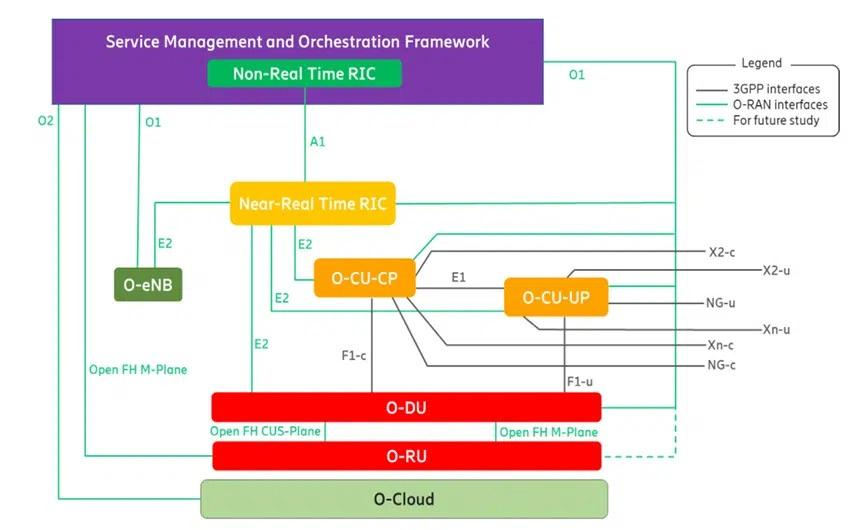
OSC Introduction

# RIC Overview

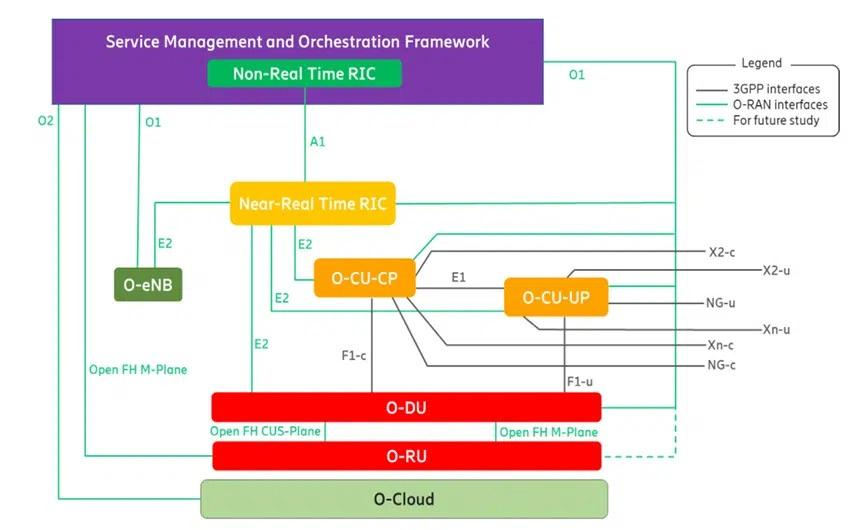


RIC is a software-defined component within the Open RAN architecture that functions to manage to do controlling and optimization within the RAN functions. RIC provides mobile operators with advanced network control, improved network performance, quick enablement of new services, AI/ML driven network control, and supports low latency applications. RIC effectively balances the RAN load, which alleviates network congestion and improves network performance. It customizes RAN functionality by optimization of regional resources, which enables quick launch of new services to build new revenue streams with personalized services. RIC provides advanced control functionality, leveraging analytics and data-drive approaches including advanced ML/AI tools to improve resource management capabilities.

The RIC facilitates multivendor interoperability by providing a common platform where various elements from different suppliers can be integrated and managed effectively. This interoperability is crucial for breaking the dominance of single-vendor networks, allowing operators more flexibility and choice in their deployments.

# Non-RT RIC Overview

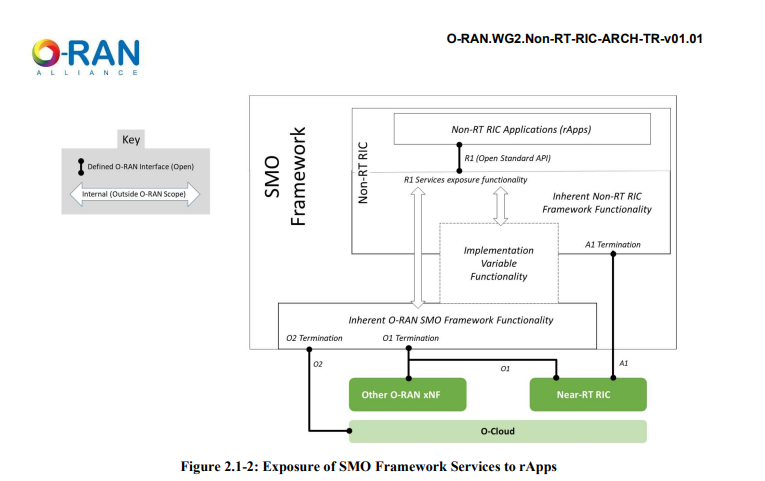
Non-RT RIC is used to manage operations that do not have tight timing. The non-RT RIC supports intelligent RAN optimization by providing policy-based guidance, ML model management and enrichment information to the Near-RT RIC function.



