O-RAN Testbeds: From Experimental Deployments to Simulation Frameworks

Juan Navarro
Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay
juan.navarro@fing.edu.uy

Alberto Castro
Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay
acastro@fing.edu.uy

Claudina Rattaro
Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay
crattaro@fing.edu.uy

***Abstract*—Open Radio Access Network (O-RAN) has emerged as a disruptive paradigm for mobile networks, aiming to disaggregate traditional RAN functions into software-based components connected via open, interoperable interfaces. By enabling multi-vendor integration, programmability through rApps and xApps, and cloud-native deployments, O-RAN promises to foster innovation, cost efficiency, and operational flexibility. However, validating its feasibility and performance requires realistic experimentation and the use of testbeds. This paper provides a review and practical evaluation of O-RAN testbeds, ranging from experimental deployments to fully simulated frameworks. We first introduce the architectural principles of O-RAN, highlighting its key components, functional splits, and open interfaces. Next, we survey relevant research testbeds and prototypes reported in the literature, analyzing their architectures, design choices, and lessons learned. Finally, we reproduce and assess four representative implementations, including the NIST automation framework and ns-3–based O-RAN simulators. A comparative analysis is presented, discussing advantages, limitations, and alignment with evolving O-RAN standards. Our findings indicate that while automation frameworks simplify reproducibility and simulation frameworks enable flexible experimentation, interoperability issues and the lack of maturity in the Non-Real-Time RAN Intelligent Controller (Non-RT RIC) remain open challenges. These insights can guide researchers and operators in designing future O-RAN testbeds and accelerating the adoption of open and smart RAN solutions.**

***Index Terms*—Open RAN, O-RAN Alliance, Testbeds, Radio Access Networks, RIC, Network Simulation**

I. Introduction

Historically, mobile network operators were tied to a small group of vendors providing proprietary hardware and software, often delivered as “black boxes”. This vendor lock-in resulted in limited interoperability, high costs, and slow innovation. While the creation of 3GPP (3rd Generation Partnership Project) was a milestone toward standardization, internal interfaces and processes remained largely closed. The advent of virtualization brought further openness in the mobile core, but the Radio Access Network (RAN) remained heavily proprietary.

To overcome these challenges, major operators formed initiatives such as Cloud RAN (C-RAN) in China and Cross RAN (xRAN) in Europe, the United States, and Japan. In 2018, these efforts converged into the Open RAN (O-RAN) Alliance, founded by AT&T, China Mobile, NTT Docomo, Orange, and T-Mobile [1]–[3]. Today, the Alliance counts more than 300 members, including global operators, technology companies (e.g., Intel, NVIDIA, Microsoft, Google), research institutions, and even traditional RAN vendors. Alongside the Alliance, the O-RAN Software Community (OSC), hosted by the Linux Foundation, plays a key role in developing open-source implementations aligned with O-RAN specifications.

O-RAN’s mission is to disaggregate and virtualize RAN components, connect them via open, well-defined interfaces, and enable interoperability across multi-vendor environments. In addition, O-RAN introduces intelligent controllers, namely the Non-Real-Time RAN Intelligent Controller (Non-RT RIC) and the Near-Real-Time RAN Intelligent Controller (Near-RT RIC), to bring automation, optimization, and AI-driven decision-making into the RAN.

In this paper, we review and compare existing O-RAN testbeds, ranging from experimental deployments to simulation-based frameworks. We classify and analyze representative implementations (srsRAN, Openairinterface (OAI), OSC RIC, FlexRIC, ns-3 extensions, among others), highlighting their advantages, limitations, and alignment with O-RAN specifications. Additionally, we reproduce and adapt selected implementations to evaluate their feasibility, interoperability issues, and practical constraints. Our contribution is twofold: (i) to consolidate the current state of O-RAN testbed research, and (ii) to offer practical insights into reproducibility and deployment trade-offs that can guide future academic and industrial efforts.

The rest of this article is structured as follows. Section II introduces the O-RAN architecture and its key components. Section III reviews related work on O-RAN testbeds, spanning real deployments, Unmanned Aerial Vehicle (UAV) experimentation, enterprise-scale prototypes, and simulation frameworks. Section IV describes our reproduced implementations, analyzing feasibility, interoperability, and limitations. Finally, Section V summarizes the findings and outlines directions for future work.

