RRAIL Project

RAN RIC and Applications Interoperability Lab (RRAIL)

Authors:

Sarat Puthenpura, ONF

Timon Sloane, ONF

Alex Stancu, ONF

Table of Contents

Introduction

Open RAN is fundamentally about disaggregating all the components of the RAN that historically have only been available as closed, proprietary, and tightly integrated solutions coming from a small number of vendors. Operators want to be able to select and assemble best-of-breed components from trusted vendors for various reasons including supply chain security and diversity, price, functionality and fostering innovation. But assembling complex solutions integrating components from multiple vendors is a difficult task. Complicating the challenge is the need for software that controls end-to-end policy and optimization across a large network footprint. This level of coordination is where current vertically integrated solutions have a significant advantage over open solutions, and it is in this area that investment is essential if open RAN solutions are to become competitive against offerings from today's incumbent vendors.

In order to assemble new open, intelligent, multi-vendor solutions, the O-RAN standards have defined the SMO, RIC and x/rApps to ultimately disaggregate intelligent control (in the form of the x/rApps) from the radio components (RU, DU, CU). **Figure 1** is a schematic diagram of the O-RAN architecture.

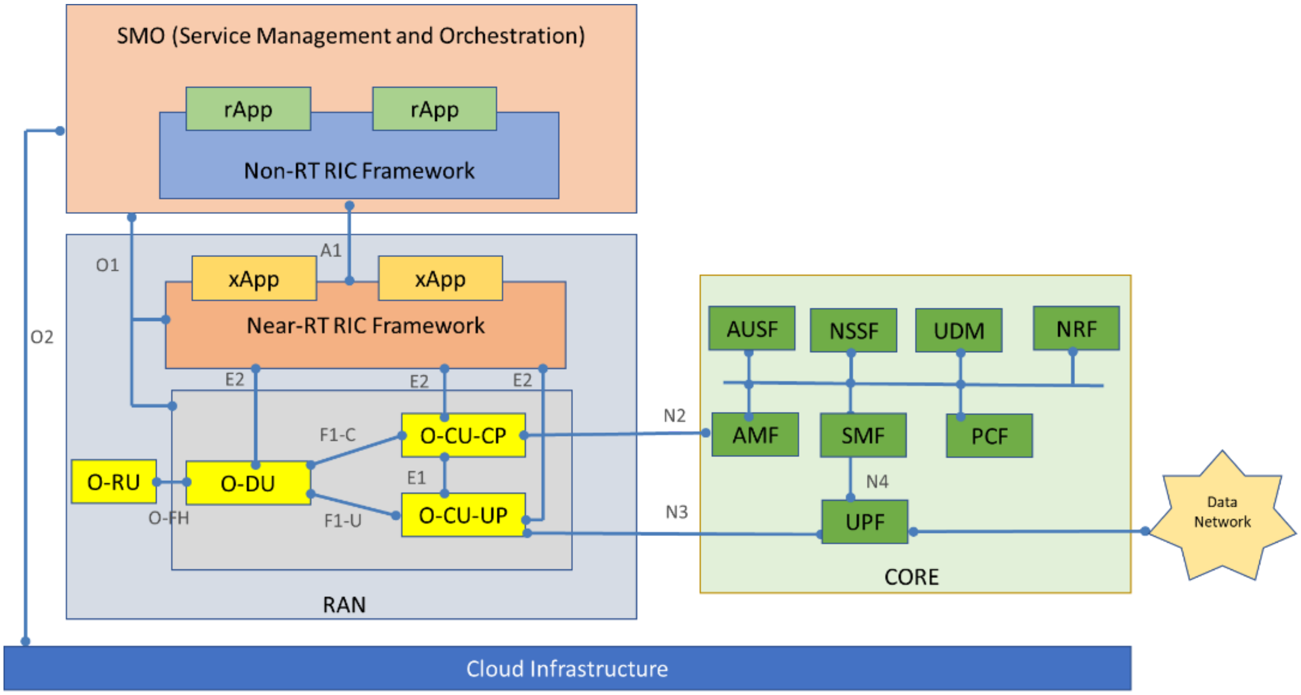

Figure 1 – O-RAN Architecture

While initial traction in open RAN has focused on the horizontal disaggregation of the RU-DU-CU, it will ultimately be the vertical disaggregation of intelligent control in x/rApps that will define the success of these solutions when competing against incumbent offerings. The real power in optimizing large RAN footprints will be in the x/rApps, and it is essential to take steps now to help incubate the availability of these apps. The ecosystem will only be successful if these x/rApps are ultimately available from a variety of vendors; without such availability vendor lock-in will persist and it will never become possible to truly mix-and-match components to build large, complex and highly optimized open RAN solutions that can compete with incumbent offerings. While the interoperability of RU/CU/DU is very important, the industry cannot wait any longer to begin to test and evaluate how RAN applications would interact with RU/CU/DU in a manner that fits into the paradigms of open RAN. The key aspects in this context are portability, interoperability, and performance of the apps (x/rApps) under different RIC and RAN implementations. In other words, the test and evaluation (T&E) efforts of RU/CU/DU interoperability should go hand in hand with x/rApp application portability, interoperability and performance T&E, with close industry coordination (rather than expecting x/rApp testing to take place serially after RU/CU/DU testing).

