O-RAN.WG3.TS.RICARCH-R004-v07.00

*Technical Specification*

**O-RAN Work Group 3 (Near-Real-time RAN Intelligent**

**Controller and E2 Interface) Near-RT RIC Architecture**

Copyright © 2025 by the O-RAN ALLIANCE e.V.

The copying or incorporation into any other work of part or all of the material available in this specification in any form without the prior written permission of O-RAN ALLIANCE e.V. is prohibited, save that you may print or download extracts of the material of this specification for your personal use, or copy the material of this specification for the purpose of sending to individual third parties for their information provided that you acknowledge O-RAN ALLIANCE as the source of the material and that you inform the third party that these conditions apply to them and that they must comply with them.

O-RAN ALLIANCE e.V., Buschkauler Weg 27, 53347 Alfter, Germany Register of Associations, Bonn VR 11238, VAT ID DE321720189

Contents

[Foreword 5](#_TOC_250113)

[Modal verbs terminology 5](#_TOC_250112)

1. [Scope 6](#_TOC_250111)
2. [References 6](#_TOC_250110)
   1. [Normative references 6](#_TOC_250109)
   2. [Informative references 7](#_TOC_250108)
3. [Definition of terms, symbols and abbreviations 7](#_TOC_250107)
   1. [Terms 7](#_TOC_250106)
   2. [Symbols 7](#_TOC_250105)
   3. [Abbreviations 7](#_TOC_250104)
4. [General Principles 8](#_TOC_250103)
5. [Near-RT RIC Architecture 8](#_TOC_250102)
   1. [Requirements 8](#_TOC_250101)
      1. [Overall Requirements 8](#_TOC_250100)
      2. [Platform Requirements 8](#_TOC_250099)
      3. [xApp Requirements 9](#_TOC_250098)
      4. [Near-RT RIC API Requirements 10](#_TOC_250097)
   2. [Overall Architecture Description 11](#_TOC_250096)
6. [Near-RT RIC Functional Description 11](#_TOC_250095)
   1. [General 11](#_TOC_250094)
   2. [Functionalities of the Near-RT RIC platform 12](#_TOC_250093)
      1. [Database and SDL 12](#_TOC_250092)
      2. [xApp Subscription Management 13](#_TOC_250091)
      3. [Conflict Mitigation 13](#_TOC_250090)
      4. [Messaging Infrastructure 13](#_TOC_250089)
      5. [Security 14](#_TOC_250088)
      6. [Management 14](#_TOC_250087)
      7. [Interface Termination 14](#_TOC_250086)
      8. [API Enablement 15](#_TOC_250085)
      9. [AI/ML Support 16](#_TOC_250084)
      10. [xApp Repository Function 16](#_TOC_250083)
   3. [xApps 16](#_TOC_250082)

[6A Services of Near-RT RIC platform and xApps 17](#_TOC_250081)

[6A.1 General 17](#_TOC_250080)

[6A.2 Services provided by Near-RT RIC platform 18](#_TOC_250079)

[6A.2.1 Overview 18](#_TOC_250078)

[6A.2.2 A1 related services 19](#_TOC_250077)

[6A.2.3 E2 related services 19](#_TOC_250076)

[6A.2.4 Management related services 20](#_TOC_250075)

[6A.2.5 SDL services 21](#_TOC_250074)

[6A.2.6 Enablement services 21](#_TOC_250073)

[6A.2.7 AI/ML services 22](#_TOC_250072)

[6A.3 Services provided by xApps 23](#_TOC_250071)

[6A.3.1 Overview 23](#_TOC_250070)

[6A.3.2 A1 related services 24](#_TOC_250069)

[6A.3.3 E2 related services 24](#_TOC_250068)

[6A.3.4 Management related services 24](#_TOC_250067)

[6A.3.5 SDL services 25](#_TOC_250066)

[6A.3.6 Enablement services 25](#_TOC_250065)

1. [Near-RT RIC APIs 25](#_TOC_250064)
   1. [Overall Description 25](#_TOC_250063)
      1. [Introduction 25](#_TOC_250062)
      2. [Near-RT RIC API approaches 25](#_TOC_250061)
      3. [Network API approach 27](#_TOC_250060)
      4. [SDK approach 28](#_TOC_250059)
      5. [Near-RT RIC API support options 28](#_TOC_250058)
   2. [Void 31](#_TOC_250057)
   3. [Void 31](#_TOC_250056)
   4. [Void 31](#_TOC_250055)
   5. [Void 31](#_TOC_250054)
   6. [Void 31](#_TOC_250053)
   7. [Void 31](#_TOC_250052)
2. [External Interfaces of Near-RT RIC 31](#_TOC_250051)
   1. [E2 Interface 31](#_TOC_250050)
   2. [A1 Interface 31](#_TOC_250049)
   3. [O1 Interface 31](#_TOC_250048)
   4. [Y1 Interface 31](#_TOC_250047)
3. [Near-RT RIC API Procedures 32](#_TOC_250046)
   1. [Disclaimer 32](#_TOC_250045)
   2. [A1 Related API Procedures 32](#_TOC_250044)
      1. [Introduction 32](#_TOC_250043)
      2. [A1 Policy procedures 33](#_TOC_250042)
      3. [A1-EI related procedures 41](#_TOC_250041)
   3. [E2 Related API Procedures 52](#_TOC_250040)
      1. [Introduction 52](#_TOC_250039)
      2. [RIC Functional Procedures 52](#_TOC_250038)
      3. [E2 Conflict Mitigation related procedures 72](#_TOC_250037)
   4. [Management API Procedures 77](#_TOC_250036)
      1. [xApp Registration procedure 77](#_TOC_250035)
      2. [xApp Deregistration procedure 79](#_TOC_250034)
      3. [Void 79](#_TOC_250033)
      4. [Create MOI 79](#_TOC_250032)
      5. [Modify MOI attributes 80](#_TOC_250031)
      6. [Delete MOI 81](#_TOC_250030)
      7. [Read MOI attributes 82](#_TOC_250029)
      8. [Notify MOI changes 83](#_TOC_250028)
      9. [Subscription Control 84](#_TOC_250027)
      10. [Fault Notification 85](#_TOC_250026)
      11. [Fault Supervision Control 86](#_TOC_250025)
      12. [Performance Data File Reporting 87](#_TOC_250024)
      13. [Report Streamed Data 87](#_TOC_250023)
      14. [Measurement Job Control 88](#_TOC_250022)
   5. [SDL API Procedures 89](#_TOC_250021)
      1. [SDL Client Registration procedure 89](#_TOC_250020)
      2. [SDL Client Deregistration procedure 90](#_TOC_250019)
      3. [Fetch Data procedure 91](#_TOC_250018)
      4. [Subscribe/Update-Notify procedure 91](#_TOC_250017)
      5. [Store Data procedure 92](#_TOC_250016)
      6. [Modify Data procedure 93](#_TOC_250015)
      7. [Subscribe/Push procedure 94](#_TOC_250014)

[9.5.7A xApp Shared Data Fetch procedure 94](#_TOC_250013)

* + 1. [B Procedures related to xApp Shared Data Subscription and Notification 95](#_TOC_250012)
    2. [Use cases of SDL APIs 97](#_TOC_250011)
    3. [SDL API data structures 104](#_TOC_250010)
  1. [API Enablement Procedures 105](#_TOC_250009)
     1. [API Discovery procedure 105](#_TOC_250008)
     2. [Procedures related to API Event Subscription and Notification 107](#_TOC_250007)
     3. [Procedures related to API Registration 110](#_TOC_250006)
  2. [AI/ML API Procedures 115](#_TOC_250005)
     1. [Data Pipelining service procedures 115](#_TOC_250004)
     2. [Training service procedures 120](#_TOC_250003)
     3. [Inference service procedures 124](#_TOC_250002)
     4. [Model Management and Exposure Service procedures 129](#_TOC_250001)

Annex A (informative): Potential conflict types among xApps for RRM optimization 134

Annex B (informative): Example of xApp's workflow 135

[Annex (informative): Change History 137](#_TOC_250000)

# Foreword

This Technical Specification (TS) has been produced by WG3 of the O-RAN Alliance.

The content of the present document is subject to continuing work within O-RAN and may change following formal O-RAN approval. Should the O-RAN Alliance modify the contents of the present document, it will be re-released by O-RAN with an identifying change of version date and an increase in version number as follows:

version xx.yy.zz where:

xx: the first digit-group is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc. (the initial approved document will have xx=01). Always 2 digits with leading zero if needed.

yy: the second digit-group is incremented when editorial only changes have been incorporated in the document. Always 2 digits with leading zero if needed.

zz: the third digit-group included only in working versions of the document indicating incremental changes during the editing process. External versions never include the third digit-group. Always 2 digits with leading zero if needed.

# Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the O-RAN Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in O-RAN deliverables except when used in direct citation.

# Scope

The present document specifies the overall Near-RT-RIC (Near-real-time RAN Intelligent Controller) architecture and functionalities, including interaction between hosted applications and common functions in the Near-RT RIC platform.

# References

## Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE 1: While any hyperlinks included in this clause were valid at the time of publication, O-RAN cannot guarantee their long-term validity.

NOTE 2: In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document in 3GPP Release 18.

The following referenced documents are necessary for the application of the present document.

1. Void.
2. O-RAN.WG3.E2GAP, “O-RAN Working Group 3, Near-Real-time RAN Intelligent Controller, E2 General Aspects and Principles”.
3. O-RAN.WG3.E2AP, “O-RAN Working Group 3, Near-Real-time RAN Intelligent Controller, E2 Application Protocol (E2AP)”.
4. O-RAN.WG10.OAM-Architecture, “O-RAN Operations and Maintenance Architecture”.
5. O-RAN.WG10.O1-Interface, “O-RAN Operations and Maintenance Interface Specification”.
6. 3GPP TS 33.401: “3GPP System Architecture Evolution (SAE); Security architecture”.
7. 3GPP TS 33.501: “Security architecture and procedures for 5G System”.
8. O-RAN.WG2.A1GAP, “O-RAN Working Group 2, A1 interface: General Aspects and Principles”.
9. O-RAN.WG2.A1AP, “O-RAN Working Group 2, A1 Interface: Application Protocol”.
10. O-RAN.WG1.OAD, “O-RAN Working Group 1, O-RAN Architecture Description”.
11. Void.
12. Void.
13. Void.
14. Void.
15. Void.
16. Void.
17. Void.
18. Void.
19. Void.
20. O-RAN.WG6.ORCH-USE-CASES, "Orchestration Use Cases and Requirements for O-RAN Virtualized RAN".
21. 3GPP TS 28.622: “Generic Network Resource Model (NRM)”.
22. O-RAN.WG11.Security-Requirements-Specification: “O-RAN Security Requirements and Controls Specification”.

## Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE 1: While any hyperlinks included in this clause were valid at the time of publication, O-RAN cannot guarantee their long-term validity.

NOTE 2: In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document in 3GPP Release 18.

The following referenced documents are not necessary for the application of the present document, but they assist the user with regard to a particular subject area.

[i.1] 3GPP TR 21.905: “Vocabulary for 3GPP Specifications”.

[i.2] O-RAN.WG2.AIML: “O-RAN Working Group 2, AI/ML Workflow description and requirements”.

# Definition of terms, symbols and abbreviations

## Terms

For the purposes of the present document, the terms given in 3GPP TR 21.905 [i.1], O-RAN WG1.OAD [10] and the following apply:

Void.

## Symbols

For the purposes of the present document, the symbols given in [i.1] and the following apply: Void.

## Abbreviations

For the purposes of the present document, the abbreviations [given in [i.1] and the following] apply: E2SM E2 Service Model

IE Information Element

LCM Life-Cycle Management

MnS Management Service

R-NIB Radio-Network Information Base

SCTP Stream Control Transmission Protocol

SDK Software Development Kit

SDL Shared Data Layer

UE-NIB UE-Network Information Base

# General Principles

The general principles for Near-RT RIC architecture are as follows:

- The Near-RT RIC architecture and internal interfaces shall be open to support 3rd party xApps.

# Near-RT RIC Architecture

## Requirements

### Overall Requirements

* + - * [REQ-Overall-001] Void.

### Platform Requirements

* + - * [REQ-Platform-Generic-001] Near-RT RIC platform shall provide a service to expose up-to-date RAN information, history of time-varying network state, as well as configurations related to E2 Nodes, Cells, Bearers, Flows, UEs, and the mapping between them;
      * [REQ-Platform-Generic-002] Near-RT RIC shall provide AI/ML support for data pipelining, model management, training and inference;
      * [REQ-Platform-Generic-003] Near-RT RIC platform shall provide a messaging infrastructure;
      * [REQ-Platform-Generic-004] Near-RT RIC platform shall provide O1 support for file management, tracing and metrics collected from Near-RT RIC platform and xApps toward SMO;
      * [REQ-Platform-Generic-005] Near-RT RIC platform shall comply with WG11 security requirements [22];
      * [REQ-Platform-Generic-006] Near-RT RIC platform shall support mitigation of E2 related conflicts between xApps;
      * [REQ-Platform-API-001] Near-RT RIC platform shall eexpose Near-RT RIC APIs;
      * [REQ-Platform-API-002] The Near-RT RIC platform APIs shall be discoverable by xApp;
      * [REQ-Platform-API-003] Void;
      * [REQ-Platform-Generic-007] Void;
      * [REQ-Platform-Generic-008] Near-RT RIC platform shall support subscription merging from multiple xApps;
      * [REQ-Platform-Generic-009] Void;
      * [REQ-Platform-Generic-010] Near-RT RIC platform shall be able to route A1 policy management messages to the registered xApps based on A1 policy type;
      * [REQ-Platform-Generic-011] Near-RT RIC platform shall control access of A1-EI types for xApps based on operator policies;
      * [REQ-Platform-Management-001] The Near-RT RIC platform shall support the capability over O1 interface to expose the list of active alarms for the Near-RT RIC platform and hosted xApps;
      * [REQ-Platform-Management-002] The Near-RT RIC platform shall have the capability to report, over the O1 interface, PM measurements associated with individual xApps;
      * [REQ-Platform-Management-003] The Near-RT RIC platform shall support the capability over O1 interface to manage individual xApp’s Administrative State.
      * [REQ-Platform-Management-004] The Near-RT RIC platform shall support the capability over O1 interface to read the current configuration of individual xApps;
      * [REQ-Platform-Management-005] The Near-RT RIC platform shall support the capability over O1 interface to notify of any change in the xApp configuration;
      * [REQ-Platform-Management-006] The Near-RT RIC platform shall support the capability over O1 interface to notify of an associated MOI creation;
      * [REQ-Platform-Management-007] The Near-RT RIC platform shall support the capability over O1 interface to notify of an associated MOI deletion;
      * [REQ-Platform-Management-008] The Near-RT RIC platform shall support the capability over O1 interface to notify of an associated MOI modification;
      * [REQ-Platform-Management-009] The Near-RT RIC platform shall support the capability over O1 interface to notify that there is a file available;
      * [REQ-Platform-Management-010] The Near-RT RIC platform shall support PM data subscriptions for xApp streamed data;
      * [REQ-Platform-Management-011] The Near-RT RIC platform shall support the capability over O1 interface to configure the xApp to produce specific PM data;
      * [REQ-Platform-Management-012] The Near-RT RIC platform shall support the file transfer capability over O1 interface.

### xApp Requirements

* + - * [REQ-xApp-Generic-001] Void;
      * [REQ-xApp-E2Related-001] Void;
      * [REQ-xApp-E2Related-002] xApp shall use Near-RT RIC APIs to make use of the Information Elements (IEs) of E2SMs that are associated with it;
      * [REQ-xApp-E2Related-003] Void;
      * [REQ-xApp-E2Related-004] Void;
      * [REQ-xApp-Generic-002] xApp shall provide collected logging, tracing and metrics information to Near-RT RIC;
      * [REQ-xApp-Descriptor-001] Void;
      * [REQ-xApp-Descriptor-002] Void;
      * [REQ-xApp-API-001] xApps shall communicate with Near-RT RIC platform via Near-RT RIC APIs;
      * [REQ-xApp-API-002] xApp shall register the Near-RT RIC APIs it produces;
      * [REQ-xApp-API-003] xApp shall be capable of discovering the Near-RT RIC APIs it consumes;
      * [REQ-xApp-Management-001] xApp shall be configurable based on config received over O1 interface;
      * [REQ-xApp-Management-002] xApp shall expose configuration options to activate handling of individual A1 policy type when the xApp supports multiple A1 policy types.

### Near-RT RIC API Requirements

* + - * [REQ-API-Generic-001] Void;
      * [REQ-API-Generic-002] Near-RT RIC APIs shall not adversely impact low-latency and high throughput operations of Near-RT RIC. Specifically, the Near-RT RIC APIs shall support the Near-RT RIC control loop of execution time from 10 milliseconds to 1 second;
      * [REQ-API-Generic-003] Near-RT RIC APIs shall be decoupled from specific implementation solutions, including a Shared Data Layer (SDL) that works as an overlay for underlying databases and enables simplified data access;
      * [REQ-API-Generic-004] Near-RT RIC APIs shall support an API repository/registry for the services provided by the Near-RT RIC platform and/or xApps;
      * [REQ-API-Generic-005] Near-RT RIC APIs shall provide means for xApps to discover the published APIs based on the xApps’ needs;
      * [REQ-API-Generic-006] Near-RT RIC APIs shall provide means to restrict xApps from discovering some published APIs based on configured policies;
      * [REQ-API-Generic-007] Near-RT RIC APIs shall aim to simplify the development of xApps and enable rapid innovation;
      * [REQ-API-Generic-008] Near-RT RIC APIs shall support xApp development in multiple programming languages (e.g., C, C++, Python, Go);
      * [REQ-API-Generic-009] Near-RT RIC APIs shall support xApp subscription management based on operators’ policies. An xApp may be restricted to interface with only a subset of E2 Nodes by such policies. Near-RT RIC shall be responsible for routing messages between this xApp and the subset of E2 Nodes;
      * [REQ-API-Generic-010] Near-RT RIC APIs shall support mitigation of E2 related conflicts between xApps;
      * [REQ-API-E2Related-001] Near-RT RIC APIs shall enable all xApps to directly use the information elements of E2SMs with which they are associated;
      * [REQ-API-E2Related-002] The Near-RT RIC E2 Related APIs shall expose event-triggered RAN information and time-varying network state.
      * [REQ-API-Management-001] The Near-RT RIC APIs for xApp management shall provide the capability to configure the xApp;
      * [REQ-API-Management-002] The Near-RT RIC APIs for xApp management shall provide the capability to change the administrative state of the xApp;
      * [REQ-API-Management-003] The Near-RT RIC APIs for xApp management shall provide the capability to read the current configuration of the xApp;
      * [REQ-API-Management-004] The Near-RT RIC APIs for xApp management shall provide the capability for the xApp to notify the Management Function of any change in the xApp configuration;
      * [REQ-API-Management-005] The Near-RT RIC APIs for xApp management shall provide the capability for the xApp to notify the Management Function of an associated MOI Creation;
      * [REQ-API-Management-006] The Near-RT RIC APIs for xApp management shall provide the capability for the xApp to notify the Management Function of an associated MOI Deletion;
      * [REQ-API-Management-007] The Near-RT RIC APIs for xApp management shall provide the capability for the xApp to notify the Management Function of a fault detection;
      * [REQ-API-Management-008] The Near-RT RIC APIs for xApp management shall provide the capability for the Consumer (SMO) to fetch the list of Active Faults on the xApp;
      * [REQ-API-Management-009] The Near-RT RIC APIs for xApp management shall provide the capability for the xApp to notify the Management Function that there is a file available;
      * [REQ-API-Management-010] The Near-RT RIC APIs for xApp management shall provide the capability for the Management Function to configure the xApp to send streaming data;
      * [REQ-API-Management-011] The Near-RT RIC APIs for xApp management shall provide the capability for the Management Function to configure the xApp to collect specific PM data.

## Overall Architecture Description

The overall O-RAN architecture specified in [10] describes the location and interfaces of Near-RT RIC, as well as possible deployment options.

A Near-RT RIC is connected to only one Non-RT RIC.

A Near-RT RIC is connected to one or more E2 Nodes as described in [2].

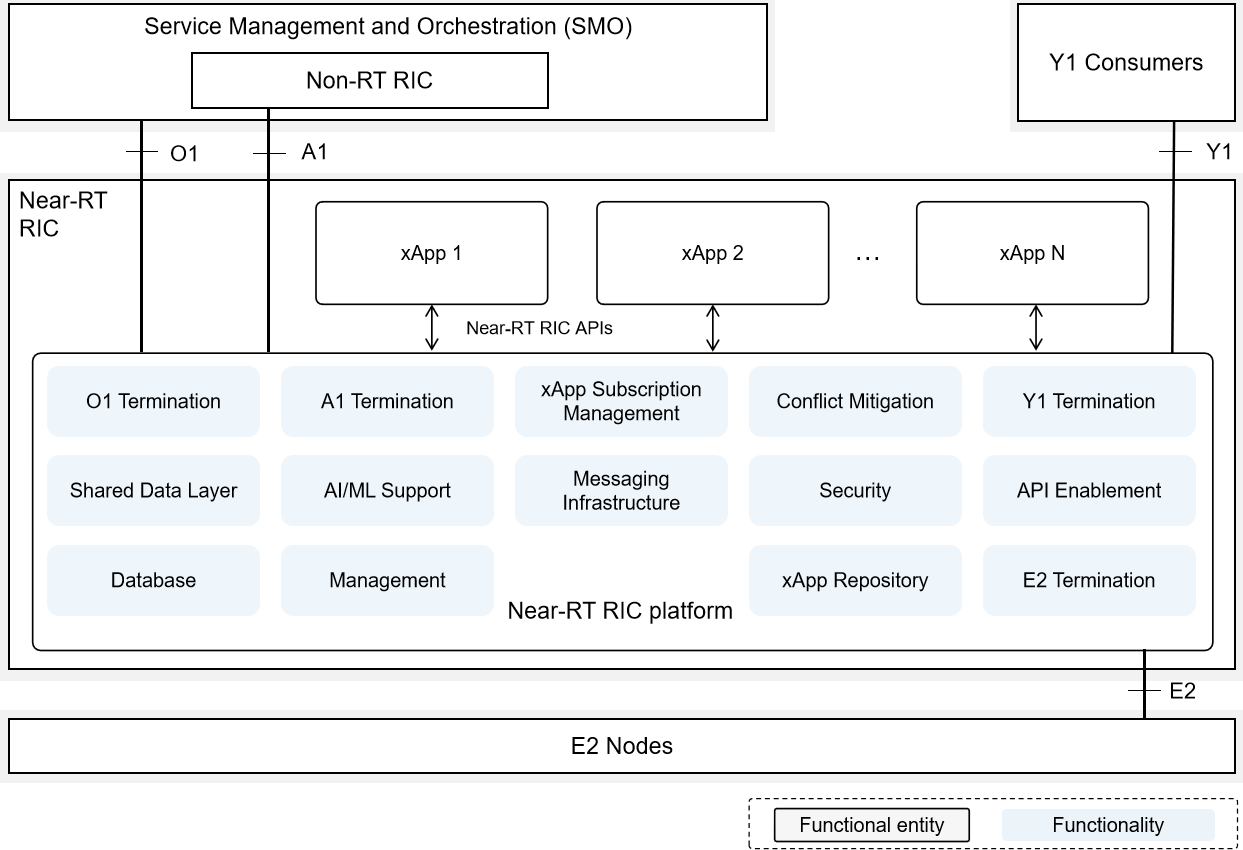
The RRM functional allocation between Near-RT RIC and E2 Node is described in [2].

# Near-RT RIC Functional Description

## General

A Near-RT RIC is composed of a Near-RT RIC platform and one or more xApps. An overview in shown in Figure 6.1-1.

Figure 6.1-1 also shows the functionalities of the Near-RT RIC platform. Those functionalities are further described in clause 6.2.



##### Figure 6.1-1: Near-RT RIC Architecture Overview

## Functionalities of the Near-RT RIC platform

### Database and SDL

#### Overview

The database and SDL functionalities enable the Near-RT RIC platform to maintain and expose RAN/UE information and other information required to support specific use cases.

#### UE-NIB

This functionality enables the Near-RT RIC platform to maintain UE related information:

* + - * + Maintaining a list of UEs and associated data;
        + Tracking and correlation of the UE identities associated with the connected E2 Nodes.

Such information may be generated and accessed by the Near-RT RIC platform or authorized xApps.

#### R-NIB

This functionality enables the Near-RT RIC platform to maintain RAN related information:

* + - * + Maintaining the configurations and near real-time information relating to connected E2 Nodes and the mappings between them.

Such information may be generated and accessed by the Near-RT RIC platform or authorized xApps.

#### SDL

This functionality enables the Near-RT RIC platform to provide SDL services for xApps, which can be used to subscribe to database notifications and to read, write and modify information stored on the database. UE-NIB, R-NIB and other use case specific information may be exposed using the SDL services.

### xApp Subscription Management

This functionality enables the Near-RT RIC platform to handle subscription requests from different xApps for E2 related data, and provide unified data distribution to xApps for those data:

* Managing subscriptions from xApps to E2 Nodes;
* Enforcing authorization of policies controlling xApp access to messages;
* Enabling merging of identical subscriptions from different xApps into a single subscription toward an E2 Node;
* Enabling audit of existing subscriptions and initiating associated corrective actions.

### Conflict Mitigation

This functionality enables the Near-RT RIC platform to detect and resolve potentially overlapping or conflicting requests from multiple xApps.

In the context of Near-RT RIC, Conflict Mitigation is about addressing conflicting interactions between different xApps. An application will typically change one or more parameters with the objective of optimizing a specific metric. Conflict Mitigation is necessary because xApps objectives may be chosen/configured such that they result in conflicting actions.

### Messaging Infrastructure

This functionality enables the Near-RT RIC platform with low-latency message delivery between Near-RT RIC internal endpoints.

* It supports registration/discovery/deletion of endpoints:
  + Registration: Endpoints register themselves to the messaging infrastructure;
  + Discovery: Endpoints are discovered by the messaging infrastructure initially and registered to the messaging infrastructure;
  + Deletion: Endpoints are deleted once they are not used anymore.
* It provides the following APIs:
  + An API for sending messages to the messaging infrastructure;
  + An API for receiving messages from the messaging infrastructure.
* It supports multiple messaging modes, e.g., point-to-point mode (e.g., message exchange among endpoints), publish/subscribe mode (e.g., real-time data dispatching from E2 termination to multiple subscriber xApps);
* It provides message routing, namely according to the message routing information, messages can be dispatched to different endpoints;
* It supports message robustness to avoid data loss during a messaging infrastructure outage/restart or to release resources from the messaging infrastructure once a message is outdated.

### Security

The security functionality given in this clause only applies to Near-RT RIC. One of the targets is to prevent malicious xApps from abusing radio network information (e.g., exporting to unauthorized external systems) and/or control capabilities over RAN functions. The security requirements for the 3GPP LTE eNB is defined in [6] and for the 5G NR gNB in [7].

NOTE: The description of security functionalities is not included in the release.

### Management

#### Void

#### OAM Management of Near-RT RIC

OAM management consists of fault, configuration, accounting, performance, file, security and other management plane services. OAM management follows O1 related management aspects defined in [4].

To support OAM management services, Near-RT RIC platform provides the following capabilities:

* + - * + Fault Management: The Near-RT RIC platform provides Near-RT RIC Fault Supervision MnS over the O1 interface as defined in [4];
        + Configuration management: The Near-RT RIC platform provides Near-RT RIC Provisioning MnS over the O1 interface as defined in [4];
        + Logging: logging is to capture information needed to operate, troubleshoot and report on the performance of the Near- RT RIC platform and its constituent components. Log records may be viewed and consumed directly by users and systems, indexed and loaded into a data storage, and used to compute metrics and generate reports. Near-RT RIC components log events according to a common logging format. Different logs can be generated (e.g., audit log, metrics log, error log and debug log).
        + Tracing: tracing mechanisms are needed to monitor the transactions or a workflow. An example subscription workflow can be broken into two traces namely, a subscription request trace followed by a response trace. Individual traces can be analysed to understand timing latencies as the workflow traverses a particular Near-RT RIC component.
        + Metrics collection: metrics for performance and fault management specific to each xApp logic and other internal functionalities are collected and published for authorized consumer (e.g., SMO). A metrics collection mechanism is needed to collect and report metrics.

### Interface Termination

#### E2 Termination

This functionality enables termination of E2 interface with the following:

* + - * + Terminating SCTP connection from each E2 Node;
        + Routing messages from xApps through the SCTP connection to an E2 Node;
        + Decoding the payload of an incoming ASN.1 message enough to determine message type;
        + Handling incoming E2 messages related to E2 connectivity;
        + Receiving and responding to the E2 Setup Request from an E2 Node;
        + Notifying xApps of the list of RAN functions supported by an E2 Node based on information derived from the E2 Setup and RIC Service Update procedures [3];
        + Notifying the newly connected E2 Node of the list of accepted functions.

#### A1 Termination

This functionality enables termination of A1 interface with the following:

* + - * + Providing a generic API by means of which Near-RT RIC can receive and send messages via A1 interface [8]. These include, e.g., A1 policies and enrichment information received from Non-RT RIC, or A1 policy feedback sent towards Non-RT RIC.
        + Sending A1 policy that is set or updated by the Non-RT RIC to suitable xApp(s) based on a list of candidate xApp(s) provided by xApp Repository functionality.

#### O1 Termination

This functionality enables termination of O1 interface.

An implementation of O1 Termination at Near-RT RIC depends on the deployment options described in [4], i.e., when Near- RT RIC is modelled as a stand-alone Managed Element.

Near-RT RIC platform communicates with SMO via O1 interface and exposes O1-related management services [5] from Near- RT RIC. For the O1 management services (MnS) shown below, Near-RT RIC platform is the MnS producer and SMO is the MnS consumer:

* Exposing provisioning management services from Near-RT RIC to O1 provisioning management service consumer;
* Translating O1 management services to Near-RT RIC internal APIs;
* Exposing FM services to report faults and events from Near-RT RIC to O1 FM service consumer;
* Exposing PM services to report bulk and real-time PM data from Near-RT RIC to O1 PM service consumer;
* Exposing file management services to download ML files, software files, etc. and upload log/trace files to/from file MnS consumer;
* Exposing communication surveillance services to O1 communication surveillance service consumer.

#### Y1 Termination

This functionality enables termination of Y1 interface.

It provides support for exposing RAN analytics information from Near-RT RIC to Y1 consumers.

### API Enablement

In the context of Near-RT RIC, the Near-RT RIC APIs 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.

This functionality provides support for registration, discovery and consumption of Near-RT RIC APIs within the Near-RT RIC scope. In particular, the API enablement services include:

* Repository / Registry services for the Near-RT RIC APIs;
* Services that allow discovery of the registered Near-RT RIC APIs;
* Services to authenticate xApps for use of the Near-RT RIC APIs;
* Services that enable generic subscription and event notification;
* Means to avoid compatibility clashes between xApps and the services they access.

The API enablement services can be accessed by the xApps via one or more enablement APIs.

NOTE: The provided enablement APIs may need to consider the level of trust related to the xApp (e.g., 3rd party xApp, RIC-owned xApp, etc.).

### AI/ML Support

#### Introduction

The AI/ML support functionality enables the Near-RT RIC platform with data pipelining, model management, training and inference, which constitute complete AI/ML workflow support for xApps to implement AI/ML algorithms. xApps may use none, part, or all of this functionality, depending on its design.

#### Data Pipelining

This functionality enables the Near-RT RIC platform with data ingestion and preparation for xApps.

The input may include E2 node data collected over E2 interface, enrichment information over A1 interface, information from xApps, and data retrieved from the Near-RT RIC database through the messaging infrastructure.

The output is data sets that are ready to be consumed by AI/ML Model. The output may be stored in an xApp specific space (e.g., a dedicated space in database) or used as input for inference (see clause 6.2.9.3).

#### A Model Management

The Near-RT RIC platform offers storage, retrieval, and version control of AI/ML models for xApps. The Near-RT RIC platform may store the model in an xApp specific space (e.g., a dedicated space in database), and tracks the version change of the model if the model is retrained or updated later.

#### Training

This functionality enables training of AI/ML Models for xApps within Near-RT RIC [i.2]. The Near-RT RIC platform provides generic and use case-independent capabilities to AI/ML-based xApps that may be useful to multiple use cases. The associated AI/ML models are managed by the Near-RT RIC platform (see clause 6.2.9.1A). The input data for training may come from an xApp specific space (e.g., dedicated space in database).

#### Inference

Near-RT RIC platform offers inference of AI/ML Models for xApps. The associated AI/ML models are managed by the Near- RT RIC platform (see clause 6.2.9.1A). The input data may be the output of data pipelining, or come from an xApp specific space (e.g., a dedicated space in database). The inferred result may be provided to the requesting xApp or stored in an xApp specific space (e.g., a dedicated space in database).

### xApp Repository Function

This functionality enables the Near-RT RIC platform with the following:

* Maintaining a list of candidate xApp(s) to A1 Termination for sending A1 policy or policy update to suitable xApp(s) based on policy type and operator policies;
* Maintaining the policy types supported in Near-RT RIC. The supported policy types are based on policy types supported by the registered xApps and operator policies;
* Performing xApp access control to requested A1-EI type based on operator policies.

## xApps

The xApps collaborate with the Near-RT RIC platform to support various specialized use cases.

The xApps may collaborate with Near-RT RIC platform to support mitigation of E2 related conflicts. xApp may be used to enhance the RRM capability in RAN.

xApp may leverage zero, one or more E2SMs.

Upon registration to the Near-RT RIC platform, an xApp informs the Near-RT RIC platform of its OAM and control information to enable relevant functionalities.

NOTE: The LCM of xApp is performed by SMO and O-Cloud as specified in [4] and [20].

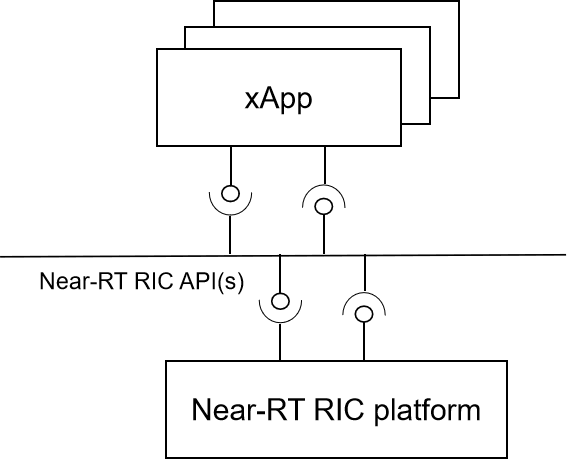
# 6A Services of Near-RT RIC platform and xApps

## 6A.1 General

Among the Near-RT RIC platform and the xApps, their capabilities are exposed as services. Both the Near-RT RIC platform and an xApp can be a service producer and/or a service consumer. Those services can be leveraged via the Near-RT RIC APIs described in clause 7.

The interactions between service consumer and service producer are generalized that the service requests are sent from the service consumer and responses and notifications are sent from the service producer. It is the service producer that handles the resources on which the service consumer performs operations. The terms service consumer and service producer thus, do not refer to the direction of the data transfer over the Near-RT RIC APIs.

A service-based representation is illustrated in Figure 6A.1-1.



##### Figure 6A.1-1: Service-based representation for internal Near-RT RIC

The services can be classified into the following categories, in regard to the relevant functionalities:

* A1 related services;
* E2 related services;
* Management related services;
* SDL services;
* Enablement services;
* AI/ML workflow services.

The services provided by the Near-RT RIC platform and the xApps are described in clause 6A.2 and 6A.3, respectively.

## 6A.2 Services provided by Near-RT RIC platform

### 6A.2.1 Overview

An overview of the services that can be provided by the Near-RT RIC platform is shown in Table 6A.2.1-1:

##### Table 6A.2.1-1: Services provided by the Near-RT RIC platform

|  |  |  |
| --- | --- | --- |
| **Service Category** | **Service Name** | **Description** |
| A1 related services | A1 EI service | This service enables the service consumer to create/update/delete/audit subscriptions to A1 Enrichment Information. It also enables the service consumer to query the Near-RT RIC platform of the supported A1 EI type(s) and the created A1 EI  job(s). |
| E2 related services | E2 Subscription service | This service enables the service consumer to create/delete/modify/audit an E2 subscription related to applicable RIC services [2] with Near-RT RIC platform. The subscribed information is delivered with E2 Indication service.  This service also enables the service consumer to be queried or notified of deleting an existing E2 subscription.  (NOTE 1) |
| E2 Indication service | This service enables the service consumer to  receive subscribed E2 information associated with an E2 subscription. |
| E2 Control service | This service enables the service consumer to access  RIC service “CONTROL” [2] with the Near-RT RIC platform. |
| E2 Guidance service | This service enables the service consumer to acquire E2 guidance information from the Near-RT RIC platform to avoid conflicts described in clause  6.2.3. |
| E2 Query service | This service enables the service consumer to access RIC service “QUERY” [2] with the Near-RT RIC  platform. |
| Management related services | xApp Registration service | This service enables service consumer to register/deregister itself with the Near-RT RIC  platform. |
| SDL services | SDL Information service | This service enables the service consumer to store and retrieve data while hiding details such as type and location of database, management operations of database layer such as high availability, scaling, load-balancing.  This service supports defined data structures used to  provide E2 Node related information. |
| Enablement services | Enabl Registration service | This service enables service consumer to register its  service. |
| Enabl Discovery service | This service enables service consumer to discover services registered by the Near-RT RIC platform and  the xApps. |
| Enabl Event service | This service enables the service consumer to create/delete subscriptions to Near-RT RIC API related events. It also enables notifications of the  subscribed Near-RT RIC API related events. |

|  |  |  |
| --- | --- | --- |
| **Service Category** | **Service Name** | **Description** |
| AI/ML workflow services | Data Pipelining service | This service enables the service consumer to ingest  and prepare data for AI/ML model training and/or inference with the Near-RT RIC platform. |
| AI/ML Model Management and Exposure service | This service enables the service consumer to store and, retrieve AI/ML models with the Near-RT RIC platform for inference and/or training, The service  also supports exposure of shared AI/ML models. |
| AI/ML Training service | This service enables the service consumer to train  AI/ML models with the Near-RT RIC platform. |
| AI/ML Inference service | This service enables the service consumer to perform AI/ML inference with the Near-RT RIC  platform. |
| NOTE 1: Support for E2 interface Global Procedures (E2 SETUP, RIC Service Update, etc.) is described in clause  9.5.8 as use cases of SDL services. | | |

### 6A.2.2 A1 related services

#### 6A.2.2.1 A1 EI service

The A1 EI service supports the following operations in Table 6A.2.2.1-1.

##### Table 6A.2.2.1-1: List of A1 EI service operations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Service Name** | **Service Operations** | **Operation**  **Semantic** | **Known**  **Consumer(s)** | **Reference** |
| A1 EI service | A1 EI query | Request/Response | xApp | 9.2.3.1 |
| A1 EI subscription setup | Request/Response | xApp | 9.2.3.2 |
| A1 EI subscription update | Request/Response | xApp | 9.2.3.3 |
| A1 EI subscription delete | Request/Response | xApp | 9.2.3.4 |
| A1 EI delivery | Notify | xApp | 9.2.3.5 |

### 6A.2.3 E2 related services

#### 6A.2.3.1 E2 Subscription service

The E2 Subscription service supports the following operations in Table 6A.2.3.1-1.

##### Table 6A.2.3.1-1: List of E2 Subscription service operations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Service Name** | **Service Operations** | **Operation**  **Semantic** | **Known**  **Consumer(s)** | **Reference** |
| E2 Subscription service | E2 Subscription | Request/Response | xApp | 9.3.2.1 |
| E2 Subscription Delete | Request/Response | xApp | 9.3.2.2 |
| E2 Subscription Delete  Query | Request/Response | xApp | 9.3.2.2A |
| E2 Subscription Delete  Notification | Notify | xApp | 9.3.2.2A |
| E2 Subscription Audit | Request/Response | xApp | 9.3.2.5 |
| E2 Subscription  Modification | Request/Response | xApp | 9.3.2.7 |

#### 6A.2.3.2 E2 Indication service

The E2 Indication service supports the following operations in Table 6A.2.3.2-1.

##### Table 6A.2.3.2-1: List of E2 Indication service operations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Service Name** | **Service Operations** | **Operation**  **Semantic** | **Known**  **Consumer(s)** | **Reference** |
| E2 Indication  service | E2 Indication | Notify | xApp | 9.3.2.3 |

#### 6A.2.3.3 E2 Control service

The E2 Control service supports the following operations in Table 6A.2.3.3-1.

##### Table 6A.2.3.3-1: List of E2 Control service operations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Service Name** | **Service Operations** | **Operation**  **Semantic** | **Known**  **Consumer(s)** | **Reference** |
| E2 Control service | E2 Control | Request/Response | xApp | 9.3.2.4 |

#### 6A.2.3.4 E2 Guidance service

The E2 Guidance service supports the following operations in Table 6A.2.3.4-1.

##### Table 6A.2.3.4-1: List of E2 Guidance service operations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Service Name** | **Service Operations** | **Operation**  **Semantic** | **Known**  **Consumer(s)** | **Reference** |
| E2 Guidance  service | E2 Guidance | Request/Response | xApp | 9.3.3.1 |
| E2 Guidance Modification | Notify | xApp | 9.3.3.5 |

#### 6A.2.3.5 E2 Query service

The E2 Query service supports the following operations in Table 6A.2.3.5-1.

##### Table 6A.2.3.5-1: List of E2 Query service operations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Service Name** | **Service Operations** | **Operation**  **Semantic** | **Known**  **Consumer(s)** | **Reference** |
| E2 Query service | E2 Query | Request/Response | xApp | 9.3.2.6 |

### 6A.2.4 Management related services

#### 6A.2.4.1 xApp Registration service

The xApp Registration service supports the following operations in Table 6A.2.4.1-1.

##### Table 6A.2.4.1-1: List of xApp Registration service operations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Service Name** | **Service Operations** | **Operation**  **Semantic** | **Known**  **Consumer(s)** | **Reference** |
| xApp Registration  service | xApp Registration | Request/Response | xApp | 9.4.1 |
| xApp Deregistration | Request/Response | xApp | 9.4.2 |

### 6A.2.5 SDL services

#### 6A.2.5.1 SDL Information service

The SDL Information service supports the following operations in Table 6A.2.5.1-1.

##### Table 6A.2.5.1-1: List of SDL Information service operations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Service Name** | **Service Operations** | **Operation**  **Semantic** | **Known**  **Consumer(s)** | **Reference** |
| SDL Information service | SDL Client Registration | Request/Response | xApp | 9.5.1 |
| SDL Client Deregistration | Request/Response | xApp | 9.5.2 |
| SDL Information Fetch | Request/Response | xApp | 9.5.3 |
| SDL Subscription  Information | Request/Response | xApp | 9.5.4, 9.5.7 |
| SDL Information Update  Notify | Notify | xApp | 9.5.4 |
| SDL Information Push | Notify | xApp | 9.5.7 |
| SDL Information Store | Request/Response | xApp | 9.5.5 |
| SDL Information Modify | Request/Response | xApp | 9.5.6 |

### 6A.2.6 Enablement services

#### 6A.2.6.1 Enabl Registration service

The Enabl Registration service supports the following operations in Table 6A.2.6.1-1.

##### Table 6A.2.6.1-1: List of Enabl Registration service operations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Service Name** | **Service Operations** | **Operation**  **Semantic** | **Known**  **Consumer(s)** | **Reference** |
| Enabl Registration service | API Registration | Request/Response | xApp | 9.6.3.1 |
| API Registration Update | Request/Response | xApp | 9.6.3.2 |
| API Deregistration | Request/Response | xApp | 9.6.3.3 |

#### 6A.2.6.2 Enabl Discovery service

The Enabl Discovery service supports the following operations in Table 6A.2.6.2-1.

##### Table 6A.2.6.2-1: List of Enabl Discovery service operations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Service Name** | **Service Operations** | **Operation**  **Semantic** | **Known**  **Consumer(s)** | **Reference** |
| Enabl Discovery  service | API Discovery | Request/Response | xApp | 9.6.1 |

#### 6A.2.6.3 Enabl Events service

The Enabl Events service supports the following operations in Table 6A.2.6.3-1.

