O-RAN.WG3.RICARCH-v02.00

Technical Specification

O-RAN Working Group 3
Near-Real-time RAN Intelligent Controller
Near-RT RIC Architecture

Copyright © 2021 by O-RAN ALLIANCE e.V.

By using, accessing or downloading any part of this O-RAN specification document, including by copying, saving, distributing, displaying or preparing derivatives of, you agree to be and are bound to the terms of the O-RAN Adopter License Agreement contained in the Annex ZZZ of this specification. All other rights reserved.

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

Revision History

Date	Revision	Author	Description
2020.02.17	01.00.00	WG3 Arch TG	Version number changed to 01.00.00 for WG3 approval
2020.11.25	01.01.00	WG3 Arch TG	Version 1.1 for Board approval after 2021 copyright update proposed by TSC
2020.03.05	02.00.05	CMCC, Nokia, CICT, Lenovo, AsiaInfo, Intel	Addition of API enablement function, and Stage-2 definitions for part of the Near-RT RIC APIs.
2020.03.06	02.00	CMCC	Incremented version for WG3 vote
2020.03.14	02.00	CMCC, Reliance Jio, Intel	Editorial updates received before TSC vote
2020.03.16	02.00	CMCC, Intel	Removal of UML code, and editorial updates during TSC vote

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

Contents

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

Foreword

This Technical Specification has been produced by the O-RAN ALLIANCE.

The contents of the present document are 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 release date and an increase in version number as follows:

Release xx.yy.zz

where:

xx the first digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc. (the initial approved document will have xx=01).

yy the second digit is incremented when editorial only changes have been incorporated in the document.

zz the third digit included only in working versions of the document indicating incremental changes during the editing process.

1 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.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. 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 the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[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-WG1.OAM Architecture, “O-RAN Operations and Maintenance Architecture”.

[5] O-RAN-WG1.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”.

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

[8] O-RAN-WG2.A1.GA&P, “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.O-RAN Architecture, “O-RAN Working Group 1, O-RAN Architecture Description”.

[11] 3GPP TS 36.401: "Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Architecture Description".

[12] 3GPP TS 38.300: “NR; NR and NG-RAN Overall Description; Stage 2”.

[13] 3GPP TS 37.324: “Evolved Universal Terrestrial Radio Access (E-UTRA) and NR; Service Data Adaptation Protocol (SDAP) specification”.

[14] 3GPP TS 38.331: “NR; Radio Resource Control (RRC); Protocol specification”.

[15] 3GPP TS 38.323: “NR; Packet Data Convergence Protocol (PDCP) specification”.

[16] 3GPP TS 38.322: “NR; Radio Link Control (RLC) protocol specification”.

[17] 3GPP TS 38.321: “NR; Medium Access Control (MAC) protocol specification”.

[18] 3GPP TS 38.201: “NR; Physical layer; General description”.

3 Definitions and Abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in TR 21.905 [1].

Near-RT RIC: O-RAN Near-Real-Time RAN Intelligent Controller: A logical function that enables near-real-time control and optimization of RAN elements and resources via fine-grained data collection and actions over E2 interface.

Non-RT RIC: O-RAN Non-Real-Time RAN Intelligent Controller: A logical function within SMO that drives the content carried across the A1 interface. It is comprised of the Non-RT RIC Framework and the Non-RT RIC Applications. Please refer to [8] for more information.

O-CU-CP: O-RAN Central Unit – Control Plane: a logical node hosting RRC [14] and the control plane part of PDCP protocol [15].

O-CU-UP: O-RAN Central Unit – User Plane: a logical node hosting the user plane part of PDCP protocol [15] and SDAP protocol [13].

O-DU: O-RAN Distributed Unit: a logical node hosting RLC [16]/MAC [17]/High-PHY [18] layers based on a lower layer functional split.

O-RU: O-RAN Radio Unit: a logical node hosting Low-PHY layer and RF processing based on a lower layer functional split. This is similar to 3GPP’s “TRP” or “RRH” but more specific in including the Low-PHY layer (FFT/iFFT, PRACH extraction).

O-eNB: An eNB [10] or ng-eNB [11] that supports E2 interface.

O1: An interface between SMO and O-RAN managed elements, for operation and management, by which FCAPS management, PNF (Physical Network Function) Software management, File management shall be achieved.

SMO: A Service Management and Orchestration system as described in [4].

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

A1: An interface between Non-RT RIC and Near-RT RIC to enable policy-driven guidance of Near-RT RIC applications/functions, and support AI/ML workflow. Please refer to [8] for more information.

E2: An interface connecting Near-RT RIC and one or more O-CU-CPs, one or more O-CU-UPs, or one or more O-DUs.

E2 Node: A logical node terminating E2 interface. In this version of the specification, O-RAN nodes terminating E2 interface are:

- for NR access: O-CU-CP, O-CU-UP, O-DU or any combination as defined in [4];
- for E-UTRA access: O-eNB.

xApp: An application designed to run on Near-RT RIC. It may consist of one or more microservices and at the point of on-boarding it identifies which data it consumes and which data it provides. It is independent of Near-RT RIC and may be provided by any third party. The E2 interface enables a direct association between xApp and RAN functionalities.

O-Cloud: O-Cloud is a cloud computing platform comprising a collection of physical infrastructure nodes that meet O-RAN requirements to host the relevant O-RAN functions (such as Near-RT RIC, O-CU-CP, O-CU-UP, and O-DU), the supporting software components (such as Operating System, Virtual Machine Monitor, Container Runtime, etc.) and the appropriate management and orchestration functions.

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply.

API	Application Programming Interface
FM	Fault Management
LCM	Life-Cycle Management
ML	Machine Learning
Non-RT RIC	Non-real-time RAN Intelligent Controller:
Near-RT RIC	Near-real-time RAN Intelligent Controller
O-CU-CP	O-RAN Central Unit – Control Plane
O-CU-UP	O-RAN Central Unit – User Plane
O-DU	O-RAN Distributed Unit
O-RU	O-RAN Radio Unit
PM	Performance Management
R-NIB	Radio-Network Information Base
SCTP	Stream Control Transmission Protocol
SMO	Service Management and Orchestration
UE-NIB	UE-Network Information Base

4 General Principles

The architecture described herein follows the architecture principles specified in [2], and the following principle(s):

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

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

5 Near-RT RIC Architecture

5.1 Requirements

The architecture described herein fulfills the Near-RT RIC requirements specified in [2] and the following requirement(s):

- Near-RT RIC shall consist of multiple xApps and a set of platform functions that are commonly used to support the specific functions hosted by xApps.

5.1.1 Platform Requirements

- Near-RT RIC shall provide a database that stores an up-to-date RAN information, history of time-varying network state, as well as configurations related to E2 Nodes, Cells, Bearers, Flows, UEs, etc., and the mapping between them. This information shall be provided as a service to any xApp that requests it.
- Near-RT RIC shall provide ML tools that support data pipelining.
- Near-RT RIC shall provide a messaging infrastructure.
- Near-RT RIC shall provide logging, tracing and metrics collected from Near-RT RIC platform and xApps toward SMO.
- Near-RT RIC shall provide security functions.
- Near-RT RIC shall support resolution of potential conflicts or overlaps of controls from xApps toward an E2 node.
- Near-RT RIC shall communicate with xApp(s) via Near-RT RIC APIs.
- Near-RT RIC shall register the Near-RT RIC APIs it produces.
- Near-RT RIC shall be capable of discovering the Near-RT RIC APIs it consumes.
- Near-RT RIC shall provide means to resolve compatibility clashes between xApps and the Near-RT RIC services they access.
- Near-RT RIC shall support subscription merging from multiple xApps to avoid unnecessary network load.

5.1.2 xApp Requirements

- xApp may enhance the RRM capabilities of Near-RT RIC.
- xApp may be associated with zero, one or more E2SMs.
- xApp shall use Near-RT RIC APIs to make use of the Information Elements (IEs) of E2SMs that are associated with it.

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

- xApp that is associated with a given E2SM shall be able to interface with any E2 Node that supports that E2SM without any intermediary xApp.
- xApp shall be able to receive event-triggered information on RAN information and time-varying network state.
- xApp shall provide collected logging, tracing and metrics information to Near-RT RIC.
- xApp shall provide a descriptor that includes the following basic information of the xApp:

  - Configuration: It includes a data dictionary for configuration data, i.e., meta data such as a YANG definition or a list of configuration parameters and their semantics. It may also include an initial configuration of xApp.

  - Control: It includes the types of data that an xApp consumes and generates, in order to perform control capabilities (e.g., xApp URL, parameters, input/output type).

  - Metrics: It includes a list of metrics (e.g., metric name, type, unit and semantics) provided by the xApp.
- The xApp descriptor shall also provide the necessary data to enable management and orchestration of the xApp, aligned with [4].
- xApps shall communicate with Near-RT RIC platform via Near-RT RIC APIs.
- xApp shall register the Near-RT RIC APIs it produces.
- xApp shall be capable of discovering the Near-RT RIC APIs they consume.

5.1.3 Near-RT RIC API Requirements

- Near-RT RIC shall provide APIs enabling the hosting of 3rd party xApps and xApps from the Near-RT RIC platform vendor.
- 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.
- Near-RT RIC shall provide APIs decoupled from specific implementation solutions, including a Shared Data Layer (SDL) that works as an overlay for underlying databases and enables simplified data access.
- Near-RT RIC shall provide an API repository/registry for the services provided by the Near-RT RIC platform and/or xApps.
- Near-RT RIC shall provide means for xApps to discover the published APIs based on the xApps' needs.
- Near-RT RIC shall provide means to restrict xApps from discovering some published APIs based on configured policies.
- Near-RT RIC shall provide APIs enabling all xApps to directly use the information elements of E2SMs with which they are associated.
- Near-RT RIC shall provide APIs aiming to simplify the development of xApps and enable rapid innovation.
- Near-RT RIC shall provide Near-RT RIC APIs supporting xApp development in multiple programming languages (e.g. C, C++, Python, Go).
- 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.

NOTE: Communication between xApps is FFS.

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

5.2 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.

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

6 Near-RT RIC Functions Description

6.1 General

Near-RT RIC hosts the following functions:

- Database, which allows reading and writing of RAN/UE information
- xApp subscription management, which merges subscriptions from different xApps and provides unified data distribution to xApps
- Conflict mitigation, which resolves potentially overlapping or conflicting requests from multiple xApps
- Messaging infrastructure, which enables message interaction amongst Near-RT RIC internal functions
- Security, which provides the security scheme for xApps
- Management services:
  - Fault management, configuration management, and performance management as a service producer to SMO;
  - Life-cycle management of xApps;
  - Logging, tracing and metrics collection, which capture, monitor and collect the status of Near-RT RIC internals and can be transferred to external system for further evaluation
- Interface Termination:
  - E2 termination, which terminates E2 interface from an E2 Node;
  - A1 termination, which terminates A1 interface from Non-RT RIC;
  - O1 termination, which terminates O1 interface from SMO
- Functions hosted by xApps, which allow services to be executed at Near-RT RIC and outcomes to be sent to E2 Nodes via E2 interface;
- API enablement function supporting capabilities related to Near-RT RIC API operations (API repository/registry, authentication, discovery, generic event subscription, etc.)

This is summarized in the figure below.

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

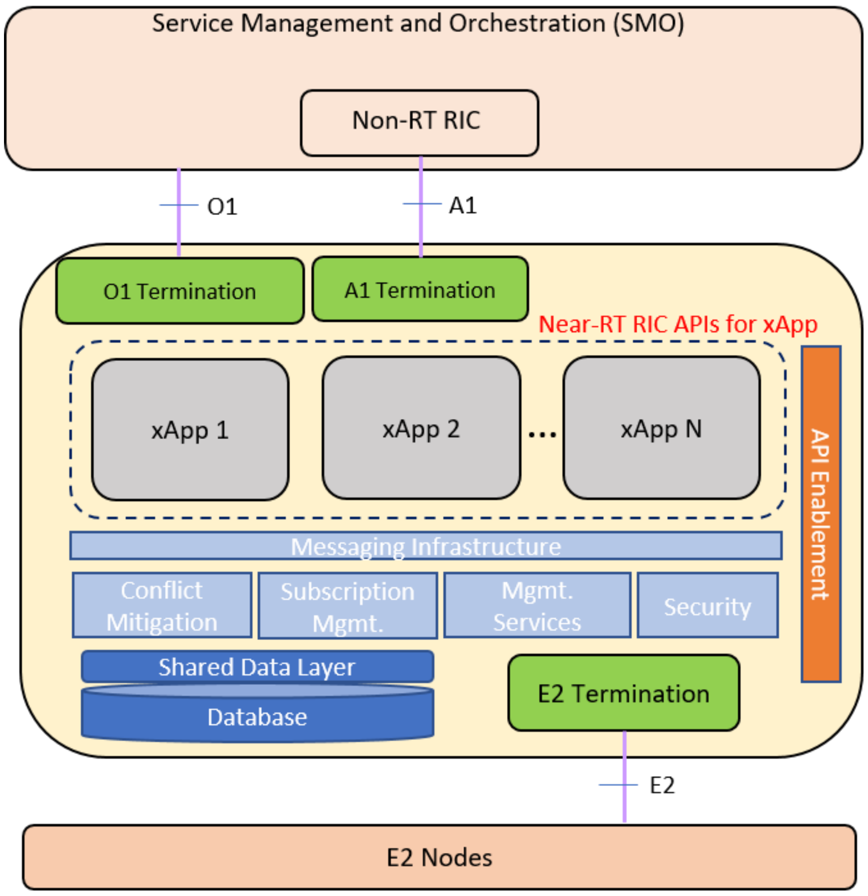


Figure 6.1-1: Near-RT RIC Internal Architecture

6.2 Platform Functions

6.2.1 Database

6.2.1.1 UE-NIB

Some xApps may generate UE related information to be stored in the UE-NIB database.

- UE-NIB maintains a list of UEs and associated data.
- UE-NIB maintains tracking and correlation of the UE identities associated with the connected E2 Nodes.

6.2.1.2 R-NIB

Some xApps may generate radio access network related information to be stored in the R-NIB database.

- The R-NIB stores the configurations and near real-time information relating to connected E2 Nodes and the mappings between them.

6.2.2 xApp Subscription Management

- xApp subscription management manages subscriptions from xApps to E2 Nodes.

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

- xApp subscription management enforces authorization of policies controlling xApp access to messages.
- xApp subscription management enables merging of identical subscriptions from different xApps into a single subscription toward an E2 Node.

6.2.3 Conflict Mitigation

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.

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 are 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.

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

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.

6.2.4 Messaging Infrastructure

Messaging infrastructure provides low-latency message delivery service 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.

6.2.5 Security

The security function given in this section 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 functions is not included in the release.

6.2.6 Management Services

6.2.6.1 Life-Cycle Management of xApp

The life-cycle management of xApps provides the following functions.

- Onboarding xApps: It receives and stores xApp descriptor that contains configuration data for the xApp.

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

- Deployment of xApps: It retrieves xApp name and other information (e.g. helm chart of the xApp) from the stored xApp descriptor and deploys the xApp.
- Resource management (RM): It does comprehensive resource provisioning/control for xApps on Near-RT RIC as well as monitors their latency and resource consumption characteristics to see if individual xApps meet their latency requirements. It may trigger alarm event when they miss the critical latency requirements.
- Termination of xApps: It terminates a running xApp if the xApp is no longer needed. The resource used by the xApp will be released.

NOTE: It is assumed that the life-cycle management of xApps is performed by SMO.

6.2.6.2 FCAPS Management of Near-RT RIC

The FCAPS management consists of fault, configuration, accounting, performance and security management. The FCAPS management follows O1 related management aspects defined in [4].

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

- 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 analyzed 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 functions are collected and published for authorized consumer (e.g., SMO). A metrics collection mechanism is needed to collect and report metrics.

6.2.7 Interface Termination

6.2.7.1 E2 Termination

- E2 Termination terminates SCTP connection from each E2 Node.
- E2 Termination routes messages from xApps through the SCTP connection to an E2 Node.
- E2 Termination decodes the payload of an incoming ASN.1 message enough to determine message type.
- E2 Termination handles incoming E2 messages related to E2 connectivity.
- E2 Termination receives and respond to the E2 Setup Request from an E2 Node.
- E2 Termination notifies 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].
- E2 Termination notifies the newly connected E2 Node of the list of accepted functions.

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

6.2.7.2 A1 Termination

A1 Termination provides 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.

6.2.7.3 O1 Termination

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.

O1 Termination communicates with SMO via O1 interface and exposes O1-related management services [5] from Near-RT RIC.

- O1 Termination exposes provisioning management services from Near-RT RIC to O1 provisioning management service consumer.
- O1 Termination supports managing xApps via NETCONF.
- O1 Termination supports translation of NETCONF to Near-RT RIC internal APIs.
- O1 Termination exposes FM services to report faults and events from Near-RT RIC to O1 FM service consumer.
- O1 Termination exposes PM services to report bulk and real-time PM data from Near-RT RIC to O1 PM service consumer.
- O1 Termination exposes file management services to download ML files, software files, etc. and upload log/trace files.
- O1 Termination exposes communication surveillance services to O1 communication surveillance service consumer.

6.2.8 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 services, and Database services.

API Enablement 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.

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

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.).

6.3 xApps

An xApp consists of xApp descriptor and its image. The image is a software package, and the xApp descriptor describes the packaging format of the image.

The xApp descriptor provides xApp management services with necessary information for the life-cycle management of the xApp, such as deployment, deletion, upgrade etc. The xApp descriptor also provides extra parameters related to the health management of the xApps, such as auto scaling when its load is too heavy or auto healing when it becomes unhealthy. The xApp descriptor provides FCAPS and control parameters when the xApp is launched.

