN software implementations using srsRAN or OAI stacks,
- SCOPE framework, which is used for data collection and control of RAN during run-time which also adds further networking and control functionalities (e.g., slicing) to the RAN software, and
- ColO-RAN provides a lightweight RIC adapted from OSC's RIC, allowing xApps/rApps to monitor KPMs and control the RAN.

Using these tools, solutions can be validated on the Colosseum emulator, for example, and then ported to heterogeneous testbeds seamlessly as described in [14]

F. Comparison of Platforms and Their Compatibility

Table I lists the perceived differences between the implementation options. Furthermore, in Fig. 2, we present the components used in currently available solutions and their combinations, and we also include other open-source CN projects not mentioned in our earlier discussion, which are compatible with some RAN implementations.

IV. xApp Implementation-Driven Ambiguities Related to O-RAN Architecture

While standardization bodies define how the O-RAN architecture should be implemented to address various applications, some ambiguities are observed while working on specific use cases. Here we focus on Beam Mobility Management (BMM-xApp) and Signaling Storm Detection xApps (SSD-xApp). The use cases related to those xApps are analyzed within O-RAN ALLIANCE's documents.

A. Example RRM xApp - Beam Mobility Management

One of the key technologies used in 5G NR is a Grid of Beams (GoB) beamforming. A UE is assigned to a specific beam (out of a static set) based on the downlink measurements of Reference Signal Received Power (RSRP). The measurements are typically carried using the Synchronization Signal Block (SSB), i.e., every 20 ms [22]. SSBs transmission for all beams lasts 5 ms. In this case, the main challenge is to deal with radio-link failures due to rapid changes in

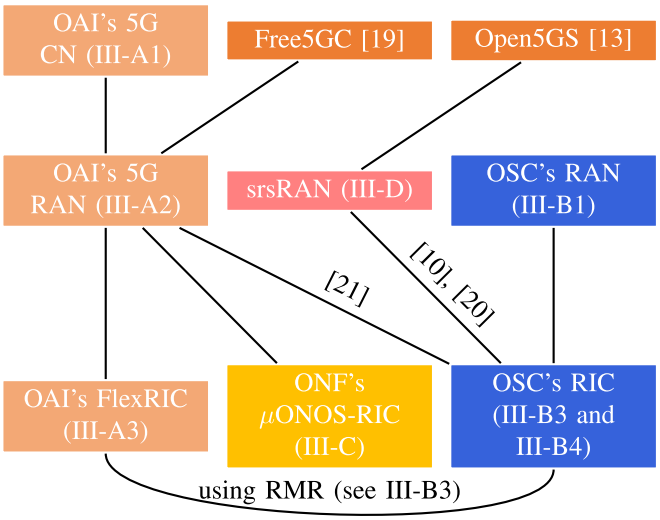


Fig. 2. Projects for building and testing a complete end-to-end 5G system with Open RAN functionalities and their compatibility. Starting from the top, the first row lists 5G CN projects, the second row mentions 5G RAN implementations, and the last row lists RIC implementations. Colors indicate the vendor: OAI (light orange); OSC (blue); ONF (yellow); srsRAN (red), and other vendors (dark orange).

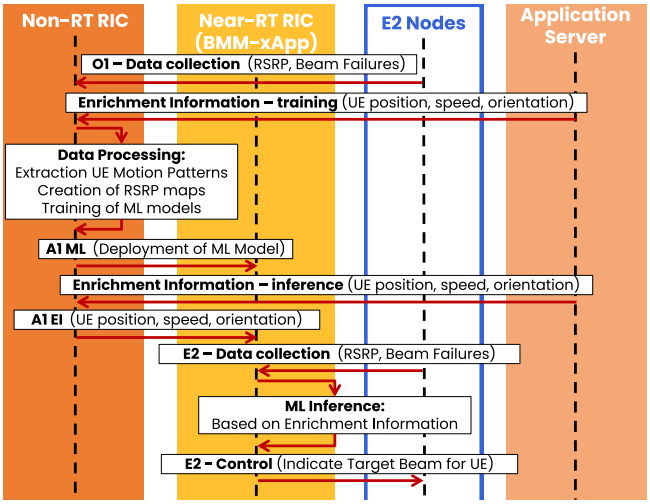


Fig. 3. The information flow between the BMM-xApp, and other O-RAN entities.

the radio environment when the UE moves fast. To avoid such situations, there is a need for AI/ML-assisted algorithms that utilize context information, e.g., UE location, to infer future target beams, possibly minimizing the number of beam reselections [23].

The challenge discussed above is addressed by the use case "AI/ML-assisted Beam Selection Optimization" from O-RAN ALLIANCE [23]. Its definition could be more specific, although O-RAN ALLIANCE specified utilized entities (e.g., Near-RT RIC) and interfaces (e.g., O1 and A1-ML). Below, we propose the remaining elements of the solution to the challenge mentioned above using the ML algorithm and data collected from E2 nodes. The information flow between the BMM-xApp, and other O-RAN entities is depicted in Fig. 3. The central concept of this development is to perform the most extensive computations related to data analysis and

TABLE I
COMPARISON OF EACH PLATFORM'S OPEN-SOURCE IMPLEMENTATIONS AND EXEMPLAR XAPPS

	OAI	**OSC**	**ONF**	**OAIC**
CN	• OAI CN	• No CN or EPC	• OMEC CN	• LTE EPC
RAN	• Better CPU and memory utilization than srsRAN [17] • Multiple UE simulation • MOSAIC5G E2 agents • No O1 interface implementation	• Radisys CU lacks integration with open-source CU and RU implementations • DU and DU App • O1 interface	• Leverages OAI's RAN modules • ONF's own RAN simulator with more features and capability to simulate a large number of UEs	• Easier to modify [17] • POWDER [12] E2 agents within srsRAN stack • Single UE simulation • No F1 interface for CU/DU split • No O1 interface
RIC	• Better CPU, memory utilization, and latency than OSC's RIC [7] • Recursive agent library for the abstraction of underlying topology • iApps have less overhead than xApps	• Completely O-RAN compliant • All O-RAN components including Non-RT RIC • Requires more resources due to containerization and microservice structure	• μONOS-RIC using ONOS modules • Code used in previous SDN activities and is therefore reliable • Good documentation • Latest version is fully O-RAN compliant	• Uses OSC's RIC
xApps	• Key performance metrics (KPMs) monitoring, slice monitoring and control, and traffic controller	• Anomaly detection, HelloWorld xApp, HW-go xApp, KPM monitoring, QoE predictor, RIC APP ML, RIC Measurement Campaign xApp, traffic steering, and GS-lite stream processing engine [18]	• onos-kpimon (KPM monitoring), onos-rsm (slice management), onos-mho (mobile handover for mobility management), onos-mlb (load balancing between cells), onos-pci (for managing PCI resources)	• Besides the xApps provided by OSC, OAIC introduced their own KPI monitor and slice control xApps
Lang. Lic.	• *C/C++* • OAIPL1.1	• *Python*, *Go* , and *C/C++* • ALV2 mostly besides CCLA4I	• *Go* • ALV2	• *C/C++* • GAGPLV3