II. O-RAN Architecture

The O-RAN architecture aims to disaggregate the traditional RAN into interoperable, software-based components connected through open interfaces [4]–[9]. Figure 1 summarizes the overall structure, showing network functions, temporal control loops (left), and deployment domains (right). The main components of its architecture are briefly described below.

A. Service Management and Orchestration (SMO) and Non-RT RIC

At the top of the architecture, the SMO provides end-to-end network management, exposing FCAPS (fault, configuration, accounting, performance, and security) capabilities and managing the distributed O-Cloud. Within the SMO resides the Non-RT RIC, which hosts rApps (non-RT RIC application) and implements the A1 interface to influence near-real-time functions. Operating at control-loop latencies of one second or more, rApps enable policy-based management, large-scale optimization, and AI/ML-driven predictions.

B. Near-RT RIC

The Near-RT RIC operates with latencies of 10 ms to 1 s, enabling closed-loop optimization of radio resources via the E2 interface. It hosts modular xApps (near-RT RIC application) that can monitor KPIs, enforce policies, and dynamically reconfigure RAN behavior. Core platform functions, including conflict mitigation, shared data layers, and subscription management, ensure the safe and coordinated execution of multiple xApps.

C. Open gNodeB and Open eNodeB

O-RAN generalizes the 3GPP base stations into Open gNodeB (O-gNB) for 5G and Open eNodeB (O-eNB) for LTE. The O-gNB is decomposed into: (i) Open Central Unit (O-CU): handles higher-layer functions such as PDCP and SDAP, including mobility and session management; runs in a centralized (often cloud) location; (ii) Open Distributed Unit (O-DU): handles lower-layer RLC, MAC, and parts of PHY, managing real-time scheduling and radio resource control; typically close to the cell site; and (iii) Open Radio Unit (O-RU): contains the RF front-end and lower PHY, performs signal transmission/reception over the air interface; connects to the DU via the Open Fronthaul (O-FH), e.g., eCPRI. The preferred functional split is 7.2x, with lower-PHY tasks performed at the O-RU and higher-PHY tasks at the O-DU, balancing transport costs, bandwidth, and latency. Both O-gNB and O-eNB connect to the Near-RT RIC via E2, to the SMO via O1, and internally through 3GPP-defined interfaces (E1, F1, Xn, X2).

D. Open Cloud

The Open Cloud (O-Cloud) provides a distributed virtualization substrate that hosts O-RAN network functions. The SMO manages it via the O2 interface, covering infrastructure management and deployment lifecycle operations. Synchronization across O-Cloud sites is crucial for coordinated RAN operations.

E. Interfaces

O-RAN introduces several new logical, open interfaces in addition to 3GPP-defined ones:

- A1: connects the Non-RT RIC and Near-RT RIC for policy guidance, enrichment information, and AI/ML model distribution.
- E2: links the Near-RT RIC with E2 nodes (O-CU, O-DU, O-RU), enabling monitoring and near-real-time control through E2AP and service models (e.g., KPM, RC).
- O1: connects SMO to all managed elements (RICs and E2 nodes) for configuration, fault, and performance management, typically using NETCONF, HTTPS, and SFTP.
- O2: connects SMO to the O-Cloud, supporting infrastructure and deployment management services.
- O-FH (CUS/M): fronthaul between O-DU and O-RU, aligned with split 7.2x, ensuring transport of user, control, synchronization, and management planes.

F. Programmability via rApps and xApps

One of O-RAN's most disruptive innovations is the programmability introduced by rApps (non-real-time) and xApps (near-real-time). These applications enable operators and third-party developers to implement customized behaviors and functionalities. For instance, during a large-scale event (e.g., a stadium concert), rApps can analyze historical traffic data and define policies. At the same time, xApps can dynamically redistribute load and adjust QoS prioritization in near real time.

In summary, O-RAN architecture introduces disaggregation, cloudification, and programmability into the RAN, with the SMO and RICs enabling cross-layer intelligence and open interfaces, ensuring multi-vendor interoperability. Given the broader ecosystem of actors beyond traditional vendors, it becomes crucial to establish testbeds and validation frameworks for xApps and rApps. Therefore, the following section reviews existing initiatives and research contributions in this area.

III. Related Work on O-RAN Testbeds

Several works have explored O-RAN implementations in experimental environments, ranging from real deployments to academic and enterprise-scale testbeds.