##### Table 6A.2.6.3-1: List of Enabl Events service operations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Service Name** | **Service Operations** | **Operation**  **Semantic** | **Known**  **Consumer(s)** | **Reference** |
| Enabl Events service | API Event Subscription | Request/Response | xApp | 9.6.2.1 |
| API Event Subscription  Delete | Request/Response | xApp | 9.6.2.2 |
| API Event Notification | Notify | xApp | 9.6.2.3 |

### 6A.2.7 AI/ML services

#### 6A.2.7.1 Data Pipelining service

The Data Pipelining service supports the following operations in Table 6A.2.7.1-1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table 6A.2.7.1-1:**  **List of Data Pipelining service operationsServic e Name** | **Service Operations** | **Operation Semantic** | **Known Consumer(s)** | **Reference** |
| Data Pipelining service | Data Pipelining Job Setup | Request/Response | xApp | 9.7.1.1 |
| Data Pipelining Job Update | Request/Response | xApp | 9.7.1.2 |
| Data Pipelining Job Delete | Request/Response | xApp | 9.7.1.3 |
| Data Pipelining Job Query | Request/Response | xApp | 9.7.1.4 |
| Data Pipelining Request | Request/Response | xApp | 9.7.1.5 |
| Data Pipelining Result Delivery | Notify | xApp | 9.7.1.5 |

#### 6A.2.7.2 Training service

The Training service supports the following operations in Table 6A.2.7.2-1.

##### Table 6A.2.7.2-1: List of Training service operations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Service Name** | **Service Operations** | **Operation**  **Semantic** | **Known**  **Consumer(s)** | **Reference** |
| Training Service | Training Job Setup | Request/Response | xApp | 9.7.2.1 |
| Training Job Delete | Request/Response | xApp | 9.7.2.2 |
| Training Job Query | Request/Response | xApp | 9.7.2.3 |
| Training Request | Request/Response | xApp | 9.7.2.4 |
| Training Result Delivery | Notify | xApp | 9.7.2.4 |

#### 6A.2.7.3 Inference service

The Inference service supports the following operations in Table 6A.2.7.3-1.

##### Table 6A.2.7.3-1: List of Inference service operations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Service Name** | **Service Operations** | **Operation**  **Semantic** | **Known**  **Consumer(s)** | **Reference** |
| Inference Service | Inference Job Setup | Request/Response | xApp | 9.7.3.1 |
| Inference Job Update | Request/Response | xApp | 9.7.3.2 |
| Inference Job Delete | Request/Response | xApp | 9.7.3.3 |
| Inference Job Query | Request/Response | xApp | 9.7.3.4 |
| Inference Request | Request/Response | xApp | 9.7.3.5 |
| Inference Result Delivery | Notify | xApp | 9.7.3.5 |

#### 6A.2.7.4 Model Management and Exposure service

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Service Name** | **Service Operations** | **Operation**  **Semantic** | **Known**  **Consumer(s)** | **Reference** |
| Model Management and Exposure Service | Model Store | Request/Response | xApp | 9.7.4.1 |
| Stored Model Query | Request/Response | xApp | 9.7.4.2 |
| Model Retrieve | Request/Response | xApp | 9.7.4.3 |
| Model Delete | Request/Response | xApp | 9.7.4.4 |

## 6A.3 Services provided by xApps

### 6A.3.1 Overview

An overview of the services that can be provided by the xApps is shown in Table 6A.3.1-1:

##### Table 6A.3.1-1: Services provided by the xApps

|  |  |  |
| --- | --- | --- |
| **Service Category** | **Service Name** | **Description** |
| A1 related services | A1 Policy service | This service enables the service consumer to create/update/delete/query an A1 policy object [8] for the xApp, and to be notified of the status of the A1 policy object. It also enables the service consumer to query the xApp about the supported A1 policy  type(s). |
| E2 related services | E2 Conflict Mitigation Assistance  service | The service enables the service consumer to obtain  assistance from the xApp for conflict mitigation. |
| Management related services  (NOTE 1) | xApp Configuration service | This service enables the service consumer to  configure the xApp. |
| xApp PM service | This service enables the service consumer to collect  PM related information from the xApp. |
| xApp FM service | This service enables the service consumer to collect  faults and events information from the xApp. |
| SDL services | xApp data sharing service | The service enables the service consumer to collect  specific data shared by the xApp. |
| NOTE 1: Trace related service is not specified in the present document. | | |

### 6A.3.2 A1 related services

#### 6A.3.2.1 A1 Policy service

The A1 Policy service supports the following operations in Table 6A.3.2.1-1.

##### Table 6A.3.2.1-1: List of A1 Policy service operations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Service Name** | **Service Operations** | **Operation**  **Semantic** | **Known**  **Consumer(s)** | **Reference** |
| A1 Policy service | A1 Policy Setup | Request/Response | Near-RT RIC  platform | 9.2.2.1 |
| A1 Policy Update | Request/Response | Near-RT RIC  platform | 9.2.2.2 |
| A1 Policy Delete | Request/Response | Near-RT RIC  platform | 9.2.2.3 |
| A1 Policy Query | Request/Response | Near-RT RIC  platform | 9.2.2.4 |
| A1 Policy Status Update | Notify | Near-RT RIC  platform | 9.2.2.5 |

### 6A.3.3 E2 related services

#### 6A.3.3.1 E2 Conflict mitigation service

##### Table 6A.3.3.1-1: List of E2 Conflict mitigation service operations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Service Name** | **Service Operations** | **Operation**  **Semantic** | **Known**  **Consumer(s)** | **Reference** |
| E2 Conflict Mitigation Assistance  Service | E2 Conflict Mitigation Assistance | Request/Response | Near-RT RIC platform | 9.3.3.6 |

### 6A.3.4 Management related services

#### 6A.3.4.1 xApp Configuration service

void

#### 6A.3.4.3 xApp PM service

void

#### 6A.3.4.4 xApp FM service

void

### 6A.3.5 SDL services

#### 6A.3.5.1 xApp data sharing service

The xApp data sharing service supports the following operations in Table 6A.3.5.1-1.

##### Table 6A.3.5.1-1: List of xApp Data sharing service operations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Service Name** | **Service Operations** | **Operation**  **Semantic** | **Known**  **Consumer(s)** | **Reference** |
| xApp Data Sharing service | xApp Shared Data Fetch | Request/Response | Near-RT RIC  platform, xApp | 9.5.7A |
| xApp Shared Data  Subscription | Request/Response | Near-RT RIC  platform, xApp | 9.5.7B.1 |
| xApp Shared Data  Subscription Delete | Request/Response | Near-RT RIC  platform, xApp | 9.5.7B.2 |
| xApp Shared Data Push | Notify | Near-RT RIC  platform, xApp | 9.5.7B.1 |

### 6A.3.6 Enablement services

In the current release of specification, no Enablement services can be provided by the xApp.

# Near-RT RIC APIs

## Overall Description

### Introduction

The Near-RT RIC APIs are a collection of well-defined interfaces providing Near-RT RIC platform and xApp services. These APIs need to explicitly define the possible types of information flows and data models. The Near-RT RIC APIs are essential to host 3rd party xApps in an inter-operable way on different Near-RT RIC platforms.

Near-RT RIC provides the following Near-RT RIC APIs for xApps:

* + - * A1 related APIs: APIs to access A1 related services;
      * E2 related APIs: APIs to access E2 related services;
      * Management APIs: APIs to access management related services;
      * SDL APIs: APIs to access Shared Data Layer related services;
      * Enablement APIs: APIs to access enablement services;
      * AI/ML workflow APIs: APIs to access AI/ML workflow services.

### Near-RT RIC API approaches

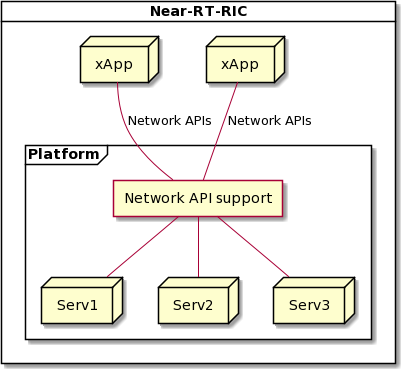
Near-RT RIC APIs may be implemented using two approaches:

* + - * **Network API approach:** Each relevant Near-RT RIC endpoint exposes a network endpoint and specifies a particular data encoding protocol and a network transport protocol that should be used to communicate with it. A different network API may be specified for each Near-RT RIC API reflecting a trade-off between different service requirements.
      * **SDK approach:** The Near-RT RIC vendor provides an SDK (software development kit). This SDK is a software library which handles all connection management and exposes a simple API for the xApp to interact with the Near- RT RIC. The interface between the SDK library embedded in the xApp and the Near-RT RIC Platform may be either vendor proprietary or aligned to the specified Network API.

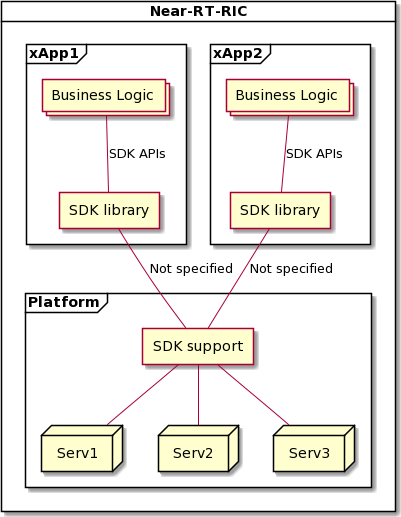
Both approaches use similar methodologies of interaction with Near-RT RIC as defined in clauses 6A and 9. Furthermore, the detailed messages and IEs for each Near-RT RIC API that the xApp will use largely stay the same in the two approaches.

These two approaches are presented in figure 7.1.2-1 for the Network API approach and figure 7.1.2-2 for the SDK approach.

Both approaches assume that the Near-RT RIC platform provides appropriate support services. These services, labelled “Network API support” and “SDK support” respectively in figures 7.1.2-1 and 7.1.2-2, may be integrated into other platform services (i.e., A1 Termination may provide the support services for A1 related APIs) or implemented as independent services.



##### Figure 7.1.2-1: Near-RT RIC platform supporting RIC APIs using Network API approach



##### Figure 7.1.2-2: Near-RT RIC platform supporting RIC APIs using SDK approach

### Network API approach

A Network API is defined in terms of a protocol stack including the definition:

* + - * An application layer protocol used to carry a set of messages which normally contain multiple Information Elements (IEs);
      * A data encoding protocol (ASN.1, Protobuf, JSON, etc.);
      * A network transport protocol (SCTP, HTTPS, gRPC, etc.);
      * Associated security and encryption methods.

For the Network API approach for Near-RT RIC APIs, it is expected that each case listed in clause 7.1.1 may require a unique protocol stack definition which offers an optimal solution balancing the trade-offs between service requirements, Near-RT RIC platform complexity and xApp implementation overheads.

Furthermore, each Network API will require a unique endpoint that the API consumer will need to discover using a combination of configuration and use of the Enablement APIs (also see clause 6A.2.6).

Open O-RAN specified Network APIs for each case listed in clause 7.1.1 are defined in terms of the corresponding protocol stack.

### SDK approach

An SDK (Software Development Kit) is defined as a set of software tools/libraries provided by platform vendors which helps developers in building applications on the platform. An SDK simplifies application development by:

* + - * Providing simple APIs to trigger commonly used functionality;
      * Handling routine management tasks “under the hood”;
      * Providing tools for debugging, building, testing applications.

For the SDK approach for Near-RT RIC APIs, it is expected that a given Near-RT RIC vendor offers an SDK library that provides a language specific implementation of each case listed in clause 7.1.1. This provides a common interface to interact with all the Near-RT RIC endpoints as well as other xApps.

The Near-RT RIC SDK library functionality would include:

* + - * Discovery of relevant endpoints for and connection to Near-RT RIC platform services and other xApps.
      * Handles all connection management of the xApp with all relevant Near-RT RIC platform endpoints as well as other xApps. This includes:
        + Authentication;
        + Registration;
        + Encryption/Decryption (if required);
        + Failovers if any endpoint crashes.
      * Presents a simple API as function definitions for sending/receiving messages;
      * Provides tools for xApp debugging, building and testing.

These services would be provided by the SDK library and so would not be exposed to the business logic within an xApp.

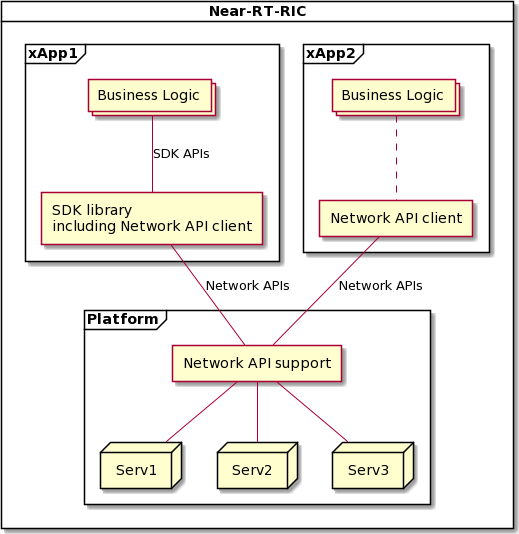
Open O-RAN specified SDK APIs for each case listed in clause 7.1.1 would need to be defined for each supported programming language.

The actual protocol stack implemented by the SDK library and used to communicate with the SDK support service in the Near- RT RIC platform may be either based on the Network APIs or left not specified and so may be vendor proprietary.

### Near-RT RIC API support options

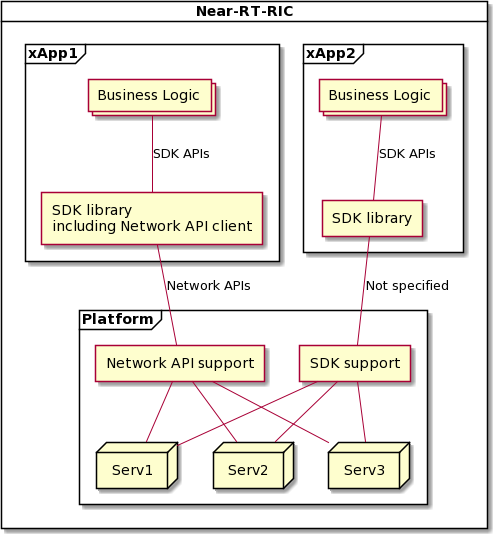
The two approaches to Near-RT RIC APIs are not necessarily mutually exclusive.

For example, an xApp designed to support the Network API approach may be implemented using either an internal SDK library that provides the Network API interface or a direct implementation the Network API interface. Both of these cases are presented in figure 7.1.5-1.



##### Figure 7.1.5-1: Near-RT RIC platform supporting RIC APIs using Network API approach with optional use of SDK APIs

Likewise, a Near-RT RIC Platform may be designed to support both the Network API approach and the SDK approach and so xApps implemented using either approach may access platform services. Both of these cases are presented in figure 7.1.5-2.



##### Figure 7.1.5-2: Near-RT RIC platform supporting both Network API and SDK approaches

## Void

## Void

## Void

## Void

## Void

## Void

# External Interfaces of Near-RT RIC

## E2 Interface

O-RAN-WG3.E2GAP [2] specifies E2 interface general aspects and principles. O-RAN-WG3.E2AP [3] specifies E2 interface application protocols.

## A1 Interface

O-RAN-WG2.A1.GA&P [8] specifies A1 interface general aspects and principles. O-RAN-WG2.A1AP [9] specifies A1 interface application protocols.

## O1 Interface

An implementation of O1 interface at Near-RT RIC depends on the deployment options described in [4], i.e., when Near-RT RIC is modelled as a stand-alone Managed Element.

O-RAN-WG1.O1-Interface [5] specifies O1 interface related aspects.

## Y1 Interface

Y1 interface allows the Y1 consumers to subscribe to or request the RAN analytics information service(s) provided by Near- RT RIC [10].

NOTE: The Stage 2 and Stage 3 aspects of Y1 interface are not addressed in the present document.

# Near-RT RIC API Procedures

## Disclaimer

Procedures described in this clause assume the Near-RT RIC platform contains the functionalities listed in clause 6.2. API endpoints associated with the platform functionalities may vary depending on implementation.

## A1 Related API Procedures

### Introduction

The following procedures are described in this clause:

##### Policy Management Service.

In this case the A1 Service Producer role (A1AP [9] clause 3.1) resides in the Near-RT RIC.

* 1. Non-RT RIC (A1 Service Consumer) originated procedures:
     + Create Single Policy procedure (A1AP [9] clause 3.2.2.2.2) using **A1 Related API: A1 Policy Setup** (request, result);
     + Query Single Policy procedure (A1AP [9] clause 3.2.2.3.2) using **A1 Related API: A1 Policy Query** (request, result);
     + Query All Policy Identifiers procedure (A1AP [9] clause 3.2.2.3.5) using **A1 Related API: A1 Policy Query**

(request, result);

* + - Query Policy Status procedure (A1AP [9] clause 3.2.2.3.6) using **A1 Related API: A1 Policy Query** (request, result);
    - Update Single Policy procedure (A1AP [9] clause 3.2.2.4.2) using **A1 Related API: A1 Policy Update** (request, result);
    - Delete Single Policy procedure (A1AP [9] clause 3.2.2.5.2) using **A1 Related API: A1 Policy Delete** (request, result);
    - Query All Policy Type Identifiers procedure (A1AP [9] clause 3.2.2.7.2) using **A1 Related API: A1 Policy Query**

(request, result);

* + - Query Single Policy Type procedure (A1AP [9] clause 3.2.2.7.3) using **A1 Related API: A1 Policy Query** (request, result).
  1. Near-RT RIC (A1 Service Producer) originated procedures:
     + Policy Status Update procedure (A1AP [9] clause 3.2.2.6.2) using **A1 Related API: A1 Policy Status** (status, ack).

##### A1AP Enrichment Information (EI) Service

In this case the A1 Service Producer role (A1AP [9] clause 3.1) resides in the Non-RT RIC.

* 1. Near-RT RIC (A1 Service Consumer) originated procedures:
     + Query EI Type Identifiers procedure (A1AP [9] clause 3.3.2.2.2) using **A1 Related API: A1 EI Query** (request, result);
     + Query EI Type procedure (A1AP [9] clause 3.3.2.2.3) using **A1 Related API: A1 EI Query** (request, result);
     + Query EI Job Identifiers procedure (A1AP [9] clause 3.3.3.2.2) using **A1 Related API: A1 EI Query** (request, result);
     + Create EI Job procedure (A1AP [9] clause 3.3.3.3.2) using **A1 Related API: A1 EI Job Setup** (request, result);
     + Query EI Job procedure (A1AP [9] clause 3.3.3.3.3) using **A1 Related API: A1 EI Query** (request, result);
     + Update EI Job procedure (A1AP [9] clause 3.3.3.3.4) using **A1 Related API: A1 EI Job Update** (request, result);
     + Delete EI Job procedure (A1AP [9] clause 3.3.3.3.5) using **A1 Related API: A1 EI Job Delete** (request, result);
     + Query EI Job Status procedure (A1AP [9] clause 3.3.3.4.2) using **A1 Related API: A1 EI Job Status Query**

(request, result).

* 1. Non-RT RIC (A1 Service Producer) originated procedures:
     + Notify EI Job Status procedure (A1AP [9] clause 3.3.3.4.3) using **A1 Related API: A1 EI Job Status Notify** (status, ack);
     + Deliver EI Job Result procedure (A1AP [9] clause 3.3.4.2.2) using **A1 Related API: A1 EI Job Result Delivery**

(result, ack).

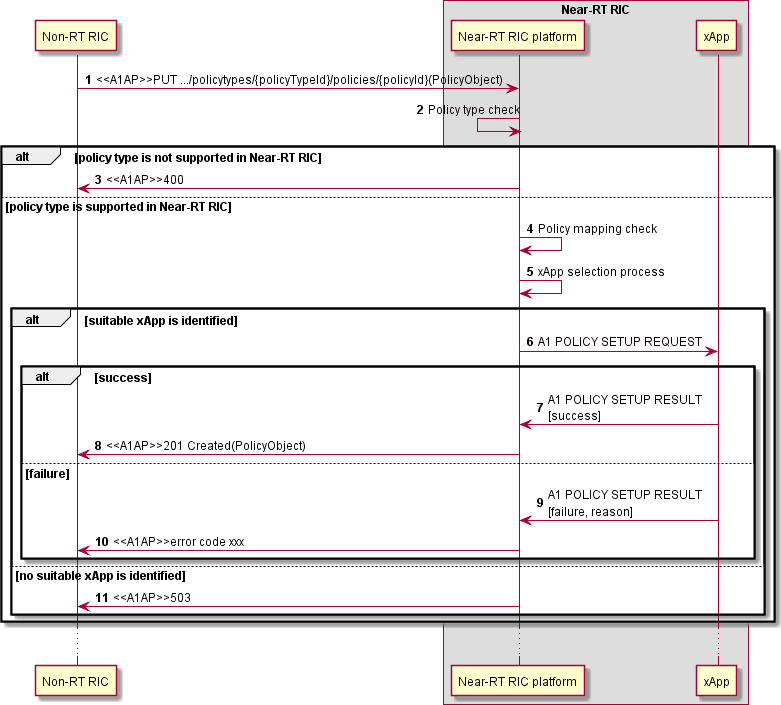
### A1 Policy procedures

#### A1 Policy Setup procedure

The purpose of A1 Policy Setup procedure in Near-RT RIC is to support the A1AP [9] Create Single Policy procedure from the Non-RT RIC and to request and activate an A1 policy enforcement toward a suitable xApp.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | The activation of a received but not activated A1 policy in Near-RT RIC. |
| Actors and Roles | * Non-RT RIC: originator of HTTP PUT request to activate an A1 policy; * xApp: responsible for the enforcement of assigned A1 policies; * Near-RT RIC platform, with the following functionalities:   + A1 Termination;   + xApp Repository. |
| Assumptions | * A1 policy types supported in Near-RT RIC are accessible by the Near-RT RIC platform; * A1 policies created in Near-RT RIC are accessible by the Near-RT RIC platform; * xApp selection needed information is accessible by the Near-RT RIC platform. xApp selection process is performed in the Near-RT RIC platform. |
| Pre conditions | * Near-RT RIC is running normally; * A1 interface is connected and activated. |
| Begins when | Non-RT RIC determines to trigger the activation of one A1 policy towards the Near-RT RIC. |
| Step 1 (M) | Non-RT RIC requests Near-RT RIC to set up an A1 policy. |
| Step 2 (M) | Near-RT RIC platform checks whether the received policy type is supported in Near-RT RIC or  not. |
|  | [ALT] Step 3 is executed if the received policy type is not supported in Near-RT RIC: |
| Step 3 (M) | A “400” response is directly returned to Non-RT RIC. |
|  | [ELSE] Steps 4-11 are executed if the received policy type is supported in Near-RT RIC: |
| Step 4 (M) | Near-RT RIC platform then checks whether the received A1 policy has been mapped to any  xApp or not. If not, Near-RT RIC platform tries to identify suitable xApp(s). (NOTE 1) |
| Step 5 (M) | Near-RT RIC platform selects suitable xApp(s). |
|  | [ALT] Steps 6-10 are executed if one or more suitable xApps are identified: |
| Step 6 (M) | Near-RT RIC platform selects a suitable xApp as the target of **A1 related API: A1**  **POLICY SETUP REQUEST** message. The message includes at least the corresponding A1 policy received from Non-RT RIC. |
|  | [ALT] Steps 7-8 are executed if the selected xApp accomplishes setup of the A1 policy: |
| Step 7-8 (M) | A success is notified to Near-RT RIC platform in **A1 related API: A1 POLICY SETUP**  **RESULT** message, and eventually to Non-RT RIC. |
|  | [ELSE] Steps 9-10 are executed if the selected xApp fails to set up the A1 policy: |
| Step 9-10 (M) | A failure is notified to Near-RT RIC platform in **A1 related API: A1 POLICY SETUP**  **RESULT** message, and eventually to Non-RT RIC. |

|  |  |
| --- | --- |
|  | [ELSE] Step 11 is executed if no suitable xApp is identified: |
| Step 11 (M) | A “503” response is directly returned to Non-RT RIC. |
| NOTE 1: The create policy and update policy procedure in A1 interface has the identical HTTP PUT Request, so when a HTTP PUT Request is received, Near-RT RIC platform cannot decide whether it’s asked for create or update from a first glance. Only after further A1 policy mapping check to find out the enforcement status of the received A1 policy, can the purpose of the received HTTP PUT request be identified. If the received A1 policy is not mapped to any xApp, Near-RT RIC platform triggers A1 Policy Setup procedure as shown  here. Otherwise, Near-RT RIC platform triggers A1 Policy Update procedure. See 9.2.2.2 for details. | |

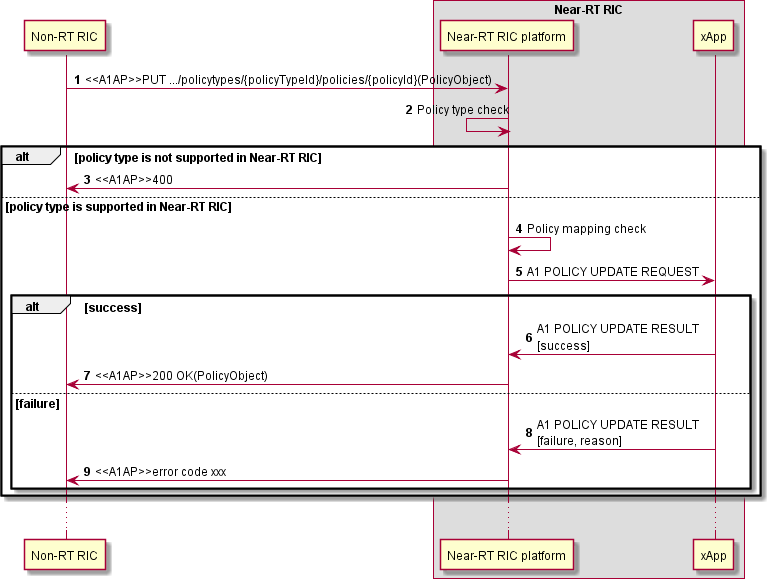


##### Figure 9.2.2.1-1: A1 Policy Setup procedure

#### A1 Policy Update procedure

The purpose of A1 Policy Update procedure in Near-RT RIC is to support A1AP Update Single Policy procedure from Non- RT RIC.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | The update of a received and already activated A1 policy in Near-RT RIC. |
| Actors and Roles | * Non-RT RIC: originator of HTTP PUT request to update an A1 policy; * xApp: responsible for the enforcement of assigned A1 policies; * Near-RT RIC platform, with the following functionalities:   + A1 Termination. |
| Assumptions | * A1 policy types supported in Near-RT RIC are accessible by the Near-RT RIC platform; * A1 policies created in Near-RT RIC are accessible by the Near-RT RIC platform. |
| Pre conditions | * Near-RT RIC is running normally; * A1 interface is connected and activated. |
| Begins when | Non-RT RIC determines to trigger the update of one A1 policy towards the Near-RT RIC. |
| Step 1 (M) | Non-RT RIC requests Near-RT RIC to update an A1 policy. |
| Step 2 (M) | Near-RT RIC platform checks whether the received policy type is supported in Near-RT RIC or  not. |
|  | [ALT] Step 3 is executed if the received A1 policy type is not supported in Near-RT RIC: |
| Step 3 (M) | A “400” response is directly returned to Non-RT RIC. |
|  | [ELSE] Steps 4-9 are executed if the received policy type is supported in Near-RT RIC: |
| Step 4 (M) | Near-RT RIC platform then checks whether the received A1 policy is mapped to any xApp or  not. |
| Step 5 (M) | (NOTE 1) If the received A1 policy is already mapped to one xApp, Near-RT RIC platform selects the xApp as the target of **A1 related API: A1 POLICY UPDATE REQUEST** message. The message includes at least the corresponding A1 policy received from Non-RT  RIC. |
|  | [ALT] Steps 6-7 are executed if the target xApp accomplishes update of the A1 policy: |
| Step 6-7 (M) | A success is notified to Near-RT RIC platform in **A1 related API: A1 POLICY UPDATE**  **RESULT** message, and eventually to Non-RT RIC. |
|  | [ELSE] Steps 8-9 are executed if the target xApp fails to update the A1 policy: |
| Step 8-9 (M) | A failure is notified to Near-RT RIC platform in **A1 related API: A1 POLICY UPDATE**  **RESULT** message, and eventually to Non-RT RIC. |
| NOTE 1: The create policy and update policy procedure in A1 interface has the identical HTTP PUT Request, so when a HTTP PUT Request is received, Near-RT RIC platform cannot decide whether it’s asked for create or update from a first glance. Only after further A1 policy mapping check to find out the enforcement status of the received A1 policy, can the purpose of the received HTTP PUT request be identified. If the received A1 policy is already mapped to an xApp, Near-RT RIC platform triggers A1 Policy Update procedure as shown  here. Otherwise, Near-RT RIC platform triggers A1 Policy Setup procedure. See 9.2.2.1 for details. | |

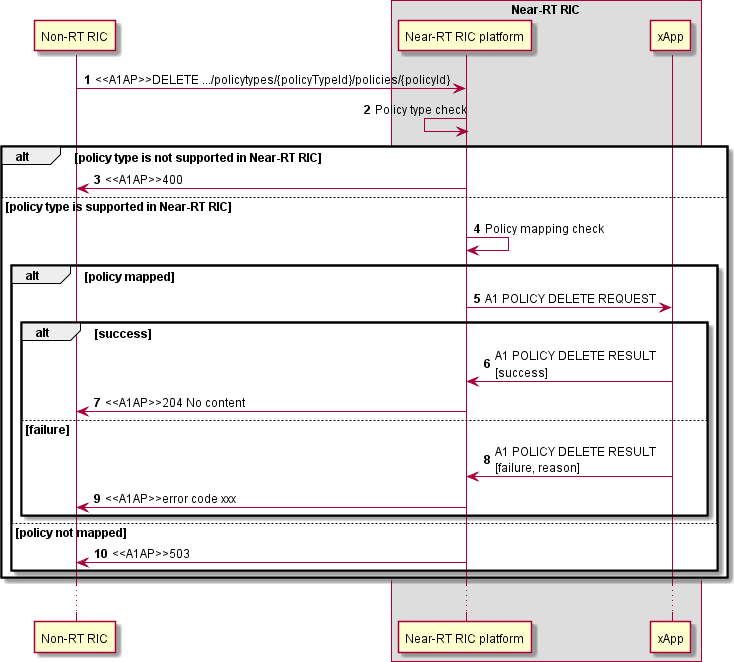


##### Figure 9.2.2.2-1: A1 Policy Update procedure

#### A1 Policy Delete procedure

The purpose of A1 Policy Delete procedure is to support the A1AP Delete Policy procedure from Non-RT RIC.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | The deactivation of a received and already activated A1 policy in Near-RT RIC. |
| Actors and Roles | * Non-RT RIC: originator of HTTP DELETE request to deactivate an A1 policy; * xApp: responsible for the deactivation of created A1 policies; * Near-RT RIC platform, with the following functionalities:   + A1 Termination. |
| Assumptions | * A1 policy types supported in Near-RT RIC are accessible by Near-RT RIC platform; * A1 policies created in Near-RT RIC are accessible by Near-RT RIC platform. |
| Pre conditions | * Near-RT RIC is running normally. * A1 interface is connected and activated. |
| Begins when | Non-RT RIC determines to trigger the deactivation of one A1 policy towards the Near-RT RIC. |
| Step 1 (M) | Non-RT RIC requests Near-RT RIC to delete an A1 policy. |
| Step 2 (M) | Near-RT RIC platform checks whether the received policy type is supported in Near-RT RIC or  not. |
|  | [ALT] Step 3 is executed if the received policy type is not supported in Near-RT RIC: |
| Step 3 (M) | A “400” response is directly returned to Non-RT RIC. |
|  | [ELSE] Steps 4-10 are executed if the received policy type is supported in Near-RT RIC: |
| Step 4 (M) | Near-RT RIC platform then checks whether the received A1 policy is mapped to any xApp or  not. |
|  | [ALT] Steps 5-9 are executed if the received A1 policy is already mapped to one xApp: |
| Step 5 (M) | Near-RT RIC platform selects the xApp as the target of **A1 related API: A1 POLICY DELETE REQUEST** message. The message includes at least the corresponding A1  policy ID received from Non-RT RIC. |
|  | [ALT] Steps 6-7 are executed if the target xApp accomplishes deletion of the A1 policy: |
| Step 6-7 (M) | A success is notified to Near-RT RIC platform in **A1 related API: A1 POLICY**  **DELETE RESULT** message, and eventually to Non-RT RIC. |
|  | [ELSE] Steps 8-9 are executed if the target xApp fails to delete the A1 policy: |
| Step 8-9 (M) | A failure is notified to Near-RT RIC platform in **A1 related API: A1 POLICY DELETE**  **RESULT** message, and eventually to Non-RT RIC. |
|  | [ELSE] Step 10 is executed if the received A1 policy is not mapped to any xApp: |
| Step 10 (M) | A “503” response is directly returned to Non-RT RIC. |



##### Figure 9.2.2.3-1: A1 Policy Delete procedure

#### A1 Policy Query procedure

The purpose of A1 Policy Query procedure is to support the following A1AP procedures initiated by the Non-RT RIC:

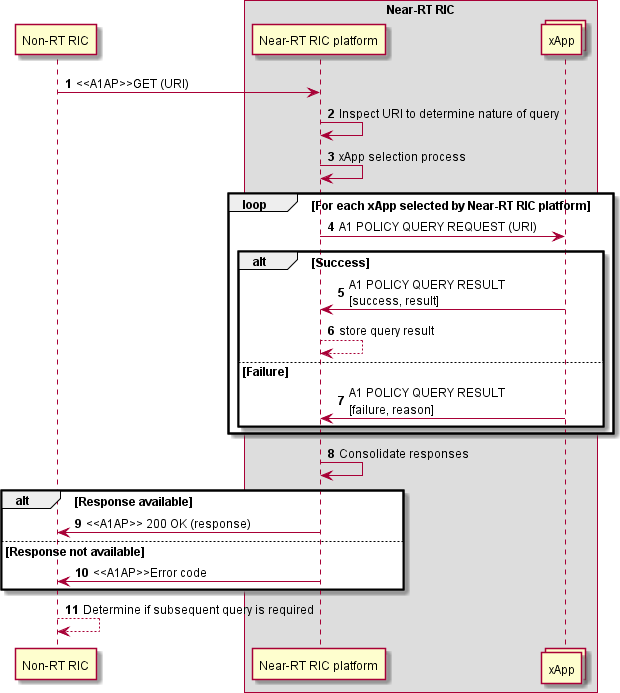
* + - * + A1AP Query Single Policy procedure;
        + A1AP Query All Policy Identifiers procedure;
        + A1AP Query Policy Status procedure;
        + A1AP Query All Policy Type Identifiers procedure;
        + A1AP Query Single Policy Type procedure.

Table 9.2.2.4-1 provides the list of expected xApp response for each Near-RT RIC Platform Policy query. The query may come from Near-RT platform or from the Non-RT RIC.

##### Table 9.2.2.4-1: Supported A1AP Policy Query procedures

|  |  |  |
| --- | --- | --- |
| **Query procedure** | **Platform Request** | **xApp Response** |
| Query Single Policy | URI containing policyId | PolicyObject |
| Query All Policy Identifiers | URI containing policyTypeId | list of current policyId |
| Query Status | URI containing policyId | PolicyStatusObject |
| Query All Policy Type Identifiers | URI | list of currently supported PolicyTypeId |
| Query Single Policy Type | URI containing PolicyTypeId | PolicyTypeObject |

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | To obtain from an xApp information concerning A1 Policies, outcome depends on nature of Request:   * Query Single Policy: xApp provides PolicyObject for a given policyId; * Query All Policy Identifiers: xApp provides the list of current policyId for a given policyTypeId; * Query Status: xApp provides current PolicyStatusObject for a given policyId; * Query All Policy Type Identifiers: xApp provides list of currently supported PolicyTypeId; * Query Single Policy Type: xApp provides PolicyTypeObject of a given PolicyTypeId. |
| Actors and Roles | * Non-RT RIC: originator of query using HTTP GET request; * xApp: provides appropriate response corresponding to nature of request; * Near-RT RIC platform, with the following functionalities:   + A1 Termination;   + xApp Repository. |
| Assumptions | * xApp selection needed information is accessible by Near-RT RIC platform. xApp selection process is performed at Near-RT RIC platform; * Near-RT RIC platform may store the results of the xApp selection process * Near-RT RIC platform may hold information related to Policy Query procedures. |
| Pre conditions | * Near-RT RIC is running normally; * A1 interface is connected and activated. |
| Begins when | Non-RT RIC determines need to trigger A1 query towards the Near-RT RIC. |
| Step 1 (M) | Non-RT RIC sends to Near-RT RIC one of the A1AP Policy Management service related Query  procedures. |
| Step 2 (M) | Near-RT RIC platform inspects content of HTTP GET to determine nature of Query. |
| Step 3 (M) | Near-RT RIC platform selects suitable xApp(s) to collect information for the query. |
|  | [LOOP] Steps 4-6 are looped for each xApp selected by Near-RT RIC platform |
| Step 4 (O) | Near-RT RIC platform may send **A1 related API: A1 POLICY QUERY REQUEST** message.  The message includes at least the corresponding A1 policy management service query resource URI received from Non-RT RIC. |
| Step 5 (O) | If the selected xApp(s) accomplishes the A1 policy management service query, a success is notified to Near-RT RIC platform in **A1 related API: A1 POLICY QUERY RESULT** message.  Near-RT RIC platform may store the result. |
| Step 6 (O) | If the selected xApp(s) fails to treat the A1 policy management service query, a failure is  notified to Near-RT RIC platform in **A1 related API: A1 POLICY QUERY RESULT** message. |
| Step 7 (M) | Near-RT RIC platform collects the responses from each xApp and/or its previously stored  information, and consolidates into a response to Non-RT RIC. |
| Step 8 (M) | If A1 policy query response available, Near-RT RIC platform sends the appropriate A1AP Policy  Management Service response to Non-RT RIC. |
| Step 9 (M) | If A1 policy query response not available, Near-RT RIC platform sends the appropriate A1AP  Policy Management Service error to Non-RT RIC (A1AP [9] clause 4.2.7.3). |
| Step 10 (O) | Depending upon the nature of the previous A1AP Policy Management Service query and the  response from the Near-RT RIC, the Non-RT RIC may initiate a subsequent A1AP Policy Management Service query. |

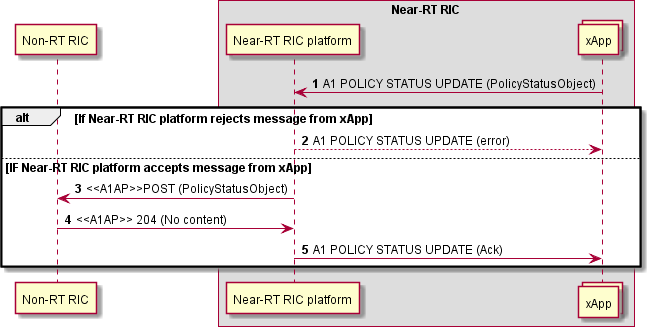


##### Figure 9.2.2.4-1: A1 Policy Query procedure

#### A1 Policy Status Update procedure

The purpose of A1 Policy Status Update procedure in Near-RT RIC is to support the A1AP Policy Status Update procedure (A1AP [9] clause 3.2.2.6) from the Near-RT RIC.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | The activation of a received but not activated A1 policy in Near-RT RIC. |
| Actors and Roles | * Non-RT RIC: receives the HTTP POST containing PolicyStatusObject; * xApp: originator of the PolicyStatusObject content to be sent to Non-RT RIC; * Near-RT RIC platform, with the following functionalities:   + A1 Termination. |
| Assumptions | * A1 policy types supported in Near-RT RIC are accessible by Near-RT RIC platform; * A1 policies created in Near-RT RIC are accessible by Near-RT RIC platform; * xApp selection needed data is accessible by the Near-RT RIC platform. xApp selection process is performed at the Near-RT RIC platform. |
| Pre conditions | * Near-RT RIC is running normally; * A1 interface is connected and activated. |
| Begins when | xApp determines need to trigger the A1 policy status update to be sent to the Non-RT RIC. |
| Step 1 (M) | xApp sends **A1 related API: A1 POLICY STATUS UPDATE** (PolicyStatusObject information  encoded as A1TD) message to Near-RT RIC platform. |
|  | [ALT] Step 2 is executed if Near-RT RIC platform rejects request: |
| Step 2 (M) | Near-RT RIC platform sends A1 Related **POLICY STATUS UPDATE (Error)** to xApp. |
|  | [ELSE] Step 3-5 are executed if Near-RT RIC platform accepts request: |
| Step 3 (M) | Near-RT RIC platform forwards content within A1AP Policy Status Update message. |
| Step 4-5 (M) | Non-RT RIC sends A1AP acknowledgement, and Near-RT RIC platform forwards to xApp  using **A1 Related POLICY STATUS UPDATE (Ack).** |



##### Figure 9.2.2.5-1: A1 Policy Status Update procedure

### A1-EI related procedures

#### A1-EI Query procedure

The purpose of A1 EI Query procedure is to support the following A1AP procedures initiated by the Near-RT RIC:

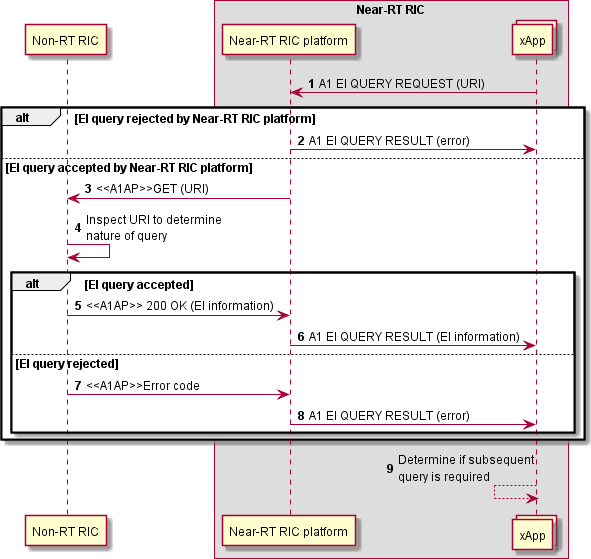
* + - * + Query EI Type Identifiers procedure;
        + Query EI Type procedure;
        + Query EI Job Identifiers procedure;
        + Query EI Job procedure;
        + Query EI Job Status procedure.

Table 9.2.3.1-1 provides the expected response from Near-RT RIC platform for each EI query request. The information may come from the platform or the Non-RT RIC.