training of the ML model in the Non-RT RIC. The Near-RT RIC receives the pre-trained ML model and uses it to infer UE target beams. First, the O1 interface is configured to provide Non-RT RIC with users' RSRP measurements and beam failure statistics from E2 nodes. The beam failure statistics are used to monitor ML model accuracy, i.e., when the observed number of beam failures increases, the ML model re-training is triggered. RSRP measurements are used to create an RSRP map for each beam, following the Radio Environment Map (REM) concept [24]. For this purpose, EI, specifically: the position, speed, and orientation of each user, is obtained from the Application Server (specifically - the location server). The obtained data are being processed in the Non-RT RIC. First, the location information is analyzed to extract the UE Motion Patterns. They are, e.g., histograms that represent the probability of future UE speed and orientation while in a particular location. A representative example can be a vehicular scenario. When users encounter a road intersection, most turn right, while only a few turn left. Next, the RSRP map is created, i.e., for each beam associated with a considered BS, the spatial distribution of RSRP is created by aligning location information from the external Application Server, and RSRP collected from E2 Nodes. The alignment can be done by comparing the data if these are accurately timestamped. These RSRP maps capture specific radio environment characteristics, e.g., obstacles in a particular location can block some beams. Both UE Motion Patterns and RSRP maps represent the radio environment and are used to train ML models. The ML models can be trained according to different optimization goals, e.g., minimizing beam reselections while maintaining users' QoS or maximizing SNR. Reinforcement Learning (RL) can be used as it learns through interaction with the environment (wireless network) [25]. After the training, the obtained ML model is transferred to Near-RT RIC via the A1 ML interface and deployed in the BMM-xApp to make inferences on target beams for UEs. To provide input to the deployed ML model, EI (precisely: location information) must be sent from the external Application Server to the BMM-xApp. This is done in a two-stage manner: first, EI is sent to the Non-RT RIC, and next, it is forwarded to the Near-RT RIC through the A1-EI. In addition, the E2 interface is configured to collect information about the RSRP and beam failures. First, the UE's localization is used in the ML inference performed by BMM-xApp, i.e., the target beam is selected. Secondly,

the BMM-xApp monitors beam failures to validate the ML model performance. If too many beam failures occur, it is a signal that the ML model is outdated. In such a case BMM-xApp can temporarily switch to the *emergency* mode in which some analytical beam management procedure based on RSRP reports is performed (e.g., [26]) until a new ML model is provided from the Non-RT RIC.

Recalling that this use-case is at its early stage of specification in O-RAN ALLIANCE, still, some implementation ambiguities are observed:

- **Location information** is currently not discussed within the O-RAN specifications; it is only mentioned as a specific type of EI message. However, it could be used by many xApps, and some of its aspects should be discussed within O-RAN ALLIANCE workgroups. The localization server should, at minimum, provide the following:
  - *Localization technique* that was used to obtain the location information; (There are many localization techniques of significantly different accuracy, e.g., an accuracy of 10 meters characterizes standard Global Navigational Satellite System (GNSS) receiver, while Real Time Kinematics (RTK) introduces only a centimeter-level error.)
  - *Available measurements* that can be provided together with the user's position, e.g., user's speed and bearing;
  - *Report time-intervals*; (If the location information is provided only once per second, the performance of BMM-xApp could be degraded as beam management can be triggered every 20 ms [22].)
  - *Delay* associated with passing the UE's localization information, which is required by Near-RT RIC, and transferred via Non-RT RIC as shown in Fig. 3. (Note that the recently introduced Y1 interface between Near-RT RIC and the Application Server can significantly lower this delay.)
- **Alignment of reported data in time**, i.e., precise timestamping of both RSRP and location information at the moment of measurement is crucial, e.g., for high-speed users who can travel a few meters during the time between the position was obtained, and the EI was received in Non-RT RIC.
- **ML Modules**
  - *Deployment of ML Modules within O-RAN architecture* should be clarified. At the current stage of standardization, there are several options in SMO and Non-RT RIC architecture where ML model training can be performed. For example, training can be performed either by a vendor-dependent module, dedicated rApp, within the rApp, or even outside of the Non-RT RIC and SMO.
  - *A1 interface* specifications, at their current state, do not explicitly define ML Model service operations [27].
- **E2 interface** lacks actions related to beam management [28], i.e., at this stage, it is unclear how BMM-xApp would enforce switching a particular user to the given beam.

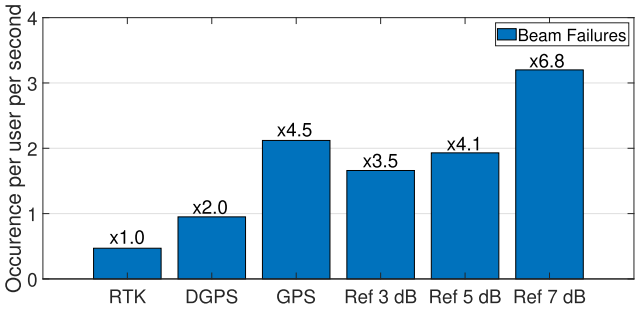


Fig. 4. Number of beam failures per user, per second versus the utilized localization technique.

To highlight the importance of the quality of location information for the BMM-xAPP relying on the REM, we have performed computer simulation studies in the scenario described in detail in [29]. The scenario considers a single Massive MIMO BS, operating at mmWaves frequency band, that supports eight beams. Within this cell, we have placed 300 UEs moving along a street, with a speed of 25 m/s, to reflect the road scenario. We have tested the BMM-xApp following the optimization goal of minimization of beam reselections while avoiding beam failures (situations when, for a given UE, the target beam has 8 dB higher RSRP than the current/source beam) under three localization techniques: RTK, Differential Global Navigation Satellite System (DGPS), and standard GPS. The standard deviations of their corresponding localization error are as follows [30]: 1 cm, 1 m, 6 m, for RTK, DGPS, and GPS, respectively. We have compared the BMM-xApp against the *Ref* beam management algorithm that relies on the static power margin [26], i.e., reselection happens for a given UE, when the target beam has RSRP higher by the margin over the current/source beam's RSRP. We have considered margins of 3, 5, and 7 dB, respectively. The resultant number of beam failures per user per second is depicted in Fig. 4. RTK provides almost perfect location information, but some beam failures occur due to channel variations. However, when additionally localization accuracy is degraded, more beam failures arise, i.e., compared to the RTK, it is about 2.0 and 4.5 times more beam failures while utilizing DGPS and GPS, respectively. Comparing the BMM-xApp that uses RTK against the *Ref* approach based on the static margin, it can be seen that number of beam failures is 3.5, 4.1, and 6.8 times higher for power margins of 3, 5, and 7 dB respectively. The BMM-xApp that utilizes slightly less accurate DGPS also outperforms *Ref*. However, when only standard GPS is available for the BMM-xApp, the accuracy of location information is not good enough, and the number of observed beam failures is worse than in the case of *Ref* with a power margin of 3 dB, and 5 dB. Thus, the information about the supported localization technique would be necessary for designing robust xApps.

