 O-RAN-WG3.RICAPI-R003-v01.00

Technical Specification

O-RAN Working Group 3  
(Near-Real-time RIC and E2 Interface Workgroup)

Near-RT RIC APIs specification

Copyright © 2023 by the O-RAN ALLIANCE e.V.

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

O-RAN ALLIANCE e.V., Buschkauler Weg 27, 53347 Alfter, Germany

Register of Associations, Bonn VR 11238, VAT ID DE321720189

Contents

Foreword 4

Modal verbs terminology 4

1 Scope 5

2 References 5

2.1 Normative references 5

2.2 Informative references 6

3 Definition of terms, symbols and abbreviations 6

3.1 Terms 6

3.2 Symbols 6

3.3 Abbreviations 6

4 General 7

4.1 Introduction 7

4.2 General Principles 7

4.3 Version Control 7

5 A1-related API 7

5.1 Overview 7

5.2 A1-related API Procedures 9

5.2.1 A1 Policy Setup procedure 9

5.2.2 A1 Policy Update procedure 11

5.2.3 A1 Policy Delete procedure 12

5.3 Elements for A1-related API Communication 14

5.3.1 General 14

5.3.2 Message Functional Definition and Content 14

5.3.3 Information Element Definitions 16

5.3.4 Message and Information Element Abstract Syntax 19

5.3.5 Message Transfer Syntax 22

6 E2-related API 22

6.1 Overview 22

6.1.1 Protocol Stack 22

6.1.2 List of E2 related APIs, procedures and messages 22

6.2 E2-related API Procedures 24

6.2.1 E2\_Subscription\_API procedures 24

6.2.2 E2\_Indication\_API procedures 28

6.2.3 E2\_Control\_API procedures 29

6.3 Elements for E2-related API Communication 31

6.3.1 General 31

6.3.2 Message Functional Definition and Content 32

6.3.3 Information Element Definitions 39

6.3.4 Message and Information Element Abstract Syntax 42

6.3.5 Message Transfer Syntax 50

7 SDL API 51

7.1 Overview 51

7.1.1 Protocol Stack 51

7.1.2 List of SDL APIs, procedures and messages 51

7.2 SDL API Procedures 52

7.2.1 SDL Client Registration procedure 52

7.2.2 SDL Client Deregistration procedure 52

7.2.3 SDL Subscribe Information procedure 52

7.2.4 SDL Information Push procedure 54

7.2.5 SDL Information Update Notify procedure 54

7.2.6 SDL Information Fetch procedure 55

7.3 Elements for SDL API Communication 56

7.3.1 General 56

7.3.2 Message Functional Definition and Content 57

7.3.3 Information Element Definitions 60

7.3.4 Message and Information Element Abstract Syntax 65

7.3.5 Message Transfer Syntax 70

8 Management API 71

8.1 Overview 71

8.1.1 Protocol Stack 71

8.1.2 List of Management APIs, procedures and messages 71

8.2 Management API Procedures 72

8.2.1 xApp Registration Procedure 72

8.3 Elements for Management API Communication 73

8.3.1 General 73

8.3.2 Message Functional Definition and Content 73

8.3.3 Information Element Definitions 75

8.3.4 Message and Information Element Abstract Syntax 76

8.3.5 Message Transfer Syntax 78

9 Enablement API 78

9.1 Overview 78

9.1.1 Protocol Stack 78

9.1.2 List of Enablement APIs, procedures and messages 78

9.2 Enablement API Procedures 79

9.2.1 Enabl\_Discovery\_API Procedures 79

9.2.2 Enabl\_Events\_API Procedures 81

9.3 Elements for Enablement API Communication 85

9.3.1 General 85

9.3.2 Message Functional Definition and Content 85

9.3.3 Information Element Definitions 89

9.3.4 Message and Information Element Abstract Syntax 91

9.3.5 Message Transfer Syntax 95

10 Common definitions across APIs 95

10.1 Information element definitions 95

10.1.1 PTI 95

10.1.2 xApp ID 96

10.1.3 API Name 96

10.1.4 API Version 96

10.1.5 A1 Policy Type ID 96

10.1.6 Global E2 Node ID 96

10.1.7 RAN Function ID 97

10.1.8 SCTP API PDU 97

10.2 Information element abstract syntax 97

10.2.1 Abstact syntax in Protocol Buffers 97

Revision history 100

History 101

# Foreword

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

# Modal verbs terminology

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

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

# 1 Scope

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 version date and an increase in version number as follows:

version xx.yy.zz

where:

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

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

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

The present document specifies the signalling and data transport protocols for Near-RT RIC APIs. The protocols are developed in accordance with the functionalities and interactions stated in [2].

# 2 References

## 2.1 Normative references

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

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

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

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

[2] O-RAN.WG3.RICARCH-v03.00.00: “O-RAN, Near-Real-time RAN Intelligent Controller, Near-RT RIC Architecture”.

[3] O-RAN.WG3.E2AP-v02.01.00: “O-RAN, Near-Real-time RAN Intelligent Controller, E2 Application Protocol(E2AP)”.

[4] Protocol buffers. Available at <https://developers.google.com/protocol-buffers/>.

[5] gRPC. Available at <https://www.grpc.io/>.

[6] O-RAN.WG11.SecReqSpecs: “O-RAN, Security Requirements Specifications”.

[7] O-RAN.WG2.A1TD-v04.00: “O-RAN, A1 interface: Type Definitions”.

[8] O-RAN.WG3.E2SM-v02.01.02: “O-RAN, Near-Real-time RAN Intelligent Controller, E2 Service Model (E2SM)”.

[9] IETF RFC 4122: “A Universally Unique IDentifier (UUID) URN Namespace”.

## 2.2 Informative references

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

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

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

[i.1]Semantic Versioning Specification. Available at <https://semver.org>.

# 3 Definition of terms, symbols and abbreviations

## 3.1 Terms

For the purposes of the present document, the following terms apply:

**Near-RT RIC**: as defined in [2].

**Near-RT RIC APIs**: interfaces that connect xApp and Near-RT RIC platform.

**Near-RT RIC Platform**: a software platform to host xApps. The Near-RT RIC is comprised of the Near-RT RIC Platform and xApps.

**xApp**: as defined in [2].

## 3.2 Symbols

No Symbol is defined in this document.

## 3.3 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [1] and the following apply:

API Application Programming Interface

Near-RT RIC Near-real-time RAN Intelligent Controller

PTI Procedure Transaction Identity

# 4 General

## 4.1 Introduction

Section 4 specifies the principles and conventions commonly applied to Near-RT RIC API design with the Network API approach as described in section 7.1.2 of [2].

## 4.2 General Principles

For each Near-RT RIC API, the following principles should be followed:

- A default solution that includes network protocol and message encoding should be defined to ensure interoperability. The default solution requires mandatory support by all Near-RT RIC platforms and xApps. Additionally, one or more alternative solutions may be defined.

The common data types applicable to several API categories should be collected in Section 10.

## 4.3 Version Control

An API version shall consist of at least 3 fields, following a MAJOR.MINOR.PATCH pattern according to the Semantic Versioning Specification [i.1] unless specified otherwise.

EXAMPLE: "1.0.0".

The basic rules as reproduced from [i.1] shall apply for incrementing the versions:

- MAJOR version is incremented if backward incompatible change is made.

- MINOR version is incremented if functional change is made in a backward compatible way.

- PATCH version is incremented if correction is made in a backward compatible way.

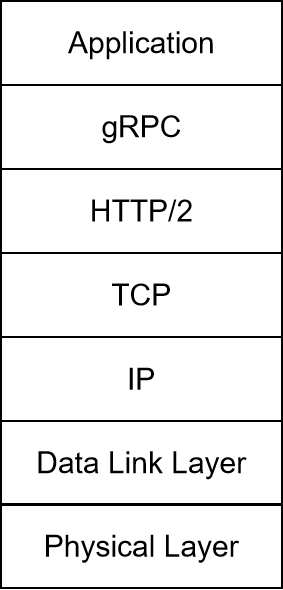
# 5 A1-related API

## 5.1 Overview

5.1.1 Protocol Stack

In this version of specification, the default protocol stack for the A1 related APIs is shown on Figure 5.1.1-1.

Note: the network layer for security protection is FFS.



**Figure 5.1.1-1: Protocol stack for A1 related APIs**

The A1 related APIs use gRPC [5] with Protocol Buffers [4] as the application layer serialization protocol.

5.1.2 List of A1 related APIs, procedures and messages

The A1 related API procedures introduced in section 9.2 of [2] are categorized under the A1 related APIs. These APIs include:

* A1 Policy API: enabling A1 policy related operations for the xApps.
* A1 Enrichment Information (A1-EI) API: enabling A1-EI related operations for the xApps.

These APIs are summarized in Table 5.1.2-1.

Table 5.1.2-1: List of A1 related APIs

|  |  |
| --- | --- |
| API Name | Description |
| A1\_Policy\_API | Service for A1 policy related operations |
| A1\_EI\_API | Service for A1 enrichment information related operations |

The table 5.1.2-2 lists the API procedures for each A1 related API.

Table 5.1.2-2: List of A1 related API procedures

|  |  |  |
| --- | --- | --- |
| API Name | API procedure | Communication type |
| A1\_Policy\_API | A1 Policy Setup | Request/Response |
| A1 Policy Update | Request/Response |
| A1 Policy Delete | Request/Response |
| A1 Policy Query | Request/Response |
| A1 Policy Status Update | Request/Response |
| A1\_EI\_API | A1 EI Subscription Setup | Request/Response |
| A1 EI Subscription Update | Request/Response |
| A1 EI Subscription Delete | Request/Response |
| A1 EI Delivery | Notify |
| A1 EI Query | Request/Response |

The table 5.1.2-3 lists the messages for each A1 related API procedure.

Table 5.1.2-3: List of A1 related API messages

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **API Procedure** | **Initiated by** | **Initiating Message** | **Outcome Message** | |
| **Successful Outcome** | **Unsuccessful Outcome** |
| A1 Policy Setup | Near-RT RIC platform | A1 POLICY SETUP REQUEST | A1 POLICY SETUP RESULT (SUCCESS) | A1 POLICY SETUP RESULT (FAILURE) |
| A1 Policy Update | Near-RT RIC platform | A1 POLICY UPDATE REQUEST | A1 POLICY UPDATE RESULT (SUCCESS) | A1 POLICY UPDATE RESULT (FAILURE) |
| A1 Policy Delete | Near-RT RIC platform | A1 POLICY DELETE REQUEST | A1 POLICY DELETE RESULT (SUCCESS) | A1 POLICY DELETE RESULT (FAILURE) |
| A1 Policy Query | Near-RT RIC platform | A1 POLICY QUERY REQUEST | A1 POLICY QUERY RESULT (SUCCESS) | A1 POLICY QUERY RESULT (FAILURE) |
| A1 Policy Status Update | xApp | A1 POLICY STATUS UPDATE | A1 POLICY STATUS UPDATE ACK | A1 POLICY STATUS UPDATE ERROR |
| A1 EI Subcription Setup | xApp | A1 EI SUBSCRIPTION SETUP REQUEST | A1 EI SUBSCRIPTION SETUP RESULT (SUCCESS) | A1 EI SUBSCRIPTION SETUP RESULT (FAILURE) |
| A1 EI Subcription Update | xApp | A1 EI SUBSCRIPTION UPDATE REQUEST | A1 EI SUBSCRIPTION UPDATE RESULT (SUCCESS) | A1 EI SUBSCRIPTION UPDATE RESULT (FAILURE) |
| A1 EI Subcription Delete | xApp | A1 EI SUBSCRIPTION DELETE REQUEST | A1 EI SUBSCRIPTION DELETE RESULT (SUCCESS) | A1 EI SUBSCRIPTION DELETE RESULT (FAILURE) |
| A1 EI Delivery | Near-RT RIC platform | A1 EI Delivery | - | - |
| A1 EI Query | xApp | A1 EI QUERY REQUEST | A1 EI QUERY RESULT (SUCCESS) | A1 EI QUERY RESULT (FAILURE) |

## 5.2 A1-related API Procedures

### 5.2.1 A1 Policy Setup procedure

#### 5.2.1.1 General

The purpose of the A1 Policy Setup procedure is to enable the execution of the transferred A1 policy.

#### 5.2.1.2 Successful operation

@startuml

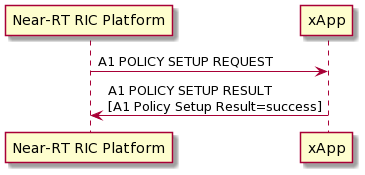
Participant rp as “Near-RT RIC Platform”

Participant ap as “xApp”

rp->ap: A1 POLICY SETUP REQUEST

ap->rp: A1 POLICY SETUP RESULT\n[A1 Policy Setup Result=success]

@enduml



**Figure 5.2.1.2‑1 : A1 Policy Setup procedure, successful operation**

The Near-RT RIC Platform initiates the procedure by sending A1 POLICY SETUP REQUEST message to xApp.

The A1 POLICY SETUP REQUEST message shall contain the information required by the xApp to setup one A1 policy including *A1 Policy Type ID* IE, *A1 Policy ID* IE, and *A1 Policy Object* IE.

Upon reception of the A1 POLICY SETUP REQUEST message, the xApp shall transform the received A1 policy into appropriate E2 / A1-EI operation(s) requirements, and shall trigger Near-RT RIC Platform to accommodate E2 / A1-EI operation(s) requirements.

If Near-RT RIC Platform is able to accommodate E2 / A1-EI operation(s) requirements, xApp replies A1 POLICY SETUP RESULT message with *A1 Policy Setup Result* IE set to *success* to Near-RT RIC Platform. And xApp should store the received *A1 Policy Type ID* IE, *A1 Policy ID* IE, *A1 Policy Object* IE as basic elements of A1 policy profile, and label the status of corresponding A1 policy as “enforced”.

#### 5.2.1.3 Unsuccessful operation

@startuml

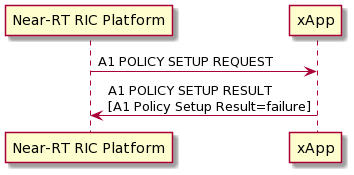
Participant rp as “Near-RT RIC Platform”

Participant ap as “xApp”

rp->ap: A1 POLICY SETUP REQUEST

ap->rp: A1 POLICY SETUP RESULT\n[A1 Policy Setup Result=failure]

@enduml



**Figure 5.2.1.3‑1 : A1 Policy Setup procedure, unsuccessful operation**

If xApp cannot transform the received A1 policy into appropriate E2 / A1-EI operation(s) requirements due to unsupported A1 policy type, xApp replies A1 POLICY SETUP RESULT message with *A1 Policy Setup Result* IE set to *failure* to Near-RT RIC Platform, possible with *Cause* IE set to *Not supported A1 policy type.*

If xApp cannot transform the received A1 policy into appropriate E2 / A1-EI operation(s) requirements due to violation of of the requirements as stated in 5.2.1.2.2, 5.2.2.2.2, 5.2.3.2.2, 5.2.4.2.2, 5.2.5.2.2, 5.2.6.2.2, 5.2.7.2.2 and 5.2.8.2.2 of [7], xApp replies A1 POLICY SETUP RESULT message with *A1 Policy Setup Result* IE set to *failure* to Near-RT RIC Platform, possible with *Cause* IE set to *Not allowed statements, restrictions and extensions.*

If Near-RT RIC Platform cannot accommodate E2 / A1-EI operation(s) requirements, xApp replies A1 POLICY SETUP RESULT message with *A1 Policy Setup Result* IE set to *failure* to Near-RT RIC Platform, possible with *Cause* IE set to *Near-RT RIC Platform cannot accommodate E2 / A1-EI operations requirements.*

#### 5.2.1.4 Abnormal conditions

If A1 policy indicated by *A1 Policy Type ID* IE, *A1 Policy ID* IE contained in A1 POLICY SETUP REQUEST message is being enforced by xApp, xApp replies A1 POLICY SETUP RESULT message with *A1 Policy Setup Result* IE set to *failure* to Near-RT RIC Platform, possible with *Cause* IE set to *A1 policy already enabled at xApp.*

### 5.2.2 A1 Policy Update procedure

#### 5.2.2.1 General

The purpose of the A1 Policy Update procedure is to update the execution of the transferred A1 policy.

#### 5.2.2.2 Successful operation

@startuml

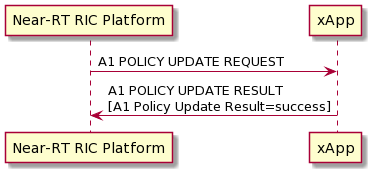
Participant rp as “Near-RT RIC Platform”

Participant ap as “xApp”

rp->ap: A1 POLICY UPDATE REQUEST

ap->rp: A1 POLICY UPDATE RESULT\n[A1 Policy Update Result=success]

@enduml



**Figure 5.2.2.2‑1 : A1 Policy Update procedure, successful operation**

The Near-RT RIC Platform initiates the procedure by sending A1 POLICY UPDATE REQUEST message to xApp.

The A1 POLICY UPDATE REQUEST message shall contain the information required by the xApp to update one A1 policy including *A1 Policy Type ID* IE, *A1 Policy ID* IE, and *A1 Policy Object* IE.

Upon reception of the A1 POLICY SETUP REQUEST message, the xApp shall transform the received A1 policy into appropriate E2 / A1-EI operation(s) requirements. If the transformed E2 / A1-EI operation(s) requirements are the same as before, xApp replies A1 POLICY UPDATE RESULT message with *A1 Policy Update Result* IE set to *success* to Near-RT RIC Platform; else, xApp shall trigger Near-RT RIC Platform to accommodate E2 / A1-EI operation(s) requirements.

If Near-RT RIC Platform to accommodate E2 / A1-EI operation(s) requirements succeeds, xApp replies A1 POLICY UPDATE RESULT message with *A1 Policy Update Result* IE set to *success* to Near-RT RIC Platform. And xApp should update the stored information of *A1 Policy Object* IE of A1 policy profile, and label the status of corresponding A1 policy as “enforced”.

#### 5.2.2.3 Unsuccessful operation

@startuml

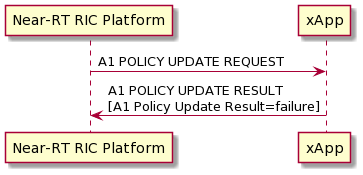
Participant rp as “Near-RT RIC Platform”

Participant ap as “xApp”

rp->ap: A1 POLICY UPDATE REQUEST

ap->rp: A1 POLICY UPDATE RESULT\n[A1 Policy Update Result=failure]

@enduml



**Figure 5.2.2.3‑1 : A1 Policy Update procedure, unsuccessful operation**

If xApp cannot transform the received A1 policy into appropriate E2 / A1-EI operation(s) requirements due to not support the corresponding A1 policy type, xApp replies A1 POLICY UPDATE RESULT message with *A1 Policy Update Result* IE set to *failure* to Near-RT RIC Platform, possible with *Cause* IE set to *Not supported A1 policy type.*

If xApp cannot transform the received A1 policy into appropriate E2 / A1-EI operation(s) requirements due to violation of the requirements as stated in 5.2.1.2.2, 5.2.2.2.2, 5.2.3.2.2, 5.2.4.2.2, 5.2.5.2.2, 5.2.6.2.2, 5.2.7.2.2 and 5.2.8.2.2 of [7], xApp replies A1 POLICY UPDATE RESULT message with *A1 Policy Update Result* IE set to *failure* to Near-RT RIC Platform, possible with *Cause* IE set to *Not allowed statements, restrictions and extensions.*

If Near-RT RIC Platform cannot accommodate E2 / A1-EI operation(s) requirements, xApp replies A1 POLICY UPDATE RESULT message with *A1 Policy Update Result* IE set to *failure* to Near-RT RIC Platform, possible with *Cause* IE set to *Near-RT RIC Platform cannot accommodate E2 / A1-EI operations requirements.*

#### 5.2.2.4 Abnormal conditions

If A1 policy indicated by *A1 Policy Type ID* IE, *A1 Policy ID* IE contained in A1 POLICY UPDATE REQUEST message is not enabled by xApp, xApp replies A1 POLICY UPDATE RESULT message with *A1 Policy Update Result* IE set to *failure* to Near-RT RIC Platform, possible with *Cause* IE set to *Not enabled A1 policy at xApp.*

### 5.2.3 A1 Policy Delete procedure

#### 5.2.3.1 General

The purpose of the A1 Policy Delete procedure is to terminate the execution of the transferred A1 policy.

#### 5.2.3.2 Successful operation

@startuml

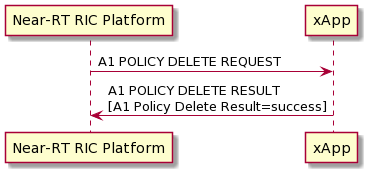
Participant rp as “Near-RT RIC Platform”

Participant ap as “xApp”

rp->ap: A1 POLICY DELETE REQUEST

ap->rp: A1 POLICY DELETE RESULT\n[A1 Policy Delete Result=success]

@enduml



**Figure 5.2.3.2‑1 : A1 Policy Delete procedure, successful operation**

The Near-RT RIC Platform initiates the procedure by sending A1 POLICY DELETE REQUEST message to xApp.

