O-Ran Software Community  
 “A” Release Requirements

Contents

[Introduction 3](#_Toc10405986)

[1. O-RAN SC Release “A” 3](#_Toc10405987)

[A. Release “A” schedule dates 3](#_Toc10405988)

[B. Target Sprint Timeline 3](#_Toc10405989)

[2. Release “A” Objective 4](#_Toc10405990)

[C. Project repos 5](#_Toc10405991)

[3. Use Case Definitions 7](#_Toc10405992)

[A. Use Case Descriptions 7](#_Toc10405993)

[B. Project use cases: 7](#_Toc10405994)

[C. Use Case Testing 8](#_Toc10405995)

[4. O-RAN Software Project Objectives 8](#_Toc10405996)

[A. Non-RT RIC: Orchestration and Automation with ONAP 8](#_Toc10405997)

[B. xAPPs 9](#_Toc10405998)

[D. O-CU 10](#_Toc10405999)

[E. O-DU 10](#_Toc10406000)

[F. O-RU 11](#_Toc10406001)

[5. External and Common Software Projects 11](#_Toc10406002)

[A. Virtualization and acceleration Layer 11](#_Toc10406003)

[B. Operations, Administrations, and Maintenance (OA&M) 11](#_Toc10406004)

[C. Common Logging and Tracing 11](#_Toc10406005)

[D. Common Software Library and Tools. 11](#_Toc10406006)

[6. Integration and Test “A” Objectives 11](#_Toc10406007)

[A. End to End O-RAN Use Case Testing 11](#_Toc10406008)

[B. Software testing should be covered by automated software testing using robot framework or other tools. https://robotframework.org/ 11](#_Toc10406009)

[7. Simulator Objectives 11](#_Toc10406010)

[A. Simulators for O-RAN components, interfaces, and messages. 11](#_Toc10406011)

[B. Simulators for external components that interoperate with O-RAN 12](#_Toc10406012)

[C. Simulators for User Equipment (UE) 12](#_Toc10406013)

[8. Addendums 12](#_Toc10406014)

[A. Contributors 12](#_Toc10406015)

[B. Definitions 13](#_Toc10406016)

[C. Near-RT RIC component details 13](#_Toc10406017)

# Introduction

O-RAN Software Community (O-RAN SC ) is a Linux Foundation Project in collaboration with the O-RAN Alliance to create an open source software solution that implements a Radio Access network implementation that aligned with O-RAN Alliance specifications.

Each release for the O-RAN SC will need to define the scope of the software for each release. Each release planning will need to set goals that advance the software but works within the limitations of the available O-ARN alliance specification, testable functions, and software development resources. Open source projects work to accomplish a solution over multiple releases, however each release should have clear objectives that advance the software projects and end to end stack objectives.

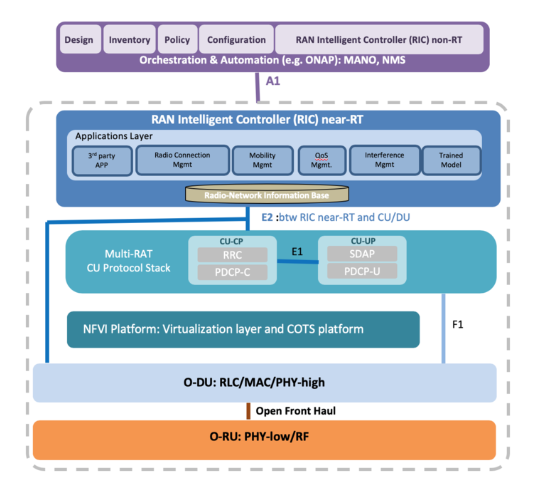


Figure 1 - O-RAN architecture

## O-RAN SC Release “A”

### Release “A” schedule dates

M0 (6/3) to M1 (6/29). This M0 involves identifying Epics, breaking down Epics into User Stories and Tasks, map Epics and US/Task to projects and dependencies, and review and finalize tasks and sub tasks in preparation for the software developers and testers to plan development sprints.

### Target Sprint Timeline

This sprint calendar is show below in Figure 2

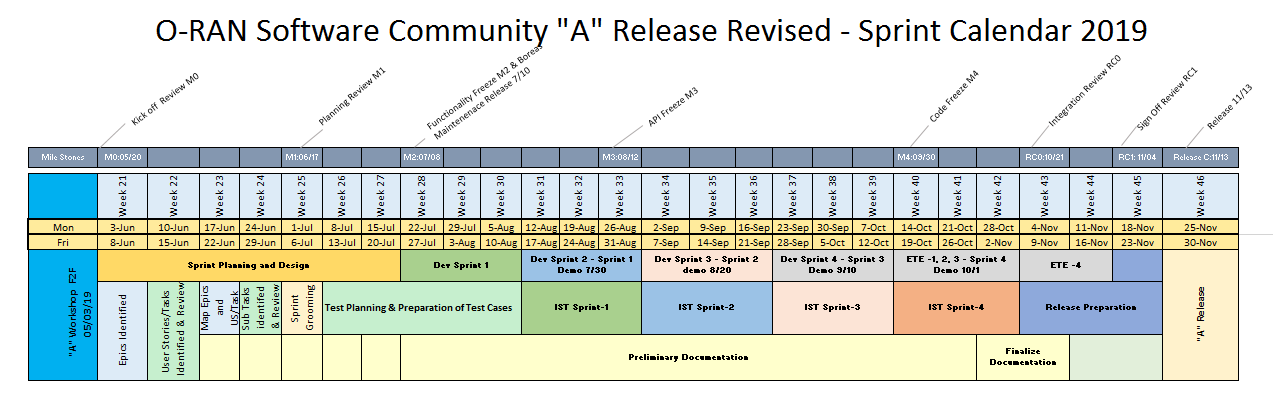


Figure 2 - Sprint calendar

## Release “A” Objective

* Demonstrate a working reference implementation of a full end-to-end O-RAN stack. The release will have some items identified as stretch goals.
  + Non-RT RIC (e.g., ONAP) implementing a basic A1 interface
  + Near-RT RIC framework implementing (see Fig. 1)
    - Application/microservices framework to host multiple xApps
    - A message bus to publish and subscribe messages inside RIC
    - A1 mediator to relay basic policies from ONAP to RIC
    - E2 Termination agent to interface with O-CU
    - R-NIB DBaaS to store RAN state information
    - E2 manager, UE manager to transact E2 data into the R-NIB
    - Dashboard to display FM and PM counters from RIC
  + A virtualization layer based on COTS platform (e.g., Akraino)
  + O-CU implementing a basic E2 interface agent
  + O-DU implementing
    - L1 functional blocks
    - L2 functional blocks
  + Open Front Haul implementation
  + O-RU