B. Example Security xApp - Signalling Storm Detection

The signaling storm attack is aimed at causing Denial of Service (DoS) in a network by occupying radio resources in a CP by an adversary or malfunctioning device [31]. Such

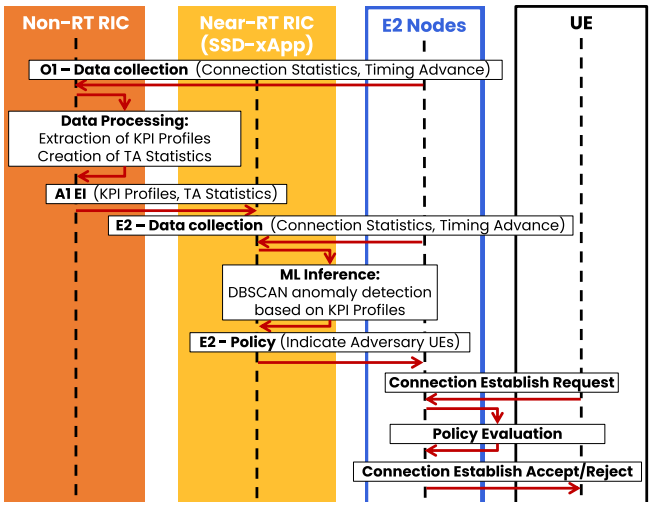


Fig. 5. The information flow between the SSD-xApp, and other O-RAN entities.

devices can persistently send control messages like registration requests that will be rejected after validation in the CN or can intentionally disconnect from the network after a successful registration. Such behavior is hazardous in the Internet of Things (IoT) networks. The IoT devices have low complexity and can be relatively easily hacked by adversaries to flood networks with CP messages, e.g., adversaries can install on the IoT device software that will constantly restart the device triggering the registration procedure. It is essential to notice that such a device will be authorized to connect to the network, and as such hard to be detected [32]. From this perspective, it is essential to equip 5G networks with an intelligent mechanism that can detect the signaling storm as close to its origin as possible, possibly at the stage of RAN. After detection, further communications with malfunctioning devices should stop to prevent flooding the CN with CP messages.

This xApp addresses a use case following requirements of O-RAN Signalling Storm Protection from [4] with a slight modification: here, both attack detection and mitigation are integrated into a single SSD-xApp to reduce the amount of communication overhead. The O-RAN ALLIANCE specifies the high-level roles of the O-RAN entities and utilized interfaces for this use case. As the other details, e.g., ML method and data exchanged with the E2 node, are missing, we propose our solution below, keeping it fully compliant with O-RAN specification. The SSD-xApp utilizes the Timing Advance (TA) parameter being computed and exchanged at the early stage of the device's registration procedure (i.e., Msg2: Random Access Response [33]). As this indirectly characterizes the distance electromagnetic wave travels between the UE and the BS, it is difficult to be falsified. As such, it can filter malfunctioning devices, creating an increased number of connection-establishment requests without interrupting CN functions, e.g., device authentication. The information flow between the SSD-xApp and other O-RAN entities is depicted in Fig. 5. It starts with configuring the O1 interface to provide Non-RT RIC with connection statistics, including registration requests, RRC connection establishment requests, etc., and related TAs extracted from Msg2. This data is processed within the Non-RT RIC to produce the so-called Key Performance Indicator (KPI) Profiles [34]. The KPI Profiles are the long-term statistics of a given KPI, e.g., the mean and standard deviation of the number of connection-establishment requests over the day. In addition, TA related to connection statistics is analyzed, e.g., in the form of histograms. The A1-EI is used to send KPI Profiles and TA statistics observed over a long period in Non-RT RIC to the SSD-xApp residing in Near-RT RIC. This step should repeat periodically, e.g., twice a day, or on an event basis, e.g., when a high number of new UEs is deployed in a factory. The SSD-xApp obtains from E2 nodes temporal information about the connection statistics (e.g., number of connection-establish requests over the last 5 minutes) and related TAs. Next, the SSD-xApp compares the long-term KPI Profile with temporal connection statistics computing the so-called anomaly values. It utilizes the unsupervised learning clustering algorithm Density-Based Spatial Clustering of Applications with Noise (DBSCAN) to detect the abnormal activity of users in the network, i.e., signaling storm. When the signaling storm is detected, the SSD-xApp analyses statistics of TA to produce a policy that will filter out connection establishment requests related to users associated with those TAs. The formulated policy is sent to the E2 Nodes via the E2 interface. Based on that policy, the E2 Node can accept or reject the connection-establish requests sent by the UEs by comparing their TAs with blacklisted TAs defined in the policy.

As with the BMM-xApp, also here some implementation ambiguities can be mentioned:

- **Resolution of TA** relies on the network configuration. A low resolution of TA will increase the number of devices having the same TA and potentially blocked. From this perspective, it might be useful to provide the xApp with some extra historical information about the UE context from the CN registers, to distinguish an adversary from a legitimate user, e.g., historical channel state information, network identifiers, etc.
- **Non-RT RIC** architecture is not specified in terms of storage processing of EI [35]. Regarding the KPI Profiles utilized by the SSD-xApp, it is unclear whether there would be some dedicated vendor-dependent Non-RT RIC module for processing and storing such xApp-provider-defined EI or whether some rApp would realize this functionality.
- **E2 interface** policy service is not clearly defined within the O-RAN specifications [28]. It might happen that E2 Nodes would not support rejecting connection establish requests based on the TA parameter.