The study *Challenges and Lessons Learned During Private 5G Open RAN Deployments* [10] analyzes multiple O-RAN trials conducted in 2023. While not a testbed itself, it highlights recurring challenges, including phased development, hardware/software integration issues, a shortage of skilled workers, vendors' reluctance to conduct interoperability tests, and difficulties with manual configuration of 5G SA. The authors emphasize the need for automation, extensive interoperability testing, and agile methodologies (CI/CD) to ensure long-term manageability.

In *Prototyping O-RAN Enabled UAV Experimentation for the AERPAW Testbed* [11], O-RAN is applied to (unmanned aerial vehicles) UAV experimentation within the AERPAW platform. Using Open5GS, OAI's gNB and UE, and FlexRIC for the E2 interface, the authors implement a Near-RT RIC that runs xApps to monitor network dynamics, demonstrating the

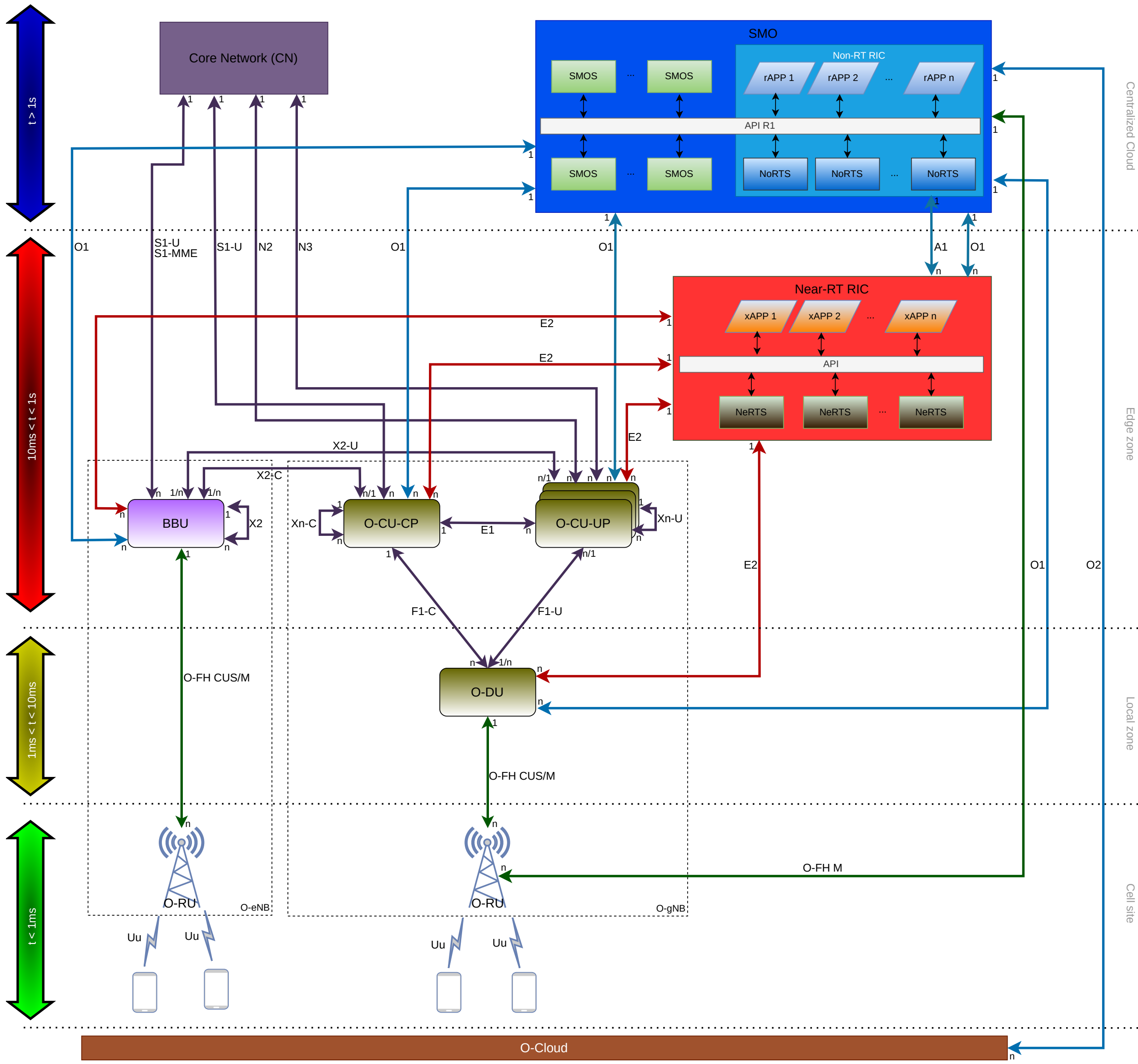

Fig. 1: O-RAN architecture illustrating the interaction among key components—Core, SMO, Non-RT RIC, Near-RT RIC, O-eNB, O-gNB, and O-Cloud—through the main interfaces (A1, O1, O2, E2, and O-FH). The figure highlights the hierarchical control structure, where the Non-RT RIC (within the SMO) manages policy and AI-driven optimization over the A1 interface, while the Near-RT RIC performs time-sensitive control via the E2 interface. Figure adapted from [6].

feasibility of combining O-RAN with aerial experimentation on real and simulated infrastructures.

The work *An Open, Programmable, Multi-vendor 5G O-RAN Testbed with NVIDIA ARC and OpenAirInterface* [12] introduces X5G, a disaggregated 5G private network testbed. Although no RIC is implemented, the testbed leverages O-RAN split 7.2x and separates DU-High and DU-Low functions across heterogeneous servers using FAPI. The setup combines OAI implementations with the NVIDIA Aerial SDK, GPUs, and commercial RU hardware, demonstrating the integration of open and proprietary components.

Enterprise-scale experimentation is presented in *Accelerating Open RAN Research Through an Enterprise-scale 5G Testbed* [13]. The platform integrates HPE Telco servers with hardware accelerators, Foxconn radios, and Kubernetes-based orchestration. Despite its realism and scalability, the integration of RIC remains unaddressed.

Other works focus on enhancing O-RAN capabilities. *TinyRIC* [14] proposes a lightweight RT-RIC for microsecond-level control, introducing tApps for resource allocation, although implementation details remain limited. *OAIC* [15] develops a complete open-source testbed that integrates Near-