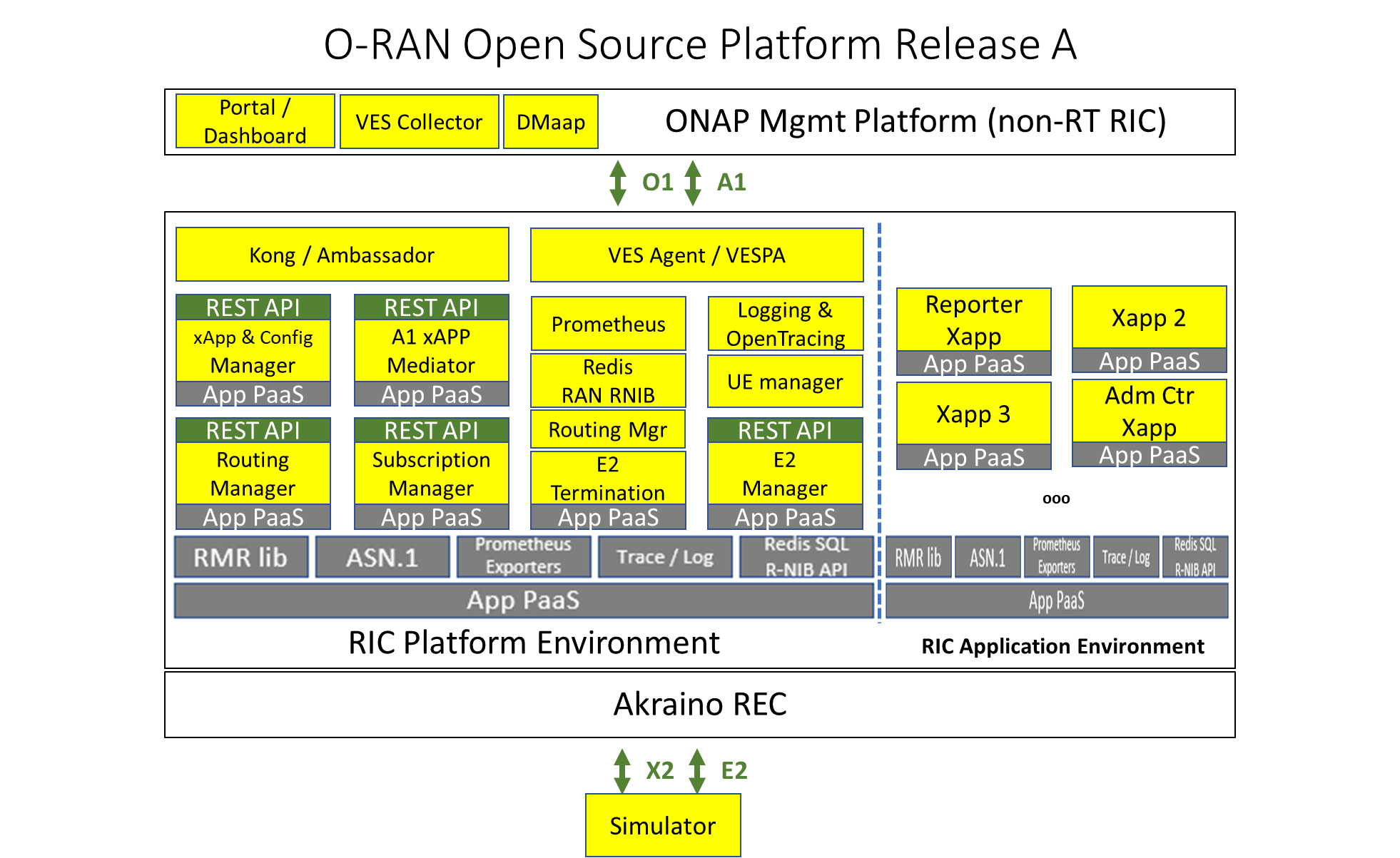


Figure 1 - near-RT RIC open source platform

Figure 1 shows an example of the near-RT RIC architecture platform deployed on a virtualized radio edge cloud (REC) infrastructure. The RIC allows xApps to be deployed and managed. The RIC also connects southbound to the RAN via E2 and northbound via A1.

### Project repos

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name | Project Key | PTL | Seed Code Repos |
| Non-realtime RAN Intelligent Controller | RICNRT | E/// |  |
|  |
| RAN Intelligent Controller Applications | RICAPP | AT&T | ric-app/admission |
| ric-app/reporter |
|  |
| RAN Intelligent Controller | RIC | Thoralf | com/asn1 |
| com/golog |
| com/log |
| ric-plt/a1 |
| ric-plt/appmgr |
| ric-plt/dbaas |
| ric-plt/dbaas/hiredis-vip |
| ric-plt/e2 |
| ric-plt/e2mgr |
| ric-plt/lib/rmr |
| ric-plt/rtmgr |
| ric-plt/sdl |
| ric-plt/sdl/config |
| ric-plt/sdlgo |
| ric-plt/sdlpy |
| ric-plt/submgr |
| ric-plt/utils |
| ric-plt/xapp-frame |
|  |
| O-RAN Central Unit | OCU | CMCC | (China Mobile integrated eNB contribution) |
|  |
| O-RAN Distributed Unit | ODU | Radisys | (Intel L1-2 contribution) |
| (Radisys L3 contribution) |
|  |
| O-RAN RU | ORU |  | Out of scope for Rel A |
| Operations Administration Management Apparatus | OAM |  | portal/ric-dashboard |
|  |
| Simulation | SIM | AT&T | (AT&T RAN side E2 simulator) |
| OTF/ |
| Infrastructure | INF | Lenovo | aal/lib |
| aal/logic |
| aal/mgmt |
| aal/virt |
| (Inspur infra monitoring contribution) |
| Akraino/blueprint |
| Integration and Testing | INT | AT&T | it/dep |
| it/dev |
| it/test |
|  |
| Documentation | DOC |  | doc |
|  |

## Use Case Definitions

The integration and test effort will focus on testing the requirements documented in each release. This will focus on end to end use case testing.