The A1 POLICY DELETE REQUEST message shall contain the information required by the xApp to delete one A1 policy including *A1 Policy Type ID* IE, *A1 Policy ID* IE.

Upon reception of the A1 POLICY DELETE REQUEST message, the xApp shall locate the E2 / A1-EI operation(s) associated with the A1 policy indicated by *A1 Policy Type ID* IE, *A1 Policy ID* IE contained in A1 POLICY DELETE REQUEST message, and shall trigger Near-RT RIC Platform to terminate the accommodation of located E2 / A1-EI operation(s).

No matter Near-RT RIC Platform to terminate the accommodation of located E2 / A1-EI operation(s) succeeds or not, xApp replies A1 POLICY DELETE RESULT message with *A1 Policy Delete Result* IE set to *success* to Near-RT RIC Platform. And xApp should delete the stored A1 policy profile associated.

#### 5.2.3.3 Unsuccessful operation

@startuml

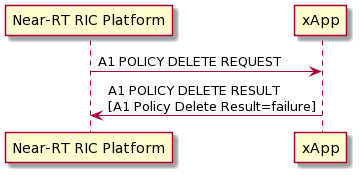
Participant rp as “Near-RT RIC Platform”

Participant ap as “xApp”

rp->ap: A1 POLICY DELETE REQUEST

ap->rp: A1 POLICY DELETE RESULT\n[A1 Policy Delete Result=failure]

@enduml



**Figure 5.2.3.3‑1 : A1 Policy Delete procedure, unsuccessful operation**

If xApp cannot locate the E2 / A1-EI operation(s) associated with the A1 policy indicated by *A1 Policy Type ID* IE, *A1 Policy ID* IE contained in A1 POLICY DELETE REQUEST message, xApp replies A1 POLICY DELETE RESULT message with *A1 Policy Delete Result* IE set to *failure* to Near-RT RIC Platform, possible with *Cause* IE set to *Not enabled A1 policy at xApp.*

#### 5.2.3.4 Abnormal conditions

Not applied.

## 5.3 Elements for A1-related API Communication

### 5.3.1 General

### 5.3.2 Message Functional Definition and Content

#### 5.3.2.1 A1 POLICY SETUP REQUEST

This message is sent by the RIC Platform and is used to request the destinated xApp to enable the execution of the nested A1 policy.

Direction: RIC Platform → xApp

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description | Criticality | Assigned Criticality |
| Message Type | M |  | 5.3.3.1 |  | YES | reject |
| A1 Policy Type ID | M |  | 5.3.3.2 |  | YES | reject |
| A1 Policy ID | M |  | 5.3.3.3 |  | YES | reject |
| A1 Policy Object | M |  | OCTET SRTING | Defined in [7]. | YES | reject |

#### 5.3.2.2 A1 POLICY SETUP RESULT

This message is sent by the xApp as response to the request to enable the execution of the nested A1 policy.

Direction: xApp → RIC Platform

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description | Criticality | Assigned Criticality |
| Message Type | M |  | 5.3.3.1 |  | YES | reject |
| A1 Policy Type ID | M |  | 5.3.3.2 |  | YES | reject |
| A1 Policy ID | M |  | 5.3.3.3 |  | YES | reject |
| A1 Policy Setup Result | M |  | 5.3.3.4 |  | YES | reject |
| Cause | O |  | 5.3.3.5 |  | YES | ignore |
| A1 Policy Object | C |  | OCTET SRTING | Defined in [7].  This IE shall present if the A1 Policy Setup Result IE is set to success.  When present, this IE is copied from the corresponding Request message. | YES | ignore |

#### 5.3.2.3 A1 POLICY UPDATE REQUEST

This message is sent by the RIC Platform and is used to request the destinated xApp to update the execution of the nested A1 policy.

Direction: RIC Platform → xApp

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description | Criticality | Assigned Criticality |
| Message Type | M |  | 5.3.3.1 |  | YES | reject |
| A1 Policy Type ID | M |  | 5.3.3.2 |  | YES | reject |
| A1 Policy ID | M |  | 5.3.3.3 |  | YES | reject |
| A1 Policy Object | M |  | OCTET SRTING | Defined in [7]. | YES | reject |

#### 5.3.2.4 A1 POLICY UPDATE RESULT

This message is sent by xApp as response to the request to update the execution of the nested A1 policy.

Direction: xApp → RIC Platform

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description | Criticality | Assigned Criticality |
| Message Type | M |  | 5.3.3.1 |  | YES | reject |
| A1 Policy Type ID | M |  | 5.3.3.2 |  | YES | reject |
| A1 Policy ID | M |  | 5.3.3.3 |  | YES | reject |
| A1 Policy Update Result | M |  | 5.3.3.4 |  | YES | reject |
| Cause | O |  | 5.3.3.5 |  | YES | ignore |
| A1 Policy Object | C |  | OCTET SRTING | Defined in [7].  This IE shall present if the A1 Policy Update Result IE is set to success.  When present, this IE is copied from the corresponding Request message. | YES | ignore |

#### 5.3.2.5 A1 POLICY DELETE REQUEST

This message is sent by the RIC Platform and is used to request the destinated xApp to terminate the execution of the nested A1 policy.

Direction: RIC Platform → xApp

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description | Criticality | Assigned Criticality |
| Message Type | M |  | 5.3.3.1 |  | YES | reject |
| A1 Policy Type ID | M |  | 5.3.3.2 |  | YES | reject |
| A1 Policy ID | M |  | 5.3.3.3 |  | YES | reject |

#### 5.3.2.6 A1 POLICY DELETE RESULT

This message is sent by xApp as response to the request to terminate the execution of the nested A1 policy.

Direction: xApp → RIC Platform

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description | Criticality | Assigned Criticality |
| Message Type | M |  | 5.3.3.1 |  | YES | reject |
| A1 Policy Type ID | M |  | 5.3.3.2 |  | YES | reject |
| A1 Policy ID | M |  | 5.3.3.3 |  | YES | reject |
| A1 Policy Delete Result | M |  | 5.3.3.4 |  | YES | reject |
| Cause | O |  | 5.3.3.5 |  | YES | ignore |

#### 5.3.2.7 A1 POLICY QUERY REQUEST

#### 5.3.2.8 A1 POLICY QUERY RESULT

#### 5.3.2.9 A1 POLICY STATUS UPDATE

#### 5.3.2.10 A1 POLICY STATUS UPDATE ACK

#### 5.3.2.11 A1 POLICY STATUS UPDATE ERROR

#### 5.3.2.12 A1-EI SUBSCRIPTION SETUP REQUEST

#### 5.3.2.13 A1-EI SUBSCRIPTION SETUP RESULT

#### 5.3.2.14 A1-EI SUBSCRIPTION UPDATE REQUEST

#### 5.3.2.15 A1-EI SUBSCRIPTION UPDATE RESULT

#### 5.3.2.16 A1-EI SUBSCRIPTION DELETE REQUEST

#### 5.3.2.17 A1-EI SUBSCRIPTION DELETE RESULT

#### 5.3.2.18 A1-EI DELIVERY

#### 5.3.2.19 A1-EI QUERY REQUEST

#### 5.3.2.20 A1-EI QUERY RESULT

### 5.3.3 Information Element Definitions

#### 5.3.3.1 Message Type

The Message Type IE identifies the message being sent. It is mandatory for all messages.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE Type and Reference | Semantics Description |
| Message Type |  |  |  |  |
| >Procedure Type | M |  | CHOICE (A1 policy setup, A1 policy update, A1 policy delete, …) |  |
| >Type of Message | M |  | CHOICE (Request, Result, …) | Refer to Table 5.1.2-3 for applicable choices for each procedure type |

#### 5.3.3.2 A1 Policy Type ID

Defined in 10.1.5.

#### 5.3.3.3 A1 Policy ID

The A1 Policy ID IE identifies one A1 policy under one A1 policy type.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE Type and Reference | Semantics Description |
| A1 Policy ID | M |  | STRING |  |

#### 5.3.3.4 Result

The Result IE indicates the procedure succeeds or not.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE Type and Reference | Semantics Description |
| Result | M |  | ENUMERATED (success, failure, …) |  |

#### 5.3.3.5 Cause

The Cause IE indicates the reason for a particular event in A1-related API.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE Type and Reference | Semantics Description |
| CHOICE *Cause Group* | M |  |  |  |
| >*A1-related API Layer* |  |  |  |  |
| >>A1-related API Layer Cause | M |  | ENUMERATED (Unspecified A1-related API layer reason,  Not supported A1 policy type,  Not allowed statements, restrictions and extensions,  Near-RT RIC Platform cannot accommodate E2 / A1-EI operations requirements,  A1 policy already enabled at xApp,  Not enabled A1 policy at xApp, …) |  |
| >*Transport Layer* |  |  |  |  |
| >>Transport Layer Cause | M |  | ENUMERATED (Unspecified transport layer reason, Transport Resource Unavailable, …) |  |
| >*Protocol* |  |  |  |  |
| >>Protocol Cause | M |  | ENUMERATED (Unspecified protocol reason,  Transfer Syntax Error,  Abstract Syntax Error (Reject),  Abstract Syntax Error (Ignore and Notify),  Message not Compatible with Receiver State,  Semantic Error, Abstract Syntax Error (Falsely Constructed Message), …) |  |
| >*Misc* |  |  |  |  |
| >>Miscellaneous Cause | M |  | ENUMERATED (Unspecified miscellaneous reason,  Control Processing Overload Hardware failure,  O&M intervention, …) |  |

The meaning of the different cause values is described in the following table. In general, “not supported” cause values indicate that the related capability is missing. On the other hand, “not available” cause values indicate that the related capability is present, but insufficient resources were available to perform the requested action.

|  |  |
| --- | --- |
| A1-related API Layer cause | Meaning |
| Unspecified A1-related API layer reason | Sent for A1-related API layer cause when none of the specified cause values applies. |
| Not supported A1 policy type | The A1 policy type is not supported. |
| Not allowed statements, restrictions and extensions | The A1 policy object is illegal due to violation of statements, restrictions and extensions required for corresponding A1 policy type. |
| Near-RT RIC Platform cannot accommodate E2 / A1-EI operations requirements | The RIC platform failed to setup xApp’s requested E2 / A1-EI operation(s). |
| A1 policy already enabled at xApp | The A1 policy indicated by A1 Policy Type ID and A1 Policy ID is enforced already. |
| Not enabled A1 policy at xApp | The A1 policy indicated by A1 Policy Type ID and A1 Policy ID is not enforced. |

|  |  |
| --- | --- |
| Transport Layer cause | Meaning |
| Unspecified transport layer reason | Sent for transport layer cause when none of the specified cause values applies. |
| Transport Resource Unavailable | The required transport resources are not available. |

|  |  |
| --- | --- |
| Protocol cause | Meaning |
| Unspecified protocol reason | Sent for protocol cause when none of the specified cause values applies. |
| Transfer Syntax Error | The received message included a transfer syntax error. |
| Abstract Syntax Error (Reject) | The received message included an abstract syntax error and the concerning criticality indicated “reject”. |
| Abstract Syntax Error (Ignore And Notify) | The received message included an abstract syntax error and the concerning criticality indicated “ignore and notify”. |
| Message Not Compatible With Receiver State | The received message was not compatible with the receiver state. |
| Semantic Error | The received message included a semantic error. |
| Abstract Syntax Error (Falsely Constructed Message) | The received message contained IEs or IE groups in wrong order or with too many occurrences. |

|  |  |
| --- | --- |
| Miscellaneous cause | Meaning |
| Unspecified miscellaneous reason | Sent for miscellaneous cause when none of the specified cause values applies. |
| Control Processing Overload | Control processing overload |
| Hardware Failure | Action related to hardware failure |
| O&M Intervention | The action is due to O&M intervention. |

### 5.3.4 Message and Information Element Abstract Syntax

#### 5.3.4.1 Solution 1: gRPC with Protocol Buffers

##### 5.3.4.1.1 Usage of gRPC with Protocol Buffers

This solution uses Protocol Buffers as message encoding, and also as the Interface Definition Language (IDL) for gRPC [5].

Table 5.3.4.1.1-1 lists the mapping of A1-related APIs and corresponding gRPC services.

**Table 5.3.4.1.1-1: Mapping of A1-related APIs and gRPC services**

|  |  |
| --- | --- |
| **API Name** | **gRPC service(s)** |
| A1\_Policy\_API | A1\_Policy\_API |

A procedure of an A1-related API is mapped to a *gRPC method.*

Table 5.3.4.1.1-2 lists the A1-related API specifications included in this document.

Table 5.3.4.1.1-2: List of A1-related APIs abstract syntax

|  |  |  |  |
| --- | --- | --- | --- |
| API Name / Contents | Clause | Version | File Name |
| A1\_Policy\_API | 5.3.4.1.2.1 | 1.0.0 | a1\_policy\_api\_ver\_1\_0\_0.proto |
| A1-related API common information elements | 5.3.4.1.3 | 1.0.0 | a1\_related\_apis\_common\_ies\_ver\_1\_0\_0.proto |
| Near-RT RIC API common information elements | 10.2.1 | 1.0.0 | near\_rt\_ric\_apis\_common\_ies\_ver\_1\_0\_0.proto |

##### 5.3.4.1.2 API Definitions

###### 5.3.4.1.2.1 A1\_Policy\_API

syntax = "proto3";

package ricapi.a1\_policy\_api.v1;

import "near\_rt\_ric\_apis\_common\_ies\_ver\_1\_0\_0.proto";

import "a1\_related\_apis\_common\_ies\_ver\_1\_0\_0.proto";

// gRPC service and rpc definition

service A1\_Policy\_API {

rpc A1PolicySetupProc(A1PolicySetupInitMsg) returns (A1PolicySetupOutMsg) {};

rpc A1PolicyUpdateProc(A1PolicyUpdateInitMsg) returns (A1PolicyUpdateOutMsg) {};

rpc A1PolicyDeleteProc(A1PolicyDeleteInitMsg) returns (A1PolicyDeleteOutMsg) {};

}

message A1PolicySetupInitMsg {

optional A1PolicySetupRequest a1\_policy\_setup\_request = 1;

}

message A1PolicySetupOutMsg {

oneof type\_of\_message {

A1PolicySetupResult a1\_policy\_setup\_result = 1;

}

}

message A1PolicyUpdateInitMsg {

optional A1PolicyUpdateRequest a1\_policy\_update\_request = 1;

}

message A1PolicyUpdateOutMsg {

oneof type\_of\_message {

A1PolicyUpdateResult a1\_policy\_update\_result = 1;

}

}

message A1PolicyDeleteInitMsg {

optional A1PolicyDeleteRequest a1\_policy\_delete\_request = 1;

}

message A1PolicyDeleteOutMsg {

oneof type\_of\_message {

A1PolicyDeleteResult a1\_policy\_delete\_result = 1;

}

}

// ---------------- messages of a1\_policy\_API ----------------

// -------- messages of A1 Policy Setup procedure --------

message A1PolicySetupRequest {

optional near\_rt\_ric\_apis\_common\_ies.v1.A1PolicyTypeId a1\_policy\_type\_id = 1;

optional a1\_related\_apis\_common\_ies.v1.A1PolicyId a1\_policy\_id = 2;

optional a1\_related\_apis\_common\_ies.v1.A1PolicyObject a1\_policy\_object = 3;

}

message A1PolicySetupResult {

optional near\_rt\_ric\_apis\_common\_ies.v1.A1PolicyTypeId a1\_policy\_type\_id = 1;

optional a1\_related\_apis\_common\_ies.v1.A1PolicyId a1\_policy\_id = 2;

optional a1\_related\_apis\_common\_ies.v1.Result a1\_policy\_setup\_result = 3;

optional a1\_related\_apis\_common\_ies.v1.Cause cause = 4;

optional a1\_related\_apis\_common\_ies.v1.A1PolicyObject a1\_policy\_object = 5;

}

// -------- messages of A1 Policy Update procedure --------

message A1PolicyUpdateRequest {

optional near\_rt\_ric\_apis\_common\_ies.v1.A1PolicyTypeId a1\_policy\_type\_id = 1;

optional a1\_related\_apis\_common\_ies.v1.A1PolicyId a1\_policy\_id = 2;

optional a1\_related\_apis\_common\_ies.v1.A1PolicyObject a1\_policy\_object = 3;

}

message A1PolicyUpdateResult {

optional near\_rt\_ric\_apis\_common\_ies.v1.A1PolicyTypeId a1\_policy\_type\_id = 1;

optional a1\_related\_apis\_common\_ies.v1.A1PolicyId a1\_policy\_id = 2;

optional a1\_related\_apis\_common\_ies.v1.Result a1\_policy\_update\_result = 3;

optional a1\_related\_apis\_common\_ies.v1.Cause cause = 4;

optional a1\_related\_apis\_common\_ies.v1.A1PolicyObject a1\_policy\_object = 5;

}

// -------- messages of A1 Policy Delete procedure --------

message A1PolicyDeleteRequest {

optional near\_rt\_ric\_apis\_common\_ies.v1.A1PolicyTypeId a1\_policy\_type\_id = 1;

optional a1\_related\_apis\_common\_ies.v1.A1PolicyId a1\_policy\_id = 2;

}

message A1PolicyDeleteResult {

optional near\_rt\_ric\_apis\_common\_ies.v1.A1PolicyTypeId a1\_policy\_type\_id = 1;

optional a1\_related\_apis\_common\_ies.v1.A1PolicyId a1\_policy\_id = 2;

optional a1\_related\_apis\_common\_ies.v1.Result a1\_policy\_delete\_result = 3;

optional a1\_related\_apis\_common\_ies.v1.Cause cause = 4;

}

##### 5.3.4.1.3 Information Element Definitions

syntax = "proto3";

package ricapi.a1\_related\_apis\_common\_ies.v1;

import "near\_rt\_ric\_apis\_common\_ies\_ver\_1\_0\_0.proto";

// ----------------------------- Information Elements -----------------------------

// --------------A--------------

message A1PolicyObject {

optional string value = 1;

}

message A1PolicyId {

optional string value = 1;

}

// --------------C--------------

message Cause {

oneof cause\_group {

A1RelatedApiLayerCause a1\_related\_api\_layer\_cause = 1;

TransportLayerCause transport\_layer\_cause = 2;

ProtocolLayerCause protocol\_layer\_cause = 3;

MiscCause misc\_cause = 4;

}

enum A1RelatedApiLayerCause {

A1\_RELATED\_API\_LAYER\_UNSPECIFIED = 0;

NOT\_SUPPORTED\_A1\_POLICY\_TYPE = 1;

NOT\_ALLOWED\_STATEMENTS\_RESTRICTIONS\_AND\_EXTENSIONS = 2;

NEAR\_RT\_RIC\_PLATFORM\_CANNOT\_ACCOMMODATE\_E2\_A1\_EI\_OPERATIONS\_REQUIREMENTS = 3;

A1\_POLICY\_ALREADY\_ENABLED\_AT\_XAPP = 4;

NOT\_ENABLED\_A1\_POLICY\_AT\_XAPP = 5;

}

enum TransportLayerCause {

TRANSPORT\_UNSPECIFIED = 0;

TRANSPORT\_RESOURCE\_UNAVAILABLE = 1;

}

enum ProtocolLayerCause {

PROTOCOL\_UNSPECIFIED = 0;

TRANSFER\_SYNTAX\_ERROR = 1;

ABSTRACT\_SYNTAX\_ERROR\_REJECT = 2;

ABSTRACT\_SYNTAX\_ERROR\_IGNORE\_AND\_NOTIFY = 3;

MESSAGE\_NOT\_COMPATIBLE\_WITH\_RECEIVER\_STATE = 4;

SEMANTIC\_ERROR = 5;

ABSTRACT\_SYNTAX\_ERROR\_FALSELY\_CONSTRUCTED\_MESSAGE = 6;

}

enum MiscCause {

MISC\_UNSPECIFIED = 0;

CONTROL\_PROCESSING\_OVERLOAD = 1;

HARDWARE\_FAILURE = 2;

OAM\_INTERVENTION = 3;

}

}

// --------------R--------------

enum Result {

RESULT\_SUCCESS = 0;

RESULT\_FAILURE = 1;

}

### 5.3.5 Message Transfer Syntax

#### 5.3.5.1 Solution 1: gRPC with Protocol Buffers

The message transfer syntax for Solution 1 is serialized Protocol Buffers [4].

# 6 E2-related API

## 6.1 Overview

### 6.1.1 Protocol Stack

In this version of specification, the default protocol stack for the E2 related APIs is shown on Figure 6.1.1-1.

Note: the network layer for security protection is FFS.

形状

低可信度描述已自动生成

**Figure 6.1.1-1: Protocol stack for E2 related APIs**

The E2 related APIs use SCTP with Protocol Buffers [4] as the application layer serialization protocol.

### 6.1.2 List of E2 related APIs, procedures and messages

The E2 related API procedures introduced in section 9.3 of [2] are categorized under the E2 related APIs. These APIs include:

* E2 Subscription API: enabling xApp to setup, modify and delete E2 subscriptions with the Near-RT RIC platform.
* E2 Indication API: enabling xApp to receive E2 indications associated with an E2 subscription.
* E2 Control API: enabling xApp to consume E2 control service from the Near-RT RIC platform.
* E2 Guidance API: enabling xApp to acquire E2 related guidance from the Near-RT RIC platform and mitigate potential conflicts with other xApps.