The definition of xApp descriptor includes:

- The basic information of xApp, including name, version, provider, URL of the xApp image, virtual resource requirements (e.g. CPU), etc. This information is used to support life-cycle management of the xApp.
- The FCAPS management specifications that specify the options of configuration, performance metrics collection, etc. for the xApp.
- The control specifications that specify the data types consumed and provided by the xApp for control capabilities (e.g. PM data that the xApp subscribes, the message type of control messages).

The xApp image contains all the files needed to deploy an xApp. An xApp can have multiple versions of images, which are tagged by version numbers.

7 Near-RT RIC APIs

7.1 Overall Description

The Near-RT RIC APIs are a collection of well-defined interfaces providing Near-RT RIC platform 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 as shown in Figure 7.1-1:

- A1 related APIs: APIs allowing to access A1 related functionality.
- E2 related APIs: APIs allowing to access E2 related functionality and associated xApp Subscription Management and Conflict Mitigation functionality.
- Management APIs: APIs allowing to access management related functionality. They are used for services such as O1 Termination, management services and logging, tracing, metrics collection.
- SDL APIs: APIs allowing to access Shared Data Layer related functionality.
- Enablement APIs: APIs between xApps and API enablement functionality.

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

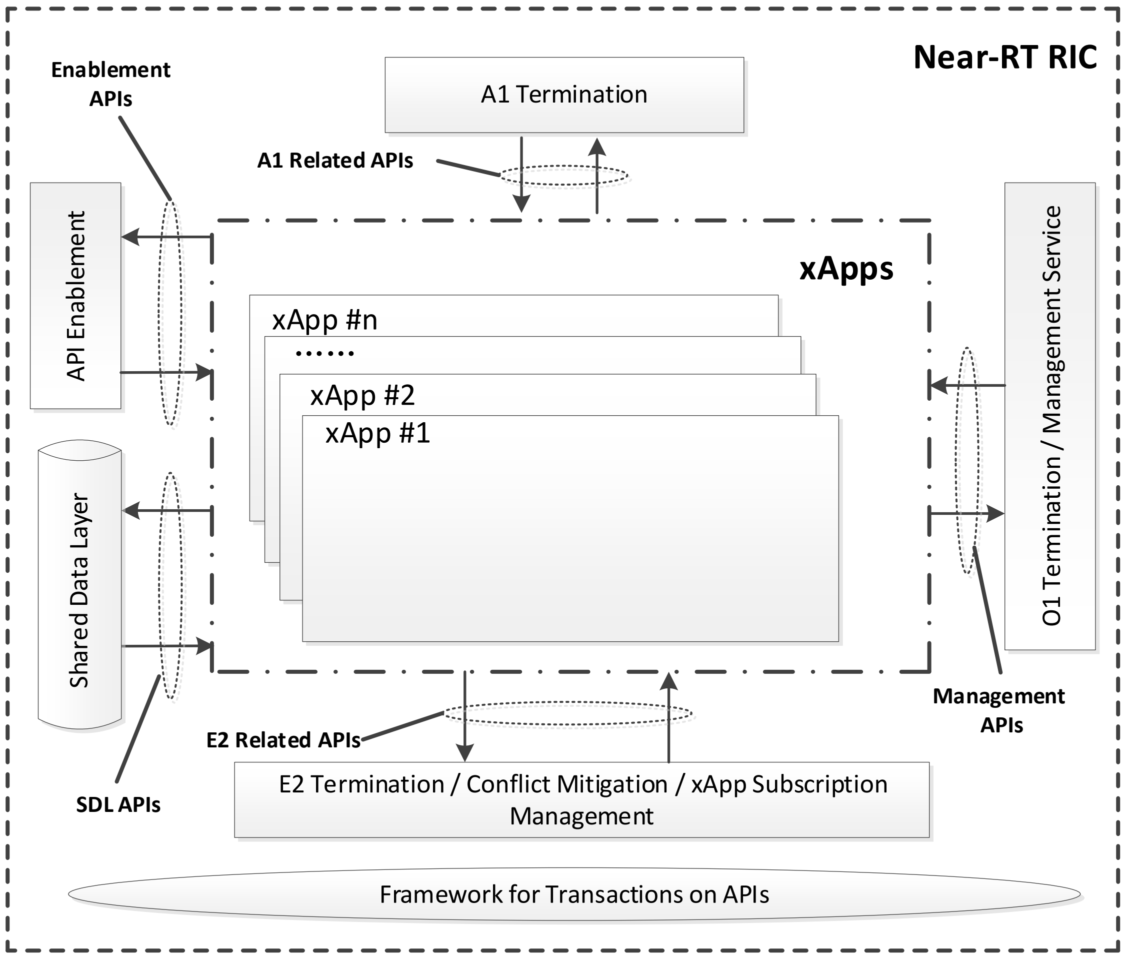


Figure 7.1-1: Overview of Near-RT RIC APIs

7.2 A1 related APIs

The xApps in Near-RT RIC provide value added services based on the policies or enrichment information or both which are transferred through A1 interface by Non-RT RIC. A1 related APIs allow access to A1 related functionality, which includes:

- Policy Enforcement API: used for policy enforcement request/response.
  - A1 Policy Setup (Request, Success/Failure)
  - A1 Policy Update (Request, Success/Failure)
  - A1 Policy Delete (Request, Success/Failure)
- Enrichment Information API: used for enrichment information transfer/response.

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

7.3 E2 related APIs

The xApps in Near-RT RIC provide value added services using RIC functional mechanisms and related procedures via E2 interface towards E2 Nodes. The E2 related APIs allow access to E2 related functionality and the associated xApp Subscription Management and Conflict Mitigation functionality, which includes:

- E2 Subscription (Request, Reject/Success/Failure)
- E2 Subscription Delete (Request, Reject/Success/Failure)
- E2 Indication
- E2 Control (Request, Reject/Success/Failure)
- E2 Guidance (Request, Response, Modification)

NOTE 1: Support for E2 messages related to E2 interface Support functions (E2 SETUP, RIC Service Update, etc.) is FFS.

NOTE 2: Addition Conflict Mitigation related messages are FFS (i.e. for "analysis results").

7.4 Management APIs

The Management APIs include the following APIs, including the xApp’s ML Model related APIs and the FCAPS related APIs.

The xApp’s ML Model related APIs include:

- ML Model Deployment Request.
- ML Model Update Request.
- ML Model Uninstall Request.

The FCAPS related APIs include:

- xApp Registration API: It supports an xApp to register to the Near-RT RIC platform after the xApp is deployed.
- xApp Deregistration API: It supports an xApp to deregister from the Near-RT RIC platform before the xApp is terminated.
- Configuration API: It transfers configurations from SMO to an xApp.
- PM API: It supports an xApp to provide PM related data to the O1 PM Consumer.
- FM API: It supports an xApp to provide faults and events information to the O1 FM Consumer.

NOTE: Trace related API is FFS.

7.5 Void

Void

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

7.6 SDL APIs

The SDL API provides a simple yet flexible way 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. The SDL API allows access to the following Shared Data Layer functionality:

- SDL Client Registration: allows xApps to request the SDL for permissions to access the database.
- SDL Client Deregistration: allows xApps to request the SDL to deregister.
- Fetch Data: allows xApps to fetch data from the database.
- Subscribe: allows xApps to subscribe to notifications related to updates of data.
- Notify: SDL can notify xApps about updates of information in the database. When an update occurs, the SDL will send notifications to those xApps that are subscribed to receive them.
- Store Data: allows xApps to store data in the database.
- Modify Data: allows xApps to modify or delete data from the database.

7.7 Enablement APIs

The Enablement APIs support exchange of API enablement related information between xApps and the API enablement functionality.

The API enablement information includes:

- API Registration: enabling the API producer to register the APIs it produces.

NOTE: An investigation of services for which the API producer is the xApp is FFS.

- API Discovery: enabling discovery of the Near-RT RIC APIs by xApps.
- API-related Event Subscription: enabling subscription/un-subscription to API-related events and event notifications.

8 External Interfaces of Near-RT RIC

8.1 E2 Interface

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

O-RAN-WG3.E2AP [3] specifies E2 interface application protocols.

8.2 A1 Interface

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

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

O-RAN-WG2.A1AP [9] specifies A1 interface application protocols.

8.3 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.

9 Near-RT RIC API Procedures

9.1 A1 related API Procedures

9.1.1 A1 Policy Setup procedure

The purpose of A1 Policy Setup procedure in Near-RT RIC is to request and activate an A1 policy enforcement toward a suitable xApp.

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

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 a A1 policy. - xApp: responsible for the enforcement of accommodated A1 policies. - Near-RT RIC Platform ○ A1 Termination: originator of A1 related API request to activate a A1 policy. ○ [Informative] xApp Repository Function: handling of A1 policy to xApp accommodation.
Assumptions	- Near-RT RIC supported A1 policy types is accessible by A1 termination. - Near-RT RIC nested A1 policies is accessible by A1 termination. - [Informative] xApp selection needed data is accessible by xApp Repository Function. xApp selection process is performed at xApp Repository Function.
Pre conditions	- Near-RT RIC is running normal. - 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)	A1 Termination checks whether the received policy type is supported in Near-RT RIC or not.
Step 3 (M)	If the received policy type is not supported in Near-RT RIC, a “400” response is directly returned to Non-RT RIC.
Step 4 (M)	If the received policy type is supported in Near-RT RIC, A1 Termination then checks whether the received A1 policy is mapped to any xApp or not.
Step 5 [Informative]	[Note] If the received A1 policy is not mapped to any xApp, A1 Termination triggers xApp selection procedure to identify a suitable xApp by sending xAPP SELECTION REQUEST to xApp Repository Function. The message includes at least A1 policy received from Non-RT RIC.
Step 6 [Informative]	xApp selection process is performed at xApp Repository Function.
Step 7 [Informative]	If suitable xApps are identified, candidate xApps are notified to A1 Termination.
Step 8(M)	A1 Termination selects a suitable one from candidate xApps 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.
Step 9~10(M)	If the selected xApp accomplishes setup of the A1 policy. A success is notified to A1 termination in **A1 related API: A1 POLICY SETUP RESULT** message, and eventually to Non-RT RIC.
Step 11~12(M)	If the selected xApp fails to set up the A1 policy. A failure is notified to A1 termination in **A1 related API: A1 POLICY SETUP RESULT** message, and eventually to Non-RT RIC.
Step 13~14(M)	If no suitable xApp is identified in step 6, a failure is notified to A1 termination [Informative]. A “503” response is directly returned to Non-RT RIC.
NOTE: As the create policy and update policy procedure in A1 interface has identical HTTP PUT Request, so when a HTTP PUT Request is received, A1 Termination cannot decide whether it’s asked for create or update from a first glance. Only after further 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, A1 Termination triggers A1 Policy Setup procedure as shown here. Otherwise, A1 Termination triggers A1 Policy Update procedure. See 9.1.2 for details.	

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

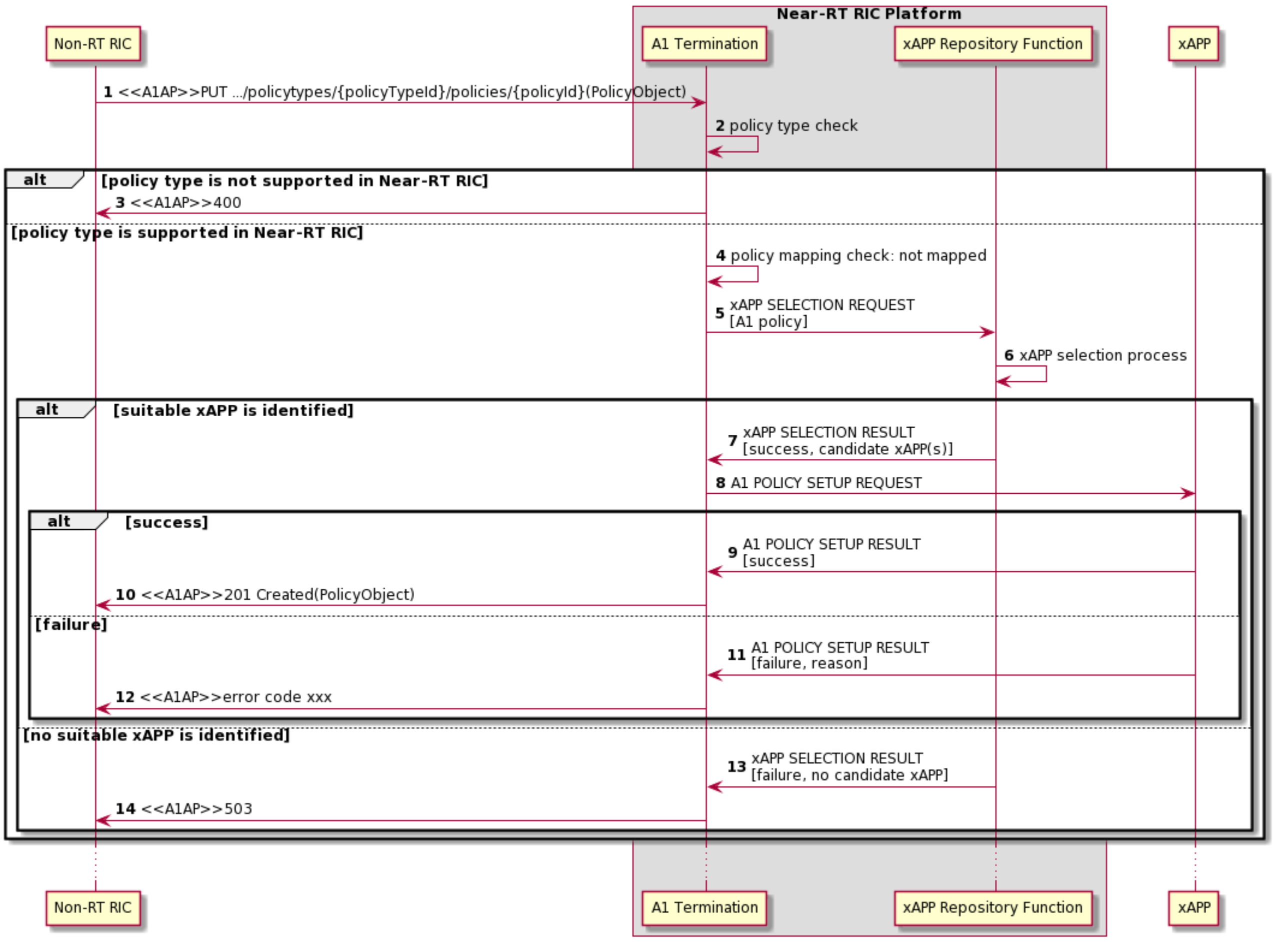

Near-RT RIC Platform
Non-RT RIC
A1 Termination
xAPP Repository Function
xAPP
1 <<A1AP>>PUT .../policytypes/{policyTypeId}/policies/{policyId}(PolicyObject)
2 policy type check
alt [policy type is not supported in Near-RT RIC]
3 <<A1AP>>400
[policy type is supported in Near-RT RIC]
4 policy mapping check: not mapped
5 xAPP SELECTION REQUEST [A1 policy]
6 xAPP selection process
alt [suitable xAPP is identified]
7 xAPP SELECTION RESULT [success, candidate xAPP(s)]
8 A1 POLICY SETUP REQUEST
alt [success]
9 A1 POLICY SETUP RESULT [success]
10 <<A1AP>>201 Created(PolicyObject)
[failure]
11 A1 POLICY SETUP RESULT [failure, reason]
12 <<A1AP>>error code xxx
[no suitable xAPP is identified]
13 xAPP SELECTION RESULT [failure, no candidate xAPP]
14 <<A1AP>>503

Figure 9.1.1-1 A1 Policy Setup procedure

9.1.2 A1 Policy Update procedure

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

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

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 a A1 policy. - xApp: responsible for the enforcement of accommodated A1 policies. - Near-RT RIC Platform o A1 Termination: originator of A1 related API request to update a A1 policy.
Assumptions	- Near-RT RIC supported A1 policy types is accessible by A1 termination. - Near-RT RIC nested A1 policies is accessible by A1 termination.
Pre conditions	- Near-RT RIC is running normal. - 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)	A1 Termination checks whether the received policy type is supported in Near-RT RIC or not.
Step 3 (M)	If the received policy type is not supported in Near-RT RIC, a “400” response is directly returned to Non-RT RIC.
Step 4 (M)	If the received policy type is supported in Near-RT RIC, A1 Termination then checks whether the received A1 policy is mapped to any xApp or not.
Step 5 (M)	[Note] If the received A1 policy is already mapped to one xApp, A1 Termination selects which 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.
Step 6~7(M)	If the target xApp accomplishes update of the A1 policy. A success is notified to A1 termination in **A1 related API: A1 POLICY UPDATE RESULT** message, and eventually to Non-RT RIC.
Step 8~9(M)	If the target xApp fails to update the A1 policy. A failure is notified to A1 termination in **A1 related API: A1 POLICY UPDATE RESULT** message, and eventually to Non-RT RIC.
NOTE: As the create policy and update policy procedure in A1 interface has identical HTTP PUT Request, so when a HTTP PUT Request is received, A1 Termination cannot decide whether it’s asked for create or update from a first glance. Only after further 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, A1 Termination triggers A1 Policy Update procedure as shown here. Otherwise, A1 Termination triggers A1 Policy Setup procedure. See 9.1.1 for details.	