### Use Case Descriptions

The use cases are categorized into two categories, namely platform and service.

### Project use cases:

#### Dashboard/portal

|  |  |
| --- | --- |
| Title | [OAM-A-F01] RIC Dashboard |
| Description | As a RIC Operator I need an initial way to manage a single RIC instance for monitoring and control during evaluation of the RIC in a network scenario. [RICPLT-R1-F3] |
| Acceptance Criteria | User interface with connectivity to a RIC instance via pre-standard interfaces. |

|  |  |
| --- | --- |
| Title | [OAM-A-F02] RIC Connectivity Dashboard |
| Description | Enable control of connecting/disconnecting the RIC to/from a gNB (via the E2 Manager) |
| Acceptance Criteria | User Interface to add or delete gNodeB E2 endpoints to the scope of a RIC instance. |

|  |  |
| --- | --- |
| Title | [OAM-A-F03] RIC Service Assurance Dashboard |
| Description | Display metrics from each xApp (how to make this generic TBD - JSON/yang/?) |
| Acceptance Criteria | User Interface to view alarms and performance metrics collected from a RIC Instance and data received from any of its xAPPs. |

|  |  |
| --- | --- |
| Title | [OAM-A-F04] RIC API Dashboard |
| Description | Provide a method to generate control APIs for each xApp in a generic way (e.g., yang definition of the interface) |
| Acceptance Criteria | User Interface to compose commands or data to exposed RIC APIs. |

|  |  |
| --- | --- |
| Title | [OAM-A-F05] O-RAN-SC Certification Dashboard |
| Description | Provide a Portal for viewing previously executed tests and to visually show current tests in progress. |
| Acceptance Criteria | User Interface to view test already completed, in progress, or scheduled to be executed. |

#### Non-RT RIC (RPGF/ONAP)

Verify ONAP Deploy (VES Collector, DMaap)

Verify Policy insertion to near-RT RIC via A1

#### Near-RT RIC

|  |  |
| --- | --- |
| Title | [RIC-A-F01] RIC Connects to gNodeB |
| Description | [RICPLT-R1-F1] RIC shall control gNB and eNB using E2 interface |
| Acceptance Criteria |  |
| Status | A release or in seed code |

|  |  |
| --- | --- |
| Title | [RIC-A-F02] RIC functions are guided through A1 |
| Description | [RICPLT-R1-F2] RIC shall be managed through NB interface (A1) |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F03] "Big Red Button" |
| Description | [RICPLT-R1-F4] Provide the ability to quick disconnect RIC from RAN during production market trials in case harm is detected. |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F04] Provide Data Storage Abstraction Layer |
| Description | [RICPLT-R1-E1] DBaaS/SDL/Redis |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F05] Provide RIC Inter-module communications |
| Description | [RICPLT-R1-E2] RMR |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F06] Provide Standardized logging library |
| Description | Logging (REC already can ship outside, we might re-use that) |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F07] Support event trace reporting |
| Description | Tracing - support for openTracing |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F08] Provide High-Available Architecture |
| Description | HA (DBaaS) |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F09] Provide DBaaS API for multiple development languages |
| Description | DB libraries for other language |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F10] Recovery after connection failure restoral |
| Description | All component should be restart automatically and reconnect |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F11] Rainy Day Interface Handling |
| Description | Retry strategy/policy |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F12] Initial Security Mechanisms |
| Description | Security? Probably minimal, but to align with potential requirements from PIZ document,i.e. read the PIZ document from security point of view |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F13] Initial Performance Metrics |
| Description | Metrics |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F14] Improve Component Implementation Performance |
| Description | Performance area (let's see where we are at end of R0) - do we need to improve latencies? |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F15] Provide Platform Healthcheck |
| Description | Something in area of testability |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F16] Map 3GPP Messages to IDs |
| Description | Provide a general way to handle and manage RIC message types (each type maps to an integer) that is aligned with 3GPP message types where appropriate. Right now (R0) message types are constants that are defined with integer values. However, there is a need to use symbolic/string names for the message types in the Dashboard and xApp Manager. Initial though: a more general (e.g., JSON) descriptor that can be translated into language specific header files. |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F16] Support IPv6 in the E2 Manager |
| Description | Support IPv6 address for the gNB (E2 Manager) |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F17] Support IPv6 in the E2 Term |
| Description | Support IPv6 address for the gNB (E2 Term) |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F18] Maintaining Message Routing after Recovery |
| Description | If an xApp fails and recovers (thanks to K8s) and its IP address changes, call Routing Manager to update the routing rules (TBD - maybe using "service name" in routing rules is sufficient to handle failure and recovery) |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F19] Maintain Subscriptions |
| Description | Receive subscription based routing updates from Subscription Manager. |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F20] Connect xAPP to Network Elements |
| Description | Populate initial routes for an xApp when it is deployed |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F21] Provide RAN to E2 Term Mapping |
| Description | Support routing to the correct E2 Termination instance based on RAN id |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F22] Persist Routing for Recovery/Restart |
| Description | Persist routing information in DB |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F24] Restore Routing after Restart or Recovery |
| Description | Handle Routing Manager restart after failure |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F25] Identify A1 messages for routing |
| Description | Add "subscription id" is the message header. |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F26] Use REST GW to selectively expose APIs |
| Description | Include Kong API gateway (or some other API gateway). No authentication yet. |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F27] RIC supports A1 Mediation |
| Description | Provide ability to route A1 messages received from the NMS to the appropriate message processing application (xAPP). |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F28] Provide NBI for configuring an xAPP |
| Description | Provide an API to the Ops Portal (REST or something different) |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F29] Support xAPP config prior to xAPP launch |
| Description | When receive a configuration update message, populate the payload in the right Redis namespace. |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F30] Provide persistent store of xAPP Config |
| Description | Possibly further steps needed after data is in Redis (e.g. ConfigMap creation) |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F31] Ability to dynamically re-assert container configmap |
| Description | ConfigMap creation on request by xApp. |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F32] xAPP needs to be able to receive RAN Messages |
| Description | Receive E2 Subscription message from RMR, compare the subscription to existing subscriptions based on the targeted gNB (from RMR header) and the E2 Subscription message payload. |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F33] RIC supports control loop latency measures |
| Description | Collect the latency for each control loop execution, report min/max/avg latency for each control loop via the Prometheus metrics collection |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F34] Provide Intra-RIC Tracing |
| Description | RMR instrumentation for open-tracing (depends on E2 termination to start using it) – Dependency on other teams to actually include open-trace support in RMR (for RMR: Espoo 2 writes prototype and demonstrate usage and AT&T platform team then integrates that into RMR formally), E2 termination, one xApp (Nokia), the open-source xApp. |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F35] RIC supports fault reporting |
| Description | RIC must emit faults |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F36] RIC as an Appliance (PNF) supports platform reporting |
| Description | Enable measuring and reporting on platform statistics |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F37] RIC needs to be able to report on Infrastructure, Platform, and Applications independently. |
| Description | Support different granularities (apps, platform) |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F38] Support O1 performance reporting |
| Description | Support platform performance metrics |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F39] RIC must support transactional logging |
| Description | Infrastructure must conform to logging guidelines (RIC, xApp, RPGF) |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F40] RIC supports reporting through NBI |
| Description | Enable the RIC to transmit the telemetry over the prescribed interface (A1/O1) |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RIC-A-F41] The RIC as a platform needs to be independently testable. |
| Description | Develop Test Strategy for the RIC Platform |
| Acceptance Criteria |  |