To highlight the importance of the ambiguities mentioned above, we have studied the potential impact of the TA resolution on the number of legitimate users being rejected from the network when adversary activity is detected. We are considering a simulation setup described in our previous work [36]: a single cell of IIoT network of a 2 km radius, with 100 randomly located, stationary legitimate IIoT sensors and five adversaries. Intervals between legitimate users' connection requests follow the exponential distribution with a

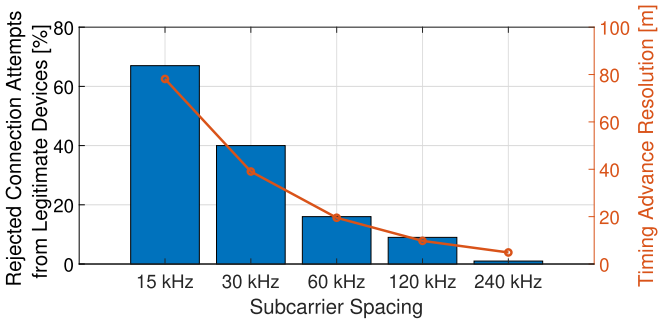


Fig. 6. The ratio of rejected connection attempts from legitimate devices, and calculated TA resolution versus the subcarrier spacing.

rate parameter equal to 5 per hour. Each adversary performs, on average three attacks per day consisting of 100 consecutive connection requests sent within the intervals of 5 s. Because the TA resolution is a function of the utilized subcarrier spacing, we have considered values proper for a 5G system: 15, 30, 60, 120, and 240 kHz, respectively. As we see in Fig. 6 the percentage of rejected connection attempts from legitimate devices drops with the subcarrier spacing, as a result of increasing TA resolution. It can be seen that for high values of subcarrier spacing, detection of adversary almost doesn't affect the performance of legitimate users, i.e., all their connection attempts are accepted. This is because of high TA resolution for subcarrier spacing of 240 kHz, i.e., about 5 m. On the other hand, while utilizing low subcarrier spacing of 15 kHz the spatial resolution of TA is significantly decreased to about 78 m. As a result, more than 60% of legitimate devices are rejected from the network because their TA is the same as the TA of the detected adversary.

V. xApp Implementation-Driven Conclusions

Contrary to the prior chapter, where we focused on the ambiguity related to xApp development, here we concentrate on issues related to the detailed application implementation on selected open RIC platforms. For comparison, we have chosen two xApps - Traffic Steering xApp (TS-xApp) and QoS-Based Resource Allocator xApp (QRA-xApp), which consider use-cases standardized by the O-RAN ALLIANCE specifications [4].

The xApps have been deployed within the environment running on the virtual machine with the Ubuntu operating system (OS). It is based on the architecture packed in Kubernetes pods and Docker images. To ensure proper implementation of the xApps, the following virtual hardware requirements are obligated: a) processor with at least 2 cores, b) Random Access Memory (RAM) with the size of min. 8 GB, c) Read-Only Memory (ROM) with a minimum size of 50 GB, d) Ubuntu OS version 20.04.5 LTS.

A. Traffic Steering xApp

TS-xApp addresses the use case #5: O-RAN Traffic Steering from [4]. It allows the dynamic switching of mobile users between cells available in the access network. The purpose of such a mechanism is to manage the current mobile traffic

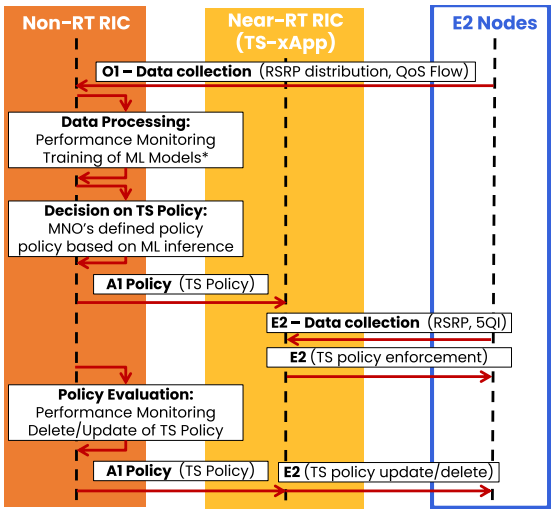


Fig. 7. The information flow between the TS-xApp, and other O-RAN entities.

to ensure the radio system's high performance. Depending on actual needs, the MNO can realize various TS targets such as guaranteeing equal traffic load for all nodes (load balancing), separating users with different Quality-of-Service (QoS) demands (service-based association), supporting the reduction of energy consumption, and many others.

In the TS xApp, the user association is performed through the E2 Interface using an O-RAN-defined handover control mechanism. The decisions about switching users among cells are made based on the RSRP distribution reports received through the E2 Interface and policies that the Non-RT RIC sends through the A1 interface. The rules, which indicate preferred and forbidden cells for a particular UE, can be found inside these policies. The preferences can be oriented to users assigned to a specific slice (slice-oriented approach) or having strictly specified identification (user-centric approach). The A1 policies are exchanged between Non-RT RIC and TS xApp in the form of JavaScript Object Notation (JSON) files, which are prepared according to the schema of the "Traffic Steering Preferences" type standardized by the O-RAN ALLIANCE [4], [37]. The information flow is depicted in Fig. 7.

TS-xApp has been integrated with the SD-RAN environment provided by the ONF; it can handle connections with the μONOS RIC components of the SD-RAN. Furthermore, it can interpret the received E2 and A1 messages correctly and suggest (to RIC) performing adequate handover operations, the results of which are reflected in the RAN Simulator. The source code of the xApp can be found in [38]. In Tab. II, the results for the TS xApp performance are presented. The considered, intentionally-simple scenario consisted of two single-cell base stations and a single UE terminal moving between the coverage areas of both BSs. Within the tests, three different UE-oriented policies were enforced. Those policies indicated the preferences for connection handling with the user by a particular cell - PREFER, AVOID, and FORBID. The UE recognized cells marked in a policy with these labels as cells by which the UE should, should not, and must not be served, respectively. Thus, referring to Tab. II, it can be

TABLE II
ASSOCIATION OF THE UE WITHIN THE NETWORK BY THE TS xAPP ACCORDING TO DIFFERENT POLICIES

POLICY NAME	USER ASSOCIATION TIME PART [%]			
	ENFORCED FOR 1st CELL		ENFORCED FOR 2nd CELL	
	1st CELL	2nd CELL	1st CELL	2nd CELL
NONE	50	50	50	50
PREFER	75	25	25	75
AVOID	25	75	75	25
FORBID	0	100	100	0

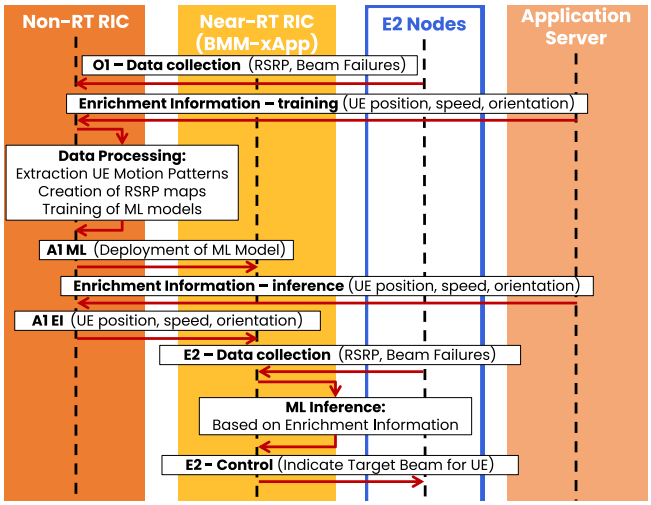


Fig. 8. The information flow between the QRA-xApp, and other O-RAN entities.