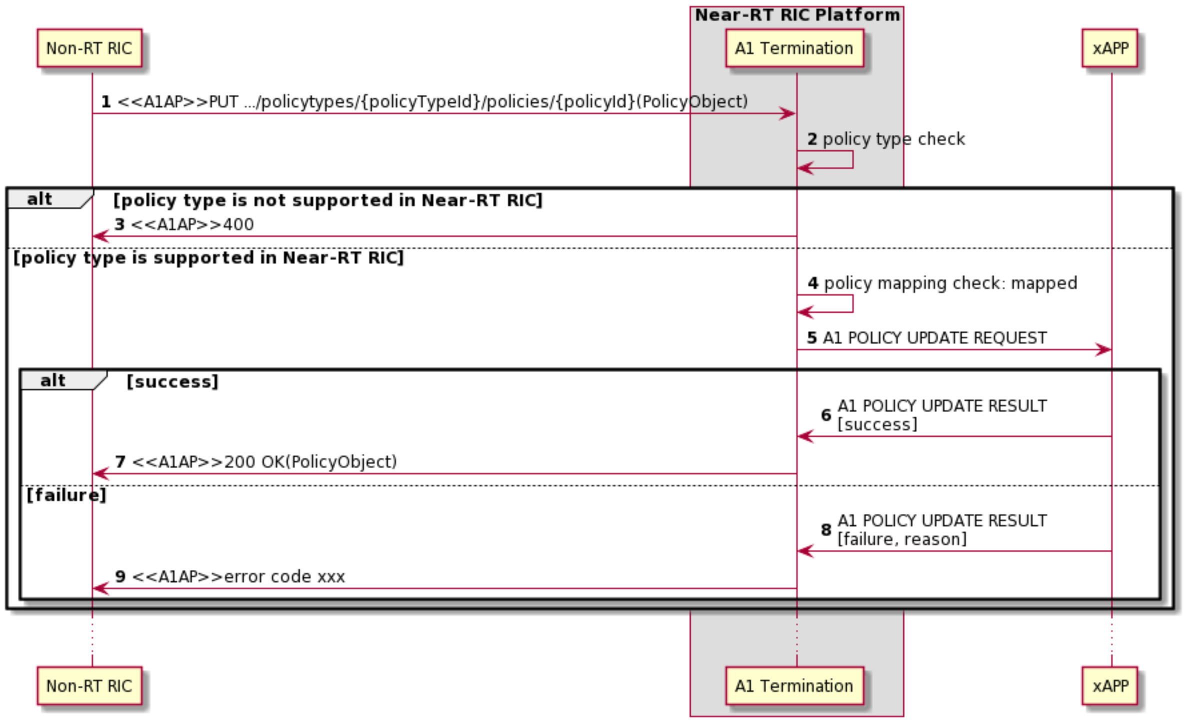

Near-RT RIC Platform
Non-RT RIC
A1 Termination
xAPP
1 <<A1AP>>PUT .../policytypes/{policyTypeId}/policies/{policyId}(PolicyObject)
2 policy type check
alt [policy type is not supported in Near-RT RIC]
3 <<A1AP>>400
[policy type is supported in Near-RT RIC]
4 policy mapping check: mapped
5 A1 POLICY UPDATE REQUEST
alt [success]
6 A1 POLICY UPDATE RESULT [success]
7 <<A1AP>>200 OK(PolicyObject)
[failure]
8 A1 POLICY UPDATE RESULT [failure, reason]
9 <<A1AP>>error code xxx

Figure 9.1.2-1 A1 Policy Update procedure

9.1.3 A1 Policy Delete procedure

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

Use Case Stage	Evolution / Specification
Goal	The deactivation of a received and already activated A1 policy in Near-RT RIC.

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

Actors and Roles	- Non-RT RIC: originator of HTTP DELETE request to deactivate a A1 policy. - xApp: responsible for the deactivation of nested A1 policies. - Near-RT RIC Platform o A1 Termination: originator of A1 related API request to delete a A1 policy.
Assumptions	- Near-RT RIC supported A1 policy types is accessible by A1 termination. - Near-RT RIC nested A1 policies is accessible by A1 termination.
Pre conditions	- Near-RT RIC is running normal. - 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)	A1 Termination checks whether the received policy type is supported in Near-RT RIC or not.
Step 3 (M)	If the received policy type is not supported in Near-RT RIC, a “400” response is directly returned to Non-RT RIC.
Step 4 (M)	If the received policy type is supported in Near-RT RIC, A1 Termination then checks whether the received A1 policy is mapped to any xApp or not.
Step 5 (M)	If the received A1 policy is already mapped to one xApp, A1 Termination selects which 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.
Step 6~7(M)	If the target xApp accomplishes delete of the A1 policy. A success is notified to A1 termination in **A1 related API: A1 POLICY DELETE RESULT** message, and eventually to Non-RT RIC.
Step 8~9(M)	If the target xApp fails to delete the A1 policy. A failure is notified to A1 termination in **A1 related API: A1 POLICY DELETE RESULT** message, and eventually to Non-RT RIC.
Step 10(M)	If the received A1 policy is not mapped to any xApp, a “503” response is directly returned to Non-RT RIC.

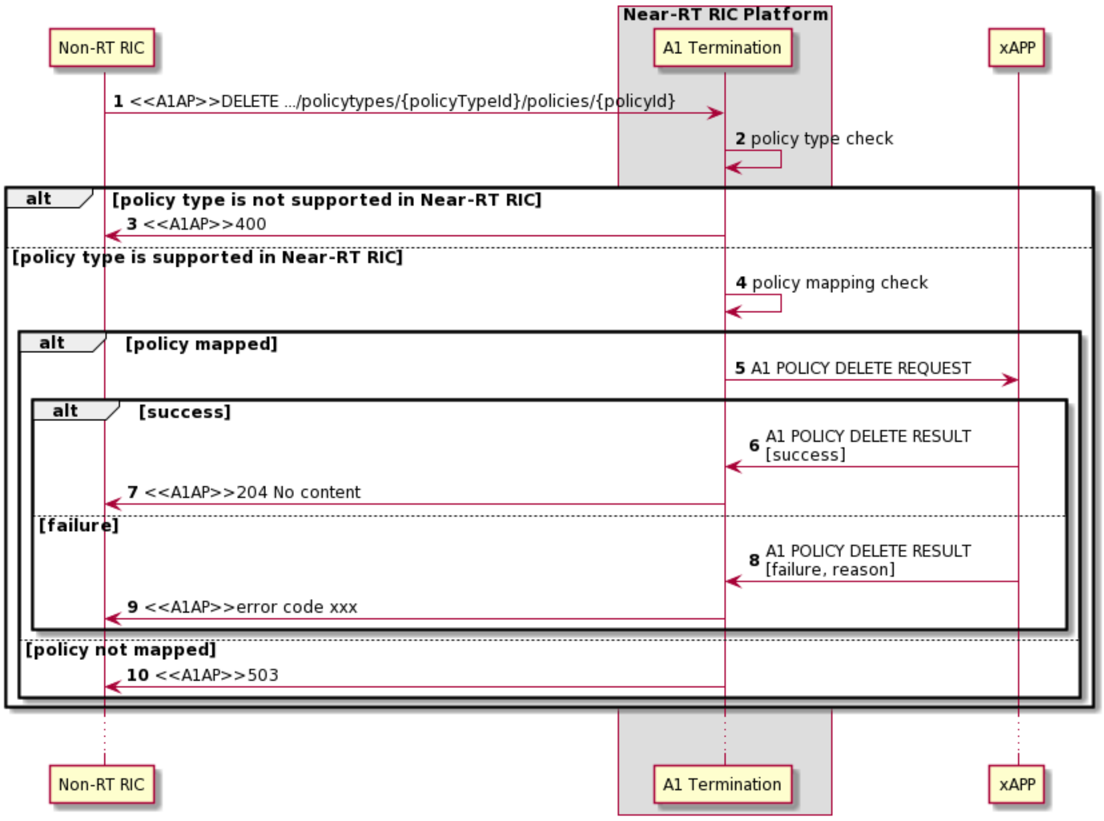

Near-RT RIC Platform
Non-RT RIC
A1 Termination
xAPP
1 <<A1AP>>DELETE .../policytypes/{policyTypeId}/policies/{policyId}
2 policy type check
alt
[policy type is not supported in Near-RT RIC]
3 <<A1AP>>400
[policy type is supported in Near-RT RIC]
4 policy mapping check
alt
[policy mapped]
5 A1 POLICY DELETE REQUEST
alt
[success]
6 A1 POLICY DELETE RESULT [success]
7 <<A1AP>>204 No content
[failure]
8 A1 POLICY DELETE RESULT [failure, reason]
9 <<A1AP>>error code xxx
[policy not mapped]
10 <<A1AP>>503

Figure 9.1.3-1 A1 Policy Delete procedure

9.2 E2 Related API Procedures

9.2.1 Introduction

Procedures described in this section assume the Near-RT RIC Platform contains the functions listed in section 6.2.

Termination points within the Platform Functions are for information and may vary depending on implementation.

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

The following procedures are described in this section:

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 Indication using **E2 Related API: E2 Indication**
- RIC Control using **E2 Related API: E2 Control** (request, reject, success, failure) and **E2 Related API: E2 Guidance** (request, response)

E2 Guidance API 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)

Other procedures related to E2AP Global support functions (E2 SETUP, RESET, etc.) and other conflict mitigation related procedures are for future study.

9.2.2 RIC functional procedures

9.2.2.1 E2 Subscription API procedure

The purpose of the E2 Subscription API procedure in the Near-RT RIC is to ensure that only validated and non-duplicate RIC Subscription Request messages are issued 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) to resolve potential conflicts and/or detect partial duplications prior to sending a E2 Subscription Request (see section 9.2.3.1)
- xApp has been configured with a trusted xApp ID
- E2 related API routes E2 Subscription Request messages for one or more E2 nodes from xApp towards an appropriate Near-RT RIC platform xApp Subscription Management function instance
- xApp Subscription Management may recover from a platform database the list of all previous successful RIC Subscriptions towards E2 Nodes listed in xApp candidate RIC Subscription request and is able to detect duplications and recover E2 Node response messages
- xApp Subscription Management may obtain guidance from Conflict Mitigation (see section 9.2.3.2)
- xApp Subscription Management is able to generate a unique REQUEST ID for each RIC Subscription request to each listed E2 Node
- xApp Subscription Management routes RIC Subscription Request messages to appropriate E2 termination instance

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

- E2 Termination systematically forwards any received RIC Subscription Response or RIC Subscription Failure to appropriate xApp Subscription Management instance
- xApp Subscription management maintains a mapping of active Subscriptions (identified by E2 Node ID and REQUEST ID) of validated xApps. E2 Termination uses the mapping when sending RIC INDICATION messages to the correct xApp or xApps

The procedure is initiated by an xApp using **E2 Related API: E2 Subscription** request to send a request of a RIC Subscription proposed for a list of E2 Nodes. 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 for a list of proposed E2 node targets rejected 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 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 transaction
- Request towards a specific E2 Node fails following acceptance by Near-RT RIC Platform but rejected by 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)

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

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 assumed to consist of: o Database o xApp Subscription Management o Conflict Mitigation o E2 Termination - E2 Node: RIC Subscription procedure
Assumptions	- xApp Subscription Management in Near-RT RIC platform has access to platform database - For E2 Subscription API procedures, xApp Subscription Management is logically placed between xApp and E2 Termination - Conflict Mitigation has access to sufficient information to both detect a potential conflict and take a decision on an optimal mitigation solution - Conflict Mitigation may initiate guidance towards other Platform Functions 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 Conflict Mitigation - xApp Subscription Management has been configured to permit E2 Subscription API requests from specific list of xApp - E2 Termination has been configured to accept incoming RIC Subscription requests from xApp Subscription Management
Begins when	xApp determines need to propose E2 Subscription API request for a list of 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 Conflict Mitigation (see section 9.2.3.1 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 list of E2 Nodes
Step 3 [Informative]	xApp Subscription Management, after optional guidance from Conflict Mitigation (see section 9.2.3.2 for details), accepts or rejects proposal from xApp for list of E2 nodes
Step 4 (M)	<u>If reject</u>, xApp Subscription Management sends **E2 related API: E2 Subscription** response to xApp (Reject with Cause for one or more E2 node in proposal list)
	<u>Loop for one or more E2 Node in list</u> of accepted RIC Subscription proposals
Step 5 [Informative]	xApp Subscription Management retrieves from a platform database the list of xApp associated with subscription for E2 Node target, RAN Function Id, Event Trigger, Action List.
Step 6 [Informative]	xApp Subscription Management counts the number of retrieved records to determine whether the Subscription proposed by xApp is a new subscription or duplicate with an existing subscription
Step 7 [Informative]	For the case that the Subscription proposed by xApp is a new subscription (i.e. no record found for E2 Node ID with Subscription matching the same contents), xApp Subscription Management assigns a unique REQUEST ID.
Step 8 [Informative]	xApp Subscription Management sends RIC Subscription Request to E2 Termination, providing E2 Node ID and REQUEST ID along with message contents proposed by xApp.
Step 9 (M)	E2 Termination selects appropriate E2 interface and sends **RIC Subscription Request** (REQUEST ID, contents) to E2 Node
Step 10 (M)	<u>If E2 Node accepts</u> Subscription then E2 Node responds with **RIC Subscription Response**.
Step 11 [Informative]	E2 Termination forwards received message to xApp Subscription Management.
Step 12 [Informative]	For each Action in subscription corresponding to an Indication message, xApp Subscription Management requests E2 termination to add xApp ID to distribution list associated with E2 Node ID, REQUEST ID and Action ID.
Step 13 [Informative]	xApp Subscription Management creates corresponding record in the platform database.
Step 14 (M)	xApp Subscription Management sends **E2 related API: E2 Subscription** response to xApp (Success, E2 Node Id)
Step 15 (M)	<u>If E2 Node rejects</u> Subscription then E2 Node responds with **RIC Subscription Failure** (with Cause).
Step 16 [Informative]	E2 Termination forwards received message to xApp Subscription Management.
Step 17 (M)	xApp Subscription Management sends **E2 related API: E2 Subscription** response to xApp (Failure with Cause, E2 node Id)

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

Step 18 [Informative]	For the case that the E2 Subscription API Request proposed by xApp is a duplicate subscription (i.e. one or more records found for E2 Node ID with Subscription matching the same contents), xApp Subscription Management extracts REQUEST ID from record.
Step 19 [Informative]	For each Action in subscription corresponding to an Indication message, xApp Subscription Management requests E2 termination to add xApp ID to distribution list associated with E2 Node ID, REQUEST ID and Action ID.
Step 20 [Informative]	xApp Subscription Management creates corresponding record in the platform database.
Step 21 (M)	xApp Subscription Management sends **E2 Related API: E2 Subscription** response to xApp (Success, E2 node Id)

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

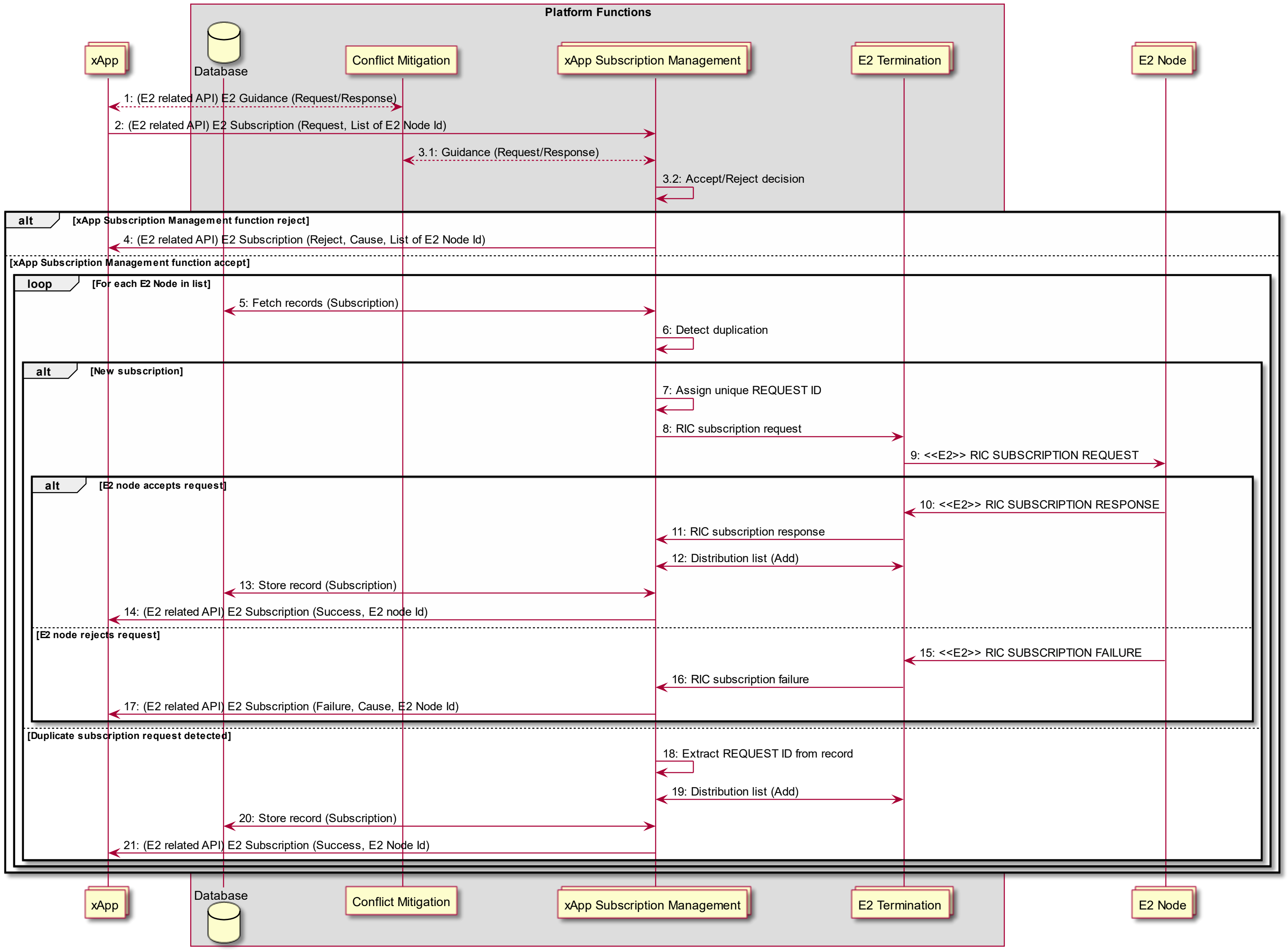

Platform Functions
xApp
Database
Conflict Mitigation
xApp Subscription Management
E2 Termination
E2 Node
1: (E2 related API) E2 Guidance (Request/Response)
2: (E2 related API) E2 Subscription (Request, List of E2 Node Id)
3.1: Guidance (Request/Response)
3.2: Accept/Reject decision
alt [xApp Subscription Management function reject]
4: (E2 related API) E2 Subscription (Reject, Cause, List of E2 Node Id)
[xApp Subscription Management function accept]
loop [For each E2 Node in list]
5: Fetch records (Subscription)
6: Detect duplication
alt [New subscription]
7: Assign unique REQUEST ID
8: RIC subscription request
9: <<E2>> RIC SUBSCRIPTION REQUEST
alt [E2 node accepts request]
10: <<E2>> RIC SUBSCRIPTION RESPONSE
11: RIC subscription response
12: Distribution list (Add)
13: Store record (Subscription)
14: (E2 related API) E2 Subscription (Success, E2 node Id)
[E2 node rejects request]
15: <<E2>> RIC SUBSCRIPTION FAILURE
16: RIC subscription failure
17: (E2 related API) E2 Subscription (Failure, Cause, E2 Node Id)
[Duplicate subscription request detected]
18: Extract REQUEST ID from record
19: Distribution list (Add)
20: Store record (Subscription)
21: (E2 related API) E2 Subscription (Success, E2 Node Id)

Figure 9.2.1-1: E2 Subscription API procedure

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

9.2.2.2 E2 Subscription Delete API Procedure

The purpose of the E2 Subscription Delete API 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 and b) that deletion requests for duplicated Subscription Requests are handled gracefully.

