O-RAN and RIC compliant solutions for Next Generation Networks

Marco Silva*, João Pedro Fonseca†‡, David Perez Abreu§*, Paulo Martins†, Paulo Duarte†, Raul Barbosa†‡, Bruno Mendes†‡, João Silva†, Adriano Goes†, Marco Araújo†, Bruno Sousa *, Marilia Curado*§, José Santos¶
*University of Coimbra, Centre for Informatics and Systems of the University of Coimbra, Department of Informatics Engineering, Coimbra, Portugal
†Capgemini Engineering, Porto, Portugal
‡Instituto de Telecomunicações and University of Aveiro, Aveiro, Portugal
§Laboratory for Informatics and Systems, Instituto Pedro Nunes
¶University of Coimbra, Centre for Mathematics of the University of Coimbra, Department of Mathematics, Coimbra, Portugal

***Abstract*—The Radio Access Network (RAN) is undergoing deep changes in the transition to beyond 5G systems. The Open-Radio Access Network (O-RAN) alliance aims to split the RAN architecture and support heterogeneity. At the same time, Open-Source Software (OSS) solutions like OpenAirInterface and srsRAN initiatives attempt to incorporate several stakeholders. This work proposes an End-to-End (E2E) orchestration framework using OSS solutions that are O-RAN compliant. A high-level architecture is presented focused on the RAN Intelligent Controller (RIC): the connection between Near-Real Time RAN Intelligent Controller (Near-RT RIC) with Service Management and Orchestration via the A1 interface and the connection with O-RAN Network Functions (E2 Nodes) via the E2 interface. The proposed architecture was validated on an experimental prototype. The main results compare state of the art OSS solutions status for deploying Near-RT RIC and RAN network functions. Our findings focused on the RAN functions interoperability with the RIC.**

***Index Terms*—Beyond 5G, O-RAN, OSS, Network Functions Virtualization, RAN Intelligent Controller**

I. Introduction

Next-generation Fifth-Generation (5G) and Sixth-Generation (6G) communication networks are transitioning from an inflexible and monolithic system to a flexible and disaggregated architecture to accommodate service heterogeneity and technology coordination [1]. Access networks are transforming digitally to serve a new generation of beyond 5G services. These networks aim to provide more throughput and lower latency on the radio technology side. Opening the Radio Access Network (RAN) to an intelligent and completely interoperable component increases its agility and effectiveness for critical services [2]. Towards this goal, the Open-Radio Access Network (O-RAN) Alliance promotes cloud-native software innovations such as microservices, containerized and service-based components, and stateless architectures. In particular, the O-RAN initiative is expected to bring new procedures to the Telecom Industry, introducing a decomposed and software-driven RAN architecture. For this, it is necessary to achieve: 1) Decomposition of the Centralized Unit (CU) and Distributed Unit (DU); 2) Disaggregation of RAN software from the CU and DU, such as Radio Connection Management and Mobility Management; and 3) Use of cloud paradigms to accelerate innovation and minimize time to market [3].

The telecommunications industry is interested in an open ecosystem for the RAN and approaches to creating digital transformation frameworks on the top for End-to-End (E2E) orchestration. This paper has been developed in the context of the Orchestration and Resource optimization for rEliable and lOw-latency Services (OREOS) project [4], which intends to design and implement an E2E orchestration platform for provisioning and managing critical services in the context of beyond 5G mobile communications. In this regard, Section II, presents the high-level reference architecture of the OREOS project. Section III describes the implementation of the primary interfaces and components related to the O-RAN architecture while also providing a comprehensive comparison between the Open-Source Software (OSS) solutions that can be used to build an approach for a system for the mobile RAN context, which will stimulate the establishment of multi-vendor RAN solutions, and helps in the research of RAN ecosystem interoperability. Section IV outlines the future research directions, placing the O-RAN architecture alongside the evolving cell-free architectures and explaining the importance of an intelligent controller in radio resources management. Finally, Section V, concludes the paper by identifying the limitations of the current OSS ecosystem and discussing the road ahead.

II. Open and Programmable Architectures

Next-generation open and programmable networks will use diverse components in which softwarization and virtualiza-

The authors wish to acknowledge and show their public appreciation towards the COMPETE 2020, Portugal 2020 - Operational Program for Competitiveness and Internationalization (POCI), European Union's ERDF (European Regional Development Fund) for providing the necessary co-funding of the project OREOS: Orchestration and Resource optimization for rEliable and lOw-latency Services (POCI/01/0247/FEDER/047222). Some of the authors are doctoral students funded in some cases by Capgemini Engineering and, in one case by the Portuguese National funding by FCT, Foundation for Science and Technology, through the individual research grant UI/BD/152295/2021.