observed that when the connection between the user and cell is marked with the PREFER label, this link is handled for 75% of the observation time. The AVOID mark causes the opposite result – the UE is served by such a cell for 25% of the observation time. Next, the FORBID label resulted in not serving the user by a given cell. Finally, in the case where no policy was enforced for the TS xApp, the UE was associated with a cell based on the RSRP report. Thus, it was noticed the user was served for 50% of the observation time by one cell and 50% by another. Please note that the whole TS algorithm could be complemented with other functionalities, such as load balancing between the base station. Thus, for a given service-based user association, the TS should keep the balance between the cell load to optimize the usage of resources. However, to achieve this, the policies shall be generated flexibly, reflecting both operator needs and the current state of the network.

B. QoS-Based Resource Allocation xApp

QRA-xApp addresses use case no. 8: O-RAN QoS Based Resource Optimization from [4]. It is responsible for splitting radio resources in the form of Physical Resource Blocks (PRBs) among available slices within the network. With the QRA xApp, the MNOs can manipulate the radio resources allocated by the scheduler to manage the networks' performance by allocating more PRBs for high-performance slices (e.g., Mobile Broadband - MBB) and simultaneously reducing the number of resources for slices demanding low data rate (e.g., Voice services).

TABLE III
RADIO RESOURCE ALLOCATION FOR DIFFERENT SCHEMAS WITHIN THE QRA xAPP

UE ID	5QI	BANDWIDTH PART [%]		
		EQUAL	PREFER-3	RESERVE
1	1	12.5	6.25	5
2	2	12.5	6.25	10
3	4	25	12.5	40
4	2	12.5	6.25	10
5	3	25	62.5	30
6	1	12.5	6.25	5

This allocation of radio resources is done to meet the SLA targets defined inside policies (passed to xApp in the form of JSON files by the Non-RT RIC through the A1 Interface) by basing on measurement reports received through the E2 Interface for a particular slice served by some gNB. The SLA targets are specified in the A1 policy file as a throughput rate expressed in [bps], which can be translated to the number of needed PRBs (and vice versa) by taking into account current propagation conditions for a slice (e.g., SNR/RSRP distribution, number of active UEs, service/slice types, etc.). This group of SLA targets specified inside A1 policies consists of UE- or slice-oriented parameters such as Guaranteed and Maximum Throughput per Slice, Maximum Throughput per UE, Maximum Number of UEs per Slice, etc. The O-RAN ALLIANCE has defined the used shape of A1 policies as the schema of policy type called "SLA Target" [4], [37]. Fig. 8 depicts the information flow between the involved entities.

QRA-xApp, similarly to TS-xApp, has been integrated and tested using the ONF's SD-RAN environment. The QRA xApp connects to the SD-RAN's μONOS RIC components. Thanks to the correct interpretation of received E2 and A1 messages, the xApp performs adequate resource-allocation-related operations, the results of which (delivered via RIC to E2 nodes) could be visible in real-time mode in the form of terminal logs.

In Tab. III, the results for the QRA xApp are presented. The considered scenario consisted of two single-cell base stations and six UEs moving simultaneously between the locations in the coverage of each BSs. Each user served within the network could belong to a different slice. All UEs connected to a specific network cell and using the same service type (denoted by the 5G QoS Identifier – 5QI) were grouped in a particular slice. Within the tests, four service types (5QI equal to 1, 2, 3, or 4) and three different schemas of radio resource allocation (EQUAL, PREFER-X, and RESERVE) have been taken into account. According to the EQUAL approach, all available PRBs were divided among existing slices equally. Next, the PREFER-X schema (where X is the number indicating the service type, i.e., the 5QI, of a particular slice – 1, 2, 3, or 4) shares all the resources among the slices in the ratio of 5:1 for ones with "preferred" service type (5QI) to the rest of them. Finally, the RESERVE approach divides all PRBs within the cell among available slices in the ratio of 5X, where X is the number that indicates the service type of a given slice (5QI). Thus, for our scenario with four different service types (1, 2,

TABLE IV
CHALLENGES FACED DURING xAPP DEVELOPMENT AND TESTING

Aspect	Challenges
Simulator	• Available RAN simulators do not provide complete functionality needed to test different specific practical scenarios (e.g., different network size, base station capabilities, network operation duration, etc.)
Conflicts	• Absence of conflict mitigation units prevents testing the operation of multiple xApps working simultaneously • Multiple A1 policies that could be turned on simultaneously should be verified against each other beforehand
SDK/API	• Standard compliance: base stations or simulators do not provide the functions or parameters needed for complete O-RAN functionality implementation • Abstraction of O-RAN messages that implement certain functionalities (e.g., RSRP monitoring, handover control, etc.) would simplify xApp development process • Interoperability between components like simulators and RICs • Exemplar xApps should be provided and they should cover the functionality of the platform as much as possible

3, and 4), the ratio of sharing the resources for RESERVE schema equals 5:10:15:20.

In Table IV, we summarize the challenges faced during the development, deployment, and testing of xApps using different platforms.

VI. CHALLENGES FOR O-RAN/INCENTIVES TO O-RAN TRIGGERED RESEARCH

Following the discussion on xApp/rApp implementation and deployment issues, we try to identify the key challenges that appear on the Open RAN development path in this section.

A. Challenge A: The Need for Intelligent Conflict Management

Intelligent RAN control functions enabled in the Near-RT RIC with the introduction of xApps allow flexibility in adapting network operation characteristics. While implementing a single application, there is no need for any mechanism responsible for conflict management; what is necessary is only the subscription functionality so that the particular xApp or rApp can request access to specific parameters or metrics through standardized service models. On the other hand, having multiple xApps/rApps, developed by various third-party providers, working simultaneously in RICs will inevitably lead to conflicts between control actions affecting the E2 Nodes finally. Thus, incorporating two (or more) xApps/rApps immediately entails the need for stable and precise solutions for conflict management [39]. The xApp/rApp developer has to be aware of the applied policy in case of any prospective conflicts - whether any priority or hierarchy between the applications shall be used and how it may impact the functioning of the application. Based on our implementation experience, it is one of the key challenges that must be effectively solved to enable reliable xApp provisioning.