#### RIC xApps

|  |  |
| --- | --- |
| Title | [RICAPP-A-F10] Support xApp policy guidance using A1 messages |
| Description | As an xApp service I need to be able to receive and apply policy guidance sent over the A1 interface, and to update the R-NIB database accordingly. |
| Acceptance Criteria | Demonstrate an ability to receive policy guidance sent over the A1 interface, make changes accordingly, and observe behavior by the xApp consistent with the sent policy guidance |

|  |  |
| --- | --- |
| Title | [RICAPP-A-F20] Support xApp configuration updates using O1 messages |
| Description | As an xApp service I need to be able to receive and apply configuration updates sent over the O1 interface, and to update R-NIB database accordingly. |
| Acceptance Criteria | Demonstrate an ability to receive configuration updates sent over the O1 interface, make changes accordingly, and observe behavior by the xApp consistent with the sent configuration updates |

|  |  |
| --- | --- |
| Title | [RICAPP-A-F30] Support xApp to generate and emit performance metrics on the A1 interface |
| Description | As the xApp service I need to be able to (a) generate data on performance metrics and (b) capture information on actions taken, and to emit these over the A1 interface in a way that Prometheus can collect them, in order to be able to monitor the performance of the xApp |
| Acceptance Criteria | Ability of Prometheus to capture and process xApp performance metrics |

|  |  |
| --- | --- |
| Title | [RICAPP-A-F40] Support xApp to generate and emit performance metrics on the O1 interface |
| Description | As the xApp service I need to be able to (a) generate data on performance metrics and (b) capture information on actions taken, and to emit these over the O1 interface in a way that Prometheus can collect them, in order to be able to monitor the performance of the xApp |
| Acceptance Criteria | Ability of Prometheus to capture and process xApp performance metrics |

|  |  |
| --- | --- |
| Title | [RICAPP-A-F50] Support xApp actions logging |
| Description | As an xApp service I need to be able to capture OA&M events into logs for debugging, error handling, identification of performance bottlenecks and to meet ASPR requirements following stated logging formats and guidelines, and to make these logs available for transport and analysis to the underlying hosting system |
| Acceptance Criteria | Demonstrate an ability to capture information into logs in a standardized format, and to make these logs available to the underlying hosting system for further disposition |

|  |  |
| --- | --- |
| Title | [RICAPP-A-F60] Develop Test Strategy for an xAPP |
| Description |  |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title | [RICAPP-A-F70] E2 Subscribe message |
| Description | As an xApp service I need to be able to construct and send E2 Subscription message as follows: Allocate an RMR message, construct the E2 subscription message as the RMR message payload (ASN.1 encoded), populate the target gNB in the RMR header ("managed element"), populate the message type as RIC\_E2\_SUBSCRIPTION and send the message. |
| Acceptance Criteria | Demonstrate the ability to generate and send an E2 subscribe message to a CU or DU and have it take effect |

|  |  |
| --- | --- |
| Title | [RICAPP-A-F80] E2 Unsubscribe message |
| Description | As an xApp service I need to be able to construct and send E2 Unsubscribe message as follows: Allocate an RMR message, construct the E2 subscription message as the RMR message payload (ASN.1 encoded), populate the target gNB in the RMR header ("managed element"), populate the message type as RIC\_E2\_UNSUBSCRIPTION and send the message. |
| Acceptance Criteria | Demonstrate the ability to generate and send an E2 unsubscribe message to a CU or DU and have it take effect |

|  |  |
| --- | --- |
| Title | [RICAPP-A-F90] Support xApp API transaction tracing |
| Description | As an xApp service I need to be able to support API transaction tracing to support API transition use and calling patterns using the OpenTracing tool |
| Acceptance Criteria | Demonstrate the ability to issue a call to an xApp and have that call and any related calls show up in a OpenTracing call trace |

|  |  |
| --- | --- |
| Title | [RICAPP-A-F100] Support xApp-generated faults |
| Description | As the xApp service I need to be able to generate faults (alarms) on error conditions and emit these over the O1 interface in a way that Prometheus (?) can collect them, in order to be able to monitor the condition of the xApp |
| Acceptance Criteria | Ability of Prometheus to capture and process xApp performance metrics (?) |

|  |  |
| --- | --- |
| Title |  |
| Description |  |
| Acceptance Criteria |  |

RPGF: RIC Policy Guidance Function (nee non-RT RIC)

|  |  |
| --- | --- |
| Title | [RICNRT-A-F10] Enable the RPGF to send policy guidance to the xApp via the A1 interface |
| Description | As a Service Management and Orchestrator I need to be able to determine appropriate policy guidance set of values that needs to be applied to a given instance of an xApp, and to send these values via the A1 interface in order to update the policies in use by that xApp. |
| Acceptance Criteria | Acceptance criteria is to demonstrate an ability to generate policy guidance values, send these over the A1 interface to a target set of xApps, have these xApps make changes accordingly to the values, and observe behavior by the xApps consistent with the sent policy guidance |