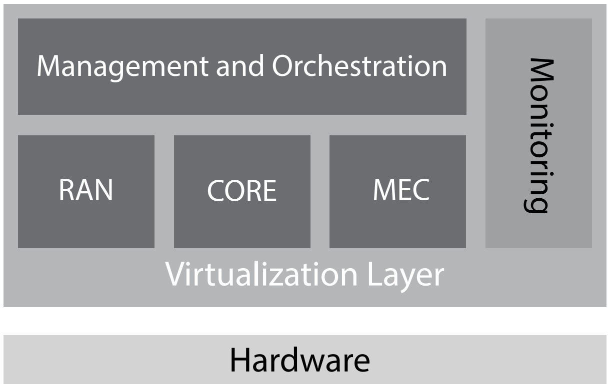

Fig. 1. OREOS Architecture

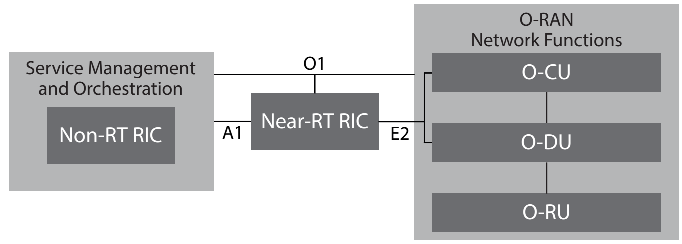

Fig. 2. O-RAN Architecture - Adapted from [9]

tion will separate services and functions from the hardware. Virtualization and management frameworks are vital in E2E management and deployment. Managing the integration of these varied sets of technologies and regulating and orchestrating network services and capabilities is challenging. Beyond 5G networks have adopted widely used and well-established cloud-computing processes and architectures to address this management issue. This section describes the OREOS and the O-RAN architectures aimed at function disassociation into different logical units that can be deployed in other locations.

A. OREOS Architecture

The industry expects the construction of an open ecosystem to create a multi-supplier RAN solution that allows for the disaggregation between hardware and software with open interfaces and virtualization, hosting software that controls and updates networks in the cloud. 5G Tango contributed to Open Source Network Function Virtualization Management and Orchestration (OSM) with an emulator for the Virtual infrastructure manager, whose support was deprecated in release 9, and started the effort to implement reference points that can be enhanced to bring network slicing capabilities to the platform. The project also suggested expanding the idea of using Operation Support Systems of the ETSI Open Source Management and Orchestration (MANO) framework for 5G into more complex frameworks [5], [6]. Also, there have been studies regarding the combination of OSM with OpenAirInterface (OAI)-Core Network to simplify the implementation of totally software-based solutions in testbeds and edge locations [7]. However, there is a lack of developments regarding integrating the RAN Intelligent Controller in E2E solutions.

The OREOS project envisions an E2E architecture based on the components depicted in Figure 1, which are: 1) Management and Orchestration; 2) RAN; 3) Core Network (CN), also known as 5G Core, and 4) Mobile Edge Computing (MEC). From these components, the RAN is critical and will be the focus of our research.

The architecture presented in Figure 1 follows the ETSI architecture [8], and the network function virtualization orchestrator is composed of 1) a virtualization layer that manages the virtualization infrastructure and the connections to the physical hardware; 2) a Management and Orchestration framework, and 3) a collection of virtual network functions that it manages. These components are equipped with southbound and northbound Application Programming Interfaces (APIs) to interact with other cellular infrastructure components, such as frameworks for governing the RAN environment and functions to bring intelligence to the network and interact with the OSS that we will explore in Section III. This architecture creates an E2E orchestration solution that explores the advantages of the O-RAN architecture and can be used in different use cases, such as Pedestrian safety within vehicle mobility; End-to-end network slicing; 3rd Generation Partnership Project (3GPP) & O-RAN alignment; 5G Self-Organizing Networks; and Intent-Based Networking. These can be merged into two major use cases: 1) Smart City and 2) Autonomous Driving. The environment in each use case runs in is of urban nature, i.e., city context, which aims to enhance the safety of people in their daily life and the quality of life with environmental monitoring. The following subsection describes the O-RAN architecture that gives the OREOS solution the distributed processing needed.

B. O-RAN Architecture

The RAN, OSS, and open interfaces are all part of the O-RAN architecture. The O-RAN architecture is designed following the 3GPP 5G New Radio (NR) RAN architecture, extending it [9]. A RAN's CU, DU, and Radio Unit (RU) are in charge of delivering various radio protocol stack functionalities, as shown in Figure 2.

The O-RAN architecture comprises the Non-Real Time RAN Intelligent Controller (Non-RT RIC), the Near-Real Time RAN Intelligent Controller (Near-RT RIC), and the Next Generation Node Bs (gNBs). The O-RAN Service Management Orchestration (SMO) structure employs the Non-RT RIC that controls decisions with a granularity greater than one second. The main objective of the Non-RT RIC in the O-RAN architecture is to allow unique RAN use cases by utilizing Artificial Intelligence/Machine Learning based algorithms. Hence, Non-RT RIC is a crucial component to allow advanced RAN technology to be applied in the O-RAN framework, staying aware of RAN properties and performance metrics to assist optimization methods.

