Release F

Summary

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 host the A1 interface (between NONRTRIC & near-RT RICs)
- Non-RT-RIC will also host the new R1 interface (between rApps and SMO/NONRTRIC services)

Overall objective for the F Release

In the F Release we focus mainly on prototyping building blocks to support Non-RT-RIC Apps ("rApps") and R1 interface concepts from O-RAN. For the R1 interface we will focus on enabling Service Management & Exposure (R1-SME), and demonstrating Data Management & Exposure (R1-DME).

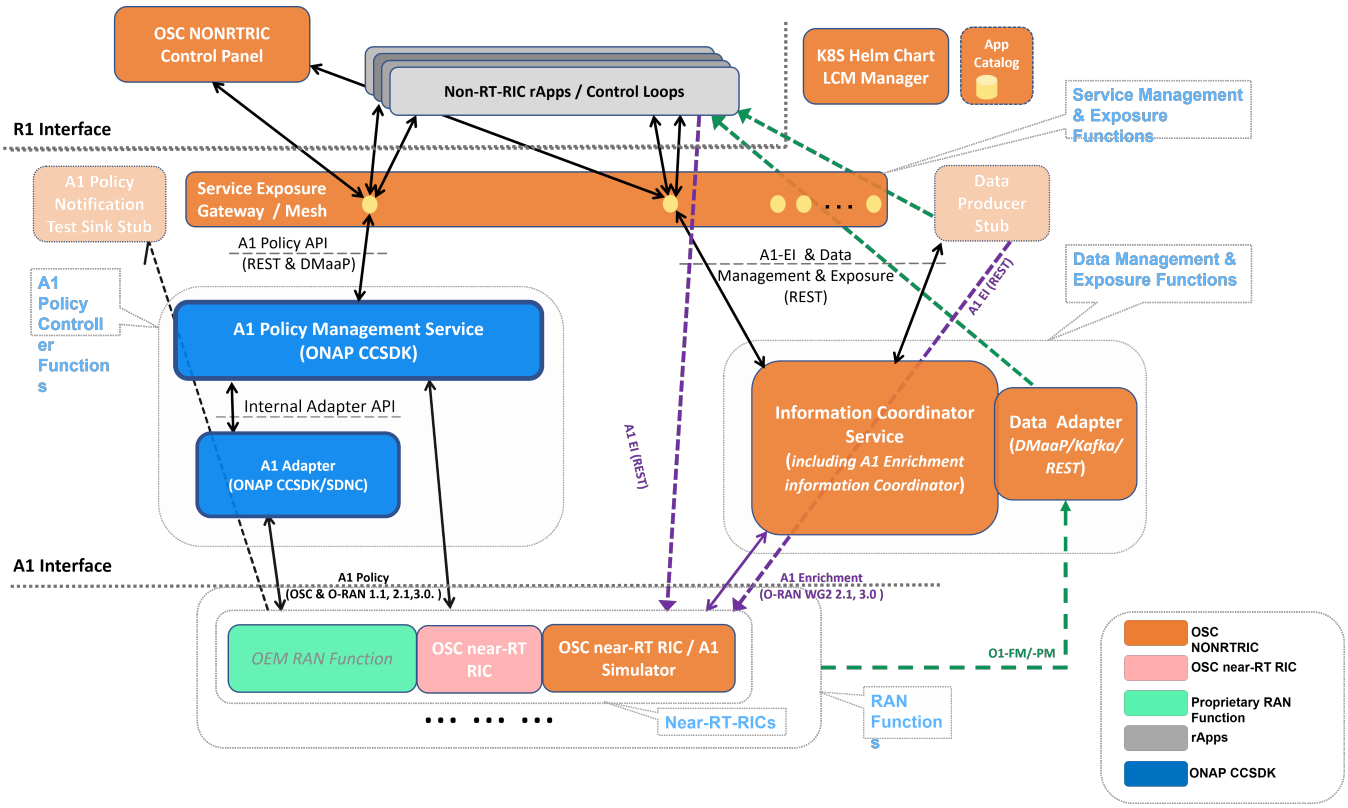
F Release Priorities

- Study & prototype Coordinated Service Exposure (SE)
  - Continue SE contribution building on the manual approaches already studied/completed.
  - Create/apply K8S configurations to isolate platform services and rApp microservices, then configure controlled secure access between service
  - Prototype CAPIF compliant API for Service/rApp registration/discovery, and service provider/consumer registration/configuration
- Data Management & Exposure (DME):
  - Pre-spec O1 PM via pre-spec R1 DME demo
  - Configure & connect to PM data - collected by SMO (ONAP)
  - Collect, Filter & Coordinate Delivery of PM data from DMaaP/Kafka to rApps over R1 (ICS)
- General activities
  - Continue to provide spec-compliant implementation of A1-Policy & A1-EI functions
  - NONRTRIC source-code repository re-organization
  - Continue to integrate and deploy SMO/NONRTRIC platform/rApps in OSC integration environment.
  - Continue to expand NONRTRIC test platform & testsets
  - Show various versions rApps implemented/deployed as holistic “Automation Compositions” (ref ONAP ACM)
  - Continue to provide & integrate strawman rApps to comply with OSC RSAC integration usecases.