These APIs are summarized in Table 6.1.2-1.

Table 6.1.2-1: List of E2 related APIs

|  |  |
| --- | --- |
| API Name | Description |
| E2\_Subscription\_API | Near-RT RIC platform service for subscription of E2 REPORT, INSERT and POLICY services |
| E2\_Indication\_API | Near-RT RIC platform service for delivery of subscribed information associated with E2 REPORT and INSERT services |
| E2\_Control\_API | Near-RT RIC platform service for E2 CONTROL |
| E2\_Guidance\_API | Near-RT RIC platform service for E2 related guidance |

The table 6.1.2-2 lists the API procedures for each E2 related API.

Table 6.1.2-2: List of E2 related API procedures

|  |  |  |
| --- | --- | --- |
| API Name | API procedure | Communication type |
| E2\_Subscription\_API | E2 Subscription | Request/Response |
| E2 Subscription Delete | Request/Response |
| E2 Subscription Delete Query | Request/Response |
| E2 Subscription Delete Notification | Notify |
| E2\_Indication\_API | E2 Indication | Notify |
| E2\_Control\_API | E2 Control | Request/Response |
| E2\_Guidance\_API | E2 Guidance | Request/Response |
| E2 Guidance Modification | Notify |

The table 6.1.2-3 lists the messages for each E2 related API procedure.

Table 6.1.2-3: List of E2 related API messages

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **API Procedure** | **Initiated by** | **Initiating Message** | **Outcome Message** | | |
| **Successful Outcome** | **Unsuccessful Outcome 1** | **Unsuccessful Outcome 2** |
| E2 Subscription | xApp | E2 SUBSCRIPTION REQUEST | E2 SUBSCRIPTION SUCCESS | E2 SUBSCRIPTION FAILURE | E2 SUBSCRIPTION REJECT |
| E2 Subscription Delete | xApp | E2 SUBSCRIPTION DELETE REQUEST | E2 SUBSCRIPTION DELETE SUCCESS | E2 SUBSCRIPTION DELETE FAILURE | E2 SUBSCRIPTION DELETE REJECT |
| E2 Subscription Delete Query | Near-RT RIC platform | E2 SUBSCRIPTION DELETE REQUIRED | E2 SUBSCRIPTION DELETE ACCEPT | E2 SUBSCRIPTION DELETE DECLINE | - |
| E2 Subscription Delete Notification | Near-RT RIC platform | E2 SUBSCRIPTION DELETE NOTIFICATION | - | - | - |
| E2 Indication | Near-RT RIC platform | E2 INDICATION PUSH | - | E2 INDICATION FAILURE | - |
| E2 Control | xApp | E2 CONTROL REQUEST | E2 CONTROL SUCCESS | E2 CONTROL FAILURE | E2 CONTROL REJECT |
| E2 Guidance | xApp | E2 GUIDANCE REQUEST | E2 GUIDANCE SUCCESS | - | - |
| E2 Guidance Modification | Near-RT RIC platform | E2 GUIDANCE MODIFICATION | - | - | - |

## 6.2 E2-related API Procedures

### 6.2.1 E2\_Subscription\_API procedures

#### 6.2.1.1 E2 Subscription Procedure

##### 6.2.1.1.1 General

This procedure is used by an xApp to request an E2 subscription to Near-RT RIC platform.

##### 6.2.1.1.2 Successful Operation

@startuml

Skin rose

skinparam ParticipantPadding 50

skinparam BoxPadding 10

skinparam lifelineStrategy solid

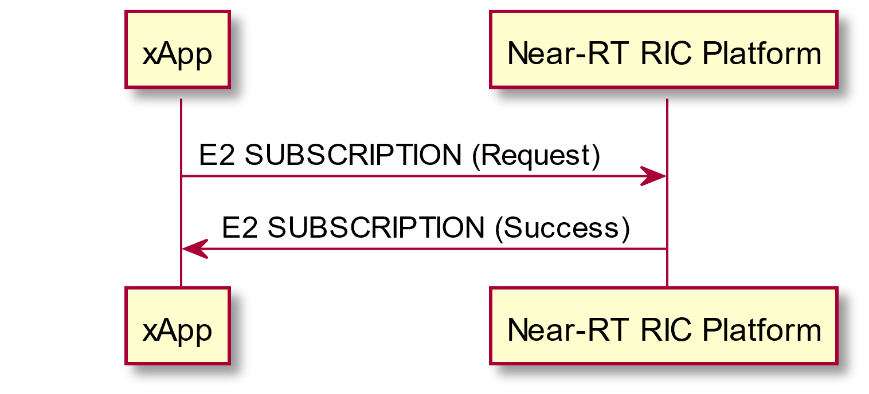
participant "xApp" as app

participant "Near-RT RIC Platform" as near

app -> near: E2 SUBSCRIPTION (Request)

near -> app: E2 SUBSCRIPTION (Success)

@enduml



**Figure 6.2.1.1.2-1: E2 Subscription procedure, successful operation**

The xApp initiates the procedure by sending an E2 SUBSCRIPTION (Request) message to the Near-RT-RIC platform. The message contains a unique *Procedure Transaction ID* IE assigned by the initiating xApp, the *Global E2 Node ID* IE, the *RAN Function ID* IE, and the *E2 Subscription Details* IE including one or more requested E2 actions.

Note: When execution order of multiple actions is not concerned, it is recommended that an E2 Subscription procedure contains a single action, to allow efficient reuse of E2 subscriptions.

At reception of the E2 SUBSCRIPTION (Request) message, the Near-RT-RIC platform shall:

- Search existing E2 subscription(s) for matched *Global E2 Node ID* IE, *RAN Function ID* IE and matched contents in *E2 Subscription Details* IE. If a matched E2 subscription exists, add the initiating xApp as a new subscriber of the E2 subscription.

- Initiate appropriate E2AP procedure(s) to establish RIC subscription with the target E2 Node, if existing E2 subscriptions are insufficient to fulfill the request.

If the subscription request is successfully fulfilled, the Near-RT RIC platform shall send an E2 SUBSCRIPTION (Success) message back to the xApp. Similar to [3], a successfully fulfilled request shall be sent if the E2 event trigger and at least one requested E2 action are admitted.

In the E2 SUBSCRIPTION (Success) message, Near-RT RIC platform shall include the associated *Procedure Transaction ID* IE, *Global E2 Node ID* IE, *RAN Function ID* IE and an *E2 Request ID* IE assigned by the Near-RT RIC platform. The message shall also indicate the admitted E2 action(s) in the *Actions Admitted List* IE, each with the associated *E2 Action ID* IE assigned by the Near-RT RIC platform. The same *E2 Request ID* IE and *E2 Action ID* IE shall be present in any subsequent E2 INDICATION message(s). The message shall indicate, if exist, the E2 action(s) not admitted in the *Actions Not Admitted List* IE with appropriate cause(s).

##### 6.2.1.1.3 Unsuccessful Operation

@startuml

Skin rose

skinparam ParticipantPadding 50

skinparam BoxPadding 10

skinparam lifelineStrategy solid

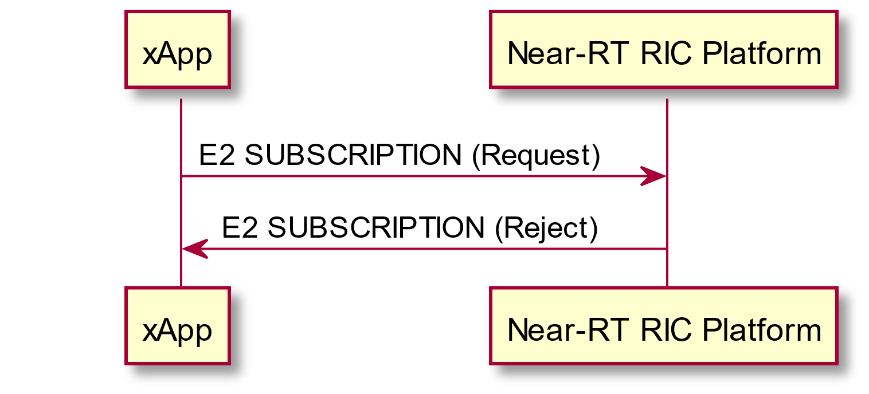
participant "xApp" as app

participant "Near-RT RIC Platform" as near

app -> near: E2 SUBSCRIPTION (Request)

near -> app: E2 SUBSCRIPTION (Reject)

@enduml



**Figure 6.2.1.1.3-1: E2 Subscription procedure, unsuccessful operation (Reject)**

If the subscription request is rejected by the Near-RT RIC platform, an E2 SUBSCRIPTION (Reject) message shall be sent back to the xApp. The message shall include associated *Procedure Transaction ID* IE which is replicated from the corresponding request, the *Global E2 Node ID* IE, the *RAN Function ID* IE and appropriate *Cause* IE.

@startuml

Skin rose

skinparam ParticipantPadding 50

skinparam BoxPadding 10

skinparam lifelineStrategy solid

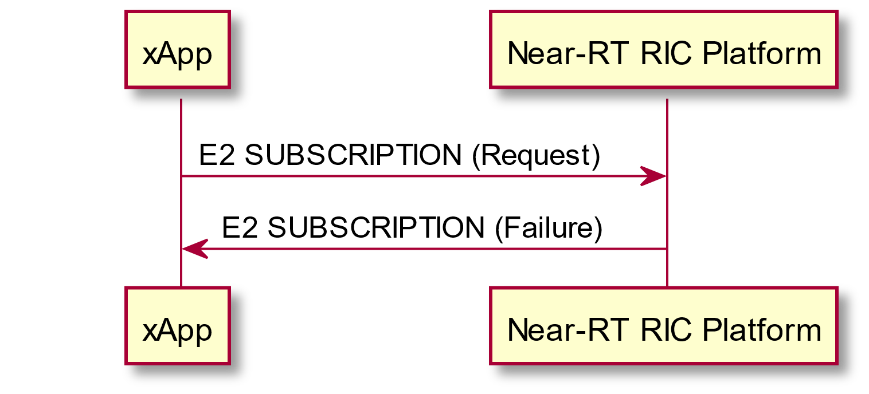
participant "xApp" as app

participant "Near-RT RIC Platform" as near

app -> near: E2 SUBSCRIPTION (Request)

near -> app: E2 SUBSCRIPTION (Failure)

@enduml



**Figure 6.2.1.1.3-2: E2 Subscription procedure, unsuccessful operation (Failure)**

If the subscription request fails due to failure in E2AP procedure, the Near-RT RIC platform shall send an E2 SUBSCRIPTION (Failure) message back to the xApp. The message shall include the associated *Procedure Transaction ID* IE which is replicated from the corresponding request, *Global E2 Node ID* IE, *RAN Function ID* IE and appropriate cause.

##### 6.2.1.1.4 Abnormal Conditions

Not applied.

#### 6.2.1.2 E2 Subscription Delete Procedure, xApp initiated

##### 6.2.1.2.1 General

This procedure is used by an xApp to unsubscribe from an E2 subscription on Near-RT RIC platform.

##### 6.2.1.2.2 Successful Operation

@startuml

Skin rose

skinparam ParticipantPadding 50

skinparam BoxPadding 10

skinparam lifelineStrategy solid

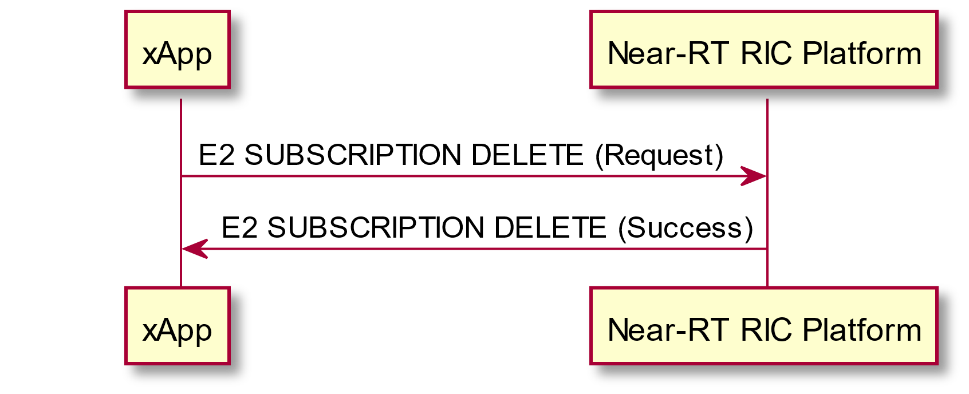
participant "xApp" as app

participant "Near-RT RIC Platform" as near

app -> near: E2 SUBSCRIPTION DELETE (Request)

near -> app: E2 SUBSCRIPTION DELETE (Success)

@enduml



**Figure 6.2.1.2.2-1: E2 Subscription Delete procedure (xApp initiated), successful operation**

The xApp initiates the procedure by sending the E2 SUBSCRIPTION DELETE (Request) message to the Near-RT-RIC platform. The message contains the *Global E2 Node ID* IE and the *E2 Request ID* IE to indicate the target E2 subscription.

At reception of the E2 SUBSCRIPTION DELETE (Request) message, the Near-RT-RIC platform shall, if the delete request is accepted by the Near-RT-RIC platform:

- Check for other subscriber(s) of the target E2 subscription. If such subscriber(s) exists, the E2 subscription shall be maintained by the Near-RT RIC platform but no longer serve the initiating xApp; if not, the E2 subscription shall be deleted.

- Initiate appropriate E2AP procedure(s) with the target E2 Node, if necessary, to map the change in the target E2 subscription to its corresponding RIC subscription.

Upon success of the above processing for the target E2 subscription, the Near-RT RIC platform shall stop delivering its associated E2 INDICATION message(s) to the initiating xApp and respond the xApp with an E2 SUBSCRIPTION DELETE(Success) message. The E2 SUBSCRIPTION DELETE (Success) message shall include the same *Global E2 Node ID* IE and *E2 Request ID* IE from the request message.

##### 6.2.1.2.3 Unsuccessful Operation

@startuml

Skin rose

skinparam ParticipantPadding 50

skinparam BoxPadding 10

skinparam lifelineStrategy solid

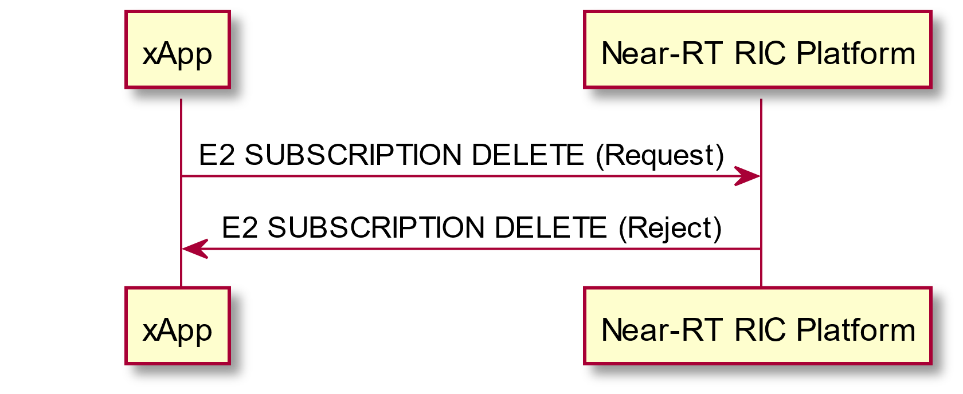
participant "xApp" as app

participant "Near-RT RIC Platform" as near

app -> near: E2 SUBSCRIPTION DELETE (Request)

near -> app: E2 SUBSCRIPTION DELETE (Reject)

@enduml



**Figure 6.2.1.2.3-1: E2 Subscription Delete procedure (xApp initiated), unsuccessful operation (Reject)**

If the Near-RT RIC platform rejects the request message, an E2 SUBSCRIPTION (Reject) message shall be sent back to the initiating xApp. The message shall include associated *Procedure Transaction ID* IE which is replicated from the corresponding request, the *Global E2 Node ID* IE, the *RAN Function ID* IE and appropriate *Cause* IE.

@startuml

Skin rose

skinparam ParticipantPadding 50

skinparam BoxPadding 10

skinparam lifelineStrategy solid

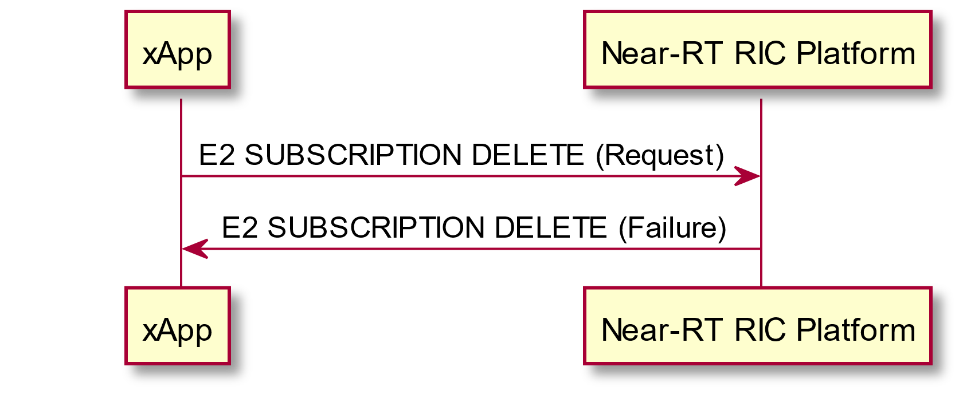
participant "xApp" as app

participant "Near-RT RIC Platform" as near

app -> near: E2 SUBSCRIPTION DELETE (Request)

near -> app: E2 SUBSCRIPTION DELETE (Failure)

@enduml



**Figure 6.2.1.2.3-2: E2 Subscription Delete procedure (xApp initiated), unsuccessful operation (Failure)**

If the associated E2AP procedure(s) fails for the E2 subscription, the Near-RT RIC platform shall send an E2 SUBSCRIPTION DELETE (Failure) message back to the initiating xApp. The message shall include associated *Procedure Transaction ID* IE which is replicated from the corresponding request, the *Global E2 Node ID* IE, the *RAN Function ID* IE and appropriate *Cause* IE.

##### 6.2.1.2.4 Abnormal Conditions

Not applied.

### 6.2.2 E2\_Indication\_API procedures

#### 6.2.2.1 E2 Indication Procedure

##### 6.2.2.1.1 General

This procedure is used by the Near-RT RIC platform to deliver to target xApp the subscribed information associated with an E2 subscription. The procedure is triggered by the reception of an E2AP RIC INDICATION message from an E2 Node.

##### 6.2.2.1.2 Successful Operation

@startuml

Skin rose

skinparam ParticipantPadding 50

skinparam BoxPadding 10

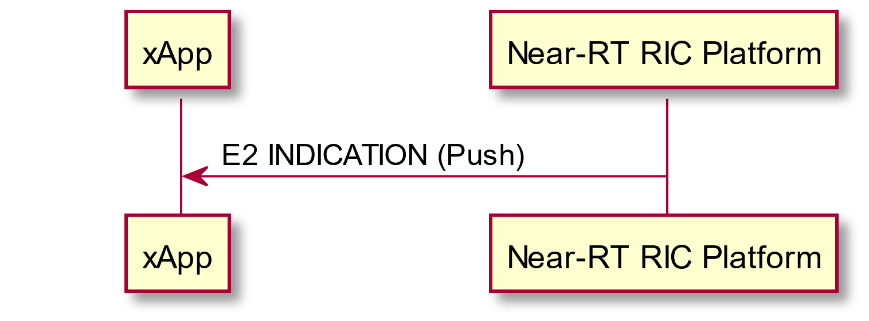
skinparam lifelineStrategy solid

participant "xApp" as app

participant "Near-RT RIC Platform" as near

near -> app: E2 INDICATION (Push)

@enduml



**Figure 6.2.2.1.2-1: E2 Indication procedure, successful operation**

The Near-RT-RIC platform initiates the procedure by sending the E2 INDICATION message containing the *Global E2 Node ID* IE, the *E2 Request ID* IE, the *E2 Action ID* IE to the target xApp. The message shall also contain the *E2 Indication Type* IE, the *E2 Indication Header* IE and the *E2 Indication Message* IE to carry the information from corresponding fields in the E2AP RIC INDICATION message. The *E2 Indication SN* IE and the *E2 Call process ID* IE shall also be included, if corresponding fields are present in the E2AP RIC INDICATION message.

At reception of the E2 INDICATION message an xApp shall:

- Use the *Global E2 Node ID* IE, the *E2 Request ID* IE and the *E2 Action ID* IE to associate the received message to its previous request for E2 subscription. The E2 Request ID and the E2 Action ID have been assigned in the E2 SUBSCRIPTION (Success) message for each requested E2 action in the E2 SUBSCRIPTION (Request) message.

- Process the information carried in the *E2 Indication SN* IE, the *E2 Indication Type* IE, the *E2 Indication Header* IE, the *E2 Indication Message* IE and the *E2 Call process ID* IE.