If one looks at the current open RAN landscape, the following points can be observed:

1. MNOs have been slow in adopting open RAN and most of the 5G is being rolled out on traditional architecture with incumbent vendors.
2. O-RAN specifications are not considered mature enough and hence vendors often have their own versions of interfaces.
3. Because of the above issue, interoperability and portability are often compromised among O-RAN components.
4. Nonetheless, the RU/CU/DU disaggregation is proceeding, supported by plug fests and similar efforts, but the application level (x/rApps) interoperability/portability with RIC and RAN is significantly lagging, which needs immediate attention since these applications drive RAN functionalities.
5. There is no efficient feedback mechanism to modulate the open standard specifications in O-RAN from an interoperability/portability testing mechanism – the current plug fests in this space seem to have limited impact to address this issue.
6. As open RAN is disaggregated in nature, there is a strong role for system integrators in this arena. However, system integrators can potentially choose their “favorite” component vendors without regard to fully open interfaces and recreate closed vertically integrated solutions. This could defeat the whole purpose of open RAN and create a new class of “incumbents.”

Open RAN – Portability and Interoperability Problem

Note that the O-RAN architecture in Figure 1 reveals that it is all about connecting a number of network functions (other than RU, the rest of the functions are virtualized or cloudified) through a number of open interfaces in a cloud native environment. This calls for a significant need for portability and interoperability, which need to be addressed, tested, and evaluated. As mentioned before, the scope of our discussions will be focused on the following aspects pertaining to x/rApps on their portability among various RIC implementations and their interoperability amongst the x/rApps and the RAN.

It is to be remembered that compliance with standards alone does not guarantee interoperability or portability (though it is a necessary first step). Different vendors can interpret the standards and specifications differently. This leads to different approaches in complying with specifications. This issue is not limited to specifications alone, and percolates into how vendors support various features and use cases throughout their product development and support life cycle. Often system integrators wrangle with varying levels of maturity of subsystems along with release and support conflicts. Hence the need for comprehensive interoperability and portability testing.

Open RAN – Application Portability

The question here is, can a given xApp be ported across all commercially available near-RT RICs and can a given rApp be ported across all commercially available non-RT RIC stacks? Additionally, RICs and x/rApps need to be tested against various cloud environments. Although these portability aspects may appear straightforward, there are several potential issues to be addressed.

First of all, let us consider cloud portability. On this front, the portability issues stem from the following aspects. The network functions (say RICs) are typically designed to run on specific hypervisors or virtualization technologies. If the target cloud platform of a Mobile Network Operator (MNO) uses a different hypervisor or virtualization technology than the vendor's development environment, compatibility issues may arise. This will then require modifications or redevelopment of the network function, which is an expensive proposition. Different cloud providers may have varying resource allocation policies, such as CPU, memory, and storage. Network functions (to be more accurate, virtual network functions or VNFs) often require specific resource allocation to meet performance requirements. Porting VNFs to different cloud environments may require adjusting resource allocation policies and optimizing performance to ensure consistent service quality, which creates various forms of customization.

It is to be noted that VNFs interact with the underlying cloud infrastructure through APIs and management interfaces. Different cloud providers may have proprietary or cloud-specific APIs, making it difficult to achieve interoperability and portability. VNFs may need to be modified to work with the APIs and interfaces of the new cloud environment. Also cloud providers have different security measures, access controls, and compliance frameworks. Porting VNFs to different cloud environments requires ensuring consistent security posture and compliance adherence, which may require customization.

Since the VNFs are sourced from different vendors by MNOs (for example non-RT RIC from vendor A and near-RT RIC from vendor B), a given VNFs from a particular vendor is often part of a larger service chain or network architecture. When porting VNFs to a different cloud environment, compatibility with the orchestration and management systems of the new environment becomes crucial. Ensuring seamless integration and functionality within the new orchestration framework can be complex. This is compounded by the fact that different cloud providers may have variations in terms of performance characteristics, including CPU, memory, and I/O capabilities. VNFs may require optimization and tuning to match the performance expectations in each MNO’s target cloud environments.

Lack of standardized interfaces and protocols between VNFs and cloud infrastructures hinders portability. Inconsistent implementations across cloud providers may require modifications or refactoring of VNFs and applications to ensure interoperability and seamless integration with network components. To address some of these critical issues, Europe's operators have pushed for a standards-based approach through initiatives such as Project Sylva [1], which has the following goals:

1. Release a cloud software framework to prioritize requirements, develop solutions to be integrated within existing open source components, and produce production-grade solutions to be leveraged within commercial products.
2. Develop a reference implementation of this cloud software framework and create an integration and validation program to accelerate adoption of network functions within the cloud.