F Release Feature Scope

- NONRTRIC Functions:
  - OSC Information Coordinator (controller – Data Management & Exposure & A1 Enrichment Information (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)
  - Initial OSC APP catalog (for registering/querying APPs)
  - K8S Helm Chart LCM Manager - for APP µServices etc. (ONAP & OSC)
  - Exposure Gateway Functions
  - Coordinated service exposure for R1 interface
  - DMaaP/Kafka Information Producer Mediator
  - Integrated A1 Adapter from ONAP (A1 Policy (A1-P) controller – mediation)
  - Integrated A1 Policy Management Service from ONAP (A1 Policy (A1-P) controller)
  - Manager for rApps / Automation Composition Manager (ONAP & OSC)

Architecture for Release F



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. **Information Coordinator Service**
6. **DMaaP/Kafka Information Producer Adapters**
7. **Initial Non-RT-RIC App Catalogue**
8. **A1 Policy Controller / Adapter**
9. **A1 Simulator (previously called Near-RT RIC A1 Interface)**
10. **Initial K8S Helm Chart LCM Manager**
11. **Test Framework**
12. **"Helloworld" O-RU Fronthaul Recovery use case**
13. **"Helloworld" O-DU Slice Assurance use case**
14. **"Healthcheck" use case**

NONRTRIC Control Panel

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

With the OSC NONRTRIC Control Panel you can:

- 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 Information Coordination Service
- Configure A1 Policy Management Service (add/remove near-rt-rics)
- Interacts with the A1-Policy Management Service & Information Coordination Service (REST NBIs) via Service Exposure gateway
- Implementation:
    - Front-end: Angular framework
    - Repo: portal/nonrtric-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.

NONRTRIC (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/nonrtric-controlpanel.git;a=tree;f=nonrtric-gateway

More information on the Spring Cloud Gateway can be found in the documentation here.

NONRTRIC (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 flavors

- 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 *nonrtric* components it will also remove Kong if it's deployed by *nonrtric* 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=nonrtric/helm
  NOTE: Kong installation done by *nonrtric* script is 'Without Database'.

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

A1 Policy Management Service (*from ONAP CCSDK – Istanbul*)

A1 Policy Controller Service above A1 Adapter 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 Adapter to connect direct to near-RT-RICs)
- Support for different Southbound connectors per near-RT-RIC - e.g. different A1 versions, different near-RT-RIC version, different A1 adapter /controllers supports different or proprietary A1 controllers/EMSs

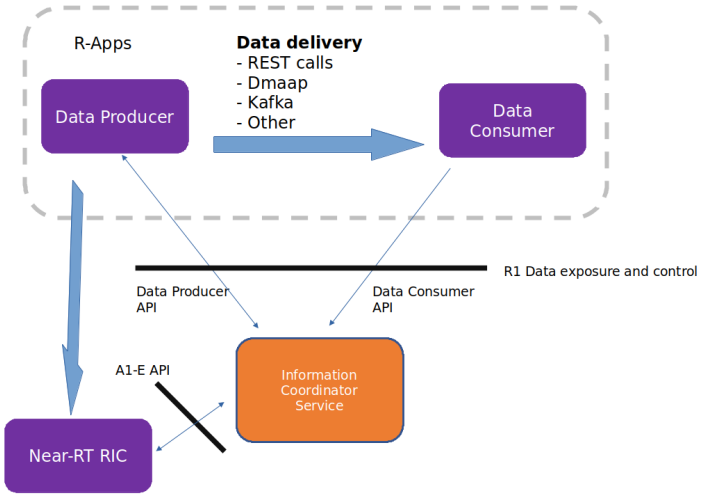
Repository and documentation about the service can be found at:

- https://gerrit.o-ran-sc.org/r/admin/repos/nonrtric/plt/a1policymanagementservice
- https://docs.o-ran-sc.org/projects/o-ran-sc-nonrtric-plt-a1policymanagementservice/en/f-release
- https://docs.onap.org/projects/onap-ccsdk-oran/en/jakarta/
- A1 Policy Management Service in ONAP

Information Coordinator Service (ICS)

A service that coordinates Information Jobs, producers and consumers. Also acts as an A1 Enrichment Information Controller.

ICS is a data subscription service which decouples data producers from data consumers. A data consumer can create a data subscription (Information Job) without any knowledge of its data producers (one subscription may involve several data producers). A data producer has the ability to produce one or several types of data (Information Type). One type of data can be produced by zero to many producers.



A data consumer can be an R-App using R1 APIs, or a near-RT RIC using the A1-E API (where the subscribed data is more specifically called "Enrichment Information").
A data consumer can have several active data subscriptions (Information Job). One Information Job consists of the type of data to produce and additional parameters, which may be different for different data types. These parameters is not defined or limited by this service and may for instance include:

- Parameters related to delivery (Kafka stream, callback URL etc.). These are different for different delivery protocols.
- Filtering information (scope, filter or other discriminators).
- Period-icy
- Other info used for aggregation

Repository and documentation about the service can be found at:

- https://gerrit.o-ran-sc.org/r/admin/repos/nonrtric/plt/informationcoordinatorservice
- https://docs.o-ran-sc.org/projects/o-ran-sc-nonrtric-plt-informationcoordinatorservice/en/f-release

APIs provided by the ICS service

A1-EI

This API is between Near-RT RIC and the Non-RT RIC. The Near-RT RIC is a data consumer, which creates Information Jobs to subscribe for data. In this context, the information is referred to as 'Enrichment Information', EI.

