

# Release D

## Primary Goals for Non-RealTime RAN Intelligent Controller (Non-RT RIC):

- The primary goal of Non-RT RIC is to support intelligent RAN optimization by providing policy-based guidance, ML model management and enrichment information to the near-RT RIC function so that the RAN can optimize, e.g., RRM under certain conditions.
- It can also perform intelligent radio resource management function in non-real-time interval (i.e., greater than 1 second).
- Non-RT RIC can use data analytics and AI/ML training/inference to determine the RAN optimization actions for which it can leverage SMO services such as data collection and provisioning services of the O-RAN nodes.
- Non-RT-RIC will define and coordinate rApps (Non-RT-RIC applications) to perform Non-RT-RIC tasks.
- Non-RT-RIC will also host the new R1 interface (between rApps and SMO/NONRTRIC services)

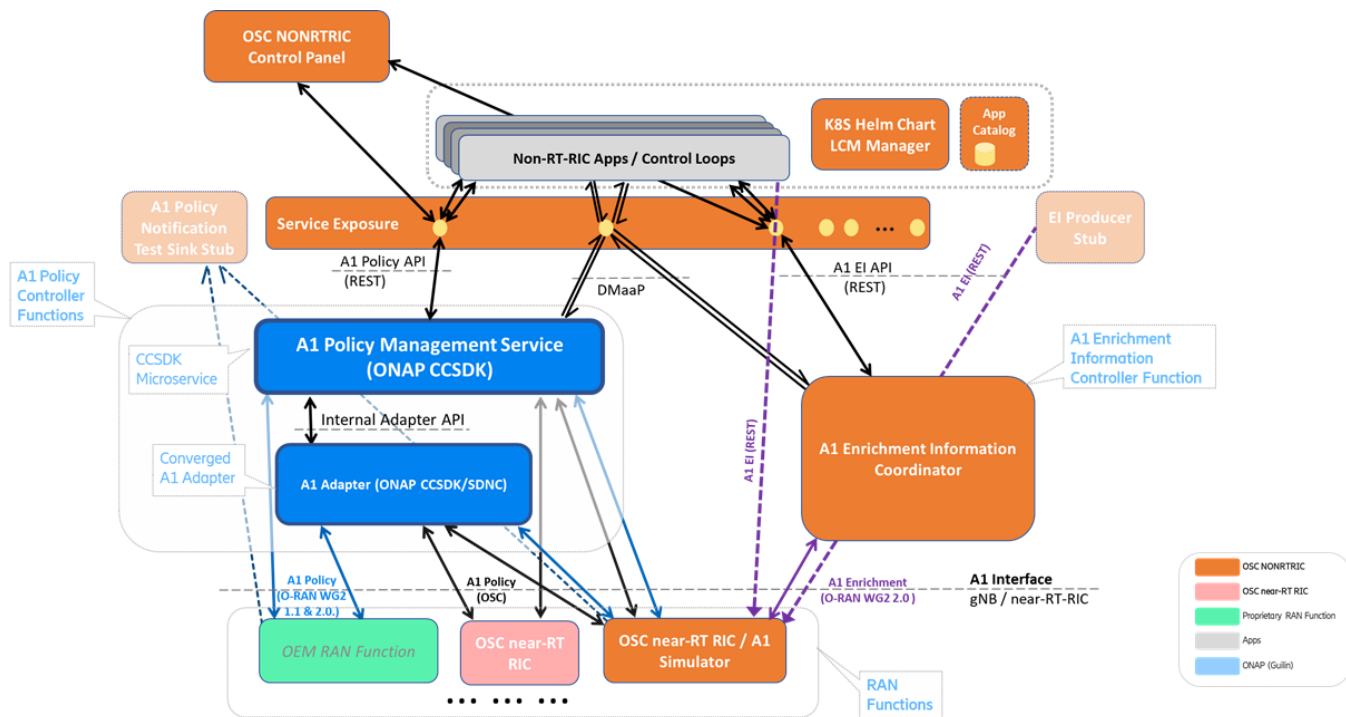
## Overall objective for the D Release:

*In the D Release we focus mainly on continued improved and evolution for the O-RAN A1 interface, and providing some building blocks to support the emerging Non-RT-RIC Apps ("rApps") and R1 interface concepts from O-RAN.*