B. Challenge B: Security

Another critical point that was immediately observable during the implementation of the xApps/rApps is related broadly to Open rAN security - both on the architectural side and from the perspective of xApp/rApp delivery by the third party. When discussing the security of an O-RAN architecture, one should note that the attack surface is expanded compared to the standard radio segment of a mobile communication network. This surface contains "traditional" attacks related to the omnipresent radio transmission medium, cyberattacks related to virtualization (softwarization) of RAN functions, i.e., attacks on xApps, rApps, and edge Artificial Intelligence (AI) algorithms residing in O-RAN and Multi-access Edge Computing entity (MEC), as well as attacks related to O-RAN interfaces.

The O-RAN specification and *openness* of the radio interface poses challenges for the entire network security. Inadequately defined and poorly secured O-RAN applications and interfaces (including the front-haul interface, O1, O2, A1, and E2) can potentially be targets of attacks. Attackers can utilize these new open interfaces to attack the system, which could lead to a denial of service, data tampering, or data leaking, all of which indirectly impact the system's security. Each O-RAN interface and function may be subject to different threats, and each threat will have a particular impact; thus, for each threat, specific security measures and solutions must be used for all aspects and assets [40]. Finally, AI and ML algorithms residing at the network edge (a consequence of the ML-as-a-Service paradigm for 5G/6G networks) become a target of a new type of attack - attacks on AL/ML. These threats can be classified as (i) *poisoning attacks* manipulating the data or the learning algorithm in the model training phase, (ii) *evasion attacks* aiming at the inference stage (test phase) based on the previously learned model, and (iii) *inference attacks* aiming at recovering the training data or their labels, discovering the model architecture and its parameters [41].

At the same time, O-RAN architecture can increase security in radio access networks because it allows for running xApps in Near-RT RIC, which can be developed to continuously monitor and analyze security threats and protect RAN from malicious and illegal access to network segments. It makes it possible to detect threats much faster before they affect the operation of the entire network. xApps can be developed for specific types of threats in a given network that can be detected closer to their occurrence. AI/ML algorithms can also be designed to improve security, e.g., by detecting various anomalies in radio traffic. Future research should aim to develop such xApps for O-RAN security despite expanded surface attacks.

C. Challenge C: The Need for Complete Automation and Testing Procedures

Another challenge raised immediately during the implementation of all the applications discussed above is the stringent need for broad automation of the xApp/rApp delivery, testing, and deployment process. As the applications can be tested, verified, and installed manually at the current stage, it is impossible to keep this stage in the future. Thus, based on the gained experience, one of the key challenges at the current stage of O-RAN development is the lack of automation related to testing and installing the xApps/rApps on the RIC platforms. The template-based approach for xApp and rApp development is discussed in [42]. A general automated, distributed, and AI-enabled testing framework has been presented in [43], to test AI models deployed in O-RAN.

This currently requires manual integration of the application every time a new one is to be deployed. There is no unified way to smoothly introduce new/upgraded xApps to the system, which consumes the resources of both the providers and the operators/customers. The xApp providers utilize the resources for this purpose instead of focusing on developing and improving the algorithms. At the same time, the customer/receiver needs to use more time to check that the xApp performs according to its design manually.

D. Challenge D: Portability

Yet another topic that yields currently cumbersome tasks is the portability of xApps/rApps between RIC platforms. What has been heavily experienced is that having the same core algorithm requires significant manual integration work to deploy it as an xApp on one RIC, with a more or less similar amount of work, when putting the same algorithm onto xApp for a different RIC. There are several reasons influencing this situation. First, the various commercial and open-source RIC platforms have different maturity levels, where each focuses on another aspect. Second, the standardization of the RICs and E2 and A1 interfaces still needs to mature enough to have a clear implementation guide for the vendors. And finally, there is a lack of a standard for SDK/API/CDK such that the xApp/rApp could be ported from one RIC to another with minimal intervention to the packaging of the xApp.

Due to the above, when having an algorithm, the xApp developer needs first to get up to speed with the RIC platform and accompanied SDK to surround the xApp with the proper interfacing. There is yet another aspect to it, which is not directly related to the RIC platform itself but to the corresponding E2 nodes, which it works with. It relates to integrating the RIC with the particular RAN software, which may utilize a different set of, e.g., E2 service models or other versions of the same E2 service model. In such a setup, the xApp may only get some of the required parameters from the E2 node, which the RIC platform works with. This requires modification in the xApp itself so that the algorithm takes into account either fewer parameters or different parameters compared to a different RIC-CU-DU constellation.

E. Challenge E: Ambiguity in Implementation - Processing Resources Optimization

Finally, from the perspective of xApp/rApp functionality design and testing, the final challenge is related to the ambiguity in implementation. While the O-RAN ALLIANCE defines use cases with examples of messages exchanged between nodes, the xApp/rApp developers should have freedom of implementation limited only by the interface specification. Only in this case long-term development and improvement of applications are possible. It will resemble a market where various products (applications) can compete and the most suitable (for a given network) solution can be implemented. For example, the BMM-xApp, as described in Sec. IV-A, can be implemented using both the ML modules in Non-RT RIC and xApp in Near-RT RIC. However, similar results, i.e., a decision of a beam reselection sent to gNB, can be obtained by a single rApp, xApp, or a combination of rApp and xApp. The various solutions may use different sets of measurements for learning purposes. The problem becomes even more significant while considering a use case not considered by the O-RAN specification. To implement such an xApp/rApp, developers must have sufficient freedom. This shows that the set of parameters exposed on interfaces should be as broad as possible. On the other hand, each Application should be constantly monitored for the amount and type of information exchanged on the interfaces. Additionally, a responsible RIC (directly or indirectly, first to get support from SMO) should take care of the computational and storage resources required by a given application. If unlimited freedom is given to developers, the application may poorly scale with, e.g., the number of UEs or operation time. The application should be *killed* if the limit is reached and reported to the community and developers.

VII. Conclusion

Open RAN as the technology is still in one of its initial phases of development. Much effort is put toward a precise and adequate definition of various standards, reflecting different aspects of the Open RAN community. Moreover, from a scientific perspective, numerous projects and activities have recently started that target many vivid and essential problems related to the fair functioning of the complete open system. However, the process should also consider the experience gained during initial implementation experiments and deployments. In this paper, we have described the lessons learned during the practical implementation of some xApps, selected based on the indications from the O-RAN ALLIANCE documents. It has been shown that from the perspective of xApp/rApp algorithmic design, the overall architecture still has a bit of ambiguity. It limits the scope of perspective investigation of the proposed solutions. Next, in-detail implementation of the selected applications led to identifying the key modifications and adjustments that could improve the impact of the open-source RIC platforms. Finally, the overall discussion on the xApp development and deployment process allowed us to identify precisely five key challenges that must be handled in the near future. As these challenges impact various aspects of the open RAN concept, it is evident that joint efforts from

academia, standardization body, and industry are necessary. We claim that with tight cooperation between these three sectors, the further development of the open, disaggregated, flexible, and modular radio access networks will be expanded.