|  |  |
| --- | --- |
| Title | [RICNRT-A-F20] Enable the RPGF to receive and process telemetry from an xApp sent via the A1 interface |
| Description | As a Service Management and Orchestrator I need to be able to receive, process and potentially store telemetry sent over the A1 interface, |
| Acceptance Criteria | Demonstrate an ability to receive telemetry sent over the A1 interface, make changes accordingly, and observe behavior by the RPGF consistent with the received telemetry |

|  |  |
| --- | --- |
| Title | [RICNRT-A-F30] Enable the RPGF to collect and process telemetry from an xApp sent via the O1 Interface |
| Description | As a Service Management and Orchestrator I need to be able to receive, process and potentially store telemetry sent over the O1 interface, |
| Acceptance Criteria | Demonstrate an ability to receive telemetry sent over the O1 interface, make changes accordingly, and observe behavior by the RPGF consistent with the received telemetry |

|  |  |
| --- | --- |
| Title | [RICNRT-A-F40] Support an extensible RPGF Platform |
| Description | As a Service Management and Orchestrator I need to be able to support an extensible RPGF platform that can support the dynamic addition, (re)configuration, (re)activation, execution, deactivation, and removal of a discrete sets of functionality packaged as discrete units (containers) |
| Acceptance Criteria | Demonstrate the ability to add, (re)configure, (re)activate, execute, deactivate and remove a discrete units of functionality |

|  |  |
| --- | --- |
| Title | [RICNRT-A-F50] Develop Test Strategy for the Non-RT RIC |
| Description |  |
| Acceptance Criteria |  |

|  |  |
| --- | --- |
| Title |  |
| Description |  |
| Acceptance Criteria |  |

Platform

Common logging

RMR

SDL/DBaaS

RNIB query library

O1

* Verify RIC platform Deploy
* Verify Heartbeat for platform components
* Verify PM/Metrics Streaming
* Verify Bulk PM
* Verify Trace Collection
* Verify Log collection
* Verify xAPP Configure/Reconfigure
* Verify xApp Deploy/ Undeploy
* Verify Fault Reporting

E2

* Verify E2 Setup Request
* Verify E2 Reset
* Verify E2 Indication
* Verify E2 Subscribe Report
* Verify E2 Subscribe Delete

A1

* A1 mediator

#### xApps

* Admission Control xApp
* Measurement, MCxapp

#### O-CU

#### O-DU

B.2 End-to-end use cases: [TBD]

Verify a simple end-to-end service demonstrating

Report subscription and KPI computation; Dashboard visualization;

Load balancing or policy based HO for select UE, etc. …

### Use Case Testing

## O-RAN Software Project Objectives

The O-RAN software community will sponsor projects and supply repos and build tools for projects supporting O-RAN Alliance aligned Radio Access Network (RAN) solution.

Each project will require a set of requirements that can drive Epics. The Epics are provide the goals for the software developers and testers. User stories are used to drive the software development team’s dev cycles. The epics and user stories all role up the high level use cases and are validated by the integration and test team.

### Non-RT RIC: Orchestration and Automation with ONAP

A project should focus on use cases supported in ONAP. The work should be sponsored in O-RAN and initial code contributions in O-RAN SC. Interoperability with ONAP should be aligned between the standards and will follow initial contributions in this project.

Non-RT RIC Functional Requirements

* ORAN contributions to support A1 and O1 interface via SDN-R; The non-RT RIC will primarily include A&AI, SDN-R and DCAE
* Non-RT RIC shall support data retrieval and analysis; the data may include performance, configuration or other data related to the application
* Non-RT RIC shall support performance monitoring and evaluation

A1 Interface Functional Requirements

* A1 interface shall support communication of policies/intents from Non-RT RIC to Near-RT RIC
* A1 interface shall support communication of enrichment information from Non-RT RIC to Near-RT RIC.
* A1 interface shall support the policy/intents feedback from Near-RT RIC to non-RT RIC

Non-RT RIC Non-Functional Requirements

* Non-RT RIC shall not update the same policy or configuration parameter for a given near-RT RIC or RAN function more often than once per second
* Non-RT RIC shall be able to update policies in several near-RT RICs (stretch) Non-RT RIC:
  + Deploys xApps in near-RT RIC;
  + Provides a dashboard to view time series of KPIs computed from measurement data obtained via E2

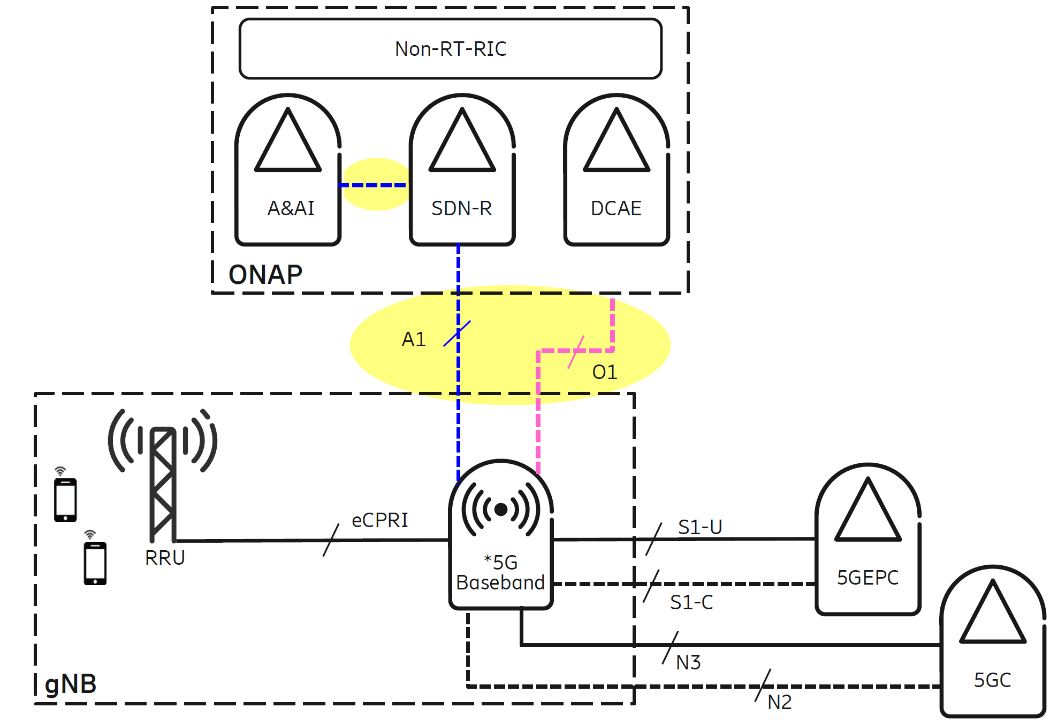


Figure 3 - A1 interface example

### xAPPs

xApps should be targeted for Release “A”. This includes open source sample xAPPs and platform applications that can we used for integration, testing, and demonstrations.