## D Feature Scope:

- NONRTRIC Functions:
  - Integrated A1 Adapter from ONAP (controller – mediation)
  - Integrated A1 Policy Management Service from ONAP (controller – A1 policies)
  - OSC A1 Enrichment Information Coordinator (controller – A1 EI Job management)
  - OSC Non-RT-RIC Control Panel (GUI – for A1-P & A1-EI Job management)
  - OSC A1 Simulator (a stateful test stub to simulate near-RT-RIC end of A1 interface – A1-P & A1-EI)
  - OSC (initial) APP catalog (for registering/querying APPs)
  - Initial K8S Helm Chart LCM Manager - for APP µServices etc. (ONAP & OSC) (**new**)
  - Initial Service Exposure Function (**new**)
- In D Release:
  - Improved A1-PMS NBI (REST & DMaaP) (Rest style alignment)
  - Runtime configuration API (REST) for A1 Policy Management Service (e.g. add/remove adapters, near-rt-rics, security certs, etc)
  - Deployment – Continued improvements for Docker & Kubernetes
  - Extended/Easier deployment options with OSC IT/DEP project (SMO/NONRTRIC deployment)
  - Improving CI/CD to support include A1 Policy controller dependencies from ONAP
  - Multi-version support ( O-RAN A1-AP v1.1, v2.0, v2.1,v3.0 & OSC pre-spec A1)
  - Improved status monitoring/notification of A1-EI Jobs
  - Further improvement in security cert management (All interfaces can now be secured using TLS)
  - Re-architect & improve usability of Non-RT-RIC Control Panel (GUI)
  - Extend NONRTRIC Control Panel to sort and filter A1 Policies
  - Extend NONRTRIC Control Panel to sort and filter A1 Enrichment Types/Jobs
  - Extend NONRTRIC Control Panel to configure A1 Policy Management Service
  - Configurable Service Exposure function – Extends/Replaces static exposure gateway in OSC Cherry
  - K8S Helm Chart LCM function for App µServices
  - Update NONRTRIC demo/test environment (one-click tests/use-cases, docker & single/multi-node K8s env)
  - OSC e2e integration use case – O-RU-FH>HelloWorld recovery
    - App to instigate O-RU-FH connection recovery after failure – via O-DU
      - Multiple implementation options – standalone µService and/or deployable ONAP-PF policy script
  - CII badging – Already achieved Bronze/Passing Grade

## Architecture for Release D:



## NONRTRIC components:

1. Non-RT-RIC Control Panel
2. Non-RT-RIC (Spring Cloud) Service Gateway
3. Non-RT-RIC (Kong) Service Exposure Prototyping
4. A1 Policy Management Service
5. Enrichment Information Coordinator
6. Initial Non-RT-RIC App Catalogue
7. SDNC A1 Controller
8. Near-RT RIC A1 Simulator
9. Initial K8S Helm Chart LCM Manager
10. Test Framework
11. "HelloWorld" O-RU Fronthaul Recovery use-case

The code base for "D" Release is in the [NONRTRIC](#), [NONRTRIC-ControlPanel](#), and [A1-Simulator](#), source repositories (Gerrit) - D Branch

## Non-RT-RIC Control Panel

Graphical user interface to interact with the NON-RT RIC services.

- View and Manage A1 policies in the RAN (near-RT-RICs)
- Graphical A1 policy creation/editing is model-driven, based on policy type's JSON schema
- View and manage producers and jobs for the Enrichment coordinator service
- Configure A1 Policy Management Service (add/remove near-rt-rics)
- Interacts with the A1-PolicyManagementService & A1-EI-Coordinator (REST NBIs) via Service Exposure gateway
- Implementation:
  - Frontend: Angular framework
  - Repo: [portal/nonrtic-controlpanel/web-frontend](#)

Please refer [this](#) developer guide to set up in your local environment. More information about Non-RT RIC control panel can be found [here](#).

## Non-RT-RIC (Spring Cloud) Service Gateway

Support Apps to use A1 Services (May be replaced by Service Exposure function in later releases)

Spring cloud Gateway provides the library to build the API Gateway for Micro-service architecture. In Non-RT-RIC we build the basic API gateway using spring cloud gateway which then exposes two Non-RT RIC functions; Policy Management Service & Enrichment Coordinator Service. You can add predicates through code or yaml and in Non-RT-RIC we prefer to use yaml.

NONRTRIC gateway code can be found at:

- <https://gerrit.o-ran-sc.org/r/gitweb?p=portal/nonrttric-controlpanel.git;a=tree;f=nonrttric-gateway;hb=refs/heads/dawn>

More information on the Spring Cloud Gateway can be found in the documentation [here](#).

## Non-RT RIC (Kong) Service Exposure Prototyping

*Support Apps to use NONRTRIC, SMO and other App interfaces*

*A building block for coming releases as the R1 Interface concept matures*

- Support dynamic registration and exposure of service interfaces to Non-RT-RIC applications (& NONRTRIC Control panel)
- Extends a static gateway function specifically for NONRTRIC Control panel
- Initial version based on Kong gateway function
- Initial exposure candidates include A1 (NONRTRIC) services & O1 (OAM/SMO) services

Kong is a cloud-native, high performance, scalable & Open source API Gateway. Kong comes in 2 flavours

- Without Database
- With Database like PostgreSQL or Cassandra

The NONRTRIC Kubernetes deployment will deploy Kong if the *installKong* flag is set to true. During uninstallation of *nonrttric* components it will also remove Kong if it's deployed by *nonrttric* script.

If the ingress enabled flag is set to true then it will create the ingress objects for A1 Policy & Enrichment Service so the Kong gateway (acts as ingress controller) will expose these functions.