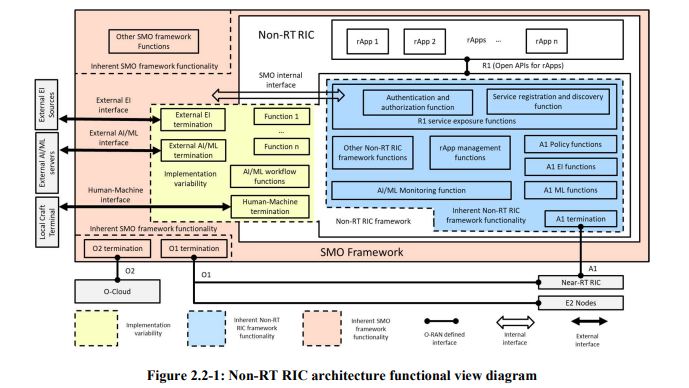
## Overall Architecture

From the architectural diagram above, non-RT RIC is portrayed as a function within the SMO and has direct connection with the A1 interface and responsible for the data sent and received via the A1 interface. It can access the SMO Framework functionality for RAN resource optimization.

The non-RT RIC can be thought of as a base platform that can be easily extended with additional applications (rApps). These rApps operate independently of the core non-RT RIC system, allowing for greater flexibility and easier updates or enhancements. The non-RT RIC exposes the SMO Frameworks functionalities to the rApps via the R1 interface.



## Non-RT RIC Architecture Functional View Diagram

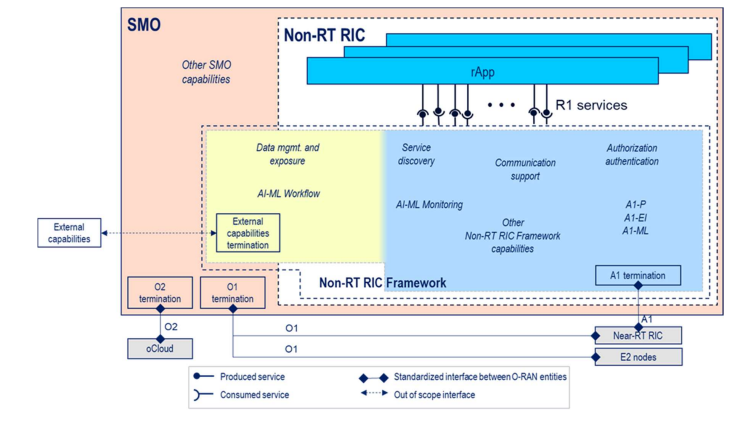


The functions within the blue part of the diagram are used to support the A1 interface and rApps, it is nature to deploy the functions inside the Non-RT RIC framework. “Other Non-RT RIC Framework Functions” refers to any other functions inside the Non-RT RIC that could be identified later in the study, for example like rApps orchestration and authentication. RApps could discover these Non-RT RIC frameworks and the other SMO frameworks via services registration and discovery function and needs to be authenticated and authorized before it can access those functions.

The yellow part of the diagram is called the implementation variability. If such a function is deployed within the Non-RT RIC, it becomes part of the Non-RT RIC framework, and its services are accessible to rApps through the "service registration and discovery function." Conversely, if the function is deployed in the SMO (Service Management and Orchestration) framework, it is not part of the Non-RT RIC framework, but its services are still discovered by rApps via the same "service registration and discovery function."

Interface termination is handled by the implementation variable functions. external EI termination, external AI/ML termination, and human-machine termination. n. External EI termination is connected to external EI sources to import enrichment information for Non-RT RIC applications. External AI/ML termination is connected to external AI/ML server for ML model importation. Human-machine termination is used to inject RAN intent manually. These terminations can reside inside either in the non-RT RIC or in the SMO.