##### Table 9.2.3.1-1: Supported A1AP Query EI procedures

|  |  |  |
| --- | --- | --- |
| **Query procedure** | **xApp request** | **Platform Response** |
| Query EI Type Identifiers | URI containing EiTypes | List of EiTypeId |
| Query EI Type | URI containing EiTypeId | List of EiTypeObject |
| Query EI Job Identifiers | URI containing EiJobs | List of current EiJobId |
| Query EI Job | URI containing EiJobId | EiJobObject |
| Query EI Job Status | URI containing EiJobId | EiJobStatusObjec |

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | To obtain from Non-RT RIC information concerning A1 EI, outcome depends on nature of Request:   * Query EI Type Identifiers: Non-RT RIC provides list of EiTypeId for a given EiTypes; * Query EI Type: Non-RT RIC provides EiTypeObject for a given EiTypeId; * Query EI Job Identifiers: Non-RT RIC provides list of current EiJobId for a given EiJobs; * Query EI Job: Non-RT RIC provides EiJobObject for a given EiJobId; * Query EI Job Status: Non-RT RIC provides EiJobStatusObject for a given EiJobId with flag Status. |
| Actors and Roles | * Non-RT RIC: provides appropriate response corresponding to nature of request; * xApp: originator of query using HTTP GET request; * Near-RT RIC platform, with the following functionalities:   + A1 Termination. |
| Assumptions | * For A1-EI query, Near-RT RIC platform is stateless and so does not store previous Query responses from Non-RT RIC |
| Pre conditions | * Near-RT RIC is running normally. * A1 interface is connected and activated. |
| Begins when | xApp determines need to trigger A1 EI query towards the Non-RT RIC. |
| Step 1 (M) | xApp sends **A1 Related EI QUERY REQUEST (URI)** to Near-RT RIC platform. |
|  | [ALT] Step 2 is executed if Near-RT RIC platform rejects the request: |
| Step 2 (O) | Near-RT RIC platform may reject request and send **A1 Related EI QUERY RESULT (error**  **code)** to xApp. |
|  | [ELSE] Steps 3-8 are executed if Near-RT RIC platform accepts the request: |
| Step 3 (M) | Near-RT RIC platform builds appropriate A1AP EI Query message including URI and sends  to non-RT RIC. |
| Step 4 (M) | Non-RT RIC inspects URI to determine nature of query and it is acceptable. |
|  | [ALT] Steps 5-6 are executed if Non-RT RIC accepts the query: |
| Step 5-6 (M) | Non-RT RIC builds appropriate response (EI information encoded using A1TD) and sends to Near-RT RIC platform, Near-RT RIC platform uses **A1 Related EI QUERY RESULT** to  send response to xApp (with EI information encoded using A1TD). |
|  | [ELSE] Steps 7-8 are executed if Non-RT RIC rejects the query: |
| Step 7-8 (M) | Non-RT RIC builds appropriate response (error code) and sends to Near-RT RIC platform, Near-RT RIC platform used **A1 Related EI QUERY RESULT (error code)** to  send response to xApp. |
| Step 9 (O) | Depending upon the nature of the previous A1AP EI query and the response from the Non-RT  RIC, the xApp may initiate a subsequent A1AP EI query. |



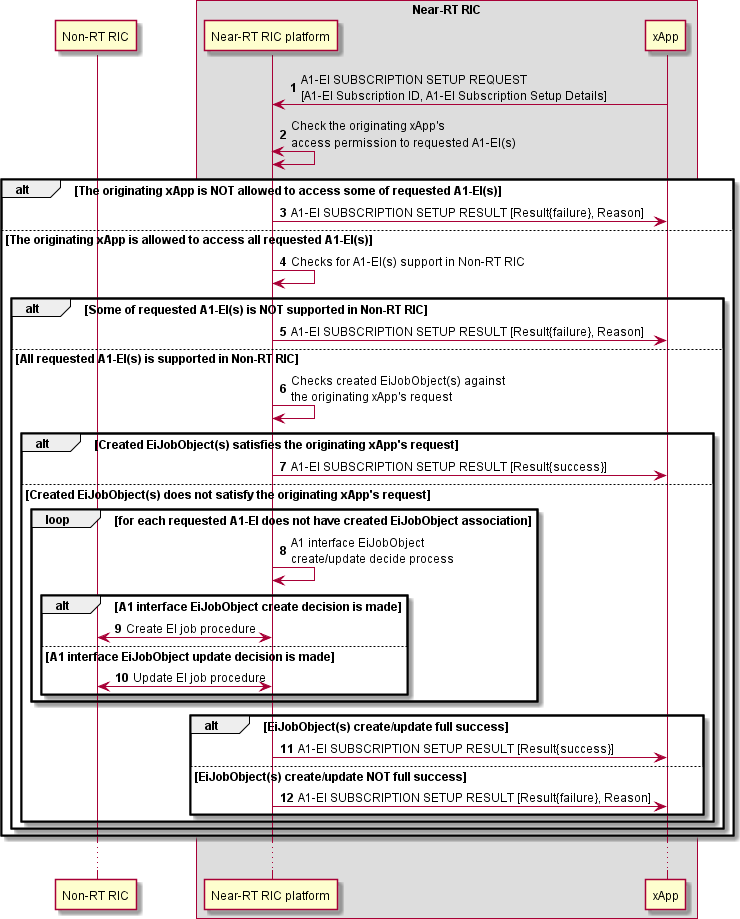
##### Figure 9.2.3.1-1: A1 EI Query procedure

#### A1-EI subscription setup procedure

The A1-EI subscription setup procedure enables xApp to setup a new A1-EI subscription toward Near-RT RIC Platform.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | To setup a new A1-EI subscription between xApp and Near-RT RIC Platform. |
| Actors and Roles | * Non-RT RIC: A1-EI producer. * xApp: requester of AI-EI subscription setup. * Near-RT RIC Platform, with the following functionalities:   + A1 Termination.   + xApp Repository. |
| Assumptions | * Created EiJobObjects in Near-RT RIC is readable by Near-RT RIC platform. * Non-RT RIC’s supported A1-EIs record in Near-RT RIC is readable by Near-RT RIC platform. * xApp subscribe multiple A1-EIs in integrated manner, i.e., subscribed A1-EIs indicated by **A1 related API: A1-EI SUBSCRIPTION SETUP REQUEST** message are dependent on each other, and only full success of all A1-EI subscriptions is viewed as message process success. * xApp's access permission to A1-EI is handled by Near-RT RIC platform. |
| Pre-conditions | * Near-RT RIC Platform is running normal. * Near-RT RIC is connected to Non-RT RIC via A1 interface. |
| Begins when | xApp determines to request subscription of A1-EI(s). |
| Step 1 (M) | xApp requests Near-RT RIC Platform to set up an A1-EI subscription by sending **A1 related**  **API: A1-EI SUBSCRIPTION SETUP REQUEST** message, with A1-EI subscription setup details included. |
| Step 2 (M) | Near-RT RIC platform checks whether the originating xApp is allowed to access all the  requested A1-EI(s). |
|  | [ALT] Step 3 is executed if the originating xApp is not allowed to access some of the requested  A1-EIs. |
| Step 3 (M) | A failure is responded to the originating xApp via **A1 related API: A1-EI SUBSCRIPTION**  **SETUP RESULT** message, possibly including appropriate reason. |
|  | [ELSE] Steps 4-12 are executed if the originating xApp is allowed to access all the requested A1-  EIs. |
| Step 4 (M) | Near-RT RIC platform checks whether all the requested A1-EIs is supported in Non-RT RIC. |
|  | [ALT] Step 5 is executed if some of the requested A1-EI(s) is NOT supported in Non-RT RIC. |
| Step 5 (M) | A failure is responded to the originating xApp via **A1 related API: A1-EI SUBSCRIPTION**  **SETUP RESULT** message, possibly including appropriate reason. |
|  | [ELSE] Steps 6-12 are executed if all requested A1-EI(s) is supported in Non-RT RIC. |
| Step 6 (M) | Near-RT RIC platform checks whether created EiJobObject(s) satisfies the originating  xApp’s request. |
|  | [ALT] Step 7 is executed if created EiJobObject(s) satisfies the originating xApp’s request. |
| Step 7 (M) | Near-RT RIC platform associates the identified created EiJobObject with the originating xApp, and responds success to the originating xApp via **A1 related API:**  **A1-EI SUBSCRIPTION SETUP RESULT** message. |
|  | [ELSE] Steps 8-12 are executed if created EiJobObject(s) does not satisfy the originating  xApp’s request. |
|  | [LOOP] Steps 8-10 are looped for each subscribed A1-EI that does not have  EiJobObject association |
| Step 8 (M) | Near-RT RIC platform makes A1 interface EiJobObject either create or update  decision. |
|  | [ALT] Step 9 is executed if A1 interface EiJobObject create decision is made. |
| Step 9 (M) | Near-RT RIC platform initiates create EI job procedure towards Non-RT RIC  and receives response from Non-RT RIC. |
|  | [ELSE] Step 10 is executed if A1 interface EiJobObject update decision is made. |
| Step 10 (M) | Near-RT RIC platform initiates update EI job procedure towards Non-RT RIC  and receives response from Non-RT RIC. |
|  | [ALT] Step 11 is executed if A1 interface EiJobObject create/update process shots full  success. |
| Step 11 (M) | Near-RT RIC platform responds success to the originating xApp via **A1 related**  **API: A1-EI SUBSCRIPTION SETUP RESULT** message. |
|  | [ELSE] Step 12 is executed if A1 interface EiJobObject create/update process shots  not full success. |

|  |  |
| --- | --- |
| Step 12 (M) | Near-RT RIC platform responds failure to the originating xApp via **A1 related API:**  **A1-EI SUBSCRIPTION SETUP RESULT** message, possibly including appropriate reason. |



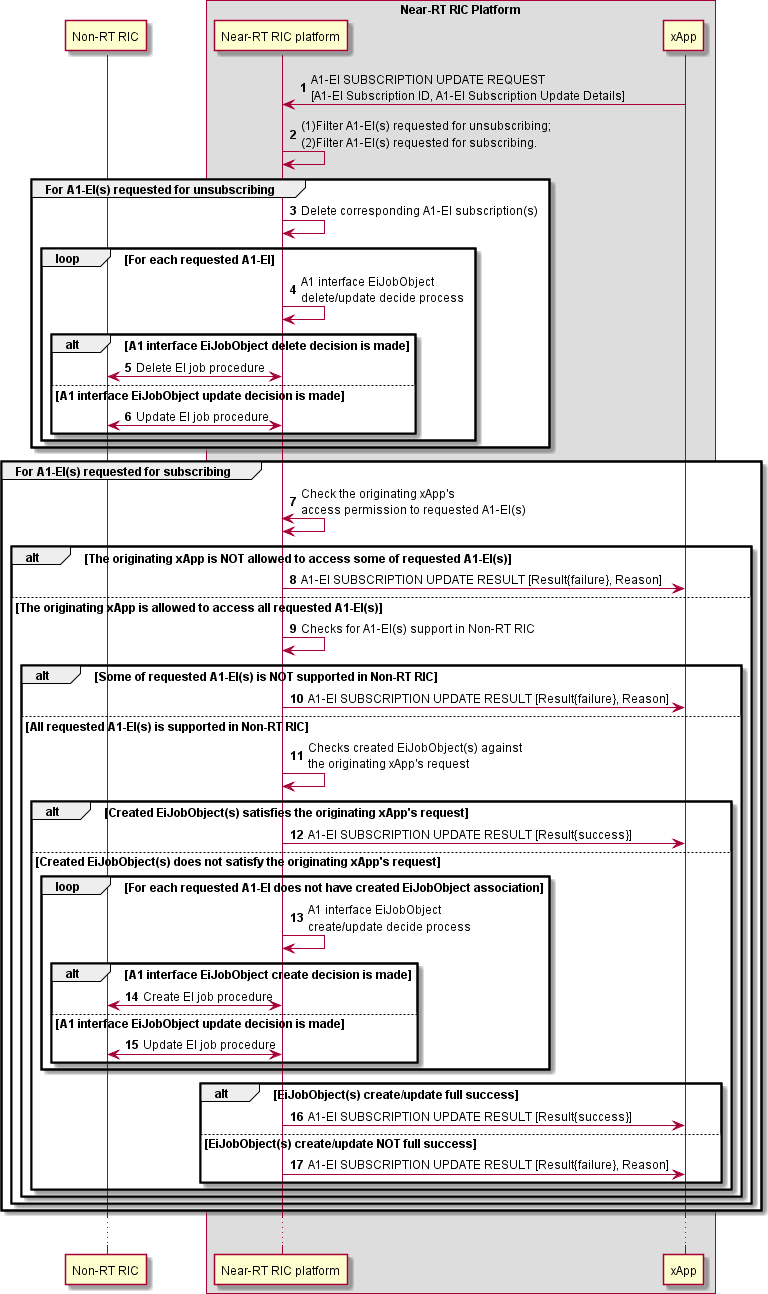
##### Figure 9.2.3.2-1: A1-EI subscription setup procedure

#### A1-EI subscription update procedure

The A1-EI subscription update procedure enables xApp to update an existing A1-EI subscription toward Near-RT RIC Platform.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | To update an existing A1-EI subscription between xApp and Near-RT RIC Platform. |
| Actors and Roles | * Non-RT RIC: A1-EI producer. * xApp: requester of AI-EI subscription update. * Near-RT RIC Platform, with the following functionalities:   + A1 Termination.   + xApp Repository. |
| Assumptions | * Created EiJobObjects in Near-RT RIC is readable by Near-RT RIC platform. * Non-RT RIC’s supported A1-EIs record in Near-RT RIC is readable by Near-RT RIC platform. * xApp subscribe multiple A1-EIs in integrated manner, i.e., subscribed A1-EIs indicated by **A1 related API: A1-EI SUBSCRIPTION SETUP REQUEST** message are dependent on each other, and only full success of all A1-EI subscriptions is viewed as message process success. * xApp's access permission to A1-EI is handled by Near-RT RIC platform. |
| Pre-conditions | * Near-RT RIC Platform is running normal. * Near-RT RIC is connected to Non-RT RIC via A1 interface. |
| Begins when | xApp determines to request update of one A1-EI subscription. |
| Step 1 (M) | xApp requests Near-RT RIC Platform to update an A1-EI subscription by sending **A1 related API: A1-EI SUBSCRIPTION UPDATE REQUEST** message, with A1-EI subscription update  details included. |
| Step 2 (M) | From **A1 related API: A1-EI SUBSCRIPTION UPDATE REQUEST** message, Near-RT RIC platform filters A1-EI(s) requested for 1) unsubscribing, and 2) subscribing (including new  subscription needs creation and current subscription needs update). |
|  | For A1-EI(s) requested for unsubscribing, execute steps 3-6. |
| Step 3 (M) | Near-RT RIC platform deletes corresponding A1-EI subscription(s). |
|  | [LOOP] Steps 4-6 are looped for each requested A1-EI |
| Step 4 (M) | Near-RT RIC platform makes A1 interface EiJobObject either delete or update decision. |
|  | [ALT] Step 5 is executed if A1 interface EiJobObject delete decision is made. |
| Step 5 (M) | Near-RT RIC platform initiates delete EI job procedure towards Non-RT RIC and receives  response from Non-RT RIC. |
|  | [ELSE] Step 6 is executed if A1 interface EiJobObject update decision is made. |
| Step 6 (M) | Near-RT RIC platform initiates update EI job procedure towards Non-RT RIC and  receives response from Non-RT RIC. |
|  | For A1-EI(s) requested for subscribing, execute steps 7-17. |
| Step 7 (M) | Near-RT RIC platform checks whether the originating xApp is allowed to access to the  requested A1-EI(s). |
|  | [ALT] Step 8 is executed if the originating xApp is not allowed to access some of the  requested A1-EI(s). |
| Step 8 (M) | A failure is responded to the originating xApp via **A1 related API: A1-EI SUBSCRIPTION**  **UPDATE RESULT** message, possibly including appropriate reason. |
|  | [ELSE] Steps 9-17 are executed if the originating xApp is allowed to access all the requested  A1-EI(s). |
| Step 9 (M) | Near-RT RIC platform checks whether the requested A1-EI(s) is supported in Non-RT  RIC. |
|  | [ALT] Step 10 is executed if some of requested A1-EI(s) is NOT supported in Non-RT  RIC. |
| Step 10 (M) | A failure is responded to the originating xApp via **A1 related API: A1-EI**  **SUBSCRIPTION UPDATE RESULT** message, possibly including appropriate reason. |
|  | [ELSE] Steps 11-17 are executed if all requested A1-EI(s) is supported in Non-RT RIC. |
| Step 11 (M) | Near-RT RIC platform checks whether created EiJobObject(s) satisfies the originating  xApp’s request. |
|  | [ALT] Step 12 is executed if created EiJobObject(s) satisfies the originating xApp’s  request. |

|  |  |
| --- | --- |
| Step 12 (M) | Near-RT RIC platform associates the identified created EiJobObject with the  originating xApp, and responds success to the originating xApp via **A1 related API: A1-EI SUBSCRIPTION UPDATE RESULT** message. |
|  | [ELSE] Steps 13-17 are executed if created EiJobObject(s) does not satisfy the  originating xApp’s request. |
|  | [LOOP] Steps 13-15 are looped for each requested A1-EI that does not have  EiJobObject association |
| Step 13 (M) | Near-RT RIC platform makes A1 interface EiJobObject either create or update  decision. |
|  | [ALT] Step 14 is executed if A1 interface EiJobObject create decision is made. |
| Step 14 (M) | Near-RT RIC platform initiates create EI job procedure towards Non-RT  RIC and receives response from Non-RT RIC. |
|  | [ELSE] Step 15 is executed if A1 interface EiJobObject update decision is  made. |
| Step 15 (M) | Near-RT RIC platform initiates update EI job procedure towards Non-RT  RIC and receives response from Non-RT RIC. |
|  | [ALT] Step 16 is executed if A1 interface EiJobObject create/update process shots  full success. |
| Step 16 (M) | Near-RT RIC platform responds success to the originating xApp via **A1 related**  **API: A1-EI SUBSCRIPTION UPDATE RESULT** message. |
|  | [ELSE] Step 17 is executed if A1 interface EiJobObject create/update process  shots not full success. |
| Step 17 (M) | Near-RT RIC platform responds failure to the originating xApp via **A1 related API: A1-EI SUBSCRIPTION UPDATE RESULT** message, possibly including  appropriate reason. |

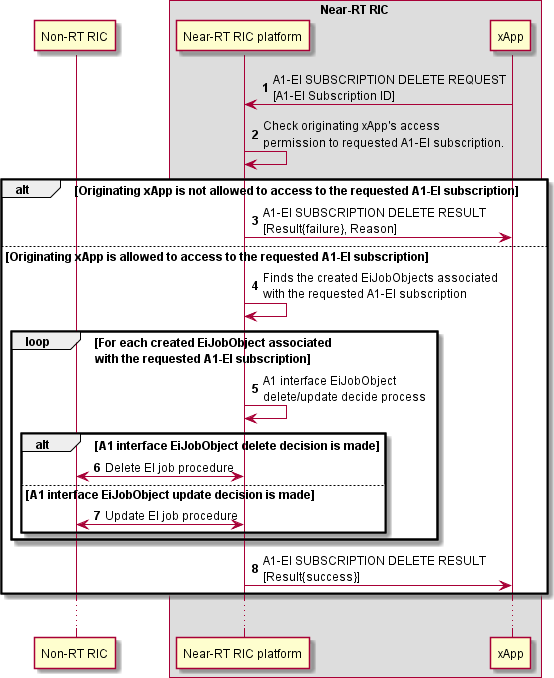


##### Figure 9.2.3.3-1: A1-EI subscription update procedure

#### A1-EI subscription delete procedure

A1-EI subscription delete procedure allows xApp to delete an existing A1-EI subscription toward Near-RT RIC Platform.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | To delete an existing A1-EI subscription between xApp and Near-RT RIC Platform. |
| Actors and Roles | * Non-RT RIC: A1-EI producer. * xApp: requester of AI-EI subscription delete. * Near-RT RIC Platform, with the following functionalities:   + A1 Termination. |
| Assumptions | * Created EiJobObjects in Near-RT RIC is readable by Near-RT RIC platform. * xApp's access permission to A1-EI subscription is handled by Near-RT RIC platform. |
| Pre-conditions | * Near-RT RIC Platform is running normal. * Near-RT RIC is connected to Non-RT RIC via A1 interface. |
| Begins when | xApp determines to request delete of one A1-EI subscription. |
| Step 1 (M) | xApp requests Near-RT RIC Platform to delete an A1-EI subscription by sending **A1 related**  **API: A1-EI SUBSCRIPTION DELETE REQUEST** message, with A1-EI subscription ID included. |
| Step 2 (M) | Near-RT RIC platform checks whether the originating xApp is allowed to access to the requested  A1-EI subscription. |
|  | [ALT] Step 3 is executed if the originating xApp is not allowed to access to the requested A1-EI  subscription. |
| Step 3 (M) | A failure is responded to the originating xApp via **A1 related API: A1-EI SUBSCRIPTION**  **DELETE RESULT** message, possibly including appropriate reason. |
|  | [ELSE] Steps 4-8 are executed if the originating xApp is allowed to access to the requested A1-  EI subscription. |
| Step 4 (M) | Near-RT RIC platform finds the created EiJobObjects associated with the requested A1-EI  subscription. |
|  | [LOOP] Steps 5-7 are looped for each created EiJobObjects associated with the requested  A1-EI subscription |
| Step 5 (M) | Near-RT RIC platform makes A1 interface EiJobObject either delete or update decision. |
|  | [ALT] Step 6 is executed if A1 interface EiJobObject delete decision is made. |
| Step 6 (M) | Near-RT RIC platform initiates delete EI job procedure towards Non-RT RIC and  receives response from Non-RT RIC. |
|  | [ELSE] Step 7 is executed if A1 interface EiJobObject update decision is made. |
| Step 7 (M) | Near-RT RIC platform initiates update EI job procedure towards Non-RT RIC and  receives response from Non-RT RIC. |
| Step 8 (M) | Near-RT RIC platform responds success to the originating xApp via **A1 related API: A1-EI**  **SUBSCRIPTION DELETE RESULT** message. |

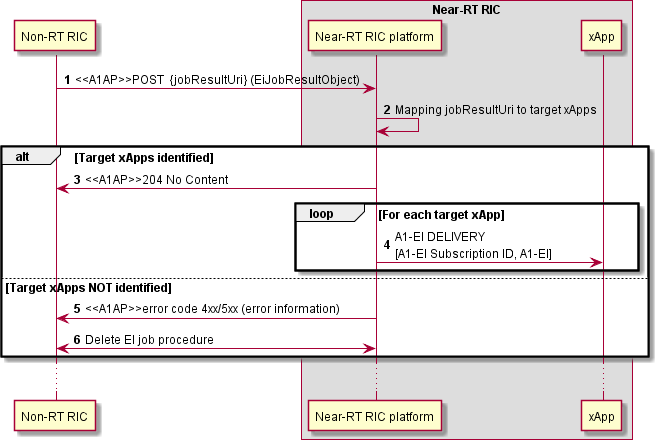


##### Figure 9.2.3.4-1: A1-EI subscription delete procedure

#### A1-EI delivery procedure

The A1-EI delivery procedure allows xApp to be deliveried with subscribed A1-EI from Near-RT RIC Platform.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | The delivery of A1-EI from Near-RT RIC Platform. |
| Actors and Roles | * Non-RT RIC: A1-EI producer. * xApp: AI-EI consumer. * Near-RT RIC Platform, with the following functionalities:   + A1 Termination. |
| Assumptions | * jobResultUri indicated by Non-RT RIC to target xApp mapping is handled by Near-RT RIC platform. |
| Pre-conditions | * Near-RT RIC Platform is running normal. * Near-RT RIC is connected to Non-RT RIC via A1 interface. |
| Begins when | Non-RT RIC has A1-EI produced for Near-RT RIC. |
| Step 1 (M) | Non-RT RIC sends a HTTP POST request to deliver the produced EiJobResultObject to Near-  RT RIC Platform. |
| Step 2 (M) | Near-RT RIC platform maps jobResultUri indicated by Non-RT RIC to target xApps. |
|  | [ALT] If at least one target xApp is identified, execute steps 3-4. |
| Step 3 (M) | Near-RT RIC platform responds Non-RT RIC with status code "204". |
|  | [LOOP] Step 4 is looped for each target xApp |
| Step 4 (M) | Near-RT RIC platform delivers the A1-EI to the target xApp by sending **A1 related API:**  **A1-EI DELIVERY** message, with associated A1-EI subscription ID included. |
|  | [ELSE] If NO target xApp is identified, execute steps 5-6. |
| Step 5 (M) | Near-RT RIC platform responds an appropriate error code and possibly error information to  Non-RT RIC. |
| Step 6 (O) | Near-RT RIC platform initiates delete EI job procedure towards Non-RT RIC to delete the  corresponding EI job and receives response from Non-RT RIC. |



##### Figure 9.2.3.5-1: A1-EI delivery procedure

## E2 Related API Procedures

### Introduction

The following procedures are described in this clause:

E2AP RIC Functional procedures:

* RIC Subscription using **E2 Related API: E2 Subscription** (request, reject, success, failure) and **E2 Related API: E2 Guidance** (request, response);
* RIC Subscription Delete using **E2 Related API: E2 Subscription Delete** (request, reject, success, failure);
* RIC Subscription Delete Required using **E2 Related API: E2 Subscription Delete Query** (required, accept, decline) and **E2 Related API: E2 Subscription Delete Notification** (Deleted);
* RIC Subscription Modification using **E2 Related API: E2 Subscription Modification** (request, reject, success, failure);
* RIC Subscription Audit using **E2 Related API: E2 Subscription Audit** (request, reject, success, failure);
* RIC Indication using **E2 Related API: E2 Indication** (push, failure);
* RIC Control using **E2 Related API: E2 Control** (request, reject, success, failure) and **E2 Related API: E2 Guidance** (request, response);
* RIC Query using **E2 Related API: E2 Query** (request, reject, success, failure).

E2 Conflict Mitigation related procedures:

* xApp initiated conflict mitigation using **E2 Related API: E2 Guidance** (request, response);
* xApp Subscription Management initiated conflict mitigation using **E2 Related API: E2 Guidance** (modification);
* Conflict mitigation related message monitoring using **E2 Related API: E2 Guidance** (modification);
* Conflict mitigation initiated conflict mitigation using **E2 Related API: E2 Guidance** (modification);
* Conflict mitigation initiated conflict detection, avoidance, and resolution related xApp assistance using **E2 Related API: E2 Conflict Mitigation Assistance** (request, response).

### RIC Functional Procedures

#### E2 Subscription procedure

The purpose of the E2 Subscription procedure in the Near-RT RIC is to enable an xApp to request subscriptions for REPORT, INSERT and/or POLICY service(s) from E2 interface, and to ensure that only validated and non-duplicate subscriptions are maintained by the Near-RT RIC over the E2 interface to the E2 Node and that duplicated E2 Subscription Request messages from xApps are handled gracefully.

This procedure is based on the following assumptions:

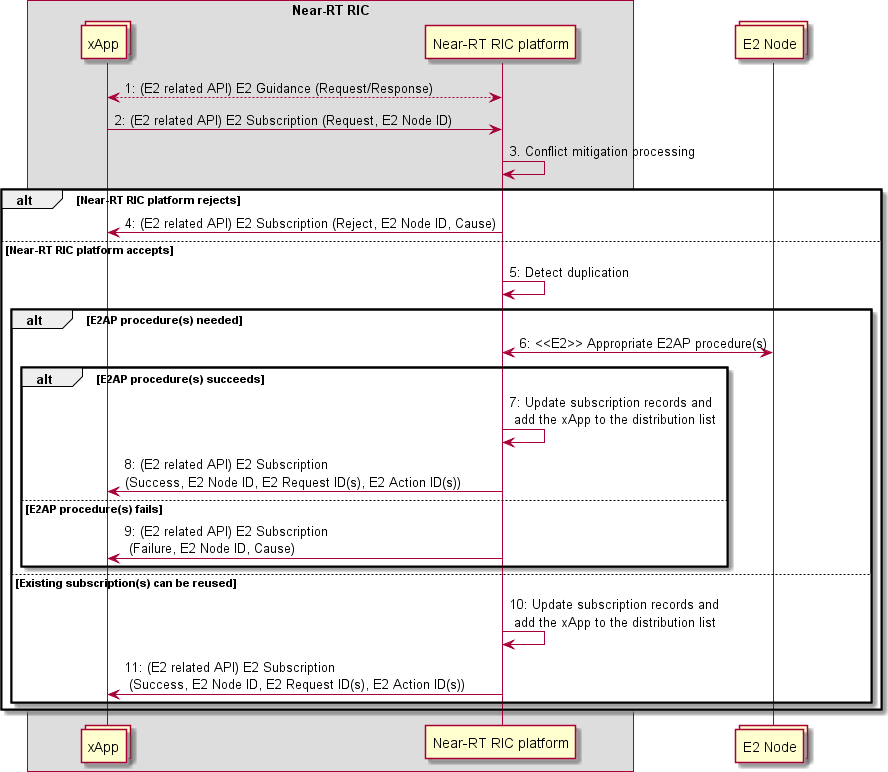
* + - * + xApp may obtain guidance from Near-RT RIC Platform (i.e., conflict mitigation processing) to resolve potential conflicts and/or detect partial duplications prior to sending a E2 Subscription Request (see clause 9.3.3.1);
        + xApp has been configured with a trusted xApp ID;
        + Near-RT RIC platform maintains all previous successful RIC Subscriptions towards the E2 Node listed in xApp’s E2 Subscription request and is able to detect duplications and recover E2 Node response messages;
        + Near-RT RIC platform may leverage its conflict mitigation functionality in managing E2 subscriptions (see clause 9.3.3.5);
        + Near-RT RIC platform decides initiation of appropriate E2AP procedure(s), if needed, to fulfil the E2 Subscription request.
        + E2 Request ID is used on E2-related APIs to identify an active RIC subscription. Near-RT Platform maintains the 1:1 mapping between E2 Request ID, used on E2 related APIs, and RIC Request ID, used on E2 interface.
        + Near-RT RIC platform routes requests for E2AP procedure(s) to appropriate E2 interface;
        + Near-RT RIC platform maintains a mapping of active Subscriptions (identified on E2 interface by E2 Node ID and RIC Request ID) of validated xApps. Near-RT RIC platform uses the mapping when sending E2 INDICATION messages to the correct xApp or xApps.
        + Near-RT RIC platform sends the E2 Subscription response prior to sending any associated E2 Indication message to the xApp.

The procedure is initiated by an xApp using **E2 Related API: E2 Subscription** request for a specific E2 Node. The following outcomes are considered:

* + - * + Request fails following rejection by Near-RT RIC platform and an **E2 related API: E2 Subscription** response (Reject) is sent to xApp without E2 interface transaction. Response contains Cause (i.e., xApp not authorized to request specific subscription);
        + Request towards a specific E2 Node is handled successfully following acceptance by Near-RT RIC platform but detected as duplicate and so acknowledged with an **E2 related API: E2 Subscription** response (Success) to xApp for a specific E2 Node without corresponding E2 interface transaction;
        + Request towards a specific E2 Node is handled successfully following acceptance by Near-RT RIC Platform and acceptance by E2 Node and so acknowledged with an **E2 related API: E2 Subscription** response (Success) to xApp for a specific E2 Node following corresponding E2 interface transaction;
        + Request towards a specific E2 Node fails following acceptance by Near-RT RIC Platform but is either rejected by E2 Node or corresponding E2AP timer expires prior to a response from E2 Node and so declined using an **E2 related API: E2 Subscription** response (Failure) to xApp for a specific E2 Node following corresponding E2 interface transaction. Response contains Cause (i.e., Request contents not accepted by E2 Node, E2AP timeout, etc.).

|  |  |
| --- | --- |
| **Use Case**  **Stage** | **Evolution / Specification** |
| Goal | E2 Subscription API procedure from xApp initiation to E2 Node and response. |
| Actors and Roles | * xApp: Originator of E2 Subscription API request; * Near-RT RIC platform, with the following functionalities:   + Database;   + xApp Subscription Management;   + Conflict Mitigation;   + E2 Termination. * E2 Node: Handling RIC Subscription related procedure(s). |
| Assumptions | * Near-RT RIC platform maintains the records of the E2 subscriptions; * Near-RT RIC platform has access to sufficient information to both detect a potential conflict and take a decision on an optimal mitigation solution; * Near-RT RIC platform may generate guidance for internal processes and/or xApp as an optional addition response to Guidance Request. |
| Pre-conditions | * E2 Node has active E2 interface to Near-RT RIC; * Near-RT RIC has recovered complete list of active RAN Functions on E2 Node and informed initiating xApp; * xApp has been authorized to issue E2 Subscription API Requests; * xApp has been authorized to request guidance from Near-RT RIC platform; |

|  |  |
| --- | --- |
|  | * Near-RT RIC platform has been configured to permit E2 Subscription API requests from specific list of xApp; |
| Begins when | xApp determines need to propose E2 Subscription API request for a specific E2 Node ID and  defines message contents (target RAN Function ID, Event Trigger, Action List). |
| Step 1 (O) | xApp can request optional E2 related API: E2 Guidance from Near-RT RIC platform (see clause  9.3.3.1 for details). |
| Step 2 (M) | xApp sends **E2 related API: E2 Subscription** request with message contents (target RAN  Function ID, Event Trigger, Action List) for a specific E2 Node. |
| Step 3 (M) | Near-RT RIC platform, after optional conflict mitigation processing (see clause 9.3.3.5 for details),  accepts or rejects proposal from xApp for specific E2 Node. |
|  | [ALT] Step 4 is executed if Near-RT RIC platform rejects the proposal: |
| Step 4 (M) | Near-RT RIC platform sends **E2 related API: E2 Subscription** response to xApp (Reject with  Cause for specific E2 Node). |
|  | [ELSE] Steps 5-11 are executed if Near-RT RIC platform accepts the proposal: |
| Step 5 (M) | Near-RT RIC platform checks the proposal against its maintained subscription records for  duplicate, and determines the subsequent operation. |
|  | [ALT] Steps 6-9 are executed if the subscription proposed by xApp necessitates E2AP procedure(s) (i.e., no record found for E2 Node ID with Subscription matching the same  contents): |
| Step 6 (M) | Near-RT RIC platform selects appropriate E2 interface and initiates appropriate E2AP  procedure(s) to E2 Node. |
|  | [ALT] Steps 7-8 are executed if the E2AP procedure(s) in Step 6 succeeds: |
| Step 7 (M) | Near-RT RIC platform updates the subscription records. For each Action in subscription  corresponding to an Indication message, Near-RT RIC platform adds the xApp ID to the distribution list associated with the E2 Node ID, RIC Request ID and Action ID. |
| Step 8 (M) | Near-RT RIC platform sends **E2 related API: E2 Subscription** response to xApp  (Success, E2 Node ID, corresponding E2 Request ID(s) and E2 Action ID(s)). |
|  | [ELSE] Steps 9 is executed if the E2AP procedure(s) in Step 7 fails: |
| Step 9 (M) | Near-RT RIC platform sends **E2 related API: E2 Subscription** response to xApp  (Failure with Cause, E2 Node ID). |
|  | [ELSE] Steps 10-11 are executed if the proposal can be fulfilled with existing subscription(s)  (i.e., one or more records found for E2 Node ID with Subscription matching the same contents): |
| Step 10 (M) | Near-RT RIC platform updates the subscription records. For each Action in subscription  corresponding to an Indication message, Near-RT RIC platform adds xApp ID to distribution list associated with E2 Node ID, RIC Request ID and Action ID. |
| Step 11 (M) | xApp Subscription Management sends E2 Related API: E2 Subscription response to xApp  (Success, E2 Node ID, corresponding E2 Request ID(s) and E2 Action ID(s)). |



##### Figure 9.3.2.1-1: E2 Subscription procedure

#### E2 Subscription Delete procedure

The purpose of the E2 Subscription Delete procedure in the Near-RT RIC is to ensure that a) only validated RIC Subscription Delete Request messages are issued by the Near-RT RIC over the E2 interface to the E2 Node; b) deletion requests for duplicated Subscription Requests are handled gracefully.

This procedure is based on the following assumptions:

* + - * + xApp has been configured with a trusted xApp ID;
        + Near-RT RIC platform maintains all active subscriptions towards the E2 Node listed in xApp’s E2 Subscription Delete request and is able to detect when and if the delete request shall result in a corresponding E2 message;
        + Near-RT RIC platform routes the subscription delete requests to an appropriate E2interface;
        + Near-RT RIC platform maintains a mapping of active subscriptions (identified on E2 interface by E2 Node ID and RIC Request ID) of the validated xApps. Near-RT RIC platform uses the mapping when sending RIC INDICATION messages to the correct xApp;

The procedure initiated by an xApp uses **E2 Related API: E2 Subscription Delete** to request subscription deletion for a specific E2 Node. The following outcomes are considered:

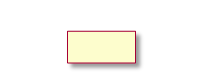
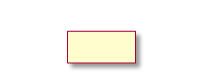
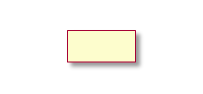
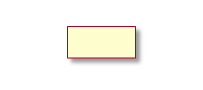
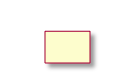
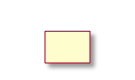
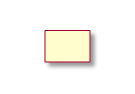
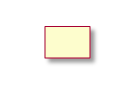
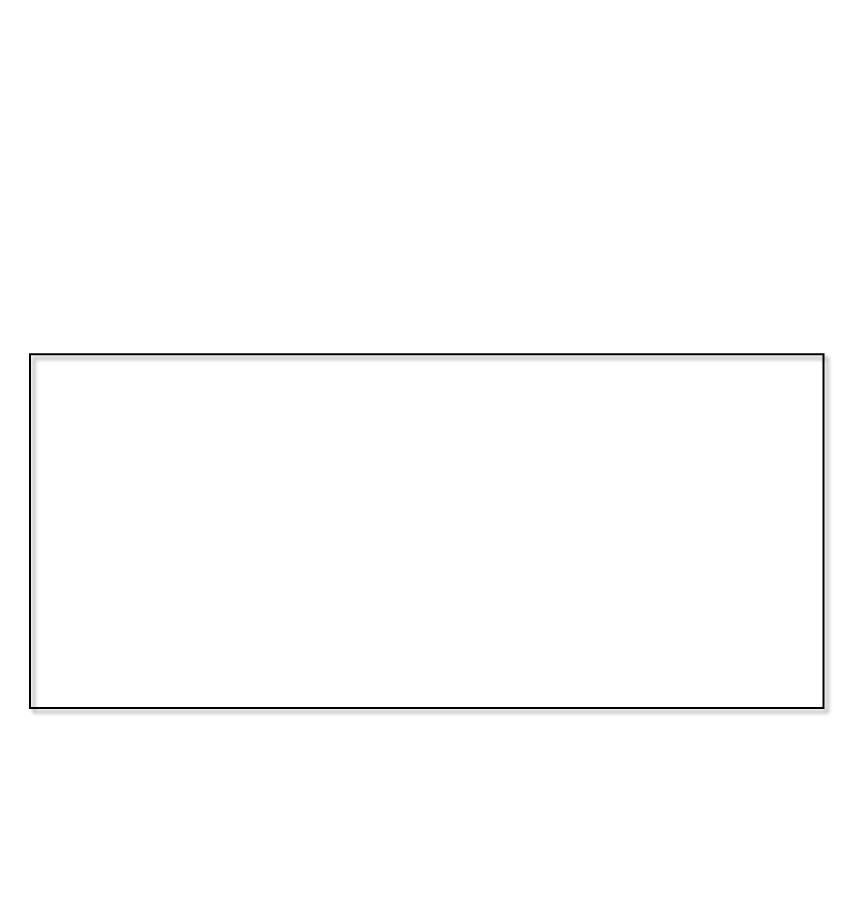
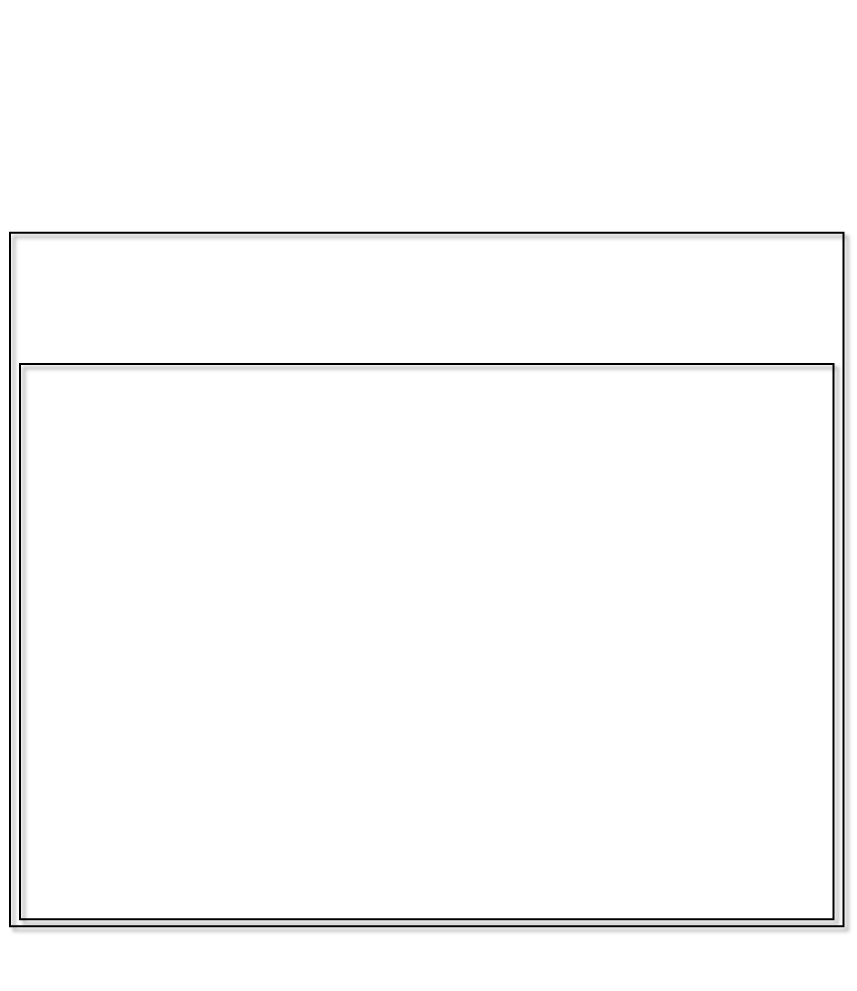
* + - * + Request fails following rejection by Near-RT RIC platform and an **E2 Related API: E2 Subscription Delete** response (Reject) is sent to xApp without E2 interface transaction (i.e., xApp not previously associated with validated subscription);
        + Request succeeds following acceptance by Near-RT RIC platform but other subscriber(s) detected for the subscription, and an **E2 Related API: E2 Subscription Delete** response (Success) is sent to xApp without corresponding E2 transaction;
        + Request succeeds following acceptance by Near-RT RIC platform and accepted by E2 Node, and an **E2 Related API: E2 Subscription Delete** response (Success) is sent to xApp following corresponding E2 transaction;
        + Request fails following acceptance by Near-RT RIC platform but is either rejected by E2 Node or corresponding E2AP timer expires prior to a response from E2 Node, and an **E2 Related API: E2 Subscription Delete** response (Failure) is sent to xApp following corresponding E2 interface transaction (i.e., RIC Request ID is not known by E2 Node, time out, etc.);

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | E2 Subscription Delete procedure from xApp initiation to E2 Node and response. |
| Actors and Roles | * xApp: Originator of E2 Subscription Delete request; * Near-RT RIC platform, with the following functionalities:   + Database;   + xApp Subscription Management;   + E2 Termination. * E2 Node: Handling RIC Subscription Delete procedure. |
| Assumptions | * Near-RT RIC platform maintains the E2 subscription records; |
| Pre conditions | * E2 Node has active E2 interface to Near-RT RIC; * Near-RT RIC has recovered complete list of active RAN Functions on E2 Node and informed initiating xApp; * Near-RT RIC platform has been configured to permit RIC Subscription Delete requests   from xApp. |
| Begins when | xApp determines need to request E2 Subscription Delete for a specific E2 Node ID and E2  Request ID. |
| Step 1 (M) | xApp sends **E2 related API: E2 Subscription Delete** request for a specific E2 Node and E2  request ID. |
| Step 2 (M) | Near-RT RIC platform confirms that xApp proposing E2 Subscription Delete has an active  subscription. |
| Step 3 (M) | Near-RT RIC platform accepts or rejects proposal from xApp. |
|  | [ALT] Step 4 is executed if xApp Subscription Management rejects the proposal: |
| Step 4 (M) | Near-RT RIC platform sends **E2 related API: E2 Subscription Delete** response to xApp  (Reject with Cause for specific E2 Node and E2 Request ID). |
|  | [ELSE] Steps 5-13 are executed if xApp Subscription Management accepts the proposal. |
| Step 5 (M) | Near-RT RIC platform checks the subscription records to determine whether or not xApp  proposing Subscription Delete is the only associated xApp. |
|  | [ALT] Steps 6-11 are executed if the xApp proposing Subscription Delete is the only xApp  associated with the subscription (i.e., a given RIC Request ID): |
| Step 6 (M) | Near-RT RIC platform selects appropriate E2 interface and sends **RIC Subscription**  **Delete Request** (RIC Request ID) to E2 Node. |
|  | [ALT] Steps 7-9 are executed if E2 Node accepts Subscription Delete: |
| Step 7 (M) | E2 Node responds with **RIC Subscription Delete Response**. |
| Step 8 (M) | Near-RT RIC platform updates the subscription records, and deletes the distribution  list(s) associated with the E2 Node ID and the RIC Request ID. |
| Step 9 (M) | Near-RT RIC platform sends **E2 related API: E2 Subscription Delete** response to  xApp (Success). |
|  | [ELSE] Steps 10-11 are executed if E2 Node rejects Subscription Delete: |
| Step 10 (M) | E2 Node responds with **RIC Subscription Delete Failure** (with Cause). |
| Step 11 (M) | Near-RT RIC platform sends **E2 related API: E2 Subscription Delete** response to  xApp (Failure with Cause). |
|  | [ELSE] Step 12 if E2 Node fails to respond prior to E2AP timer expires |

|  |  |
| --- | --- |
| Step 12 (M) | Near-RT RIC platform E2AP timer expires and sends **E2 related API: E2**  **Subscription Delete** response (Failure with Cause) to xApp. |
|  | [ELSE] Steps 13-14 are executed if the xApp proposing Subscription Delete is one of a  number of xApp associated with the subscription (i.e., a given RIC Request ID): |
| Step 13 (M) | Near-RT RIC platform updates the subscription records, and removes the xApp from  distribution list(s) associated with the E2 Node ID and the RIC Request ID. |
| Step 14 (M) | Near-RT RIC platform sends **E2 related API: E2 Subscription Delete** response to xApp  (Success). |

xApp

Near RT RIC platform



E Node

1 (E related A I) E ubscription elete (Request E Node I E Request I )

Confirm that xApp had pre iously subscribed (i.e. record found for xApp I )

3 Accept Reject decision

4 (E related A I) E ubscription elete (Reject E Node I E Request I Cause)

5 Count alid records

E RIC CRI TION E ETE RE E T (RIC Request I )

7 E RIC CRI TION E ETE RE ON E

pdate the subscription records and delete the distribution list

(E related A I) E ubscription elete ( uccess E Node I E Request I )

10 E RIC CRI TION E ETE AI RE

11 (E related A I) E ubscription elete ( ailure E Node I E Request I Cause)

1 a. E A time out

1 b. (E related A I) E ubscription elete ( ailure E Node I E Request I Cause)

13 pdate the subscription records

and delete the xApp from the distribution list

14 (E related A I) E ubscription elete ( uccess E Node I E Request I )

xApp

Near RT RIC platform

E Node

##### Figure 9.3.2.2-1: E2 Subscription Delete procedure

#### A E2 Subscription Delete Query procedure and E2 Subscription Delete Notification procedure

The purpose of the E2 Subscription Delete Query procedure and E2 Subscription Delete Notification Procedure in the Near-RT RIC is to ensure that deletion request of the existing subscriptions from an E2 Node is handled properly.