While this is a great idea, the cloud portability problem needs to be addressed now, and it is not sufficient to hope that its complexities will be addressed progressively by initiatives such as Project Sylva.

Now the question of xApp/rApp portability with various RIC implementations. Here also lack of mature O-RAN standards hurts portability. Let us examine this closely.

Applications are typically designed to run on specific platforms or environments. Porting x/rApps to various RICs requires ensuring compatibility with each RIC's platform, including the operating system, runtime environment, and software dependencies. Both x/rApps interact with the RICs through APIs and interfaces. Differences in API/interface versions, protocols, or data models can pose challenges that need to be addressed.

Data model mapping requires serious attention. Both x/rApps often rely on specific data models that define the structure and format of the information they require to ingest or create. Mapping the data models used by the x/rApps to the data models supported by the RICs can turn out to be a complex task, especially if there are significant differences or mismatches.

Applications (x/rApps) almost invariably have performance requirements or expectations that need to be met in the RIC environment. The RIC's resource allocation, processing capabilities and performance characteristics may differ from the original platform where the applications were developed, necessitating performance analysis and optimization efforts during the porting and associated validation process.

The orchestration and management frameworks used by the applications and the RIC may differ. Porting applications to the RIC involves adapting or integrating with the RIC's orchestration and management systems, ensuring proper coordination, configuration and monitoring of the application within the RIC environment.

Open RAN – Application Interoperability (Interoperability between x/rApps)

To understand this problem, we use the following example. Consider rApp A doing energy saving by cell on/off (switching some cell on/off based on the load seen by cell sector) and rApp B doing traffic offload (moving traffic from a heavily loaded cell to a lightly loaded cell in an adjacent site).

Referring to **Figure 2**, let us assume that Site B is lightly loaded and further that Sector B3 is very lightly loaded, such that all cells except say one (to keep that alive for coverage purpose) are put to sleep by an Energy Savings rApp.

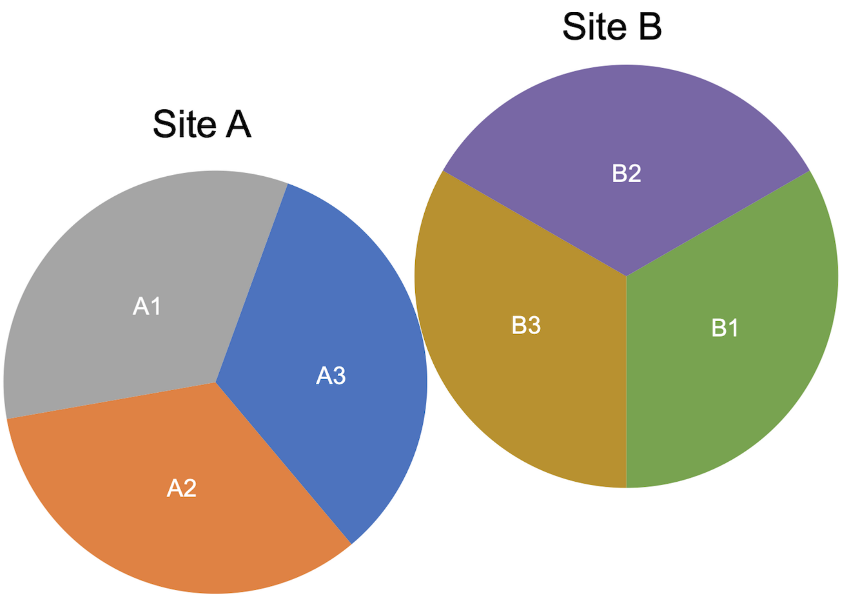

Figure 2 – Example of application coordination

Now let us assume that load on Site A is increasing heavily (especially say on Sector A3) and that a separate Traffic Steering xApp is trying to offload some of the traffic on Sector A3 to the cells in Sector B3 to preserve service quality. But since there may not be enough spare capacity for the offload (since the Energy Savings rApp has already removed some capacity from Sector B3), the operation could fail. This clearly illustrates the need for coordination between the x/rApps.

Under the O-RAN architecture, this sort of coordination and conflict resolution among x/rApps is done via policies. The specific policies and coordination mechanisms will depend on the nature of the applications, the complexity of interactions, and the requirements of the use case. Effective policy coordination ensures that the x/rApps work harmoniously to deliver a seamless user experience and achieve their respective objectives efficiently and securely.

