Release E

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 applications (rApps) 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 define and coordinates rApps (Non-RT-RIC applications) to perform Non-RT-RIC tasks.
- Non-RT-RIC hosts the new R1 interface (between rApps and SMO/NONRTRIC services)

Overall objective for the E Release

In the E Release we focus mainly on studying and providing some building blocks to support the emerging Non-RT-RIC Apps ("rApps") and R1 interface concepts from O-RAN.
Support and improvement of functions for the O-RAN A1 interface continue.

E Release Priorities

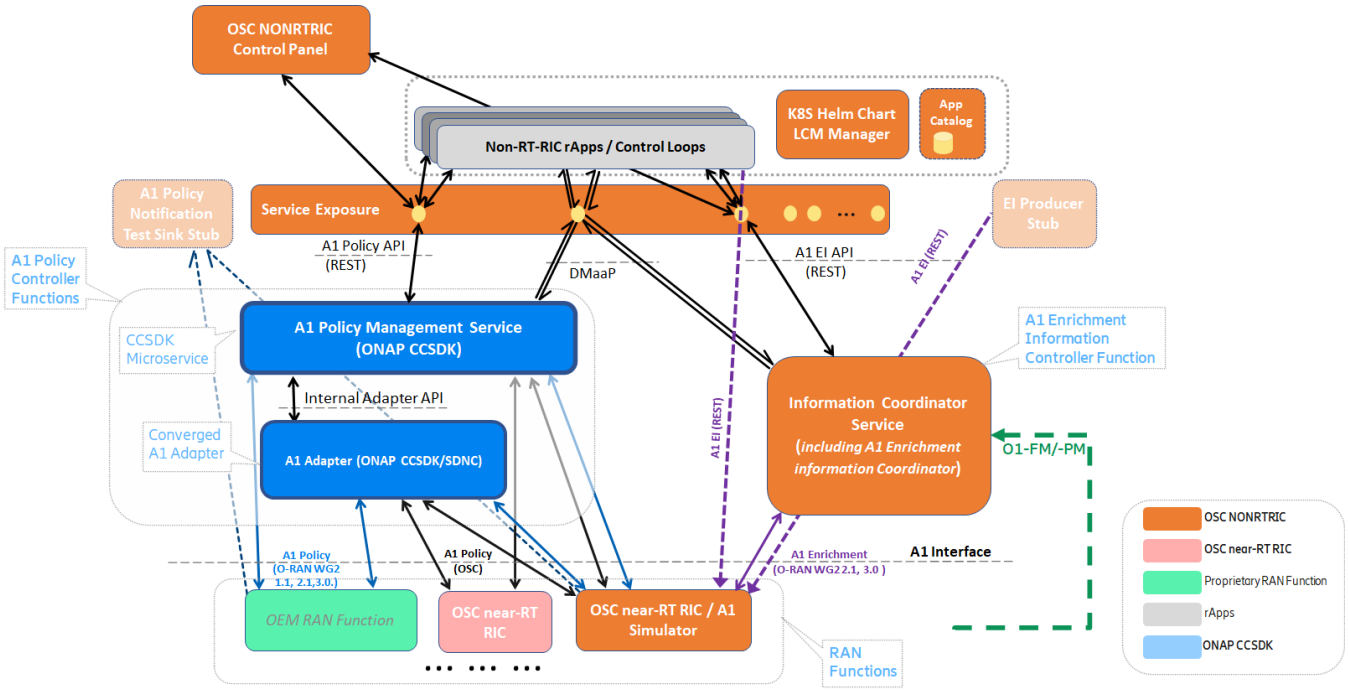
- ONAP Control Loop -> O-RAN rApp : “The *rApp-ification* of ONAP Control Loops”
  - Adopt ONAP CL work as a starting point, continue to identify gaps, then extend
  - Identify & motivate where an rApp is different from a CL
  - Types of rApps:
    - Microservice-based rApps
    - Non-Microservice-based rApps
- NONRTRIC Service Exposure/Gateway -> O-RAN R1 : “The *R1-ification* of Service Exposure”
  - Service-independent aspects
  - Types of exposure support in R1:
    - Microservice-based rApps & Service
    - Non-Microservice-based rApps & Service
- Use cases of rApps & Exposing specific Services via R1
  - Requirements drivers & demonstrators
  - O-RU FH recovery (multiple), O-DU Slice Assurance (multiple), Existing Function Tests, various other use cases in ONAP
- Continued Evolution & Support for A1 functions