RT and Non-RT RICs with srsRAN and OSC software, providing automated validation of xApps and rApps. Similarly, *Prototyping Next Generation O-RAN Research Testbeds with SDRs* [16] demonstrates interoperability between srsRAN and a Near-RT RIC, implementing KPIMON and slicing xApps on COTS hardware.

Some proposals address specific challenges. *CloudRIC* [17] introduces resource sharing across multiple O-DUs, reducing costs and energy consumption by virtualizing accelerators. Meanwhile, the NIST blueprint [18] provides a comprehensive guide for building replicable O-RAN testbeds using open-source stacks (srsRAN, Open5GS, FlexRIC, OSC RIC), which are validated through SDR-based experiments.

Finally, simulation-based efforts extend O-RAN research. *O-RAN with Machine Learning in ns-3* [19] implements a logical Near-RT RIC for LTE handover optimization using ML. In contrast, *Programmable and Customized Intelligence for Traffic Steering* [20] integrates ns-3 simulations with a physical Near-RT RIC deployment, enabling experimentation with traffic steering xApps.

In summary, O-RAN testbed research spans real deployments, UAV platforms, enterprise-scale prototypes, and simulation frameworks. While many initiatives validate interoperability and xApp/rApp experimentation, challenges remain in integrating RICs, automating configuration, and achieving realistic end-to-end performance.

IV. Implementations

After reviewing the theoretical aspects of O-RAN and related testbeds, we reproduced and adapted selected implementations to evaluate their feasibility and limitations. For clarity, each case is numbered as *Implementation 1*, *Implementation 2*, and so on.

A. Implementation 1: NIST Automation Framework (srsRAN combined with OSC RIC)

The first implementation builds on the NIST *O-RAN-Testbed-Automation* tool [18], which automates the deployment of a nearly complete O-RAN testbed on a single node. This setup integrates Open5GS as the 5G Core, srsRAN for gNB and UE, and the OSC Near-RT RIC deployed as Kubernetes microservices. The RIC includes several key modules: `e2mgr`, which manages the E2 connections with RAN nodes; `e2term`, the E2 interface termination point; `appmgr`, which manages the lifecycle of xApps; Redis, a database for storing configuration and state information; and Prometheus, a monitoring system that collects and stores internal metrics from the Near-RT RIC. Communication between the gNB and RIC takes place over the E2 interface, while the radio channel is emulated using ZeroMQ. The overall architecture is depicted in Figure 2.