REFERENCES

[1] *O-RAN ALLIANCE*. Accessed: Oct. 8, 2023. [Online]. Available: https://www.o-ran.org

[2] *O-Ran Architecture Description, V.8.0*, O-RAN ALLIANCE WG, O-RAN ALLIANCE e.V., Alfter, Germany, Mar. 2023.

[3] *Near-Real-Time Ran Intelligent Controller Near-RT RIC Architecture, V.2.1*, O-RAN ALLIANCE WG3, O-RAN ALLIANCE e.V., Alfter, Germany, 2022.

[4] *Use Cases Detailed Specification, V.8.0*, O-RAN ALLIANCE WG1, O-RAN ALLIANCE e.V., Alfter, Germany, Jul. 2022.

[5] Telecom Infra Project, *Open RAN Project Website*, 501(c)(6) Non-Profit, USA, 2022.

[6] *OpenAirInterface—5G Software Alliance for Democratising Wireless Innovation*. Accessed: Sep. 21, 2022. [Online]. Available: https://openairinterface.org/

[7] R. Schmidt, M. Irazabal, and N. Nikaein, "FlexRIC: An SDK for next-generation SD-RANs," in *Proc. 17th Int. Conf. Emerg. Netw. EXperiments Technol.*, Dec. 2021, pp. 411–425, doi: 10.1145/3485983.3494870.

[8] *O-RAN Software Community*. Accessed: Sep. 13, 2022. [Online]. Available: https://o-ran-sc.org/

[9] *Open Networking Foundation*. Accessed: Oct. 6, 2022. [Online]. Available: https://opennetworking.org/

[10] *Open AI Cellular (OAIC)*. Accessed: Nov. 8, 2022. [Online]. Available: https://www.openaicellular.org

[11] *srsRAN—Your Own Mobile Network*. Accessed: Nov. 23, 2022. [Online]. Available: https://www.srslte.com/

[12] J. Breen et al., "POWDER: Platform for open wireless data-driven experimental research," in *Proc. 14th Int. Workshop Wireless Netw. Testbeds, Experim. Eval. Characterization*. New York, NY, USA: Association for Computing Machinery, Sep. 2020, pp. 17–24.

[13] S. Lee. *Open5GS*. Accessed: Jun. 2, 2022. [Online]. Available: https://open5gs.org/open5gs/

[14] L. Bonati, M. Polese, S. D'Oro, S. Basagni, and T. Melodia, "Open-RAN gym: AI/ML development, data collection, and testing for O-RAN on PAWR platforms," *Comput. Netw.*, vol. 220, Jan. 2023, Art. no. 109502.

[15] L. Bonati, M. Polese, S. D'Oro, S. Basagni, and T. Melodia, "OpenRAN gym: An open toolbox for data collection and experimentation with AI in O-RAN," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2022, pp. 518–523.

[16] M. Polese, L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, "ColO-RAN: Developing machine learning-based xApps for open RAN closed-loop control on programmable experimental platforms," *IEEE Trans. Mobile Comput.*, vol. 22, no. 10, pp. 5787–5800, Oct. 2023.

[17] F. Gringoli, P. Patras, C. Donato, P. Serrano, and Y. Grunenberger, "Performance assessment of open software platforms for 5G prototyping," *IEEE Wireless Commun.*, vol. 25, no. 5, pp. 10–15, Oct. 2018.

[18] *RIC Applications (RICAPP)—RIC Applications—Confluence*. Accessed: Sep. 18, 2022. [Online]. Available: https://wiki.o-ran-sc.org/pages/viewpage.action?pageId=1179662

[19] *Free5GC*. Accessed: Sep. 24, 2022. [Online]. Available: https://www.free5gc.org/

[20] D. Johnson, D. Maas, and J. Van Der Merwe, "NexRAN: Closed-loop RAN slicing in POWDER—A top-to-bottom open-source open-RAN use case," in *Proc. 15th ACM Workshop Wireless Network Testbeds, Exp. Eval. Characterization*. New York, NY, USA: Association for Computing Machinery, 2022, pp. 17–23.

[21] *O-RAN Alliance—Virtual Exhibition*. Accessed: Oct. 16, 2022. [Online]. Available: https://www.virtualexhibition.o-ran.org/classic/generation/2021/category/intelligent-ran-control-demonstrations/sub/intelligent-control/140

[22] C. N. Barati, S. Dutta, S. Rangan, and A. Sabharwal, "Energy and latency of beamforming architectures for initial access in mmWave wireless networks," *J. Indian Inst. Sci.*, vol. 100, no. 2, pp. 281–302, Apr. 2020.

[23] *Massive MIMO Use Cases, Technical Report, V.1.0*, O-RAN ALLIANCE WG1, O-RAN ALLIANCE e.V., Alfter, Germany, Jul. 2022.

[24] J. Perez-Romero et al., "On the use of radio environment maps for interference management in heterogeneous networks," *IEEE Commun. Mag.*, vol. 53, no. 8, pp. 184–191, Aug. 2015.

[25] H. Lee, Y. Jang, J. Song, and H. Yeon, "O-RAN AI/ML workflow implementation of personalized network optimization via reinforcement learning," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2021, pp. 1–6.

[26] F. Abinader, C. Rom, K. Pedersen, S. Hailu, and N. Kolehmainen, "System-level analysis of mmWave 5G systems with different multi-panel antenna device models," in *Proc. IEEE 93rd Veh. Technol. Conf. (VTC-Spring)*, Apr. 2021, pp. 1–6.

[27] *A1 Interface: Application Protocol, V.3.02*, O-RAN ALLIANCE WG2, O-RAN ALLIANCE e.V., Alfter, Germany, Jul. 2022.

[28] *Near-Real-Time Ran Intelligent Controller E2 Service Model (E2SM), Ran Control, V.1.03*, O-RAN ALLIANCE WG3, O-RAN ALLIANCE e.V., Alfter, Germany, Oct. 2022.

[29] M. Hoffmann and P. Kryszkiewicz, "Beam management driven by radio environment maps in O-RAN architecture," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2023, pp. 1–6.

[30] P. Misra and P. Enge, *Global Position Systems: Signals, Measurements and Performance*. Lincoln, MA, USA: Ganga-Jamuna Press, 2006.

[31] F. Francois, O. H. Abdelrahman, and E. Gelenbe, "Impact of signaling storms on energy consumption and latency of LTE user equipment," in *Proc. IEEE IEEE 17th Int. Conf. High Perform. Comput. Commun., 7th Int. Symp. Cyberspace Saf. Secur., IEEE 12th Int. Conf. Embedded Softw. Syst.*, Aug. 2015, pp. 1248–1255.

[32] M. Pavloski, "Detecting and mitigating storm attacks in mobile access to the cloud," in *Proc. IEEE Int. Conf. Fog Comput. (ICFC)*, Jun. 2019, pp. 53–58.