ONF RRAIL Initiative

RRAIL is creating and operating a lab that assembles multi-vendor open RAN solutions, integrating RAN and RIC stacks as well as x/rApps from different vendors, and applying innovative test approaches and methodologies to assess x/rApp interoperability and portability. In parallel with assembling commercial RICs and x/rApp in the lab, an open source RIC and x/rApp stack will also be advanced and leveraged to test new concepts and ideas to advance the portability/interoperability aspects of open RAN, thereby promoting technology advancements and associated supply chain enhancements for 5G and beyond.

RRAIL Scope

At its core, RRAIL is a test lab environment, powered by cutting-edge tools and supporting processes to achieve the following:

1. Test the portability of xApps and rApps in multiple near-RT RIC/non-RT RIC combinations and SMO environments.
2. Test the interoperability among xApps, rApps, and among xApps and rApps. Evaluate how policy coordination works and how effective it is. For instance, an rApp which does cell on/off for energy saving from vendor A should be able to coordinate with a load balancing r/xApp from vendor B.
3. Test the interoperability between other RAN elements like RU and CU from an application perspective, and the ability of third-party x/rApps to control network elements from different vendors.
4. Test and evaluate support within different cloud environments, including:
    a. Virtualization – supports different hypervisors; and/or
    b. Cloud – supports different cloud hosting providers and/or private cloud solutions.

Figure 3 shows the scope of RRAIL and how it will complement traditional interoperability testing efforts like OTIC (O-RAN Testing and Integration Centers). Referring to Figure 3, many of the testing efforts focus on the interoperability among RU/CU/DU – which we may call horizontal testing and evaluation (T&E), indicated by the blue wide arrow. The scope of RRAIL is the T&E of the vertical application stack (near-RT RIC/xApps, non-RT RIC/rApps) along with how these applications (xApps/rApps) interoperate with various horizontal RAN solutions (RU/CU/DU) - shown by the wide green arrows.

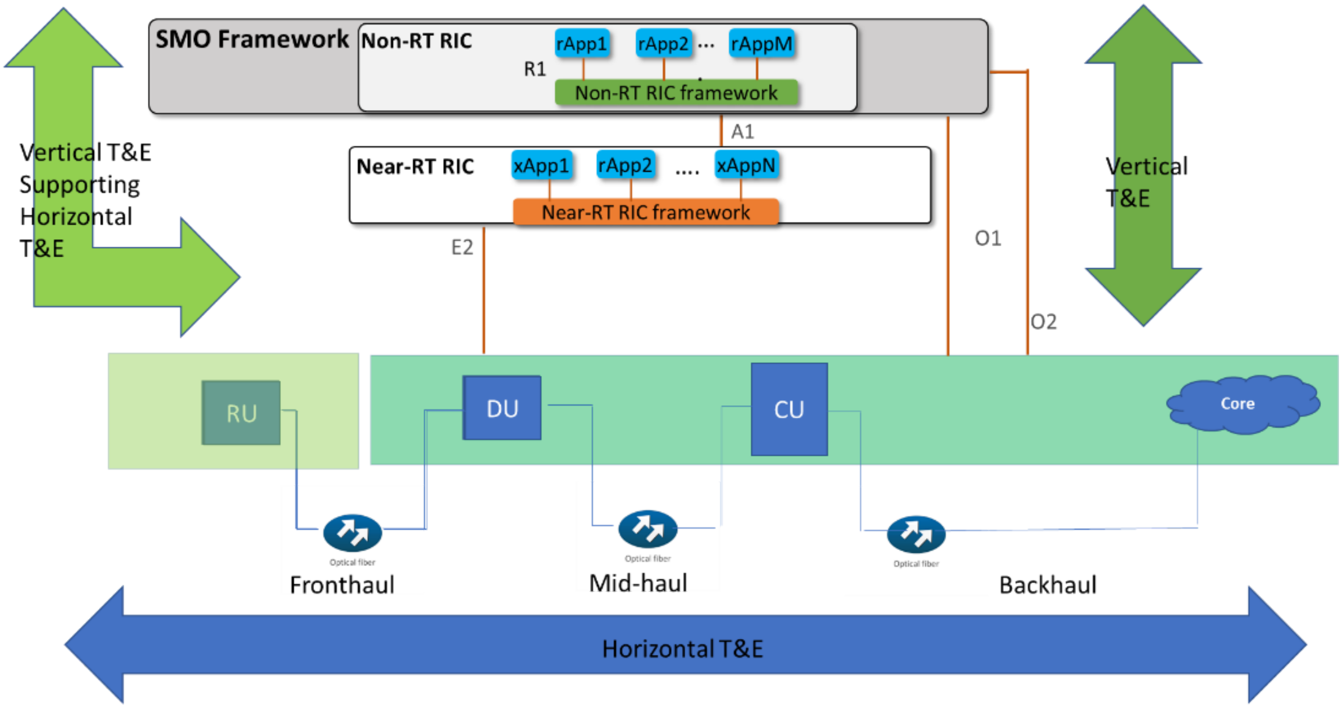

Figure 3 – RRAIL Scope

Given RRAIL's unique focus on the RIC and x/rApps, this effort will be complementing/supplementing the testing practices and labs that are already in place, like Rutgers WINLAB, COSMOS, POWDER, TIP, i14y and various OTIC initiatives.