##### 6.2.2.1.3 Unsuccessful Operation

@startuml

Skin rose

skinparam ParticipantPadding 50

skinparam BoxPadding 10

skinparam lifelineStrategy solid

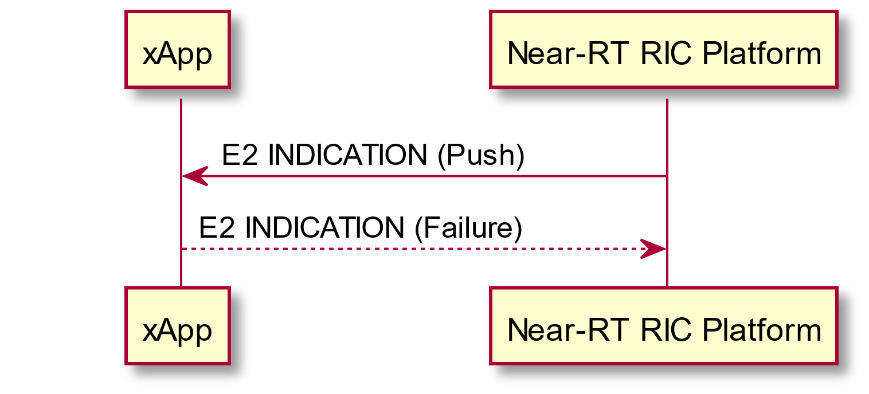
participant "xApp" as app

participant "Near-RT RIC Platform" as near

near -> app: E2 INDICATION (Push)

app --> near: E2 INDICATION (Failure)

@enduml



**Figure 6.2.2.1.3-1: E2 Indication procedure, unsuccessful operation (Failure)**

If an error is detected in the received message, the xApp may send E2 INDICATION (Failure) message back to the Near-RT RIC platform. The message shall include the associated *Global E2 Node ID* IE, *E2 Request ID* IE, *RAN Function ID* IE, *E2 Action ID* IE, and appropriate cause. The *E2 Indication SN* IE shall also be included if present in the corresponding E2 INDICATION (Push) message.

##### 6.2.2.1.4 Abnormal Conditions

Not applied.

### 6.2.3 E2\_Control\_API procedures

#### 6.2.3.1 E2 Control procedure

##### 6.2.3.1.1 General

This procedure is used by an xApp to request for E2 control service to Near-RT RIC platform.

##### 6.2.3.1.2 Successful Operation

@startuml

Skin rose

skinparam ParticipantPadding 50

skinparam BoxPadding 10

skinparam lifelineStrategy solid

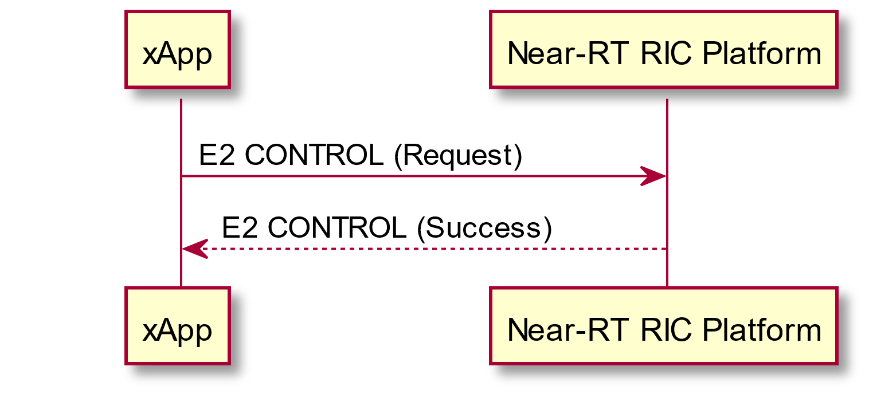
participant "xApp" as app

participant "Near-RT RIC Platform" as near

app -> near: E2 CONTROL (Request)

near 🡪 app: E2 CONTROL (Success)

@enduml



**Figure 6.2.3.1.2-1: E2 Contol procedure, successful operation**

The xApp initiates the procedure by sending the E2 CONTROL (Request) message to the Near-RT-RIC platform, containing the *Global E2 Node ID* IE, the *RAN Function ID* IE, optionally the *E2 Call Process ID* IE, *E2 Control Header* IE, *E2 Control Message* IE, optionally the *E2 Control Ack Request* IE and optionally the *E2 Control Timer* IE.

At reception of the E2 CONTROL (Request) message, the Near-RT-RIC platform shall:

- Evaluate whether to reject the request due to potential control conflict, when such evaluation is required by conflict mitigation related configuration.

- Initiate appropriate E2AP RIC Control procedure with the target E2 Node, if the request is not rejected by the Near-RT-RIC platform.

The Near-RT-RIC platform may use the value in the *E2 Control Timer* IE, if present, as a one-time value of the E2AP timer TRICcontrol in Clause 9.5 of [3], which is valid only for this transaction.

If the control request is successfully performed, and if the optional *E2 Control Ack Request* IE in the request is absent or set to “Ack”, an E2 CONTROL (Success) message shall be responded to the initiating xApp. The E2 CONTROL (Success) message shall include the *Global E2 Node ID* IE, the *RAN Function ID* IE, optionally the *E2 Call Process ID* IE, and optionally the *E2 Control Outcome* IE. The xApp may use the information contained in the optional *E2 Control Outcome* IE to determine subsequent actions.

If the control request is successfully performed, and if the optional *E2 Control Ack Request* IE in the request is set to “NoAck”, an E2 CONTROL (Success) message is not needed and xApp may consider it success.

##### 6.2.3.1.3 Unsuccessful Operations

@startuml

Skin rose

skinparam ParticipantPadding 50

skinparam BoxPadding 10

skinparam lifelineStrategy solid

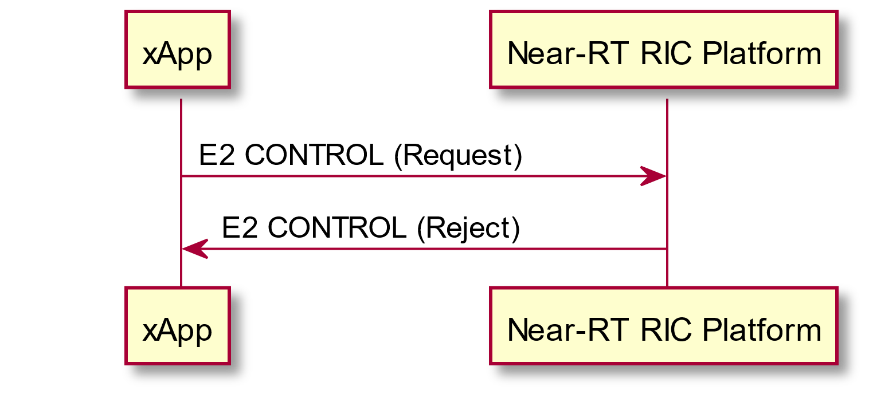
participant "xApp" as app

participant "Near-RT RIC Platform" as near

app -> near: E2 CONTROL (Request)

near -> app: E2 CONTROL (Reject)

@enduml



**Figure 6.2.3.1.3-1: E2 Control procedure, unsuccessful operation (Reject)**

If the request is rejected by the Near-RT RIC platform, an E2 CONTROL (Reject) message shall be responded to the initiating xApp with appropriate *Cause* IE. The xApp may use the information contained in the *Cause* IE to determine subsequent actions.

@startuml

Skin rose

skinparam ParticipantPadding 50

skinparam BoxPadding 10

skinparam lifelineStrategy solid

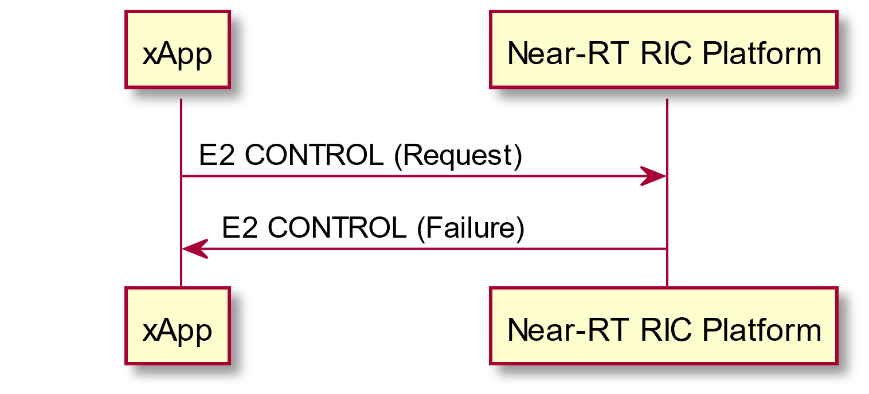
participant "xApp" as app

participant "Near-RT RIC Platform" as near

app -> near: E2 CONTROL (Request)

near -> app: E2 CONTROL (Failure)

@enduml



**Figure 6.2.3.1.3-2: E2 Control procedure, unsuccessful operation (Failure)**

If the request is accepted by the Near-RT RIC platform but failed in the E2AP RIC Control procedure, the Near-RT RIC platform shall send an E2 CONTROL (Failure) message back to the xApp with appropriate *Cause* IE. The xApp may use the information contained in the *Cause* IE and the optional *E2 Control Outcome* IE to determine subsequent actions.

##### 6.2.3.1.4 Abnormal Conditions

Not applied.

## 6.3 Elements for E2-related API Communication

### 6.3.1 General

### 6.3.2 Message Functional Definition and Content

#### 6.3.2.1 E2 SUBSCRIPTION (Request)

This message is sent by an xApp to the Near-RT RIC platform for an E2 Subscription.

Direction: xApp -> Near-RT RIC platform

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **IE/Group Name** | **Presence** | **Range** | **IE type and reference** | **Semantics description** | **Criticality** | **Assigned Criticality** |
| Message Type | M |  | 6.3.3.1 |  | YES | Reject |
| Procedure Transaction ID | M |  | 10.1.1 | Value assigned by the initiating xApp. | YES | Reject |
| Global E2 Node ID | M |  | 10.1.6 |  | YES | Reject |
| RAN Function ID | M |  | 10.1.7 |  | YES | Reject |
| E2 Subscription Details |  |  |  |  | YES | Reject |
| >E2 Event Trigger Definition | M |  | RIC Event Trigger Definition, Clause 9.2.9 of [3] |  | - |  |
| >Sequence of Actions |  | 1.. <maxnoofE2Actions> |  |  | EACH | Ignore |
| >>Requested E2 Action ID | M |  | RIC Action ID, Clause 9.2.10 of [3] | Assigned by initiating xApp to uniquely identify an action within the sequence of actions. It should not be assumed that the E2 Action ID in the subsequent E2 SUBSCRIPTION (Success) message or RIC Action ID in the corresponding E2AP message has the same value. | - |  |
| >>E2 Action Type | M |  | RIC Action Type, Clause 9.2.11 of [3] |  | - |  |
| >>E2 Action Definition | O |  | RIC Action Definition, Clause 9.2.12 of [3] |  | - |  |
| >>E2 Subsequent Action | O |  | RIC Subscription Action, Clause 9.2.13 of [3] |  | - |  |

|  |  |
| --- | --- |
| **Range bound** | **Explanation** |
| maxnoofE2Actions | Maximum no. of E2 Actions requested in the E2 subscription. Value is 16. |

#### 6.3.2.2 E2 SUBSCRIPTION (Success)

This message is sent by the Near-RT RIC platform after the xApp’s subscription request is successfully accepted with respect to the target E2 Node.

Direction: Near-RT RIC platform -> xApp

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **IE/Group Name** | **Presence** | **Range** | **IE type and reference** | **Semantics description** | **Criticality** | **Assigned Criticality** |
| Message Type | M |  | 6.3.3.1 |  | YES | Reject |
| Procedure Transaction ID | M |  | 10.1.1 | Value replicated from corresponding request. | YES | Reject |
| Global E2 Node ID | M |  | 10.1.6 | Value replicated from corresponding request. | YES | Reject |
| E2 Request ID | M |  | 6.3.3.2 | Value assigned by the Near-RT RIC platform to indicate the associated E2 subscription in Near-RT RIC. | YES | Reject |
| RAN Function ID | M |  | 10.1.7 | Value replicated from corresponding request. | YES | Reject |
| Actions Admitted List |  | 1.. <maxnoofE2Actions> |  |  | YES | Reject |
| >Requested E2 Action ID | M |  | RIC Action ID, Clause 9.2.10 of [3] | Value replicated from corresponding request message | - | - |
| >E2 Action ID | M |  | RIC Action ID, Clause 9.2.10 of [3] | Value assigned by Near-RT RIC platform to indicate in the associated E2 subscription the action ID that corresponds to the *Requested E2 Action ID* IE in the E2 SUBSCRIPTION REQUEST message. | - | - |
| Actions Not Admitted List |  | 0.. <maxnoofE2Actions> |  |  | YES | Reject |
| >Requested E2 Action ID | M |  | RIC Action ID, Clause 9.2.10 of [3] | Value replicated from corresponding request message | - | - |
| >Cause | M |  | 6.3.3.4 |  | - | - |

|  |  |
| --- | --- |
| **Range bound** | **Explanation** |
| maxnoofE2Actions | Maximum no. of E2 Actions requested in the E2 subscription. Value is 16. |

#### 6.3.2.3 E2 SUBSCRIPTION (Reject)

This message is sent by the Near-RT RIC platform after the xApp’s subscription request is rejected by Near-RT RIC platform with respect to the target E2 Node.

Direction: Near-RT RIC platform -> xApp

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **IE/Group Name** | **Presence** | **Range** | **IE type and reference** | **Semantics description** | **Criticality** | **Assigned Criticality** |
| Message Type | M |  | 6.3.3.1 |  | YES | Reject |
| Procedure Transaction ID | M |  | 10.1.1 | Value replicated from corresponding request. | YES | Reject |
| Global E2 Node ID | M |  | 10.1.6 | Value replicated from corresponding request. | YES | Reject |
| RAN Function ID | M |  | 10.1.7 | Value replicated from corresponding request. | YES | Reject |
| Cause | M |  | 6.3.3.4 |  | YES | Reject |
| Criticality Diagnostics | O |  | 6.3.3.5 |  | YES | Ignore |

#### 6.3.2.4 E2 SUBSCRIPTION (Failure)

This message is sent by the Near-RT RIC platform after the xApp’s subscription request is accepted by Near-RT RIC platform but failed by E2 Node, with respect to the target E2 Node.

Direction: Near-RT RIC platform -> xApp

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **IE/Group Name** | **Presence** | **Range** | **IE type and reference** | **Semantics description** | **Criticality** | **Assigned Criticality** |
| Message Type | M |  | 6.3.3.1 |  | YES | Reject |
| Procedure Transaction ID | M |  | 10.1.1 | Value replicated from corresponding request. | YES | Reject |
| Global E2 Node ID | M |  | 10.1.6 | Value replicated from corresponding request. | YES | Reject |
| RAN Function ID | M |  | 10.1.7 | Value replicated from corresponding request. | YES | Reject |
| Cause | M |  | 6.3.3.4 |  | YES | Reject |
| Criticality Diagnostics | O |  | 6.3.3.5 |  | YES | Ignore |

#### 6.3.2.5 E2 SUBSCRIPTION DELETE (Request)

This message is sent by the xApp to the Near-RT RIC platform to request deletion of one target E2 subscription.

Direction: xApp -> Near-RT RIC platform

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **IE/Group Name** | **Presence** | **Range** | **IE type and reference** | **Semantics description** | **Criticality** | **Assigned Criticality** |
| Message Type | M |  | 6.3.3.1 |  | YES | Reject |
| Procedure Transaction ID | M |  | 10.1.1 |  | YES | Reject |
| Global E2 Node ID | M |  | 10.1.6 |  | YES | Reject |
| E2 Request ID | M |  | 6.3.3.2 |  | YES | Reject |
| RAN Function ID | M |  | 10.1.7 |  | YES | Reject |

#### 6.3.2.6 E2 SUBSCRIPTION DELETE (Success)

This message is sent by the Near-RT RIC platform to the xApp to indicate success for the delete request of the target E2 subscription.

Direction: Near-RT RIC platform -> xApp

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **IE/Group Name** | **Presence** | **Range** | **IE type and reference** | **Semantics description** | **Criticality** | **Assigned Criticality** |
| Message Type | M |  | 6.3.3.1 |  | YES | Reject |
| Procedure Transaction ID | M |  | 10.1.1 | Value replicated from corresponding request. | YES | Reject |
| Global E2 Node ID | M |  | 10.1.6 | Value replicated from corresponding request. | YES | Reject |
| E2 Request ID | M |  | 6.3.3.2 | Value replicated from corresponding request. | YES | Reject |
| RAN Function ID | M |  | 10.1.7 | Value replicated from corresponding request. | YES | Reject |

#### 6.3.2.7 E2 SUBSCRIPTION DELETE (Reject)

This message is sent by the Near-RT RIC platform to the xApp when the Near-RT RIC platform rejects the delete request for the target E2 subscription.

Direction: Near-RT RIC platform -> xApp

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **IE/Group Name** | **Presence** | **Range** | **IE type and reference** | **Semantics description** | **Criticality** | **Assigned Criticality** |
| Message Type | M |  | 6.3.3.1 |  | YES | Reject |
| Procedure Transaction ID | M |  | 10.1.1 | Value replicated from corresponding request. | YES | Reject |
| Global E2 Node ID | M |  | 10.1.6 | Value replicated from corresponding request. | YES | Reject |
| E2 Request ID | M |  | 6.3.3.2 | Value replicated from corresponding request. | YES | Reject |
| RAN Function ID | M |  | 10.1.7 | Value replicated from corresponding request. | YES | Reject |
| Cause | M |  | 6.3.3.4 |  | YES | Ignore |
| Criticality Diagnostics | O |  | 6.3.3.5 |  | YES | Ignore |

#### 6.3.2.8 E2 SUBSCRIPTION DELETE (Failure)

This message is sent by the Near-RT RIC platform to the xApp when the delete request for a target E2 subscription fails due to failure in the associated E2AP procedure(s).

Direction: Near-RT RIC platform -> xApp

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **IE/Group Name** | **Presence** | **Range** | **IE type and reference** | **Semantics description** | **Criticality** | **Assigned Criticality** |
| Message Type | M |  | 6.3.3.1 |  | YES | Reject |
| Procedure Transaction ID | M |  | 10.1.1 | Value replicated from corresponding request. | YES | Reject |
| Global E2 Node ID | M |  | 10.1.6 | Value replicated from corresponding request. | YES | Reject |
| E2 Request ID | M |  | 6.3.3.2 | Value replicated from corresponding request. | YES | Reject |
| RAN Function ID | M |  | 10.1.7 | Value replicated from corresponding request. | YES | Reject |
| Cause | M |  | 6.3.3.4 |  | YES | Ignore |
| Criticality Diagnostics | O |  | 6.3.3.5 |  | YES | Ignore |

#### 6.3.2.9 E2 SUBSCRIPTION DELETE QUERY (Required)

#### 6.3.2.10 E2 SUBSCRIPTION DELETE QUERY (Accept)

#### 6.3.2.11 E2 SUBSCRIPTION DELETE QUERY (Decline)

#### 6.3.2.12 E2 SUBSCRIPTION DELETE NOTIFICATION

#### 6.3.2.13 E2 INDICATION (Push)

This message is sent by the Near-RT RIC platform to the xApp for delivery of subscribed information associated with an E2 subscription.

Direction: Near-RT RIC platform -> xApp

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **IE/Group Name** | **Presence** | **Range** | **IE type and reference** | **Semantics description** | **Criticality** | **Assigned Criticality** |
| Message Type | M |  | 6.3.3.1 |  | YES | Reject |
| Global E2 Node ID | M |  | 10.1.6 |  | YES | Reject |
| E2 Request ID | M |  | 6.3.3.2 |  | YES | Reject |
| RAN Function ID | M |  | 10.1.7 |  | YES | Reject |
| E2 Action ID | M |  | RIC Action ID, Clause 9.2.10 of [3]. |  |  |  |
| E2 Indication SN | O |  | RIC Indication SN, Clause 9.2.14 of [3]. |  | YES | Reject |
| E2 Indication Type | M |  | RIC Indication Type, Clause 9.2.15 of [3]. |  | YES | Reject |
| E2 Indication Header | M |  | RIC Indication Header, Clause 9.2.17 of [3]. |  | YES | Reject |
| E2 Indication Message | M |  | RIC Indication Message, Clause 9.2.16 of [3]. |  | YES | Reject |
| E2 Call Process ID | O |  | RIC Call Process ID, Clause 9.2.18 of [3]. |  | YES | Reject |

#### 6.3.2.14 E2 INDICATION (Failure)

This message is sent by the xApp to the Near-RT RIC platform if an error is detected in E2 INDICATION (Push) message.