The Near-RT RIC executes a control loop with a significantly tighter timing constraint (with a decision interval as low as 10 ms), relying on various start, stop, override, or control primitives in the `E2 Nodes` for instance radio resource management [10]. The control of the RAN Functions running in the `E2 Nodes` may be accessed by various apps (called xApps) installed on the Near-RT RIC. The xApps can be produced by third-party companies and even pulled from a shared marketplace. For example, an operator can control handover processes and optimize scheduling policies. xApps enable RAN components and their resources in near real-time to use the `E2 Interface`. The remaining components of the O-RAN architecture concern the O-RAN network functions (O-RAN Central Unit (O-CU)/O-RAN Distributed Unit (O-DU)/O-RAN Radio Unit (O-RU)) that are also called `E2 Nodes`. The O-CU entity is the O-RAN entity that operates the RAN stack's higher layer protocols, such as Radio Resource Management (RRM). O-CU needs to execute specific resource allocation algorithms and drive O-DU correspondingly. O-CU will communicate with Near-RT RIC to deliver specific Key Performance Indicator (KPI) measures while also receiving dynamic advice via the `E2 Interface`. The O-DU is the O-RAN entity that operates the RAN stack's lower layer protocols, including the Medium Access Control (MAC) layer, which is the layer that schedules Physical Resource Blocks (PRB) to User Equipments (UEs). O-DU will receive setup parameters through the `O1 Interface` and perform dynamic optimization advice from the Near-RT RIC via the `E2 Interface`.

O-RAN is currently standardizing the interfaces between each presented component. The two Radio Intelligent Controllers communicate via the `A1 Interface`, while the Non-RT RIC communicates with the O-RUs via the `O1` interface. The `A1 Interface` enables policy-based advice for use cases. The `O1 Interface` has operation and management functions and is used to obtain detailed performance data and defect information from O-RAN nodes in addition to specific configuration options. The Near-RT RIC exposes the `E2 Interface` to O-RAN Network Functions, also known as `E2 Nodes`. This interface drives the `E2 Node` design and behavior. The Near-RT RIC can then influence radio resource allocation, schedule policies, and change/create configuration policies that may impact network performance characteristics. The Near-RT RIC can set and receive specific performance data from `E2 Nodes` over the `E2 Interface`. PRB usage, average latency, and data radio bearer metrics are examples of specific KPIs that are already specified in 3GPP.

In this paper, our focus relies on Near-RT RIC and its connection via the `A1 Interface` to the Service Management Orchestration as well as its connection via the `E2 Interface` to the remaining O-RAN Network Functions - `E2 Nodes`, to create an E2E solution for next-generation open and programmable radio networks. In the next section, we explore the OSS available to implement on the OREOS architecture to fulfil this.

III. Experimentation of O-RAN OSS Solutions

Taking into account the simplified O-RAN architecture, depicted in Figure 2, we decided to focus our experiences on the connections between the Non-RT RIC and the Near-RT RIC via the `A1 Interface` and the Near-RT RIC; and the `E2 Nodes` via the `E2 Interface`. These connections were chosen because they represent the central software components of the O-RAN architecture and critical elements for managing beyond 5G network functions. The OREOS project has defined the use of Open Network Automation Platform (ONAP) [11] for the MANO component. This platform follows the ETSI NFV architecture, extending it with Service Orchestration capabilities. Moreover, it is also aligned with the software/APIs being developed by the O-RAN Software Community (O-RAN SC). ONAP in cooperation with O-RAN SC developed specific procedures and APIs to interact with the Near-RT RICs and `E2 Nodes`, via specific terminators for the `A1` and `O1` interfaces.

The capabilities of the O-RAN Near-RT RIC, `E2 Nodes` and its inner components have already been specified by the O-RAN Alliance and have an implementation by the O-RAN SC community. However, there are several OSS solutions aiming at the implementation of O-RAN Architecture components while exposing the required `A1`, `O1`, and `E2 Interfaces`. A study detailing the most relevant OSS solutions, considering the compliance with the O-RAN architecture, is herein detailed.

A. A1 Interface

Focusing on the `A1 Interface`, it is known that ONAP exposes a Southbound Interface that allows it to connect to `A1` Mediators exposed by Near-RT RIC. This interface follows the `A1AP` protocol [12], which specifies the exchanges between Non-RT RIC and Near-RT RIC, using `HTTPS` and messages in `JSON` format.

To test compliance with the O-RAN Software Community, we explored different approaches to connect Near-RT RIC solutions connection to ONAP's `A1 Terminator`. The tested Near-RT RICs were µ-Open Network Operating System (µONOS) [13], Open Radio Access Network RAN Intelligent Controller (O-RAN RIC) [14] and Flexible RAN Intelligent Controller (FlexRIC) [15] [16], as no other OSS was available. Moreover, we prioritized Near-RT RICs, whose integration with ONAP has already been adopted by the community.