The intent is not to benchmark activity comparisons between different RIC (non-RT, near-RT) implementations, or different x/rApps solutions from different vendors. Such activities are not in scope and are not to be executed in RRAIL. RRAIL is more concerned with achieving compatibility between different components rather than comparing two specific competing components from different vendors.

In terms of beneficiaries of this T&E Lab, ultimately it will be the consumers, but the entire ecosystem has much to gain from this initiative:

- Operators – They will have an expanded supply chain with best-of-breed solutions, rendering cost and operational efficiencies, aided by detailed information on application portability, interoperability, and performance.
- Existing vendors – They will be able to get timely and rich information about the portability, interoperability, and performance of their products to refine their product portfolio quickly. This will foster rapid technological advancement due to the

collaborative environment and the resulting ecosystem advantages, as opposed to pairwise business agreements.

- New vendors, especially small startups – They will gain access to a comprehensive T&E environment, one they could not afford to assemble on their own. Additionally, the ability to interact and leverage the resulting ecosystem of both operators and vendors will open up new business opportunities.

RRAIL Architecture

Figure 4 shows the vision of what will be tested in RRAIL. As mentioned before, the lab will consist of both commercial vendor RIC stacks and an open source stack. While each will be separate stacks, cross testing between the stacks will also be possible. To be more specific, a commercial RAN simulator can support the open source RIC/SMO stacks, or an open source RAN simulator can possibly pair with commercial RIC/SMO stacks. Thus, RRAIL is a lab and testing effort combining different components (commercial and/or open source). Similarly, Near-RT RIC, Non-RT RIC and SMO can be from different vendors including open source.

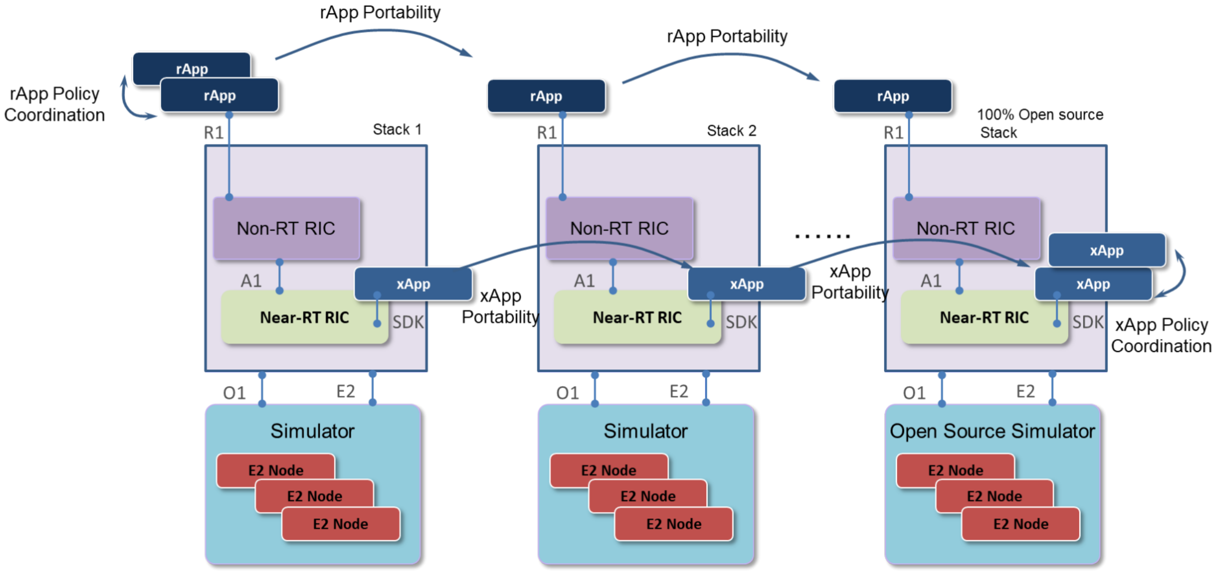

Figure 4 – RRAIL Structure

Examples of software stacks that fit into the RRAIL framework are:

- RAN Simulator: Products from Viavi and Keysight (commercial stacks); SD-RAN RANSIM (open source stacks from ONF), NS3 etc.
- SMO/Non-RT RIC/Near-RT RIC: Products from Juniper and VMware (commercial stacks); SD-RAN (open source near-RT RIC from ONF), non-RT RIC and Lightweight SMO from O-RAN Software Community (open source), etc.