E Release Feature Scope

- NONRTRIC Functions:
  - Integrated A1 Adapter from ONAP (A1 Policy (A1-P) controller – mediation)
  - Integrated A1 Policy Management Service from ONAP (A1 Policy (A1-P) controller)
  - rApp/Control Loop Manager (ONAP & OSC)
  - 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 Information Producer Mediator/Adapter (multiple)

- In E Release:
  - Deployment, Integration & Configuration– Continued improvements for Docker & Kubernetes
  - Extended/Easier deployment options with OSC IT/DEP project (SMO/NONRTRIC deployment)
  - O-RAN A1-AP evolution (v3.0)
  - Evolution of A1-EI functions to a more generic Information Coordination functions, moving beyond A1-EI
  - Further improvement in security management
  - 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 D-release
  - 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
  - OSC e2e integration use case – O-DU-HelloWorld-SliceAssurance
    - Closed loop tuning of RRM policies to assure Slice performance - via O-DU
      - Multiple implementation options – standalone µService and/or deployable ONAP-PF policy script

Architecture for Release E



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. **Near-RT RIC A1 Simulator**
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**

The code base for "E" Release is in the NONRTRIC, NONRTRIC-ControlPanel, and A1-Simulator, source repositories (Gerrit: 'e-release' 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 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:
  - Frontend: 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.

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

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 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

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 *nonrtric* script is 'Without Database'.

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

A1 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

Documentation about the service can be found at:

- https://docs.o-ran-sc.org/projects/o-ran-sc-nonrtric/en/e-release/
- https://docs.onap.org/projects/onap-ccsdk-oran/en/latest/
- A1 Policy Management Service in ONAP

Information Coordinator Service

Coordinate/Register Information Types, Producers, Consumers, and Jobs.
Coordinate/Register A1-EI Types, Producers, Consumers, and Jobs (A1 Enrichment Information Job Coordination).

- Maintains a registry of:
  - Information Types / schemas
  - Information Producers
  - Information Consumers
  - Information Jobs
- Information Query API (e.g. per producer, per consumer, per types)
- Query status of Information jobs
- After Information-type/Producer/Consumer/Job is successfully registered delivery/flow can happen directly between Information Producers and Information Consumers
- The Information Coordinator Service natively supports the O-RAN A1 Enrichment Information (A1-EI) interface, supporting coordination A1-EI Jobs where information (A1-EI)flow from the SMO/Non-RT-RIC/rApps to near-RT-RICs over the A1 interface.

Documentation about the service can be found at:

- https://docs.o-ran-sc.org/projects/o-ran-sc-nonrtric/en/e-release/

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 DMaaP and Kafka mediation:
  - Release E - Run in Docker#RuntheDmaapAdaptorServiceDockerContainer
  - https://gerrit.o-ran-sc.org/r/gitweb?p=nonrtric.git;a=tree;f=dmaap-adaptor-java;hb=refs/heads/e-release
- A version implemented in Go - Supporting DMaaP mediation:
  - Release E - Run in Docker#RuntheDmaapMediatorProducerDockerContainer
  - https://gerrit.o-ran-sc.org/r/gitweb?p=nonrtric.git;a=tree;f=dmaap-mediator-producer;hb=refs/heads/e-release

(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/e-release/

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

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

Documentation about the adapter / controller can be found at:

- https://docs.o-ran-sc.org/projects/o-ran-sc-nonrtric/en/e-release/
- https://docs.onap.org/projects/onap-ccsdk-oran/en/latest/
- CCSDK A1 Adapter for A1 Policies in ONAP

Near-RT-RIC 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

Documentation about the simulator can be found at:

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

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 E: O-RU Fronthaul Recovery usecase

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/e-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 E: O-DU Slice Assurance usecase

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/odusliceassurance;hb=refs/heads/e-release