The procedure for the onboarding of a new `A1 Node` is well documented [17] and is straightforward. Either the Near-RT RIC can communicate with ONAP or it fails the connection; in that case, the Healthcheck operation reports that the Near-RT RIC is not Ready for `A1` Management. Considering this, we were able to connect both O-RAN Near-RT RIC and µONOS to ONAP. Furthermore, we could have both platforms connected simultaneously to ONAP for testing the connection with the `E2 Nodes`. The FlexRIC implements the internal logic of a Near-RT RIC, focusing primarily on the `E2 Interface`. However, to the best of our knowledge, there is no OSS solution adapter to connect this platform to a Non-RT RIC via `A1` or `O1`.

B. E2 Interface

When evaluating the `E2 Interface`, it was necessary to focus on the communication between Near-RT RIC and the `E2 Nodes`. This communication leverages the E2 Application Protocol (E2AP), which has been defined by the O-RAN Alliance [18]. The protocol, described in Abstract Syntax Notation One (ASN.1), allows for signalling communication between the Near-RT RIC and the `E2 Nodes` [19]. It provides management capabilities over `E2 Nodes` and encapsulates the E2 Service Models (E2SMs). Two Services Procedures modules are part of the E2AP, the Near-RT RIC Functional Procedures, and the E2AP Global Procedures. These E2AP Global procedures enable the following services: Report, Control, Install, and Policy required by the xApps, to entail the actions related to the setup of the `E2 Nodes` into the Near-RT RIC. It is important to mention that there are mainly three versions of E2AP protocol currently in use; namely, *E2AP v1.0.0*, *E2AP v1.0.1*, and *E2AP v2.0.0*. The *E2AP v1.0.1* is backwards compatible with *E2AP v1.0.0*.

E2 Service Modules - When the setup of a `E2 Node` is successful, it is possible to access the functions in that specific node. Each `E2 Node` can implement RAN functions according to its capabilities. Therefore each RAN function has its E2SM to ensure that there is a uniform interface regarding the interaction between xApps installed in the Near-RT RIC and RAN. The Near-RT RICs make use of a E2SM or allow the creation of custom E2SM [19] depending on the `E2 Nodes` capabilities. An E2SM is described using ASN.1 interface language [20]. The current version of O-RAN Alliance *E2SM v02.01* [21], defines the following E2SMs:

- *Key Performance Measurement (KPM) v02.03*: provides RAN function handling reporting the cell-level performance measurement [22];
- *RAN Controller (RC) v01.03*: provides effective control of RAN components such as connected mode mobility [23] [24];
- *Cell Configuration and Control (CCC) v02.01*: provides control for handling and definition of the UE over `E2 Interface` [25];
- *Network Interface (NI) V01.00*: provides network interface for the RAN [26].

The version of the protocols configured between Near-RT RIC and `E2 nodes` is essential for communication between these two components. They should implement and share identical versions of E2AP and E2SM.

C. E2 Interface Implementations

This section assesses the implementation of the `E2 Interface` on µONOS, O-RAN RIC and FlexRIC. More specifically, this study focuses on E2AP and E2SM versions employed and the artifacts used during the integration of Near-RT RIC with a `E2 Node`. The most relevant information collected from our integration tests with a focus on E2AP support across the Near-RT RICs is depicted in Table I. This cross-validation is essential to ensure that the `E2 Node` can connect to the Near-RT RIC. When the registration of the `E2 Node` is possible, the Near-RT RIC can manage it using the supported E2SMs. In Table II, we present the current E2SMs supported by each Near-RT RIC. The main findings of the experimental work carried out are detailed next.

TABLE I
E2AP VERSION SUPPORT FOR DIFFERENT NEAR-RT RICS

	E2AP v01.00	E2AP v01.01	E2AP v02.00
µONOS RIC	✓	✓	✓
O-RAN Near-RT RIC	✓	✓	✓
FlexRIC	✓	-	-

1) µONOS: Regardings this OSS, *sdRAN-in-a-box* [13] was deployed, leveraging helm charts to deploy the several components into a Kubernetes (k8s) cluster. µONOS implements the `E2 Interface`, providing all the E2AP versions:

- *E2AP V01.00* until tag *v0.7.0*;
- *E2AP V01.01* from previous to *v0.8.13*;
- *E2AP V02.00* to latest (*v0.11.17*).

Concerning the E2SMs, µONOS, by the time of testing it, had implemented most of the E2SM described by O-RAN. Moreover, µONOS has created several custom E2SM, listed in Table II.

2) O-RAN RIC: Focusing on O-RAN RIC [27], different E2AP versions are used, over the different software releases:

- *E2AP V01.00* (up to release cherry);
- *E2AP V01.01* (releases D & E);
- *E2AP V02.00* (releases F & G).

The O-RAN SC has also created a simulator for a `E2 Node`, allowing the test of several xApps on the Near-RT RIC. Concerning the E2SMs, O-RAN RIC implements them, but the supported operations change over time. The deployment of O-RAN RIC is also helm chart based, allowing for the simple deployment of most components. An additional component, *xApp_Onboarder*, is required to onboard the xApps into a dedicated network namespace. An alternative is using helm chart commands directly to onboard the xApps.