The two procedures are based on the following assumptions:

* + - * + The Near-RT RIC platform maintains the list of xApps associated with the subscription removal request from an E2 Node and, if decides to accept, uses the E2 related API: E2 Subscription Delete Notification (Deleted) to delete subscriptions toward xApps. Prior to determining whether or not to accept the subscription delete required request, the Near-RT RIC platform may send E2 related API: E2 Subscription Delete Query (required) to each xApp with an active subscription, each xApp may respond with E2 related API: E2 Subscription Delete (accept) or E2 related API: E2 Subscription Delete (decline).The procedures may be initiated by theNear-RT RIC platform, when RIC Subscription Delete Required is received from an E2 Node. The Near-RT RIC platform may use the **E2 Related API: E2 Subscription Delete Query (required)** to consult xApps. The following outcomes are considered:
        + xApp accepts request to delete subscription and sends E2 Related API: E2 Subscription Delete Query (accept) to accept subscription deletion.
        + xApp declines request to delete subscription and sends E2 Related API: E2 Subscription Delete Query (decline) to inform Near-RT RIC platform.

In both cases, the Near-RT RIC platform may decide whether to delete the subscription.

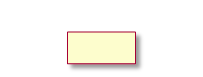
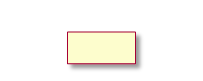
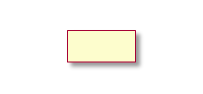
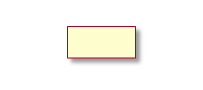
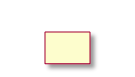
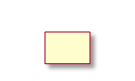
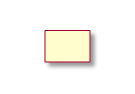
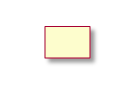
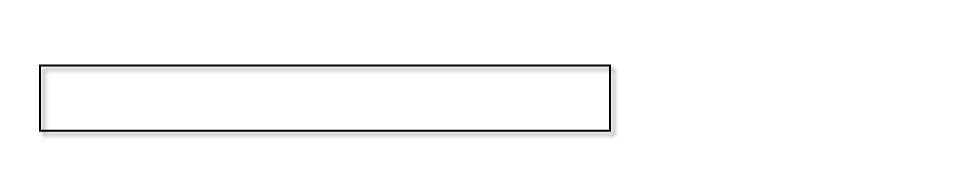
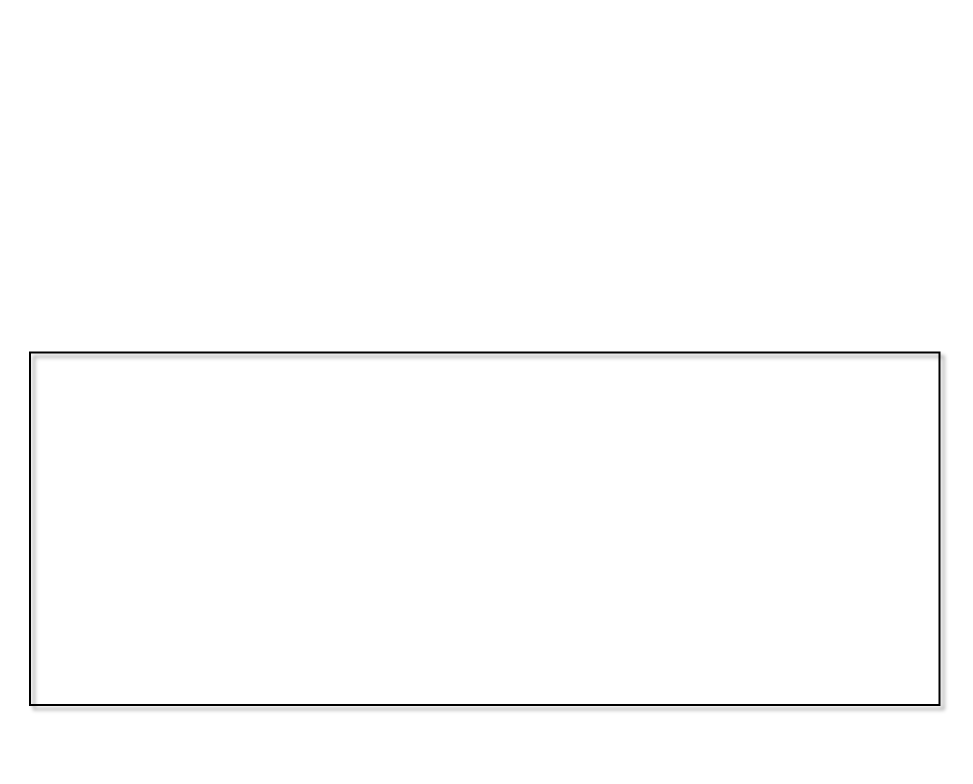
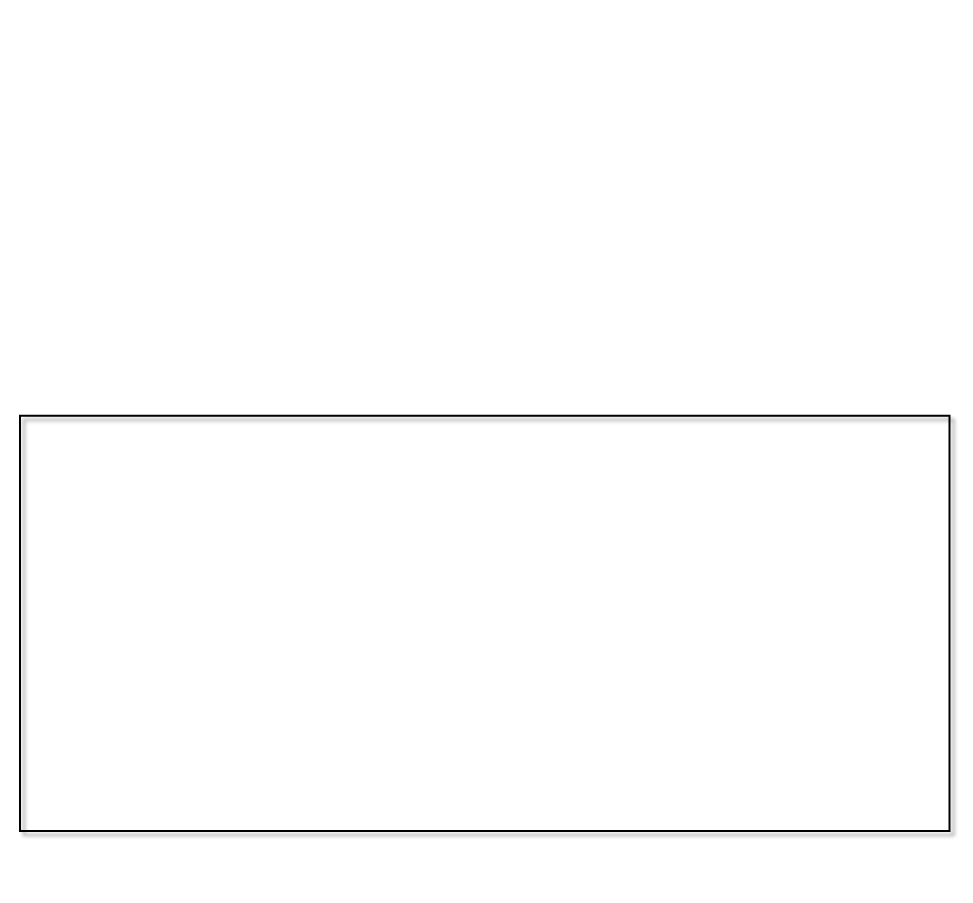
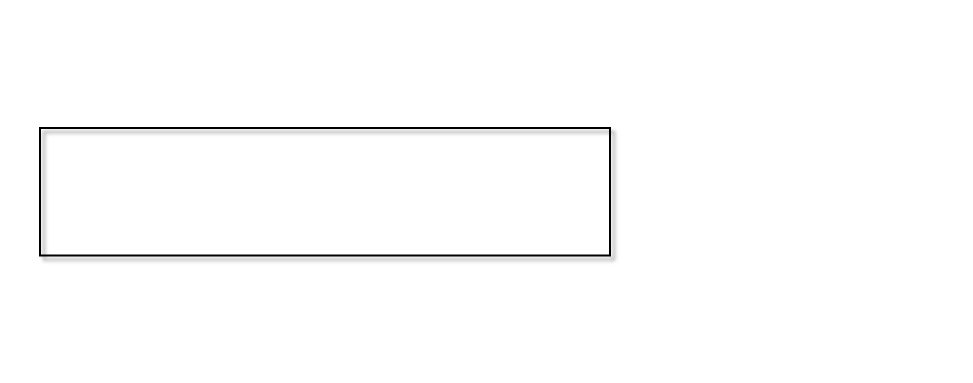
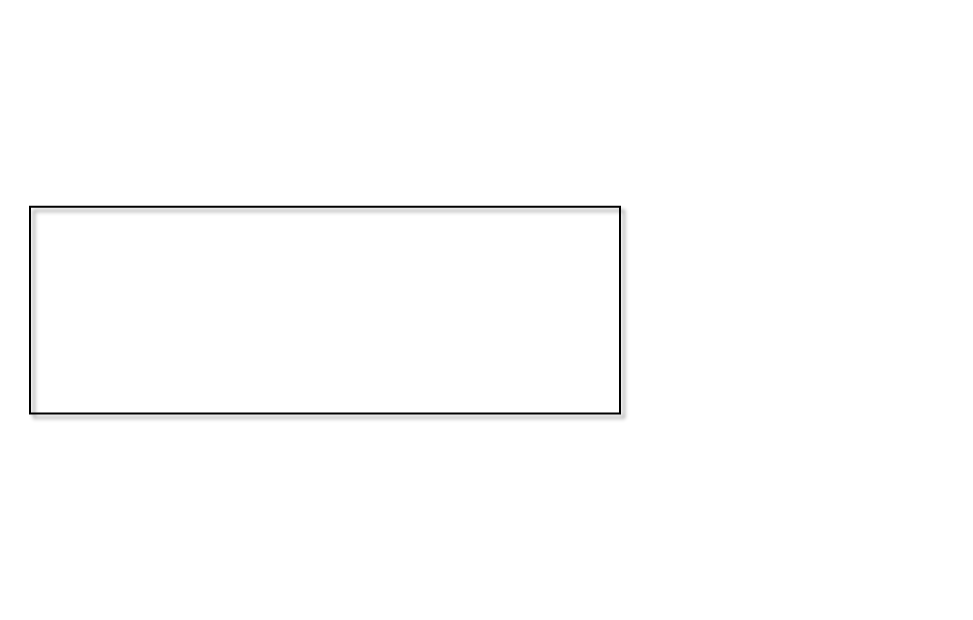
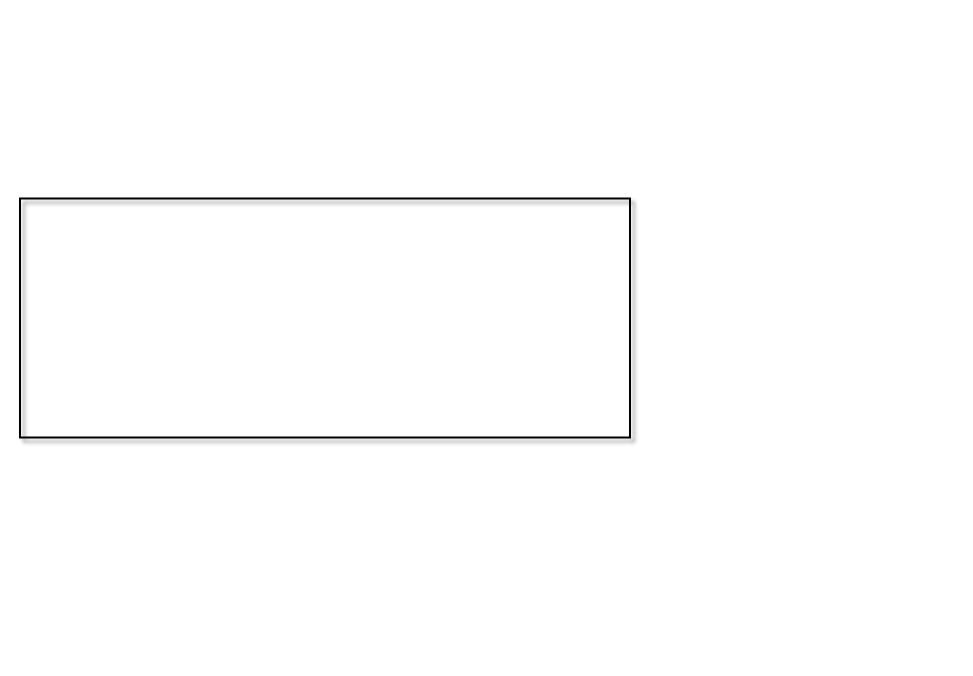
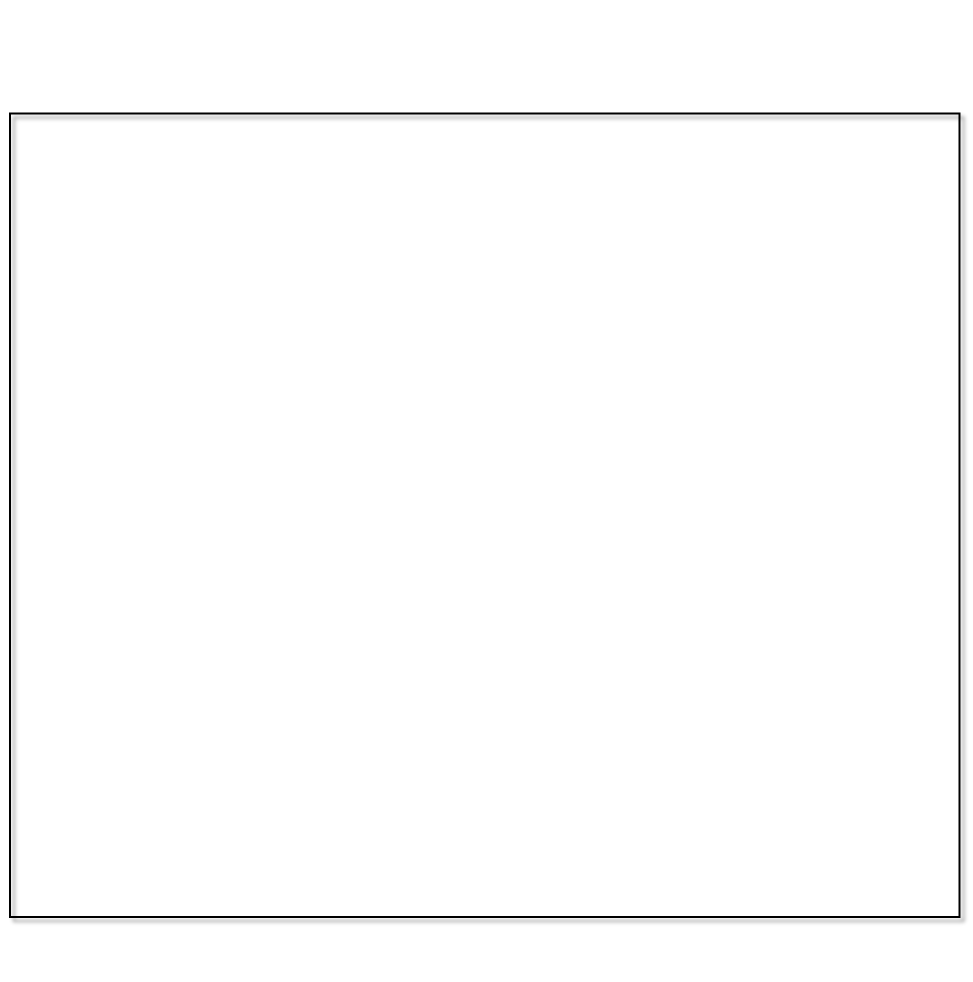
The Near-RT RIC platform may also decide whether to delete the subscription without consulting xApps.

If the Near-RT RIC platform decides to delete the subscription, it uses **E2 Related API: E2 Subscription Delete Notification**

(Deleted) to inform each xApp for the eventual outcome.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | RIC subscription removal initiated from an E2 Node. |
| Actors and Roles | * xApp: Recipient of E2 Subscription Delete query and notification; * Near-RT RIC platform, with the following functionalities:   + Database;   + xApp Subscription Management;   + E2 Termination. * E2 Node: Originator of RIC Subscription Delete Required. |
| Assumptions | * Near-RT RIC platform maintains the records of E2 subscriptions; |
| Pre conditions | * E2 Node has active E2 interface to Near-RT RIC; * Near-RT RIC has recovered complete list of active RAN Functions on E2 Node and informed xApps; * Near-RT RIC platform has been configured to accept incoming RIC subscription removal request from an E2 Node; * Near-RT RIC platform has been configured to permit RIC Subscription removal request from an E2 Node. |
| Begins when | E2 Node determines a need to request removal of some RIC subscriptions that has been  subscribed toward itself from Near-RT RIC. |
| Step 1 (M) | E2 Node sends **RIC Subscription Delete Required** (a list of RIC Request IDs and RAN  Function IDs with cause). |
|  | [LOOP] Steps 2-12 are looped for each Subscription requested to be deleted: |
| Step 2 (M) | Near-RT RIC platform checks the subscription records to find the list of xApps associated  with RIC subscription requested to be deleted from the E2 Node. |
|  | [OPT] Steps 3-5 may be executed. For each xApp with RIC Subscription requested to be  deleted: |
| Step 3 (O) | Near-RT RIC platform sends **E2 related API: E2 Subscription Delete Query** Required to xApp (with E2 Node ID, E2 Request ID and Cause) to request xApp advice on required  subscription deletion. |
| Step 4 (O) | If xApp accepts, xApp responds with **E2 related API: E2 Subscription Delete Query**  Accept to Near-RT RIC platform (with E2 Node ID, E2 Request ID). |
| Step 5 (O) | If xApp rejects, xApp responds with **E2 related API: E2 Subscription Delete Query**  Decline to Near-RT RIC platform (with E2 Node ID, E2 Request ID). |
| Step 6 (M) | Near-RT RIC platform takes decision whether to accept/reject RIC subscription delete. |

|  |  |
| --- | --- |
|  | [OPT] Steps 7-12 are executed if Near-RT RIC platform accepts: |
| Step 7 (M) | Near-RT RIC platform selects appropriate E2 interface and sends **RIC Subscription**  **Delete Request** (RIC Request ID) to E2 Node. |
|  | [ALT] Steps 8-10 are executed if E2 Node accepts Subscription Delete: |
| Step 8 (M) | E2 Node responds with **RIC Subscription Delete Response.** |
| Step 9 (M) | Near-RT RIC platform updates the subscription records, and deletes the distribution  list associated with the E2 Node ID and the RIC Request ID. |
| Step 10 (M) | For each concerned xApp, Near-RT RIC platform sends **E2 related API: E2 Subscription Delete Notification** (with E2 Node ID, E2 Request ID, Cause) to inform  xApp that subscription has been deleted. |
|  | [ELSE] Steps 11-12 are executed if E2 Node rejects Subscription Delete: |
| Step 11 (M) | E2 Node responds with **RIC Subscription Delete Failure** (with Cause). |
| Step 12 (M) | Near-RT RIC platform decides subsequent processing (e.g., retry RIC Subscription  Delete). The failure outcome is not needed to inform the xApp. |
|  | [ELSE] Step 13 is executed if E2 Node fails to respond prior to E2AP timer expires |
| Step 13 (M) | Near-RT RIC platform E2AP timer expires and decides subsequent processing (e.g.,  retry RIC Subscription Delete). The failure outcome is not needed to inform the xApp. |



xApp

Near RT RIC platform

E Node

1. E RIC CRI TION E ETE RE IRE

. Check subscription records for concerned xApps (RIC Request I E Node I )

. Accept Reject decision

7. E RIC CRI TION E ETE RE E T

. E RIC CRI TION E ETE RE ON E

. pdate subscriptio

1 . ecide subseque (e.g. retry RIC subsc

13. E A time out. (e.g. retry RIC subsc

xApp

Near RT RIC platform

E Node

ecide subsequent processing ription delete)

nt processing ription delete)

CRI TION E ETE AI RE

11. E RIC

10. (E related A I) E ubscription elete Notification ( eleted E Node I E Request I Cause)

n records and delete the distribution list

5. (E related A I) E ubscription elete uery ( ecline E Node I E Request I )

4. (E related A I) E ubscription elete uery (Accept E Node I E Request I )

3. (E related A I) E ubscription elete uery (Required E Node I E Request I Cause)

##### Figure 9.3.2.2A-1: E2 Subscription Delete Query and E2 Subscription Delete Notification procedure

#### E2 Indication procedure

The purpose of the E2 Indication procedure in the Near-RT RIC is to ensure delivery of RIC INDICATION messages to one or more validated xApps.

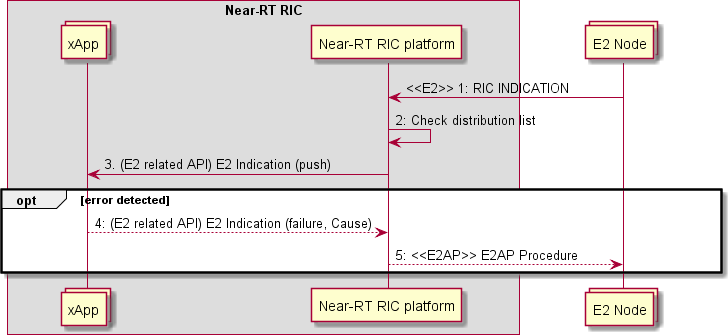
This procedure is based on the following assumptions:

* + - * + Near-RT RIC platform maintains a list for each E2 Node of validated xApps and the corresponding assigned E2 Request ID and E2 Action ID(s) associated with an E2 node ID, RIC REQUEST ID and RIC Action ID(s);
        + E2 related API ensures delivery of E2 Indication messages from Near-RT RIC platform to all validated xApp;
        + Near-RT RIC platform may support distribution of messages to multiple destinations.

The following outcomes are considered:

* + - * + Message is received, using E2 related API: E2 Indication (push), and successfully processed by xApp;
        + Message is received, using E2 related API: E2 Indication (push), but an error is detected by the xApp and so declined using an E2 related API: E2 Indication (failure) from xApp for a specific E2 Node, E2 Request ID, E2 Action ID and E2 Indication SN. Response contains Cause (i.e., Request contents not accepted by xApp).

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | E2 Indication procedure from E2 Node initiation to xApp reception. |
| Actors and Roles | * xApp: Originator of RIC subscription request; * Near-RT RIC platform, with the following functionalities:   + Database;   + xApp Subscription Management;   + E2 Termination. * E2 Node: Originator of RIC Indication procedure. |
| Assumptions | * For RIC Indication procedures, Near-RT RIC platform uses E2 related API to directly send received messages to one or more xApp. |
| Pre conditions | * E2 Node has active E2 interface to Near-RT RIC; * Near-RT RIC has recovered complete list of active RAN Functions on E2 Node and informed initiating xApp; * Near-RT RIC platform has distribution list associating Indication messages with xApps. |
| Begins when | E2 Node creates RIC Indication Message (RIC Request ID, RAN Function ID, RIC Action ID,  optional RIC Indication SN, Indication Type, Indication Header, Indication Message, optional Call Process ID). |
| Step 1 (M) | E2 Node sends **RIC Indication** message to Near-RT RIC platform. |
| Step 2 (M) | Near-RT RIC platform checks distribution list for E2 Node ID, RIC REQUEST ID, RAN  Function ID, RIC Action ID. |
| Step 3 (M) | Near-RT RIC platform sends **E2 related API: E2 Indication** (push, E2 Node ID, E2 Request  ID, E2 Action ID, E2 Indication SN, Message) to xApp(s). |
| Step 4 (O) | If an error is detected in the received message, the xApp may send **E2 related API: E2**  **Indication** (failure, E2 Node ID, E2 Request ID, E2 Action ID, E2 Indication SN, Cause). |
| Step 5 (O) | The Near-RT RIC platform may send an appropriate E2AP procedure message |



##### Figure 9.3.2.3-1: E2 Indication procedure

#### E2 Control procedure

The purpose of the E2 Control procedure in the Near-RT RIC is to ensure that only authorized xApp may initiate RIC Control Request messages issued by the Near-RT RIC over the E2 interface to the E2 Node.

This procedure is based on the following assumptions:

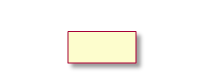
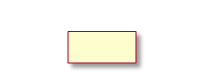
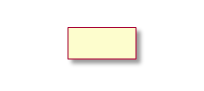
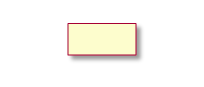
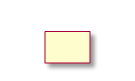
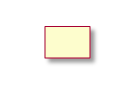
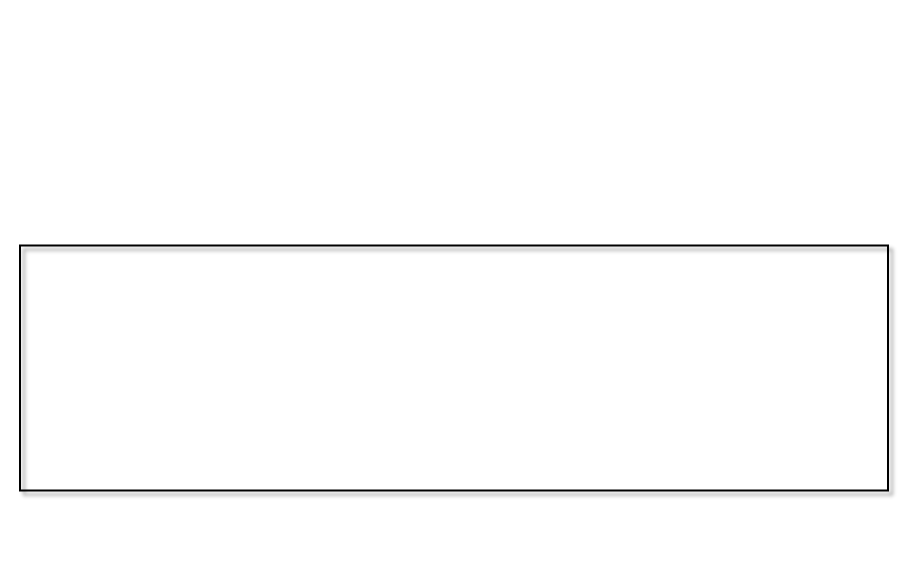
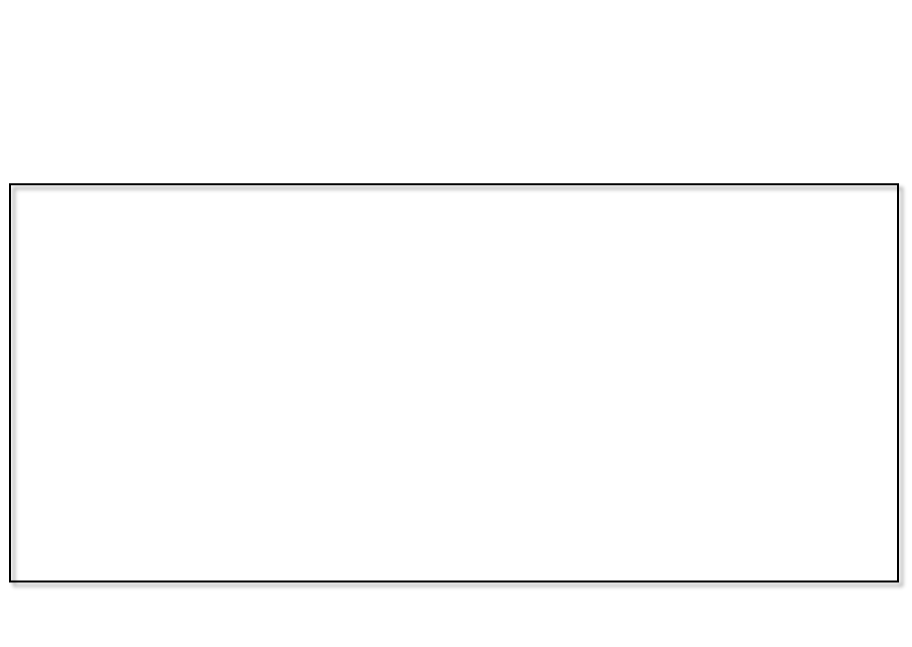
* + - * + xApp may obtain guidance from Near-RT RIC platform to resolve potential conflicts prior to sending a E2 Control Request (see clause 9.3.3.1);
        + xApp has been configured with a trusted xApp ID;
        + Near-RT RIC platform may be configured to route E2 Control Request messages for a E2 node from xApp either to trigger conflict mitigation for acceptance or directly towards an appropriate E2interface;
        + Near-RT RIC platform ensures that only authorized xApp may send E2 Control Request messages to appropriate E2 interface;
        + Near-RT RIC platform ensures that it systematically forwards any received RIC Control Response or RIC Control Failure to appropriate xApp.

The procedure is initiated by an xApp using E2 Related API: E2 Control request to send a request of a RIC Control for an E2 Node. The following outcomes are considered:

* + - * + Request successful following acceptance by Near-RT RIC platform and accepted by E2 Node and so an **E2 Related API: E2 Control** response (Success with outcome) is sent to xApp following corresponding E2 transaction;
        + Request fails following acceptance by Near-RT RIC platform but is either rejected by E2 Node or corresponding E2AP timer expires prior to a response from E2 Node and so an **E2 Related API: E2 Control** response (Failure with cause) is sent to xApp following corresponding E2 interface transaction (i.e., Request contents is not compliant, E2AP timeout, etc.);
        + Request rejected by Near-RT Platform and so an **E2 Related API: E2 Control** response (Reject) is sent to xApp (i.e., rejected due to conflict mitigation).

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | E2 Control procedure from xApp initiation to E2 Node and response. |
| Actors and Roles | * xApp: Originator of E2 Control request; * Near-RT RIC platform, with the following functionalities:   + E2 Termination. * E2 Node: RIC Control procedure. |
| Assumptions | * For E2 Control procedures, Near-RT RIC platform may forward the xApp's control request directly to the E2 interface; * Near-RT RIC platform has access to sufficient information to both detect a potential conflict and take a decision on an optimal mitigation solution. |
| Pre-conditions | * E2 Node has active E2 interface to Near-RT RIC; * Near-RT RIC has recovered complete list of active RAN Functions on E2 Node and informed initiating xApp; * xApp has been authorized to send RIC Control requests for a specific scope (E2 Node list, RAN Function, etc.); * xApp has been authorized to request guidance from Near-RT RIC platform; * Near-RT RIC platform has been configured to accept incoming E2 Control requests from authorized xApp and, either trigger conflict mitigation or forward directly to E2interface. |
| Begins when | xApp determines need to propose RIC Control procedure for a E2 Node and defines message  contents (RAN Function ID, Call process ID, Control Header, Control message). |
| Step 1 (O) | xApp may request **E2 related API: E2 Guidance** from Near-RT RIC platform (see clause  9.3.3.1 for details) |
| Step 2 (M) | xApp sends **E2 related API: E2 Control** request with message contents (RAN Function ID, Call  process ID, Control Header, Control message) for an E2 Node. |
| Step 3 (M) | Near-RT RIC platform processes request. |

|  |  |
| --- | --- |
|  | [ALT] Step 4 is executed if Near-RT RIC platform rejects proposed E2 Control message. |
| Step 4 (M) | **E2 related API: E2 Control** response (Reject with Cause) message is sent to xApp. |
|  | [ELSE] Step 5-10 are executed if Near-RT RIC platform accepts proposed E2 Control message: |
| Step 5 (M) | Near-RT RIC platform selects appropriate E2 interface and assigns an appropriate RIC  Request ID. |
| Step 6 (M) | Near-RT RIC platform sends **RIC Control Request** (RIC Request ID, contents) to E2 Node. |
|  | [ALT] Steps 7-8 are executed if E2 Node accepts RIC Control Request: |
| Step 7 (M) | E2 Node responds with **RIC Control Acknowledge.** |
| Step 8 (M) | Near-RT RIC platform sends **E2 Related API: E2 Control** response (Success with  Outcome) to xApp. |
|  | [ELSE] Steps 9-10 are executed if E2 Node rejects RIC Control Request: |
| Step 9 (M) | E2 Node responds with **RIC Control Failure** (with Cause). |
| Step 10 (M) | Near-RT RIC platform sends **E2 Related API: E2 Control** response (Failure with Cause)  to xApp. |
|  | [ELSE] Step 11 is executed if E2 Node fails to respond prior to E2AP timer expires |
| Step 11 (M) | Near-RT RIC platform E2AP timer expires and sends **E2 Related API: E2 Control**  response (Failure with Cause) to xApp. |



xApp

Near RT RIC platform

E Node

1 (E related A I) uidance (Request Response)

. (E related A I) E Control (Request)

3. Near RT RIC platform processes request

5 elect appropriate E interface and assigns RIC Request I

E RIC CONTRO RE E T

7 E RIC CONTRO AC NO E E

. (E related A I) E Control ( uccess)

E RIC CONTRO AI RE

11a. E A time out

xApp

Near RT RIC platform

E Node

11b. (E related A I) E Control ( ailure)

10. (E related A I) E Control ( ailure)

4. (E related A I) E Control (Reject)

##### Figure 9.3.4-1 E2 Control Procedure

#### E2 Subscription Audit procedure

The purpose of the E2 Subscription Audit procedure in the Near-RT RIC is to enable an xApp to initiate an audit of one or more existing E2 subscriptions.

This procedure is based on the following assumptions:

* + - * + xApp has been configured with a trusted xApp ID;
        + Near-RT RIC platform maintains all active subscriptions towards the E2 Node listed in xApp’s E2 Subscription Audit request and is able to detect when and if the audit request shall result in a corresponding E2 message;
        + Near-RT RIC platform routes the subscription audit requests to an appropriate E2 interface;
        + Near-RT RIC platform maintains a mapping of active subscriptions (identified on E2 interface by E2 Node ID and RIC Request ID) of the validated xApps. Near-RT RIC platform uses the mapping when sending RIC INDICATION messages to the correct xApp;

The procedure initiated by an xApp uses **E2 Related API: E2 Subscription Audit** to request subscription audit for a specific E2 Node. The following outcomes are considered:

* + - * + Request fails following rejection by Near-RT RIC platform and an **E2 Related API: E2 Subscription Audit** response (Reject) is sent to xApp without E2 interface transaction (i.e., xApp not previously associated with one or more validated subscription);
        + Request succeeds following acceptance by Near-RT RIC platform and accepted by E2 Node, and an **E2 Related API: E2 Subscription Audit** response (Success) is sent to xApp following corresponding E2 interface transaction;
        + Request succeeds following acceptance by Near-RT RIC platform and an **E2 Related API: E2 Subscription Audit**

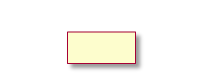
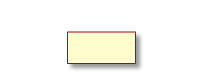
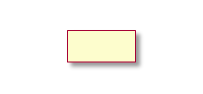
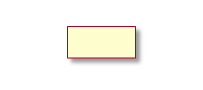
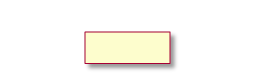
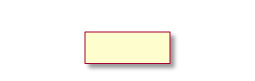
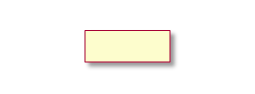
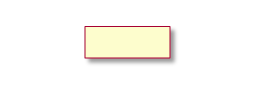
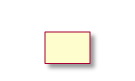
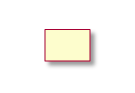
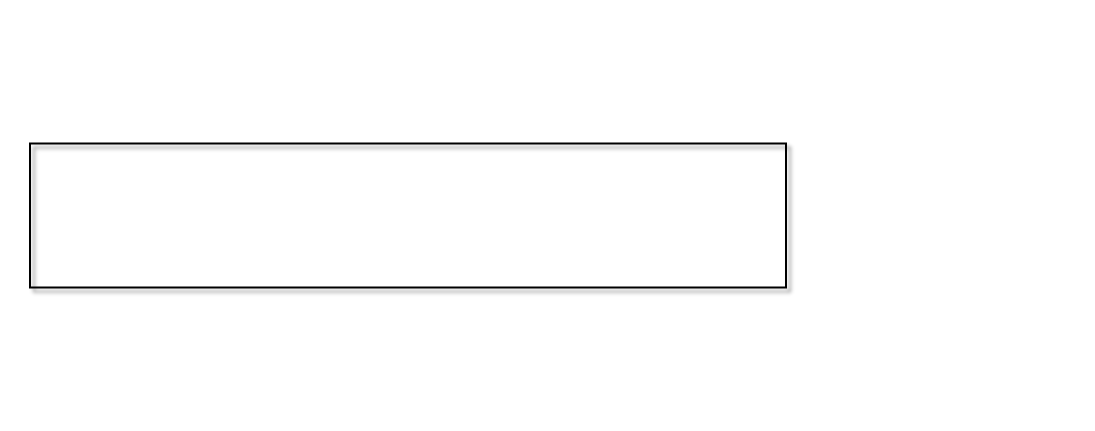
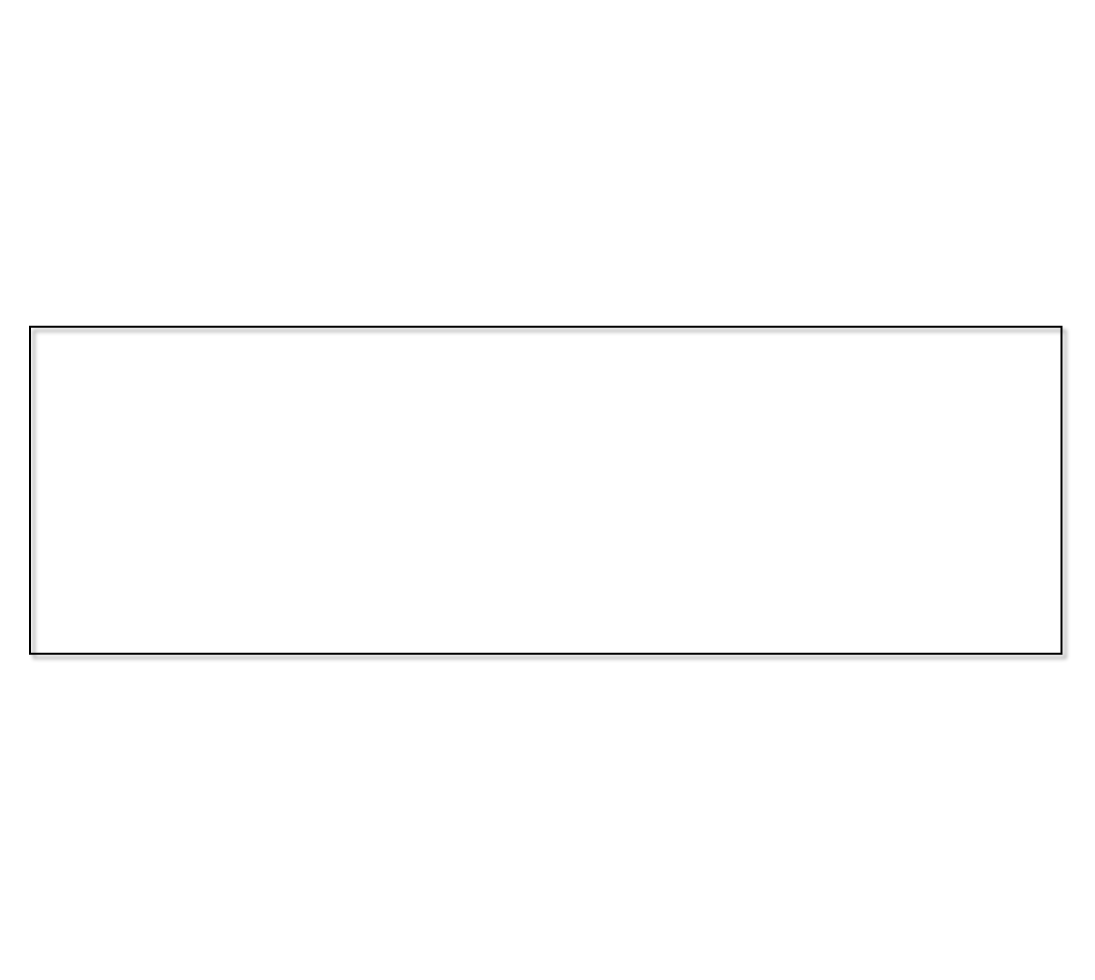
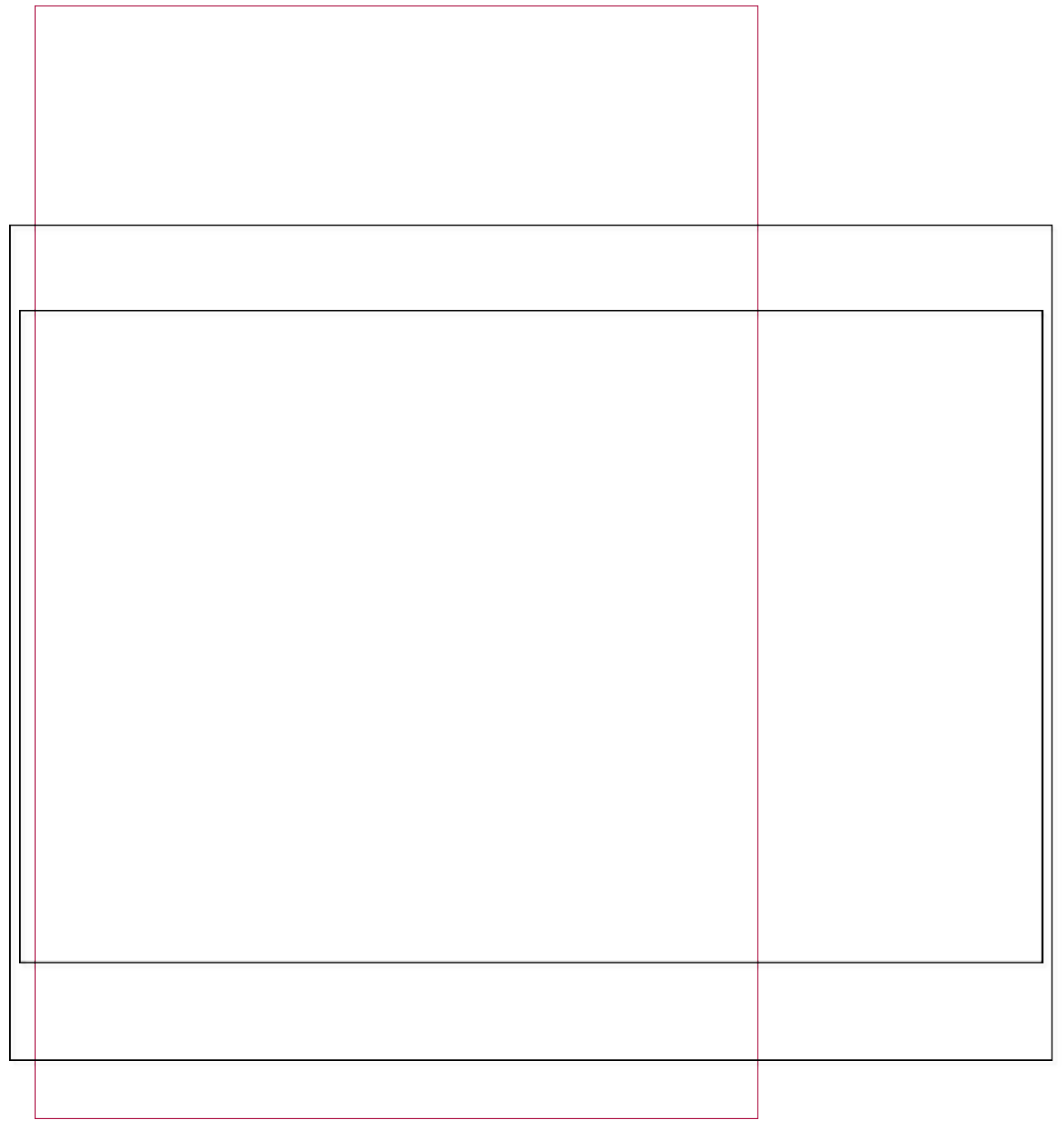
response (Success) is sent to xApp without E2 interface transaction;

* + - * + Request fails following acceptance by Near-RT RIC platform but is either rejected by E2 Node or corresponding E2AP timer expires prior to a response from E2 Node, and an **E2 Related API: E2 Subscription Audit** response (Failure) is sent to xApp following corresponding E2 interface transaction (i.e., RIC Request ID is not known by E2 Node, E2AP timeout, etc.);
        + Request fails following acceptance by Near-RT RIC platform and an **E2 Related API: E2 Subscription Audit**

response (Failure) is sent to xApp without E2 interface transaction.