This procedure is based on the following assumptions:

- E2 related API routes RIC Subscription Delete Requests for one or more E2 nodes from xApp towards an appropriate xApp Subscription Management instance
- xApp has been configured with a trusted xApp ID
- xApp Subscription Management may recover from the platform database the list of all previous successful RIC Subscriptions towards E2 Nodes listed in xApp candidate RIC Subscription Delete Request and is able to detect when and if the delete request shall result in a corresponding E2 message
- xApp Subscription Management routes RIC Subscription Delete Request messages to appropriate E2 termination instance
- E2 Termination systematically forwards any received RIC Subscription Delete Response or RIC Subscription Delete Failure to appropriate xApp Subscription Management instance
- xApp Subscription management maintains a mapping of active Subscriptions (identified by E2 Node ID and REQUEST ID) of validated xApps. E2 Termination uses the mapping when sending RIC INDICATION messages to the correct xApp or xApps

The procedure is initiated by an xApp using **E2 Related API: E2 Subscription Delete** request to send a request of a RIC Subscription Delete proposed for a list of E2 Nodes. The following outcomes are considered:

- Request fails following rejection by xApp Subscription Management and an **E2 Related API: E2 Subscription Delete** response (Failure) is sent to xApp without E2 interface transaction (i.e. xApp not previously associated with validated subscription)
- Request successful following acceptance by xApp Subscription Management but detected as duplicate and so an **E2 Related API: E2 Subscription Delete** response (Success) is sent to xApp without corresponding E2 transaction

  Request successful following acceptance by xApp Subscription Management and accepted by E2 Node and so an **E2 Related API: E2 Subscription Delete** response (Success) is sent to xApp following corresponding E2 transaction
- Request fails following acceptance by xApp Subscription Management but rejected by E2 Node and so an **E2 Related API: E2 Subscription Delete** response (Failure) is sent to xApp following corresponding E2 interface transaction (i.e. REQUEST ID is not known by E2 Node)

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

Use Case Stage	Evolution / Specification
Goal	E2 Subscription Delete API procedure from xApp initiation to E2 Node and response
Actors and Roles	- xApp: Originator of E2 Subscription Delete API request - Near-RT RIC Platform o Database o xApp Subscription Management o E2 Termination - E2 Node: RIC Subscription Delete procedure
Assumptions	- xApp Subscription Management in Near-RT RIC has access to platform database - For E2 Subscription Delete API procedures, xApp Subscription Management is logically placed between xApp and E2 Termination
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 Subscription Management has been configured to permit RIC Subscription Delete requests from xApp - E2 Termination has been configured to accept incoming RIC subscription delete requests from xApp Subscription Management
Begins when	xApp determines need to request E2 Subscription Delete API for a list of (E2 Node ID / REQUEST ID) pairs
Step 1 (M)	xApp sends **E2 related API: E2 Subscription Delete** request for a list of E2 Nodes.
Step 2 [Informative]	xApp Subscription Management accepts or rejects proposal from xApp
Step 3 (M)	<u>If reject</u>, xApp Subscription Management sends **E2 related API: E2 Subscription Delete** response to xApp (Reject with Cause for one or more E2 node in proposal list)
	Loop for one or more E2 Node in list of accepted RIC Subscription Delete proposals
Step 4 [Informative]	xApp Subscription Management retrieves list of xApp associated with subscription for E2 Node and Request ID.
Step 5 [Informative]	xApp subscription Management confirms that xApp proposing Subscription Delete has an active subscription
Step 6 (M)	xApp Subscription Management sends **E2 related API: E2 Subscription Delete** response to xApp (Reject with Cause for E2 node)
Step 7 [Informative]	xApp Subscription Management counts the number of retrieved records to determine whether or not xApp proposing Subscription Delete is the only associated xApp
Step 8 [Informative]	For the case that xApp proposing Subscription Delete is the only xApp associated with the subscription (i.e. a given REQUEST ID) xApp Subscription Management sends RIC Subscription Delete Request to E2 Termination, providing E2 Node ID and REQUEST ID
Step 9 (M)	E2 Termination selects appropriate E2 interface and sends **RIC Subscription Delete Request** (REQUEST ID) to E2 Node
Step 10 (M)	<u>If E2 Node accepts</u> Subscription Delete then E2 Node responds with **RIC Subscription Delete Response**
Step 11 [Informative]	E2 Termination forwards received message to xApp Subscription Management.
Step 12 [Informative]	xApp Subscription Management requests E2 termination to delete distribution list(s) associated with E2 Node ID and REQUEST ID.
Step 13 [Informative]	xApp Subscription Management deletes corresponding record in the platform database.
Step 14 (M)	xApp Subscription Management sends **RIC API: E2 related E2 Subscription Delete** response to xApp (Success)
Step 15 (M)	<u>If E2 Node rejects</u> Subscription Delete then E2 Node responses with **RIC Subscription Delete Failure** (with Cause):
Step 16 [Informative]	E2 Termination forwards received message to xApp Subscription Management.
Step 17 (M)	xApp Subscription Management sends **E2 related API: E2 Subscription Delete** response to xApp (Failure with Cause)
Step 18 [Informative]	For the case that xApp proposing Subscription Delete is one of a number of xApp associated with the subscription (i.e. a given REQUEST ID) xApp Subscription Management requests E2 termination to remove xApp from distribution list(s) associated with E2 Node ID and REQUEST ID.

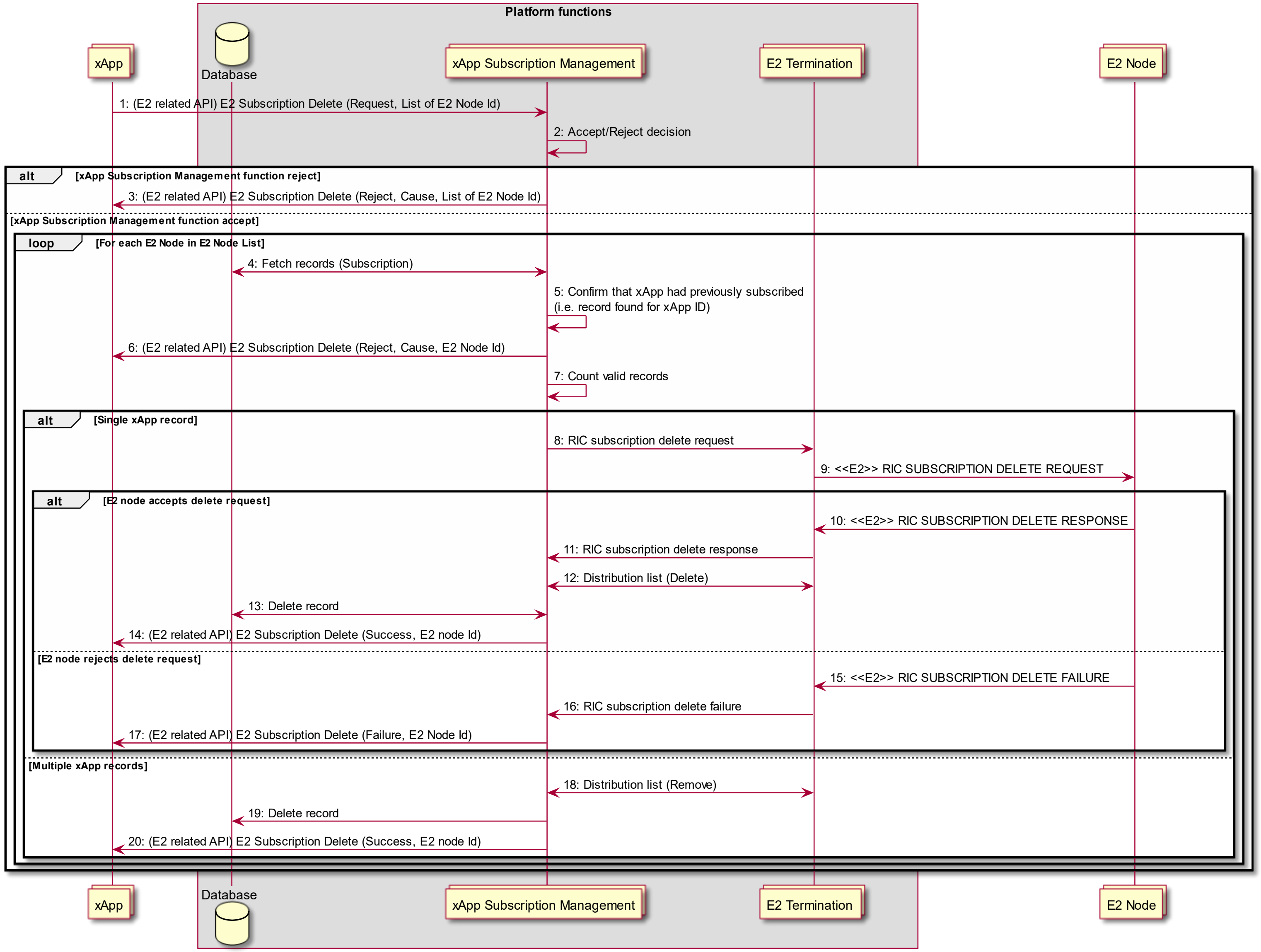
Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

Step 19 [Informative]	xApp Subscription Management deletes corresponding record in the platform database.
Step 20 (M)	xApp Subscription Management sends **E2 related API: E2 Subscription Delete** response to xApp (Success)

1

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

1


Platform functions
xApp
Database
xApp Subscription Management
E2 Termination
E2 Node
1: (E2 related API) E2 Subscription Delete (Request, List of E2 Node Id)
2: Accept/Reject decision
alt
[xApp Subscription Management function reject]
3: (E2 related API) E2 Subscription Delete (Reject, Cause, List of E2 Node Id)
[xApp Subscription Management function accept]
loop
[For each E2 Node in E2 Node List]
4: Fetch records (Subscription)
5: Confirm that xApp had previously subscribed (i.e. record found for xApp ID)
6: (E2 related API) E2 Subscription Delete (Reject, Cause, E2 Node Id)
7: Count valid records
alt
[Single xApp record]
8: RIC subscription delete request
9: <<E2>> RIC SUBSCRIPTION DELETE REQUEST
alt
[E2 node accepts delete request]
10: <<E2>> RIC SUBSCRIPTION DELETE RESPONSE
11: RIC subscription delete response
12: Distribution list (Delete)
13: Delete record
14: (E2 related API) E2 Subscription Delete (Success, E2 node Id)
[E2 node rejects delete request]
15: <<E2>> RIC SUBSCRIPTION DELETE FAILURE
16: RIC subscription delete failure
17: (E2 related API) E2 Subscription Delete (Failure, E2 Node Id)
[Multiple xApp records]
18: Distribution list (Remove)
19: Delete record
20: (E2 related API) E2 Subscription Delete (Success, E2 node Id)

2

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

Figure 9.2.2-1: E2 Subscription Delete API procedure

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

9.2.2.3 E2 Indication API Procedure

The purpose of the E2 Indication API 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:

- E2 Termination maintains a list of validated xApps associated with an E2 node ID and REQUEST ID
- E2 related API ensures delivery of RIC Indication messages from E2 termination to all validated xApp
- Messaging infrastructure may support distribution of messages to multiple destinations

Use Case Stage	Evolution / Specification
Goal	E2 Indication API procedure from E2 Node initiation to xApp reception
Actors and Roles	- xApp: Originator of RIC subscription request - Near-RT RIC Platform ○ Database ○ xApp Subscription Management ○ E2 Termination - E2 Node: Originator of RIC Indication procedure
Assumptions	- For RIC Indication procedures, E2 Termination 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 - E2 Termination has distribution list associating Indication messages with xApps
Begins when	E2 Node creates RIC Indication Message (REQUEST ID, RAN Function Id, Action ID, Sequence number, Indication Type, Indication Header, Indication Message, optional Call Process ID)
Step 1 (M)	E2 Node sends **RIC Indication** message to E2 termination
Step 2 [Informative]	If not available, E2 Termination fetches distribution list from platform database for E2 node Id, REQUEST ID, RAN Function Id, Action ID
Step 3 (M)	E2 Termination forwards Indication message to associated xApps using either: a) If messaging infrastructure in Near-RT RIC supports distribution of messages to list of destinations, E2 Termination sends **E2 related API: E2 Indication** to list of xApps b) If messaging infrastructure in Near-RT RIC only supports distribution of messages to single destinations, E2 Termination sends **E2 related API: E2 Indication** to each xApp in list

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

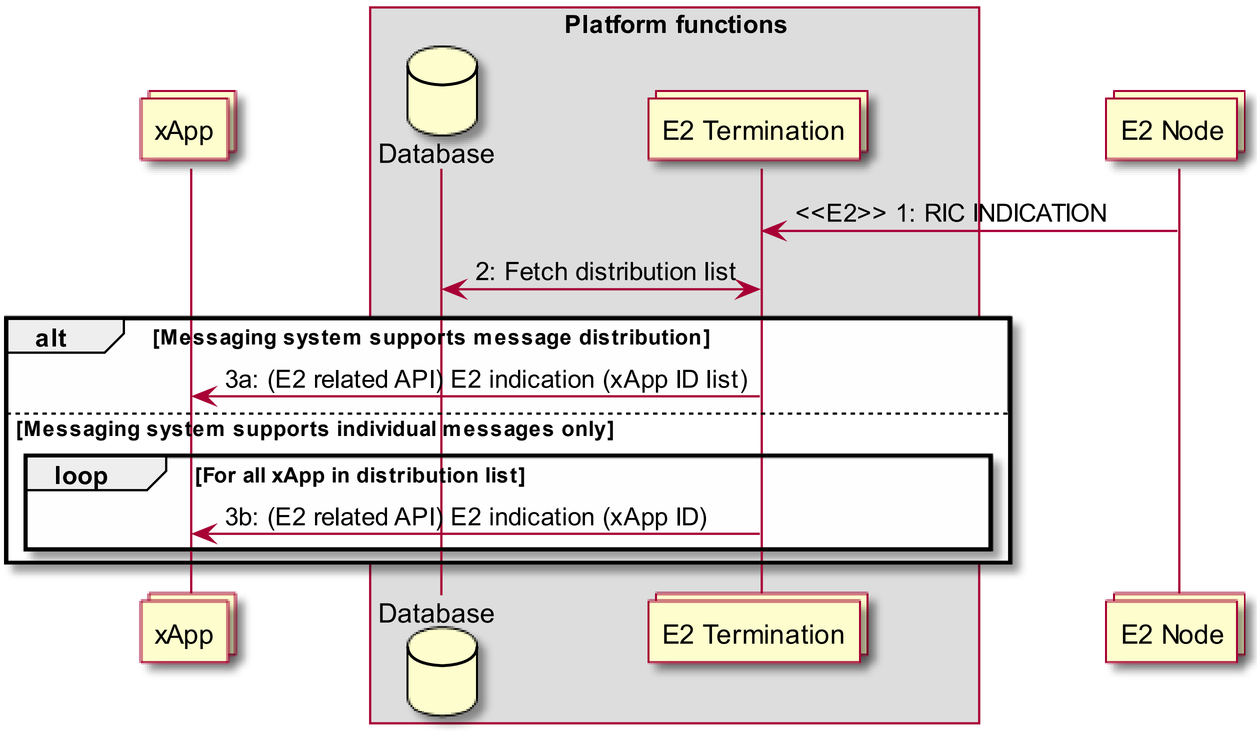

Platform functions
xApp
Database
E2 Termination
E2 Node
<<E2>> 1: RIC INDICATION
2: Fetch distribution list
alt
[Messaging system supports message distribution]
3a: (E2 related API) E2 indication (xApp ID list)
[Messaging system supports individual messages only]
loop
[For all xApp in distribution list]
3b: (E2 related API) E2 indication (xApp ID)
xApp
Database
E2 Termination
E2 Node

Figure 9.2.3-1: E2 Indication API procedure

9.2.2.4 E2 Control API Procedure

The purpose of the E2 Control API 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 Conflict Mitigation to resolve potential conflicts prior to sending a E2 Control API Request (see section 9.2.3.1)
- xApp has been configured with a trusted xApp ID
- E2 related API may be configured to route E2 Control API Request messages for a E2 node from xApp either towards Conflict Mitigation for acceptance (see section 9.2.3.4) or directly towards an appropriate E2 Termination instance
- E2 related API ensures that only authorized xApp may send E2 Control API Request messages to appropriate E2 Termination instance
- E2 related API ensures that E2 Termination 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 rejected by 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)
- Request rejected by Near-RT Platform and so an **E2 Related API: E2 Control** response (Reject) is sent to xApp (i.e. rejected by Conflict Mitigation)