Direction: xApp -> Near-RT RIC platform

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **IE/Group Name** | **Presence** | **Range** | **IE type and reference** | **Semantics description** | **Criticality** | **Assigned Criticality** |
| Message Type | M |  | 6.3.3.1 |  | YES | Reject |
| Global E2 Node ID | M |  | 10.1.6 |  | YES | Reject |
| E2 Request ID | M |  | 6.3.3.2 |  | YES | Reject |
| RAN Function ID | M |  | 10.1.7 |  | YES | Reject |
| E2 Action ID | M |  | RIC Action ID, Clause 9.2.10 of [3]. |  |  |  |
| E2 Indication SN | O |  | RIC Indication SN, Clause 9.2.14 of [3]. |  | YES | Reject |
| Cause | M |  | 6.3.3.4 |  | YES | Reject |
| Criticality Diagnostics | O |  | 6.3.3.5 |  | YES | Ignore |

#### 6.3.2.15 E2 CONTROL (Request)

This message is sent by the xApp to the Near-RT RIC platform for an E2 control request.

Direction: xApp -> Near-RT RIC platform

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **IE/Group Name** | **Presence** | **Range** | **IE type and reference** | **Semantics description** | **Criticality** | **Assigned Criticality** |
| Message Type | M |  | 6.3.3.1 |  | YES | Reject |
| Procedure Transaction ID | M |  | 10.1.1 | Value assigned by the initiating xApp. | YES | Reject |
| Global E2 Node ID | M |  | 10.1.6 |  | YES | Reject |
| RAN Function ID | M |  | 10.1.7 |  | YES | Reject |
| E2 Call Process ID | O |  | RIC Call Process ID, Clause 9.2.18 of [3]. |  | YES | Reject |
| E2 Control Header | M |  | RIC Control Header, Clause 9.2.20 of [3]. |  | YES | Reject |
| E2 Control Message | M |  | RIC Control Message, Clause 9.2.19 of [3]. |  | YES | Reject |
| E2 Control Ack Request | O |  | RIC Control Ack Request, Clause 9.2.21 of [3]. |  | YES | Reject |
| E2 Control Timer | O |  | 6.3.3.6 |  | YES | Ignore |

#### 6.3.2.16 E2 CONTROL (Success)

This message is sent by the Near-RT RIC platform to the xApp for a successfully performed E2 control request.

Direction: Near-RT RIC platform -> xApp

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **IE/Group Name** | **Presence** | **Range** | **IE type and reference** | **Semantics description** | **Criticality** | **Assigned Criticality** |
| Message Type | M |  | 6.3.3.1 |  | YES | Reject |
| Procedure Transaction ID | M |  | 10.1.1 | Value replicated from corresponding request. | YES | Reject |
| Global E2 Node ID | M |  | 10.1.6 |  | YES | Reject |
| RAN Function ID | M |  | 10.1.7 |  | YES | Reject |
| E2 Call Process ID | O |  | RIC Call Process ID, Clause 9.2.18 of [3]. |  | YES | Reject |
| E2 Control Outcome | O |  | RIC Control Outcome, Clause 9.2.25 of [3]. |  | YES | Reject |

#### 6.3.2.17 E2 CONTROL (Reject)

This message is sent by the Near-RT RIC platform to the xApp if the corresponding E2 control request is rejected by the Near-RT RIC platform.

Direction: Near-RT RIC platform -> xApp

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **IE/Group Name** | **Presence** | **Range** | **IE type and reference** | **Semantics description** | **Criticality** | **Assigned Criticality** |
| Message Type | M |  | 6.3.3.1 |  | YES | Reject |
| Procedure Transaction ID | M |  | 10.1.1 | Value replicated from corresponding request. | YES | Reject |
| Global E2 Node ID | M |  | 10.1.6 |  | YES | Reject |
| RAN Function ID | M |  | 10.1.7 |  | YES | Reject |
| Cause | M |  | 6.3.3.4 |  | YES | Ignore |
| Criticality Diagnostics | O |  | 6.3.3.5 |  | YES | Ignore |

#### 6.3.2.18 E2 CONTROL (Failure)

This message is sent by the Near-RT RIC platform to the xApp if the corresponding E2 control request is accepted by the Near-RT RIC platform but fails in the E2 Node.

Direction: Near-RT RIC platform -> xApp

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **IE/Group Name** | **Presence** | **Range** | **IE type and reference** | **Semantics description** | **Criticality** | **Assigned Criticality** |
| Message Type | M |  | 6.3.3.1 |  | YES | Reject |
| Procedure Transaction ID | M |  | 10.1.1 | Value replicated from corresponding request. | YES | Reject |
| Global E2 Node ID | M |  | 10.1.6 |  | YES | Reject |
| RAN Function ID | M |  | 10.1.7 |  | YES | Reject |
| E2 Call Process ID | O |  | RIC Call Process ID, Clause 9.2.18 of [3]. |  | YES | Reject |
| E2 Control Outcome | O |  | RIC Control Outcome, Clause 9.2.25 of [3]. |  | YES | Reject |
| Cause | M |  | 6.3.3.4 |  | YES | Ignore |
| Criticality Diagnostics | O |  | 6.3.3.5 |  | YES | Ignore |

#### 6.3.2.19 E2 GUIDANCE (Request)

#### 6.3.2.20 E2 GUIDANCE (Response)

#### 6.3.2.21 E2 GUIDANCE (Modification)

### 6.3.3 Information Element Definitions

#### 6.3.3.1 Message Type

This IE uniquely identifies the message being sent. It is mandatory for all messages.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE Type and Reference | Semantics Description |
| Message Type |  |  |  |  |
| >Procedure Type | M |  | CHOICE (E2 Subscription, E2 Subscription Delete, E2 Subscription Delete Query, E2 Subscription Delete Notification, E2 Indication, E2 Control, …) |  |
| >Type of Message | M |  | CHOICE (Request, Success, Reject, Failure, …) | Refer to Table 6.1.2-3 for applicable choices for each procedure type |

#### 6.3.3.2 E2 Request ID

This IE is assigned by Near-RT RIC platform to uniquely identify an E2 subscription among all the E2 subscriptions associated with the same E2 Node.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description |
| E2 Request ID | M |  | INTEGER (0 … 4294967295) |  |

#### 6.3.3.3 Void

#### 6.3.3.4 Cause

This IE is used to indicate the reason for a particular event for the E2-related APIs.

Note: Not all causes from E2AP are applicable.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description |
| CHOICE Cause Group | M |  |  |  |
| >*E2-related API Cause* |  |  |  |  |
| >E2-related API Cause | M |  | 6.3.3.4a. |  |
| >*E2AP Cause* |  |  |  |  |
| >>E2AP Cause | M |  | Cause, Clause 9.2.1 of [3]. |  |

6.3.3.4a E2-related API Cause

The purpose of the *E2-related API Cause* IE is to indicate the reason for a particular event for the E2-related APIs.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description |
| CHOICE *E2-related API* *Cause Group* | M |  |  |  |
| >*Application Layer* |  |  |  |  |
| >>Application Layer Cause | M |  | ENUMERATED  (Unspecified,  Global E2Node ID invalid, E2 Node Reset or Disconnected,  RAN Function ID invalid,  Action not supported,  Excessive actions,  Duplicate action,  E2 Request ID unknown,  Control conflict,  E2 Indication unknown,  E2AP Procedure timeout,  Essential for operation, …) |  |
| >*Transport Layer* |  |  |  |  |
| >>Transport Layer Cause | M |  | ENUMERATED  (Unspecified,  Transport Resource Unavailable, ...) |  |
| >*Protocol* |  |  |  |  |
| >>Protocol Cause | M |  | ENUMERATED  (Unspecified, Transfer Syntax Error,  Abstract Syntax Error (Reject),  Abstract Syntax Error (Ignore and Notify),  Message not Compatible with Receiver State,  Semantic Error,  Abstract Syntax Error (Falsely Constructed Message), ...) |  |
| >*Misc* |  |  |  |  |
| >>Miscellaneous Cause | M |  | ENUMERATED  (Unspecified, Control Processing Overload,  Hardware Failure,  O&M Intervention, ...) |  |

The meaning of the different cause values is described in the following table. In general, "not supported" cause values indicate that the related capability is missing. On the other hand, "not available" cause values indicate that the related capability is present, but insufficient resources were available to perform the requested action.

|  |  |
| --- | --- |
| **Application Layer cause** | Meaning |
| Unspecified | Sent when none of the cause values below applies but still the cause is Application Layer related. |
| Global E2Node ID invalid | Global E2Node ID invalid. |
| E2 Node Reset or Disconnected | E2Node Status is Reset or Disconnected. |
| RAN Function ID invalid | Requested RAN Function ID invalid. |
| Action not supported | Requested Action not supported by RAN function. |
| Excessive actions | Excessive number of actions requested for RAN Function. |
| Duplicate action | Same action requested more than once in same subscription request. |
| E2 Request ID unknown | E2 Request ID sent to Near-RT RIC is unknown. |
| Control Conflict | Requested Control command causes conflict. |
| E2 Indication unknown | The E2 Indication message is unknown, i.e. The received indication message is not subscribed to by the xApp. |
| E2AP Procedure timeout | An E2AP procedure triggered by the E2-related API procedure is time-out. |
| Essential for operation | xApp is unwilling to delete an E2 subscription essential for its operation. |

|  |  |
| --- | --- |
| **Transport Layer cause** | Meaning |
| Unspecified | Sent when none of the cause values below applies but still the cause is Transport Layer related. |
| Transport Resource Unavailable | The required transport resources are not available. |

|  |  |
| --- | --- |
| **Protocol cause** | Meaning |
| Unspecified | Sent when none of the below cause values applies but still the cause is Protocol related. |
| Transfer Syntax Error | The received message included a transfer syntax error. |
| Abstract Syntax Error (Reject) | The received message included an abstract syntax error and the concerning criticality indicated "reject". |
| Abstract Syntax Error (Ignore And Notify) | The received message included an abstract syntax error and the concerning criticality indicated "ignore and notify". |
| Message Not Compatible With Receiver State | The received message was not compatible with the receiver state. |
| Semantic Error | The received message included a semantic error. |
| Abstract Syntax Error (Falsely Constructed Message) | The received message contained IEs or IE groups in wrong order or with too many occurrences. |

| **Miscellaneous cause** | **Meaning** |
| --- | --- |
| Unspecified Failure | Sent when none of the below cause values applies and the cause is not related to any of the categories Application Layer, Transport Layer, Protocol. |
| Control Processing Overload | Control processing overload. |
| Hardware Failure | Action related to hardware failure. |
| O&M Intervention | The action is due to O&M intervention. |

#### 6.3.3.5 Criticality Diagnostics

The *Criticality Diagnostics* IE is sent by the xApp or the Near-RT RIC Platform when parts of a received message have not been comprehended or were missing. When applicable, it contains information about which IEs were not comprehended or were missing.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description |
| **Information Element Criticality Diagnostics** |  | *0 .. <maxnoofErrors>* |  |  |
| >IE Criticality | M |  | ENUMERATED (reject, ignore, notify) | The IE Criticality is used for reporting the criticality of the triggering IE. The value 'ignore' shall not be used. |
| >IE ID | M |  | ENUMERATED (Message Type, Procedure Transaction ID, Global E2 Node ID, RAN Function ID…) | The IE ID of the not understood or missing IE. |
| >Type of Error | M |  | ENUMERATED (not understood, missing, ...) |  |

|  |  |
| --- | --- |
| **Range bound** | Explanation |
| *maxnoofErrors* | Maximum no. of IE errors allowed to be reported with a single message. Value is 256. |

#### 6.3.3.6 E2 Control Timer

This IE specifies the suggested time to wait before timeout for the RIC Control procedure in E2AP[3].

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description |
| E2 Control Timer | M |  | INTEGER (10..10000, ...) | Unit in milliseconds. |

### 6.3.4 Message and Information Element Abstract Syntax

#### 6.3.4.1 Solution 1: SCTP with Protocol Buffers

##### 6.3.4.1.1 Usage of SCTP with Protocol Buffers

This solution uses Protocol Buffers as message encoding, and SCTP as the transport protocol.

The PDU on the SCTP connection uses the structure as defined in Section 10.1.8 where its *API Payload* IE serves as a container for individual API messages in Section 6.3.4.1.2 which can be E2SubscriptionApiPayload or E2IndicationApiPayload or E2ControlApiPayload.

Table 6.3.4.1.1-1 lists the E2 related API abstract syntax included in this document.

Table 6.3.4.1.1-1: List of E2 related APIs abstract syntax

|  |  |  |  |
| --- | --- | --- | --- |
| API Name / Contents | Clause | Version | File Name |
| E2\_Subscription\_API | 6.3.4.1.2.1 | 1.0.0 | e2\_subscription\_api\_ver\_1\_0\_0.proto |
| E2\_Indication\_API | 6.3.4.1.2.2 | 1.0.0 | e2\_indication\_api\_ver\_1\_0\_0.proto |
| E2\_Control\_API | 6.3.4.1.2.3 | 1.0.0 | e2\_control\_api\_ver\_1\_0\_0.proto |
| E2 related API common information elements | 6.3.4.1.3 | 1.0.0 | e2\_related\_apis\_common\_ies\_ver\_1\_0\_0.proto |
| Near-RT RIC API common information elements | 10.2.1 | 1.0.0 | near\_rt\_ric\_apis\_common\_ies\_ver\_1\_0\_0.proto |

##### 6.3.4.1.2 API Definitions

###### 6.3.4.1.2.1 E2\_Subscription\_API

syntax = "proto3";

package ricapi.e2\_subscription\_api.v1;

import "near\_rt\_ric\_apis\_common\_ies\_ver\_1\_0\_0.proto";

import "e2\_related\_apis\_common\_ies\_ver\_1\_0\_0.proto";

// top-level PDU

message E2SubscriptionApiPayload {

oneof procedure\_type {

E2SubscriptionProc e2\_subscription\_proc = 1;

E2SubscriptionDeleteProc e2\_subscription\_delete\_proc = 2;

E2SubscriptionDeleteQueryProc e2\_subscription\_delete\_query\_proc = 3;

E2SubscriptionDeleteNotificationProc e2\_subscription\_delete\_notification\_proc = 4;

}

}

// ---------------- procedures of E2\_Subscription\_API ----------------

message E2SubscriptionProc {

oneof type\_of\_message {

E2SubscriptionRequest e2\_subscription\_request = 1 ;

E2SubscriptionSuccess e2\_subscription\_success = 2 ;

E2SubscriptionReject e2\_subscription\_reject = 3 ;

E2SubscriptionFailure e2\_subscription\_failure = 4 ;

}

}

message E2SubscriptionDeleteProc {

oneof type\_of\_message {

E2SubscriptionDeleteRequest e2\_subscription\_delete\_request = 1 ;

E2SubscriptionDeleteSuccess e2\_subscription\_delete\_success = 2 ;

E2SubscriptionDeleteReject e2\_subscription\_delete\_reject = 3 ;

E2SubscriptionDeleteFailure e2\_subscription\_delete\_failure = 4 ;

}

}

message E2SubscriptionDeleteQueryProc {

oneof type\_of\_message {

E2SubscriptionDeleteQueryRequired e2\_subscription\_delete\_query\_required = 1 ;

E2SubscriptionDeleteQueryAccept e2\_subscription\_delete\_query\_accept = 2 ;

E2SubscriptionDeleteQueryDecline e2\_subscription\_delete\_query\_reject = 3 ;

}

}

message E2SubscriptionDeleteNotificationProc {

oneof type\_of\_message {

E2SubscriptionDeleteNotificationDeleted e2\_subscription\_delete\_notification\_deleted = 1 ;

}

}

// ---------------- messages of E2\_Subscription\_API ----------------

// -------- messages of E2 Subscription procedure --------

message E2SubscriptionRequest {

optional near\_rt\_ric\_apis\_common\_ies.v1.ProcedureTransactionId procedure\_transaction\_id = 1;

optional near\_rt\_ric\_apis\_common\_ies.v1.GlobalE2NodeId global\_e2\_node\_id = 2;

optional near\_rt\_ric\_apis\_common\_ies.v1.RanFunctionId ran\_function\_id = 3;

optional E2SubscriptionDetails e2\_subscription\_details = 4;

message E2SubscriptionDetails {

optional e2\_related\_apis\_common\_ies.v1.E2EventTriggerDefinition e2\_event\_trigger\_definition = 1;

repeated E2ActionToBeSetupItem sequence\_of\_actions = 2 ;

message E2ActionToBeSetupItem {

optional e2\_related\_apis\_common\_ies.v1.E2ActionId requested\_e2\_action\_id = 1 ;

optional e2\_related\_apis\_common\_ies.v1.E2ActionType e2\_action\_type = 2 ;

optional e2\_related\_apis\_common\_ies.v1.E2ActionDefinition e2\_action\_definition = 3 ;

optional e2\_related\_apis\_common\_ies.v1.E2SubsequentAction e2\_subsequent\_action = 4 ;

}

}

}

message E2SubscriptionSuccess {

optional near\_rt\_ric\_apis\_common\_ies.v1.ProcedureTransactionId procedure\_transaction\_id = 1;

optional near\_rt\_ric\_apis\_common\_ies.v1.GlobalE2NodeId global\_e2\_node\_id = 2;

optional e2\_related\_apis\_common\_ies.v1.E2RequestId e2\_request\_id = 3;

optional near\_rt\_ric\_apis\_common\_ies.v1.RanFunctionId ran\_function\_id = 4;

repeated E2ActionAdmittedItem e2\_actions\_admitted\_list = 5;

repeated E2ActionNotAdmittedItem e2\_actions\_not\_admitted\_list = 6;

message E2ActionAdmittedItem {

optional e2\_related\_apis\_common\_ies.v1.E2ActionId requested\_e2\_action\_id = 1 ;

optional e2\_related\_apis\_common\_ies.v1.E2ActionId e2\_action\_id = 2 ;

}

message E2ActionNotAdmittedItem {

optional e2\_related\_apis\_common\_ies.v1.E2ActionId requested\_e2\_action\_id = 1 ;

optional e2\_related\_apis\_common\_ies.v1.Cause cause = 2 ;

}

}

message E2SubscriptionFailure {

optional near\_rt\_ric\_apis\_common\_ies.v1.ProcedureTransactionId procedure\_transaction\_id = 1;

optional near\_rt\_ric\_apis\_common\_ies.v1.GlobalE2NodeId global\_e2\_node\_id = 2;

optional near\_rt\_ric\_apis\_common\_ies.v1.RanFunctionId ran\_function\_id = 3;

optional e2\_related\_apis\_common\_ies.v1.Cause cause = 4;

optional e2\_related\_apis\_common\_ies.v1.CriticalityDiagnostics criticality\_diagnostics = 5;

}

message E2SubscriptionReject {

optional near\_rt\_ric\_apis\_common\_ies.v1.ProcedureTransactionId procedure\_transaction\_id = 1;

optional near\_rt\_ric\_apis\_common\_ies.v1.GlobalE2NodeId global\_e2\_node\_id = 2;

optional near\_rt\_ric\_apis\_common\_ies.v1.RanFunctionId ran\_function\_id = 3;

optional e2\_related\_apis\_common\_ies.v1.Cause cause = 4;

optional e2\_related\_apis\_common\_ies.v1.CriticalityDiagnostics criticality\_diagnostics = 5;

}

// -------- messages of E2 Subscription Delete procedure --------

message E2SubscriptionDeleteRequest {

optional near\_rt\_ric\_apis\_common\_ies.v1.ProcedureTransactionId procedure\_transaction\_id = 1;

optional near\_rt\_ric\_apis\_common\_ies.v1.GlobalE2NodeId global\_e2\_node\_id = 2;

optional e2\_related\_apis\_common\_ies.v1.E2RequestId e2\_request\_id = 3;

optional near\_rt\_ric\_apis\_common\_ies.v1.RanFunctionId ran\_function\_id = 4;

}

message E2SubscriptionDeleteSuccess {

optional near\_rt\_ric\_apis\_common\_ies.v1.ProcedureTransactionId procedure\_transaction\_id = 1;

optional near\_rt\_ric\_apis\_common\_ies.v1.GlobalE2NodeId global\_e2\_node\_id = 2;

optional e2\_related\_apis\_common\_ies.v1.E2RequestId e2\_request\_id = 3;

optional near\_rt\_ric\_apis\_common\_ies.v1.RanFunctionId ran\_function\_id = 4;

}

message E2SubscriptionDeleteFailure {

optional near\_rt\_ric\_apis\_common\_ies.v1.ProcedureTransactionId procedure\_transaction\_id = 1;

optional near\_rt\_ric\_apis\_common\_ies.v1.GlobalE2NodeId global\_e2\_node\_id = 2;

optional e2\_related\_apis\_common\_ies.v1.E2RequestId e2\_request\_id = 3;

optional near\_rt\_ric\_apis\_common\_ies.v1.RanFunctionId ran\_function\_id = 4;

optional e2\_related\_apis\_common\_ies.v1.Cause cause = 5;

optional e2\_related\_apis\_common\_ies.v1.CriticalityDiagnostics criticality\_diagnostics = 6;

}

message E2SubscriptionDeleteReject {

optional near\_rt\_ric\_apis\_common\_ies.v1.ProcedureTransactionId procedure\_transaction\_id = 1;

optional near\_rt\_ric\_apis\_common\_ies.v1.GlobalE2NodeId global\_e2\_node\_id = 2;

optional e2\_related\_apis\_common\_ies.v1.E2RequestId e2\_request\_id = 3;

optional near\_rt\_ric\_apis\_common\_ies.v1.RanFunctionId ran\_function\_id = 4;

optional e2\_related\_apis\_common\_ies.v1.Cause cause = 5;

optional e2\_related\_apis\_common\_ies.v1.CriticalityDiagnostics criticality\_diagnostics = 6;