This procedure may result in the Near-RT RIC platform issuing **E2 related API: E2 Subscription Delete Notification** (with E2 Node ID, E2 Request ID, Cause) message(s) to inform other xApp that a subscription has been deleted. This may also occur if the Near-RT RIC platform independently initiates a RIC Subscription Audit procedure.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | E2 Subscription Audit procedure from xApp initiation to E2 Node and response. |
| Actors and Roles | * xApp: Originator of E2 Subscription Audit request; * Near-RT RIC platform, with the following functionalities:   + Database;   + xApp Subscription Management;   + E2 Termination. * E2 Node: Handling RIC Subscription Audit procedure. |
| Assumptions | * Near-RT RIC platform maintains the E2 subscription records; |
| Pre conditions | * E2 Node has active E2 interface to Near-RT RIC; * Near-RT RIC has recovered complete list of active RAN Functions on E2 Node and informed initiating xApp; * Near-RT RIC platform has been configured to permit RIC Subscription Audit requests from xApp. |
| Begins when | xApp determines need to request E2 Subscription Audit for a specific E2 Node ID and one or  more E2 Request ID. |
| Step 1 (M) | xApp sends **E2 related API: E2 Subscription Audit** request for a specific E2 Node and one or  more E2 request ID. |
| Step 2 (M) | Near-RT RIC platform confirms that xApp proposing E2 Subscription Audit has an active  subscription associated with each E2 Request ID in the list provided by the xApp. |
| Step 3 (M) | Near-RT RIC platform accepts or rejects proposal from xApp. |
|  | [ALT] Step 4 is executed if Near-RT RIC platform rejects the proposal: |
| Step 4 (M) | Near-RT RIC platform sends **E2 related API: E2 Subscription Audit response** to xApp  (Reject with Cause for specific E2 Node and one or more E2 Request ID). |
|  | [ELSE] Steps 5-15 are executed if Near-RT RIC platform accepts the proposal. |
|  | [ALT] Steps 5-12 are executed if the Near-RT RIC executes a corresponding E2 interface  transaction. |
| Step 5 (M) | Near-RT RIC platform selects appropriate E2 interface and sends **RIC Subscription**  **Audit Request** (one or more RIC Request ID including subscriptions from xApp) to E2 Node. |
|  | [ALT] Steps 6-7 are executed if E2 Node accepts Subscription Audit: |
| Step 6 (M) | E2 Node responds with **RIC Subscription Audit Response**, with lists of confirmed,  unknown and/or missing subscriptions. |
| Step 7 (M) | Near-RT RIC platform sends **E2 related API: E2 Subscription Audit** response to  xApp (Success). |
|  | [ELSE] Steps 8-9 are executed if E2 Node rejects Subscription Audit: |
| Step 8 (M) | E2 Node responds with **RIC Subscription Audit Failure** (with Cause). |
| Step 9 (M) | Near-RT RIC platform sends **E2 related API: E2 Subscription Audit** response to  xApp (Failure with Cause). |
|  | [ELSE] Step 10 is executed if E2 Node fails to respond prior to E2AP timer expires |
| Step 10 (M) | Near-RT RIC platform E2AP timer expires and sends **E2 Related API: E2**  **Subscription Audit** response (Failure with Cause) to xApp. |
| Step 11 (M) | Near-RT RIC platform updates subscription records. |
| Step 12 (O) | If one or more subscriptions are listed as missing, then the Near-RT RIC platform may  initiate **RIC Subscription Delete** procedure to delete subscription on E2 Node. |
|  | [ELSE] Steps 13-14 are executed if the Near-RT RIC determines that xApp response may be  generated without a corresponding E2 interface transaction. |
|  | [ALT] Step 12 is executed if the result is a success: |
| Step 13 (M) | Near-RT RIC platform sends **E2 related API: E2 Subscription Audit** response to  xApp (Success). |
|  | [ELSE] Step 14 is executed if the result is a failure: |
| Step 14 (M) | Near-RT RIC platform sends **E2 related API: E2 Subscription Audit** response to  xApp (Failure with Cause). |
| Step 15 (O) | If one or more subscriptions are listed as unknown, then Near-RT RIC platform may send **E2**  **related API: E2 Subscription Delete Notification** (with E2 Node ID, E2 Request ID, Cause) to inform concerned xApps that subscription has been deleted. |



xApp

Other xApp

Near RT RIC platform

E Node

1 (E related A I) E ubscription Audit (Request E Node I ist of E Request I )

Confirm that xApp had pre iously subscribed (i.e. record found for xApp I )

3 Accept Reject decision

4 (E related A I) E ubscription Audit

(Reject E Node I ist of E Request I Cause)

7 (E related A I) E ubscription Audit

( uccess E Node I Confirmed nknown and or

issing list of E Request I )

(E related A I) E ubscription Audit ( ailure E Node I Cause)

10b. (E related A I) E ubscription Audit ( ailure)

13 (E related A I) E ubscription Audit

( uccess E Node I Confirmed nknown and or

issing list of E Request I )

14 (E related A I) E ubscription Audit ( ailure E Node I Cause)

15a (E related A I) E ubscription elete Notification ( eleted E Node I E Request I Cause)

xApp

Other xApp

Near RT RIC platform

E Node

15b (E related A I) E ubscription elete Notification ( eleted E Node I E Request I Cause)

11 pdate the subscription records

1 E RIC ubscription elete procedure (RIC Request I )

10a. E A time out

E RIC CRI TION A IT AI RE

E RIC CRI TION A IT RE ON E (Confirmed nknown and or

issing list of RIC Request I )

5 E RIC CRI TION A IT RE E T (Audit flag ist of RIC Request I )

##### Figure 9.3.2.5-1: E2 Subscription Audit procedure

#### E2 Query procedure

The purpose of the E2 Query procedure in the Near-RT RIC is to ensure that only authorized xApp may initiate RIC Query Request messages issued by the Near-RT RIC over the E2 interface to the E2 Node.

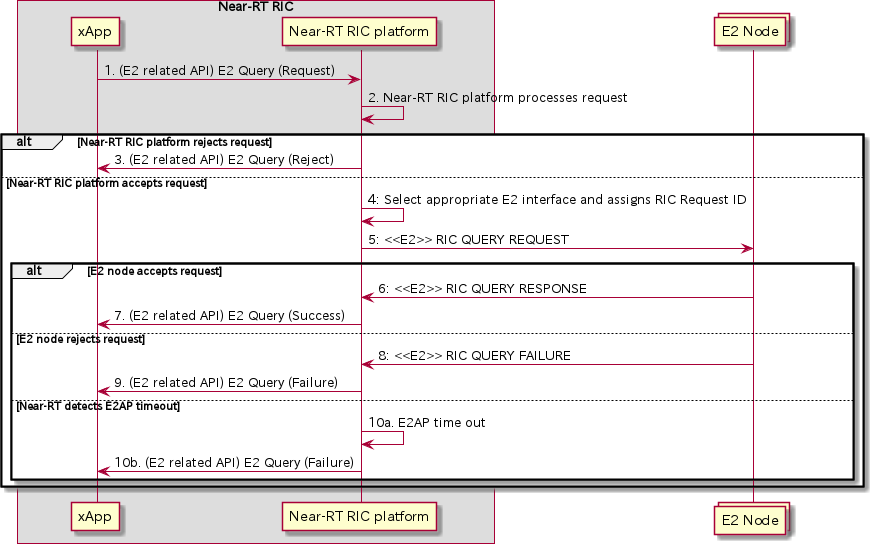
This procedure is based on the following assumptions:

* + - * + xApp has been configured with a trusted xApp ID;
        + Near-RT RIC platform may be configured to route E2 Query Request messages for a E2 node from xApp trigger directly towards an appropriate E2interface;
        + Near-RT RIC platform ensures that only authorized xApp may send E2 Query Request messages to appropriate E2 interface;
        + Near-RT RIC platform ensures that it systematically forwards any received RIC Query Response or RIC Query Failure to appropriate xApp.

The procedure is initiated by an xApp using E2 Related API: E2 Query request to send a request of a RIC Query for an E2 Node. The following outcomes are considered:

* + - * + Request successful following acceptance by Near-RT RIC platform and accepted by E2 Node and so an **E2 Related API: E2 Query** response (Success with outcome) is sent to xApp following corresponding E2 transaction;
        + Request fails following acceptance by Near-RT RIC platform but is either rejected by E2 Node or corresponding E2AP timer expires prior to a response from E2 Node and so an **E2 Related API: E2 Query** response (Failure with cause) is sent to xApp following corresponding E2 interface transaction (i.e., Request contents is not compliant, E2AP timeout, etc.);
        + Request rejected by Near-RT Platform and so an **E2 Related API: E2 Query** response (Reject) is sent to xApp.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | E2 Query procedure from xApp initiation to E2 Node and response. |
| Actors and Roles | * xApp: Originator of E2 Query request; * Near-RT RIC platform, with the following functionalities:   + E2 Termination. * E2 Node: RIC Query procedure. |
| Assumptions | * For E2 Query procedures, Near-RT RIC platform may forward the xApp's query request directly to the E2 interface; |
| Pre-conditions | * E2 Node has active E2 interface to Near-RT RIC; * Near-RT RIC has recovered complete list of active RAN Functions on E2 Node and informed initiating xApp; * xApp has been authorized to send RIC Query requests for a specific scope (E2 Node list, RAN Function, etc.); * Near-RT RIC platform has been configured to permit RIC Query requests from xApp. |
| Begins when | xApp determines need to propose RIC Query procedure for a E2 Node and defines message  contents (RAN Function ID, Query Header, Query Definition). |
| Step 1 (M) | xApp sends **E2 related API: E2 Query** request with message contents (RAN Function ID,  Query Header, Query Definition) for an E2 Node. |
| Step 2 (M) | Near-RT RIC platform processes request. |
|  | [ALT] Step 3 is executed if Near-RT RIC platform rejects proposed E2 Query message. |
| Step 3 (M) | **E2 related API: E2 Query** response (Reject with Cause) message is sent to xApp. |
|  | [ELSE] Step 4-10 are executed if Near-RT RIC platform accepts proposed E2 Query message: |
| Step 4 (M) | Near-RT RIC platform selects appropriate E2 interface and assigns an appropriate RIC  Request ID. |
| Step 5 (M) | Near-RT RIC platform sends **RIC Query Request** (RIC Request ID, contents) to E2 Node. |
|  | [ALT] Steps 6-7 are executed if E2 Node accepts RIC Query Request: |
| Step 6 (M) | E2 Node responds with **RIC Query Response** |
| Step 7 (M) | Near-RT RIC platform sends **E2 Related API: E2 Query** response (Success with  Outcome) to xApp. |
|  | [ELSE] Steps 8-9 are executed if E2 Node rejects RIC Query Request: |
| Step 8 (M) | E2 Node responds with **RIC Query Failure** (with Cause). |
| Step 9 (M) | Near-RT RIC platform sends **E2 Related API: E2 Query** response (Failure with Cause)  to xApp. |
|  | [ELSE] Step 10 is executed if E2 Node fails to respond prior to E2AP timer expires |
| Step 10 (M) | Near-RT RIC platform E2AP timer expires and sends **E2 Related API: E2 Query**  response (Failure with Cause) to xApp. |



##### Figure 9.3.2.6-1: E2 Query Procedure

#### E2 Subscription Modification procedure

The purpose of the E2 Subscription Modification procedure in the Near-RT RIC is to ensure that only validated RIC Subscription Modification Request messages are issued by the Near-RT RIC over the E2 interface to the E2 Node are handled gracefully.

This procedure is based on the following assumptions:

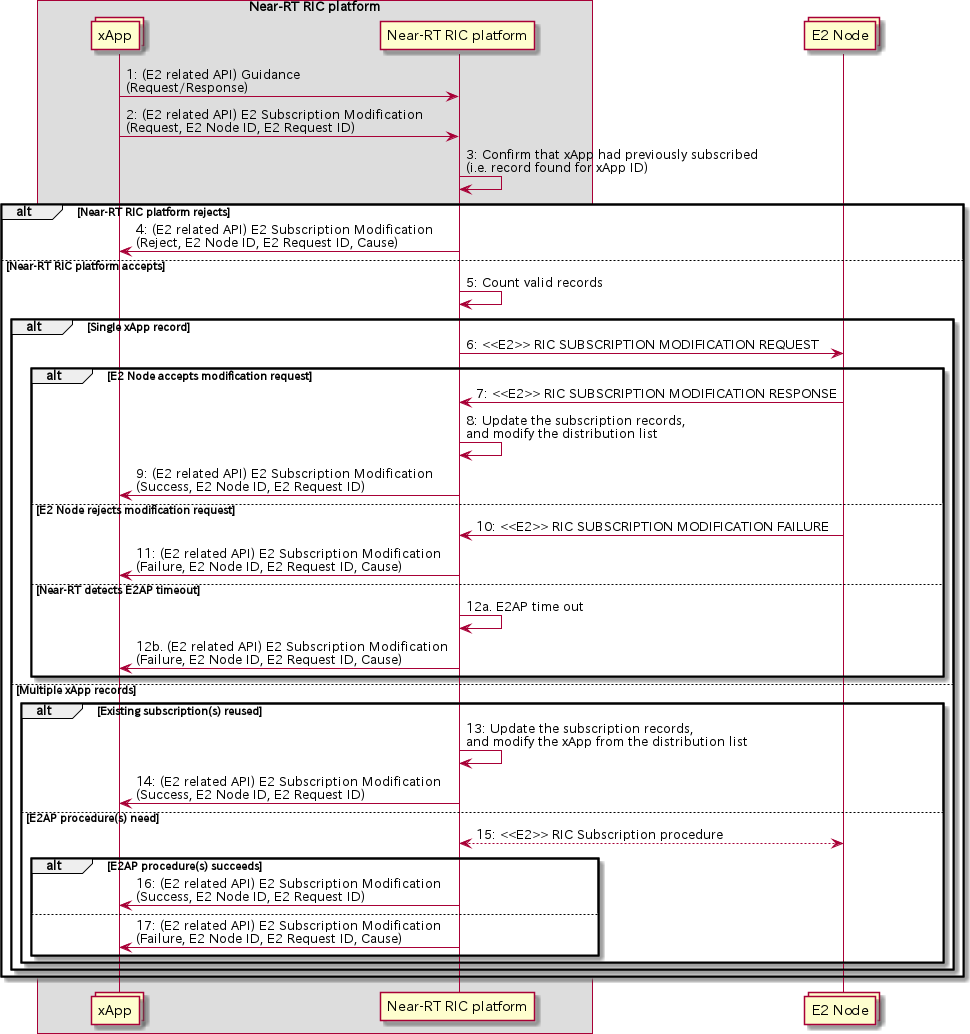
* + - * + xApp may obtain guidance from Near-RT RIC Platform (i.e., conflict mitigation processing) to resolve potential conflicts and/or detect partial duplications prior to sending a E2 Subscription Modification Request (see clause 9.3.3.1);
        + xApp has been configured with a trusted xApp ID;
        + Near-RT RIC platform maintains all previous successful RIC Subscriptions towards the E2 Node listed in xApp’s E2 Subscription request and is able to detect duplications and recover E2 Node response messages;
        + Near-RT RIC platform may leverage its conflict mitigation functionality in managing E2 subscriptions (see clause 9.3.3.5);
        + Near-RT RIC platform decides initiation of appropriate E2AP procedure(s), if needed, to fulfil the E2 Subscription request.
        + E2 Request ID is used on E2-related APIs to identify an active RIC subscription. Near-RT Platform maintains the 1:1 mapping between E2 Request ID, used on E2 related APIs, and RIC Request ID, used on E2 interface.
        + Near-RT RIC platform routes the subscription moodification requests to an appropriate E2interface;
        + Near-RT RIC platform maintains a mapping of active Subscriptions (identified on E2 interface by E2 Node ID and RIC Request ID) of validated xApps. Near-RT RIC platform uses the mapping when sending E2 INDICATION messages to the correct xApp or xApps.
        + Near-RT RIC platform sends the E2 Subscription Modification response prior to sending any associated E2 Indication message to the xApp.

The procedure initiated by an xApp uses **E2 Related API: E2 Subscription Modification** to request subscription modify for a specific E2 Node. The following outcomes are considered:

* + - * + Request fails following rejection by Near-RT RIC platform and an **E2 Related API: E2 Subscription Modification** response (Reject) is sent to xApp without E2 interface transaction (i.e., xApp not previously associated with validated subscription);
        + Request succeeds following acceptance by Near-RT RIC without initiating a corresponding E2 transaction; and an **E2 Related API: E2 Subscription Modification** response (Success) is sent to xApp;
        + Request succeeds following acceptance by Near-RT RIC platform and accepted by E2 Node, and an **E2 Related API: E2 Subscription Modification** response (Success) is sent to xApp following corresponding E2 transaction;
        + Request fails following acceptance by Near-RT RIC platform but is either rejected by E2 Node or corresponding E2AP timer expires prior to a response from E2 Node, and an **E2 Related API: E2 Subscription Modification** response (Failure) is sent to xApp following corresponding E2 interface transaction (i.e., RIC Request ID is not known by E2 Node, time out, etc.);

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | E2 Subscription Modification procedure from xApp initiation to E2 Node and response. |
| Actors and Roles | * xApp: Originator of E2 Subscription Modification request; * Near-RT RIC platform, with the following functionalities:   + Database;   + xApp Subscription Management;   + Conflict Mitigation;   + E2 Termination. * E2 Node: Handling RIC Subscription Modification procedure. |
| Assumptions | * Near-RT RIC platform maintains the E2 subscription records; * Near-RT RIC platform has access to sufficient information to both detect a potential conflict and take a decision on an optimal mitigation solution; * Near-RT RIC platform may generate guidance for internal processes and/or xApp as an optional addition response to Guidance Request. |
| Pre conditions | * E2 Node has active E2 interface to Near-RT RIC; * Near-RT RIC has recovered complete list of active RAN Functions on E2 Node and informed initiating xApp; * xApp has been authorized to issue E2 Subscription Modification API Requests; * xApp has been authorized to request guidance from Near-RT RIC platform; * Near-RT RIC platform has been configured to permit RIC Subscription Modification requests from xApp. |
| Begins when | xApp determines need to request E2 Subscription Modification for a specific E2 Node ID and E2  Request ID. |
| Step 1 (O) | xApp can request optional E2 related API: E2 Guidance from Near-RT RIC platform (see clause  9.3.3.1 for details). |
| Step 2 (M) | xApp sends **E2 related API: E2 Subscription Modification** request for a specific E2 Node and  E2 request ID. |
| Step 3 (M) | Near-RT RIC platform, after optional conflict mitigation processing (see clause 9.3.3.5 for  details), accepts or rejects proposal from xApp for specific E2 Node. |
|  | [ALT] Step 4 is executed if Near-RT RIC platform rejects the proposal: |
| Step 4 (M) | Near-RT RIC platform sends **E2 related API: E2 Subscription Modification** response to  xApp (Reject with Cause for specific E2 Node). |
|  | [ELSE] Steps 5-17 are executed if Near-RT RIC platform accepts the proposal: |
| Step 5 (M) | Near-RT RIC platform checks the subscription records to determine whether or not xApp  proposing Subscription Modification is the only associated xApp. |
|  | [ALT] Steps 6-12 are executed if the xApp proposing Subscription Modification is the only  xApp associated with the subscription (i.e., a given RIC Request ID): |
| Step 6 (M) | Near-RT RIC platform selects appropriate E2 interface and sends **RIC Subscription**  **Modification Request** (RIC Request ID) to E2 Node. |

|  |  |
| --- | --- |
|  | [ALT] Steps 7-9 are executed if E2 Node accepts Subscription Modification: |
| Step 7 (M) | E2 Node responds with **RIC Subscription Modification Response**. |
| Step 8 (M) | Near-RT RIC platform updates the subscription records, and modify the distribution  list(s) associated with the E2 Node ID and the RIC Request ID. |
| Step 9 (M) | Near-RT RIC platform sends **E2 related API: E2 Subscription Modification**  response to xApp (Success). |
|  | [ELSE] Steps 10-11 are executed if E2 Node rejects Subscription Modification: |
| Step 10 (M) | E2 Node responds with **RIC Subscription Modification Failure** (with Cause). |
| Step 11 (M) | Near-RT RIC platform sends **E2 related API: E2 Subscription Modification**  response to xApp (Failure with Cause). |
|  | [ELSE] Step 12 is executed if E2 Node fails to respond prior to E2AP timer expires |
| Step 12 (M) | Near-RT RIC platform E2AP timer expires and sends **E2 related API: E2**  **Subscription Modification** response (Failure with Cause) to xApp. |
|  | [ELSE] Steps 13-17 are executed if the xApp proposing Subscription Modification is one of a  number of xApp associated with the subscription (i.e., a given RIC Request ID): |
|  | [ALT] Steps 13-14 executed if the proposal can be fulfilled with existing subscription(s) (i.e., one or more records found for E2 Node ID with Subscription matching the same  contents): |
| Step 13 (M) | Near-RT RIC platform updates the subscription records. For each Action in subscription corresponding to an Indication message, Near-RT RIC platform adds  xApp ID to distribution list associated with E2 Node ID, RIC Request ID and Action ID. |
| Step 14 (M) | xApp Subscription Management sends **E2 Related API: E2 Subscription**  **Modification** response to xApp (Success, E2 Node ID, corresponding E2 Request ID(s) and E2 Action ID(s)). |
|  | [ELSE] Steps 15-17 are executed if the subscription proposed by xApp necessitates  E2AP procedure(s) (i.e., no record found for E2 Node ID with Subscription matching the same contents): |
| Step 15 (M) | Near-RT RIC platform selects appropriate E2 interface and initiates appropriate E2AP  procedure(s) to E2 Node. |
|  | [ALT] Step 16 is executed if the E2AP procedure(s) in Step 15 succeeds: |
| Step 16 (M) | Near-RT RIC platform sends **E2 related API: E2 Subscription Modification**  response to xApp (Success, E2 Node ID, corresponding E2 Request ID(s) and E2 Action ID(s)). |
|  | [ELSE] Step 17 is executed if the E2AP procedure(s) in Step 15 fails: |
| Step 17 (M) | Near-RT RIC platform sends **E2 related API: E2 Subscription Modification**  response to xApp (Failure with Cause, E2 Node ID). |



##### Figure 9.3.2.7-1: E2 Subscription Modification procedure

### E2 Conflict Mitigation related procedures

#### E2 Guidance procedure

The purpose of the xApp initiated E2 Guidance request/response API procedure in the Near-RT RIC is to allow authorized xApp to obtain guidance from the Near-RT RIC platform (with the conflict mitigation functionality) prior to initiating an action.

Guidance from Near-RT RIC platform may include:

* + - * + Indication on whether or not the xApp proposed E2 Related API message or series of messages may result in a conflict with E2 related API messages from other xApps;
        + Recommendations on how the proposed E2 Related message or series of messages should be modified to avoid conflict;
        + Modification of previous guidance to other xApps and/or Near-RT RIC platform internal processes.

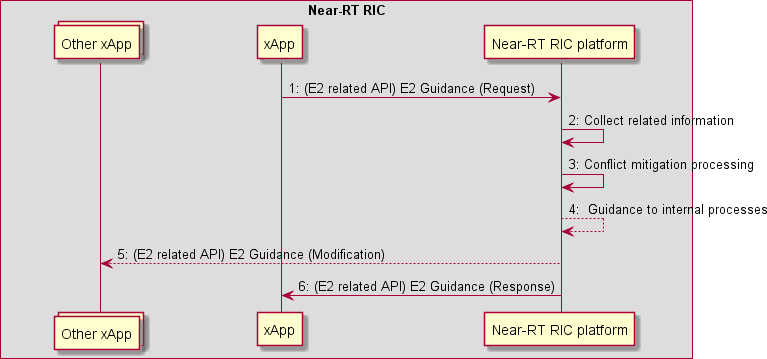
This procedure is based on the following assumptions:

* + - * + xApp may use E2 Related API to obtain guidance from Near-RT RIC platform to resolve potential conflicts prior to initiating a RIC function procedure;
        + xApp may use Near-RT RIC platform response in a subsequent procedure (i.e., for a RIC Functional Procedure).

This procedure is initiated by an xApp using **E2 Related API: E2 Guidance** request. The following outcomes are considered:

* + - * + Near-RT RIC Platform provides guidance to requesting xApp using E2 Related API: E2 Guidance response;
        + Near-RT RIC Platform provides modified guidance to another xApp and/or its internal processes.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | xApp initiation of Conflict Mitigation guidance. |
| Actors and Roles | * xApp: Originator of Conflict Mitigation guidance request; * Near-RT RIC platform, with the following functionalities:   + Database;   + Conflict mitigation. |
| Assumptions | * Near-RT RIC platform has access to sufficient information to both detect a potential conflict and take a decision on an optimal mitigation solution; * Near-RT RIC platform may initiate guidance towards other xApp as an optional addition response to Guidance Request. |
| Pre-conditions | * xApp has been authorized to request guidance from Near-RT RIC platform for conflict mitigation; * xApp has been assigned xApp ID. |
| Begins when | xApp determines need to request guidance from Near-RT RIC platform for conflict mitigation. |
| Step 1 (M) | xApp sends **E2 Related API E2 Guidance** Request to Near-RT RIC platform. |
| Step 2 (M) | Near-RT RIC platform collects related information. |
| Step 3 (M) | Near-RT RIC platform processes request. |
| Step 4-5 (O) | Near-RT RIC platform may signal conflict and/or provide guidance to internal processes and/or  other xApps (see clause 9.3.3.5). |
| Step 6 (M) | Near-RT RIC platform sends **E2 Related API E2 Guidance** response to xApp. |
| Ends with | xApp continues processing using guidance response. |



##### Figure 9.3.3.1-1: xApp initiated E2 guidance request/response procedure

#### Void

#### Void

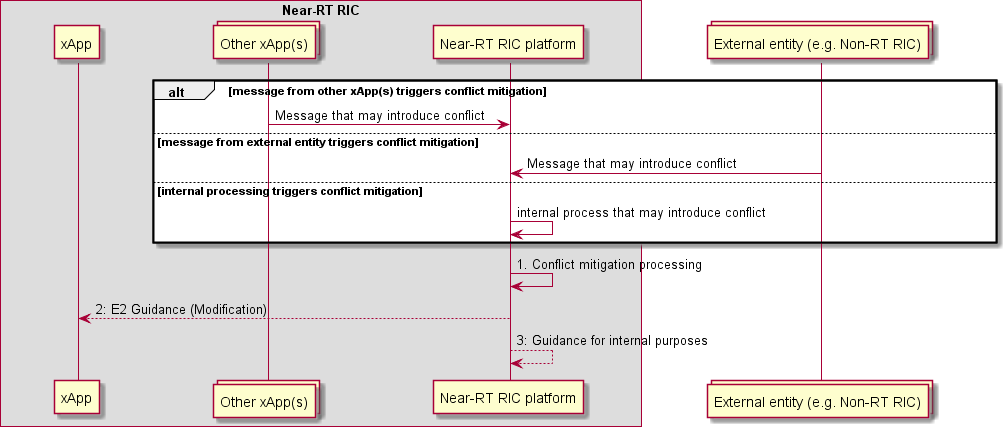
#### Void

#### E2 Guidance modification procedure

The purpose of the E2 Guidance modification procedure is to allow Near-RT RIC platform to signal a modified guidance to the xApp, which can be triggered by the Near-RT RIC platform’s internal process (e.g., xApp subscription management), or by a message from the other xApps or external interfaces.

The guidance may also be used for Near-RT RIC platform internal purposes.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | Near-RT RIC platform indicates guidance to an xApp without request from the xApp. |
| Actors and Roles | * xApp * Near-RT RIC platform, with the following functionalities:   + Database   + Conflict mitigation * External entity (e.g., E2 Node) |
| Assumptions | * Near-RT RIC platform is capable of collecting related information, and generating proper guidance for an xApp. |
| Pre-conditions | * Message from xApp, message from external interfaces, or Near-RT RIC platform’s internal process triggers conflict mitigation in Near-RT RIC platform. |
| Step 1 (M) | Near-RT RIC platform identifies affected xApps and performs conflict mitigation  processing. |
| Step 2 (O) | Near-RT RIC platform may provide guidance to the affected xApp using E2 Guidance  modification message. |
| Step 3 (O) | Near-RT RIC platform may also use the guidance for internal purposes. |



##### Figure 9.3.3.5-1 E2 Guidance Modification Procedure

#### E2 Conflict Mitigation Assistance procedure

The purpose of the Near-RT RIC platform initiated E2 Conflict Mitigation Assistance request/response API procedure in the Near-RT RIC is to allow authorized helper and conflicting xApps to provide assistance to Near-RT RIC platform during conflict detection, conflict avoidance, and conflict resolution.

Assistance from helper and conflicting xApps may consist of the following:

* + - * + Helper xApp: E2SM processing related to conflict detection, avoidance, and resolution. This may include a helper xApp providing the platform with support for data analysis, optimization, or other E2SM related computations;
        + Helper xApp: Assistance may include a helper xApp collecting and processing relevant data from E2 Nodes or Near- RT RIC platform;
        + Conflicting xApps: Assitance for computation of an avoidance solution, e.g., conflicting xApps may agree to a platform-recommended mutually exclusive subset of E2SM defined RIC services;
        + Conflicting xApps: Upon request, conflicting xApps may provide the platform with E2SM related information (e.g., RIC services, RAN parameters or RAN KPIs of interest to xApps and preferences regarding these).

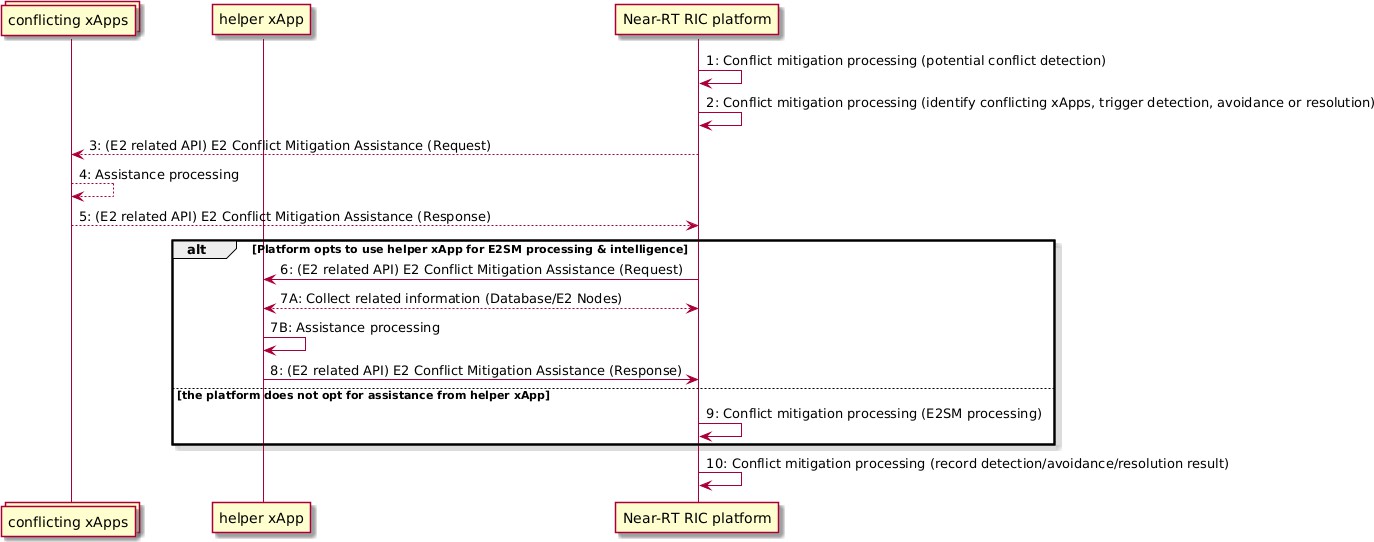
This procedure is based on the following assumptions:

* + - * + Whether an xApp supports a certain type of assistance listed above is assumed to be known to the platform before requesting the assistance;
        + The Near-RT RIC platform can request assistance from an xApp if and only if the xApp supports it. Furthermore, the decision to request this assistance is Near-RT RIC platform implementation specific.

This procedure is initiated by Near-RT RIC platform using **E2 Related API: E2 Conflict Mitigation Assistance** request. The xApps provide the required assitance using E2 Related API: E2 Conflict Mitigation Assistance response.

The procedure for E2 Conflict Mitigation Assistance for conflict detection, avoidance, or resolution is provided below:

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | Near-RT RIC platform initiated Conflict mitigation (detection, avoidance or resolution) using E2  Conflict Mitigation Assistance. |
| Actors and Roles | * Conflicting or potentially conflicting xApps, as producers of E2 Conflict Mitigation assistance (towards Near-RT RIC platform) related to detection, avoidance, or resolution of conflicts. This assistance may comprise the following:   + Assitance for computation of an avoidance solution;   + Providing the platform with E2SM related information relevant to a conflict. * Helper xApp as producer of E2 Conflict Mitigation assistance (towards Near-RT RIC platform) which may comprise the following:   + E2AP and E2SM processing support to the platform;   + Data analysis, optimization, and data collection from platform or E2 Nodes; * Near-RT RIC platform as consumer of E2 Conflict Mitigation assistance, |
| Assumptions | * Near-RT RIC platform has access to sufficient information to detect a potential conflict; * xApps can provide assistance for conflict mitigation to the Near-RT RIC platform; * Near-RT RIC platform has access to sufficient information to determine which xApps offer which type of conflict mitigation assistance. |
| Pre-conditions | * A previous event (e.g., xApp initiated E2 Guidance Request or E2 Subscription/Control Request) has triggered conflict mitigation in the Near-RT RIC Platform. * In response, the platform has triggered one of the conflict detection, avoidance or resolution processes. |
| Begins when | A triggering event as mentioned in the pre-conditions. |
| Step 1 (M) | Near-RT RIC platform detects potential conflict. |
| Step 2 (M) | Near-RT RIC platform identifies potentially conflicting xApps and triggers conflict detection,  avoidance, or resolution |
| Step 3 (O) | Near-RT RIC platform sends **E2 Conflict Mitigation Assistance** request to xApps. |
| Step 4 (O) | xApps process the request and generate Assistance information. |
| Step 5 (O) | xApps send **E2 Conflict Mitigation Assistance** response to Near-RT RIC platform. |
|  | [ALT] Steps 6-8 are executed if the Near-RT RIC platform opts to use helper xApp for E2SM  processing and intelligence. |
| Step 6 (M) | Near-RT RIC platform sends **E2 Conflict Mitigation Assistance** to the helper xApp. |
| Step 7A (O) | The helper xApp may gather additional information from the platform (e.g., any available  dependency information between RAN control parameters and KPIs) or E2 Nodes (e.g., KPIs) using appropriate procedures. |
| Step 7B (M) | Helper xApp procesess the request and generates the Assistance information. |
| Step 8 (M) | Helper xApp sends **E2 Conflict Mitigation Assistance** response to Near-RT RIC platform. |
|  | [ELSE] Step 9 is executed if Near-RT RIC platform does not opt for assistance from a helper  xApp. |
| Step 9 (M) | Near-RT RIC platform perform the required E2SM processing itself. |
| Step 10 (M) | Near-RT RIC platform generates the information/result related to detection, avoidance, or  resolution. |
| Ends with | Conflict detection, avoidance, or resolution. |



##### Figure 9.3.3.6-1: E2 Conflict Mitigation Processing with Assistance

## Management API Procedures

### xApp Registration procedure

This procedure registers the xApp to the Near-RT RIC platform so that it can be managed by the SMO via the Near-RT RIC platform. When the xApp is deployed, it will register itself as a managed application to the Near-RT RIC platform by passing the relevant info through the API. Relevant info may include the xApp name, vendor, software version, YANG schemas for configuration management, faults raised, metrics generated, ports for messaging, supported commands (e.g., health check, aliveness probe) and other information needed by the Near-RT RIC platform and/or the SMO for managing the xApp.

In response to an xApp Registration, Near-RT RIC platform performs the following:

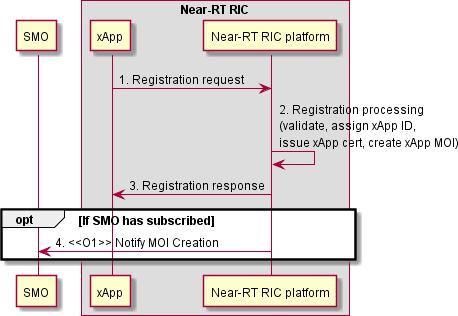
* Authenticates the xApp;
* Validates that this xApp can run on this Near-RT RIC platform; e.g., RIC has the capacity;
* Assigns an ID to the xApp;
* May act as Registration Authority (RA) of the operator Certification Authority (CA), and fetch from the CA a new End Entity (EE) operator certificate for the xApp containing the assigned xApp ID in SubjectAltName (SAN) field.

NOTE: Refer to [22] clause 5.1.3.2 for more detailed information on the security procedure.

* Creates an xApp Managed Object Instance (MOI) in the Near-RT RIC Config DB and populates it with the relevant information passed from the xApp;
* Sends a Notify MOI Creation to SMO if subscribed.

The data type of the xApp ID shall be a character string that uniquely identifies the xApp instance. The format of this string shall be a Universally Unique Identifier (UUID) version 4 (as described in IETF RFC 4122).

|  |  |  |
| --- | --- | --- |
| **Use Case**  **Stage** | **Evolution / Specification** | **<<Uses>>**  **Related use** |
| Goal | To register the newly deployed xApp to the Near-RT RIC  platform. |  |
| Actors and Roles | * xApp: originator of the xApp registration procedure; * Near-RT RIC platform, with the following functionalities:   + Management |  |
| Assumptions | The API channel is established successfully between xApp and  the Near-RT RIC platform. |  |
| Pre conditions | The xApp is deployed onto the Near-RT RIC platform. As part of the initial configuration security information used in the security procedure defined in [22] clause 5.1.3.2 is provided to the xApp. xApp has established a secure connection with the Near-RT RIC  platform. |  |
| Begins when | An xApp is deployed and ready for registration. |  |
| Step 1 (M) | xApp send xApp registration request to the Near-RT RIC platform. passing relevant information needed to manage the  xApp. |  |
| Step 2 (M) | Near-RT RIC platform processes the application registration, which includes (not an exhaustive list):   * Performing authentication and validity checks; * If the checks pass, assigning an xApp ID for the xApp; * Performing the security procedure specified in [22] clause 5.1.3.2; * Creating an xApp Managed Object Instance in the Near-RT RIC Config DB in compliance with the YANG schemas; * Populating xApp MOI with the xApp ID and the initial xApp configuration. |  |
| Step 3 (M) | The Near-RT RIC platform send the registration response to the  xApp. If the response indicates the registration fails, it will send the registration response with failure results and failure cause. |  |
| Step 4 (O) | If SMO has subscribed to CM Notifications from the Near-RT RIC, Near-RT RIC platform formats and sends a Notify MOI Creation to SMO to notify SMO that a MOI has been created for  a new xApp. |  |



##### Figure 9.4.1-1: xApp Registration procedure

### xApp Deregistration procedure

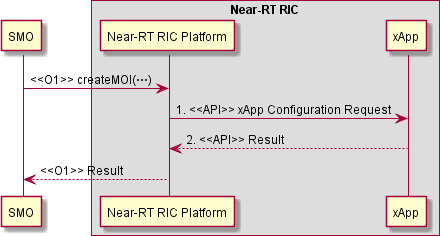
This procedure deregisters the xApp from the Near-RT RIC platform. The message flow is not specified in the present document.