Key Insights: The setup was straightforward and reproducible, benefiting from mature open-source components. However, interoperability issues were encountered during the deployment of certain xApps (e.g., KPIMON) due to version

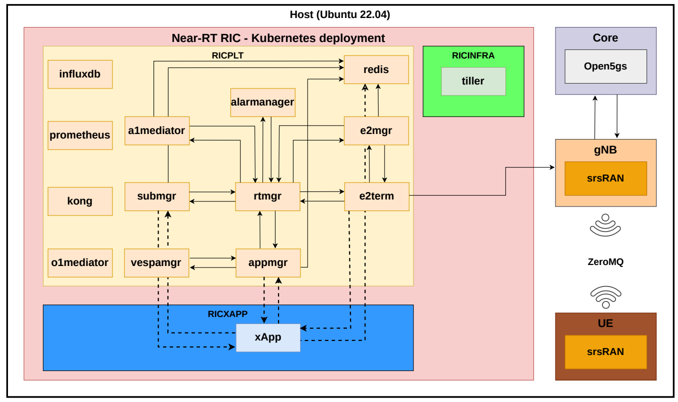

Fig. 2: Implementation 1. On the host, the OSC Near-RT RIC is deployed as a Kubernetes infrastructure comprising several pods across three namespaces. In addition, the srsRAN gNB and UE are installed directly on the host, together with the Open5GS core.

mismatches between srsRAN and the OSC RIC, highlighting ongoing synchronization challenges across independently evolving projects.

B. Implementation 1.1: Non-RT RIC Extension

As an extension, we evaluated the optional deployment of the OSC Non-RT RIC (Release L). Although the framework includes policy management, rApp handling, and information coordination, in practice, several pods failed to run correctly. They interfered with the Near-RT RIC, confirming that Non-RT RIC development is still less mature than Near-RT RIC solutions.

C. Implementation 2: NIST Automation with OAI + FlexRIC

The second configuration offered by NIST integrates Open5GS, OAI gNB/UE with E2 support, and the FlexRIC controller [21]. Refer to Figure 3 for additional details. Unlike the OSC RIC, FlexRIC is not directly an O-RAN Near-RT RIC but rather a lightweight Software Development Kit (SDK) for building specialized, service-oriented SD-RAN (Software-Defined Radio Access Network) controllers. One such controller could serve as the Near-RT RIC. FlexRIC adopts a modular, flexible architecture based on two core components (a server and an agent), augmented with internal applications (iApps) and service models that enable the programming of custom RAN functions. In this implementation, air channels are simulated using OAI's RFSimulator.

Key Insights: This solution is highly customizable and lighter than the OSC RIC, meeting the timing requirements of low-latency applications. Nonetheless, it diverges from the official O-RAN specification, raising questions about long-term alignment with community standards.

D. Implementation 3: ns-3 with Real RIC Integration

The third implementation is based on ns-O-RAN [20], which connects ns-3 simulations to a real Near-RT RIC. A fork of the mmwave module (in particular *ns-o-ran-ns3-mmwave*

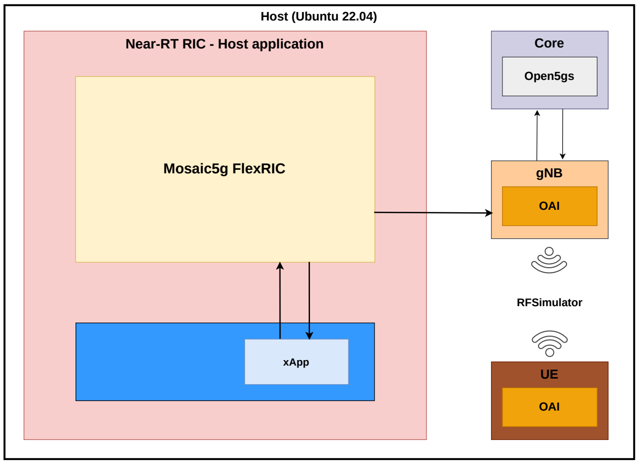

Fig. 3: Implementation 2. On the host, the OAI FlexRIC Near-RT RIC is installed. Additionally, the OAI gNB and UE are installed directly on the host, along with the Open5GS core.