3) FlexRIC: FlexRIC has been developed in collaboration with the OAI development team. It implements the `E2 Interface` both from the Near-RT RIC and `E2 Nodes` perspective. Focusing on the `E2 Nodes`, this project has adopted the decision to develop `E2 Agents` compatible with both OAI and Software Radio Systems Radio Access Network (srsRAN). The `E2 Agent` provides a generic interface to which the RAN functions can connect, extending them to provide support for `E2` over the existing capabilities. Given that the `E2 Agent` is using the E2AP protocol, it is possible to connect a `E2 Node` to several Near-RT RIC solutions if the same version of the protocol is kept among them. The FlexRIC project has adopted *E2AP v1.0.0*. The `E2 Agent` solution patches OAI and srsRAN with *E2AP v1.0.0* code binaries. As part of this work, efforts were made to upgrade the protocol version, but the migration failed due to the way `E2 Nodes`

TABLE II
NEAR-RT RIC E2SM SUPPORT

	Standard O-RAN E2SMs	Custom E2SMs
µONOS RIC	KPMv1 (partially), KPMv2, RC (pre-standard)	Mobile Handover, RAN Slicing
O-RAN Near-RT RIC	KPMv1, KPMv2, NI, RC	Supports custom E2SM
FlexRIC	KPM v2	Slice Control, Traffic Control, NGAP/GTP, PDCP, RLC, MAC

are implemented. The flexibility of the FlexRIC `E2 Agents` allows for the connection of these to O-RAN RIC. However, only *E2AP v.1.0.0* is available for this setup. This solution allows the use of O-RAN RIC specific procedures, like the support of `A1` `O1` interfaces and support for different xApps, amongst other capabilities. It is important to note that FlexRIC has developed a protocol that enhances E2AP, extending the standard capabilities. The goal behind this development was to support xApps and perform some of the procedures of the Near-RT RIC directly on the RAN Functions associated with said xApps without waiting for prior Near-RT RIC specific mediation. This approach is more flexible to changes, allowing for an easy reconfiguration of the `E2 Nodes` as the xApps can now perform four RIC procedures.

FlexRIC supports only *KPM v2* from the standard E2SM, but it has developed several custom E2SMs that gather information from the current status of the RAN functions. Moreover, it may implement a different sort of encoding technique (ASN.1, flat buffer, plain) for the E2SMs.

D. Controlling the RAN via E2

The main goal of the O-RAN Architecture (Figure 2) is ensuring the control over RAN functions programmatically, allowing for quick configuration changes when needed. Although all Near-RT RIC platforms present some form of `E2 Node` Simulator, we focus our study on NR OSS implementations of RAN Functions: OAI and srsRAN. Both have limitations that are discussed below.

1) E2 Node Simulation: Both O-RAN RIC and FlexRIC have created their `E2 Node` Simulators for testing the communication using the E2AP protocol. These `E2 Nodes` solutions allow us to see a normal `E2` Setup Request procedure and the logs into the Near-RT RIC. This enables a direct comparison of logs recorded on the Near-RT RIC platforms by the O-RAN compliant simulators with the logs of the real RAN functions, like OAI and srsRAN, which leverage OSS `E2` implementations.

2) OAI RAN: The OAI RAN project [28], during the testing and implementation phase of the work performed in this paper, did not have any information regarding `E2 Nodes`. However, at the time of writing this manuscript, a Merge Request (MR) was opened that brought `E2 Nodes` into OAI [28]. Our tests used the fork of OAI from the µONOS project [29], which implements a connection between OAI and Near-RT RIC. Therefore we could test the E2AP and E2SM protocols support. The information related to µONOS, present in Tables I and II also considers the interaction with `E2 Node`. One inconvenience of the µONOS implementation of a OAI `E2 Node` is that this software only provided `E2` integration for Long Term Evolution (LTE) binaries and not to NR. Nevertheless, an integration of OAI with µONOS was obtained, as well as the connection via `E2`. Moreover, the collection of metrics using *KPMON* xApp and the ability of slice creation via *onos.cli* was achieved. For the integration we used *E2AP V02.00 E2SM-KPM V02.00*, and the custom RAN Slicing E2SM.

3) srsRAN: srsRAN is a Fourth-Generation (4G)/5G software radio suite, just like OAI RAN. The native repository of srsRAN has no `E2 Node` implementation at the time of writing [30]. Our study of the platform found two srsRAN repositories that implement `E2` communication. These were *srsLTE* [31] and *srsRAN-E2* [32]. Upon closer inspection, we found that *srsRAN-E2* is an improvement of *srsLTE*, so we deployed and tested it.

Focusing on the characteristics of *srsRAN-E2*, it employs *E2AP V01.00* and *E2AP V01.01*. At the same time, it supports two standard E2SM: *KPM V01.00* and *NI V01.00*, and provides a custom one: *GNB-NRT V04.01*. Regarding the E2AP version selection, all the versions must be defined during the building stage. Performance and service models can be added using the ASN.1 compiler, if necessary, since modifications and adjustments may occur.