### Void

### Create MOI

An xApp must be instantiated before its associated MOI can be created. The SMO interacts with the MOI to manage the xApp. The Near-RT RIC platform is responsible for updating the xApp over the Management API to ensure that its attribute values are aligned with those of the MOI.

|  |  |  |
| --- | --- | --- |
| **Use Case Stage** | **Evolution / Specification** | **<<Uses>>**  **Related use** |
| Goal | To Create a Managed Object Instance for an xApp. |  |
| Actors and Roles | * Near RT-RIC platform * xApp |  |
| Assumptions | The API channel is established successfully between xApp and the Near-  RT RIC platform. |  |
| Preconditions | * xApp is instantiated; * MOI is not yet created. |  |
| Begins when | Create MOI operation received over O1 |  |
| Step 1 (M) | Near-RT RIC platform sends an APIConfirgurationWrite updating the xApp  with the attribute values from the newly created MOI |  |
| Step 2 (M) | After applying the configuration data to the xApp, a configuration response  is sent over the management API to the Near-RT RIC platform. |  |
| Ends when | Create MOI Response is sent over O1. |  |

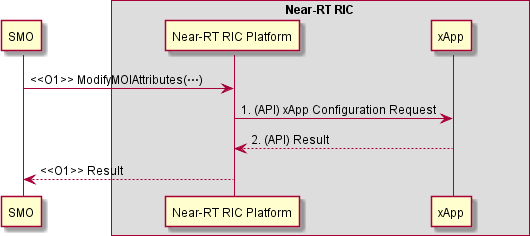


##### Figure 9.4.4-1: xApp Create MOI

### Modify MOI attributes

The Near-RT RIC platform is responsible for updating the xApp over the Management API to ensure its attribute values are aligned with those of the MOI.

|  |  |  |
| --- | --- | --- |
| **Use Case Stage** | **Evolution / Specification** | **<<Uses>>**  **Related use** |
| Goal | To change an existing xApp MOI configuration. |  |
| Actors and Roles | * Near RT-RIC platform * xApp |  |
| Assumptions | * The API channel is established successfully between xApp and the Near-RT RIC platform. |  |
| Pre conditions | * The xApp is installed successfully on the Near-RT RIC platform and registered to the SMO; * The MOI for the target xApp already exists. |  |
| Begins when | Near-RT RIC platform receives the configuration message from SMO to  indicate that there’s a configuration update for the xApp. |  |
| Step 1 (M) | Near-RT RIC platform sends an APIConfirgurationWrite updating the xApp  with the attribute values contained in the O1 Modify MOI message. |  |
| Step 2 (M) | The xApp sends a Configuration Response. |  |
| Ends when | Near-RT RIC platform notifies SMO of success or error. |  |

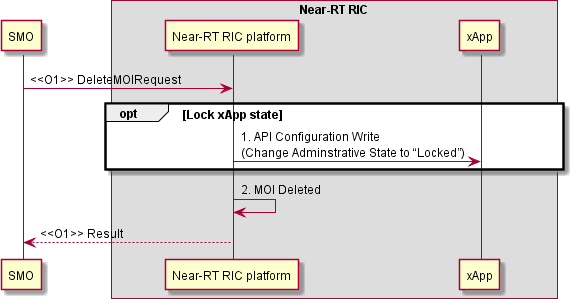


##### Figure 9.4.5-1: xApp Modify MOI

### Delete MOI

This procedure removes the MOI for an an xApp when an edit-config delete request is received from the SMO over the O1 interface. See [5] clause 2.1.3. The SMO sends the deleteMOI request via a NETCONF command over the O1 to the Near-RT RIC platform. The Near-RT RIC platform may optionally take the xApp out of service sending a confirmation of the state change before the MOI gets deleted. Once the MOI is deleted, the Near-RT RIC platform sends a NETCONF response confirming the success or failure to the MnS Consumer (SMO).

|  |  |  |
| --- | --- | --- |
| **Use Case Stage** | **Evolution / Specification** | **<<Uses>>**  **Related use** |
| Goal | To remove an existing xApp MOI. |  |
| Actors and Roles | * Near-RT RIC platform * xApps |  |
| Assumptions | The API channel is established successfully between xApp and the Near-  RT RIC platform. |  |
| Pre conditions | The MOI exists on the target Near-RT RIC. |  |
| Begins when | Near-RT RIC platform receives the edit-config Delete message from SMO  to indicate that the SMO wants to delete the MOI. |  |
| Step 1 (O) | Near-RT RIC platform may optionally lock the administrative state of the  xApp. |  |
| Step 2 (M) | Near-RT RIC platform deletes the MOI. |  |
| Ends when | The MnS Consumer (SMO) receives the O1 response. |  |

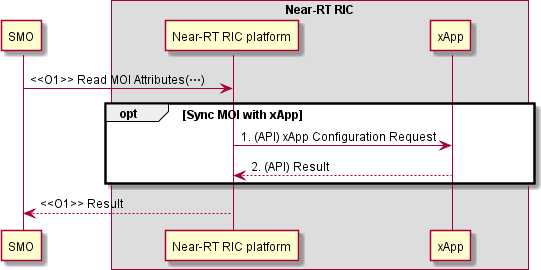


##### Figure 9.4.6-1: xApp Delete MOI

### Read MOI attributes

The O1 Read MOI Attributes service is supported by the xApp API Configuration Read. On receipt of a Read MOI Attributes request for an xApp, the Near-RT RIC platform uses the API Configuration Read to retrieve the current configuration values.

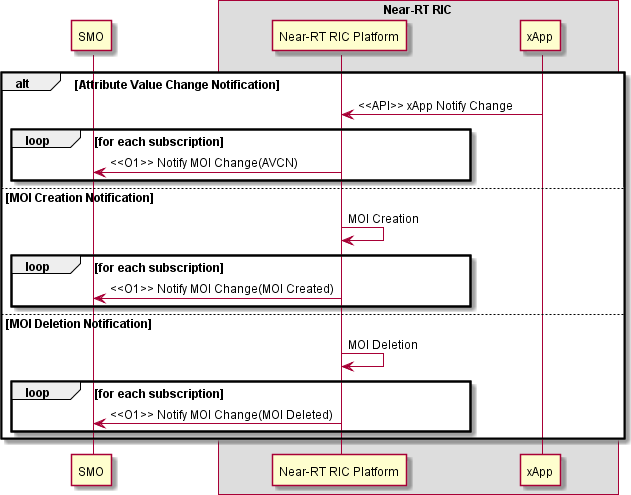
|  |  |  |
| --- | --- | --- |
| **Use Case Stage** | **Evolution / Specification** | **<<Uses>>**  **Related use** |
| Goal | To Read the current attribute values for an existing xApp MOI. |  |
| Actors and Roles | * Near-RT RIC platform * xApps |  |
| Assumptions | * The API channel is established successfully between xApp and the Near-RT RIC platform. |  |
| Pre conditions | * The MOI exists on the target Near-RT RIC. |  |
| Begins when | Near-RT RIC platform receives the Read MOI Attributes request from SMO  to indicate that the SMO wants the current xApp attributes. |  |
| Step 1 (O) | Near-RT RIC platform sends xApp Configuration Read  (Optional because the information is already available in the MOI in the Near-RT RIC platform ). |  |
| Step 2 (O) | xApp makes the operational configuration available to the Near-RT RIC  platform. |  |
| Ends when | The MnS Consumer (SMO) receives the O1 response. |  |



##### Figure 9.4.7-1: Read MOI Attributes

### Notify MOI changes

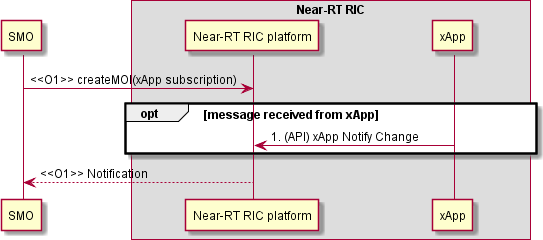
|  |  |  |
| --- | --- | --- |
| **Use Case Stage** | **Evolution / Specification** | **<<Uses>>**  **Related use** |
| Goal | To notify Near-RT RIC platform that some change has occurred in  the xApp configuration |  |
| Actors and Roles | * Near-RT RIC platform * xApps |  |
| Assumptions | * The API connection is established successfully between xApp and the Near-RT RIC platform. |  |
| Pre conditions | * The MOI exists on the target Near-RT RIC. |  |
|  | [ALT 1] Attribute Value Change Notification (AVCN) |  |
| Begins when | The Near-RT RIC platform receives an xApp Notify change message  from the xApp. |  |
| Step 1 (M) | The Near-RT RIC platform sends O1 Attribute Value Change  Notification to the O1 Consumer (SMO). |  |
|  | [ALT 2] MOI Creation Notification |  |
| Begins when | A Managed Object Instance representing xApp has been created. |  |
| Step 1 (M) | The Near-RT RIC platform sends O1 MOI Creation Notification to the  O1 Consumer (SMO). |  |
|  | [ALT 3] MOI Deletion Notification |  |
| Begins when | A Managed Object Instance representing xApp has been Deleted. |  |
| Step 1(M) | The Near-RT RIC platform sends O1 MOI Deletion Notification to the  O1 Consumer (SMO). |  |
|  | [ALL CASES] |  |
| Ends when | The MnS Consumer (SMO) receives the O1 response. |  |



##### Figure 9.4.8-1: Notify MOI Changes

### Subscription Control

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | Allows a MnS Consumer to subscribe to notifications emitted by a MnS Provider. |
| Actors and Roles | * Near-RT RIC platform * xApps |
| Assumptions | * The API channel is established successfully between xApp and the Near-RT RIC platform. |
| Pre conditions | * An MOI exists on the target Near-RT RIC representing the xApp functionality. |
| Begins when | Near-RT RIC platform receives a Create MOI message with arguments defining the MOI as a  Subscription and associating it with one or more xApps and a consumer (SMO). |
| Step 1 (M) | A Subscription Control MOI reflecting the scope and filtering requirements for the subscription is  created and associated with the instance of Managed Application representing the xApp. |
| Ends when | The MnS Consumer (SMO) receives the O1 response. |

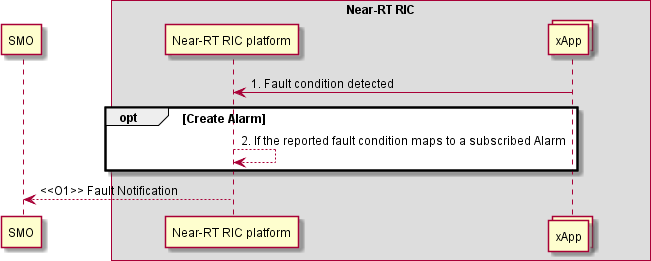


##### Figure 9.4.9-1: xApp Subscription Control

### Fault Notification

This procedure enables the xApps to notify the Near-RT RIC of a Fault Condition. Near-RT RIC platform may create an Alarm notification based on the fault reported over the API, and supplement it with additional information from the Alarm Dictionary. The notification is also used to notify Alarms changes.

|  |  |  |
| --- | --- | --- |
| **Use case stage** | **Evolution/Specification** | **<<Uses>>**  **Related use** |
| Goal | xApp informs the Near-RT RIC of the existence of a Fault condition |  |
| Actors and Roles | * xApp * Near-RT RIC platform |  |
| Assumptions |  |  |
| Pre-conditions |  |  |
| Begins when | A fault condition is detected at the xApp. |  |
| Step 1 (M) | xApp Management API sends a fault notification. |  |
| Step 2 (M) | Near-RT RIC platform determines whether to create an Alarm notification for the  fault. |  |
| Ends when | xApp sends Fault Notification to the Near-RT RIC. |  |



##### Figure 9.4.10-1: Fault Notification

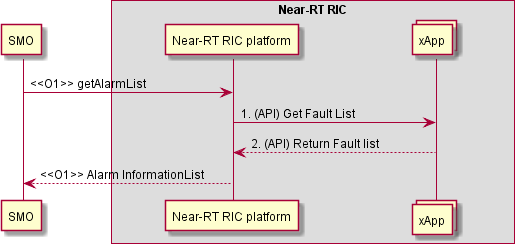
### Fault Supervision Control

The O1 Interface supports the 3GPP “getAlarmList”. The Alarm list for the xApps is maintained on the Near-RT RIC.

The xApp can notify the Near-RT RIC platform of a detected fault. Each xApp shall maintain a list of any faults that are active. These faults are notified to the Near RT RIC platform where the necessary Alarms are created.

The SMO (consumer) may need to verify which alarms are currently active and so may request that the Near-RT RIC (producer) provides an updated list of Alarms. In order to verify that its Alarm list correctly reflects the state of faults at the xApp, the Near RT RIC platform can request an updated list of faults from the xApp.

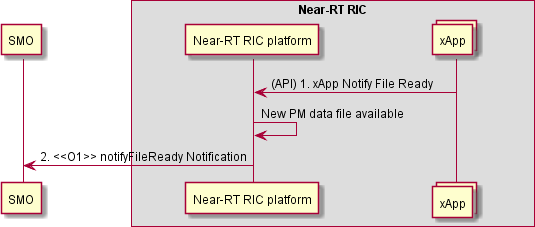
|  |  |  |
| --- | --- | --- |
| **Use case stage** | **Evolution/Specification** | **<<Uses>>**  **Related use** |
| Goal | To validate that the Near-RT RIC Alarm list for a given xApp correctly reflects the  fault status of that xApp |  |
| Actors and Roles | * SMO * Near-RT RIC platform * xApp |  |
| Assumptions |  |  |
| Pre-conditions | Target xApp in service. |  |
| Begins when | O1 consumer (SMO) requests Near-RT RIC platform to fetch the current Alarm  List. |  |
| Step 1 (M) | Near-RT RIC platform requests xApp to provide the list of current faults. |  |
| Step 2 (M) | xApp sends list of current faults to Near-RT RIC platform. |  |
| Ends when | O1 consumer (SMO) receives current Alarm list. |  |



##### Figure 9.4.11-1: Get Alarm List

### Performance Data File Reporting

|  |  |  |
| --- | --- | --- |
| **Use case stage** | **Evolution/Specification** | **<<Uses>>**  **Related use** |
| Goal | xApp informs Near-RT RIC platform that a file is ready and provides the location  for that file. |  |
| Actors and Roles | * SMO * Near-RT RIC platform * xApp |  |
| Assumptions |  |  |
| Pre-conditions | PerfMetricJob exists and includes a valid stream target value. |  |
| Begins when | PM Streaming configuration created. |  |
| Step 1 (M) | xApp sends API notification that a file is ready also providing the location of that  file. |  |
| Step 2 (M) | <<O1>> File Ready notification sent. |  |
| Ends when | SMO receives notification. |  |

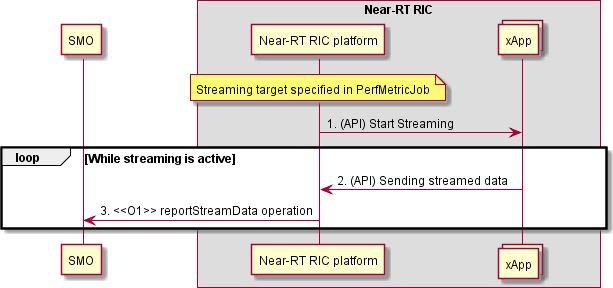


##### Figure 9.4.12-1: Performance Data File Reporting

### Report Streamed Data

When the ReportingCtrl Attribute of the PerfMetricJob ([21] Clause 4.3.33) is set to streamTarget, the Near-RT RIC shall stream the PM data to the target consumer destination based on the value of the “streamTarget” attribute.

|  |  |  |
| --- | --- | --- |
| **Use case stage** | **Evolution/Specification** | **<<Uses>>**  **Related use** |
| Goal | Report Streamed Data |  |
| Actors and Roles | * SMO * Near-RT RIC platform * xApp |  |
| Assumptions |  |  |
| Pre-conditions | PerfMetricJob exists and includes a valid stream target value.  A streaming connection has been established between the Near-RT RIC and the SMO. |  |
| Begins when | PM Streaming configuration created. |  |
| Step 1 (M) | (API) Near-RT RIC platform directs xApp to start streaming data. |  |
| Step 2 (M) | (API) xApp streams data to Near-RT RIC platform. |  |
| Step 3 (M) | Near-RT RIC platform streams PM data to subscriber. |  |
| Ends when | SMO receives streamed data. |  |

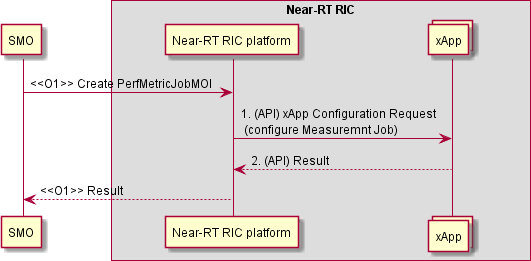


##### Figure 9.4.13-1 Report Streamed Data

### Measurement Job Control

Both the selection of the data at the xApp and the formatting of the data at the Near-RT RIC platform are in scope for this procedure. The CreateMeasurementJobControl request received over O1 also determines whether the PM data would be file based or streamed.

|  |  |  |
| --- | --- | --- |
| **Use case stage** | **Evolution/Specification** | **<<Uses>>**  **Related use** |
| Goal | To configure the collection of PM measurements |  |
| Actors and Roles | * SMO * Near-RT RIC platform * xApp |  |
| Assumptions |  |  |
| Pre-conditions | Target xApp in service. |  |
| Begins when | “Create erf etricJob” recei ed on O1 interface. |  |
| Step 1 (M) | Configure xApp to collect and forward PM data. |  |
| Step 2 (M) | PM collection response. |  |
| Ends when | Successful creation of PM PerformanceMetric Job confirmed on O1. |  |



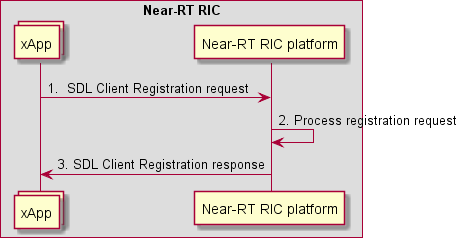
##### Figure 9.4.14-1 Measurement Job Control

## SDL API Procedures

### SDL Client Registration procedure

The SDL Client Registration procedure in the Near-RT RIC enables an xApp to register with the Near-RT RIC platform for permission to access the database.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | xApp requests Near-RT RIC platform as a client for the permission to access database. |
| Actors and Roles | * xApp: Originator of SDL Client Registration request; * Near-RT RIC platform: Response to the request of xApp. |
| Pre-conditions | * SDL API services have been registered and authorized by Near-RT RIC platform. |
| Begins when | xApp determines to register with Near-RT RIC platform to access the database in the Near-RT-  RIC platform. |
| Step 1 (M) | xApp sends SDL Client Registration request to Near-RT RIC platform to obtain permission to  access the database. |
| Step 2 (M) | Near-RT RIC platform arranges the permission for the xApp to access the database. |
| Step 3 (M) | Near-RT RIC platform sends the response of successful or failed registration to the xApp. |

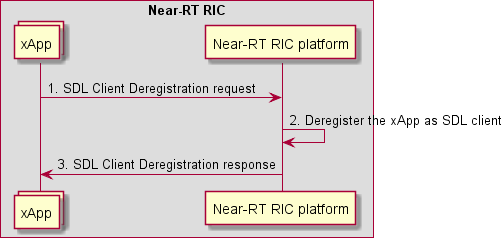


##### Figure 9.5.1-1: SDL Client Registration procedure

### SDL Client Deregistration procedure

The SDL Client Deregistration procedure in the Near-RT RIC enables an xApp to request the Near-RT RIC platform to release the registration as SDL client.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | xApp requests the Near-RT RIC platform to release the registration as SDL client |
| Actors and Roles | * xApp: Originator of SDL Client Deregistration request; * Near-RT RIC platform: Response to the request of xApp. |
| Pre-conditions | * SDL API services have been registered with Near-RT RIC platform; * xApp has successfully registered as a client to the SDL before. |
| Begins when | xApp determines to release the registration with Near-RT RIC platform. |
| Step 1 (M) | xApp sends Client Deregistration request to Near-RT RIC platform to release the connection  with database. |
| Step 2 (M) | Near-RT RIC platform deregisters the xApp as a SDL client. |
| Step 3 (M) | Near-RT RIC platform sends the response of successful or failed deregistration to the xApp. |

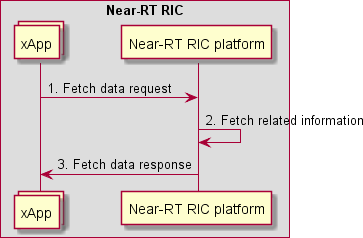


##### Figure 9.5.2-1: SDL Client Deregistration procedure

### Fetch Data procedure

The Fetch Data procedure in the Near-RT RIC enables an xApp to request data for which it is authorized as the SDL client for local processing.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | xApp fetches data as the SDL client. |
| Actors and Roles | * xApp: Originator of Fetch Data request; * Near-RT RIC platform: Response to the request of xApp. |
| Pre-conditions | * SDL API services have been registered with Near-RT RIC platform; * xApp has successfully registered as a client to the SDL before; * xApp has been authorized to read the data in the database. |
| Begins when | xApp determines to fetch relevant data as the SDL client from the Near-RT-RIC platform. |
| Step 1 (M) | xApp requests to Near-RT RIC platform to fetch data from database. |
| Step 2 (M) | Near-RT RIC platform fetches xApp required information from database. |
| Step 3 (M) | Near-RT RIC platform sends the requested data or the failure response to specific xApp. |



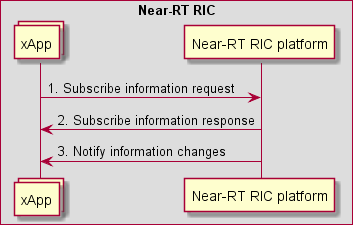
##### Figure 9.5.3-1: Fetch Data procedure

### Subscribe/Update-Notify procedure

The Subscribe/Update-Notify procedure in the Near-RT RIC allows the xApp to subscribe to SDL notifications for the summary of authorized data changes. The Near-RT RIC platform will notify the summary of information changes to the xApp, and the xApp may subsequently request updated data through the Fetch Data procedure.

The notifications contain only summary of changes to the information after the subscription.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | xApp subscribes with the Near-RT RIC platform for data changes and receives related  notifications. |
| Actors and Roles | * xApp: Originator of Subscribe request; * Near-RT RIC platform: Response to the request of xApp, and notify the subscribed information to xApp. |
| Pre-conditions | * SDL API services have been registered by Near-RT RIC platform; * xApp has successfully registered as a client to the SDL before. |
| Begins when | xApp determines to subscribe for notifications related to summary of information changes in the  Near-RT-RIC platform database. |
| Step 1 (M) | xApp sends the request to Near-RT RIC platform to subscribe for notifications related to  summary of SDL information changes. |
| Step 2 (M) | Near-RT RIC platform responds to the xApp subscription. |
| Step 3 (M) | Near-RT RIC platform sends the notification for the subscribed summary of information changes to the xApp. The xApp may further send a request to Near-RT RIC platform to fetch the updated  information. |

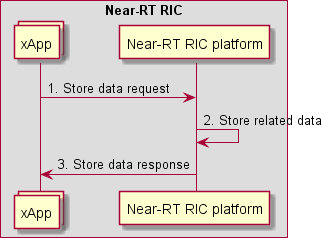


##### Figure 9.5.4-1: Subscribe/Update-Notify procedure

### Store Data procedure

The Store Data procedure in the Near-RT RIC ensures that the xApp may request Near-RT RIC platform to insert data to the database.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | xApp requests to store data in the database. |
| Actors and Roles | * xApp: Originator of Store Data request; * Near-RT RIC platform: Response to the request of xApp. |
| Pre-conditions | * SDL API services have been registered by Near-RT RIC platform; * xApp has successfully registered as a client to the SDL before; * xApp has been authorized to write the data in the database. |
| Begins when | xApp determines to store relevant data to the database in the Near-RT-RIC platform. |
| Step 1 (M) | xApp sends a request to Near-RT RIC platform to store related information and try to insert data  to database. |
| Step 2 (M) | Near-RT RIC platform stores xApp requested information in database. |
| Step 3 (M) | Near-RT RIC platform responds to the xApp, including success and failure operation. |

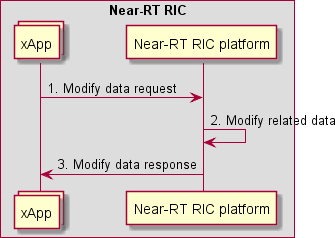


##### Figure 9.5.5-1: Store Data procedure

### Modify Data procedure

The Modify Data procedure in the Near-RT RIC allows the xApp to request Near-RT RIC platform to update or delete data from database.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | xApp intends to modify or delete data stored by the SDL. |
| Actors and Roles | * xApp: Originator of Modify Data request; * Near-RT RIC platform: Response to the request of xApp. |
| Pre-conditions | * SDL API services have been registered by Near-RT RIC platform; * xApp has successfully registered as a client to the SDL before; * xApp has been authorized to write the data in the database. |
| Begins when | xApp determines to modify relevant data stored by the database in the Near-RT-RIC platform. |
| Step 1 (M) | xApp sends a request to Near-RT RIC platform to modify related information, including update  and delete data in database. |
| Step 2 (M) | Near-RT RIC platform modifies or deletes xApp requested information in database. |
| Step 3 (M) | Near-RT RIC platform responds to the xApp, including success and failure operation. |



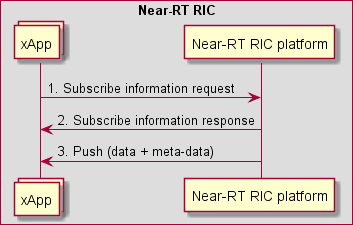
##### Figure 9.5.6-1: Modify Data procedure

### Subscribe/Push procedure

The Subscribe/Push procedure in the Near-RT RIC allows the xApp to subscribe to the Near-RT RIC platform for authorized data in the database.

The pushes may be used to contain a given set of information, or only the changed part of the information after the subscription.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | xApp subscribes with the Near-RT RIC platform for data changes and receives related push  messages. |
| Actors and Roles | * xApp: Originates the Subscribe request; * Near-RT RIC platform: Responds to the xApp request and pushes the subscribed data and meta-data to the xApp. |
| Pre-conditions | * SDL API services have been registered by Near-RT RIC platform; * xApp has successfully registered as a client to the SDL. |
| Begins when | xApp subscribes for push messages related to information in the database in the Near-RT RIC  platform. |
| Step 1 (M) | xApp sends the request to Near-RT RIC platform to subscribe for push messages related to  information. |
| Step 2 (M) | Near-RT RIC platform responds to the xApp subscription. |
| Step 3 (M) | Near-RT RIC platform sends the push message with data and meta-data for the subscribed  information to the xApp. |

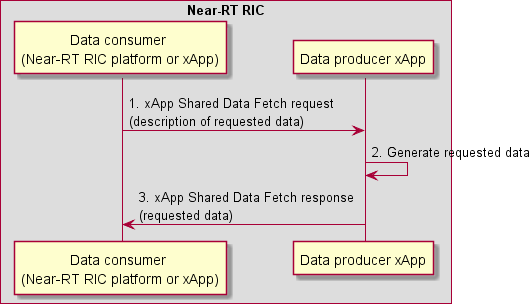


##### Figure 9.5.7-1: Subscribe/Push procedure

### 9.5.7A xApp Shared Data Fetch procedure

The xApp Shared Data Fetch procedure enables the data consumer (Near-RT RIC platform or an xApp) to fetch specific shared data from the data producer xApp.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | Data consumer (Near-RT RIC platform or xApp) fetches shared data from the data producer  xApp. |
| Actors and Roles | * Data producer xApp; * Data consumer: Originator of xApp Shared Data Fetch request, which may be the Near-RT RIC platform or an xApp. |
| Pre-conditions | * Data producer xApp has successfully registered its xApp Data Sharing service with Near-RT RIC platform; * Data consumer has discovered the xApp Data Sharing service. |
| Begins when | The data consumer determines to fetch relevant data from the data producer xApp. |
| Step 1 (M) | The data consumer sends a request to the data producer xApp for specific shared data. |
| Step 2 (M) | The data producer xApp generates the requested data. |
| Step 3 (M) | The data producer xApp responds with the requested data. |



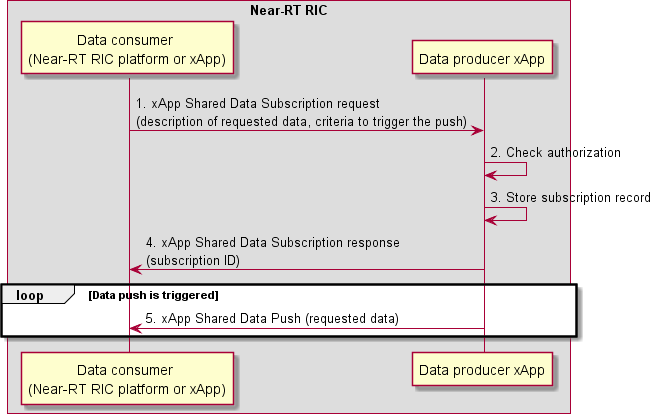
##### Figure 9.5.7A-1: xApp Shared Data Fetch procedure

### B Procedures related to xApp Shared Data Subscription and Notification

#### 9.5.7B.1 Procedures for xApp Shared Data Subscribe/Push

The xApp Shared Data Subscribe/Push procedures allow the data consumer (Near-RT RIC platform or an xApp) to subscribe to and receive specific shared data from the data producer xApp.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | Data consumer (Near-RT RIC platform or xApp) subscribes to and receives shared data from  the data producer xApp. |
| Actors and Roles | * Data producer xApp; * Data consumer: Originator of the xApp Shared Data Subscription request, which may be the Near-RT RIC platform or an xApp. |
| Pre-conditions | * Data producer xApp has successfully registered its xApp Data Sharing service with Near-RT RIC platform; * Data consumer has discovered the xApp Data Sharing service. |
| Begins when | The data consumer determines to subscribe to specific shared data from the data producer  xApp. |
| Step 1 (M) | The data consumer sends a subscription request to the data producer xApp for shared data. The request contains the description for the requested data, and the criteria to trigger the data  push. |
| Step 2 (M) | The data producer xApp checks whether the data consumer is authorized to access the  subscribed data. |
| Step 3 (M) | The data producer xApp accepts the request and stores the subscription record. |
| Step 4 (M) | The data producer xApp responds to the data consumer with the subscription ID. |
|  | [LOOP] Step 5 is executed when the criteria to trigger the data push are met. |
| Step 5 (M) | The data producer xApp pushes the subscribed data to the data consumer. |

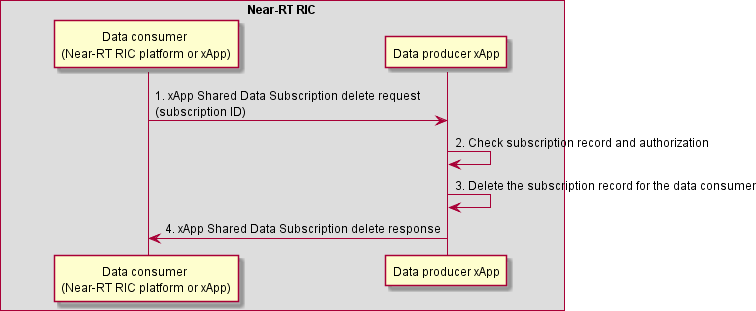


##### Figure 9.5.7B.1-1: xApp Shared Data Subscribe/Push procedures

#### 9.5.7B.2 xApp Shared Data Subscription Delete procedure

The xApp Shared Data Subscription Delete procedure allows the data consumer (Near-RT RIC platform or an xApp) to unsubscribe to specific shared data from the data producer xApp.

|  |  |  |
| --- | --- | --- |
| **Use Case Stage** | **Evolution / Specification** | **<<Uses>>**  **Related use** |
| Goal | Data consumer (Near-RT RIC platform or xApp) unsubscribes to shared  data from the data producer xApp. |  |
| Actors and Roles | * Data producer xApp; * Data consumer: Originator of the xApp Shared Data Subscription Delete request, which may be the Near-RT RIC platform or an xApp. |  |
| Pre-conditions | The data consumer has subscribed to shared data from the data producer  xApp. |  |
| Begins when | The data consumer determines to unsubscribe to specific shared data from  the data producer xApp. |  |
| Step 1 (M) | The data consumer sends a subscription delete request to the data  producer xApp. The request contains the subscription ID. |  |
| Step 2 (M) | The data producer xApp checks the subscription record associated with the  subscription ID and checks whether the data consumer is authorized to unsubscribe. |  |
| Step 3 (M) | The data producer xApp deletes the subscription record for the data  consumer. |  |
| Step 4 (M) | The data producer xApp sends a subscription delete response. |  |



##### Figure 9.5.7B.2-1: xApp Shared Data Subscription Delete procedure

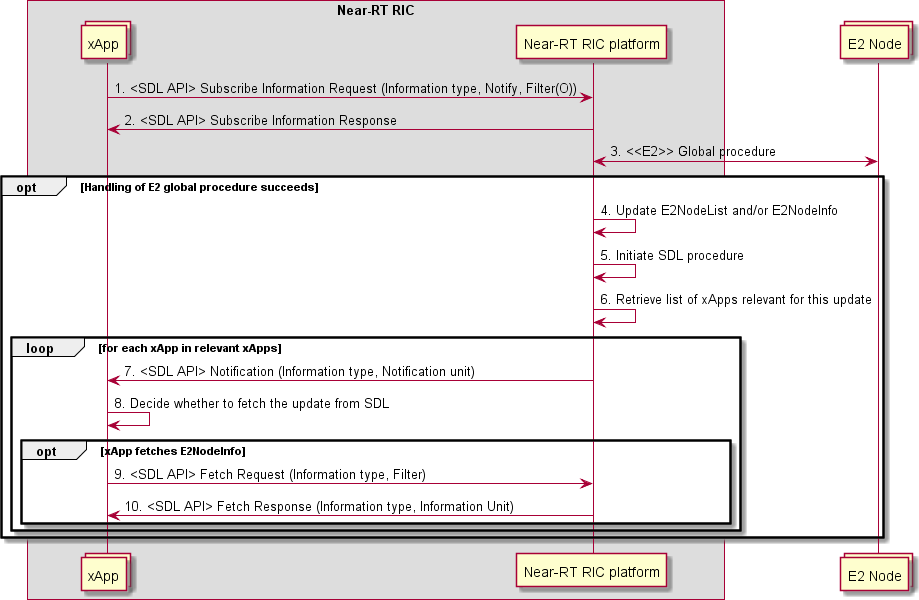
### Use cases of SDL APIs

#### Subscribe/Update-Notify for E2NodeInfo and E2NodeList

The SDL Subscribe/Update-Notify procedure (clause 9.5.4) along with the SDL fetch procedure (clause 9.5.3) may be used for getting notifications and fetching E2 Node related information such as E2NodeList (clause 9.5.9.2) and E2NodeInfo (clause 9.5.9.3) changes (obtained as the result of an E2AP Global Procedure such as: E2 Setup, Reset, E2 Removal, RIC Service Update or E2 Node Configuration Update).

As noted in clause 9.5.4, notifications will contain only the summary of changes to E2NodeInfo after the subscription. To get the complete state of E2NodeInfo, the xApp may do a fetch.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | Inform the xApp about changes to E2 Node information |
| Actors and Roles | * E2 Nodes: O-CU, O-DUs supporting the E2 interface. * xApps: Requires information about changes to E2 Node information. * Near-RT RIC platform:   + exposing an SDL API to the xApps;   + Termination of E2 connections. |
| Pre-conditions | * SDL API services have been registered and authorized by Near-RT RIC platform; * xApp has successfully registered as a client to the SDL before; * xApp has been authorized to read the data in the database. |
| Begins when | xApp decides it is interested in E2NodeInfo updates. |
| Step 1 (M) | xApp sends request to Near-RT RIC platform to subscribe for notifications related to E2NodeInfo  changes. |
| Step 2 (M) | Near-RT RIC platform responds to xApp subscription. |
| Step 3 (M) | E2 Node or Near-RT RIC initiates an E2AP Global procedure (i.e., E2 Setup, Reset, E2  Removal, RIC Service Update or E2 Node Configuration Update). |
|  | [OPT] Steps 4-10 are executed if the E2AP Global Procedure succeeds: |
| Step 4 (M) | Near-RT RIC platform updates the Database. |
| Step 5 (M) | Near-RT RIC platform triggers SDL procedure. |
| Step 6 (M) | Near-RT RIC platform retrieves list of xApps that would be interested in this update . |
|  | [LOOP] Steps 7-10 are looped for each relevant xApp. |
| Step 7 (M) | Near-RT RIC platform sends a notification of subscribed E2NodeList update or  E2NodeInfo update to the xApp which includes the E2NodeID. |
| Step 8 (M) | xApp makes decision on whether to fetch the updated information from the Near-RT RIC  platform. |
|  | [OPT] Steps 9-10 are executed if the xApp decides that it needs the update: |
| Step 9 (M) | xApp sends a fetch request to the Near-RT RIC platform which includes the E2NodeID. |
| Step 10 (M) | Near-RT RIC platform responds with a fetch response (E2NodeList or E2NodeInfo). |



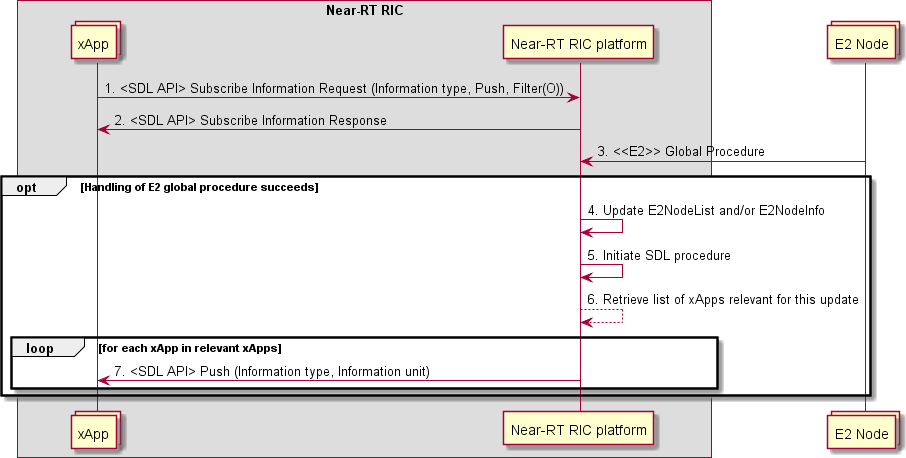
##### Figure 9.5.8.1-1: Subscribe/Notify procedures for E2NodeInfo and E2NodeList

#### Subscribe/Push for E2NodeInfo and E2NodeList

The SDL Subscribe/Push procedure (clause 9.5.7) may be used for getting E2 Node related information pushes such as E2NodeList (clause 9.5.9.2) and E2NodeInfo (clause 9.5.9.3) (obtained as the result of an E2 Global procedure such as E2 Setup, Reset, E2 Removal, RIC Service Update or E2 Node Configuration Update).

As noted in clause 9.5.7, the pushes may contain only the changed to E2 Node related information after the subscription, or a complete set of E2 Node related information.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution/Specification** |
| Goal | Inform the xApp about changes to E2 Node information. |
| Actors and Roles | * E2 Nodes: O-CU, O-DUs supporting the E2 interface; * xApps: Requires information about changes to E2 Node information; * Near-RT RIC platform:   + exposing an SDL API to the xApps;   + Termination of E2 node connections. |
| Pre-conditions | * SDL API services have been registered and authorized by Near-RT RIC platform; * xApp has successfully registered as a client to the SDL before; * xApp has been authorized to read the data in the database. |
| Begins when | xApp decides it is interested in E2NodeInfo updates. |
| Step 1 (M) | xApp sends request to Near-RT RIC platform to subscribe for pushes related to E2NodeList  and/or E2NodeInfo. |
| Step 2 (M) | Near-RT RIC platform responds to xApp subscription. |
| Step 3 (M) | E2 Node or Near-RT RIC initiates an E2AP Global Procedure (i.e., E2 Setup, Reset, E2  Removal, RIC Service Update or E2 Node Configuration Update). |
|  | [OPT] Steps 4-7 are executed if the E2AP Global procedure succeeds: |
| Step 4 (M) | Near-RT RIC platform updates the Database. |
| Step 5 (M) | Near-RT RIC platform triggers SDL procedure. |
| Step 6 (M) | Near-RT RIC platform retrieves list of xApps that would be interested in this update. |
| Step 7 (M) | For each relevant xApp, Near-RT RIC platform sends a push of E2NodeList or E2NodeInfo to  the xApp which includes the E2NodeID. |



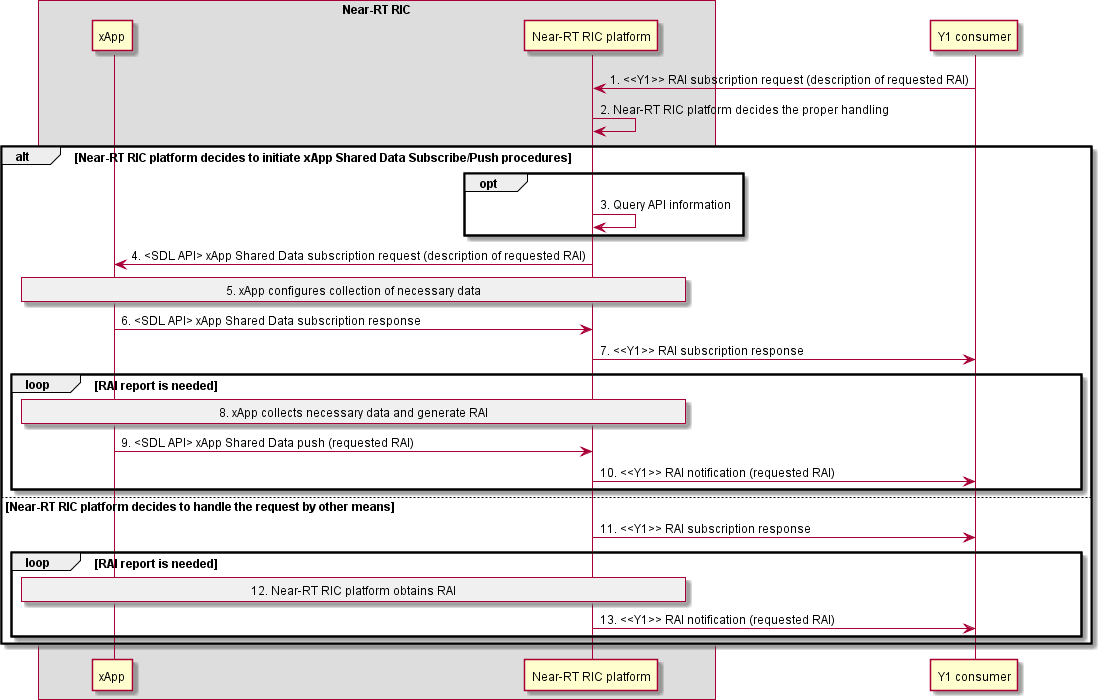
##### Figure 9.5.8.2-1: Subscribe/Push procedures for E2NodeInfo and E2NodeList

#### xApp data sharing for RAN analytics information exposure

An xApp may act as a producer of some types of RAI. Such RAI may be exposed to the Near-RT RIC platform, and further exposed via Y1 interface. In this use case, the Near-RT RIC platform uses the xApp Shared Data Subscribe/Push procedures or the xApp Shared Data Fetch procedure to collect RAI from the RAI producer xApp.