NONRTRIC Kubernetes deployment can be found at:

- <https://gerrit.o-ran-sc.org/r/gitweb?p=it/dep.git;a=tree;f=nonrttric/helm>

More information on Kong API Gateway can be found a,

<https://docs.konghq.com/gateway-oss/>

<https://github.com/Kong/charts/blob/main/charts/kong/README.md>

### **NOTE:**

Kong installation done by *nonrttric* script is Without Database.

## A1 Policy Management Service (from ONAP CCSDK – Honolulu)

*A1 Controller Service above A1 Adaptor that provides:*

- Unified REST & DMaaP APIs for managing A1 Policies in all near-RT-RICs
- Operations:
  - Query A1 Policy Types in near-RT-RICs
  - Create/Query/Update/Delete A1 Policy Instances in near-RT-RICs
  - Query Status for A1 Policy Instances
- Maintains (persistent) cache of RAN's A1 Policy information
  - Support RAN-wide view of A1 Policy information
  - Streamline A1 traffic
  - Enable (optional) re-synchronization after inconsistencies / near-RT-RIC restarts
  - Added support for multiple near-RT-RICs (& multi-version support)
- Unified REST & DMaaP NBI
- Converged ONAP & O-RAN-SC A1 Adapter/Controller functions in ONAP SDNC/CCSDK
  - (Optionally deploy without A1 Adaptor to connect direct to near-RT-RICs)

Documentation about the service can be found at:

- <https://docs.o-ran-sc.org/projects/o-ran-sc-nonrttric/en/dawn/>

## A1 Enrichment Information Job Coordination Service

*Coordinate/Register A1-EI Types, Producers, Consumers, and Jobs.*

- Maintains a registry of:
  - A1-EI Data Types / schemas
  - A1-EI Producers
  - A1-EI Consumers
  - A1-EI Jobs
- A1-EI Query API (e.g. per producer, per consumer, per types)
- Query status of A1-EI jobs
- Monitors all near-RT-RICs and recovers from inconsistencies
- After EI-type/Producer/Consumer/Job is successfully registered delivery/flow can happen directly between A1-EI Producers (in SMO/NONRTRIC domain) and A1-EI consumers (near-RT-RICs in RAN domain)

- Being extended to coordinate non-A1 Enrichment Information exchange between NONRTRIC Apps

Documentation about the service can be found at:

- <https://docs.o-ran-sc.org/projects/o-ran-sc-nonrtric/en/dawn/>

## (Initial) Non-RT RIC APP catalog

*Register for NONRTRIC APPs*

- APPs can be registered / queried
- Limited functionality/integration for now
- A building block for coming releases as the R-APP concept matures

<https://docs.o-ran-sc.org/projects/o-ran-sc-nonrtric/en/dawn/>

## A1 Adaptor / Controller (from ONAP CCSDK – Honolulu):

*Mediation point for A1 interface termination in SMO/NONRTRIC*

- Implemented as CCSDK OSGI Feature/Bundles
- A1 REST southbound
- RESTCONF Northbound
- NETCONF YANG > RESTCONF adapter
- Mapping logic / Provider
- Can be included in an any controller based on ONAP CCSDK
- OSC Dawn will continue to support both a dedicated A1-controller persona (*NONRTRIC*) and an SDNC/SDNR persona (*IT/DEP, OAM/SMO*)

Documentation about the adaptor can be found at:

Documentation about the service can be found at:

- <https://docs.o-ran-sc.org/projects/o-ran-sc-nonrtric/en/dawn/>

## Near-RT-RIC Simulator:

*Stateful A1 test stub*

- Used to create multiple stateful A1 providers (simulated near-rt-rics)
- Supports A1-Policy and A1-EnrichmentInformation
- Implemented as a Python application
- Swagger-based northbound interface, so easy to change the A1 profile exposed (e.g. A1 version, A1 Policy Types, A1-E1 consumers, etc)
- All A1-AP versions supported

Documentation about the simulator can be found at:

- <https://docs.o-ran-sc.org/projects/o-ran-sc-sim-a1-interface/en/latest/>

## Initial K8S Helm Chart LCM Manager

*Onboard, start, stop, and modify Non-RT-RIC App μServices as Helm Charts*

*A building block for coming releases as the R-APP concept matures*

- Interfaces that accepts Non-RT-RIC App μServices Helm Charts
- Support basic LCM operations
- Onboard, Start, Stop, Modify, Monitor
- Initial version co-developed with v. similar functions in ONAP
- Limited functionality/integration for now

## NONRTRIC Test Platform

Information about the test platform can be found at:

- <https://wiki.o-ran-sc.org/display/RICNR/Function+Test>

## "Helloworld" O-RU Fronthaul Recovery use-case

Information about the use case can be found at:

- [Release D: Simple script for O-RU & O-DU use case](#)

Code for the use case can be found at:

- <https://gerrit.o-ran-sc.org/r/gitweb?p=nonrtric.git;a=tree;f=test/usecases/oruclosedlooprecovery;hb=refs/heads/dawn>