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

Use Case Stage	Evolution / Specification
Goal	E2 Control API procedure from xApp initiation to E2 Node and response
Actors and Roles	- xApp: Originator of E2 Control API request - Near-RT RIC Platform o E2 Termination - E2 Node: RIC Control procedure
Assumptions	- For E2 Control API procedures, xApp may directly send messages to E2 Termination - Conflict Mitigation 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 Conflict Mitigation - E2 Termination and E2 related API has been configured to accept incoming E2 Control API requests from authorized xApp and forward either to Conflict Mitigation or directly to E2 Termination
Begins when	xApp determines need to propose RIC Control procedure for a E2 Node and defines message contents (REQUEST ID, RAN Function Id, Call process ID, Control Header, Control message)
Step 1 (O)	xApp may request **E2 related API: E2 Guidance** from Conflict Mitigation (see section 9.2.3.1 for details)
Step 2 [Informative]	xApp assigns a unique REQUEST ID
Step 3 (O)	xApp sends **E2 related API: E2 Control** request with message contents (REQUEST ID, RAN Function Id, Call process ID, Control Header, Control message) for a E2 Node. Message may be routed either a) Directly to E2 Termination b) Or message may be routed to Conflict Mitigation (see section 9.2.3.4 for details)
Step 4 [Informative]	If message was routed to Conflict Mitigation, Conflict Mitigation processes request
Step 5 [Informative]	If message was routed to Conflict Mitigation and Conflict Mitigation accepts proposed RIC Control message, then message is forwarded to appropriate E2 Termination instance
Step 6 (O)	If message was routed to Conflict Mitigation and Conflict Mitigation rejects proposed RIC Control message, then **RIC API: E2 related E2 Control** response (Reject with Cause) message sent to xApp.
Step 7 (M)	E2 Termination receives incoming proposed RIC Control message, selects appropriate E2 interface and sends **RIC Control Request** (REQUEST ID, contents) to E2 Node.
Step 8 (M)	If E2 Node accepts RIC Control Request, then E2 Node responds with **RIC Control Response**
Step 9 (M)	E2 Termination sends **E2 Related API: E2 Control** response (Success with Outcome) to xApp, routing either a) directly or b) via Conflict Mitigation.
Step 10 (O)	If response is routed via Conflict Mitigation, then **E2 Related API: E2 Control** response (Success with Outcome) is sent to xApp
Step 11 (M)	If E2 Node rejects RIC Control Request, then E2 Node responds with **RIC Control Failure** (with Cause)
Step 12 (M)	E2 Termination sends **E2 Related API: E2 Control** response (Failure with Cause) to xApp, routing either a) directly or b) via Conflict Mitigation
Step 13 (O)	If response is routed via Conflict Mitigation, then **E2 Related API: E2 Control** response (Failure with Cause) is sent to xApp

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

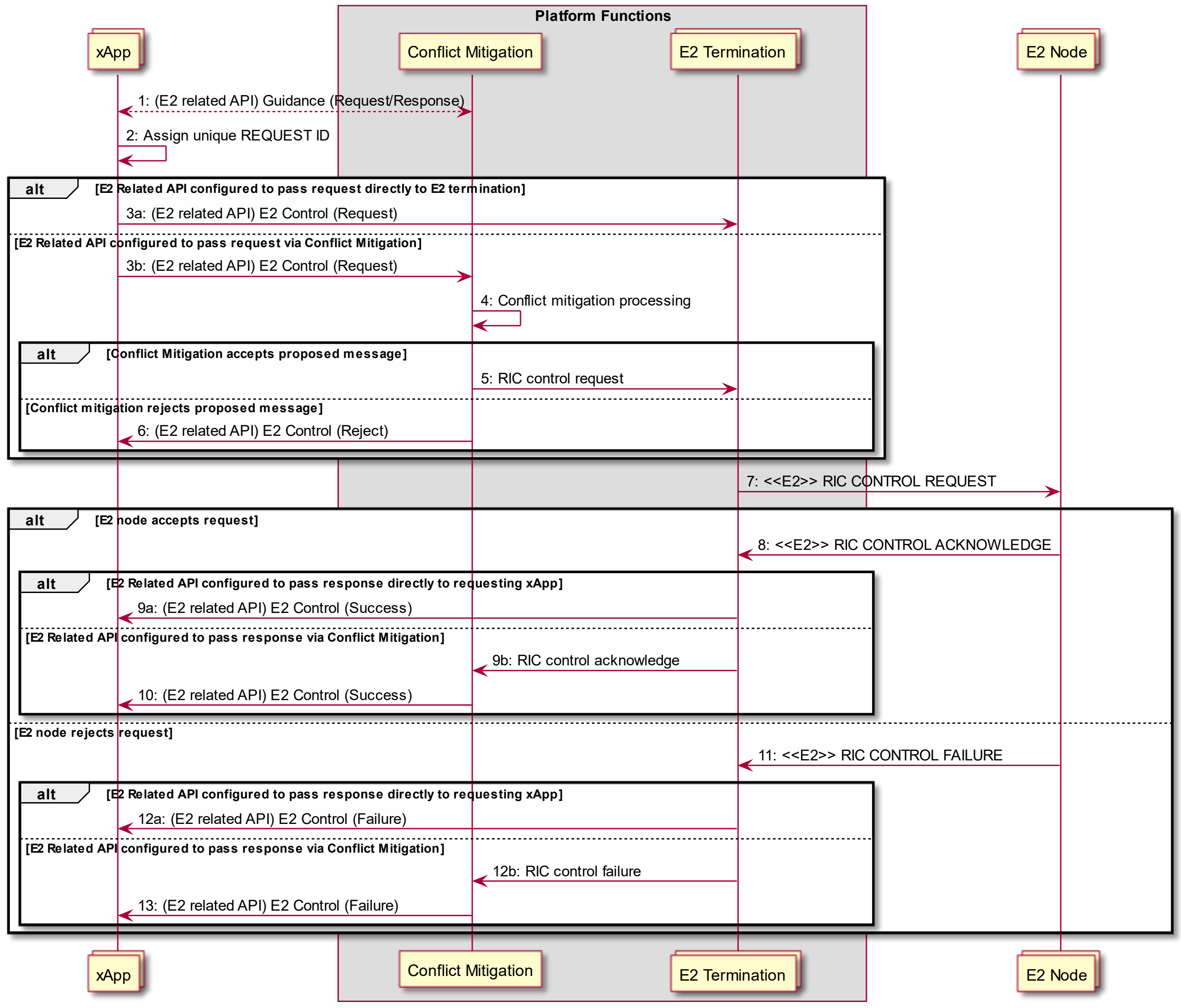

Platform Functions
xApp
Conflict Mitigation
E2 Termination
E2 Node
1: (E2 related API) Guidance (Request/Response)
2: Assign unique REQUEST ID
alt
[E2 Related API configured to pass request directly to E2 termination]
3a: (E2 related API) E2 Control (Request)
[E2 Related API configured to pass request via Conflict Mitigation]
3b: (E2 related API) E2 Control (Request)
4: Conflict mitigation processing
alt
[Conflict Mitigation accepts proposed message]
5: RIC control request
[Conflict mitigation rejects proposed message]
6: (E2 related API) E2 Control (Reject)
7: <<E2>> RIC CONTROL REQUEST
alt
[E2 node accepts request]
8: <<E2>> RIC CONTROL ACKNOWLEDGE
alt
[E2 Related API configured to pass response directly to requesting xApp]
9a: (E2 related API) E2 Control (Success)
[E2 Related API configured to pass response via Conflict Mitigation]
9b: RIC control acknowledge
10: (E2 related API) E2 Control (Success)
[E2 node rejects request]
11: <<E2>> RIC CONTROL FAILURE
alt
[E2 Related API configured to pass response directly to requesting xApp]
12a: (E2 related API) E2 Control (Failure)
[E2 Related API configured to pass response via Conflict Mitigation]
12b: RIC control failure
13: (E2 related API) E2 Control (Failure)

Figure 9.2.4-1 E2 Control API Procedure

9.2.3 E2 Guidance API related procedures

9.2.3.1 E2 Guidance request/response API 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 Conflict Mitigation platform function prior to initiating an action.

Guidance from Conflict Mitigation 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 other platform functions

This procedure is based on the following assumptions:

- xApp may use E2 Related API to obtain guidance from Conflict Mitigation to resolve potential conflicts prior to initiating a RIC function procedure

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

- xApp may use Conflict Mitigation 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 other platform function using **E2 Related API: E2 Guidance** response

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
	  o Database
	  o Conflict Mitigation

Assumptions	- Conflict Mitigation has access to sufficient information to both detect a potential conflict and take a decision on an optimal mitigation solution - Conflict Mitigation may initiate guidance towards other Platform Functions and/or xApp as an optional addition response to Guidance Request
Pre-conditions	- xApp has been authorized to request guidance from Conflict Mitigation - xApp has been assigned xApp ID
Begins when	xApp determines need to request guidance from Conflict Mitigation
Step 1 (M)	xApp sends **E2 Related API E2 Guidance** Request to Conflict Mitigation
Step 2 [Informative]	Conflict Mitigation may recover related information from platform database
Step 3 [Informative]	Conflict Mitigation processes request
Step 4 [Informative]	Conflict Mitigation may signal conflict and/or provide guidance to another platform function
Step 5 (O)	Conflict Mitigation may signal conflict and/or provide guidance to another xApp using **E2 Related API E2 Guidance** (modification)
Step 6 (M)	Conflict Mitigation sends **E2 Related API E2 Guidance** response to xApp
Ends with	xApp continues processing using Conflict Mitigation guidance response

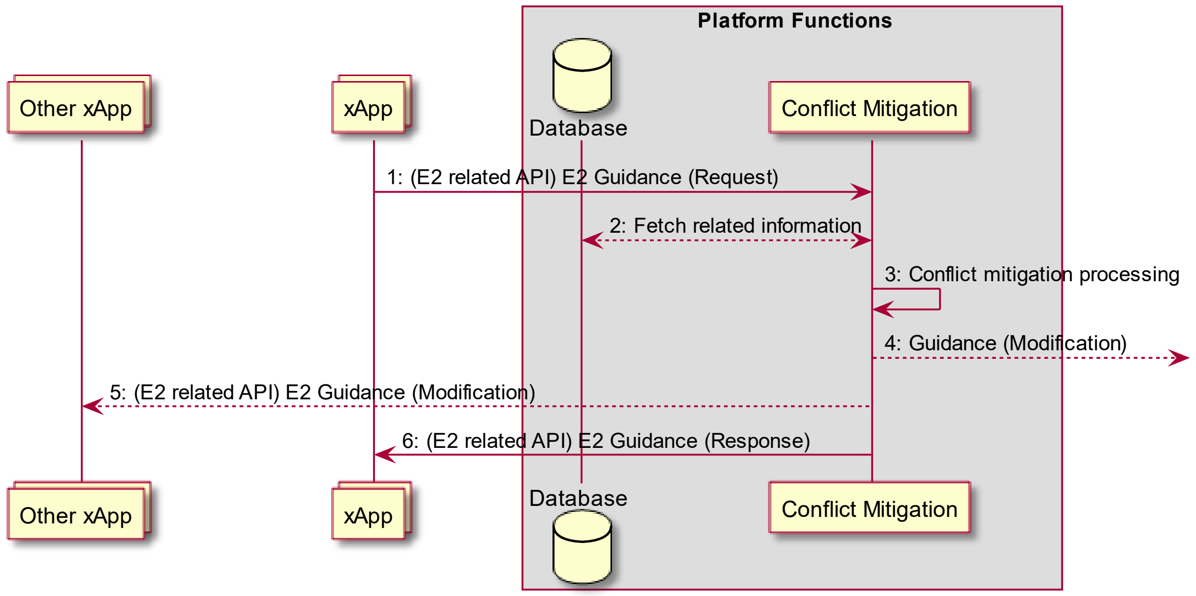

Platform Functions
Other xApp
xApp
Database
Conflict Mitigation
1: (E2 related API) E2 Guidance (Request)
2: Fetch related information
3: Conflict mitigation processing
4: Guidance (Modification)
5: (E2 related API) E2 Guidance (Modification)
6: (E2 related API) E2 Guidance (Response)

Figure 9.2.3.1-1: xApp initiated E2 guidance request/response procedure

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

9.2.3.2 E2 Guidance modification API procedure: xApp Subscription Management initiated

The purpose of the Near-RT RIC Platform internal xApp Subscription Management initiated E2 Guidance modification procedure is to allow Conflict Mitigation to signal a potential modification of E2 Guidance as the result of an xApp Subscription Management request to obtain guidance from the Conflict Mitigation platform function prior to handling an action.

This procedure is based on the following assumptions:

- xApp Subscription Management may request guidance from Conflict Mitigation to resolve potential conflicts prior to initiating a RIC function procedure
- Conflict Mitigation may need to subsequentially modify previous provided Guidance to an xApp

The following outcomes are considered:

- Conflict Mitigation provides guidance to xApp Subscription Management
- Conflict Mitigation provides modified guidance to another Near-RT RIC platform function
- Conflict Mitigation provides modified guidance to an xApp using **E2 Related API: E2 Guidance** modification

Use Case Stage	Evolution / Specification
Goal	xApp Subscription Management initiation of Conflict Mitigation guidance resulting in potential E2 Related API: E2 Guidance modification sent to an xApp
Actors and Roles	- Near-RT RIC Platform ○ Database ○ Conflict Mitigation ○ xApp Subscription Management
Assumptions	- Conflict Mitigation has access to sufficient information to both detect a potential conflict and take a decision on an optimal mitigation solution - Conflict Mitigation may initiate guidance towards other Platform Functions and/or xApp as an optional addition response to Guidance Request
Pre-conditions	- xApp Subscription Management has been configured to request guidance from Conflict Mitigation
Begins when	xApp Subscription Management receives incoming request
Step 1 [Informative]	xApp Subscription Management requests guidance from Conflict Mitigation
Step 2 [Informative]	Conflict Mitigation may recover related information from platform database
Step 3 [Informative]	Conflict Mitigation processes request
Step 4 [Informative]	Conflict Mitigation may signal conflict and/or provide guidance to another platform function
Step 5 (O)	Conflict Mitigation may signal conflict and/or provide guidance to another xApp using **E2 Related API: E2 Guidance** (modification)
Step 6 [Informative]	Conflict Mitigation responds to xApp Subscription Management
Step 7 [Informative]	xApp Subscription Management may consider Conflict Mitigation response to determine response to incoming request
End with	xApp Subscription Management sends response to incoming request

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

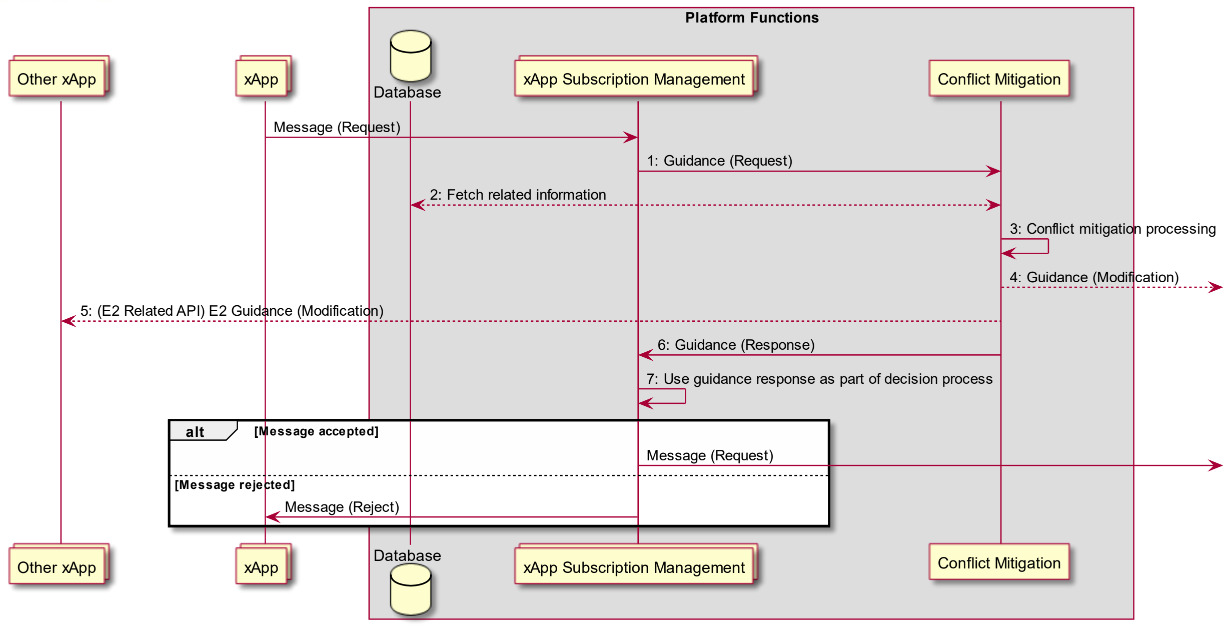

Platform Functions
Other xApp
xApp
Database
xApp Subscription Management
Conflict Mitigation
Message (Request)
1: Guidance (Request)
2: Fetch related information
3: Conflict mitigation processing
4: Guidance (Modification)
5: (E2 Related API) E2 Guidance (Modification)
6: Guidance (Response)
7: Use guidance response as part of decision process
alt
[Message accepted]
Message (Request)
[Message rejected]
Message (Reject)

Figure 9.2.3.2-1: E2 Guidance modification API procedure: xApp Subscription Management initiated

9.2.3.3 E2 Guidance modification API procedure: message monitoring initiated

The purpose of the Conflict mitigation related message monitoring procedure in the Near-RT RIC is to allow the Conflict Mitigation platform function to monitor ongoing transactions involving other Platform Functions.