#### Sample xAPP – Admission Control xApp to allow basic admission control functionality of UEs based on a simple policy (e.g., time of day)

#### Sample xApp – Reporting xApp that computes KPIs from RNIB data (e.g., UE throughput, cell load) and provides a dashboard view in non-RT RIC

#### Sample platform application – UE Manager platform application to read E2 messages and extract specific information elements (IEs) into the near-RT RIC database (RNIB). The RNIB reflects the most recent RAN information state.

1. The Near RT-RIC should have a focus for this release. An initial framework should be put in place with support for xAPP development and operation.

#### The CU should target an initial software deliverables with limited functionality. This would be included as part of the integration and test plans.

* Near-RT RIC shall support interpreting policies and intents
* Near-RT RIC shall support deployment and execution of xApps; Life cycle management (LCM) of Xapps (e.g., start/stop/deploy/undeploy)
* Near-RT RIC platform should provide an up to date, consistent, and query-able view of UE and RAN config, context, state, and performance data to xApps running in the RIC; The RIC has a Radio Network Information Base (RNIB) which holds an accurate, updated information of the RAN (including GNB, ENB etc) state. The RNIB is a part of the overall RIC database / persistent storage which serves the needs of the various RIC and xAPP functions. We assume that whenever there is an update in the RAN state, the RIC receives REPORT messages from the RAN over E2, and this information is processed by an E2 Manager xAPP which updates the RNIB
* Near-RT RIC platform should support xApp execution with input data, ML model inference, and policy execution, to generate the corresponding RRM enforcement decisions.
* Near-RT RIC platform should be able to meet the necessary latency requirements for xApp execution (e.g., 10 msec)

### O-CU

The O-RAN CU is targeted for Release “A”.

#### The O-CU should recruit seed code contributions from contributors.

#### The O-CU should target an initial software deliverable with limited functionality.

#### O-CU shall implement a basic E2 interface. This would be included as part of the integration and test plans.

### O-DU

The O-RAN DU is targeted for Release “A”.

#### The DU should recruit seed code contributions from contributors.

#### The DU should target a proof of concept leveraging seed code contributions.

### O-RU

The O-RAN RU is not the focus of Release “A”.

#### The RU should focus plans for future releases.

## External and Common Software Projects

This chapter provides dependencies on other open source projects (Akraino, Acumos, ONAP, etc.) that need to be leveraged to realize an end-to-end ORAN deployment.

### Virtualization and acceleration Layer

### Operations, Administrations, and Maintenance (OA&M)

### Common Logging and Tracing

* Develop a standard logging mechanism

### Common Software Library and Tools.

* Develop an open source ASN.1 library

## Integration and Test “A” Objectives

The integration and test effort will focus on testing the requirements documented in each release. This will focus on end to end test and use case testing.

### End to End O-RAN Use Case Testing

### Software testing should be covered by automated software testing using robot framework or other tools. <https://robotframework.org/>

## Simulator Objectives

Simulators are very important to O-RAN Software Community. Developers and testers need simulators to allow the teams to work on parts of the software without having to have all part available. This is very important when RF interfaces and User Equipment (UE) interoperation are required.

In early releases simulators are very important to allow partial implementations of O-RAN interfaces and modules. The needs are to allow software to be developed with a reasonable level of isolation or until other software can be completed.

### Simulators for O-RAN components, interfaces, and messages.

#### E2 interface similar is needed for near RT-RIC and xAPPs. The E2 interface connects the RIC to the RAN Control Plane.

The roles of the E2 interface are:

* To manage the SCTP connections to the RAN E2 interface
* To manage the connection lifecycle of the RAN (CRUD) using the XSetup xApp
* To receive/send E2 messages from and to the RAN devices over SCTP
* To strip the E2 protocol envelope from received messages and extract the underlying RAN E2 Service Model containers for the xApps
* To send the RAN E2 Service Model containers to the xApps that we registered to receive them using the UTA library
* To receive RAN E2 Service Model containers from the Xapps using the UTA library wrap them with E2 wrapper and send them to the relevant E2 interface
* To take care of re-transmissions, failures, timeouts, etc

The set of E2 operations that need to be supported by the gNB simulator are

| Initiated by | Elementary Procedure | Initiating Message | Successful Outcome | Unsuccessful Outcome |
| --- | --- | --- | --- | --- |
| Response message | Response message |
| RIC | RIC Subscription (r1) | RIC SUBSCRIPTION REQUEST (r1) | RIC SUBSCRIPTION RESPONSE (r1) | RIC SUBSCRIPTION FAILURE(>r2) |  |
| RIC | RIC Subscription delete (r1) | RIC SUBSCRIPTION DELETE REQUEST (r1) | RIC SUBSCRIPTION DELETE RESPONSE (r1) | RIC SUBSCRIPTION DELETE FAILURE(>r2) |  |
| RIC or RAN | E2 Reset (r1) |  |  |  |  |
| RIC | E2 Setup (r1) | [EN-DC]/- X2 SETUP REQUEST | [EN-DC]/- X2 SETUP RESPONSE | [EN-DC]/- X2 SETUP FAILURE |  |
| RIC | Resource Status Reporting Initiation (r2) | RESOURCE STATUS REQUEST | RESOURCE STATUS RESPONSE | RESOURCE STATUS FAILURE |  |

| Initiated by | Elementary Procedure | Initiating Message | Which xApp? |
| --- | --- | --- | --- |
| RAN | RIC Indication | RIC INDICATION (r1) |  |
| RAN | Resource Status Reporting | RESOURCE STATUS UPDATE (r1) |  |
| RAN or RIC | Error Indication | ERROR INDICATION (r1) |  |

Legend: r1 is targeted for A release, r2 is B release

### Simulators for external components that interoperate with O-RAN

#### Not Required

### Simulators for User Equipment (UE)

#### Not Required

## Addendums

There will be different sections to support the document and its objectives.