The RRAIL open source stack is aimed at providing a flexible environment to facilitate xApp/rApp development to promote R&D activities in the open RAN space. This in turn would be a catalyst to create a forum for community/ecosystem interactions. Also, an open source stack would help vendors develop customized and scalable commercial stacks with richer functionality using applicable open source software components.

However, it is to be remembered that the open source stack is NOT a replacement or substitute for commercial stacks due to the fact that it is not of industrial strength or production grade, nor is it supported by a commercial backer.

The RRAIL Open Source stack is being built on the ONF SD-RAN project, which was initiated with the vision of building open source components for the mobile RAN space, complementing O-RAN's focus on architecture and interfaces by building and trialing O-RAN compliant open source components. This has been aimed at fostering the creation of true multi-vendor RAN solutions and help invigorate innovation across the RAN ecosystem. As the O-RAN specifications and industry needs evolve, SD-RAN continues to adapt to these needs.

Figure 5 is a more detailed view of of each of these software stacks:

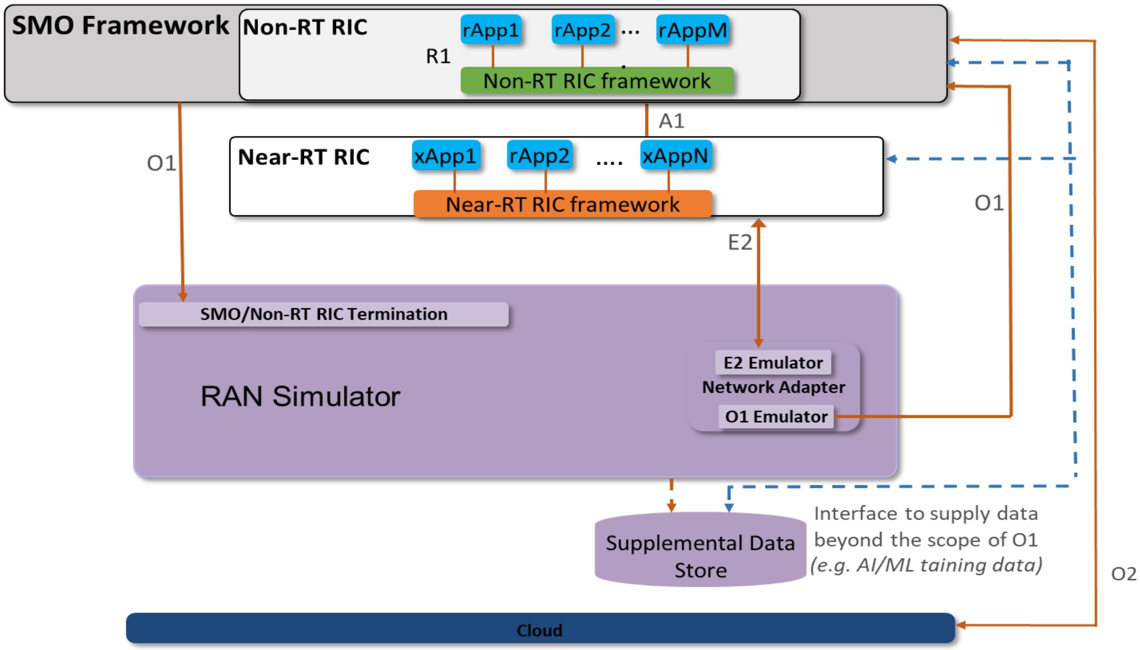

Figure 5 – Architecture of each software stack within RRAIL (based on O-RAN architecture)

Referring to Figure 5, the key components are:

- SMO - A design environment for rapid application development. A common data collection platform for management of RAN data as well as mediation for O1, O2 and A1 interfaces. Supports licensing, access control and AI/ML lifecycle management, along with legacy north-bound interfaces (NBI).
- Non-RT RIC/rApps - part of the SMO Framework, centrally deployed in the service provider network, which enables non-real-time (> 1 second) control of RAN elements and their resources through specialized applications called rApps.
- Near-RT RIC/xApps - resides within a telco edge cloud or regional cloud and is responsible for intelligent edge control of RAN nodes and resources with actions that typically take 10 milliseconds to one second to complete. It receives policy guidance from the non-RT RIC and provides policy feedback to the non-RT RIC through specialized applications called xApps.
- RAN Simulator – Simulates RAN at scale to test interoperability and portability. A detailed dissemination of the RAN simulator is provided in a later section.
- E2 and O1 emulators – Interface emulators per O-RAN specification.
- Supplemental data store – A data store for data that is needed extraneous to E2 and O1 interfaces (e.g. data needed to train rApps functioning using AI/ML approaches, network inventory etc.), transmitted to SMO and RICs via a special interface. [Note:

This capability may vary widely among vendors, depending on their architecture and implementation.]

Target Test Cases

We now look at a couple of initial target test cases, which are based on current industry needs. By no means is this an exhaustive list, and the priority could change from time to time.