Despite both `E2 Node` and Near-RT RICs use `E2AP V01.01`, it was impossible to establish a connection, mainly because of decoding error messages. Therefore handshake was not achieved. Interestingly, all the tested Near-RT RICs reported the same error *OCTET Decode Error*, which mainly occurred on the `E2` setup request process.

E. Integration of E2AP Entities

In this section, we compare several solutions for deploying Near-RT RICs and RAN functions, therefore allowing `E2` connection. We have aimed at integrating only NR RAN functions. Our findings are depicted in Table III. It is relevant to point out that our tests focused mainly on the interoperability of the RAN functions with the Near-RT RICs. This not only considers direct communication with the `E2 nodes` but in the case of µONOS and O-RAN RIC, the connection is made using an indirect link leveraging the `E2 Agent` patches developed by FlexRIC, which are using *E2AP v1.0.0*.

In terms of controlling `E2 Nodes`, FlexRIC is the best approach to connect to a NR solution (see Table III) to later on connect `E2 Nodes` leveraging the `E2 Agents` abstraction that implements E2AP and E2SM.

TABLE III
SUMMARY ON THE INTEGRATION OF E2AP ENTITIES

Near-RT RIC\RAN Function	OAI	srsRAN
µONOS	✓(LTE only)	-
O-RAN RIC	-	-
FlexRIC	✓	✓

F. O-RAN Integration Findings

O-RAN Alliance's open-source components are incomplete and require additional integration to implement real and robust components. RAN software packages such as OAI RAN and srsRAN, as detailed in subsection III-D, are not completely developed for enabling NR networks because further coding and interoperability testing are required. In O-RAN, integration and interoperability are important issues since work and collaboration are required to establish a development environment, standard optimization metrics, and standard test and validation methods to realize the full potential of O-RAN.

As presented during this section, FlexRIC still lacks solutions that enable connection to Non-RT RIC via `A1` or `O1 Interface`, which is critical for improving network efficiency in both the short and long run. Furthermore, the FlexRIC `E2` devices only enable connections to O-RAN RIC via *E2AP v1.0.0.* Furthermore, OAI and µONOS integration only occurs via `E2` integration for LTE binaries, implying that NR networks are not supported. These constraints must be handled because organizations cannot validate the feasibility of the NR network standard that has been created.

IV. O-RAN FUTURE RESEARCH DIRECTIONS

O-RAN is a radio access network focused on building and supporting open, intelligent, and virtualized radio access networks. The O-RAN Alliance's primary research directions include the following: 1) Virtualization: which allows users to interact with the network via a web-based interface; 2) Artificial intelligence and machine learning: Which improve network performance, lower costs, and boost network efficiency; 3) Interoperability: Different vendors' equipment and solutions may smoothly coexist; and 4) Automation and orchestration: Automating radio access network administration and developing new network orchestration frameworks.

An O-RAN architecture enables independent software vendors to interact with radio resources in real-time, allowing them to influence how network resources are computed and assign resources to specific use cases, such as traffic steering, enhancing Quality of Service (QoS), and controlling Quality of Experience (QoE). The separation of RU/DU/CU allows for traffic to be directed from RUs to a pool of DUs, control traffic to be directed from a DU to a pool of CUs, and traffic to be directed from RAN to a collection of 5G core resources. For example, to forecast when it will be advantageous to offload Protocol Data Unit (PDU) sessions from the 5G radio network to another one or a Wi-Fi network, operators can feed infrastructure and application data from their access networks

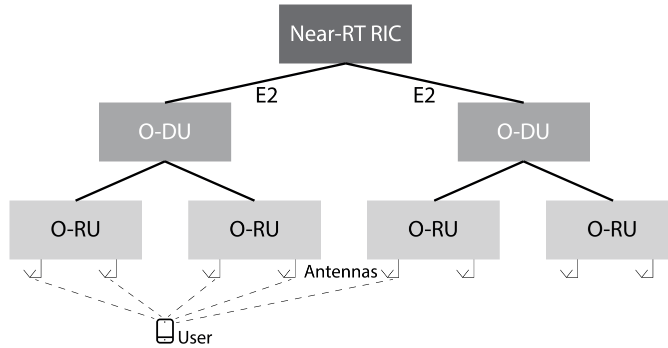

Fig. 3. CF-mMIMO based on O-RAN Architecture - Adapted from [34]

from UE environments (such as Wi-Fi, 5G private networks, and so on), and O-RAN.

New wireless network architectures are emerging that divide a base station's control plane and user plane operations, allowing them to be managed and operated by distinct entities such as emerging cell-free architectures described next.

A. Cell-free

The Cell-free massive multiple-input multiple-output (CF-mMIMO) systems demand precise granularity of processing choices (processing at Access Point (AP) versus Central Processing Unit (CPU), Inter-CPU coordination), which may be provided by hardware and software supporting the O-RAN disaggregation concept. A CF-mMIMO network has numerous dispersed APs that serve the User Equipment Units collaboratively through coherent joint transmission and reception utilizing the same time-frequency resources [33].