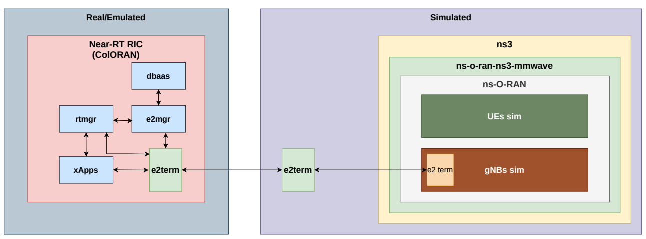

Fig. 4: Implementation 3. In the real or emulated world, the Near-RT EIC is deployed, while the rest of the RAN is simulated within ns-3.

[22]) enables 5G simulation with E2 support. The *ns-O-RAN* module is also installed, which extends *ns-o-ran-ns3-mmwave* to enable the termination of multiple E2 interfaces in the simulation elements. See Figure 4. It is also necessary to install the *o-ran-e2sim* (https://github.com/wineslab/o-ran-e2sim) application, a fork of the OSC *sim-e2-interface* (https://github.com/o-ran-sc/sim-e2-interface) project. Its role is to provide E2 termination for the simulation and to serve as a direct connection point to the Near-RT RIC. Finally, the Colosseum Near-RT RIC (ColORAN) [23] provides a reduced version of the OSC RIC. It allows simulated UEs and gNBs in ns-3 to interoperate with a physical RIC.

Key Insights: The approach expands the range of experimentation by combining simulated RAN scenarios with live RIC control. However, it depends on outdated RIC and ns-3 versions, making it rigid and less adaptable.

E. Implementation 4: Fully Simulated ns-3 O-RAN

Finally, the NIST ns-3 O-RAN module [19] enables fully simulated O-RAN experiments. It models all key components, including a logical Near-RT RIC with xApp (logic modules), data repository, conflict mitigation, and E2 termination. Nodes implement reporters and triggers to feed metrics to the RIC.

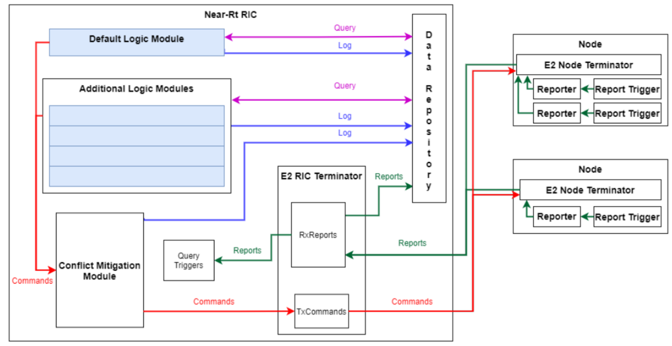

Fig. 5: Implementation 4. Near-RT RIC with its modules and their internal communication, as well as the E2 nodes and their connection to the Near-RT RIC.

The framework also supports ML inference through ONNX or PyTorch integration. The diagram illustrating this implementation is shown in Figure 5.

Key Insights: This solution offers a flexible, fully simulated environment suitable for algorithmic development and xApp validation. Its main limitation is the steep learning curve of ns-3 and C++ for new users. Additionally, a major constraint is that it currently operates with the E2 eNB node for LTE; therefore, integration with 5G simulators, such as 5G-LENA, is required to enable more up-to-date next-generation scenarios [24].

F. Summary

Overall, the reviewed implementations cover a spectrum from automated single-node deployments to hybrid simulation–real integration and fully simulated environments. While automation frameworks (such as NIST) ease reproducibility, interoperability issues remain common. FlexRIC offers performance advantages but diverges from OSC specifications. Simulation-based solutions offer flexibility for algorithm testing, but are limited by the maturity and complexity of the tools used. These insights inform the design of future O-RAN testbeds, particularly in terms of automation, interoperability, and alignment with evolving standards.

To better compare the reviewed initiatives, Table I summarizes the main characteristics of the selected O-RAN testbed implementations. The table highlights the technology stack, the adopted RIC solution, the type of radio or channel emulation, and their main advantages and limitations. This comparative perspective offers a clear overview of how different approaches balance realism, flexibility, and compliance with O-RAN specifications, providing insights into the trade-offs researchers and practitioners face.

V. Conclusion

This paper has presented a review and hands-on evaluation of O-RAN testbeds, bridging the gap between theoretical specifications and practical implementations. We began by outlining the O-RAN architecture, emphasizing its disaggregated design, cloud-native foundations, and the key role of open

TABLE I: Comparison of O-RAN Testbed Implementations.