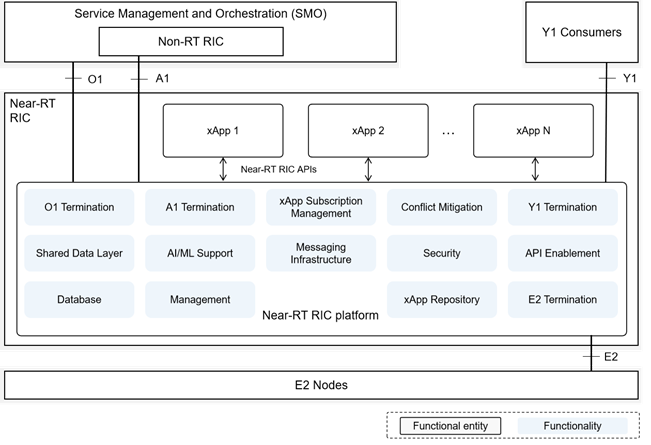
## Non-RT RIC Architecture Service-based View



# Near-RT RIC

Near RT RIC operates within a time frame that is shorter than traditional RAN management systems but not as immediate as the millisecond-level reactions required for some physical layer functions. The Near-RT RIC hosts one or more xApps that use E2 interface to collect near real-time information (e.g., on a UE basis or a Cell basis) and provide value added services. The Near-RT RIC control over the E2 Nodes is steered via the policies and the enrichment data provided via A1 from the Non-RT RIC. Based on the available data, the Near-RT RIC generates the RAN analytics information and exposes it via Y1 interface.

* Near RT RIC can only be connected to one non-RT RIC
* Near-RT RIC can be connected to multiple E2 Nodes (O-CU, O-DU)



## Near RT RIC Functions:

### DB and [SDL (Shared Data Layer)](https://wiki.o-ran-sc.org/pages/viewpage.action?pageId=20874400)

* **UE-NIB:** Functionality to maintain UE related info such as UE list and associated data and Tracking UE Identity. This information may be needed for the Near-RT RIC platform or an authorized xApps.
* **R-NIB:** Functionality to maintain RAN related info such as real-time information of the connected E2 nodes and mappings within them. This information may be needed for the Near-RT RIC platform or an authorized xApps.
* **SDL:** Provide SDL service for xApps. Can be used to subscribe to database notifications and to read, write, and modify information stored in the database

### xApp Subscription Management

This functionality is to enable the Near-RT RIC to handle subscription requests from the xApps for E2 related data.

* Managing subscriptions from xApps to E2 nodes
* Authorization of policies controlling xApp access to messages
* Merging identical subscriptions from different xApps into a single subscription toward an E2 Node
* Auditing existing subscriptions and do corrective actions

### Conflict Mitigation

Detect and resolve overlapping requests from multiple xApps.

### Messaging Infrastructure

Enables low latency message delivery between internal endpoints

* **Registration, discovery, and deletion of endpoints.**
* **API for sending and receiving message**
* **Multiple messaging mode:** P2P, pub/sub mode
* **Message routing:** sending message to different endpoint based on the routing info
* **Message robustness:** to avoid data loss during outage/restart or to relase resource from the messaging infrastructure once a message is outdated.

### Security

Mainly to prevent malicious xApps from abusing radio network info and control capabilities over RAN functions. Defined in 3GPP TS 33.501: “Security architecture and procedures for 5G System”.

### Management

[**OAM**](https://docs.o-ran-sc.org/projects/o-ran-sc-oam/en/latest/overview.html) **Management of Near-RT RIC.** This consist of fault, config, accounting, performance, file, security, and other management plane service

* **Fault Management**:
  + Near-RT RIC provides Fault Supervision MnS over the O1 interface.
* **Configuration Management**:
  + Near-RT RIC provides Provisioning MnS over the O1 interface.
* **Logging**:
  + Captures information for operation, troubleshooting, and performance reporting.
  + Log records can be viewed directly, indexed, stored, and used for metrics and reports.
  + Common logging format used for different logs: audit log, metrics log, error log, and debug log.
* **Tracing**:
  + Monitors transactions or workflows.
  + Example: Subscription workflow divided into subscription request trace and response trace.
  + Analyzes individual traces to understand timing latencies.
* **Metrics Collection**:
  + Collects and publishes performance and fault management metrics for each xApp logic and internal functionalities.
  + Metrics collection mechanism needed for reporting to authorized consumers (e.g., SMO).

### Interface Termination

Interface termination for these following interfaces:

* E2
* O1
* Y1

### API Enablement

API can be categorized based on the interaction with the Near-RT RIC platform and can be related to E2-related services, A1-related services, Management related services, and SDL services.

* Repository/Registry services for the Near-RT RIC APIs
* Services for discovery of the registered Near-RT RIC API
* Service for authenticating xApps use of Near-RT RIC APIs
* Service for enabling generic subscrription and event notification
* Tools to avoid compatibility clash between xApps and services they access

### AI/ML SUpport

* Data Pipelining
* Model Management
* Training
* Inference