Data producer API

This API is provided by the Non-RT RIC platform and is intended to be part of the O-RAN R1 interface. The API is for use by different kinds of data producers and provides support for:

- Registry of supported information types and which parameters needed to setup a subscription.
- Registry of existing data producers.
- Callback API provided by producers to setup subscriptions.

Data consumer API

This API is provided by the Non-RT RIC platform and is intended to be part of the O-RAN R1 interface. The API is for use by different kinds of data consumers and provides support for:

- Querying of available types of data to consume.
- Management of data subscription jobs
- Optional callback API provided by consumers to get notification on added and removed information types.

Service status

This API provides a means to monitor the health of this service.

DMaaP/Kafka Information Producer Adapters

Configurable mediators to take information from DMaaP (& Kafka) and present it as a coordinated Information Producer

Two alternative implementations to allow Information Consumers to consume DMaaP or Kafka events as coordinated Information Jobs.
These configurable adapters/mediators act producers of Information Coordinator Service (ICS) jobs by polling topics in DMaaP Message Router (MR) or Kafka and pushing the messages to a consumer.

- A version implemented in Java (Spring) - Supporting filtered DMaaP and Kafka mediation:
  - https://gerrit.o-ran-sc.org/r/admin/repos/nonrtric/plt/dmaapadapter
  - Release F - Run in Docker#RuntheDmaapAdaptorServiceDockerContainer
  - https://docs.o-ran-sc.org/projects/o-ran-sc-nonrtric-plt-dmaapadapter/en/f-release
- A version implemented in Go - Supporting DMaaP mediation:
  - https://gerrit.o-ran-sc.org/r/admin/repos/nonrtric/plt/dmaapmediatorproducer
  - Release F - Run in Docker#RuntheDmaapMediatorProducerDockerContainer
  - https://docs.o-ran-sc.org/projects/o-ran-sc-nonrtric-plt-dmaapmediatorproducer/en/f-release

(Initial) NONRTRIC 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*

Repository and documentation about the service can be found at:

- https://gerrit.o-ran-sc.org/r/admin/repos/nonrtric/plt/rappcatalogue
- https://docs.o-ran-sc.org/projects/o-ran-sc-nonrtric-plt-rappcatalogue/en/f-release

A1 Policy Controller / Adapter (*from ONAP CCSDK – Istanbul*)

Optional mediation point for A1 Policy 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

Documentation about the adapter / controller can be found at:

- https://docs.onap.org/projects/onap-ccsdk-oran/en/jakarta/
- CCSDK A1 Adapter for A1 Policies in ONAP

A1 Simulator

Stateful A1 test stub

- Used to create multiple stateful A1 providers (simulated near-rt-rics)
- Supports A1-Policy (A1-P) and A1-Enrichment Information (A1-EI)
- 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-EI consumers, etc)
- All A1-AP versions supported
- Call-out feature to add applications-specific A1-Policy behaviors (REST & Kafka call-outs supported)

Repository and documentation about the service can be found at:

- https://gerrit.o-ran-sc.org/r/admin/repos/sim/a1-interface

- https://docs.o-ran-sc.org/projects/o-ran-sc-sim-a1-interface/en/f-release

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 rApp 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

Repository and documentation about the service can be found at:

- https://gerrit.o-ran-sc.org/r/admin/repos/nonrtric/plt/helmmanager
- https://docs.o-ran-sc.org/projects/o-ran-sc-nonrtric-plt-rappcatalogue/en/f-release

NONRTRIC Test Platform

Information about the test platform can be found at:

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

Use Cases

"Helloworld" O-RU Fronthaul Recovery use case

A very simplified closed-loop rApp use case to re-establish front-haul connections between O-DUs and O-RUs if they fail. Not intended to to be 'real-world'

Information about the use case can be found at:

- Release F: O-RU Fronthaul Recovery usecase

Repository and documentation about the service can be found at:

- https://gerrit.o-ran-sc.org/r/admin/repos/nonrtric/rapp/orufhrecovery
- https://docs.o-ran-sc.org/projects/o-ran-sc-nonrtric-rapp-orufhrecovery/en/f-release

"Helloworld" O-DU Slice Assurance use case

A very simplified closed-loop rApp use case to re-prioritize a RAN slice's radio resource allocation priority if sufficient throughput cannot be maintained. Not intended to to be 'real-world'

Information about the use case can be found at:

- Release F: O-DU Slice Assurance usecase

Repository and documentation about the service can be found at:

- https://gerrit.o-ran-sc.org/r/admin/repos/nonrtric/rapp/ransliceassurance
- https://docs.o-ran-sc.org/projects/o-ran-sc-nonrtric-rapp-ransliceassurance/en/f-release