This procedure is based on the following assumptions:

- Other Platform functions (i.e. E2 termination, xApp Subscription Management) may be configured to forward messages related to xApp transactions to Conflict Mitigation
- Conflict Mitigation may use information obtained when formulating responses in a subsequent procedure and/or modify previous Guidance to another Near-RT RIC platform function
- Conflict Mitigation may use information to provide modified guidance to an xApp using **E2 Related API: E2 Guidance** modification

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

Use Case Stage	Evolution / Specification
Goal	Platform function forward xApp related messages to Conflict Mitigation resulting in potential E2 Related API: E2 Guidance modification sent to an xApp
Actors and Roles	- Near-RT RIC Platform ○ Conflict Mitigation ○ Other Platform Function ▪ xApp Subscription Management ▪ E2 Termination ▪ Etc.
Assumptions	- Conflict Mitigation is capable of exploiting collected xApp related messages
Pre-conditions	- Other Platform Functions have been configured to copy and forward selected xApp related message to Conflict Mitigation
Step 1 [Informative]	Message received by other Platform Function
Step 2 [Informative]	Other Platform Function handles incoming message and sends outgoing message
Step 3 [Informative]	Other Platform Function applies message filter and determines that message is to be copied and sent to Conflict Mitigation. If filter conditions match then forwards copy of incoming message to Conflict mitigation
Step 4 [Informative]	Conflict Mitigation processes forwarded message
Step 5 [Informative]	Conflict Mitigation may signal conflict and/or provide updated guidance to another platform function
Step 6 (O)	Conflict Mitigation may signal conflict and/or provide updated guidance to another xApp using **E2 Related API: E2 Guidance** (modification)

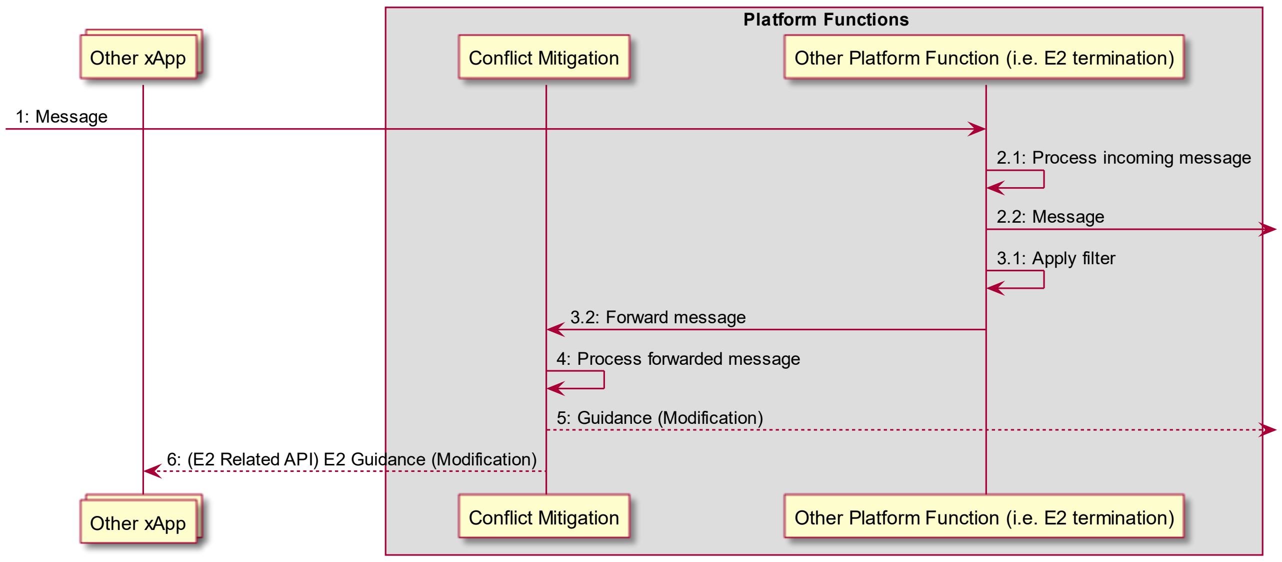

Platform Functions
Other xApp
Conflict Mitigation
Other Platform Function (i.e. E2 termination)
1: Message
2.1: Process incoming message
2.2: Message
3.1: Apply filter
3.2: Forward message
4: Process forwarded message
5: Guidance (Modification)
6: (E2 Related API) E2 Guidance (Modification)
Other xApp
Conflict Mitigation
Other Platform Function (i.e. E2 termination)

Figure 9.2.3.3-1: E2 Guidance modification API procedure: message monitoring initiated

9.2.3.4 E2 Guidance modification API procedure: Conflict Mitigation initiated

The purpose of the Conflict Mitigation initiated conflict mitigation procedure in the Near-RT RIC is to allow Conflict Mitigation to directly intervene during the handling an action.

This procedure is based on the following assumptions:

- Messages from xApp or other platform functions may be redirected towards Conflict Mitigation to resolve potential conflicts prior to initiating a RIC function procedure
- xApp would not be aware of the redirection if conflict mitigation accepts the proposed message

The following outcomes are considered:

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

- Conflict Mitigation accepts proposed message and forwards to other platform function
- Conflict Mitigation rejects proposed message and sends a rejection message to the initiating xApp or other platform function providing both cause for rejection and optionally guidance

Use Case Stage	Evolution / Specification	<<Uses>> Related use
Goal	Conflict Mitigation initiation of Conflict Mitigation guidance resulting in potential E2 Related API: E2 Guidance modification sent to an xApp	

Actors and Roles	- xApp or other platform function
	- Near-RT RIC Platform
	    o Database
	    o Conflict Mitigation	

Assumptions	- Messages from xApp or other platform function may be redirected to Conflict Mitigation - Conflict Mitigation has access to sufficient information to both detect a potential conflict and take a decision on an optimal mitigation solution - Conflict Mitigation may initiate guidance towards other Platform Functions and/or xApp as an optional addition response to Guidance Request	
Pre-conditions	- Message infrastructure has been configured to redirect specified messages to Conflict Mitigation	
Begins when	xApp or other platform function sends message to platform function	RIC functional procedures
Step 1 [Informative]	Message infrastructure redirects message from xApp to Conflict Mitigation	
Step 2 [Informative]	Conflict Mitigation may recover related information from SDL	
Step 3 [Informative]	Conflict Mitigation processes request	
Step 4 [Informative]	If accepted, Conflict Mitigation may forward to E2 termination for delivery	
Step 5 [Informative]	If rejected, Conflict Mitigation may send a rejection message to xApp including indication of cause and optionally guidance information	
Step 6 (O)	Conflict Mitigation may signal conflict and/or provide guidance to another platform function or xApp	
Step 7 (O)	Conflict Mitigation may signal conflict and/or provide updated guidance to another xApp using **E2 Related API: E2 Guidance** (modification)	
Ends with	xApp receives eventual response from E2 Node	

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

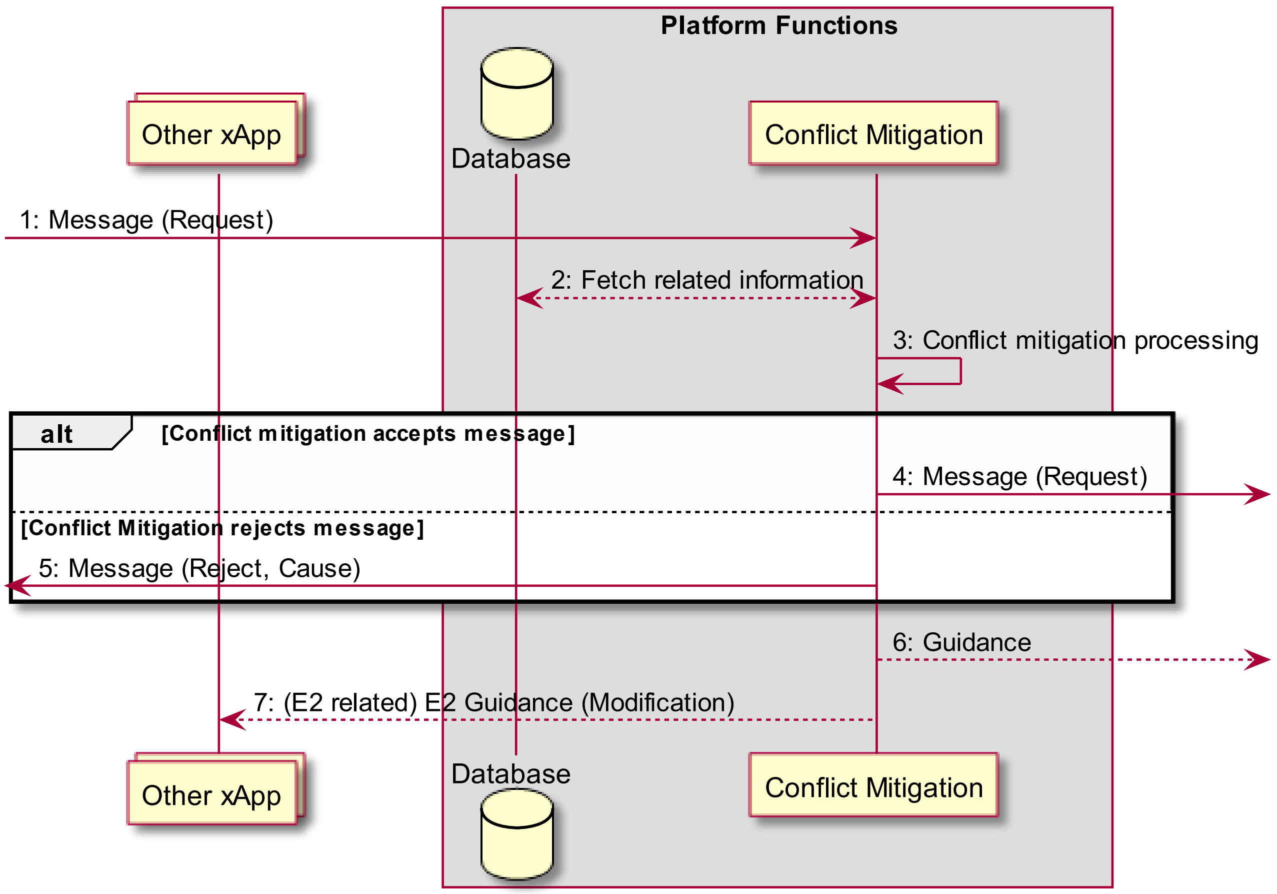

Platform Functions
Other xApp
Database
Conflict Mitigation
1: Message (Request)
2: Fetch related information
3: Conflict mitigation processing
alt
[Conflict mitigation accepts message]
4: Message (Request)
[Conflict Mitigation rejects message]
5: Message (Reject, Cause)
6: Guidance
7: (E2 related) E2 Guidance (Modification)
Other xApp
Database
Conflict Mitigation

Figure 9.2.3.4-1: E2 Guidance modification API procedure: Conflict Mitigation initiated

9.3 Management API Procedures

9.3.1 xApp Registration procedure

This procedure registers the xApp application 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
- 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

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

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. - Management Service in Near-RT RIC platform: handle registration related message from/to xApp and O1 termination and may do initial validation of xApp.	
Assumptions	The API channel is established successfully between xApp and Management Service in the Near-RT RIC platform.	
Pre conditions	The xApp is deployed onto 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 Management Service Component in 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 - 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 Management Service 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, O1 Terminations formats and sends a Notify MOI Creation to SMO to notify SMO that a MOI has been created for a new xApp.	

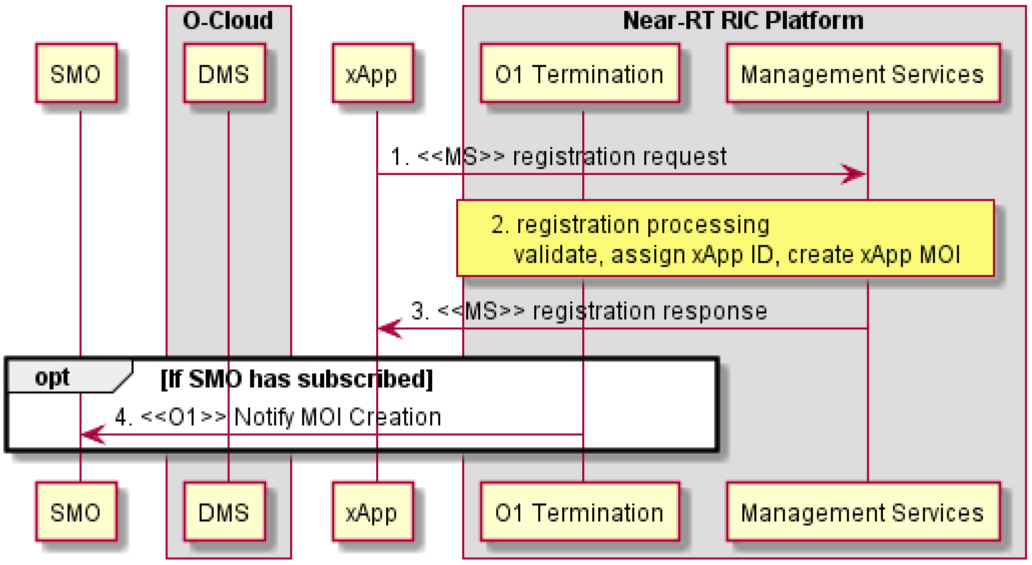

O-Cloud
Near-RT RIC Platform
SMO
DMS
xApp
O1 Termination
Management Services
1. <<MS>> registration request
2. registration processing
validate, assign xApp ID, create xApp MOI
3. <<MS>> registration response
opt
[If SMO has subscribed]
4. <<O1>> Notify MOI Creation
SMO
DMS
xApp
O1 Termination
Management Services

Figure 9.3.1-1: xApp Registration procedure

9.3.2 xApp Deregistration procedure

This procedure deregisters the xApp from the Near-RT RIC platform.

The message flow is FFS.

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

9.3.3 xApp Configuration procedure

This procedure configures an xApp when a configuration request is received from the SMO over the O1. SMO sends the configuration request via a NETCONF command over the O1 to O1 Termination in the Near-RT RIC platform. O1 Termination validates and saves the configuration persistently in the Config DB and responds to the SMO. If the configuration is for an xApp, Management Services notifies the xApp of the new configuration via the Configuration API. xApp takes the new configuration into use.

Use Case Stage	Evolution / Specification	<<Uses>> Related use
Goal	To configure a registered xApp.	
Actors and Roles	- xApp: the object of xApp configuration procedure - Near-RT RIC platform: Management Service: handle the configuration related message from/to xApp and O1 termination. O1 Termination: receive and send message from/to Management Service and SMO.	
Assumptions	- The API channel is established successfully between xApp and Management Service in the Near-RT RIC platform,	
Pre conditions	- The xApp is installed successfully on the Near-RT RIC platform and registered to the SMO.	
Begins when	O1 Termination receives the configuration message from SMO to indicate that there's configuration for the xApp.	
Step 1(M) [informative]	Near-RT RIC determines that the configuration is for an xApp.	
Step 2(M)	The Management Service identify the reconfiguration info related to the xApp and send the xApp configuration message to the xApp via the API.	
Step 3(M)	The xApp takes the new configuration into use and response the Management Service with the xApp configuration response message to indicate the configuration result (success or failure).	