The O-RAN architecture is divided into three components: O-CU, O-DU, and O-RU. For centralized New Generation RAN, many O-CU and O-DU can be linked to the Near-RT RIC. The CF-mMIMO network components shown before may be connected to the O-RAN nodes. The O-DU symbolizes the CPU, whereas the O-RU represents the AP. The CF-mMIMO architecture based on O-RAN is depicted in Figure 3. In this situation, the Near-RT RIC is vital for setup issues that may impair CF-mMIMO performance, especially to allow these architectures with advanced signal processing techniques, such as massive Multiple Input and Multiple Output (MIMO) and beamforming, to improve the capacity and coverage of cell-free networks. This will be covered in the following subsection.

B. Near-RT RIC for Radio Resources

The Near-RT RIC maintains RAN components and their resources by performing optimization tasks. The Near-RT RIC has three effects on RAN performance: 1) Network Intelligence: The performance of the RAN is measured and reported. This generates data in standardized forms, which may be used to design new algorithms and regulations. 2) Resource Assurance: Ensuring that devices/users/services obtain the required performance (for example, through radio link control optimization, handover optimization, or priority). 3) Resource

Control: The RAN system must remain operational when several user groups compete for limited resources. Near-RT RIC is primarily incremental and complementary to the currently established 3GPP architecture. It can increase the performance of existing RAN nodes by bringing new capabilities; by being an open platform, it has the potential to attract software developers to the radio domain. RAN Intelligent Controller services will be used on a variety of services due to the xApps capabilities, including: 1) Context-based dynamic handover management vehicle-to-everything; 2) Dynamic radio resource allocation for traffic steering; 3) Quality-of-service/quality-of-experience optimization; 4) Massive MIMO beamforming optimization; 5) RAN sharing; 6) RAN slice service assurance; 7) Multi-vendor slice performance management; 8) Dynamic spectrum sharing; 9) Network slice subnet instance resource allocation optimization. Beyond 5G cell-free architectures, research on Near-RT RIC controllers for radio resources will cover the following areas: 1) creating efficient algorithms for radio resource allocation and management to facilitate high data rate and low latency communications; 2) Creating innovative methodologies for real-time radio resource monitoring and measurement to allow dynamic network optimization in cell-free networks; and 3) Exploring the integration of Near-RT RIC with emerging technologies such as the Internet of Things and edge computing to support new use cases and applications.

V. Conclusion

This paper examined the state of OSSs solutions to develop an orchestration solution able to deliver an E2E NR network compliant with the O-RAN. Our attention considered the `A1 Interface` connections between the service management orchestration module and the Near-RT RIC. We prioritized our focus on the interface `E2` connections between the Near-RT RIC and O-RAN Network Functions - `E2 Nodes`. Furthermore, we have explored a variety of heterogeneous solutions to connect the mentioned modules and illustrated the main practical challenges of using OSS solutions for an E2E network solution. In terms of future work, the new OAI native `E2` implementation should be tested, and its outputs and performance compared against using FlexRIC `E2 Node` encapsulation with OAI.

References

[1] M. Giordani, M. Polese, M. Mezzavilla, S. Rangan, and M. Zorzi, "Toward 6g networks: Use cases and technologies," *IEEE Communications Magazine*, vol. 58, no. 3, pp. 55–61, 2020.

[2] L. Bonati, M. Polese, S. D'Oro, S. Basagni, and T. Melodia, "Open, programmable, and virtualized 5g networks: State-of-the-art and the road ahead," *Computer Networks*, vol. 182, p. 107516, dec 2020.

[3] I. Amazon Web Services, "Open radio access network architecture on aws," 2022. Online; accessed 27 December 2022.

[4] OREOS, "Moving forward to ultra-reliable and low latency solutions," 2022. Online; accessed 20 December 2022.

[5] T. Soenen, W. Tavernier, M. Peuster, F. Vicens, G. Xilouris, S. Kolometsos, M.-A. Kourtis, and D. Colle, "Empowering network service developers: Enhanced nfv devops and programmable mano," *IEEE Communications Magazine*, vol. 57, no. 5, pp. 89–95, 2019.

[6] P. Trakadas, P. Karkazis, H. C. Leligou, T. Zahariadis, F. Vicens, A. Zurita, P. Alemany, T. Soenen, C. Parada, J. Bonnet, E. Fotopoulou, A. Zafeiropoulos, E. Kapassa, M. Touloupou, and D. Kyriazis, "Comparison of management and orchestration solutions for the 5g era," *Journal of Sensor and Actuator Networks*, vol. 9, no. 1, 2020.

[7] T. Dreibholz, "Flexible 4g/5g testbed setup for mobile edge computing using openairinterface and open source mano," in *Workshops of the international conference on advanced information networking and applications*, pp. 1143–1153, Springer, 2020.

[8] C. Rotsos, D. King, A. Farshad, J. Bird, L. Fawcett, N. Georgalas, M. Gunkel, K. Shiomoto, A. Wang, A. Mauthe, N. Race, and D. Hutchison, "Network service orchestration standardization: A technology survey," *Computer Standards & Interfaces*, vol. 54, pp. 203–215, 2017. SI: Standardization SDN&NFV.