1. using xApp Shared Data Subscribe/Push procedures

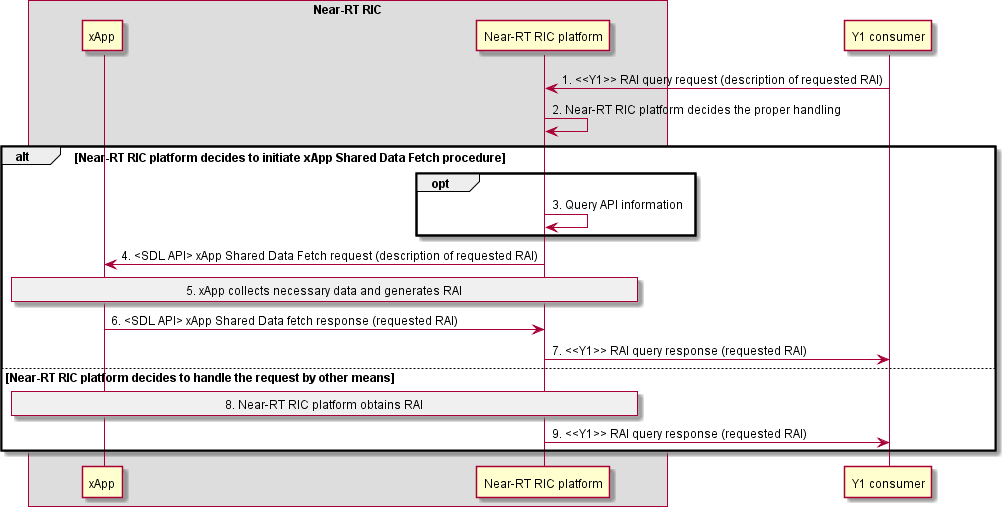
|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | Near-RT RIC platform exposes via Y1 interface the RAI produced by the xApp. |
| Actors and Roles | * Y1 consumer * xApp: Producer of RAI. * Near-RT RIC platform, with the following functionalities:   + Y1 termination;   + API enablement. |
| Pre-conditions | * RAI producer xApp has registered the API information of its xApp Data Sharing API information using the Enablement APIs; * Y1 consumer has been provided with the supported RAI types. |
| Begins when | Y1 consumer determines to subscribe to specific RAI from the Near-RT RIC. |
| Step 1 (M) | Y1 consumer requests RAI with a Y1 RAI subscription request to Near-RT RIC platform. |
| Step 2 (M) | Near-RT RIC platform decides how to handle the request. |
|  | [ALT] Steps 3-10 are executed if Near-RT RIC platform decides to initiate xApp Shared Data  Subscribe/Push procedures with the RAI producer xApp. |
| Step 3 (O) | Near-RT RIC platform queries the API information registered by the RAI producer xApp. |
| Step 4 (M) | Near-RT RIC platform forwards the RAI request to the RAI producer xApp with an xApp  Shared Data subscription request. |
| Step 5 (M) | RAI producer xApp configures collection of necessary data (e.g., by creating E2  subscriptions and/or A1-EI subscription). |
| Step 6 (M) | RAI producer xApp responds to Near-RT RIC platform with an xApp Shared Data  subscription response. |
| Step 7 (M) | Near-RT RIC platform responds to Y1 consumer with a Y1 RAI subscription response, which  contains the result in Step 5. |
|  | [LOOP] Steps 8-10 are executed when RAI report is needed. |
| Step 8 (M) | RAI producer xApp collects necessary data and generates RAI. |
| Step 9 (M) | RAI producer xApp sends RAI to the Near-RT RIC platform with an xApp Shared Data  Push. |
| Step 10 (M) | Near-RT RIC platform forwards the RAI to Y1 consumer. |
|  | [ELSE] Steps 11-13 are executed if Near-RT RIC platform decides to handle the request by  other means (e.g., reusing an existing xApp Shared Data subscription, or using xApp Shared Data Fetch). |
| Step 11 (M) | Near-RT RIC platform sends the response to Y1 consumer with a Y1 RAI subscription  response. |
|  | [LOOP] Steps 12-13 are executed when RAI report is needed. |
| Step 12 (M) | Near-RT RIC platform obtains RAI. |
| Step 13 (M) | Near-RT RIC platform forwards the RAI to Y1 consumer. |



##### Figure 9.5.8.3-1: xApp Shared Data Subscribe/Push procedures for RAN analytics information exposure

1. using xApp Shared Data Fetch procedure

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | Near-RT RIC platform exposes via Y1 interface the RAI produced by the xApp. |
| Actors and Roles | * Y1 consumer * xApp: Producer of RAI. * Near-RT RIC platform, with the following functionalities:   + Y1 termination;   + API enablement. |
| Pre-conditions | * RAI producer xApp has registered the API information of its xApp Data Sharing API information using the Enablement APIs; * Y1 consumer has been provided with the supported RAI types. |
| Begins when | Y1 consumer determines to query specific RAI from the Near-RT RIC. |
| Step 1 (M) | Y1 consumer requests RAI with a Y1 RAI query request to Near-RT RIC platform. |
| Step 2 (M) | Near-RT RIC platform decides how to handle the request. |
|  | [ALT] Steps 3-7 are executed if Near-RT RIC platform decides to initiate xApp Shared Data  Fetch procedure with the RAI producer xApp. |
| Step 3 (O) | Near-RT RIC platform queries the API information registered by the RAI producer xApp. |
| Step 4 (M) | Near-RT RIC platform forwards the RAI request to the RAI producer xApp with an xApp  Shared Data Fetch request. |
| Step 5 (M) | RAI producer xApp collects necessary data and generates RAI. |
| Step 6 (M) | RAI producer xApp sends the RAI to Near-RT RIC platform with an xApp Shared Data Fetch  response. |
| Step 7 (M) | Near-RT RIC platform forwards the RAI to Y1 consumer with a Y1 RAI query response. |
|  | [ELSE] Steps 8-9 are executed if Near-RT RIC platform decides to handle the request by other  means (e.g., using/reusing an xApp Shared Data subscription). |
| Step 8 (M) | Near-RT RIC platform obtains RAI. |
| Step 9 (M) | Near-RT RIC platform forwards the RAI to Y1 consumer with a Y1 RAI query response. |



##### Figure 9.5.8.3-2: xApp Shared Data Fetch procedure for RAN analytics information exposure

### SDL API data structures

#### Overview

The following data structures are defined:

* + - * + E2NodeList: List of known E2 Node ID and the associated E2 Node State
        + E2NodeInfo: E2 Node component configuration list and RAN function list

E2NodeList and E2NodeInfo are used to inform xApps of change to E2 Node state and information as the result of E2AP Global Procedures. The relationship between these data structures and E2AP Global Procedures is described in table 9.5.9.1-1.

Table 9.5.9.1-2 presents the actions that xApp may perform under different E2 Node State conditions.

NOTE 1: E2AP Error Indication procedure does not result in a change to SDL API data as most error cases are assumed to be handled internally within the Near-RT RIC platform. An appropriate E2 Related API (Failure, Cause) message may be used to inform xApp if required.

NOTE 2: E2AP Reset procedure handling is assumed to be assisted by an appropriate platform mechanism with the Near- RT RIC first using the appropriate SDL API mechanism (see clause 9.5.8) to inform xApp that a RESET has occurred (i.e., E2 Node State = "Reset") and then a second SDL API mechanism is used to inform xApp that the E2 Node is again available (i.e. E2 Note State = "Connected"). It is assumed that the Near-RT RIC platform will delete all associated subscription records during the period between these two SDL API mechanisms and that xApp will use the information E2 Node is again available (i.e., E2 Note State = "Connected") to trigger any required E2 Subscription (Request) messages to re-establish E2 subscriptions.

NOTE 3: E2AP E2 Removal procedure handling and E2 link failure handling are assumed to be assisted by an appropriate platform mechanism with the Near-RT RIC using the appropriate SDL API mechanism (see clause 9.5.8) to inform xApp that the E2 Node is not available (i.e., E2 Node State = "Disconnect").

##### Table 9.5.9.1-1: SDL API handling of E2AP Global procedures

|  |  |  |
| --- | --- | --- |
| **Global**  **procedure** | **E2NodeList change** | **E2NodeInfo change** |
| E2 Setup | **Add E2NodeList record** (E2 Node ID, E2 Node State =  "Connected") | **Add E2NodeInfo record** (E2 Node ID, RAN Function list, E2 Node component configuration  list) |
| Reset | **Change E2NodeList record** (E2 Node State = "Reset").  After using an appropriate Near- RT RIC mechanism, **Change E2NodeList record** (E2 Node  State to "Connected") | - |
| RIC Service  Update | - | **Add, Modify and/or Remove E2NodeInfo**  **record** (E2 Node ID, RAN Function List) |
| E2 Node  Configuration Update | - | **Add, Modify and/or Remove E2NodeInfo**  **record** (E2 Node ID, E2 Node Component configuration list) |
| E2 Removal | **Change E2NodeList record** (E2 Node State = "Disconnected").  After **Delete E2NodeInfo record**, **Delete E2NodeList record** (E2  Node ID) | **Delete E2NodeInfo record** (E2 Node ID) |
| E2 link failure | **Change E2NodeList record** (E2 Node State = "Disconnected").  After **Delete E2NodeInfo record**, **Delete E2NodeList record** (E2  Node ID) | **Delete E2NodeInfo record** (E2 Node ID) |

##### Table 9.5.9.1-2: Expected xApp responses to E2 Node State

|  |  |
| --- | --- |
| **E2 Node State** | **xApp actions** |
| Connected | Discover E2 Node ID in E2NodeList and obtain E2NodeInfo (after E2 Setup).  Initiate E2 Related APIs (after E2 Setup and Reset). |
| Reset | Delete internal records of ongoing E2 Related API procedures.  Monitor for changes in E2 Node State. |
| Disconnected | Delete internal records of ongoing E2 Related API procedures.  Delete internal records of E2 Node ID. |

#### E2NodeList

The E2NodeList data structure contains a list of known E2 Nodes and the associated E2 Node State. Each record contains:

* + - * + E2 Node ID
        + E2 Node State ("Connected", "Reset", "Disconnected")

E2NodeList records may be deleted when the corresponding E2NodeInfo record has been deleted. It is recommended to maintain this record for sufficient time for all xApps to complete clean-up mechanisms.

#### E2NodeInfo

The E2NodeInfo data structure contains information on a given E2 Node. Each record contains:

* + - * + E2 Node ID
        + List of declared E2 Node component configuration information
        + List of declared RAN Functions

E2NodeInfo records shall be deleted after corresponding E2NodeState is set to "Disconnected" and sufficient time has been allowed for all xApps to complete clean-up mechanisms.

## API Enablement Procedures

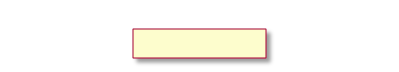
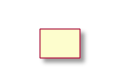
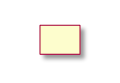
### API Discovery procedure

This procedure enables the xApp(s) to discover the available Near-RT RIC APIs.

|  |  |  |
| --- | --- | --- |
| **Use Case Stage** | **Evolution / Specification** | **<<Uses>>**  **Related use** |
| Goal | To allow the xApp to discover which APIs are available and how to access  them. |  |
| Actors and Roles | * xApp: Requester of the available Near-RT RIC APIs. * Near-RT RIC platform, with the following functionalities:   + API Enablement;   + Database |  |
| Assumptions | For initiating the discovery, the xApp has registered to Near-RT RIC platform and is aware of the API information of the Enablement APIs. |  |
| Pre conditions | * The xApp is deployed to Near-RT RIC platform. * The API information of Enablement APIs has been provided to the xApp (as pre-configuration by SMO or by other means). * One or more Near-RT RIC platform services are configured to   provide their API information with the API Enablement functionality. |  |
| Begins when | xApps wants to discover the available Near-RT RIC APIs. |  |
| Step 1 (M) | xApp sends a API discovery request to the Near-RT RIC platform. It includes the xApp identity and includes query information. Query information may include criteria for discovering matching APIs. |  |
| Step 2 (M) | Near-RT RIC platform verifies the identity of the xApp. |  |
| Step 3 (M) | Near-RT RIC platform searches the stored Near-RT RIC API(s) information,  as per the query information in the discovery request. |  |
| Step 4 (M) | Near-RT RIC platform applies the discovery policy and performs filtering of  the retrieved Near-RT RIC APIs information. |  |
| Step 5 (M) | The Near-RT RIC platform sends a API discovery response to the xApp with the API information about the APIs for which the xApp has the required  authorization. |  |

xApp

Near RT RIC platform



Near RT RIC platform ser ice a ailable

tore A I information

1. A I disco ery request

. Near RT RIC platform erifies the identity of the xApp

1. earch stored A I information
2. Apply disco ery policy and information filtering
3. A I disco ery response

xApp Near RT RIC platform

##### Figure 9.6.1-1: API Discovery procedure

### Procedures related to API Event Subscription and Notification

In this clause, the procedures for subscription / subscription deletion and API event notification are provided. These procedures allow the xApps to receive information on API related events.

Such events correspond to Near-RT RIC API related events like the following:

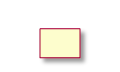
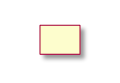
|  |  |
| --- | --- |
| **Events** | **Events Description** |
| Availability of Near-RT RIC APIs | Availability events of Near-RT RIC APIs (e.g.,  active, inactive) |
| Near-RT RIC API updated | Events related to change in Near-RT RIC API  information |

#### Procedure for API Event Subscription

The purpose of this procedure is to enable xApps to subscribe to monitor events related to the provided Near-RT RIC APIs.

|  |  |  |
| --- | --- | --- |
| **Use Case Stage** | **Evolution / Specification** | **<<Uses>>**  **Related use** |
| Goal | To allow the xApp to subscribe to receive Near-RT RIC API related event  notifications. |  |
| Actors and Roles | * xApp: Requester to subscribe to notifications of events related to the Near-RT RIC APIs; * Near-RT RIC platform, with the following functionalities:   + API enablement.   + Database |  |
| Assumptions | * For initiating the subscription, the xApp has registered to RIC platform and is aware of the API information of Enablement APIs. Also, xApp has discovered the provided Near-RT RIC APIs. |  |
| Pre-conditions | * The xApp is registered to Near-RT RIC platform; * The API information of Enablement APIs has been provided to the xApp (as pre-configuration by SMO); * The xApp has discovered the registered Near-RT RIC APIs (see 9.6.1). |  |
| Begins when | xApp has discovered the available Near-RT RIC APIs and wants to subscribe for event notifications. |  |
| Step 1 (M) | xApp sends an API event subscription request to the Near-RT RIC platform in order to receive notification of events. This request includes the subscribing xApp identifier as well as the event criteria:   * Event criteria may include event type information, e.g., API failure event, new API available event, API unavailable event, API version change event, API location change event, etc. |  |
| Step 2 (M) | The Near-RT RIC platform checks whether the xApp is authorized to  monitor the requested event(s). |  |
| Step 3 (M) | The Near-RT RIC platform stores the approved subscription information. |  |
| Step 4 (M) | The Near-RT RIC platform sends an API event subscription response to the xApp. This response indicates the success or failure of the event  subscription operation. |  |

subscription



xApp

Near RT RIC platform

1. A I e ent subscription request

. Check authori ation for A I e ent

3. tore e ent subscription

4. A I e ent subscription response

(success or failure)

xApp

Near RT RIC platform

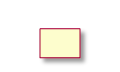
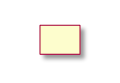
##### Figure 9.6.2.1-1: Subscription for API-related events

#### Procedure for API Event Subscription Delete

The purpose of this procedure is to enable xApps to un-subscribe from events monitoring related to the provided Near-RT RIC APIs.

|  |  |  |
| --- | --- | --- |
| **Use Case Stage** | **Evolution / Specification** | **<<Uses>>**  **Related use** |
| Goal | To allow the xApp to un-subscribe from Near-RT RIC API related event  notifications. |  |
| Actors and Roles | * xApp: Requester to un-subscribe from monitoring events related to the Near-RT RIC APIs; * Near-RT RIC platform, with the following functionalities:   + API Enablement.   + Database |  |
| Assumptions | For initiating the subscription delete, the xApp has registered to Near-RT  RIC platform and is aware of the API information of the Enablement APIs. Also, xApp has subscribed to events. |  |
| Pre-conditions | The xApp has subscribed to Near-RT RIC API related events (see 9.6.2.1). |  |
| Begins when | xApp wants to un-subscribe for Near-RT RIC API related events. |  |
| Step 1 (M) | xApp sends an API event subscription delete request to the Near-RT RIC  platform with the information of the subscribed event and includes the xApp identifier and the event subscription identifier. |  |
| Step 2 (M) | The Near-RT RIC platform, upon receiving the API event subscription delete request from the xApp, checks for the API event subscription corresponding to the xApp and further checks if the subscribing entity is  authorized to un-subscribe. |  |
| Step 3 (M) | Near-RT RIC platform deletes the subscription information for the xApp. |  |
| Step 4 (M) | The API Enablement sends an API event subscription delete response to  the xApp. This response indicates the success or failure of the event subscription delete operation. |  |

xApp Near RT RIC platform



1. A I e ent subscription delete request

. Check authori ation for A I e ent subscription

3. elete the record of the A I e ent subscription

4. A I e ent subscription delete response

(success or failure)

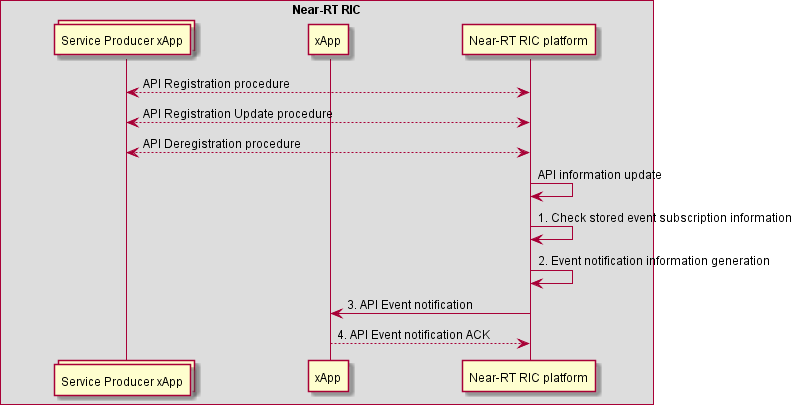
xApp Near RT RIC platform

##### Figure 9.6.2.2-1: Subscription Delete for API-related events

#### Procedure for API Event Notification

This procedure shows sending the event notification to the subscribed xApp, based on a trigger event captured at the Near-RT RIC platform.

|  |  |  |
| --- | --- | --- |
| **Use Case Stage** | **Evolution / Specification** | **<<Uses>>**  **Related use** |
| Goal | To notify the xApp on events related to the Near-RT RIC APIs, based on  subscription. |  |
| Actors and Roles | * xApp: Receiving event notifications based on subscription; * Service producer xApp: Initiating API Registration, API Registration update or API Deregistration procedure * Near-RT RIC platform, with the following functionalities:   + API Enablement   + Database |  |
| Assumptions | For receiving a notification, the xApp has registered to Near-RT RIC platform and has subscribed to events. |  |
| Pre-conditions | The xApp has subscribed to Near-RT RIC API related events (see 9.6.2.1). |  |
| Begins when | A trigger event criterion is met at the Near-RT RIC platform (i.e., API information update from a Near-RT RIC platform service and/or Service producer xApp). Such criterion has been configured during subscription of  the xApp. |  |
| Step 1 (M) | Near-RT RIC platform checks corresponding event subscriptions |  |
| Step 2 (M) | Near-RT RIC platform generates an event related to APIs, which is to be  consumed by the subscribed xApp(s) |  |
| Step 3 (M) | The Near-RT RIC platform sends event notifications to all the subscribing  xApp(s) that have subscribed for the event matching the criteria. |  |
| Step 4 (O) | The xApp may send an event notification acknowledgement to Near-RT RIC platform. It depends on the capabilities of the transport used for  notifications whether or not such acknowledgement can be sent. |  |



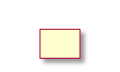
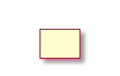
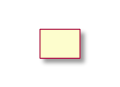
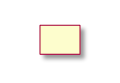
##### Figure 9.6.2.3-1: API Event Notification

### Procedures related to API Registration

#### API Registration procedure

This procedure enables the xApp to register as a service producer.

|  |  |  |
| --- | --- | --- |
| **Use Case Stage** | **Evolution / Specification** | **<<Uses>>**  **Related use** |
| Goal | To allow the xApp to register as a Near-RT RIC API service producer and  provide information on how to access the service. |  |
| Actors and Roles | * Service Producer xApp: Producer of a Near-RT RIC API. * xApp: Receiving event notifications based on subscription; * Near-RT RIC platform, with the following functionalities:   + API Enablement;   + Database |  |
| Assumptions | For initiating the API Registration procedure, the xApp has registered as an  xApp to Near-RT RIC platform and is aware of the API information of the Enablement APIs. |  |
| Pre conditions | * The xApp is deployed to Near-RT RIC platform. * The API information of Enablement APIs has been provided to the xApp (as pre-configuration by SMO or by other means). * The xApp is configured and authorized to consume API Registration service |  |
| Begins when | xApps wants to register as a producer of a service in the Near RT RIC. |  |
| Step 1 (M) | xApp sends an API Registration request to the Near-RT RIC platform. It  includes the xApp identity and supported API information. |  |
| Step 2 (M) | Near-RT RIC platform verifies the identity of the xApp. |  |
| Step 3 (M) | Near-RT RIC platform verifies that the xApp is authorized to act as a  Service Producer. |  |
| Step 4 (M) | Near-RT RIC platform adds stored API information. |  |
| Step 5 (M) | The Near-RT RIC platform sends API Registration response to the xApp. This response indicates the success or failure of the API Registration  operation. |  |
| Ends with | Near-RT RIC platform may initiate API Event notification procedure towards  one or more other xApps that had previously established a corresponding API Event Subscription. |  |



xApp

er ice roducer xApp

Near RT RIC platform

Near RT RIC platform ser ice a ailable

1. A I Registration request

. Near RT RIC platform erifies the identity of the xApp

1. Near RT RIC platform erifies

that xApp may act as er ice roducer

1. tore A I information

A I E ent Notification

xApp

er ice roducer xApp

Near RT RIC platform

5. A I Registration response ( uccess or ailure)

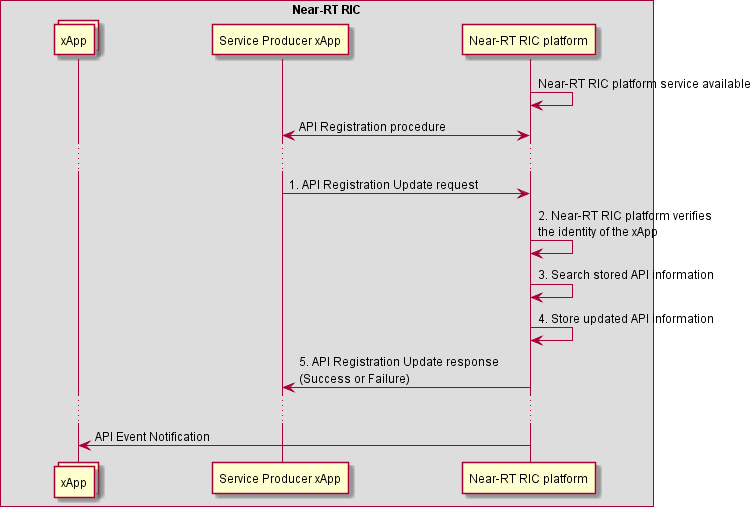
##### Figure 9.6.3.1-1: API Registration procedure

#### API Registration Update procedure

This procedure enables a previously registered xApp service producer to update API information.

|  |  |  |
| --- | --- | --- |
| **Use Case Stage** | **Evolution / Specification** | **<<Uses>>**  **Related use** |
| Goal | To allow the xApp to update provided service and/or information on how to  access the services it produces. |  |
| Actors and Roles | * Service Producer xApp: Producer of a Near-RT RIC API. * xApp: Receiving event notifications based on subscription; * Near-RT RIC platform, with the following functionalities:   + API Enablement;   + Database |  |
| Assumptions | For initiating the API Registration Update procedure, the xApp has registered to Near-RT RIC platform and is aware of the API information of  the Enablement APIs. |  |
| Pre conditions | * The xApp is deployed to Near-RT RIC platform. * The API information of Enablement APIs has been provided to the xApp (as pre-configuration by SMO or by other means). * The xApp has previously completed API Registration procedure |  |
| Begins when | xApps wants to update information concerning a Near-RT RIC API. |  |
| Step 1 (M) | xApp sends an API Registration update request to the Near-RT RIC platform. It includes the xApp identity and updated information on supported API. |  |
| Step 2 (M) | Near-RT RIC platform verifies the identity of the xApp. |  |
| Step 3 (M) | Near-RT RIC platform searches the stored Near-RT RIC API(s) information. |  |

|  |  |  |
| --- | --- | --- |
| Step 4 (M) | Near-RT RIC platform updates stored API information |  |
| Step 5 (M) | The Near-RT RIC platform sends an API Registration Update response to the xApp. This response indicates the success or failure of the API  Registration Update operation. |  |
| Ends with | Near-RT RIC platform may initiate API Event notification procedure towards  one or more other xApps that had previously established a corresponding API Event Subscription. |  |

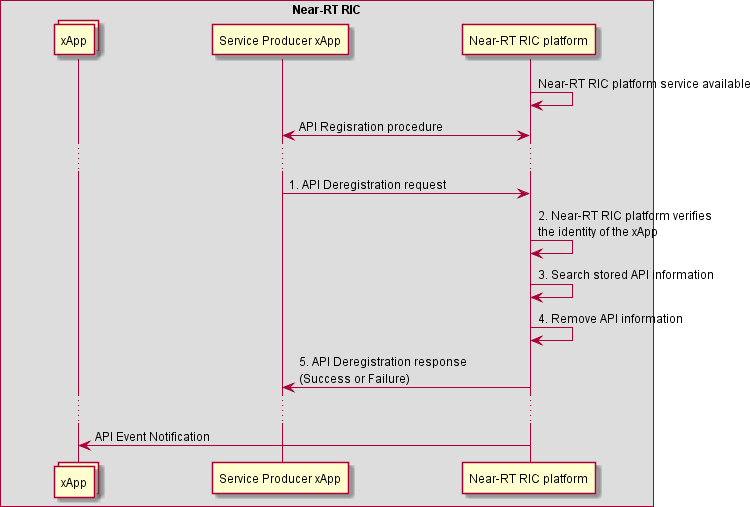


##### Figure 9.6.3.2-1: API Registration Update procedure

#### API Deregistration procedure

This procedure enables a previously registered xApp service producer to deregister as a service producer.

|  |  |  |
| --- | --- | --- |
| **Use Case Stage** | **Evolution / Specification** | **<<Uses>>**  **Related use** |
| Goal | To allow the xApp to deregister as a Near-RT RIC API service producer. |  |
| Actors and Roles | * Service Producer xApp: Producer of a Near-RT RIC API. * xApp: Receiving event notifications based on subscription * Near-RT RIC platform, with the following functionalities:   + API Enablement;   + Database |  |
| Assumptions | For initiating the API Registration Update procedure, the xApp has  registered to Near-RT RIC platform and is aware of the API information of the Enablement APIs. |  |
| Pre conditions | * The xApp is deployed to Near-RT RIC platform. * The API information of Enablement APIs has been provided to the xApp (as pre-configuration by SMO or by other means). * The xApp has previously completed API Registration procedure |  |
| Begins when | xApps wants to deregister as a service producer of a Near-RT RIC API. |  |
| Step 1 (M) | xApp sends an API Deregistration request to the Near-RT RIC platform. It includes the xApp identity and supported API information. |  |
| Step 2 (M) | Near-RT RIC platform verifies the identity of the xApp. |  |
| Step 3 (M) | Near-RT RIC platform searches the stored Near-RT RIC API(s) information. |  |
| Step 4 (M) | Near-RT RIC platform removes stored API information. |  |
| Step 5 (M) | The Near-RT RIC platform sends an API Deregistration response to the xApp. This response indicates the success or failure of the API  Deregistration operation. |  |
| Ends with | Near-RT RIC platform may initiate API Event notification procedure towards  one or more other xApps that had previously established a corresponding API Event Subscription. |  |



##### Figure 9.6.3.3-1: API Deregistration procedure

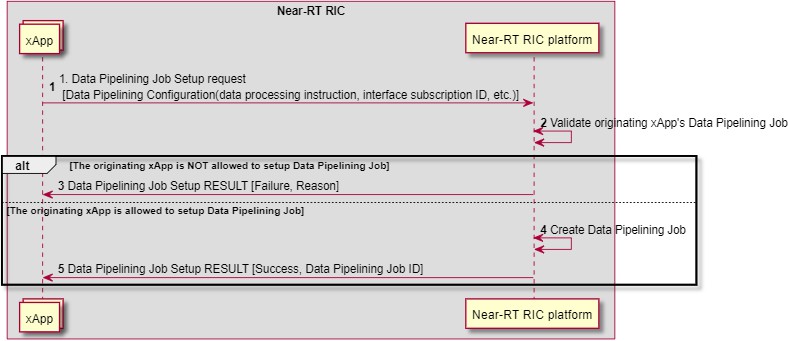
## AI/ML API Procedures

### Data Pipelining service procedures

#### Data Pipelining Job Setup procedure

The Data Pipelining Job Setup procedure enables an xApp to setup a new data pipelining job in the Near-RT RIC platform.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | xApp requests to setup a data pipelining job in the Near-RT RIC platform. |
| Actors and Roles | * xApp: Originator of data pipelining job setup request; * Near-RT RIC platform: Response to the request of xApp. |
| Pre-conditions | * AI/ML workflow services have been registered by Near-RT RIC platform. |
| Begins when | xApp determines to setup a Data Pipelining Job in the Near-RT RIC platform. |
| Step 1 (M) | xApp sends a request to Near-RT RIC platform to setup a data pipelining job. |
| Step 2 (M) | Near-RT RIC platform checks whether the originating xApp is allowed to setup the data  pipelining job. |
|  | [ALT] Step 3 is executed if the originating xApp is not allowed to setup the data pipelining job. |
| Step 3 (M) | A failure is responded to the originating xApp, possibly including appropriate reason. |
|  | [ELSE] Steps 4-5 are executed if the originating xApp is allowed to setup the data pipelining job. |
| Step 4 (M) | Near-RT RIC platform create corresponding data pipelining job. |
| Step 5 (M) | Near-RT RIC platform responds success to the originating xApp. |

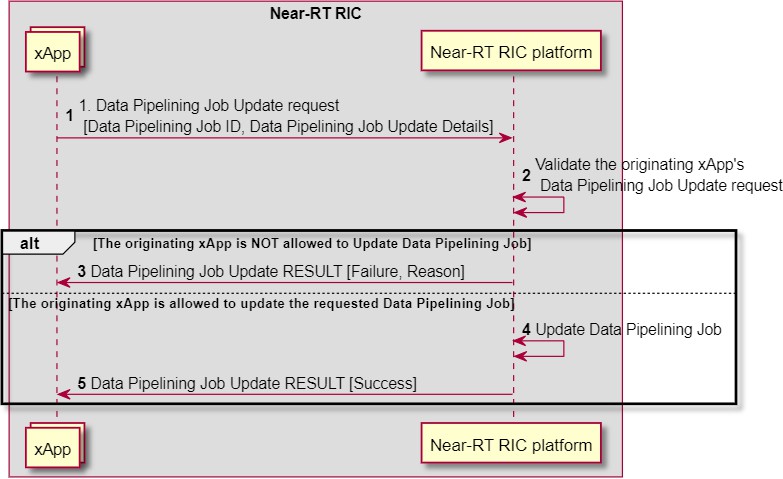


##### Figure 9.7.1.1-1: Data Pipelining Job Setup procedure

#### Data Pipelining Job Update procedure

The Data Pipelining Job Update procedure enables an xApp to update a running data pipelining job in the Near-RT RIC platform.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | xApp requests to update a data pipelining job in the Near-RT RIC platform. |
| Actors and Roles | * xApp: Originator of data pipelining job update request; * Near-RT RIC platform: Response to the request of xApp. |
| Pre-conditions | * AI/ML workflow services have been registered by Near-RT RIC platform; * At least one Data Pipelining Job setup by xApp is running in the Near-RT RIC platform. |
| Begins when | xApp determines to update a Data Pipelining Job in the Near-RT-RIC platform. |
| Step 1 (M) | xApp sends a request to Near-RT RIC platform to update a data pipelining job. |
| Step 2 (M) | Near-RT RIC platform checks whether the originating xApp is allowed to update the requested  data pipelining job. |
|  | [ALT] Step 3 is executed if the originating xApp is not allowed to update the requested data  pipelining job. |
| Step 3 (M) | A failure is responded to the originating xApp, possibly including appropriate reason. |
|  | [ELSE] Steps 4-5 are executed if the originating xApp is allowed to update the requested data  pipelining job. |
| Step 4 (M) | Near-RT RIC platform updates corresponding data pipelining job. |
| Step 5 (M) | Near-RT RIC platform responds success to the originating xApp. |

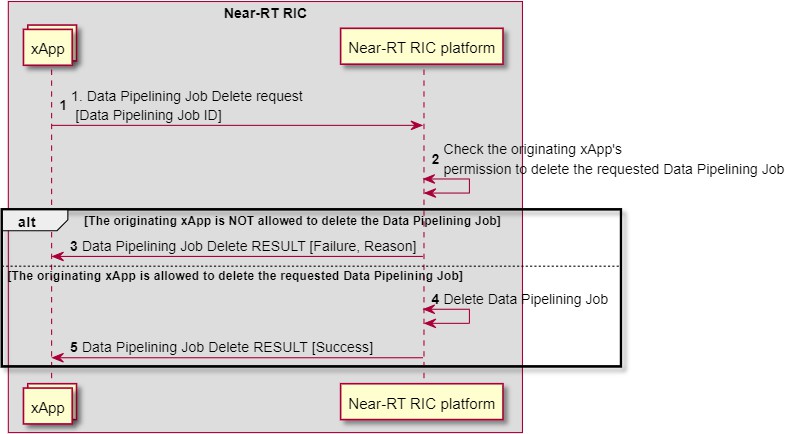


##### Figure 9.7.1.2-1: Data Pipelining Job Update procedure

#### Data Pipelining Job Delete procedure

The Data Pipelining Job Delete procedure enables an xApp to delete a running data pipelining job in the Near-RT RIC Platform.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | xApp requests to delete a data pipelining job in the Near-RT RIC platform. |
| Actors and Roles | * xApp: Originator of data pipelining job delete request; * Near-RT RIC platform: Response to the request of xApp. |
| Pre-conditions | * AI/ML workflow services have been registered by Near-RT RIC platform; * At lease one data pipelining job setup by xApp is running in the Near-RT RIC platform. |
| Begins when | xApp determines to delete a data pipelining job in the Near-RT-RIC platform. |
| Step 1 (M) | xApp sends a request to Near-RT RIC platform to delete a data pipelining job. |
| Step 2 (M) | Near-RT RIC platform checks whether the originating xApp is allowed to delete the requested  data pipelining job |
|  | [ALT] Step 3 is executed if the originating xApp is not allowed to delete the requested data  pipelining job. |
| Step 3 (M) | A failure is responded to the originating xApp, possibly including appropriate reason. |
|  | [ELSE] Steps 4-5 are executed if the originating xApp is allowed to delete the requested data  pipelining job. |
| Step 4 (M) | Near-RT RIC platform deletes corresponding data pipelining job. |
| Step 5 (M) | Near-RT RIC platform responds success to the originating xApp. |

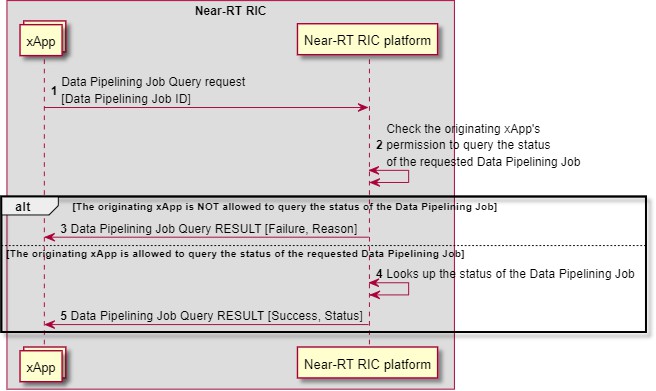


##### Figure 9.7.1.3-1: Data Pipelining Job Delete procedure

#### Data Pipelining Job Query procedure

The Data Pipelining Job Query procedure enables an xApp to query the status of a running data pipelining job in the Near-RT RIC platform.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | xApp requests to query the status of a data pipelining job in the Near-RT RIC platform. |
| Actors and Roles | * xApp: Originator of data pipelining job query request; * Near-RT RIC platform: Response to the request of xApp. |
| Pre-conditions | * AI/ML workflow services have been registered by Near-RT RIC platform; * At least one data pipelining job setup by xApp is running in the Near-RT RIC platform. |
| Begins when | xApp determines to query the status of a data pipelining job in the Near-RT-RIC platform. |
| Step 1 (M) | xApp sends a request to Near-RT RIC platform to query the status of a data pipelining job. |
| Step 2 (M) | Near-RT RIC platform checks whether the originating xApp is allowed to query the status of the  requested data pipelining job. |
|  | [ALT] Step 3 is executed if the originating xApp is not allowed to query the status of the  requested data pipelining job. |
| Step 3 (M) | A failure is responded to the originating xApp, possibly including appropriate reason. |
|  | [ELSE] Steps 4-5 are executed if the originating xApp is allowed to query the status of the  requested data pipelining job. |
| Step 4 (M) | Near-RT RIC platform looks up the status of corresponding data pipelining job. |
| Step 5 (M) | Near-RT RIC platform responds success to the originating xApp, including the queried  status. |

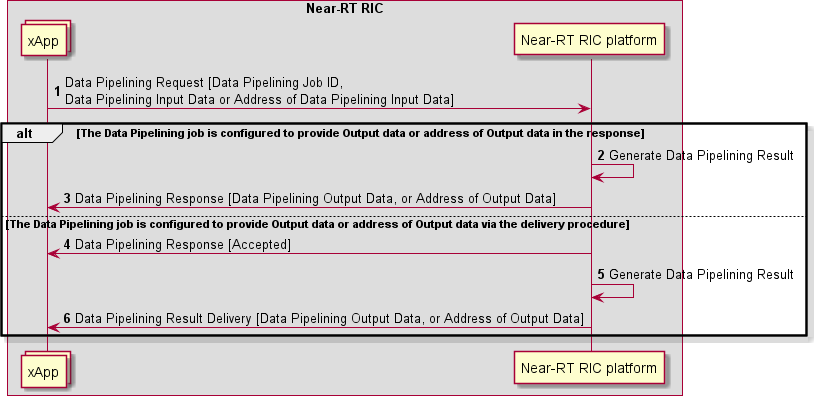


##### Figure 9.7.1.4-1: Data Pipelining Job Query procedure

#### Data Pipelining Request and Result Delivery procedure

The Data Pipelining Request and Result Delivery procedure allows an xApp to request a data pipelining result from Near-RT RIC platform through providing data pipelining input data.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | xApp requests to get a data pipelining result from the Near-RT RIC platform. |
| Actors and Roles | * xApp: Originator of data pipelining request; * Near-RT RIC platform: Response to the request of xApp. |
| Pre-conditions | * AI/ML workflow services have been registered by Near-RT RIC platform; * At least one data pipelining job setup by xApp is running in the Near-RT RIC platform. |
| Begins when | xApp has new data pipelining input data or address to the new data pipelining input data and  determine to use running data pipelining job to generates data pipelining results for xApp. |
| Step 1 (M) | xApp sends Near-RT-RIC platform Data Pipelining request. |
|  | [ALT] Step 2-3 are executed if the data pipelining job is configured to provide output data or  address of output data in the response. |
| Step 2 (M) | Near-RT-RIC platform generates data pipelining results. |
| Step 3 (M) | Near-RT-RIC platform responds to xApp, with data pipelining result. |
|  | [ELSE] Steps 4-6 are executed if the data pipelining job is configured to provide output data or  address of output data via the delivery procedure. |
| Step 4 (M) | xApp is responded by the Near-RT RIC platform. |
| Step 5 (M) | Near-RT-RIC platform generates data pipelining results. |
| Step 6 (M) | Near-RT-RIC platform delivers the data pipelining result to the xApp. |



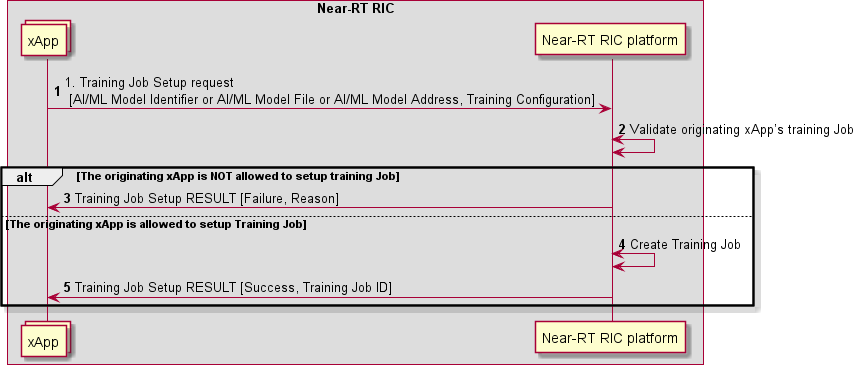
##### Figure 9.7.1.5-1: Data Pipelining Request and Result Delivery procedures

### Training service procedures

#### Training Job Setup procedure

The Training Job Setup procedure enables an xApp to setup a new training job in the Near-RT RIC platform.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | xApp requests to setup a training job in the Near-RT RIC platform. |
| Actors and Roles | * xApp: Originator of training job setup request; * Near-RT RIC platform: Response to the request of xApp. |
| Pre-conditions | * AI/ML workflow services have been registered by Near-RT RIC platform; |
| Begins when | xApp determines to setup a training job in the Near-RT-RIC platform. |
| Step 1 (M) | xApp sends a request to Near-RT RIC platform to setup a training job. |
| Step 2 (M) | Near-RT RIC platform checks whether the originating xApp is allowed to setup the training job. |
|  | [ALT] Step 3 is executed if the originating xApp is not allowed to setup the training job. |
| Step 3 (M) | A failure is responded to the originating xApp, possibly including appropriate reason. |
|  | [ELSE] Steps 4-5 are executed if the originating xApp is allowed to setup the training job. |
| Step 4 (M) | Near-RT RIC platform create corresponding training job. |
| Step 5 (M) | Near-RT RIC platform responds success to the originating xApp. |

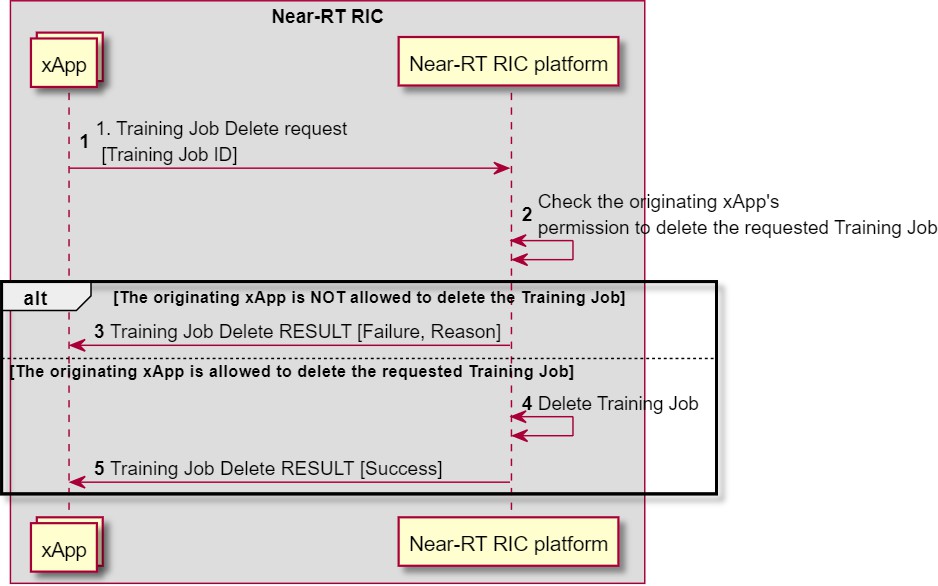


##### Figure 9.7.2.1-1: Training Job Setup procedure

#### Training Job Delete procedure

The Training Job Delete procedure enables an xApp to delete a running training job in the Near-RT RIC platform.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | xApp requests to delete a training job in the Near-RT RIC platform. |
| Actors and Roles | * xApp: Originator of training job delete request; * Near-RT RIC platform: Response to the request of xApp. |
| Pre-conditions | * AI/ML workflow services have been registered by Near-RT RIC platform; * At lease one training job setup by xApp is running in the Near-RT RIC platform. |
| Begins when | xApp determines to delete a training job in the Near-RT-RIC platform. |
| Step 1 (M) | xApp sends a request to Near-RT RIC platform to delete a training job. |
| Step 2 (M) | Near-RT RIC platform checks whether the originating xApp is allowed to delete the requested  training job |
|  | [ALT] Step 3 is executed if the originating xApp is not allowed to delete the requested training  job. |
| Step 3 (M) | A failure is responded to the originating xApp, possibly including appropriate reason. |
|  | [ELSE] Steps 4-5 are executed if the originating xApp is allowed to delete the requested training  job. |
| Step 4 (M) | Near-RT RIC platform deletes corresponding training job. |
| Step 5 (M) | Near-RT RIC platform responds success to the originating xApp. |