}

// -------- messages of E2 Subscription Delete Query procedure --------

message E2SubscriptionDeleteQueryRequired {

optional near\_rt\_ric\_apis\_common\_ies.v1.ProcedureTransactionId procedure\_transaction\_id = 1;

optional near\_rt\_ric\_apis\_common\_ies.v1.GlobalE2NodeId global\_e2\_node\_id = 2;

optional e2\_related\_apis\_common\_ies.v1.E2RequestId e2\_request\_id = 3;

optional near\_rt\_ric\_apis\_common\_ies.v1.RanFunctionId ran\_function\_id = 4;

optional e2\_related\_apis\_common\_ies.v1.Cause cause = 5;

}

message E2SubscriptionDeleteQueryAccept {

optional near\_rt\_ric\_apis\_common\_ies.v1.ProcedureTransactionId procedure\_transaction\_id = 1;

optional near\_rt\_ric\_apis\_common\_ies.v1.GlobalE2NodeId global\_e2\_node\_id = 2;

optional e2\_related\_apis\_common\_ies.v1.E2RequestId e2\_request\_id = 3;

optional near\_rt\_ric\_apis\_common\_ies.v1.RanFunctionId ran\_function\_id = 4;

}

message E2SubscriptionDeleteQueryDecline {

optional near\_rt\_ric\_apis\_common\_ies.v1.ProcedureTransactionId procedure\_transaction\_id = 1;

optional near\_rt\_ric\_apis\_common\_ies.v1.GlobalE2NodeId global\_e2\_node\_id = 2;

optional e2\_related\_apis\_common\_ies.v1.E2RequestId e2\_request\_id = 3;

optional near\_rt\_ric\_apis\_common\_ies.v1.RanFunctionId ran\_function\_id = 4;

optional e2\_related\_apis\_common\_ies.v1.Cause cause = 5;

}

// -------- messages of E2 Subscription Delete Notification procedure --------

message E2SubscriptionDeleteNotificationDeleted {

optional near\_rt\_ric\_apis\_common\_ies.v1.ProcedureTransactionId procedure\_transaction\_id = 1;

optional near\_rt\_ric\_apis\_common\_ies.v1.GlobalE2NodeId global\_e2\_node\_id = 2;

optional e2\_related\_apis\_common\_ies.v1.E2RequestId e2\_request\_id = 3;

optional near\_rt\_ric\_apis\_common\_ies.v1.RanFunctionId ran\_function\_id = 4;

optional e2\_related\_apis\_common\_ies.v1.Cause cause = 5;

}

###### 6.3.4.1.2.2 E2\_Indication\_API

syntax = "proto3";

package ricapi.e2\_indication\_api.v1;

import "near\_rt\_ric\_apis\_common\_ies\_ver\_1\_0\_0.proto";

import "e2\_related\_apis\_common\_ies\_ver\_1\_0\_0.proto";

// top-level PDU

message E2IndicationApiPayload {

oneof procedure\_type {

E2IndicationProc e2\_indication\_proc = 1;

}

}

// ---------------- procedures of E2\_Subscription\_API ----------------

message E2IndicationProc {

oneof type\_of\_message {

E2IndicationPush e2\_indication\_push = 1 ;

E2IndicationFailure e2\_indication\_failure = 2 ;

}

}

// ---------------- messages of E2\_Indication\_API ----------------

// -------- messages of E2 Indication procedure --------

message E2IndicationPush {

optional near\_rt\_ric\_apis\_common\_ies.v1.GlobalE2NodeId global\_e2\_node\_id = 1;

optional e2\_related\_apis\_common\_ies.v1.E2RequestId e2\_request\_id = 2;

optional near\_rt\_ric\_apis\_common\_ies.v1.RanFunctionId ran\_function\_id = 3;

optional e2\_related\_apis\_common\_ies.v1.E2ActionId e2\_action\_id = 4;

optional e2\_related\_apis\_common\_ies.v1.E2IndicationSN e2\_indication\_sn = 5;

optional e2\_related\_apis\_common\_ies.v1.E2IndicationType e2\_indication\_type = 6;

optional e2\_related\_apis\_common\_ies.v1.E2IndicationHeader e2\_indication\_header = 7;

optional e2\_related\_apis\_common\_ies.v1.E2IndicationMessage e2\_indication\_message = 8;

optional e2\_related\_apis\_common\_ies.v1.E2CallProcessId e2\_call\_process\_id = 9;

}

message E2IndicationFailure {

optional near\_rt\_ric\_apis\_common\_ies.v1.GlobalE2NodeId global\_e2\_node\_id = 1;

optional e2\_related\_apis\_common\_ies.v1.E2RequestId e2\_request\_id = 2;

optional near\_rt\_ric\_apis\_common\_ies.v1.RanFunctionId ran\_function\_id = 3;

optional e2\_related\_apis\_common\_ies.v1.E2ActionId e2\_action\_id = 4;

optional e2\_related\_apis\_common\_ies.v1.E2IndicationSN e2\_indication\_sn = 5;

optional e2\_related\_apis\_common\_ies.v1.Cause cause = 6;

optional e2\_related\_apis\_common\_ies.v1.CriticalityDiagnostics criticality\_diagnostics = 7;

}

###### 6.3.4.1.2.3 E2\_Control\_API

syntax = "proto3";

package ricapi.e2\_control\_api.v1;

import "near\_rt\_ric\_apis\_common\_ies\_ver\_1\_0\_0.proto";

import "e2\_related\_apis\_common\_ies\_ver\_1\_0\_0.proto";

// top-level PDU

message E2ControlApiPayload {

oneof procedure\_type {

E2ControlProc e2\_control\_proc = 1;

}

}

// ---------------- procedures of E2\_Control\_API ----------------

message E2ControlProc {

oneof type\_of\_message {

E2ControlRequest e2\_control\_request = 1;

E2ControlSuccess e2\_control\_success = 2;

E2ControlReject e2\_control\_reject = 3;

E2ControlFailure e2\_control\_failure = 4;

}

}

// ---------------- messages of E2\_Control\_API ----------------

// -------- messages of E2 Control procedure --------

message E2ControlRequest {

optional near\_rt\_ric\_apis\_common\_ies.v1.ProcedureTransactionId procedure\_transaction\_id = 1;

optional near\_rt\_ric\_apis\_common\_ies.v1.GlobalE2NodeId global\_e2\_node\_id = 2;

optional near\_rt\_ric\_apis\_common\_ies.v1.RanFunctionId ran\_function\_id = 3;

optional e2\_related\_apis\_common\_ies.v1.E2CallProcessId e2\_call\_process\_id = 4;

optional e2\_related\_apis\_common\_ies.v1.E2ControlHeader e2\_control\_header = 5;

optional e2\_related\_apis\_common\_ies.v1.E2ControlMessage e2\_control\_message = 6;

optional e2\_related\_apis\_common\_ies.v1.E2ControlAckRequest e2\_control\_ack\_request = 7;

optional uint32 e2\_control\_timer = 8;

}

message E2ControlSuccess {

optional near\_rt\_ric\_apis\_common\_ies.v1.ProcedureTransactionId procedure\_transaction\_id = 1;

optional near\_rt\_ric\_apis\_common\_ies.v1.GlobalE2NodeId global\_e2\_node\_id = 2;

optional near\_rt\_ric\_apis\_common\_ies.v1.RanFunctionId ran\_function\_id = 3;

optional e2\_related\_apis\_common\_ies.v1.E2CallProcessId e2\_call\_process\_id = 4;

optional e2\_related\_apis\_common\_ies.v1.E2ControlOutcome e2\_control\_outcome = 5;

}

message E2ControlReject {

optional near\_rt\_ric\_apis\_common\_ies.v1.ProcedureTransactionId procedure\_transaction\_id = 1;

optional near\_rt\_ric\_apis\_common\_ies.v1.GlobalE2NodeId global\_e2\_node\_id = 2;

optional near\_rt\_ric\_apis\_common\_ies.v1.RanFunctionId ran\_function\_id = 3;

optional e2\_related\_apis\_common\_ies.v1.Cause cause = 4;

optional e2\_related\_apis\_common\_ies.v1.CriticalityDiagnostics criticality\_diagnostics = 5;

}

message E2ControlFailure {

optional near\_rt\_ric\_apis\_common\_ies.v1.ProcedureTransactionId procedure\_transaction\_id = 1;

optional near\_rt\_ric\_apis\_common\_ies.v1.GlobalE2NodeId global\_e2\_node\_id = 2;

optional near\_rt\_ric\_apis\_common\_ies.v1.RanFunctionId ran\_function\_id = 3;

optional e2\_related\_apis\_common\_ies.v1.E2CallProcessId e2\_call\_process\_id = 4;

optional e2\_related\_apis\_common\_ies.v1.E2ControlOutcome e2\_control\_outcome = 5;

optional e2\_related\_apis\_common\_ies.v1.Cause cause = 6;

optional e2\_related\_apis\_common\_ies.v1.CriticalityDiagnostics criticality\_diagnostics = 7;

}

##### 6.3.4.1.3 Information Element Definitions

syntax = "proto3";

package ricapi.e2\_related\_apis\_common\_ies.v1;

import "near\_rt\_ric\_apis\_common\_ies\_ver\_1\_0\_0.proto";

// ----------------------------- Information Elements -----------------------------

// --------------E--------------

// From E2AP Clause 9.2.11

enum E2ActionType {

E2ACTION\_TYPE\_REPORT = 0;

E2ACTION\_TYPE\_INSERT = 1;

E2ACTION\_TYPE\_POLICY = 2;

};

// From E2AP Clause 9.2.21

enum E2ControlAckRequest {

E2\_CONTROL\_ACK\_REQUEST\_NO\_ACK = 0;

E2\_CONTROL\_ACK\_REQUEST\_ACK = 1;

}

// From E2AP Clause 9.2.15

enum E2IndicationType {

E2\_INDICATION\_TYPE\_REPORT = 0;

E2\_INDICATION\_TYPE\_INSERT = 1;

}

// --------------C--------------

message Cause {

oneof cause\_group {

E2RelatedApiCause e2\_related\_api\_cause = 1;

E2apCause e2ap\_cause = 2;

}

message E2apCause {

oneof cause\_group {

RicRequest ric\_request = 1;

RicService ric\_service = 2;

E2Node e2\_node = 3;

Transport transport = 4;

Protocol protocol = 5;

Misc misc = 6;

}

enum E2Node {

E2NODE\_COMPONENT\_UNKNOWN = 0;

}

enum Misc {

CONTROL\_PROCESSING\_OVERLOAD = 0;

HARDWARE\_FAILURE = 1;

OM\_INTERVENTION = 2;

MISC\_UNSPECIFIED = 3;

}

enum Protocol {

TRANSFER\_SYNTAX\_ERROR = 0;

ABSTRACT\_SYNTAX\_ERROR\_REJECT = 1;

ABSTRACT\_SYNTAX\_ERROR\_IGNORE\_AND\_NOTIFY = 2;

MESSAGE\_NOT\_COMPATIBLE\_WITH\_RECEIVER\_STATE = 3;

SEMANTIC\_ERROR = 4;

ABSTRACT\_SYNTAX\_ERROR\_FALSELY\_CONSTRUCTED\_MESSAGE = 5;

PROTOCOL\_UNSPECIFIED = 6;

}

enum RicRequest {

RAN\_FUNCTION\_ID\_INVALID = 0;

ACTION\_NOT\_SUPPORTED = 1;

EXCESSIVE\_ACTIONS = 2;

DUPLICATE\_ACTION = 3;

DUPLICATE\_EVENT\_TRIGGER = 4;

FUNCTION\_RESOURCE\_LIMIT = 5;

REQUEST\_ID\_UNKNOWN = 6;

INCONSISTENT\_ACTION\_SUBSEQUENT\_ACTION\_SEQUENCE = 7;

CONTROL\_MESSAGE\_INVALID = 8;

RIC\_CALL\_PROCESS\_ID\_INVALID = 9;

CONTROL\_TIMER\_EXPIRED = 10;

CONTROL\_FAILED\_TO\_EXECUTE = 11;

SYSTEM\_NOT\_READY = 12;

RIC\_REQUEST\_UNSPECIFIED = 13;

}

enum RicService {

RAN\_FUNCTION\_NOT\_SUPPORTED = 0;

EXCESSIVE\_FUNCTIONS = 1;

RIC\_RESOURCE\_LIMIT = 2;

}

enum Transport {

TRANSPORT\_UNSPECIFIED = 0;

TRANSPORT\_RESOURCE\_UNAVAILABLE = 1;

}

}

message E2RelatedApiCause {

oneof cause\_group {

ApplicationLayer application\_layer = 1;

Transport transport = 2;

Protocol protocol = 3;

Misc misc = 4;

}

enum ApplicationLayer {

APPLICATION\_LAYER\_UNSPECIFIED = 0;

GLOBAL\_E2\_NODE\_ID\_INVALID = 1;

E2\_NODE\_RESET\_OR\_DISCONNECTED = 2;

RAN\_FUNCTION\_ID\_INVALID = 3;

ACTION\_NOT\_SUPPORTED = 4;

EXCESSIVE\_ACTIONS = 5;

DUPLICATE\_ACTION = 6;

E2\_REQUEST\_ID\_UNKNOWN = 7;

CONTROL\_CONFLICT = 8;

E2\_INDICATION\_UNKNOWN = 9;

E2AP\_PROCEDURE\_TIMEOUT = 10;

ESSENTIAL\_FOR\_OPERATION = 11;

}

enum Misc {

MISC\_UNSPECIFIED = 0;

CONTROL\_PROCESSING\_OVERLOAD = 1;

HARDWARE\_FAILURE = 2;

OM\_INTERVENTION = 3;

}

enum Protocol {

PROTOCOL\_UNSPECIFIED = 0;

TRANSFER\_SYNTAX\_ERROR = 1;

ABSTRACT\_SYNTAX\_ERROR\_REJECT = 2;

ABSTRACT\_SYNTAX\_ERROR\_IGNORE\_AND\_NOTIFY = 3;

MESSAGE\_NOT\_COMPATIBLE\_WITH\_RECEIVER\_STATE = 4;

SEMANTIC\_ERROR = 5;

ABSTRACT\_SYNTAX\_ERROR\_FALSELY\_CONSTRUCTED\_MESSAGE = 6;

}

enum Transport {

TRANSPORT\_UNSPECIFIED = 0;

TRANSPORT\_RESOURCE\_UNAVAILABLE = 1;

}

}

}

message CriticalityDiagnostics {

repeated IeCriticalityDiagnosticsItem ie\_criticality\_diagnostics\_list = 1;

message IeCriticalityDiagnosticsItem {

optional IeCriticality ie\_criticality = 1;

optional IeId ie\_id = 2;

optional TypeOfError type\_of\_error = 3;

}

enum IeCriticality {

IE\_CRITICALITY\_REJECT = 0;

IE\_CRITICALITY\_IGNORE = 1;

IE\_CRITICALITY\_NOTIFY = 2;

}

enum IeId {

IE\_ID\_CAUSE = 0;

IE\_ID\_CRITICALITY\_DIAGNOSTICS = 1;

IE\_ID\_E2\_ACTION\_ID = 2;

IE\_ID\_E2\_ACTIONS\_ADMITTED\_LIST = 3;

IE\_ID\_E2\_ACTIONS\_NOT\_ADMITTED\_LIST = 4;

IE\_ID\_E2\_CALL\_PROCESS\_ID = 5;

IE\_ID\_E2\_CONTROL\_ACK\_REQUEST = 6;

IE\_ID\_E2\_CONTROL\_HEADER = 7;

IE\_ID\_E2\_CONTROL\_MESSAGE = 8;

IE\_ID\_E2\_CONTROL\_OUTCOME = 9;

IE\_ID\_E2\_CONTROL\_TIMER = 10;

IE\_ID\_E2\_INDICATION\_HEADER = 11;

IE\_ID\_E2\_INDICATION\_MESSAGE = 12;

IE\_ID\_E2\_INDICATION\_SN = 13;

IE\_ID\_E2\_INDICATION\_TYPE = 14;

IE\_ID\_E2\_REQUEST\_ID = 15;

IE\_ID\_E2\_SUBSCRIPTION\_DETAILS = 16;

IE\_ID\_GLOBAL\_E2\_NODE\_ID = 17;

IE\_ID\_PROCEDURE\_TRANSACTION\_ID = 18;

IE\_ID\_RAN\_FUNCTION\_ID = 19;

IE\_ID\_SEQUENCE\_OF\_ACTIONS = 20;

}

enum TypeOfError {

TYPE\_OF\_ERROR\_NOT\_UNDERSTOOD = 0;

TYPE\_OF\_ERROR\_MISSING = 1;

}

}

// --------------E--------------

message E2ActionDefinition {

optional bytes value = 1;

};

message E2ActionId {

optional uint32 value = 1;

};

message E2CallProcessId {

optional bytes value = 1;

};

message E2ControlHeader {

optional bytes value = 1;

};

message E2ControlMessage {

optional bytes value = 1;

};

message E2ControlOutcome {

optional bytes value = 1;

};

message E2EventTriggerDefinition {

optional bytes value = 1;

};

message E2IndicationHeader {

optional bytes value = 1;

};

message E2IndicationMessage {

optional bytes value = 1;

};

message E2IndicationSN {

optional uint32 value = 1;

};

message E2RequestId {

optional uint32 value = 1;

};

// From E2AP Clause 9.2.13

message E2SubsequentAction {

optional SubsequentActionType subsequent\_action\_type = 1;

optional TimeToWait e2\_time\_to\_wait = 2;

enum SubsequentActionType {

SUBSEQUENT\_ACTION\_TYPE\_CONTINUE = 0;

SUBSEQUENT\_ACTION\_TYPE\_WAIT = 1;

};

enum TimeToWait {

TIME\_TO\_WAIT\_W1MS = 0;

TIME\_TO\_WAIT\_W2MS = 1;

TIME\_TO\_WAIT\_W5MS = 2;

TIME\_TO\_WAIT\_W10MS = 3;

TIME\_TO\_WAIT\_W20MS = 4;

TIME\_TO\_WAIT\_W30MS = 5;

TIME\_TO\_WAIT\_W40MS = 6;

TIME\_TO\_WAIT\_W50MS = 7;

TIME\_TO\_WAIT\_W100MS = 8;

TIME\_TO\_WAIT\_W200MS = 9;

TIME\_TO\_WAIT\_W500MS = 10;

TIME\_TO\_WAIT\_W1S = 11;

TIME\_TO\_WAIT\_W2S = 12;

TIME\_TO\_WAIT\_W5S = 13;

TIME\_TO\_WAIT\_W10S = 14;

TIME\_TO\_WAIT\_W20S = 15;

TIME\_TO\_WAIT\_W60S = 16;

}

};

### 6.3.5 Message Transfer Syntax

#### 6.3.5.1 Solution 1: SCTP with Protocol Buffers

The message transfer syntax for Solution 1 is serialized Protocol Buffers [4].

# 7 SDL API

## 7.1 Overview

### 7.1.1 Protocol Stack

In this version of specification, the default protocol stack for the SDL APIs is shown on Figure 7.1.1-1.

Note: The security protocols are specified in [6].

形状

低可信度描述已自动生成

**Figure 7.1.1-1: Protocol stack for SDL APIs**

The SDL APIs use SCTP with Protocol Buffers [4] as the application layer serialization protocol.

### 7.1.2 List of SDL APIs, procedures and messages

The SDL API procedures introduced in section 9.5 of [2] are categorized under the SDL APIs. These APIs include:

* SDL Information API: enabling storage and retrieval of information by xApps.

These APIs are summarized in Table 7.1.2-1.

Table 7.1.2-1: List of SDL APIs

|  |  |
| --- | --- |
| API Name | Description |
| SDL\_Information\_API | Near-RT RIC platform service for SDL information operation |

The table 7.1.2-2 lists the API procedures for each SDL API.

Table 7.1.2-2: List of enablement API procedures

|  |  |  |
| --- | --- | --- |
| API Name | API procedure | Communication type |
| SDL\_Information\_API | SDL Information Subscribe | Request/Response |
| SDL Information Push | Notify |
| SDL Information Update Notify | Notify |
| SDL Information Fetch | Request/Response |

The table 7.1.2-3 lists the messages for each SDL API procedure.

Table 7.1.2-3: List of enablement API messages

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **API Procedure** | **Initiated by** | **Initiating Message** | **Outcome Message** | |
| **Successful Outcome** | **Unsuccessful Outcome** |
| SDL Information Subscribe | xApp | Subscribe Information REQUEST | Subscribe Information RESPONSE | Subscribe Information FAILURE |
| SDL Information Push | Near-RT RIC platform | Information PUSH | - | - |
| SDL Information Update Notify | Near-RT RIC platform | INFORMATION UPDATE NOTIFY | - | - |
| SDL Information Fetch | xApp | INFORMATION FETCH REQUEST | INFORMATION FETCH RESPONSE | INFORMATION FETCH FAILURE |

## 7.2 SDL API Procedures

### 7.2.1 SDL Client Registration procedure

### 7.2.2 SDL Client Deregistration procedure

### 7.2.3 SDL Subscribe Information procedure

#### 7.2.3.1 General

This procedure is used by the xApp to subscribe to the SDL for the information that it is interested to obtain. The xApp can optionally define a set of filters to specify the type of the data to be subscribed. The Near-RT RIC platform acknowledges the subscribe results in the response message.