[9] O.-R. Alliance, "O-ran alliance introduces 48 new specifications," tech. rep., Open-RAN Alliance, 2021. Online; accessed 12 January 2022.

[10] A. S. Abdalla, P. S. Upadhyaya, V. K. Shah, and V. Marojevic, "Toward next generation open radio access networks: What o-ran can and cannot do!," *IEEE Network*, vol. 36, no. 6, pp. 206–213, 2022.

[11] L. Foundation, "Open network automation platform." https://www.onap.org. Online; accessed 18 January 2023.

[12] O. Alliance, "O-ran a1 interface: Application protocol 3.01 (oran. wg2. a1ap-v03. 01)," *Technical Specification*, 2021.

[13] O. Project, "Micro-onos code repository." https://github.com/onosproject/sdran-in-a-box. Online; accessed 18 January 2023.

[14] O.-R. SC, "O-ran ric documentation." https://docs.o-ran-sc.org/en/latest/projects.html#near-realtime-ran-intelligent-controller-ric. Online; accessed 18 January 2023.

[15] T. O. S. A. (OSA), "Flexric code repository." https://gitlab.eurecom.fr/mosaic5g/flexric. Online; accessed 18 January 2023.

[16] R. Schmidt, M. Irazabal, and N. Nikaein, "Flexric: An sdk for next-generation sd-rans," in *Proceedings of the 17th International Conference on Emerging Networking EXperiments and Technologies*, CoNEXT '21, (New York, NY, USA), p. 411–425, Association for Computing Machinery, 2021.

[17] O. SC, "A1 policy management service." https://docs.onap.org/projects/onap-ccsdk-oran/en/latest/offeredapis/pms-api.html. Online; accessed 18 January 2023.

[18] O. Alliance, "O-ran working group 3: Near-real-time ran intelligent controller-e2 application protocol (e2ap)," *ORAN-WG3. E2AP-KPM-v01. 00*, 2020.

[19] M. Polese, L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, "Understanding o-ran: Architecture, interfaces, algorithms, security, and research challenges," *arXiv preprint arXiv:2202.01032*, 2022.

[20] G. Neufeld and S. Vuong, "An overview of asn. 1," *Computer Networks and ISDN Systems*, vol. 23, no. 5, pp. 393–415, 1992.

[21] O. Alliance, "O-ran near-real-time ran intelligent controller e2 service model (e2sm) 2.01 (o-ran.wg3.e2sm-v02.01)," *Technical Specification*, March 2022.

[22] O. Alliance, "O-ran e2 service model (e2sm), o-ran.wg3.e2sm-kpm-v02.03," *Technical Specification*, October 2022.

[23] E. Gures, I. Shayea, A. Alhammadi, M. Ergen, and H. Mohamad, "A comprehensive survey on mobility management in 5g heterogeneous networks: Architectures, challenges and solutions," *IEEE Access*, vol. 8, pp. 195883–195913, 2020.

[24] O. Alliance, "O-ran e2 service model (e2sm), ran control 1.03o-ran.wg3.e2sm-rc-v01.03," *Technical Specification*, October 2022.

[25] O. Alliance, "O-ran e2 service model (e2sm), cell configuration and control 1.0 (o-ran.wg3.e2sm-ccc-v01.00)," *Technical Specification*, October 2022.

[26] O. Alliance, "O-ran near-real-time ran intelligent controller e2 service model (e2sm), ran function network interface (ni) 1.0," *Technical Specification*, February 2020.

[27] O.-R. SC, "O-ran ric repository." https://gerrit.o-ran-sc.org/r/ric-plt/ric-dep. Online; accessed 18 January 2023.

[28] T. O. A. Alliance, "Oai repository." https://gitlab.eurecom.fr/oai/openairinterface5g. Online; accessed 18 January 2023.

[29] T. O. A. A. O. Project, "Onos fork of oai." https://github.com/onosproject/openairinterface5g. Online; accessed 18 January 2023.

[30] S. R. Systems, "srsran is a 4g/5g software radio suite developed by srs." https://github.com/srsran/srsran. Online; accessed 18 January 2023.

[31] D. Johnson, D. Maas, and J. Van Der Merwe, "Open source ran slicing on powder: A top-to-bottom o-ran use case," in *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '21, (New York, NY, USA), p. 507–508, Association for Computing Machinery, 2021.

[32] e. a. Software Radio Systems, OpenAICellular, "Open ai cellular srsran-e2 repository." https://github.com/openaicellular/srsRAN-e2. Online; accessed 18 January 2023.

[33] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-free massive mimo versus small cells," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1834–1850, 2017.

[34] V. Ranjbar, A. Girycki, M. A. Rahman, S. Pollin, M. Moonen, and E. Vinogradov, "Cell-free mmimo support in the o-ran architecture: A phy layer perspective for 5g and beyond networks," *IEEE Communications Standards Magazine*, vol. 6, no. 1, pp. 28–34, 2022.