RAN Energy Saving

The Energy Savings use case has gained significant momentum of late. ONF was an early advocate, spearheading a community initiative called **SMaRT-5G**, which is structured as a phased series of increasingly powerful PoCs designed to enable MNOs to start using the results from each PoC on both new disaggregated open RAN solutions as well as with traditional RAN architectures. The PoCs are being implemented on ONF RRAIL, and the first phase has already been demonstrated. **Figure 6** summarizes the phases of the SMaRT-5G project.

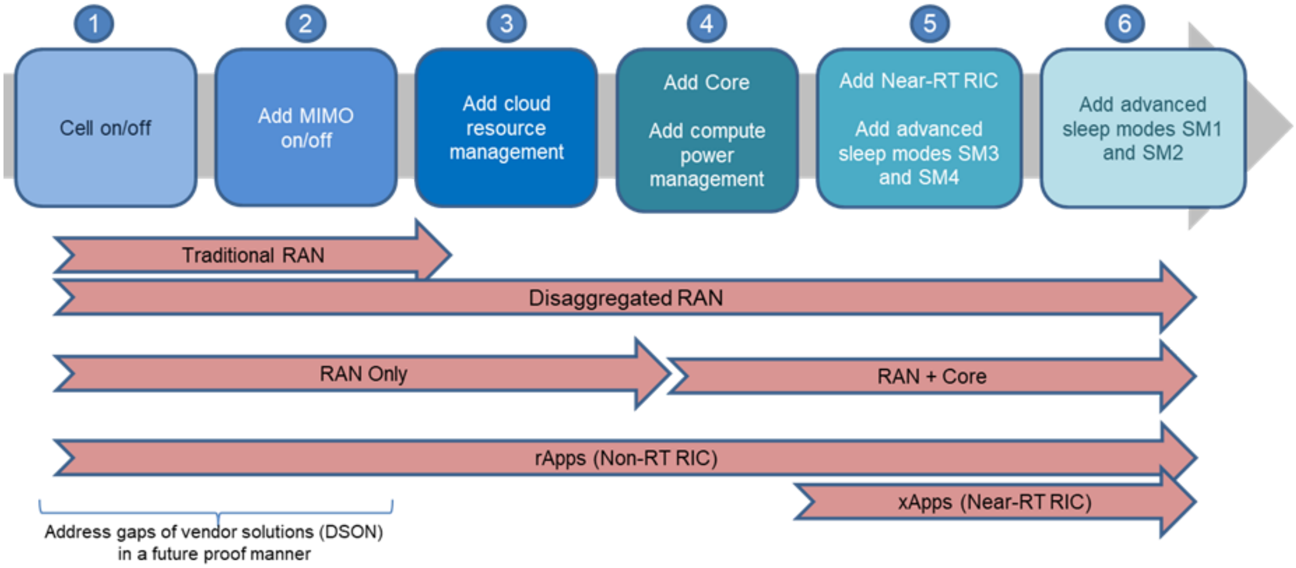

Figure 6 - SMaRT-5G

- Phase 1 - The first phase of SMaRT-5G is to demonstrate optimized cell on/off capabilities, which involves turning carrier frequencies on and off in a mobile network, based on load. ONF completed the first demonstration of this capability in October 2023 on a pure open source stack.

- Phase 2 - MIMO and RF channel switch on/off. During low traffic, the radio/antenna configuration can be changed from 8x8 to 4x4 or 2x2 MIMO, to save energy. Under Massive MIMO (M-MIMO), in low traffic cases, the M-MIMO panel can be put in deep sleep partially or fully, and progressively, saving energy; coverage impacts should be compensated when MIMO configurations are manipulated.
- Phase 3 – In addition to radio resources, cloud resources are scaled in/out (in a disaggregated RAN architecture where DU/CU are virtualized network functions).
- Phase 4 – Over and above cloud resources supporting RAN, Core capacity is also modulated based on load. In addition, CPU supporting the infrastructure is power optimized by adjusting voltage clock frequency along with disabling some CPU functions under low load conditions.
- Phase 5 and 6 – Advanced Sleep Modes (ASMs) are implemented where sub-components of the base station are progressively shut down depending on the activation and the deactivation times (transition time) of the different components. Phase 5 requires a transition time as low as 10 milliseconds whereas Phase 6 needs a lower transition time of 71 microseconds.

For details, please refer to the ONF white paper on SMaRT-5G [2].

Note the RRAIL architecture will be evolving to encompass Core as well to fully support the later phases of the SMaRT-5G PoC roadmap.

Traffic Steering

The Traffic Steering use case involves the detection and response to traffic load variation and interference. In a nutshell, it is all about moving a device in the same geographical area between different frequency layers (within a site or among a cluster of neighboring sites) to achieve specific goals (often throughput) when an overload condition or interference condition is detected. See **Figure 7**.