### Contributors

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Company | Contact Info |  |
| John Murray | AT&T | jfm@research.att.com | 5/5 |
| Rittwik Jana / David Kinsey | AT&T | [rjana@research.att.com](mailto:rjana@research.att.com); dk8126@att.com | 5/20 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### Definitions

|  |  |
| --- | --- |
| Term | Description |
| Not Required | This release does not require any addition work on this topic for this release. The current software may be adequate or additional requirements may appear in the future. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

### Near-RT RIC component details

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Is docker container /binary?** | **Component** |  | **Description** |
| **Near-RT  RIC Platform** |  | xApp Manager |  | Manages xApps |
|  | Configuration Manager |  |  |
|  | Routing Manager |  |  |
| yes | E2 Termination |  | Setup E2 connection between RIC and gNB |
| yes | E2 Manager |  | Manages E2 connections |
| yes | DBaaS (Redis instantiation) |  | R-NIB |
| no, library | ASN.1 encoding/decoding library |  | Message library |
| no, library | AppPaaS - SDL (Shared data layer) |  | R-NIB database abstraction layer |
| no, library | AppPaas - Logging |  |  |
| no, library | AppPaaS - message library and routing table |  |  |
| no, library | OpenTracing |  |  |
|  | A1 Mediator |  | A1 interface impl. |
|  | VES agent |  | Collects PM data |
|  | UE manager |  |  |
|  | Subscription manager |  |  |
|  | DBaSS |  |  |
| **Aux** |  | VES collector |  |  |
|  | DMaaP |  |  |
| yes,in ONAP | RIC Dashboard |  |  |
|  | Nexus Repo for deployment |  |  |
| **xApps** | yes | Admission Control xApp |  |  |
| yes | Measurement reporting xApp |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Domain** | **Idea, Epic or User Story?** | **Original Section** | **Original WIKI Sub-Section / Prior ID** | **Epic (N2=0) / User story (N2>0) / Idea (N<0)** |
| CM | Epic | R1 PI | R1 Features and Enablers | **[RICPLT-R1-F4] "Big Red Button"** |
| CM | Epic | R1 Scope | Any xApp (general requirements) | 7. Support Configuration management via configuration management framework (including re-reading configuration via k8s configMaps and using inotify() for changes - however, so far no xApp parameter identified that would have to be dynamically modified. E.g. we could have a parameter that modifies the semantics of group A and group B). |
| CM | Epic | R1 Scope | Any xApp (general requirements) | Note: Generic parts of CM pipeline must be a separate feature (under A1/O1 below) |
| CM | Epic | R1 Scope | Configuration Management Framework for ConfigMaps | 1. It shall be possible to instantiate xApps with different configuration (enough if same xApps type is only instantiated once) |
| CM | Epic | R1 Scope | Configuration Management Framework for ConfigMaps | 2. It shall be possible to update configuration of xApps while it is running. |
| CM | Epic | R1 Scope | O1 Mediator (TBD) | 1. Provide an API to the Ops Portal (REST or something different) |
| CM | Epic | R1 Scope | O1 Mediator (TBD) | 2. When receive a configuration update message, populate the payload in the right Redis namespace. |
| CM | Epic | R1 Scope | O1 Mediator (TBD) | 3. Possibly further steps needed after data is in Redis (e.g. ConfigMap creation) |
| CM | Epic | R1 Scope | O1 Mediator (TBD) | 4. ConfigMap creation on request by xApp. |
| Dashboard | Epic | R1 Scope | Top-down approach to defining the scope | **[[RICPLT-R1-F3] RIC Dashboard](https://rancodev.atlassian.net/wiki/spaces/CM/pages/36733069/%5BRICPLT-R1-F3%5D+RIC+Dashboard)** |
| Dashboard | Epic | R1 Scope | RIC Dashboard | 1. Enable control of connecting/disconnecting the RIC to/from a gNB (via the E2 Manager) |
| Dashboard | Epic | R1 Scope | RIC Dashboard | 2. Display metrics from each xApp (how to make this generic TBD - JSON/yang/?) |
| Dashboard | Epic | R1 Scope | RIC Dashboard | 3. Provide a method to generate control APIs for each xApp in a generic way (e.g., yang definition of the interface) |
| Demos | Epic | R1 Scope | "E2E components": | 3. Demos |
| gNB | Epic | R1 Scope | E2 Manager | 1. Support IPv6 address for the gNB |
| gNB | Epic | R1 Scope | E2 Termination | 1. Support IPv6 address for the gNB |
| LCM | Epic | R1 Scope | xApp Manager | 1. If an xApp fails and recovers (thanks to K8s) and its IP address changes, call Routing Manager to update the routing rules (TBD - maybe using "service name" in routing rules is sufficient to handle failure and recovery) |
| LCM | Epic | R1 Scope | Integration | 1. CI: |
| LCM | Epic | R1 Scope | Integration | 2. “One-click” improvements |
| LCM | Epic | R1 Scope | Integration | 3. xApp descriptor |
| LCM | Epic | R1 Scope | Integration | 5. Resource manager |
| LCM | Epic | R1 Scope | Integration | 6. Networking |
| LCM | Epic | R1 Scope | Integration | 7. HA design |
| Logging | Epic | R1 Scope | Non-Functional Requirements | 1. Logging (REC already can ship outside, we might re-use that) |
| Logging | Epic | R1 Scope | Non-Functional Requirements | 2. Tracing - support for openTracing |
| Logging | Epic | R1 Scope | Any xApp (general requirements) | 9. Support tracing via OpenTracing (at least one xApp in R1) |
| Logging | Epic | R1 Scope | VESPA (VES Agent that takes information from Prometheus and sends it to VES Collector in VES format): | 1. Whatever needs to be done to get the desired metrics to the VES Collector (requirements for xApps, Prometheus) |
| Logging | Epic | R1 Scope | Logging | 1. Python library for logging |
| Logging | Epic | R1 Scope | Open-tracing | 1. RMR instrumentation for open-tracing (depends on E2 termination to start using it) – Dependency on other teams to actually include open-trace support in RMR (for RMR: Espoo 2 writes prototype and demonstrate usage and AT&T platform team then integrates that into RMR formally), E2 termination, one xApp (Nokia), the open-source xApp. |