#### 7.2.3.2 Successful Operation

@startuml

skinparam ParticipantPadding 50

skinparam BoxPadding 10

skinparam lifelineStrategy solid

participant "Near-RT RIC Platform" as near

participant "xApp" as app

app -> near: SUBSCRIBE INFORMATION REQUEST

near -> app: SUBSCRIBE INFORMATION RESPONSE

@enduml

Graphical user interface, text, application, chat or text message, email

Description automatically generated

**Figure 7.2.3.2-1: Subscribe Information procedure, successful operation**

The xApp initiates the procedure by sending the SUBSCRIBE INFORMATION REQUEST message containing the *Information Type* IE to be subscribed and the *Delivery Method* IE of the information. The xApp can optionally specify the *Subscribe Filter List* IE to further define the subset of the data to be subscribed within the Information Type.

When the Near-RT RIC platform receives the request, it searches the local database for the requested information. If the optional filter definition list is specified, the Near-RT RIC platform performs further filter evaluation to find the matching data. The following operation is conducted to evaluate the data filtering:

* A Filter Definition is satisfied when the logical statement “X p Y” is True, where:
  + X is the filter type, represented by the *Filter Type* IE,
  + p is the filter operator, represented by the *Filter Condition* IE,
  + Y is the set of the target values, represented by the *Filter Condition Value* IE,
* Multiple filter conditions can be specified, in which case, a logical AND will be performed across all filters.

After the Near-RT RIC platform processes the request, it responds to the xApp with the SUBSCRIBE INFORMATION RESPONSE or SUBSCRIBE INFORMATION FAILURE message.

#### 7.2.3.3 Unsuccessful Operation

@startuml

skinparam ParticipantPadding 50

skinparam BoxPadding 10

skinparam lifelineStrategy solid

participant "Near-RT RIC Platform" as near

participant "xApp" as app

app -> near: SUBSCRIBE INFORMATION REQUEST

near -> app: SUBSCRIBE INFORMATION FAILURE

@enduml

Graphical user interface, text, application, chat or text message, email

Description automatically generated

**Figure 7.2.3.3-1: Subscribe Information procedure, unsuccessful operation**

If the Near-RT RIC platform can not accept the request it shall respond with the SUBSCRIBE INFORMATION FAILURE message and the proper cause. If the message contains *Time To Wait* IE the xApp shall wait at least the indicated time before reinitiating the request to the Near-RT RIC platform.

#### 7.2.3.4 Abnormal Conditions

If the Near-RT RIC platform receives a SUBSCRIBE INFORMATION REQUEST message containing *Information Type* IE that is not supported by the Near-RT RIC platform, or the optional filter definition is invalid, the Near-RT RIC platform shall send the SUBSCRIBE INFORMATION FAILURE message to the xApp with an appropriate cause value.

### 7.2.4 SDL Information Push procedure

#### 7.2.4.1 General

This procedure is used by the Near-RT RIC platform to send the latest information from the local database to the xApp based on what the xApp has subscribed with the *Delivery Method* IE set to Push. It will be triggered by subsequent information updates in the local database that impact the subscribed information.

#### 7.2.4.2 Successful Operation

@startuml

skinparam ParticipantPadding 50

skinparam BoxPadding 10

skinparam lifelineStrategy solid

participant "Near-RT RIC Platform" as near

participant "xApp" as app

near -> app: INFORMATION PUSH

@enduml

Graphical user interface, text, application

Description automatically generated

**Figure 7.2.4.2-1: Information Push procedure, successful operation**

The Near-RT RIC initiates the procedure by sending the INFORMATION PUSH message containing the *Information Block* IE. Information Block can contain multiple *Information Unit* IEs due to various updates as indicated by the associated *Modification Type* IE.

#### 7.2.4.3 Unsuccessful Operation

Not applied.

#### 7.2.4.4 Abnormal Conditions

Not applied.

### 7.2.5 SDL Information Update Notify procedure

#### 7.2.5.1 General

This procedure is used by the Near-RT RIC platform to notify the xApp that certain information subscribed by the xApp has been updated and available to be fetched. The information is limited to that with *Delivery Method* IE set to Notify in the subscribe information request.

#### 7.2.5.2 Successful Operation

@startuml

skinparam ParticipantPadding 50

skinparam BoxPadding 10

skinparam lifelineStrategy solid

participant "Near-RT RIC Platform" as near

participant "xApp" as app

near -> app: INFORMATION UPDATE NOTIFY

@enduml

Graphical user interface, text, application

Description automatically generated

**Figure 7.2.5.2-1: Information Update Notify procedure, successful operation**

The Near-RT RIC platform initiates the procedure by sending the INFORMATION UPDATE NOTIFY message containing the *Notification List* IE, which includes the list of the *Notification Unit* IEs to indicate which E2 Node has updated the information type and the the cause of the update via the *Modification Type* IE.

Once the xApp has received the message, xApp can determine to retrieve the updated information through the Information Fetch procedure described in section 7.2.4.

#### 7.2.5.3 Unsuccessful Operation

Not applied.

#### 7.2.5.4 Abnormal Conditions

Not applied.

### 7.2.6 SDL Information Fetch procedure

#### 7.2.6.1 General

This procedure is used by the xApp to request certain information from the SDL. The xApp can optionally define a set of filters to further specify the type of the data to be fetched. The Near-RT RIC platform delivers the requested data in the response message.

#### 7.2.6.2 Successful Operation

@startuml

skinparam ParticipantPadding 50

skinparam BoxPadding 10

skinparam lifelineStrategy solid

participant "Near-RT RIC Platform" as near

participant "xApp" as app

app -> near: INFORMATION FETCH REQUEST

near -> app: INFORMATION FETCH RESPONSE

@enduml

**Graphical user interface, text, application, chat or text message, email

Description automatically generated**

**Figure 7.2.6.2-1: SDL Information Fetch procedure, successful operation**

The xApp initiates the procedure by sending the INFORMATION FETCH REQUEST message containing the Information Type IE, which specifies the information type xApp wishes to fetch. For each information, the xApp can optionally specify a further Fetch Filter List IE to define the subset of the data to be retrieved within the subscribed information. The filtering of data is exactly the same as the filtering mentioned for SDL Subscribe Information procedure [7.2.1].

When the Near-RT RIC platform receives the request, it searches the local database for the fetched information. If the optional filter condition list is specified, the Near-RT RIC platform performs further filter evaluation to find the matching data.

If the SDL contains matching data for the fetch request, the Near-RT RIC platform sends the requested data to the xApp in the INFORMATION FETCH RESPONSE message.

#### 7.2.6.3 Unsuccessful Operation

If the data corresponding to a fetch request is not found, the Near-RT RIC platform sends an INFORMATION FETCH FAILURE message with an appropriate failure cause.

#### 7.2.6.4 Abnormal Conditions

If the Near-RT RIC platform receives a INFORMATION FETCH REQUEST message containing *Information Type* IE that is not supported by the Near-RT RIC platform, or the optional filter definition is invalid, the Near-RT RIC platform shall send the INFORMATION FETCH FAILURE message to the xApp with an appropriate cause value.

## 7.3 Elements for SDL API Communication

### 7.3.1 General

Sub clauses 7.3.2 and 7.3.3 describe the structure of the messages and information elements required for the SDL API in tabular format.

### 7.3.2 Message Functional Definition and Content

#### 7.3.2.1 SDL CLIENT REGISTRATION REQUEST

#### 7.3.2.2 SDL CLIENT REGISTRATION RESPONSE

#### 7.3.2.3 SDL CLIENT REGISTRATION FAILURE

#### 7.3.2.4 SDL CLIENT DE-REGISTRATION REQUEST

#### 7.3.2.5 SDL CLIENT DE-REGISTRATION RESPONSE

#### 7.3.2.6 SUBSCRIBE INFORMATION REQUEST

This message is sent by an xApp to the Near-RT RIC platform to subscribe to the interested information.

Direction: xApp -> Near-RT RIC platform

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description | Criticality | Assigned Criticality |
| Message Type | M |  | 7.3.3.1 |  | YES | Reject |
| xApp Request ID | M |  | 7.3.3.2 |  | YES | Reject |
| Information Type | M |  | 7.3.3.3 |  | YES | Reject |
| Delivery Method | M |  | 7.3.3.4 |  | YES | Reject |
| **Subscribe Filter List** |  | *0..<maxFilterID>* |  | If the Subscribe Filter List IE is absent, no filtering is applied and all data in Information Type IE is included. If multiple filters are specified, a logical AND is performed across all filters. | - |  |
| >Filter Definition | M |  | 7.3.3.5 |  |  |  |

|  |  |
| --- | --- |
| Range bound | Explanation |
| *maxFilterID* | Maximum number of filter conditions to be applied on an information subscribe request. Value is 256. |

#### 7.3.2.7 SUBSCRIBE INFORMATION RESPONSE

This message is sent by the Near-RT RIC platform after successful acceptance of an xApp’s request to subscribe to the information in the SDL.

Direction: Near-RT RIC platform -> xApp

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description | Criticality | Assigned Criticality |
| Message Type | M |  | 7.3.3.1 |  | YES | Reject |
| xApp Request ID | M |  | 7.3.3.2 | Replicated from corresponding request message. | YES | Reject |

#### 7.3.2.8 SUBSCRIBE INFORMATION FAILURE

This message is sent by the Near-RT RIC platform after failure to accept an xApp’s request to subscribe to information in the SDL.

Direction: Near-RT RIC platform -> xApp

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description | Criticality | Assigned Criticality |
| Message Type | M |  | 7.3.3.1 |  | YES | Reject |
| xApp Request ID | M |  | 7.3.3.2 | Replicated from corresponding request messge. | YES | Reject |
| Subscription Reject Cause | M |  | 7.3.3.9 |  |  |  |
| Time to Wait | O |  | 7.3.3.13 | If absent, no time to wait is required. |  |  |

#### 7.3.2.9 INFORMATION PUSH

This message is sent by the Near-RT RIC platform to an xApp containing the information subscribed by the xApp.

Direction: Near-RT RIC platform -> xApp

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description | Criticality | Assigned Criticality |
| Message Type | M |  | 7.3.3.1 |  | YES | Reject |
| xApp Request ID | M |  | 7.3.3.2 | Corresponding to the subscription request message. |  |  |
| Information Type | M |  | 7.3.3.3 |  |  |  |
| **Information Block** |  | *1..<maxInfoUnits>* |  |  |  |  |
| > Modification Type | M |  | 7.3.3.10 |  |  |  |
| > Information Unit | M |  | 7.3.3.11 | Content depends on Information Type. |  |  |

|  |  |
| --- | --- |
| Range bound | Explanation |
| *maxInfoUnits* | Maximum number of information units that can be carried in an Information Block. Value is 256. |

#### 7.3.2.10 INFORMATION UPDATE NOTIFY

This message is sent by the Near-RT RIC platform to inform an xApp that some of the information subscribed by the xApp has been updated.

Direction: Near-RT RIC platform -> xApp

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description | Criticality | Assigned Criticality |
| Message Type | M |  | 7.3.3.1 |  | YES | Reject |
| xApp request ID | M |  | 7.3.3.2 | Corresponding to the subscription request message. | YES | Reject |
| Information Type | M |  | 7.3.3.3 |  |  |  |
| **Notification Block** |  | *1..<maxNotificationUnits>* |  |  |  |  |
| > Modification Type | M |  | 7.3.3.10 |  |  |  |
| > Notification Unit | M |  | [7.3.3.12](#_Notification_Unit) |  |  |  |

|  |  |
| --- | --- |
| Range bound | Explanation |
| *maxNotificationUnits* | Maximum number of notification units that can be carried in an Update Notify message. Value is 256. |

#### 7.3.2.11 INFORMATION FETCH REQUEST

This message is sent by an xApp to the Near-RT RIC platform to request certain information subscribed by the xApp.

Direction: xApp -> Near-RT RIC platform

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description | Criticality | Assigned Criticality |
| Message Type | M |  | 7.3.3.1 |  | YES | Reject |
| xApp request ID | M |  | 7.3.3.2 |  | YES | Reject |
| Information Type | M |  | 7.3.3.3 |  |  |  |
| **Fetch Filter List** |  | *0..<maxFilterID>* |  | If the *Filter List* IE is absent, no filtering is applied and all data in *Information Type* IE is included.  If multiple filter definitions are specified, a logical AND is performed across all filters. | - |  |
| >Filter Definition | M |  | 7.3.3.5 |  |  |  |

#### 7.3.2.12 INFORMATION FETCH RESPONSE

This message is sent by the Near-RT RIC platform to an xApp containing the information requested by the xApp.

Direction: Near-RT RIC platform -> xApp

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description | Criticality | Assigned Criticality |
| Message Type | M |  | 7.3.3.1 |  | YES | Reject |
| xApp Request ID | M |  | 7.3.3.2 | Corresponding to the fetch request message. | YES | Reject |
| Information Type | M |  | 7.3.3.3 |  |  |  |
| **Information Block** |  | *1..<maxInfoUnits>* |  |  |  |  |
| > Information Unit | M |  | 7.3.3.11 | Content depends on Information Type. |  |  |

#### 7.3.2.13 INFORMATION FETCH FAILURE

This message is sent by the Near-RT RIC platform to an xApp if the Information fetch request failed.

Direction: Near-RT RIC platform -> xApp

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description | Criticality | Assigned Criticality |
| Message Type | M |  | 7.3.3.1 |  | YES | Reject |
| xApp Request ID | M |  | 7.3.3.2 | Corresponding to the fetch request | YES | Reject |
| Fetch Failure Cause | M |  | 7.3.3.18 | Cause for failure | YES | Reject |

#### 7.3.2.14 STORE DATA REQUEST

#### 7.3.2.15 STORE DATA RESPONSE

#### 7.3.2.16 STORE DATA FAILURE

#### 7.3.2.17 MODIFY DATA REQUEST

#### 7.3.2.18 MODIFY DATA RESPONSE

#### 7.3.2.19 MODIFY DATA FAILURE

### 7.3.3 Information Element Definitions

#### 7.3.3.1 Message Type

The Message Type IE identifies the message being sent. It is mandatory for all messages.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE Type and Reference | Semantics Description |
| Message Type |  |  |  |  |
| >Procedure Type | M |  | CHOICE (SDL Subscription Information, SDL Information Push, SDL Information Update Notify, SDL Information Fetch, …) |  |
| >Type of Message | M |  | CHOICE (Request, Response, Failure, …) | Refer to Table 7.1.2-3 for applicable choices for each procedure type |

#### 7.3.3.2 xApp Request ID

The xApp request ID IE uniquely identifies each request made by the xApp. It is mandatory for all requests and needs to be replicated in their corresponding response.  
For SDL subscriptions, the xApp request ID is also used as the “subscription identifier” and is used in all related Information Push and Information Update Notify messages.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description |
| xApp Request ID | M |  | INTEGER (0 … 4294967295) |  |

#### 7.3.3.3 Information Type

The Information Type IE indicates the type of information the xApp wants to subscribe from the database.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description |
| Information Type | M |  | ENUMERATED (E2NodeList, E2NodeComponentConfig, E2NodeRanfunction, …) |  |

#### 7.3.3.4 Delivery Method

The Delivery Method IE indicates the delivery method in which the xApp wants the information to be delivered.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description |
| Delivery Method | M |  | ENUMERATED (Notify, Push, …) |  |

#### 7.3.3.5 Filter Definition

The Filter Definition IE defnes a filter on the information that the xApp wants to obtain from SDL.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description |
| Filter Type | M |  | 7.3.3.6 |  |
| Filter Condition | M |  | 7.3.3.7 |  |
| Filter Condition Value | M |  | 7.3.3.8 |  |

#### 7.3.3.6 Filter Type

The Filter Type IE defines the filtering condition type to identify the information the xApp wants to be delivered.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description |
| Filter Type | M |  | ENUMERATED (E2NodeID, ComponentType, ServiceModel, …)  Content depends on the Information Type. See following table for the allowed settings. |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Filter Type** | **Information Type** | | |
| E2NodeList | E2NodeComponentConfig | E2NodeRanFunction |
| E2NodeID | NO | YES | YES |
| ComponentType | YES | YES | NO |
| ServiceModel | YES | NO | YES |

#### 7.3.3.7 Filter Condition

The Filter Condition IE defines the filtering condition to identify the information the xApp wants to be delivered.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description |
| Filter Condition | M |  | ENUMERATED (contained In, not contained In, equals, greanterthan, lessthan, …) |  |

#### 7.3.3.8 Filter Condition Value

The Filter Condition Value IE defines the filtering condition value to identify the information the xApp wants to be delivered.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description |
| **CHOICE** *Filter Condition Value* |  |  |  |  |
| >E2Node ID List |  | 1..65536 |  |  |
| >>E2 Node ID |  |  | 7.3.3.14 |  |
| >>E2Node State |  |  | 7.3.3.20 |  |
| >E2 Node Component Type List |  | 1..65536 |  |  |
| >>E2 Node Component Type |  |  | 7.3.3.15 |  |
| >RAN function OID List |  | 1..65536 |  |  |
| >>RAN function OID |  |  | 7.3.3.16 |  |

#### 7.3.3.9 Subscription Reject Cause

The Subscription Reject Cause IE indicates the reason the SDL rejected the subscription.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description |
| Subscription Reject Cause | M |  | ENUMERATED (SDL reached capacity, Information type not supported, Invalid filter, Unspecified, …) |  |

#### 7.3.3.10 Modification Type

The Modification Type IE informs the type of modification that the SDL underwent.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description |
| Modification Type | M |  | ENUMERATED (Added, Modified, Removed, ….) |  |

#### 7.3.3.11 Information Unit

The Information Unit IE contains the information that the SDL sends to the xApp.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description |
| CHOICE Information Unit | M |  | Contents depend on Information Type. See following table for allowed settings. |  |
| **>E2 Node List Unit** |  |  |  |  |
| >>E2 Node ID |  |  | 7.3.3.14 |  |
| >>E2 Node State |  |  | 7.3.3.20 |  |
| **>E2 Node Component Config Unit** |  |  |  |  |
| >>E2 Node ID | M |  | 7.3.3.14 |  |
| **>>E2 Node Component Config List** |  | *1..<maxofE2NodeComponents>* |  |  |
| >>>E2 Node Component Type | M |  | 7.3.3.15 |  |
| >>>E2 Node Component ID | O |  | Defined in clause 9.2.32 of [3]. |  |
| >>>E2 Node Component Configuration | M |  | 7.3.3.19 |  |
| >>> Last Modified E2Node Version | M |  | 7.3.3.17 |  |
| **> RAN Function Unit** |  |  |  |  |
| >>E2NodeID | M |  | 7.3.3.14 |  |
| **>>RAN Function List** | M | *1..<maxofRANfunctionID>* | *maxofRANfunctionIDs* is defined in clause 9.1.2.2 of [3]. |  |
| >>> Ran Function ID | M |  | 10.1.7 |  |
| >>> RAN function Definition | M |  | Defined in clause 9.2.23 of [3]. |  |
| >>> RAN function OID | M |  | Defined in clause 9.2.31 of [3]. |  |
| >>> Last Modified E2Node Version | M |  | 7.3.3.17 |  |

|  |  |
| --- | --- |
| Information Type | Information Unit |
| E2NodeList | E2NodeList Unit |
| E2NodeComponentConfig | E2NodeComponentConfig Unit |
| E2NodeRanFunction | E2NodeRanfunction Unit |

#### 7.3.3.12 Notification Unit

The Notification Unit IE contains the notification the xApp wants to be delivered.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description |
| Notification Unit |  |  |  |  |
| >E2NodeID | M |  | 7.3.3.14 |  |
| >Current E2Node Version | M |  | INTEGER | SDL maintains version number for state of each E2Node. Increments it whenever any ranfunciton or component config changes.  This is the current version number for the E2NodeID. |
| >E2 Node State | O |  | 7.3.3.20 |  |

#### 7.3.3.13 Time to wait

This IE defines the minimum allowed waiting times.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description |
| Time to wait | M |  | INTEGER (1..300) | Unit in seconds. |

#### 7.3.3.14 E2 Node ID

Defined in 10.1.6.

#### 7.3.3.15 E2 Node Component Type

Follow the definition of E2 Node Component Interface Type defined in E2AP clause 9.2.26.

#### 7.3.3.16 RAN function Service Model OID

Follow the definition of RAN function Service model OID defined in E2AP clause 9.2.31.

#### 7.3.3.17 Last Modified E2Node Version

SDL maintains version number for state of each E2Node. Increments it whenever any RAN Function IE or E2 Node Component Config IE changes.

This is the version number when the particular RAN Function IE or E2 Node Component Config IE last changed.