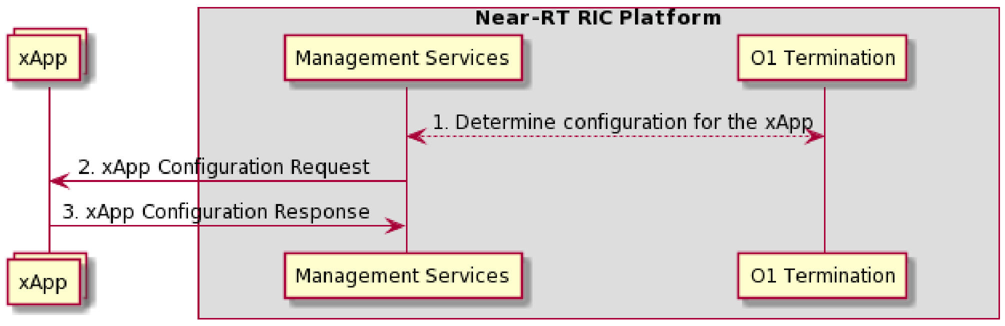

Near-RT RIC Platform
xApp
Management Services
O1 Termination
1. Determine configuration for the xApp
2. xApp Configuration Request
3. xApp Configuration Response
xApp
Management Services
O1 Termination

Figure 9.2.3-1: xApp Configuration procedure

9.4 SDL API Procedures

9.4.1 SDL Client Registration procedure

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

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

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

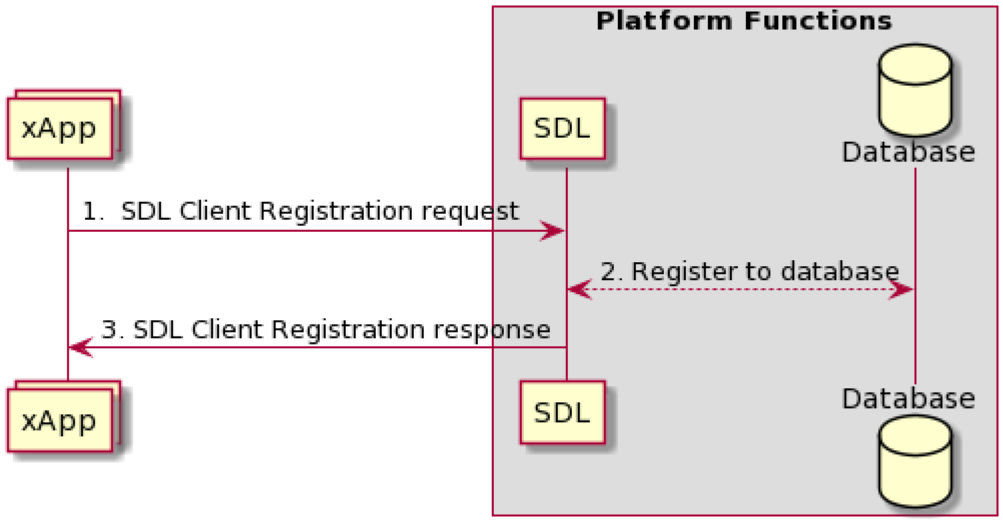


Figure 9.4.1-1: SDL Client Registration procedure

9.4.2 SDL Client Deregistration procedure

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

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

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

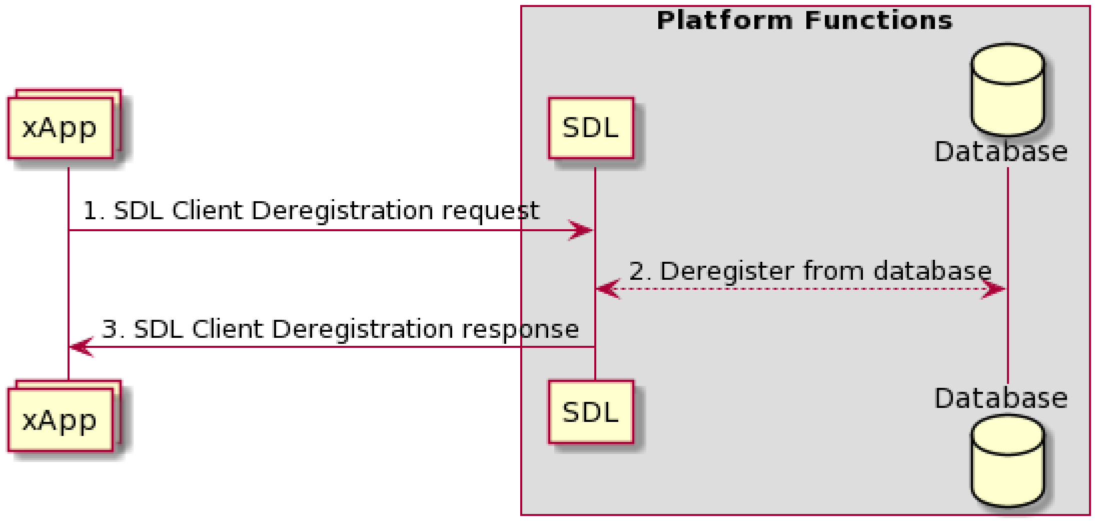

Platform Functions
xApp
SDL
Database
1. SDL Client Deregistration request
2. Deregister from database
3. SDL Client Deregistration response
xApp
SDL
Database

Figure 9.4.2-1: SDL Client Deregistration procedure

9.4.3 Fetch Data procedure

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

Use Case Stage	Evolution / Specification
Goal	xApp fetches data from the SDL.
Actors and Roles	- xApp: Originator of Fetch Data request - SDL: Response to the request of xApp
Pre-conditions	- SDL API services have been registered with API enablement. - 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 from the SDL in the Near-RT-RIC platform.
Step 1 (M)	xApp requests to SDL to fetch data from database.
Step 2 [Informative]	SDL fetches xApp required information from database (internal implementation).
Step 3 (M)	SDL sends the requested data or the failure response to specific xApp.

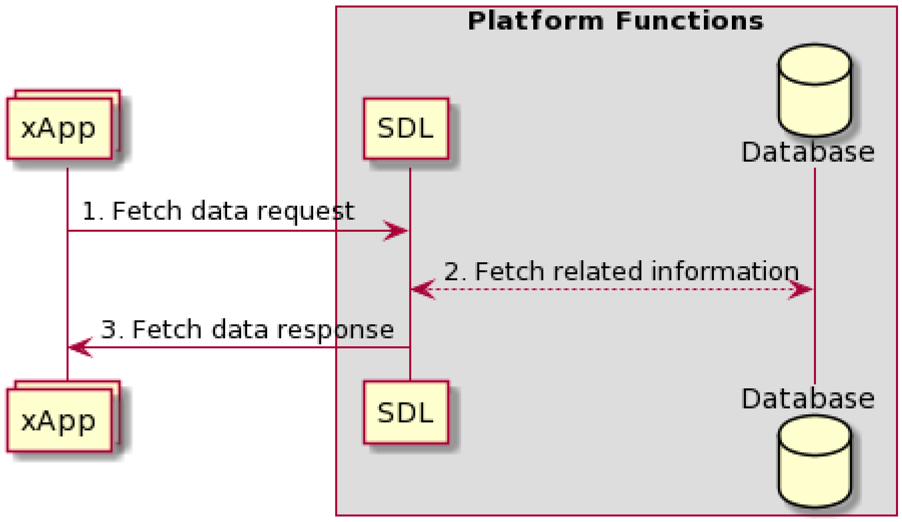


Figure 9.4.3-1: Fetch Data procedure

9.4.4 Subscribe/Notify procedure

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

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

Use Case Stage	Evolution / Specification
Goal	XApp subscribes with the SDL for data changes and receives related notifications.
Actors and Roles	- xApp: Originator of Subscribe request - SDL: Response to the request of xApp, and notify the subscribed information to xApp.
Pre-conditions	- SDL API services have been registered by API enablement. - xApp has successfully registered as a client to the SDL before.
Begins when	xApp determines to subscribe for notifications related to information changes in the SDL in the Near-RT-RIC platform.
Step 1 (M)	xApp sends the request to SDL to subscribe for notifications related to information changes.
Step 2 (M)	SDL responds to the xApp subscription.
Step 3 [Informative]	SDL subscribes related information from database (internal implementation).
Step 4 (M)	SDL sends the notification for the subscribed information changes to the xApp. The xApp can further send a request to SDL to fetch the updated information.

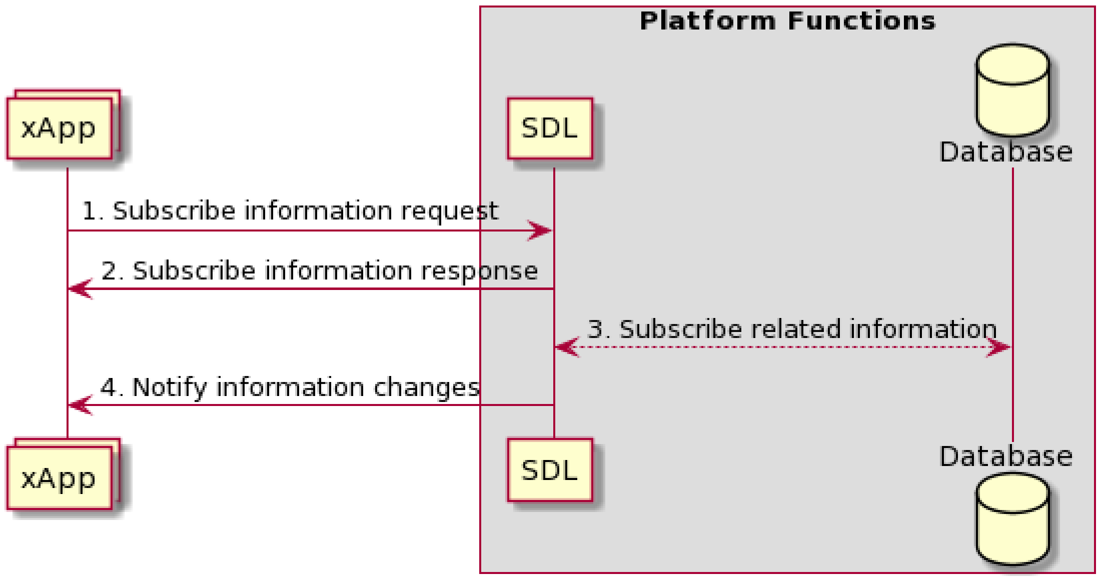

Platform Functions
xApp
SDL
Database
1. Subscribe information request
2. Subscribe information response
3. Subscribe related information
4. Notify information changes
xApp
SDL
Database

Figure 9.4.4-1: Subscribe/Notify procedure

9.4.5 Store Data procedure

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

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

Use Case Stage	Evolution / Specification
Goal	XApp requests to store data in the SDL.
Actors and Roles	- xApp: Originator of Store Data request - SDL: Response to the request of xApp
Pre-conditions	- SDL API services have been registered by API enablement. - 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 SDL to store related information and try to insert data to database.
Step 2 [Informative]	SDL stores xApp requested information in database (internal implementation).
Step 3 (M)	SDL responds to the xApp, including success and failure operation.

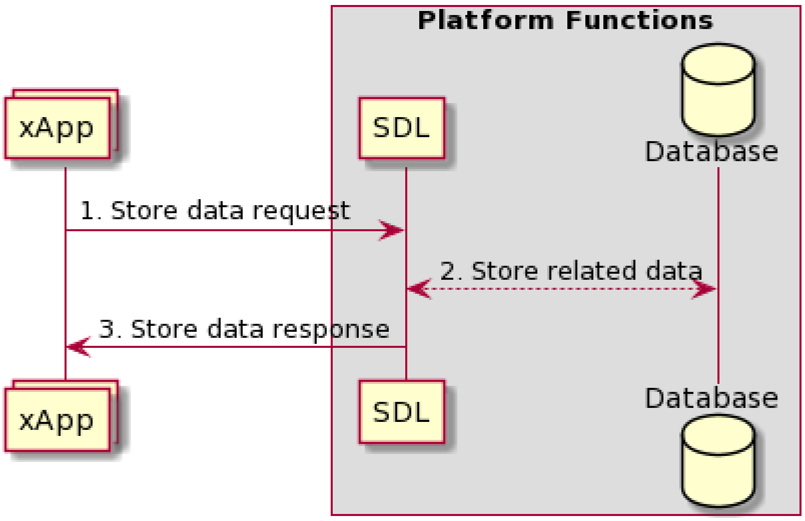


Figure 9.4.5-1: Store Data procedure

9.4.6 Modify Data procedure

The Modify Data procedure in the Near-RT RIC allows the xApp to request SDL 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 - SDL: Response to the request of xApp
Pre-conditions	- SDL API services have been registered by API enablement. - 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 SDL to modify related information, including update and delete data in database.
Step 2 [Informative]	SDL modifies or deletes xApp requested information in database (internal implementation).
Step 3 (M)	SDL responds to the xApp, including success and failure operation.

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

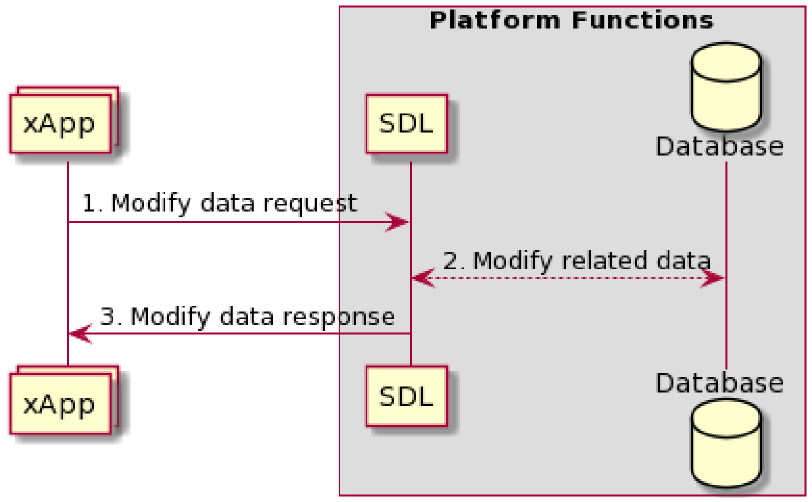


Figure 9.4.6-1: Modify Data procedure

9.5 API Enablement Procedures

9.5.1 Near-RT RIC API Discovery procedure

This procedure enables the xApp(s) to discover the Near-RT RIC APIs which are offered by the Near-RT RIC platform.

Use Case Stage	Evolution / Specification	<<Uses>> Related use
Goal	To allow the xApp to discover which APIs are provided by the Near-RT RIC platform and how to access them.	
Actors and Roles	- xApp: Requester of the available Near-RT RIC APIs in RIC platform and potential Near-RT RIC API consumer. - API enablement: providing the Near-RT RIC API information to the xApps.	
Assumptions	For initiating the discovery, the xApp has registered to RIC platform and is aware of the Enablement API info.	
Pre conditions	- The xApp is deployed to RIC platform. - The Enablement API info has been provided to the xApp (as pre-configuration by SMO or by other means.	
Begins when	xApps wants to discover the APIs which are provided by the RIC platform (via API enablement).	
Step 1 (M)	xApp sends a Near-RT RIC API discovery request to the API enablement. It includes the xApp identity and includes query information. Query information may include criteria for discovering matching APIs (e.g. API type, Serving Area Information, interfaces, protocols)	
Step 2 [Informative]	API enablementverifies the identity of the xApp. API enablement retrieves the stored Near-RT RIC API(s) information, as per the query information in the discovery request. Also, the API enablement applies the discovery policy and performs filtering of the retrieved near-RT RIC APIs information.	
Step 3 (M)	The API enablement function sends a near-RT RIC API discovery response to the xApp with the information about the APIs for which the xApp has the required authorization.	

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

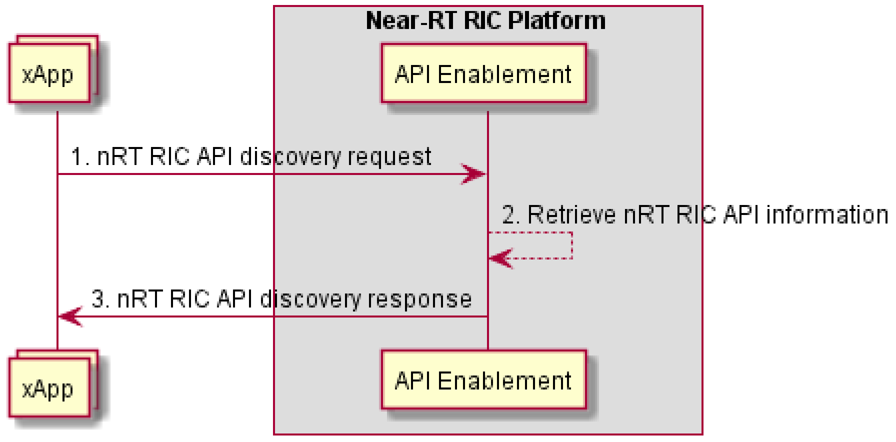

Near-RT RIC Platform
xApp
API Enablement
1. nRT RIC API discovery request
2. Retrieve nRT RIC API information
3. nRT RIC API discovery response
xApp
API Enablement

Figure 9.4.1-1: Near-RT RIC API Discovery procedure

9.5.2 Procedures related to API Event Subscription and Notification

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

9.5.2.1 Procedure for Event Subscription

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

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

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 API enablement to monitor events related to the near-RT RIC APIs - API enablement: Providing the subscription to Near-RT RIC API events	
Assumptions	- For initiating the subscription, the xApp has registered to RIC platform and is aware of the Enablement API info. Also, xApp has discovered the provided Near-RT RIC APIs.	
Pre-conditions	- The xApp is registered to RIC platform. - The Enablement API information has been provided to the xApp (as pre-configuration by SMO. - The xApp has discovered the registered near-RT RIC APIs (9.5.1)	
Begins when	xApp has discovered the near-RT RIC APIs as provided by the RIC platform and wants to subscribe for event notifications.	
Step 1 (M)	xApp sends an event subscription request to the API enablement 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 [Informative]	The API enablement checks whether the xApp is authorized to monitor the requested event(s)	
Step 3 (M)	The API enablement stores the approved subscription information.	
Step 4 (M)	The API enablement sends an API event subscription response to the xApp. This response indicates the success or failure of the event subscription operation.	

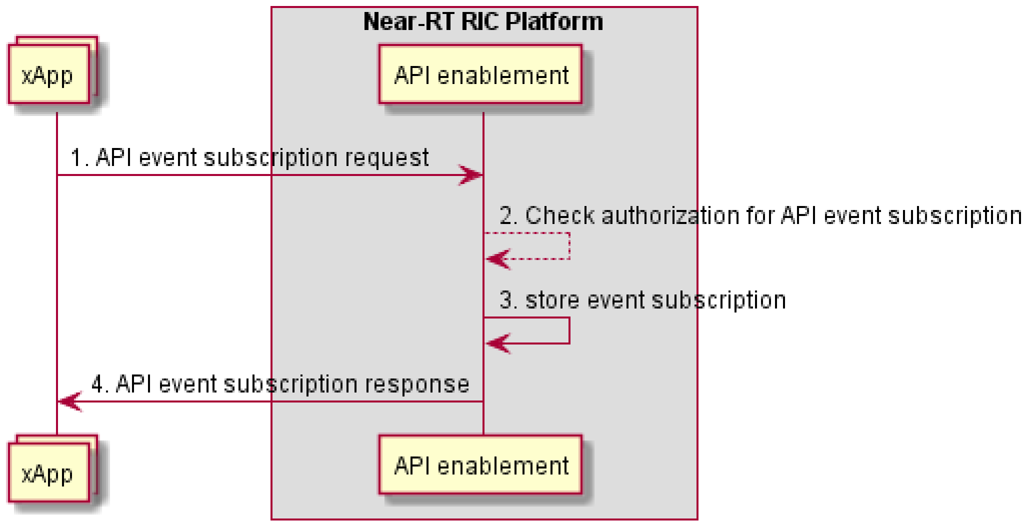

Near-RT RIC Platform
xApp
API enablement
1. API event subscription request
2. Check authorization for API event subscription
3. store event subscription
4. API event subscription response
xApp
API enablement

Figure 9.4.2.1-1: Subscription for API-related events

9.5.2.2 Procedure for 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.

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

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 to API enablement from monitoring events related to the near-RT RIC APIs - API enablement: Removing the subscription of xApp to Near-RT RIC API events	
Assumptions	For initiating the subscription delete, the xApp has registered to RIC platform and is aware of the Enablement API info. Also, xApp has subscribed to events.	
Pre-conditions	The xApp has subscribed to near-RT RIC API related events (9.5.2.1)	
Begins when	xApp wants to un-subscribe for near-RT RIC API related events.	
Step 1 (M)	xApp sends an event subscription delete request to the API enablement with the information of the subscribed event and includes the xApp identifier and the event subscription identifier	
Step 2 [Informative]	The API enablement, upon receiving the event subscription delete request from the xApp, checks for the event subscription corresponding to the xApp and further checks if the subscribing entity is authorized to un-subscribe.	
Step 3 (M)	API enablement deletes the subscription information for the xApp.	
Step 3 (M)	The API enablement sends an API event un-subscription response to the xApp. This response indicates the success or failure of the event un-subscription operation.	