| Logging | Epic | R1 Scope | Integration | 9. Logging support: |
| N/A | Epic | R1 Scope | Non-Functional Requirements | 6. Retry strategy/policy |
| N/A | Epic | R1 Scope | Non-Functional Requirements | 8. Metrics |
| N/A | Epic | R1 Scope | Non-Functional Requirements | 9. Performance area (let's see where we are at end of R0) - do we need to improve latencies? |
| N/A | Epic | R1 Scope | Non-Functional Requirements | 10. something in area of testability? Bharath to consider |
| N/A | Epic | R1 Scope | Component-specific requirement listing (tentative - work in progress) | < Section Header> |
| N/A | Epic | R1 Scope | Configuration Management Framework for ConfigMaps | Suggestion: Espoo 1 (not yet discussed with Espoo 1 team) |
| OTF | Epic | R1 Scope | "E2E components": | 1. Testing: |
| Persistence | Epic | R1 Scope | Top-down approach to defining the scope | **[RICPLT-R1-E1] DBaaS/SDL/Redis** |
| Persistence | Epic | R1 Scope | Non-Functional Requirements | 3. HA (DBaaS) |
| Persistence | Epic | R1 Scope | Non-Functional Requirements | 4. DB libraries for other language |
| Persistence | Epic | R1 Scope | DBaaS/SDL/Redis: | 1. Provide higher-level database access operations as a Golang library for xApp writers and for E2 Termination and/or E2- Manager writers. TODO-decide data format (Google protobuf) and how versioning is handled. |
| PM | Epic | R1 Scope | Any xApp (general requirements) | 1. Export xApp specific metrics in a way that Prometheus can collect them (mechanism to be determined) |
| PM | Epic | R1 Scope | Any xApp (general requirements) | 8. Support metrics of the xApp by providing a Prometheus data provider. In R1 we must have at least one xApp and one RIC component (e.g. E2 termination/E2 manager) providing metrics. |
| PM | Epic | R1 Scope | Resource Manager: | 1. Collect the latency for each control loop execution, report min/max/avg latency for each control loop via the Prometheus metrics collection |
| RIC | Epic | R1 Scope | Top-down approach to defining the scope | **[[RICPLT-R1-F1] RIC shall control gNB and eNB using E2 interface](https://rancodev.atlassian.net/wiki/spaces/CM/pages/36733032/%5BRICPLT-R1-F1%5D+RIC+shall+control+gNB+using+E2+interface)** |
| RIC-A1 | Epic | R1 Scope | Top-down approach to defining the scope | **[[RICPLT-R1-F2] RIC shall be managed through NB interface (A1)](https://rancodev.atlassian.net/wiki/spaces/CM/pages/36601978)** |
| RIC-MB | Epic | R1 Scope | Top-down approach to defining the scope | **[RICPLT-R1-E2] RMR** |
| RIC-MB | Epic | R1 Scope | RIC "Platform" (common services) | 1. Provide a general way to handle and manage RIC message types (each type maps to an integer) that is aligned with 3GPP message types where appropriate. Right now (R0) message types are constants that are defined with integer values. However, there is a need to use symbolic/string names for the message types in the Dashboard and xApp Manager. Initial though: a more general (e.g., JSON) descriptor that can be translated into language specific header files. |
| RIC-MB | Epic | R1 Scope | Routing Manager | 1. Receive subscription based routing updates from Subscription Manager. |
| RIC-MB | Epic | R1 Scope | Routing Manager | 2. Populate initial routes for an xApp when it is deployed |
| RIC-MB | Epic | R1 Scope | Routing Manager | 3. Support routing to the correct E2 Termination instance based on RAN id |
| RIC-MB | Epic | R1 Scope | Routing Manager | 4. Persist routing information in DB |
| RIC-MB | Epic | R1 Scope | Routing Manager | 5. Handle Routing Manager restart after failure |
| RIC-MB | Epic | R1 Scope | RMR | 1. Add "subscription id" is the message header. |
| RIC-MB | Epic | R1 Scope | A1 Mediator | 1. Receive REST calls and translate them to RMR calls and/or Redis call. |
| RIC-MB | Epic | R1 Scope | A1 Mediator | 2. Include Kong API gateway (or some other API gateway). No authentication yet. |
| RIC-MB | Epic | R1 Scope | A1 Mediator | Note: Competing proposals from Nokia and AT&T (RMR or Redis based). Also an open question if the A1 Mediator is xApp specific (i.e., each xApp has one of its own) or shared. If we agree on the north-bound interface, all options may be supportable. |
| RIC-MB | Epic | R1 Scope | Subscription Manager | 1. Receive E2 Subscription message from RMR, compare the subscription to existing subscriptions based on the targeted gNB (from RMR header) and the E2 Subscription message payload. |
| Security | Epic | R1 Scope | Non-Functional Requirements | 7. Security? Probably minimal, but to align with potential requirements from PIZ document,i.e. read the PIZ document from security point of view |
| Security | Epic | R1 Scope | Integration | 8. Security design |
| Security | Epic | R1 Scope | STILL OPEN-  TODO - security | Probably minimal, but to align with potential requirements from PIZ document,i.e. read the PIZ document from security point of view |
| Simulators | Epic | R1 Scope | "E2E components": | 2. Simulators |
| xApp | Epic | R1 Scope | Any xApp (general requirements) | 2. Construct and send E2 Subscription message as follows: Allocate an RMR message, construct the E2 subscription message as the RMR message payload (ASN.1 encoded), populate the target gNB in the RMR header ("managed element"), populate the message type as RIC\_E2\_SUBSCRIPTION and send the message. |
| xApp | Epic | R1 Scope | Any xApp (general requirements) | 3. Construct E2 Unsubcsribe messages the same way. |
| xApp | Epic | R1 Scope | Any xApp (general requirements) | 4. Process E2 Subscription response as follows: .. |
| xApp | Epic | R1 Scope | Any xApp (general requirements) | 5. ... |
| xApp | Epic | R1 Scope | Any xApp (general requirements) | 6. If xApp support policy guidance (A1) or configuration (O1) updates, implement the necessary mechanism to receive the updates: |
| xApp | Epic | R1 Scope | Integration | 4. Maintain the requirement documents (HW, SW) |