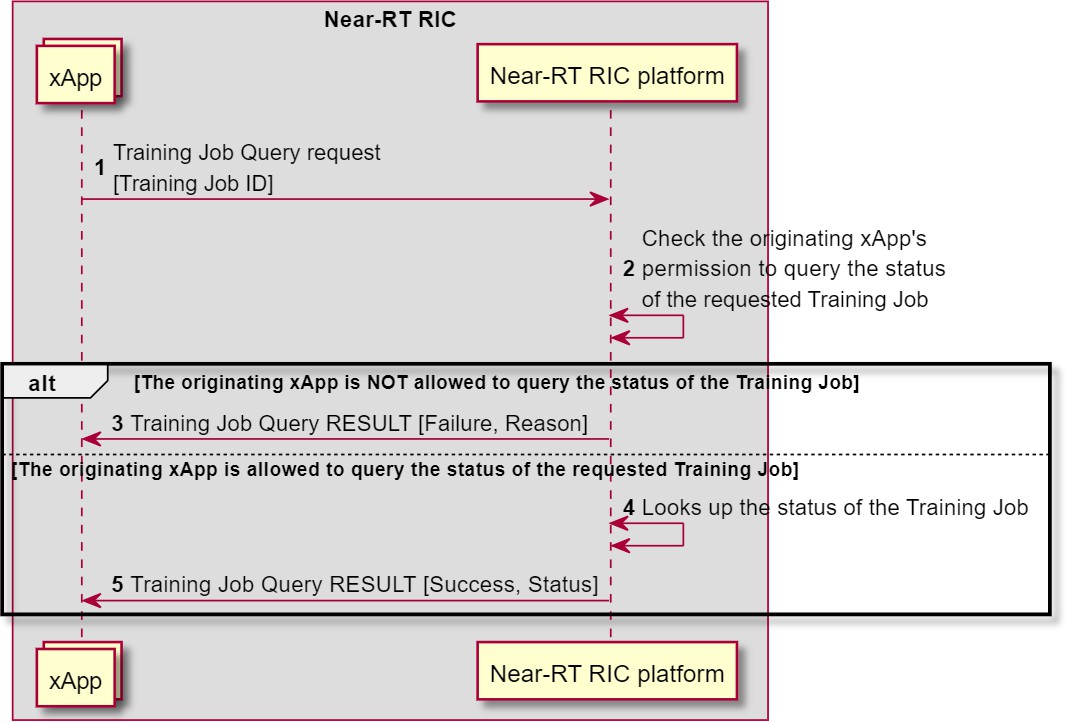


##### Figure 9.7.2.2-1: Training Job Delete procedure

#### Training Job Query procedure

The Training Job Query procedure enables an xApp to query the status of a running training job in the Near-RT RIC platform.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | xApp requests to query the status of a training job in the Near-RT RIC platform. |
| Actors and Roles | * xApp: Originator of training job query request; * Near-RT RIC platform: Response to the request of xApp. |
| Pre-conditions | * AI/ML workflow services have been registered by Near-RT RIC platform; * At least one training job setup by xApp is running in the Near-RT RIC platform. |
| Begins when | xApp determines to query the status of a training job in the Near-RT-RIC platform. |
| Step 1 (M) | xApp sends a request to Near-RT RIC platform to query the status of a training job. |
| Step 2 (M) | Near-RT RIC platform checks whether the originating xApp is allowed to query the status of the  requested training job. |
|  | [ALT] Step 3 is executed if the originating xApp is not allowed to query the status of the  requested training job. |
| Step 3 (M) | A failure is responded to the originating xApp, possibly including appropriate reason. |
|  | [ELSE] Steps 4-5 are executed if the originating xApp is allowed to query the status of the  requested training job. |
| Step 4 (M) | Near-RT RIC platform looks up the status of corresponding training job. |
| Step 5 (M) | Near-RT RIC platform responds success to the originating xApp, including the queried  status. |

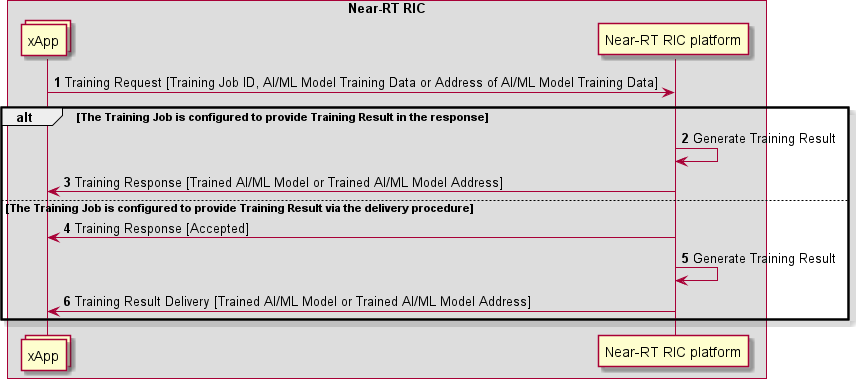


##### Figure 9.7.2.3-1: Training Job Query procedure

#### Training Request and Result Delivery procedure

The Training Request and Result Delivery procedure allows an xApp to request training result of an AI/ML model from Near- RT RIC platform through providing AI/ML model training data.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | xApp requests to get a training result from the Near-RT RIC platform. |
| Actors and Roles | * xApp: Originator of AI/ML model training request; * Near-RT RIC platform: Response to the request of xApp. |
| Pre-conditions | * AI/ML workflow services have been registered by Near-RT RIC platform; * At least one training job setup by xApp is running in the Near-RT RIC platform. |
| Begins when | Near-RT-RIC platform or xApp has new AI/ML model training data or Address of new AI/ML  model training data and determine to use running training job to train AI/ML model for xApp. |
| Step 1 (M) | xApp sends Near-RT-RIC platform training request. |
|  | [ALT] Step 2-3 are executed if the training job is configured to provide training result in the  response. |
| Step 2 (M) | Near-RT-RIC platform generates training results. |
| Step 3 (M) | Near-RT-RIC platform responds to xApp, with training result. |
|  | [ELSE] Steps 4-6 are executed if the training job is configured to provide training result via the  delivery procedure. |
| Step 4 (M) | xApp is responded by the Near-RT RIC platform. |
| Step 5 (M) | Near-RT-RIC platform generates training results. |
| Step 6 (M) | Near-RT-RIC platform delivers the training result to xApp. |



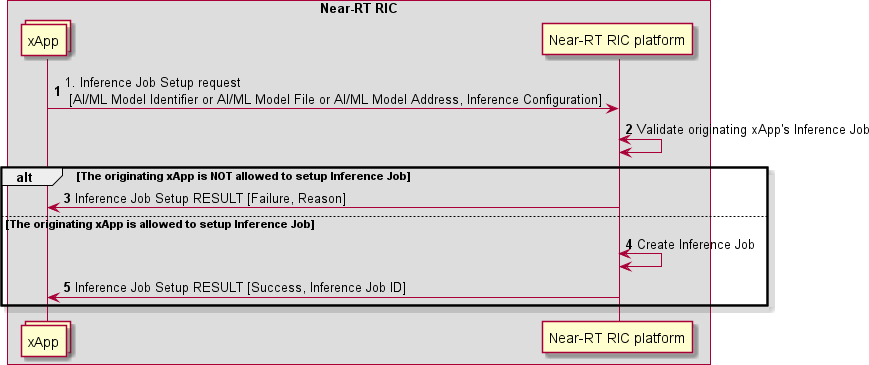
##### Figure 9.7.2.4-1: Training Request and Result Delivery procedures

### Inference service procedures

#### Inference Job Setup procedure

The Inference Job Setup procedure enables an xApp to setup a new inference job in the Near-RT RIC platform.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | xApp requests to setup an inference job in the Near-RT RIC platform. |
| Actors and Roles | * xApp: Originator of inference job setup request; * Near-RT RIC platform: Response to the request of xApp. |
| Pre-conditions | * AI/ML workflow services have been registered by Near-RT RIC platform; |
| Begins when | xApp determines to setup an inference job in the Near-RT-RIC platform. |
| Step 1 (M) | xApp sends a request to Near-RT RIC platform to setup an inference job. |
| Step 2 (M) | Near-RT RIC platform checks whether the originating xApp is allowed to setup inference job. |
|  | [ALT] Step 3 is executed if the originating xApp is not allowed to setup inference job. |
| Step 3 (M) | A failure is responded to the originating xApp, possibly including appropriate reason. |
|  | [ELSE] Steps 4-5 are executed if the originating xApp is allowed to setup inference job. |
| Step 4 (M) | Near-RT RIC platform create corresponding inference job. |
| Step 5 (M) | Near-RT RIC platform responds success to the originating xApp. |

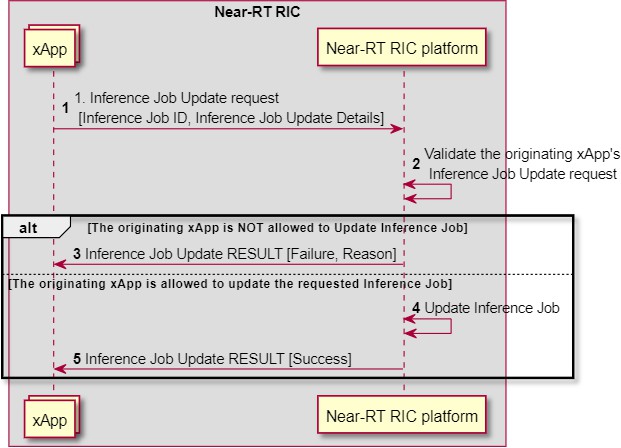


##### Figure 9.7.3.1-1: Inference Job Setup procedure

#### Inference Job Update procedure

The Inference Job Update procedure enables an xApp to update a running inference job in the Near-RT RIC platform.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | xApp requests to update an inference job in the Near-RT RIC platform. |
| Actors and Roles | * xApp: Originator of inference job update request; * Near-RT RIC platform: Response to the request of xApp. |
| Pre-conditions | * AI/ML workflow services have been registered by Near-RT RIC platform; * At least one inference job setup by xApp is running in the Near-RT RIC platform. |
| Begins when | xApp determines to update an Inference Job in the Near-RT-RIC platform. |
| Step 1 (M) | xApp sends a request to Near-RT RIC platform to update an inference job. |
| Step 2 (M) | Near-RT RIC platform checks whether the originating xApp is allowed to update the requested  inference job. |
|  | [ALT] Step 3 is executed if the originating xApp is not allowed to update the requested inference  job. |
| Step 3 (M) | A failure is responded to the originating xApp, possibly including appropriate reason. |
|  | [ELSE] Steps 4-5 are executed if the originating xApp is allowed to update the requested  inference job. |
| Step 4 (M) | Near-RT RIC platform updates corresponding inference job. |
| Step 5 (M) | Near-RT RIC platform responds success to the originating xApp. |

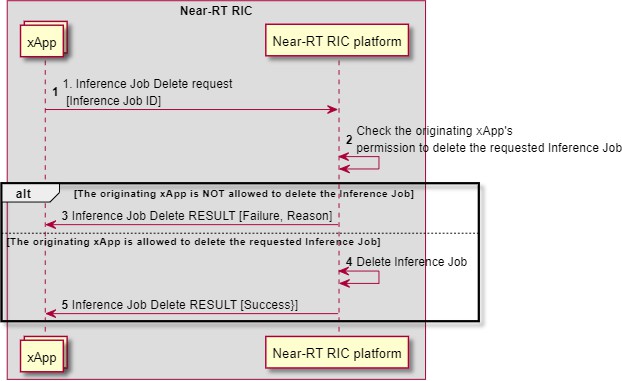


##### Figure 9.7.3.2-1: Inference Job Update procedure

#### Inference Job Delete procedure

The Inference Job Delete procedure enables an xApp to delete a running inference job in the Near-RT RIC platform.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | xApp requests to delete an inference job in the Near-RT RIC platform. |
| Actors and Roles | * xApp: Originator of inference job delete request; * Near-RT RIC platform: Response to the request of xApp. |
| Pre-conditions | * AI/ML workflow services have been registered by Near-RT RIC platform; * At least one inference job setup by xApp is running in the Near-RT RIC platform. |
| Begins when | xApp determines to delete an inference job in the Near-RT-RIC platform. |
| Step 1 (M) | xApp sends a request to Near-RT RIC platform to delete an inference job. |
| Step 2 (M) | Near-RT RIC platform checks whether the originating xApp is allowed to delete the requested  inference job. |
|  | [ALT] Step 3 is executed if the originating xApp is not allowed to delete the requested inference  job. |
| Step 3 (M) | A failure is responded to the originating xApp, possibly including appropriate reason. |
|  | [ELSE] Steps 4-5 are executed if the originating xApp is allowed to delete the requested  inference job. |
| Step 4 (M) | Near-RT RIC platform deletes corresponding inference job. |
| Step 5 (M) | Near-RT RIC platform responds success to the originating xApp. |

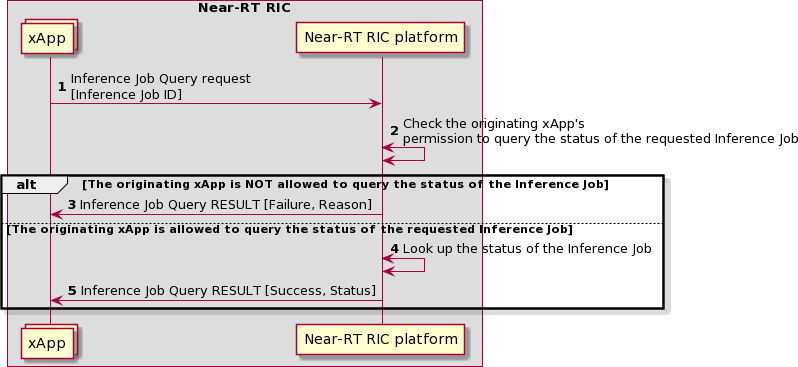


##### Figure 9.7.3.3-1: Inference Job Delete procedure

#### Inference Job Query procedure

The Inference Job Query procedure enables an xApp to query the status of a running inference job in the Near-RT RIC platform.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | xApp requests to query the status of an inference job in the Near-RT RIC platform. |
| Actors and Roles | * xApp: Originator of inference job query request; * Near-RT RIC platform: Response to the request of xApp. |
| Pre-conditions | * AI/ML workflow services have been registered by Near-RT RIC platform; * At least one inference job setup by xApp is running in the Near-RT RIC platform. |
| Begins when | xApp determines to query the status of an Inference Job in the Near-RT-RIC platform. |
| Step 1 (M) | xApp sends a request to Near-RT RIC platform to query the status of an inference job. |
| Step 2 (M) | Near-RT RIC platform checks whether the originating xApp is allowed to query the status of the  requested inference job. |
|  | [ALT] Step 3 is executed if the originating xApp is not allowed to query the status of the  requested inference job. |
| Step 3 (M) | A failure is responded to the originating xApp, possibly including appropriate reason. |
|  | [ELSE] Steps 4-5 are executed if the originating xApp is allowed to query the status of the  requested inference job. |
| Step 4 (M) | Near-RT RIC platform looks up the status of corresponding inference job. |
| Step 5 (M) | Near-RT RIC platform responds success to the originating xApp, including the queried  status. |

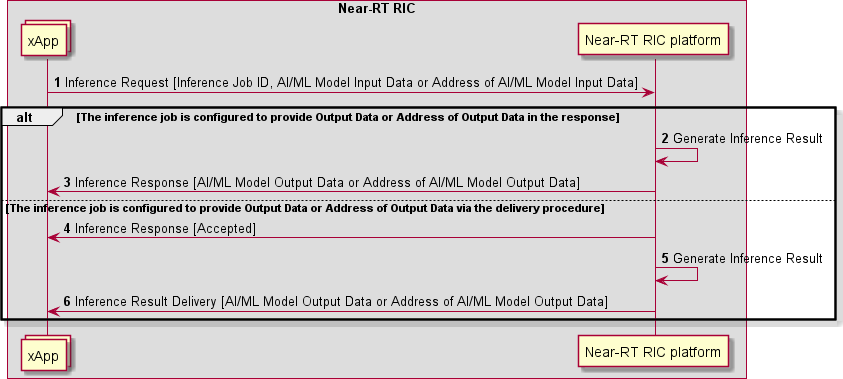


##### Figure 9.7.3.4-1: Inference Job Query procedure

#### Inference Request and Result Delivery procedures

The Inference Request and Result Delivery procedure allows an xApp to request an AI/ML model inference result from Near- RT RIC platform through providing model input data.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | xApp requests to get an inference result from the Near-RT RIC platform. |
| Actors and Roles | * xApp: Originator of AI/ML model inference request; * Near-RT RIC platform: Response to the request of xApp. |
| Pre-conditions | * AI/ML workflow services have been registered by Near-RT RIC platform; * At least one inference job setup by xApp is running in the Near-RT RIC platform. |
| Begins when | Near-RT-RIC platform has new model input data and determine to use running inference job to  generates inference results for xApp. |
| Step 1 (M) | xApp sends Near-RT-RIC platform inference request. |
|  | [ALT] Step 2-3 are executed if the Inference Job is configured to provide output data in the  response. |
| Step 2 (M) | Near-RT-RIC platform generates inference results. |
| Step 3 (M) | Near-RT-RIC platform responds to xApp, with inference result. |
|  | [ELSE] Steps 4-6 are executed if the inference job is configured to provide output data via the  delivery procedure. |
| Step 4 (M) | xApp is responded by the Near-RT RIC platform. |
| Step 5 (M) | Near-RT-RIC platform generates inference results. |
| Step 6 (M) | Near-RT-RIC platform delivers the inference result to xApp. |



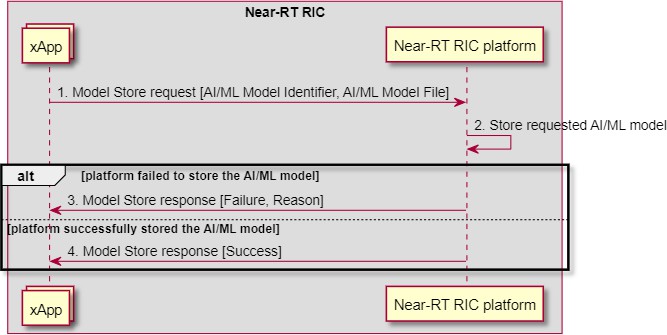
##### Figure 9.7.3.5-1: Inference Request and Result Delivery procedures

### Model Management and Exposure Service procedures

#### Model Store procedure

The Model Store procedure in the Near-RT RIC allows an xApp to request to store an AI/ML model in the Near-RT RIC platform. An AI/ML Model is identified by a model identifier.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | xApp requests to store an AI/ML model in the Near-RT RIC platform. |
| Actors and Roles | * xApp: Originator of Model Store request; * Near-RT RIC platform, with the following functionalities:   + AI/ML support. |
| Pre-conditions | * AI/ML workflow services have been registered by Near-RT RIC platform; |
| Begins when | xApp determines to store an AI/ML model in the Near-RT-RIC platform. |
| Step 1 (M) | xApp sends a request to Near-RT RIC platform to store an AI/ML model. |
| Step 2 (M) | Near-RT RIC platform stores the AI/ML model in an xApp specific space. |
|  | [ALT] Step 3 is executed if the platform failed to stored the AI/ML model. |
| Step 3 (M) | A failure is responded to the originating xApp, possibly including appropriate reason. |
|  | [ELSE] Step 4 is executed if the platform successfully stored the AI/ML model. |
| Step 4 (M) | Near-RT RIC platform responds success to the originating xApp. |

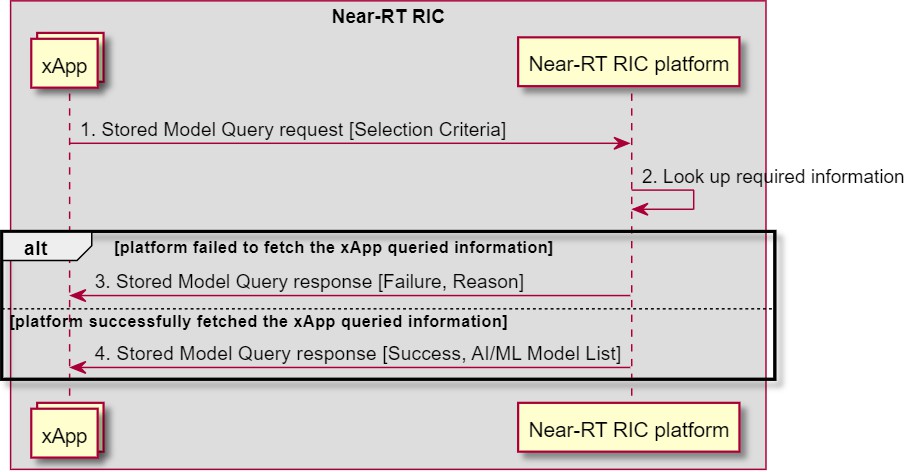


##### Figure 9.7.4.1-1: Model Store procedure

#### Stored Model Query procedure

The Stored Model Query procedure in the Near-RT RIC enables an xApp to query the list of stored AI/ML models in the Near- RT RIC.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | An xApp queries the list of stored AI/ML models from the Near-RT RIC. |
| Actors and Roles | * xApp: Originator of Stored Model Query request; * Near-RT RIC platform, with the following functionalities:   + AI/ML support. |
| Pre-conditions | * AI/ML workflow services have been registered with Near-RT RIC platform; * At least one AI/ML model has been stored by xApp in the Near-RT RIC Platform |
| Begins when | xApp determines to query the list of stored AI/ML models that it has stored from the Near-RT  RIC platform. |
| Step 1 (M) | xApp requests to Near-RT RIC platform to query stored AI/ML models that meets provided  selection criteria. |
| Step 2 (M) | Near-RT RIC platform fetches xApp queried information. |
|  | [ALT] Step 3 is executed if the platform failed to fetch the xApp queried information. |
| Step 3 (M) | A failure is responded to the originating xApp, possibly including appropriate reason. |
|  | [ELSE] Step 4 is executed if the platform successfully fetched the xApp queried information. |
| Step 4 (M) | Near-RT RIC platform responds success to the originating xApp, with requested AI/ML  model list. |

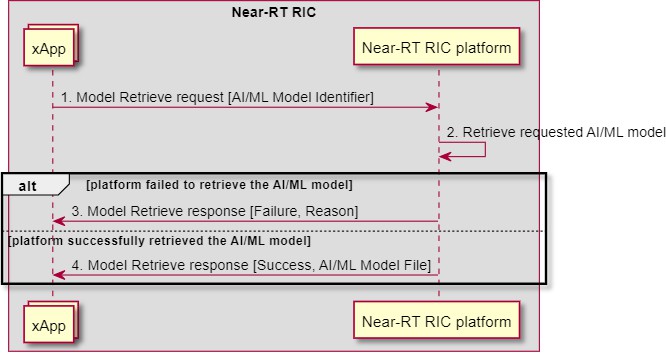


##### Figure 9.7.4.2-1: Stored Model Query procedure

#### Model Retrieve procedure

The Model Retrieve procedure in the Near-RT RIC enables an xApp to retrieve an AI/ML model it has stored in the Near-RT RIC.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | xApp retrieves an AI/ML model it has stored from the Near-RT RIC platform. |
| Actors and Roles | * xApp: Originator of Model Fetch request; * Near-RT RIC platform, with the following functionalities:   + AI/ML support. |
| Pre-conditions | * AI/ML workflow services have been registered with Near-RT RIC platform; * At least one AI/ML model has been stored by xApp in the Near-RT RIC Platform |
| Begins when | xApp determines to retrieve an AI/ML model from the Near-RT-RIC platform. |
| Step 1 (M) | xApp requests to Near-RT RIC platform to retrieve an AI/ML model from the Near-RT-RIC  platform. |
| Step 2 (M) | Near-RT RIC platform retrieves xApp required AI/ML model from Near-RT-RIC platform. |
|  | [ALT] Step 3 is executed if the platform failed to retrieve the AI/ML model. |
| Step 3 (M) | A failure is responded to the originating xApp, possibly including appropriate reason |
|  | [ELSE] Step 4 is executed if the platform successfully retrieved the AI/ML model. |
| Step 4 (M) | Near-RT RIC platform responds success to the originating xApp, with AI/ML Model File. |

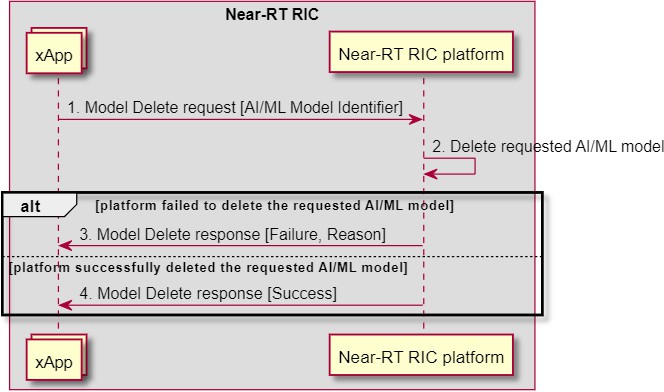


##### Figure 9.7.4.3-1: Model Retrieve procedure

#### Model Delete procedure

The Model Delete procedure in the Near-RT RIC allows an xApp to request to Near-RT RIC platform to delete an AI/ML model it has stored from the platform.

|  |  |
| --- | --- |
| **Use Case Stage** | **Evolution / Specification** |
| Goal | xApp intends to delete an AI/ML model it has stored in the Near-RT RIC. |
| Actors and Roles | * xApp: Originator of Model Delete request; * Near-RT RIC platform, with the following functionalities:   + AI/ML support. |
| Pre-conditions | * AI/ML workflow services have been registered by Near-RT RIC platform; * xApp has already stored the AI/ML model in the Near-RT RIC platform it requests to delete. |
| Begins when | xApp determines to delete an AI/ML model in the Near-RT-RIC platform. |
| Step 1 (M) | xApp sends a request to Near-RT RIC platform to delete an AI/ML model in the platform. |
| Step 2 (M) | Near-RT RIC platform deletes xApp requested AI/ML model in platform. |
|  | [ALT] Step 3 is executed if the platform failed to delete the requested AI/ML model |
| Step 3 (M) | A failure is responded to the originating xApp, possibly including appropriate reason. |
|  | [ELSE] Steps 4 are executed if the platform successfully deleted the requested AI/ML model |
| Step 4 (M) | Near-RT RIC platform responds success to the originating xApp |



##### Figure 9.7.4.4-1: Model Delete procedure

Annex A (informative):

Potential conflict types among xApps for RRM optimization

The control target of the radio resource management can be a cell, a UE or a bearer, etc. The control contents of the radio resource management can cover access control, bearer control, handover control, QoS control, resource assignment and so on. The control time span indicates the valid control duration which is expected by the control request. The conflicts of control can be illustrated as below.

1. Direct Conflicts: The conflicts can be observed directly by Conflict Mitigation. Some cases are described as below:
   * Two or more xApps request different settings for the very same configuration of one or more parameters of a Control Target. Conflict mitigation processes the requests and decides on a resolution.
   * The new request from an xApp may conflict with the running configuration resulting from a previous request of another or the same xApp.
   * The total requested resources from different xApps may exceed the limitation of the RAN system, e.g., the sum of resources required by the two different xApps may be far beyond the resource limitation of the RAN system.
2. Indirect Conflicts: The conflicts cannot be observed directly, nevertheless, some dependence among the parameters and resources that the xApps target can be observed. Conflict Mitigation may anticipate the possible conflicts and take actions to mitigate them. For instance, different xApps target different configuration parameters to optimize the same metric according to the respective objective. Even though this will not result in conflicting parameter settings, it may have uncontrollable or inadvertent system impacts. One example of such indirect conflicts can occur when the changes required by one xApp create a system impact which is equivalent to a parameter change targeted by another xApp. E.g., antenna tilts and measurement offsets are different control points, but they both impact the handover boundary.
3. Implicit Conflicts: The conflicts cannot be observed directly, even the dependence between xApps is not obvious. For instance, different xApps may optimize different metrics and (re-)configure different parameters. Nonetheless, optimizing one metric may have implicit, unwanted, and maybe adversary side effects on one of the metrics optimized by another xApp. E.g., protecting throughput metrics for GBR users may degrade non-GBR metrics or even cell throughput.

For mitigating these conflicts, different approaches exist:

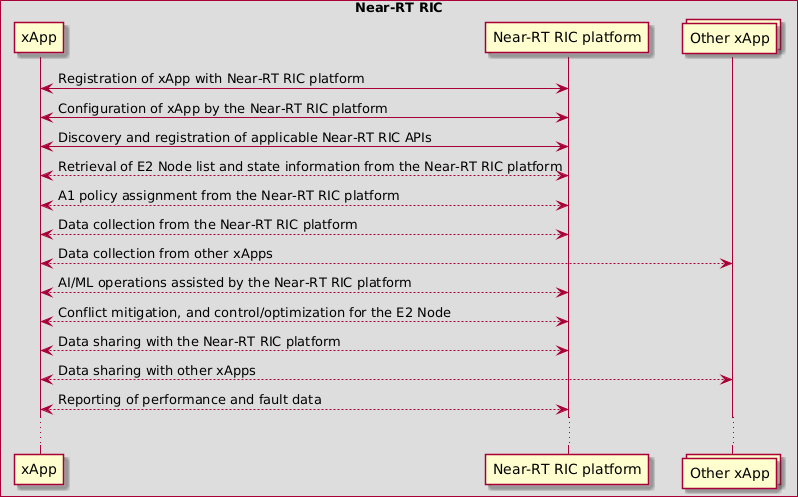
1. Direct conflicts typically can be mitigated by pre-action coordination, i.e., the xApps or a Conflict Mitigation component needs to make the final determination on whether any specific change is made, or in which order the changes are applied.
2. Indirect conflicts can be resolved by post-action verification. Here, the actions are executed and the effects on the target metric are observed. Based on the observations, the system has to decide on potential corrections, e.g., rolling back one of the xApp actions.
3. Implicit conflicts are the most difficult to mitigate since these dependencies are difficult or impossible to observe and therefore hard to model in any mitigation scheme. In some cases, it may be possible to design around such conflicts by ensuring that use cases (xApps) target different parameters, thus falling back to approach 2), but preferably, a generic approach to managing such conflicts is established.

The individual xApp goals are defined by A1 policies, but it is also important to define utility metrics that incorporate the relative importance of each of the metrics targeted by the xApps as well as the importance of the optimization (use case). A Conflict Mitigation function may also use ML approaches, e.g., Reinforcement Learning, to a-priori assess, for each proposed change, the likely probability of degrading a metric versus the potential improvement.

Annex B (informative):

Example of xApp's workflow

Exemplified in this annex is a high-level description of an xApp's workflow, which may occur between the xApp and the Near- RT RIC platform after the xApp is instantiated on O-Cloud. Note that some operations may not be performed, and that the sequence and repetitions of the operations may vary from this example, depending on the use cases and the xApp's business logic.



##### Figure B-1: Example of xApp's workflow after its instantiation

Figure B-1 shows the following operations for the xApp:

* Registration of xApp with Near-RT RIC platform: This is achieved via the xApp registration procedure in clause 9.4.1.
* Configuration of xApp by the Near-RT RIC platform: This is achieved via the Management API procedures in clause 9.4.
* Discovery and registration of applicable Near-RT RIC APIs: This applies to the Near-RT RIC APIs that are not preconfigured, and is achieved via the API Enablement procedures in clause 9.6.
* Retrieval of E2 Node list and state information from the Near-RT RIC platform: This is achieved via the SDL API procedures in clause 9.5.
* A1 policy assignment from the Near-RT RIC platform: This is achieved via the A1 Policy API procedures in clause 9.2.2.
* Data collection from the Near-RT RIC platform: This is achieved via the E2 related API procedures in clause 9.3.2, the A1-EI API procedures in clause 9.2.3, and the SDL API procedures in clause 9.5.
* Data collection from other xApps: This is achieved via the SDL API procedures in clause 9.5.
* AI/ML operations assisted by the Near-RT RIC platform: This is achieved via the AI/ML API procedures in clause 9.7.
* Conflict mitigation, and control/optimization for the E2 Node: This is achieved via the E2 related API procedures in clause 9.3.
* Data sharing with the Near-RT RIC platform: This is achieved via the SDL API procedures in clause 9.5.
* Data sharing with other xApps: This is achieved via the SDL API procedures in clause 9.5.
* Reporting of performance and fault data: This is achieved via the Management API procedures in clause 9.4.

NOTE: The xApp’s communications with the Near-RT RIC platform and other xApps are protected according to the security requirements specified in WG11.

# Annex (informative): Change History

|  |  |  |
| --- | --- | --- |
| **Date** | **Revision** | **Description** |
| 2019.11.01 | 00.00.00 | Initial Skeleton |
| 2019.11.11 | 00.00.01 | Edits to Near-RT RIC functions description (Section 6) |
| 2019.11.19 | 00.01.00 | Approved skeleton based on WG3\_2019.11.19\_CMCC\_ORAN-WG3.RIC-  ARCH.Skeleton\_v2 |
| 2019.12.04 | 00.01.01 | Intermediate version based on WG3 2019-12-2\_ATT-CICT-CMCC\_CR\_nRT-RIC-  Arch\_Section45\_v3 |
| 2019.12.13 | 00.01.02 | Intermediate version based on WG3 2019.12.09\_JIO\_CR\_ORAN-  WG3.RICARCH.0\_running v00.01.01\_v2 |
| 2019.12.20 | 00.01.03 | Intermediate version based on WG3 2019-12\_16\_CMCC\_ATT\_CR\_nRT-RIC-  Arch\_Section6 1\_v4.1 |
| 2020.01.21 | 00.01.04 | Intermediate version based on |
| 2020.01.22 | 00.01.05 | WG3\_2020-1\_14\_ATT\_CR\_NearRTRICArchSection 6.docx |
| 2020.02.04 | 00.01.06 | WG3\_2020-01-10\_Lenovo\_CR\_NearRTRICArch\_section6.2.4\_v2.1\_xxcmt.docx |
| 2020.02.12 | 00.01.07 | Intermediate version based on |
| 2020.02.12 | 00.01.08 | WG3\_2020 01 21\_Nokia\_CICT\_CMCC\_CR\_near-RT RIC Architecture Merging Conflict  Mitigation\_v0.1a\_clean |
| 2020.02.13 | 00.01.09 | WG3\_2020-01-21 \_Lenovo\_CR\_NearRTRICArch\_section6.3\_v1.4\_xApp |
| 2020.02.14 | 00.01.10 | Intermediate version based on WG3 TG approval and |
| 2020.02.17 | 01.00.00 | Version number changed to 01.00.00 for WG3 approval |
| 2020.02.24 | 01.00.00 | Editorial corrections collected during WG3 approval process |
| 2020.10.22 | 01.00.01 | Editorial corrections collected from Intel about Non-RT RIC |
| 2020.10.28 | 01.00.02 | The name of API was changed to Near-RT RIC API according to the agreement on TG  Conference #33 |
| 2020.11.03 | 01.00.03 | Addition of CR adopted during TG Conference #33 |
| 2020.11.04 | 01.00.04 | "RIC framework" was changed to "RIC platform" according to the agreement on TG  Conference #34 |
| 2020.11.06 | 01.00.05 | Addition of CRs adopted during TG Conference #34 |
| 2020.11.08 | 01.01.00 | Editorial corrections collected during WG3 email reflector final review |
| 2020.11.17 | 01.01.00 | Withdraw section 7.7 from RICARCH v1.1 according to the comment during WG3 email  reflector final review |
| 2020.11.25 | 01.01.00 | Version 1.1 for Board approval after 2021 copyright update proposed by TSC |
| 2020.12.14 | 02.00.01 | Merged the approved CR CICT-2020.07.28-WG3-CR-xxxx-prdcedures-in-A1-related-  APIs-v1.4.1.docx according to the agreement on TG Conference #34. |
| 2021.02.03 | 02.00.02 | Discuss the next step of Section 7.7 from the CR ONF-AO-2020.10.27-WG3-CR-001-  Open API design-v03.docx. |
| 2021.02.23 | 02.00.03 | Merged the approved CR NOK-2020.11.26-WG3-CR-0001-RICARCHsection9.x-v6.docx  according to the agreement in TG conference #39 |
| 2021.03.02 | 02.00.04 | Deleted section 7.7, updated section 9.1 using the newly agreed CR CICT-2020.07.28-  WG3-CR-0002-procedures in A1 related APIs-v2.0.0.docx, removed (M) and (O) for the informative steps in section 9 according to the agreement on TG Conference #40, . |
| 2021.03.05 | 02.00.05 | Addition of API enablement function, and Stage-2 definitions for part of the Near-RT RIC  APIs. |
| 2021.03.06 | 02.00 | Incremented version for WG3 vote |
| 2021.03.14 | 02.00 | Editorial updates received before TSC vote |
| 2021.03.16 | 02.00 | Removal of UML code, and editorial updates during TSC vote |
| 2021.05.27 | 02.01.01 | Clean version with UML code |
| 2021.06.15 | 02.01.02 | Incorporated CRs: |
| 2021.10.12 | 02.01.03 | FB.AO-2021.04.23-O-RAN-CR-1001-SDL Subscribe Push-v01.docx |
| 2021.11.08 | 02.01.04 | INT.AO-2021.05.04-WG3-CR-0004-RICARCH-AIMLsupport\_v2.docx |
| 2022.04.07 | 02.01.05 | Implemented CRs:  (1) CMCC CR-0002 in ARCH#65; (2) NOK.AO CR-0008 in ARCH#67; (3) NOK.AO CR- 0009 in ARCH#67; |
| 2022.05.06 | 02.01.06 | Implemented CRs:  (1) NOK CR-0011 in ARCH#68; (2) NOK CR-0005 in ARCH#68; |

|  |  |  |
| --- | --- | --- |
| 2022.07.15 | 02.01.07 | Implemented CRs:  (1) CICT CR-0005 in ARCH#69; (2) NOK CR-0013 in ARCH#71; (3) CMCC CR-0003 in ARCH#73; (4) CMCC CR-0004 in ARCH#73; |
| 2022.07.22 | 02.01.08 | Editorial updates. |
| 2022.07.29 | 02.01.09 | Editorial updates during WG3 electronic vote. |
| 2022.08.02 | 03.00 | Version incremented as 03.00 for TSC |
| 2022.09.16 | 03.00.00 | Initial baseline |
| 2022.11.11 | 03.00.01 | Addition of CRs:  <CMCC-WG3-CR-0005-RICARCH-Add Y1 descriptions for Near-RT RIC architecture- v2.docx>  <NOK.AO-2022.10.28-WG3-CR-0017-RICARCH 5.1.3 Management API  Requirements\_rev2.docx>  <NOK.AO-2022.09.28-WG3-CR-0014-RICARCH 9.4.5 Management APIs\_rev5.docx>  <NOK.AO-2022.09.28-WG3-CR-0015-RICARCH 9.4.6 Management APIs\_rev4.docx>  <NOK.AO-2022.09.28-WG3-CR-0016-RICARCH 9.4.7 Management APIs\_rev5.docx>  Minor editorial clean-ups. |
| 2022.11.17 | 03.00.02 | Changes reflecting remarks received during WG3 approval process   * Update copyright year 2023, and release number R003. * Fix xApp/Near-RT RIC platform in the “Actors and Roles” of tables in 9.4.8 and 9.4.13. * Fix the labels for a few steps in 9.4.8, 9.4.12 and 9.4.13 as “[informative]” instead of “(M)”. * Add missing step descriptions in 9.4.8. * Other minor editorial clean-ups. |
| 2022.11.20 | 04.00 | Version incremented to 04.00 for TSC. |
| 2022.12.13 | 04.00.00 | Initial baseline |
| 2023.03.29 | 04.00.01 | Addition of CRs:   * <CICT-2022.08.26-WG3-CR-0011-Function updates related to direct A1 policy to suitable xApps\_v4.docx> * <CMCC-2023.02.21-WG3-CR-0006-RICARCH-Editorial Updates to Section 4 and 5- v02.docx>   Minor editorial clean-ups. |
| 2023.07.10 | 04.00.02 | Addition of CRs:   * CMCC-2023.03.31-WG3-CR-0007-RICARCH-Improvement in Section 6-v05.docx * CMCC.AO-2023.05.08-WG3-CR-0010-Service definitions for Near-RT RIC APIs- v04.docx   Typo fix in the caption of Fig 6.1-1. |
| 2023.07.17 | 04.00.03 | Addition of CR:  - CMCC-2023.06.14-WG3-CR-0011-RICARCH-Fix the use of Near-RT RIC platform and its functionalities in clause 9 -v03.docx  Minor editorial clean-ups. |
| 2023.07.21 | 04.00.04 | Addition of CR:  CMCC-2023.04.18-WG3-CR-0009-AIML Support Function-v05.docx Updated to new O-RAN TS template. |
| 2023.07.28 | 04.00.05 | Resolution of review comments during WG3 poll. |
| 2023.07.29 | 05.00 | Version increment for TSC approval. |
| 2023.08.08 | 05.00.00 | Initial baseline |
| 2023.11.24 | 05.00.01 | Addition of CRs:   * <NEC-2023.08.13-WG3-CR-0017-RIC ARCH Clarification on E2 INDICATION message order\_v3.docx> * <CMCC.AO-2023.07.10-WG3-CR-0013-Addition of text for service consumer and producer in clause 6A-v03.docx> * <NOK-2023.06.21-WG3-CR-0018-registration procedure with secure xApp ID assignment\_v05.docx> * <CMCC-2023.09.20-WG3-CR-0014-RICARCH-Services for AIML-v02.docx> * <NOK-2023.10.27-WG3-CR-0019-RICARCH-Subscription Audit-v02.docx> * <NOK-2023.02.10-WG3-CR-0015-RICARCH-APIEnablementAPIextension-v07.docx> Adding the full change history.   Editorial corrections and improvements. |
| 2024.03.11 | 05.00.02 | Addition of CRs:  - <CMCC-2023.11.07-WG3-CR-0017-RICARCH-xApp Data Sharing Service for RAI sharing from xApps-v1.docx> |

|  |  |  |
| --- | --- | --- |
|  |  | * <CMCC-2024.01.30-WG3-CR-00019-RICARCH-AIML data pipelining services- v02.docx> * <CMCC-2024.01.30-WG3-CR-00018-RICARCH-AIML training services-v02.docx> * <CMCC-2023.09.20-WG3-CR-00016-RICARCH-AIML inference services-v02.docx> * <CMCC-2024.02.05-WG3-CR-0020-RICARCH-Stage 2 enhancements to SDL   Information API-v01.docx>   * <NOK-2024.02.21-WG3-CR-0021-RICARCH-TimeoutCases-v01.docx>   Editorial corrections and improvements. |
| 2024.03.21 | 05.00.03 | Addition of CRs:  - <NOK-2024.03.12-WG3-CR-00025-RICARCH-Clause-5,6,6A,7,9-CORRECTIONS-  v02.docx>   * <NOK-2024.03.12-WG3-CR-00024-RICARCH-AIML data pipelining services- CORRECTIONS-v01.docx> * <NOK-2024.03.12-WG3-CR-00023-RICARCH-AIML training services-CORRECTIONS-   v03.docx>   * <NOK-2024.03.12-WG3-CR-00022-RICARCH-AIML inference services- CORRECTIONS-v03.docx> |
| 2024.03.31 | 05.00.04 | Editorial corrections during WG3 review for March 2024 publication train. |
| 2024.04.01 | 06.00 | Version incremented as 06.00 for TSC |
| 2024.08.20 | 06.00.01 | Addition of CRs:   * <CUC-2024.4.9-WG3-CR-0001-RICARCH-some editorial corrections\_v01.docx> * <CUC-2024.4.15-WG3-CR-0002-RICARCH-Correct message in E2 Control procedure\_v01.docx>   - <NOK-2023.11.29-WG3-CR-0020-RICARCH-Terms-v03.docx>   * <NOK-2024.04.03-WG3-CR-0026-RICArch OAM requirements\_v03.docx> * <CMCC-2023.06.30-WG3-CR-0012-RICARCH-Service definition for A1 Policy API- v03.docx> * <CMCC-2023.09.20-WG3-CR-00015-RICARCH-model management and exposure services-v04.docx> * <[NOK-2024.06.05-WG3-CR-0027-RICArch requirement corrections v05.docx](https://oranalliance.atlassian.net/wiki/download/attachments/3211657238/NOK-2024.06.05-WG3-CR-0027-RICArch%20requirement%20corrections%20v05.docx?api=v2)> * <[CMCC-2024.05.20-WG3-CR-0021-RICARCH-An informative example of xApp's](https://oranalliance.atlassian.net/wiki/download/attachments/3211657238/CMCC-2024.05.20-WG3-CR-0021-RICARCH-An%20informative%20example%20of%20xApp%27s%20workflow-v05.docx?api=v2) [workflow-v05.docx](https://oranalliance.atlassian.net/wiki/download/attachments/3211657238/CMCC-2024.05.20-WG3-CR-0021-RICARCH-An%20informative%20example%20of%20xApp%27s%20workflow-v05.docx?api=v2)>   Editorial corrections and improvements. |
| 2024.11.12 | 06.00.02 | Addition of CRs:   * <NEC-2024.10.29-WG3-CR-0034-RICARCH-Add E2 Query-v02.docx> * <NEC-2024.10.29-WG3-CR-0035-RICARCH-Add E2 Subscription Modification- v02.docx> * <NOK-2024.10.15-WG3-CR-0028-RICARCH-ConMit-adding-general-support- descriptions-v03.docx> * <NOK-2024.10.15-WG3-CR-0029-RICARCH-ConMit-adding-E2 Conflict Mitigation Section-v04.docx> * <CMCC-2024.11.13-WG3-CR-0022-RICARCH-Reference cleanups and updates- v01.docx>   Editorial corrections and improvement. |
| 2024.12.05 | 06.00.03 | Editorial improvements during the WG3 poll. |
| 2024.12.05 | 07.00 | Version incremented to 07.00 for TSC. |