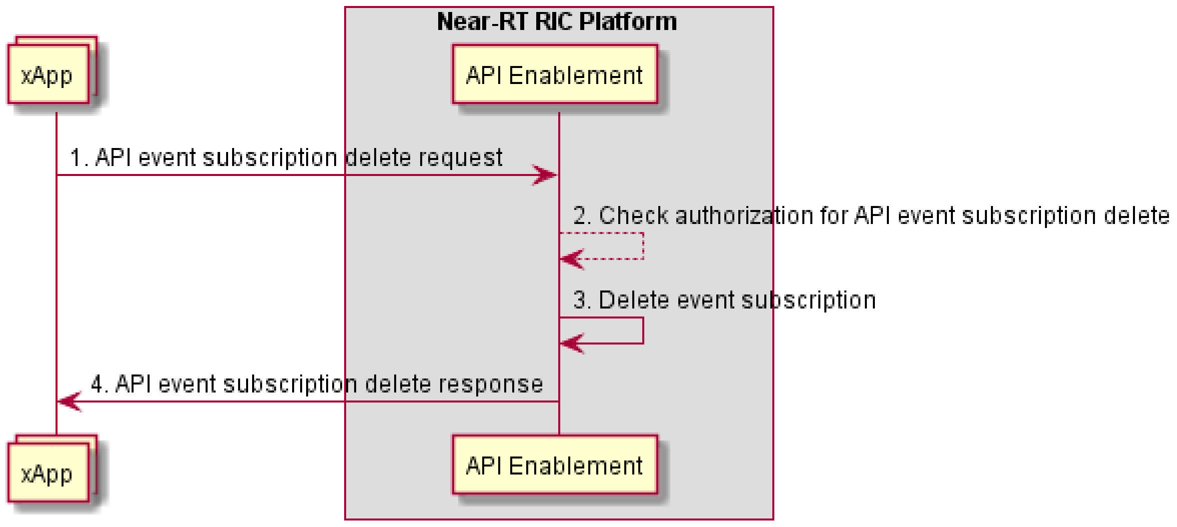

Near-RT RIC Platform
xApp
API Enablement
1. API event subscription delete request
2. Check authorization for API event subscription delete
3. Delete event subscription
4. API event subscription delete response
xApp
API Enablement

Figure 9.5.2.2-1: Subscription Delete for API-related events

9.5.2.3 Procedure for Event Notification

This procedure shows sending the event notification to the subscribed xApp, based on a trigger event captured at the API enablement.

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

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 - API enablement: Generating/ triggering event notifications and sending them to the subscribed xApps	
Assumptions	For receiving a notification, the xApp has registered to RIC platform and has subscribed to events.	
Pre-conditions	The xApp has subscribed to near-RT RIC API related events (9.5.2.1)	
Begins when	A trigger event criterion is met at the API Enablement. Such criterion has been configured during subscription of the xApp.	
Step 1 (M)	API enablement generates an event related to APIs, which is to be consumed by the subscribed xApp(s). For the generated event, the API enablement retrieves the list of corresponding event subscriptions.	
Step 2 (M)	The API enablement sends event notifications to all the subscribing xApp (s) that have subscribed for the event matching the criteria.	
Step 3 (O)	The xApp may send an event notification acknowledgement to API enablement. It depends on the capabilities of the transport used for notifications whether or not such acknowledgement can be sent.	

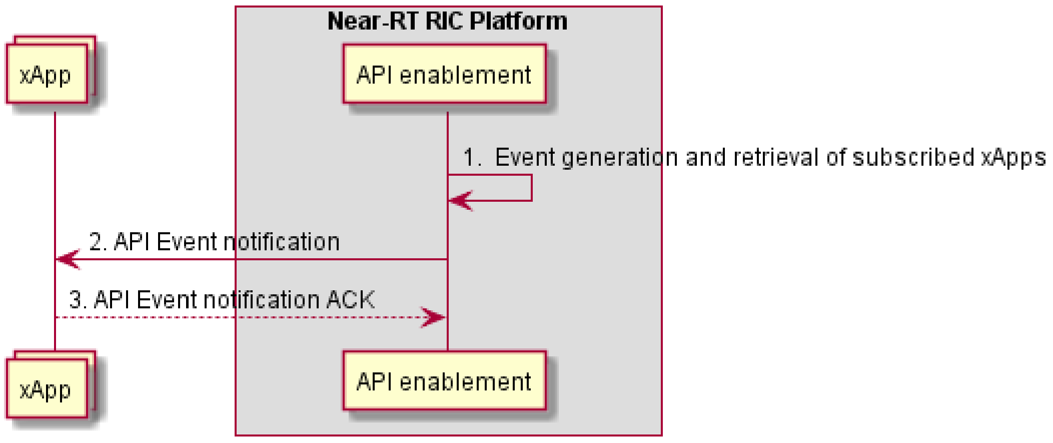

Near-RT RIC Platform
xApp
API enablement
1. Event generation and retrieval of subscribed xApps
2. API Event notification
3. API Event notification ACK
xApp
API enablement

Figure 9.5.2.3-1: API Event Notification

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

Annex ZZZ: O-RAN Adopter License Agreement

BY DOWNLOADING, USING OR OTHERWISE ACCESSING ANY O-RAN SPECIFICATION, ADOPTER AGREES TO THE TERMS OF THIS AGREEMENT.

This O-RAN Adopter License Agreement (the "Agreement") is made by and between the O-RAN Alliance and the entity that downloads, uses or otherwise accesses any O-RAN Specification, including its Affiliates (the "Adopter").

This is a license agreement for entities who wish to adopt any O-RAN Specification.

Section 1: DEFINITIONS

1.1 "Affiliate" means an entity that directly or indirectly controls, is controlled by, or is under common control with another entity, so long as such control exists. For the purpose of this Section, "Control" means beneficial ownership of fifty (50%) percent or more of the voting stock or equity in an entity.

1.2 "Compliant Implementation" means any system, device, method or operation (whether implemented in hardware, software or combinations thereof) that fully conforms to a Final Specification.

1.3 "Adopter(s)" means all entities, who are not Members, Contributors or Academic Contributors, including their Affiliates, who wish to download, use or otherwise access O-RAN Specifications.

1.4 "Minor Update" means an update or revision to an O-RAN Specification published by O-RAN Alliance that does not add any significant new features or functionality and remains interoperable with the prior version of an O-RAN Specification. The term "O-RAN Specifications" includes Minor Updates.

1.5 "Necessary Claims" means those claims of all present and future patents and patent applications, other than design patents and design registrations, throughout the world, which (i) are owned or otherwise licensable by a Member, Contributor or Academic Contributor during the term of its Member, Contributor or Academic Contributorship; (ii) such Member, Contributor or Academic Contributor has the right to grant a license without the payment of consideration to a third party; and (iii) are necessarily infringed by a Compliant Implementation (without considering any Contributions not included in the Final Specification). A claim is necessarily infringed only when it is not possible on technical (but not commercial) grounds, taking into account normal technical practice and the state of the art generally available at the date any Final Specification was published by the O-RAN Alliance or the date the patent claim first came into existence, whichever last occurred, to make, sell, lease, otherwise dispose of, repair, use or operate a Compliant Implementation without infringing that claim. For the avoidance of doubt in exceptional cases where a Final Specification can only be implemented by technical solutions, all of which infringe patent claims, all such patent claims shall be considered Necessary Claims.

1.6 "Defensive Suspension" means for the purposes of any license grant pursuant to Section 3, Member, Contributor, Academic Contributor, Adopter, or any of their Affiliates, may have the discretion to include in their license a term allowing the licensor to suspend the license against a licensee who brings a patent infringement suit against the licensing Member, Contributor, Academic Contributor, Adopter, or any of their Affiliates.

Section 2: COPYRIGHT LICENSE

2.1 Subject to the terms and conditions of this Agreement, O-RAN Alliance hereby grants to Adopter a nonexclusive, nontransferable, irrevocable, non-sublicensable, worldwide copyright license to obtain, use and modify O-RAN Specifications, but not to further distribute such O-RAN Specification in any modified or unmodified way, solely in furtherance of implementations of an O-RAN

Specification.

2.2 Adopter shall not use O-RAN Specifications except as expressly set forth in this Agreement or in a separate written agreement with O-RAN Alliance.

Section 3: FRAND LICENSE

3.1 Members, Contributors and Academic Contributors and their Affiliates are prepared to grant based on a separate Patent License Agreement to each Adopter under Fair Reasonable And Non- Discriminatory (FRAND) terms and conditions with or without compensation (royalties) a nonexclusive, non-transferable, irrevocable (but subject to Defensive Suspension), non-sublicensable, worldwide patent license under their Necessary Claims to make, have made, use, import, offer to sell, lease, sell and otherwise distribute Compliant Implementations; provided, however, that such license shall not extend: (a) to any part or function of a product in which a Compliant Implementation is incorporated that is not itself part of the Compliant Implementation; or (b) to any Adopter if that Adopter is not making a reciprocal grant to Members, Contributors and Academic Contributors, as set forth in Section 3.3. For the avoidance of doubt, the foregoing licensing commitment includes the distribution by the Adopter's distributors and the use by the Adopter's customers of such licensed Compliant Implementations.

3.2 Notwithstanding the above, if any Member, Contributor or Academic Contributor, Adopter or their Affiliates has reserved the right to charge a FRAND royalty or other fee for its license of Necessary Claims to Adopter, then Adopter is entitled to charge a FRAND royalty or other fee to such Member, Contributor or Academic Contributor, Adopter and its Affiliates for its license of Necessary Claims to its licensees.

3.3 Adopter, on behalf of itself and its Affiliates, shall be prepared to grant based on a separate Patent License Agreement to each Members, Contributors, Academic Contributors, Adopters and their Affiliates under Fair Reasonable And Non-Discriminatory (FRAND) terms and conditions with or without compensation (royalties) a nonexclusive, non-transferable, irrevocable (but subject to Defensive Suspension), non-sublicensable, worldwide patent license under their Necessary Claims to make, have made, use, import, offer to sell, lease, sell and otherwise distribute Compliant Implementations; provided, however, that such license will not extend: (a) to any part or function of a product in which a Compliant Implementation is incorporated that is not itself part of the Compliant Implementation; or (b) to any Members, Contributors, Academic Contributors, Adopters and their Affiliates that is not making a reciprocal grant to Adopter, as set forth in Section 3.1. For the avoidance of doubt, the foregoing licensing commitment includes the distribution by the Members', Contributors', Academic Contributors', Adopters' and their Affiliates' distributors and the use by the Members', Contributors', Academic Contributors', Adopters' and their Affiliates' customers of such licensed Compliant Implementations.

Section 4: TERM AND TERMINATION

4.1 This Agreement shall remain in force, unless early terminated according to this Section 4.

4.2 O-RAN Alliance on behalf of its Members, Contributors and Academic Contributors may terminate this Agreement if Adopter materially breaches this Agreement and does not cure or is not capable of curing such breach within thirty (30) days after being given notice specifying the breach.

4.3 Sections 1, 3, 5 - 11 of this Agreement shall survive any termination of this Agreement. Under surviving Section 3, after termination of this Agreement, Adopter will continue to grant licenses (a) to entities who become Adopters after the date of termination; and (b) for future versions of O-RAN Specifications that are backwards compatible with the version that was current as of the date of termination.

Section 5: CONFIDENTIALITY

Adopter will use the same care and discretion to avoid disclosure, publication, and dissemination of O-RAN Specifications to third parties, as Adopter employs with its own confidential information, but no less than reasonable care. Any disclosure by Adopter to its Affiliates, contractors and consultants should be subject to an obligation of confidentiality at least as restrictive as those contained in this Section. The foregoing obligation shall not apply to any information which is: (1) rightfully known by Adopter without any limitation on use or disclosure prior to disclosure; (2) publicly available through no fault of Adopter; (3) rightfully received without a duty of confidentiality; (4) disclosed by O-RAN Alliance or a Member, Contributor or Academic Contributor to a third party without a duty of confidentiality on such third party; (5) independently developed by Adopter; (6) disclosed pursuant to the order of a court or other authorized governmental body, or as required by law, provided that Adopter provides reasonable prior written notice to O-RAN Alliance, and cooperates with O-RAN Alliance and/or the applicable Member, Contributor or Academic Contributor to have the opportunity to oppose any such order; or (7) disclosed by Adopter with O-RAN Alliance's prior written approval.

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

Section 6: INDEMNIFICATION

Adopter shall indemnify, defend, and hold harmless the O-RAN Alliance, its Members, Contributors or Academic Contributors, and their employees, and agents and their respective successors, heirs and assigns (the "Indemnitees"), against any liability, damage, loss, or expense (including reasonable attorneys' fees and expenses) incurred by or imposed upon any of the Indemnitees in connection with any claims, suits, investigations, actions, demands or judgments arising out of Adopter's use of the licensed O-RAN Specifications or Adopter's commercialization of products that comply with O-RAN Specifications.

Section 7: LIMITATIONS ON LIABILITY; NO WARRANTY

EXCEPT FOR BREACH OF CONFIDENTIALITY, ADOPTER'S BREACH OF SECTION 3, AND ADOPTER'S INDEMNIFICATION OBLIGATIONS, IN NO EVENT SHALL ANY PARTY BE LIABLE TO ANY OTHER PARTY OR THIRD PARTY FOR ANY INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES RESULTING FROM ITS PERFORMANCE OR NON-PERFORMANCE UNDER THIS AGREEMENT, IN EACH CASE WHETHER UNDER CONTRACT, TORT, WARRANTY, OR OTHERWISE, AND WHETHER OR NOT SUCH PARTY HAD ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES. O-RAN SPECIFICATIONS ARE PROVIDED "AS IS" WITH NO WARRANTIES OR CONDITIONS WHATSOEVER, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE. THE O-RAN ALLIANCE AND THE MEMBERS, CONTRIBUTORS OR ACADEMIC CONTRIBUTORS EXPRESSLY DISCLAIM ANY WARRANTY OR CONDITION OF MERCHANTABILITY, SECURITY, SATISFACTORY QUALITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, ERROR-FREE OPERATION, OR ANY WARRANTY OR CONDITION FOR O-RAN SPECIFICATIONS.

Section 8: ASSIGNMENT

Adopter may not assign the Agreement or any of its rights or obligations under this Agreement or make any grants or other sublicenses to this Agreement, except as expressly authorized hereunder, without having first received the prior, written consent of the O-RAN Alliance, which consent may be withheld in O-RAN Alliance's sole discretion. O-RAN Alliance may freely assign this Agreement.

Section 9: THIRD-PARTY BENEFICIARY RIGHTS

Adopter acknowledges and agrees that Members, Contributors and Academic Contributors (including future Members, Contributors and Academic Contributors) are entitled to rights as a third-party beneficiary under this Agreement, including as licensees under Section 3.

Section 10: BINDING ON AFFILIATES

Execution of this Agreement by Adopter in its capacity as a legal entity or association constitutes that legal entity's or association's agreement that its Affiliates are likewise bound to the obligations that are applicable to Adopter hereunder and are also entitled to the benefits of the rights of Adopter hereunder.

Section 11: GENERAL

This Agreement is governed by the laws of Germany without regard to its conflict or choice of law provisions.

This Agreement constitutes the entire agreement between the parties as to its express subject matter and expressly supersedes and replaces any prior or contemporaneous agreements between the parties, whether written or oral, relating to the subject matter of this Agreement.

Adopter, on behalf of itself and its Affiliates, agrees to comply at all times with all applicable laws, rules and regulations with respect to its and its Affiliates' performance under this Agreement, including without limitation, export control and antitrust laws. Without limiting the generality of the foregoing, Adopter acknowledges that this Agreement prohibits any communication that would violate the antitrust laws.

By execution hereof, no form of any partnership, joint venture or other special relationship is created between Adopter, or O-RAN Alliance or its Members, Contributors or Academic Contributors. Except as expressly set forth in this

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.

Agreement, no party is authorized to make any commitment on behalf of Adopter, or O-RAN Alliance or its Members, Contributors or Academic Contributors.

In the event that any provision of this Agreement conflicts with governing law or if any provision is held to be null, void or otherwise ineffective or invalid by a court of competent jurisdiction, (i) such provisions will be deemed stricken from the contract, and (ii) the remaining terms, provisions, covenants and restrictions of this Agreement will remain in full force and effect.

Any failure by a party or third party beneficiary to insist upon or enforce performance by another party of any of the provisions of this Agreement or to exercise any rights or remedies under this Agreement or otherwise by law shall not be construed as a waiver or relinquishment to any extent of the other parties' or third party beneficiary's right to assert or rely upon any such provision, right or remedy in that or any other instance; rather the same shall be and remain in full force and effect.

Copyright © 2021 by O-RAN ALLIANCE e.V. All Rights Reserved.
Your use is subject to the terms of the O-RAN Adopter License Agreement in the Annex ZZZ.