Implementation	Stack (Core / RAN)	RIC	Radio / Channel	Advantages	Limitations
1 (NIST)	Open5GS / srsRAN	OSC Near-RT RIC (Kubernetes microservices)	ZeroMQ (simulated channel)	Easy deployment; uses widely adopted open-source tools; official OSC RIC	Interoperability issues with xApps; version mismatches; lacks Non-RT RIC maturity
1.1 (NIST ext.)	Same as Impl. 1	OSC Non-RT RIC (Release L)	ZeroMQ	Includes rApp support; introduces A1 Policy Management	Several pods fail; conflicts with Near-RT RIC; not production ready
2 (NIST–OAI)	Open5GS / OAI gNB & UE	FlexRIC (lightweight SDK)	RFSimulator (OAI)	Lightweight, customizable; supports low-latency apps; good OAI integration	Not official OSC RIC; diverges from standard; learning curve for FlexRIC
3 (ns-O-RAN)	ns-3 simulation / modified mmWave module	ColORAN (reduced OSC RIC)	Simulated in ns-3	Mixes real RIC with simulated RAN; enables flexible experiments	Outdated ns-3 and RIC versions; rigid, less adaptable
4 (ns3-ORAN)	Fully simulated ns-3	Logical RIC (modules for xApps, data, conflict mitigation)	Fully simulated in ns-3	Fully simulated, highly flexible; supports ML integration via ONNX/PyTorch	Requires ns-3 & C++ expertise; steep learning curve; simulation only. LTE only.

interfaces, including A1, E2, O1, and O2. We then surveyed existing experimental efforts reported in the literature, spanning private 5G deployments, enterprise-scale prototypes, and simulation environments.

Building on this background, we implemented and analyzed four representative O-RAN configurations: (i) the NIST automation framework with srsRAN and OSC RIC, (ii) an OAI-based stack with FlexRIC, (iii) ns-3 with real RIC integration, and (iv) a fully simulated ns-3 O-RAN environment. The comparative study highlights the advantages and limitations of these frameworks: automation frameworks enable reproducibility but face interoperability issues; FlexRIC offers lightweight customization but diverges from official standards; hybrid simulation–real setups expand research flexibility but rely on outdated components; and fully simulated frameworks support rapid prototyping but require substantial expertise in ns-3.

From these insights, two key challenges emerge: ensuring interoperability across rapidly evolving open-source components and strengthening the maturity of Non-RT RIC functionalities. Addressing these issues will be critical for building scalable and sustainable O-RAN deployments.

Future work will focus on integrating AI/ML-driven rApps and xApps, automating end-to-end orchestration, and exploring hardware–software co-design for energy-efficient O-RAN implementations. In addition, we plan to evaluate more realistic prototypes by incorporating commercial off-the-shelf (COTS) UEs, adding radio-frequency interfaces via Software-Defined Radio (SDR) platforms, and validating performance under practical deployment conditions. Another of our future research lines focuses on adapting existing ns-3 developments to support 5G and RIC integration. Such directions will help transform O-RAN from experimental prototypes into production-ready, intelligent, and sustainable networks.

References

[1] O.-R. Alliance, "Governance of o-ran alliance e.v. in compliance with wto principles," O-RAN Alliance, Tech. Rep., 2023.

[2] O-RAN Alliance e.V., "O-ran alliance overview," Jul. 2025, white paper (23 pages), available at https://mediastorage.o-ran.org/white-papers/O-RAN.Overview-of-the-O-RAN-ALLIANCE-presentation.pdf.

[3] O.-R. Alliance. O-ran alliance website. Available: https://www.o-ran.org/.

[4] ——. Specifications of the o-ran alliance working groups. [Online]. Available: https://specifications.o-ran.org/specifications

[5] W. Rouwet, *Open Radio Access Network (O-RAN) Systems Architecture and Design*, 1st ed. Londres, UK: Academic Press, Elsevier, 2022.

[6] I. C. Wong, A. Chopra, S. Rajagopal, and R. Jana, *Open RAN The Definitive Guide*, 1st ed. New Jersey, USA: Wiley, 2024.

[7] M. Polese, L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, "Understanding o-ran: Architecture, interfaces, algorithms, security, and research challenges," *IEEE Comumunications Surveys & Tutorials*, vol. 25, no. 2, pp. 1376–1411, 2023.

[8] A. L.-G. Simona Marinova, "Intelligent o-ran beyond 5g: Architecture, use cases, challenges, and opportunities," *IEEE Access*, vol. 12, pp. 27 088–27 114, 2024.