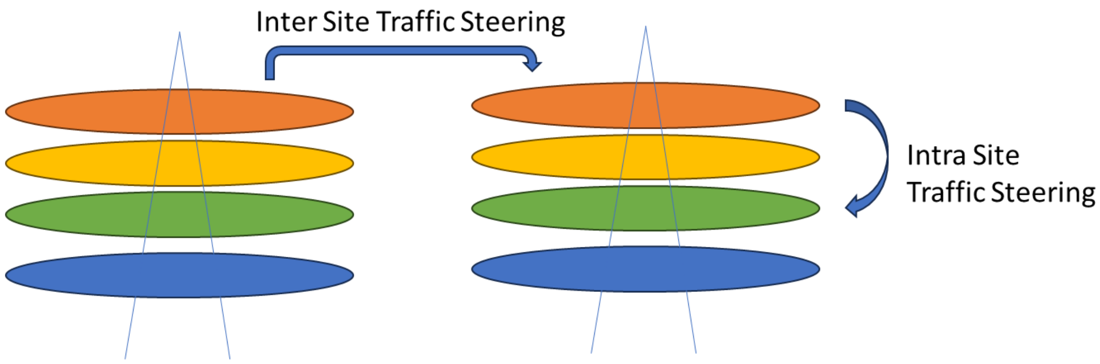

Figure 7 – Traffic Steering

The potential benefits of traffic steering include load balancing among different cells, traffic offloading, interference mitigation (by moving traffic from a highly interfered band to a less interfered band), etc., thus improving overall network Quality of Service (QoS) and possible overall network capacity as well.

It is to be noted that idle mode load balancing (IMLB) will also be considered under this use case, which is a technique used to distribute the mobile devices across different cells or sectors, even when they are not actively transmitting data. The goal of IMLB is to ensure that even when mobile devices are not actively using the network, they are connected to the most appropriate cell based on factors such as signal strength, interference, cell congestion, and overall network load, so that the user experience will be optimal when they actively start using the network.

Also, note that traffic steering can be combined with RAN energy saving in a way that traffic can be moved among cells to maximize the availability of capacity to be temporarily turned off, thus enhancing energy saving. In fact, the first phase of the SMaRT-5G project integrated both an Energy Savings rApp and a Traffic Steering xApp to be able to proactively move traffic away from cells that were candidates for being idled to save power.

Other Possible Test Cases

While the above two test cases are of high priority, examples of some of the other possible test cases are summarized below in a table, courtesy of O-RAN WG1. Details are omitted for brevity.

Test Case	Brief Description
RAN Slice SLA Assurance	Based on RAN specific slice SLA requirements, fine-tune RAN behavior to assure RAN slice SLAs dynamically.
Flight Path Based Dynamic UAV Resource Allocation	Flight path based dynamic UAV Radio Resource Allocation, allowing operators to adjust radio resource allocation policies through the O-RAN architecture, reducing unnecessary handover and improving radio resource utilization.
Multi-vendor Slices	Enablement of multiple slices with functions provided from different vendors.
Dynamic Spectrum Sharing	Enable adaptation and control of radio resource allocation policies to dynamically share radio spectrum between 4G and 5G networks over the O-RAN architecture, particularly when the 4G/5G CU/DU are from different vendors.
Signaling Storm Protection (DDoS Attack)	Detect and mitigate signaling storm DDoS attacks quickly and close to the network edge, minimizing affected network nodes while the legitimate UEs will continue to get service as usual.

Concluding Remarks

ONF has already launched a community effort to implement RRAIL and has started implementing Phase 1 of SMaRT-5G on RRAIL. A demonstration took place at FYUZ in October 2023.

ONF is soliciting additional collaborators from the vendor, operator and academic communities to further grow to this effort.

References

1. The Linux Foundation. “Linux Foundation Europe Announces Project Sylva to Create Open Source Telco Cloud Software Framework to Complement Open Networking Momentum.” The Linux Foundation, November 2022. https://www.linuxfoundation.org/press/linux-foundation-europe-announces-project-sylva-to-create-open-source-telco-cloud-software-framework-to-complement-open-networking-momentum

2. S. Puthenpura, T. Sloane, M. Dryjański, M. Ranganathan, V. Singh, A, Tiwari, C. Maciocco, and N. K. Shankaranarayanan. “SMaRT-5G Project - Sustainable Mobile and RAN Transformation (SMaRT).” The Open Networking Foundation, June 2023. https://opennetworking.org/wp-content/uploads/2023/07/ONF-SMaRT-5G-White-Paper-v11.pdf

About ONF

The Open Networking Foundation (ONF) is an operator-driven, community-led nonprofit consortium fostering and democratizing innovation in software-defined programmable networks. Through ecosystem building, advocacy, research, and education, ONF is accelerating the state-of-the-art in open networking and catalyzing creation and adoption of open disaggregated solutions leveraging open source software.