[33] *Technical Specification Group Radio Access Network; NR; Medium Access Control (MAC) Protocol Specification (Release 17)*, document TS 38.214, V.17.2.0, 3GPP, Sep. 2022.

[34] L. Bodrog, M. Kajo, S. Kocsis, and B. Schultz, "A robust algorithm for anomaly detection in mobile networks," in *Proc. IEEE 27th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Sep. 2016, pp. 1–6.

[35] *Non-RT RIC Architecture, V.2.0*, O-RAN ALLIANCE WG2, O-RAN ALLIANCE e.V., Alfter, Germany, Jul. 2022.

[36] M. Hoffmann and P. Kryszkiewicz, "Signaling storm detection in IIoT network based on the open RAN architecture," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Hoboken, NJ, USA, 2023, pp. 1–2. [Online]. Available: https://ieeexplore.ieee.org/document/10226043, doi: 10.1109/INFOCOMWKSHPS57453.2023.10226043.

[37] *A1 Interface: Type Definitions, V.4.0*, O-RAN ALLIANCE WG2, O-RAN ALLIANCE e.V., Alfter, Germany, Oct. 2022.

[38] *Rimedo Labs Traffic Steering xApp*. Accessed: Sep. 25, 2022. [Online]. Available: https://github.com/RIMEDO-Labs/rimedo-ts

[39] C. Adamczyk and A. Kliks, "Conflict mitigation framework and conflict detection in O-RAN near-RT RIC," *IEEE Commun. Mag.*, early access, May 18, 2023, doi: 10.1109/MCOM.018.2200752.

[40] C. T. Shen et al., "Security threat analysis and treatment strategy for ORAN," in *Proc. 24th Int. Conf. Adv. Commun. Technol. (ICACT)*, Feb. 2022, pp. 417–422.

[41] C. Benzaïd and T. Taleb, "AI for beyond 5G networks: A cyber-security defense or offense enabler?" *IEEE Netw.*, vol. 34, no. 6, pp. 140–147, Nov. 2020.

[42] A. Kliks, M. Dryjanski, V. Ram, L. Wong, and P. Harvey, "Towards autonomous open radio access networks," *ITU J. Future Evolving Technol.*, vol. 4, no. 2, pp. 251–268, 2023.

[43] B. Tang, V. K. Shah, V. Marojevic, and J. H. Reed, "AI testing framework for next-G O-RAN networks: Requirements, design, and research opportunities," *IEEE Wireless Commun.*, vol. 30, no. 1, pp. 70–77, Feb. 2023.

Marcin Hoffmann (Graduate Student Member, IEEE) received the M.Sc. degree (Hons.) in electronics and telecommunication from the Poznań University of Technology in 2019, where he is currently pursuing the Ph.D. degree with the Institute of Radiocommunications. He is also a Senior Research and Development Engineer with the Rimedo Labs working on O-RAN software development solutions. His research interests are the utilization of machine learning and location-dependent information for the purpose of network management.

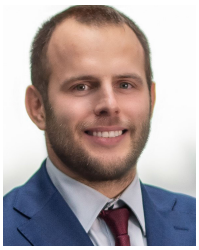
Salim Janji (Graduate Student Member, IEEE) received the B.Sc. degree in electrical engineering from the University of Balamand in 2017 and the M.Sc. degree in electronics and telecommunications from the Institute of Radiocommunications in 2020. He is currently pursuing the Ph.D. degree with PUT focusing on UAV base stations. He is also a Teaching Assistant. He is also a Research and Development Engineer with the Rimedo Labs. Besides UAV base stations, his research interests include reinforcement learning in wireless networks, reconfigurable intelligent surfaces (RIS), and O-RAN.

Adam Samorzewski (Graduate Student Member, IEEE) received the master's degree in electronics and telecommunications from the Poznan University of Technology, where he is currently pursuing the Ph.D. degree. He is currently a Research and Development Engineer with the Rimedo Labs developing software for O-RAN systems. His research interests include sustainable radio resource management in wireless systems supplied by renewable energy sources, wireless systems, radio resource management, and energy saving.

Łukasz Kułacz (Member, IEEE) received the Ph.D. degree in telecommunications from the Poznan University of Technology (PUT), Poland, in 2022. He is currently a Research and Teaching Assistant with the Institute of Radiocommunications, PUT. He is also a Senior Research and Development Engineer and the Technical Team Leader with the Rimedo Labs. He is involved in both, national, and international research projects. His research interests include the software-defined radio and utilization of context information for radio network operation improvement, in particular in the radio resource allocation process improvement.

Cezary Adamczyk is currently pursuing the Ph.D. degree with the Poznan University of Technology, conducting research in the field of radio resource optimization in open RAN, currently focusing on mitigation of conflicts between control decisions. He is also a Specification Engineer at one of the world's leading RAN provider, working on the development of O-RAN Alliance standards for the Open Fronthaul interface and their integration into the company's products.

Marcin Dryjański (Senior Member, IEEE) received the Ph.D. degree (Hons.) from the Poznan University of Technology in September 2019. Over the past 15 years, he was a Research and Development Engineer and a Consultant, a Technical Trainer, the Technical Leader, and an Advisor. He has been involved in 5G design since 2012, when he was the Work-Package Leader in the FP7 5GNOW Project. Currently, he is the CEO and a Principal Consultant with the Rimedo Labs. He is the coauthor of many articles on 5G and LTE-advanced pro and open RAN.



Pawel Kryszkiewicz (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees (Hons.) in telecommunications from the Poznan University of Technology (PUT), Poland, in 2010 and 2015, respectively. He is currently an Associate Professor with the Institute of Radiocommunications, PUT. He was involved in a number of international research projects. His main fields of interests are multicarrier signal design for green communications and problems related to the practical implementation of massive MIMO systems.

Adrian Kliks (Senior Member, IEEE) received the Postdoctoral degree in technical computer science and telecommunications in February 2019. He is currently an Associate Professor with the Poznan University of Technology and has taken part in numerous industrial and commissioned projects. He leads OPUS projects on V2X communication and RISes. In 2014 and 2016, he was the IEEE Membership Development/Web Visibility Chair of the EMEA Area. Since 2019, he has been the Editor-in-Chief of the *Journal of Telecommunications and Information Technology* of the Institute of Communications. A Vice Chair of the IEEE VTS Polish Chapter. He is the Co-Founder of the PUT spin-off company—Rimedo Labs.

Hanna Bogucka (Senior Member, IEEE) is currently a Professor with the Institute of Radiocommunications, PUT. She is involved in research in wireless cognitive and green communication. She is a member of the Polish Academy of Sciences. She also serves as the Member-at-Large of the IEEE Communications Society Board of Governors and Europe Regional Chair for the IEEE ComSoc Fog/Edge Industry Community.