[9] Y. Huang, Q. Sun, N. Li Z, J. Huang, H. Ding, and C.L., "Validation of current o-ran technologies and insights on the future evolution," *IEEE*, vol. 42, no. 2, pp. 487–505, 2024. [Online]. Available: https://doi.org/10.3390/s24103242

[10] X. Krasniqi, E. Hajrizi, and B. Qehaja, "Challenges and lessons learned during private 5g open ran deployments," in *2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, 2023, pp. 1–6.

[11] J. Moore, A. S. Abdalla, C. Ueltschey, and V. Marojevic, "Prototyping o-ran enabled uav experimentation for the aerpaw testbed," 2024. [Online]. Available: https://arxiv.org/abs/2411.04027

[12] D. Villa, I. Khan, F. Kaltenberger, N. Hedberg, R. S. Da Silva, A. Kelkar, C. Dick, S. Basagni, J. M. Jornet, T. Melodia, M. Polese, and D. Koutsonikolas, "An open, programmable, multi-vendor 5g o-ran testbed with nvidia arc and openairinterface," in *IEEE INFOCOM 2024 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2024, pp. 1–6.

[13] P. Bahl, M. Balkwill, X. Foukas, A. Kalia, D. Kim, M. Kotaru, Z. Lai, S. Mehrotra, B. Radunovic, S. Saroiu, C. Settle, A. Verma, A. Wolman, F. Y. Yan, and Y. Zhang, "Accelerating open ran research through an enterprise-scale 5g testbed," in *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, ser. ACM MobiCom '23. New York, NY, USA: Association for Computing Machinery, 2023. [Online]. Available: https://doi.org/10.1145/3570361.3615745

[14] C. Liu, G. Aravinthan, A. Kak, and N. Choi, "Tinyric: Supercharging o-ran base stations with real-time control," in *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, ser. ACM MobiCom '23. New York, NY, USA: Association for Computing Machinery, 2023. [Online]. Available: https://doi.org/10.1145/3570361.3615743

[15] P. S. Upadhyaya, N. Tripathi, J. Gaeddert, and J. H. Reed, "Open ai cellular (oaic): An open source 5g o-ran testbed for design and testing of ai-based ran management algorithms," *IEEE Network*, vol. 37, no. 5, pp. 7–15, 2023.

[16] P. S. Upadhyaya, A. S. Abdalla, V. Marojevic, J. H. Reed, and V. K. Shah, "Prototyping next-generation o-ran research testbeds with sdrs," 2022. [Online]. Available: https://arxiv.org/abs/2205.13178

[17] L. L. Schiavo, G. Garcia-Aviles, A. Garcia-Saavedra, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez, "Cloudric: Open radio access network (o-ran) virtualization with shared heterogeneous computing," in *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, ser. ACM MobiCom '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 558–572. [Online]. Available: https://doi.org/10.1145/3636534.3649381

[18] P. Liu, K. Lee, F. Cintron, S. Wuthier, B. Savaliya, D. Montgomery, and R. Rouil, "Blueprint for deploying 5g o-ran testbeds: A guide to using diverse o-ran software stacks," 2024-10-23 04:10:00 2024. [Online]. Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=958753

[19] W. Garey, R. A. Rouil, E. Black, T. Ropitault, and W. Gao, "O-ran with machine learning in ns-3," 2023-06-28 04:06:00 2023.

[20] A. Lacava, M. Polese, R. Sivaraj, R. Soundrarajan, B. S. Bhati, T. Singh, T. Zugno, F. Cuomo, and T. Melodia, "Programmable and customized intelligence for traffic steering in 5g networks using open ran architectures," *IEEE Transactions on Mobile Computing*, pp. 1–16, 2023.

[21] 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*, ser. CoNEXT '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 411–425. [Online]. Available: https://doi.org/10.1145/3485983.3494870

[22] M. Mezzavilla, M. Zhang, M. Polese, R. Ford, S. Dutta, S. Rangan, and M. Zorzi, "End-to-end simulation of 5g mmwave networks," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2237–2263, 2018.

[23] 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 Transactions on Mobile Computing*, pp. 1–14, July 2022.

[24] N. Patriciello, S. Lagen, B. Bojovic, and L. Giupponi, "An e2e simulator for 5g nr networks," *Simulation Modelling Practice and Theory*, vol. 96, p. 101933, Nov. 2019.