#### 7.3.3.18 Fetch Failure Cause

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description |
| Fetch Failure Cause | M |  | ENUMERATED (Invalid filter, Information type not supported, No information present, Unspecified, …) |  |

#### 7.3.3.19 E2Node Component Configuration

Follow the definition of E2 Node Component Configuration Update defined in E2AP clause 9.2.27. However, all the 3GPP content would be configurations, and not configuration updates.

#### 7.3.3.20 E2Node State

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description |
| E2Node State | M |  | ENUMERATED (connected, reset, disconnected, …) |  |

|  |  |
| --- | --- |
| E2 Node State | Meaning |
| connected | E2 Node has completed E2 Setup procedure.  Near-RT RIC platform has at least one validated E2NodeComponentConfig and E2NodeRanFunction record available for xApps using SDL API.  xApp may use E2 related API to request E2 Subscription and E2 Control procedures |
| reset | E2 connection to E2 Node has been reset using E2AP Reset procedure.  Near-RT RIC platform has at least one validated E2NodeComponentConfig and E2NodeRanFunction record available for xApps using SDL API.  Previous E2 Subscriptions have been deleted  E2 related API services not available |
| disconnected | E2 connection to E2 Node is not available  xApp shall delete records for E2 Node |

### 7.3.4 Message and Information Element Abstract Syntax

7.3.4.1 Solution 1: SCTP with Protocol Buffers

7.3.4.1.1 Usage of SCTP with Protocol Buffers

This solution uses Protocol Buffers as message encoding, and SCTP as the transport protocol.

The PDU on the SCTP connection uses the structure as defined in Section 10.1.8 where its *API Payload* IE serves as a container for individual API messages in Section 7.3.4.1.2 which can be SdlInformationApiPayload.

Table 7.3.4.1.1-1 lists the SDL API specifications included in this document.

|  |  |  |  |
| --- | --- | --- | --- |
| **API Name / Contents** | **Clause** | **Version** | **File Name** |
| SDL\_Information\_API | 7.3.4.1.2.1 | 1.0.0 | sdl\_information\_api\_ver\_1\_0\_0.proto |
| SDL API common information elements | 7.3.4.1.3 | 1.0.0 | sdl\_apis\_common\_ies\_ver\_1\_0\_0.proto |
| Near-RT RIC API common information elements | 10.2.1 | 1.0.0 | near\_rt\_ric\_apis\_common\_ies\_ver\_1\_0\_0.proto |

##### 7.3.4.1.2 API Definitions

###### 7.3.4.1.2.1 SDL\_Information\_API

syntax = "proto3";

package ricapi.sdl\_information\_api.v1;

import "near\_rt\_ric\_apis\_common\_ies\_ver\_1\_0\_0.proto";

import "sdl\_apis\_common\_ies\_ver\_1\_0\_0.proto";

// top-level PDU

message SdlInformationApiPayload {

oneof procedure\_type {

SdlSubscribeInformationProc sdl\_subscribe\_information\_proc = 1;

SdlInformationFetchProc sdl\_information\_fetch\_proc = 2;

SdlInformationPushProc sdl\_information\_push\_proc = 3;

SdlInformationUpdateNotifyProc sdl\_information\_update\_notify\_proc = 4;

}

}

// ---------------- procedures of SDL\_Information\_API ----------------

message SdlSubscribeInformationProc {

oneof type\_of\_message {

SubscribeInformationRequest subscribe\_information\_request = 1;

SubscribeInformationResponse subscribe\_information\_response = 2;

SubscribeInformationFailure subscribe\_information\_failure = 3;

}

}

message SdlInformationFetchProc {

oneof type\_of\_message {

InformationFetchRequest information\_fetch\_request = 1;

InformationFetchResponse information\_fetch\_response = 2;

InformationFetchFailure information\_fetch\_failure = 3;

}

}

message SdlInformationPushProc {

oneof type\_of\_message {

InformationPush information\_push = 1;

}

}

message SdlInformationUpdateNotifyProc {

oneof type\_of\_message {

InformationUpdateNotify information\_update\_notify = 1;

}

}

// ---------------- messages of SDL\_Information\_API ----------------

// -------- messages of SDL Subscribe Information procedure --------

message SubscribeInformationRequest {

optional sdl\_apis\_common\_ies.v1.XappRequestId xapp\_request\_id = 1;

optional sdl\_apis\_common\_ies.v1.InformationType information\_type = 2;

optional sdl\_apis\_common\_ies.v1.DeliveryMethod delivery\_method = 3;

repeated SubscribeFilterItem subscribe\_filter\_list = 4;

message SubscribeFilterItem {

optional sdl\_apis\_common\_ies.v1.FilterDefinition filter\_definition = 1;

}

}

message SubscribeInformationResponse {

optional sdl\_apis\_common\_ies.v1.XappRequestId xapp\_request\_id = 1;

}

message SubscribeInformationFailure {

optional sdl\_apis\_common\_ies.v1.XappRequestId xapp\_request\_id = 1;

optional sdl\_apis\_common\_ies.v1.SubscriptionRejectCause subscription\_reject\_cause = 2;

optional uint32 time\_to\_wait = 3;

}

// -------- messages of SDL Information Fetch procedure --------

message InformationFetchRequest {

optional sdl\_apis\_common\_ies.v1.XappRequestId xapp\_request\_id = 1;

optional sdl\_apis\_common\_ies.v1.InformationType information\_type = 2;

repeated FetchFilterItem fetch\_filter\_list = 3;

message FetchFilterItem {

optional sdl\_apis\_common\_ies.v1.FilterDefinition filter\_definition = 1;

}

}

message InformationFetchResponse {

optional sdl\_apis\_common\_ies.v1.XappRequestId xapp\_request\_id = 1;

optional sdl\_apis\_common\_ies.v1.InformationType information\_type = 2;

repeated InformationBlockItem information\_block = 3;

message InformationBlockItem {

optional sdl\_apis\_common\_ies.v1.InformationUnit information\_unit = 1;

}

}

message InformationFetchFailure {

optional sdl\_apis\_common\_ies.v1.XappRequestId xapp\_request\_id = 1;

optional sdl\_apis\_common\_ies.v1.FetchFailureCause fetch\_failure\_cause = 2;

}

// -------- messages of SDL Information Push procedure --------

message InformationPush {

optional sdl\_apis\_common\_ies.v1.XappRequestId xapp\_request\_id = 1;

optional sdl\_apis\_common\_ies.v1.InformationType information\_type = 2;

repeated InformationBlockItem information\_block = 3;

message InformationBlockItem {

optional sdl\_apis\_common\_ies.v1.ModificationType modification\_type = 1;

optional sdl\_apis\_common\_ies.v1.InformationUnit information\_unit = 2;

}

}

// -------- messages of SDL Information Update Notify procedure --------

message InformationUpdateNotify {

optional sdl\_apis\_common\_ies.v1.XappRequestId xapp\_request\_id = 1;

optional sdl\_apis\_common\_ies.v1.InformationType information\_type = 2;

repeated NotificationBlockItem notification\_block = 3;

message NotificationBlockItem {

optional sdl\_apis\_common\_ies.v1.ModificationType modification\_type = 1;

optional sdl\_apis\_common\_ies.v1.NotificationUnit notification\_unit = 2;

}

}

##### 7.3.4.1.3 Information Element Definitions

syntax = "proto3";

package ricapi.sdl\_apis\_common\_ies.v1;

import "near\_rt\_ric\_apis\_common\_ies\_ver\_1\_0\_0.proto";

// ----------------------------- Information Elements -----------------------------

// --------------A--------------

// From E2AP Clause 9.2.32

message AmfName {

optional string value = 1;

}

// --------------D--------------

enum DeliveryMethod {

DELIVERY\_METHOD\_NOTIFY = 0;

DELIVERY\_METHOD\_PUSH = 1;

}

// --------------E--------------

// From E2AP Clause 9.2.27

message E2NodeComponentConfiguration {

optional bytes e2\_node\_component\_request\_part = 1;

optional bytes e2\_node\_component\_response\_part = 2;

}

// From E2AP Clause 9.2.32

message E2NodeComponentId {

oneof e2\_node\_component\_interface\_type {

NgInterfaceType ng = 1;

XnInterfaceType xn = 2;

E1InterfaceType e1 = 3;

F1InterfaceType f1 = 4;

W1InterfaceType w1 = 5;

S1InterfaceType s1 = 6;

X2InterfaceType x2 = 7;

}

message NgInterfaceType {

optional AmfName amf\_name = 1;

}

message XnInterfaceType {

optional GlobalNgRanNodeId global\_ng\_ran\_node\_id = 1;

}

message E1InterfaceType {

optional near\_rt\_ric\_apis\_common\_ies.v1.GnbCuUpId gnb\_cu\_up\_id = 1;

}

message F1InterfaceType {

optional near\_rt\_ric\_apis\_common\_ies.v1.GnbDuId gnb\_du\_id = 1;

}

message W1InterfaceType {

optional near\_rt\_ric\_apis\_common\_ies.v1.NgEnbDuId ng\_enb\_du\_id = 1;

}

message S1InterfaceType {

optional MmeName mme\_name = 1;

}

message X2InterfaceType {

optional near\_rt\_ric\_apis\_common\_ies.v1.GlobalEnbId global\_enb\_id = 1;

optional near\_rt\_ric\_apis\_common\_ies.v1.GlobalEnGnbId global\_en\_gnb\_id = 2;

}

}

// From E2AP Clause 9.2.26

enum E2NodeComponentType {

E2\_NODE\_COMPONENT\_TYPE\_NG = 0;

E2\_NODE\_COMPONENT\_TYPE\_XN = 1;

E2\_NODE\_COMPONENT\_TYPE\_E1 = 2;

E2\_NODE\_COMPONENT\_TYPE\_F1 = 3;

E2\_NODE\_COMPONENT\_TYPE\_W1 = 4;

E2\_NODE\_COMPONENT\_TYPE\_S1 = 5;

E2\_NODE\_COMPONENT\_TYPE\_X2 = 6;

}

enum E2NodeState {

E2\_NODE\_STATE\_CONNECTED = 0;

E2\_NODE\_STATE\_RESET = 1;

E2\_NODE\_STATE\_DISCONNECTED = 2;

}

// --------------F--------------

enum FetchFailureCause {

FETCH\_FAILURE\_CAUSE\_UNSPECIFIED = 0;

FETCH\_FAILURE\_CAUSE\_INVALID\_FILTER = 1;

FETCH\_FAILURE\_CAUSE\_INFORMATION\_TYPE\_NOT\_SUPPORTED = 2;

FETCH\_FAILURE\_CAUSE\_NO\_INFORMATION\_PRESENT = 3;

}

enum FilterCondition {

FILTER\_CONDITION\_CONTAINTED\_IN = 0;

FILTER\_CONDITION\_NOT\_CONTAINTED\_IN = 1;

FILTER\_CONDITION\_EQUALS = 2;

FILTER\_CONDITION\_GREATER\_THAN = 3;

FILTER\_CONDITION\_LESS\_THAN = 4;

}

message FilterConditionValue {

oneof filter\_condition\_value\_type {

E2NodeIdList e2\_node\_id\_list = 1;

E2NodeComponentTypeList e2\_node\_component\_type\_list = 2;

RanFunctionOidList ran\_function\_oid\_list = 3;

}

message E2NodeIdList {

repeated E2NodeIdItem list = 1;

message E2NodeIdItem {

optional near\_rt\_ric\_apis\_common\_ies.v1.GlobalE2NodeId e2\_node\_id = 1;

optional E2NodeState e2\_node\_state = 2;

}

}

message E2NodeComponentTypeList {

repeated E2NodeComponentTypeItem list = 1;

message E2NodeComponentTypeItem {

optional E2NodeComponentType e2\_node\_component\_type = 1;

}

}

message RanFunctionOidList {

repeated RanFunctionOidItem list = 1;

message RanFunctionOidItem {

optional RanFunctionOid ran\_function\_oid = 1;

}

}

}

message FilterDefinition {

optional FilterType filter\_type = 1;

optional FilterCondition filter\_condition = 2;

optional FilterConditionValue filter\_condition\_value = 3;

}

enum FilterType {

FILTER\_TYPE\_E2\_NODE\_ID = 0;

FILTER\_TYPE\_COMPONENT\_TYPE = 1;

FILTER\_TYPE\_SERVICE\_MODEL = 2;

}

// --------------G--------------

// From E2AP Clause 9.2.32

message GlobalNgRanNodeId {

oneof ng\_ran\_node\_choice {

near\_rt\_ric\_apis\_common\_ies.v1.GlobalGnbId global\_gnb\_id = 1;

near\_rt\_ric\_apis\_common\_ies.v1.GlobalNgEnbId global\_ng\_enb\_id = 2;

}

}

// --------------I--------------

enum InformationType {

INFORMATION\_TYPE\_E2NODELIST = 0;

INFORMATION\_TYPE\_E2NODECOMPONENTCONFIG = 1;

INFORMATION\_TYPE\_E2NODERANFUNCTION = 2;

}

message InformationUnit {

oneof information\_unit\_type {

E2NodeListUnit e2\_node\_list\_unit = 1;

E2NodeComponentConfigUnit e2\_node\_component\_config\_unit = 2;

RanFunctionUnit ran\_function\_unit = 3;

}

message E2NodeListUnit {

optional near\_rt\_ric\_apis\_common\_ies.v1.GlobalE2NodeId e2\_node\_id = 1;

optional E2NodeState e2\_node\_state = 2;

}

message E2NodeComponentConfigUnit {

optional near\_rt\_ric\_apis\_common\_ies.v1.GlobalE2NodeId e2\_node\_id = 1;

repeated E2NodeComponentConfigItem e2\_node\_component\_config\_list = 2;

message E2NodeComponentConfigItem {

optional E2NodeComponentType e2\_node\_component\_type = 1;

optional E2NodeComponentId e2\_node\_component\_id = 2;

optional E2NodeComponentConfiguration e2\_node\_component\_configuration = 3;

optional LastModifiedE2NodeVersion last\_modified\_e2\_node\_version = 4;

}

}

message RanFunctionUnit {

optional near\_rt\_ric\_apis\_common\_ies.v1.GlobalE2NodeId e2\_node\_id = 1;

repeated RanFunctionItem ran\_function\_list = 2;

message RanFunctionItem {

optional near\_rt\_ric\_apis\_common\_ies.v1.RanFunctionId ran\_function\_id = 1;

optional RanFunctionDefinition ran\_function\_definition = 2;

optional RanFunctionOid ran\_function\_oid = 3;

optional LastModifiedE2NodeVersion last\_modified\_e2\_node\_version = 4;

}

}

}

// --------------L--------------

message LastModifiedE2NodeVersion {

optional uint32 value = 1;

}

// --------------M--------------

enum ModificationType {

TYPE\_ADDED = 0;

TYPE\_MODIFIED = 1;

TYPE\_REMOVED = 2;

}

// From E2AP Clause 9.2.32

message MmeName {

optional string value = 1;

}

// --------------N--------------

message NotificationUnit {

optional near\_rt\_ric\_apis\_common\_ies.v1.GlobalE2NodeId e2\_node\_id = 1;

optional LastModifiedE2NodeVersion current\_e2\_node\_version = 2;

optional E2NodeState e2\_node\_state = 3;

}

// --------------R--------------

// From E2AP Clause 9.2.23

message RanFunctionDefinition {

optional bytes value = 1;

}

// From E2AP Clause 9.2.31

message RanFunctionOid {

optional string value = 1;

}

// --------------S--------------

enum SubscriptionRejectCause {

SUBSCRIPTION\_REJECT\_CAUSE\_UNSPECIFIED = 0;

SUBSCRIPTION\_REJECT\_CAUSE\_SDL\_REACHED\_CAPACITY = 1;

SUBSCRIPTION\_REJECT\_CAUSE\_INFORMATION\_TYPE\_NOT\_SUPPORTED = 2;

SUBSCRIPTION\_REJECT\_CAUSE\_INVALID\_FILTER = 3;

}

// --------------T--------------

message TimeToWait {

optional uint32 value = 1;

}

// --------------X--------------

message XappRequestId {

optional uint32 value = 1;

}

### 7.3.5 Message Transfer Syntax

7.3.5.1 Solution 1: SCTP with Protocol Buffers

The message transfer syntax for Solution 1 is serialized Protocol Buffers [4].

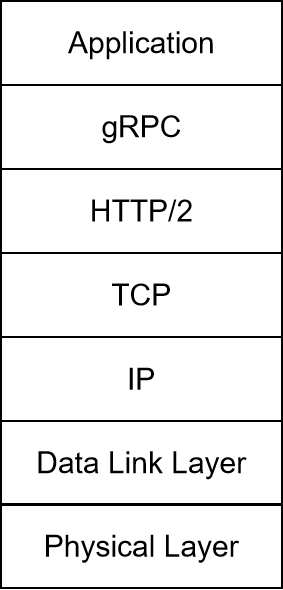
# 8 Management API

## 8.1 Overview

### 8.1.1 Protocol Stack

In this version of specification, the default protocol stack for the Management APIs is shown on Figure 8.1.1-1.

Note: The security protocols are specified in [6].



**Figure 8.1.1-1: Protocol stack for Management APIs**

The Management APIs use gRPC [5] with Protocol Buffers [4] as the application layer serialization protocol.

### 8.1.2 List of Management APIs, procedures and messages

The Management API procedures introduced in section 9.4 of [2] are categorized under the Management APIs. These APIs include:

* xApp Registration API: enabling registration of xApps with the Near-RT RIC platform.

These APIs are summarized in Table 8.1.2-1.

Table 8.1.2-1: List of Management APIs

|  |  |
| --- | --- |
| API Name | Description |
| Mgmt\_xApp\_Registration\_API | Near-RT RIC platform service for xApp registration |

The table 8.1.2-2 lists the API procedures for each Management API.

Table 8.1.2-2: List of Management API procedures

|  |  |  |
| --- | --- | --- |
| API Name | API procedure | Communication type |
| Mgmt\_xApp\_Registration\_API | xApp Registration | Request/Response |

The table 8.1.2-3 lists the messages for each management API procedure.

Table 8.1.2-3: List of Management API messages

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **API Procedure** | **Initiated by** | **Initiating Message** | **Outcome Message** | |
| **Successful Outcome** | **Unsuccessful Outcome** |
| xApp Registration | xApp | XAPP REGISTRATION REQUEST | XAPP REGISTRATION SUCCESS | XAPP REGISTRATION FAILURE |

## 8.2 Management API Procedures

### 8.2.1 xApp Registration Procedure

#### 8.2.1.1 General

This procedure is used by an xApp who wants to register in Near-RT RIC platform.

#### 8.2.1.2 Successful Operation

@startuml

Skin rose

skinparam ParticipantPadding 50

skinparam BoxPadding 10

skinparam lifelineStrategy solid

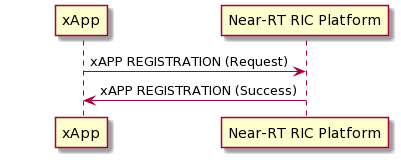
participant "xApp" as app

participant "Near-RT RIC Platform" as near

app -> near: xAPP REGISTRATION (Request)

near -> app: xAPP REGISTRATION (Success)

@enduml



**Figure 8.2.1.2-1: xApp Registration procedure, successful operation**

The xApp initiates the procedure by sending an xApp Registration (Request) message to the Near-RT-RIC platform. The message contains a unique *Procedure Transaction ID* IE assigned by the initiating xApp, the optional *xApp Description* IE*,* and the *Required APIs and Data Types* IE that specifies the APIs and data types needed for normal operation of the xApp.

At reception of the xApp Registration (Request) message, the Near-RT-RIC platform shall:

- Check xApp’s compatibility with the Near-RT RIC platform, using information in the *Required APIs and Data Types* IE.

- Assign an xApp ID and create the profile for the xApp.

If the registration request is successfully fulfilled, the Near-RT RIC platform shall send an xApp Registration (Success) message back to the xApp.

In the xApp Registration (Success) message, the Near-RT RIC platform shall include the associated *Procedure Transaction ID* IE and the *xApp ID* IE.

#### 8.2.1.3 Unsuccessful Operation

@startuml

Skin rose

skinparam ParticipantPadding 50

skinparam BoxPadding 10

skinparam lifelineStrategy solid

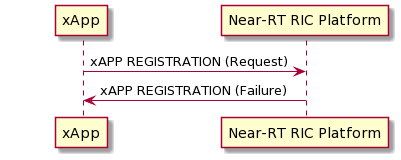
participant "xApp" as app

participant "Near-RT RIC Platform" as near

app -> near: xAPP REGISTRATION (Request)

near -> app: xAPP REGISTRATION (Failure)

@enduml



**Figure 8.2.1.3-1: xApp Registration procedure, unsuccessful operation**

If the registration request fails, the Near-RT RIC platform shall send an xApp Registration (Failure) message back to the xApp. The message shall include the associated *Procedure Transaction ID* IE replicated from the corresponding request and appropriate cause.

#### 8.2.1.4 Abnormal Conditions

Not applicable.

## 8.3 Elements for Management API Communication

### 8.3.1 General

### 8.3.2 Message Functional Definition and Content

#### 8.3.2.1 xAPP REGISTRATION REQUEST

This message is sent by an xApp to the Near-RT RIC platform for xApp Registration.

Direction: xApp -> Near-RT RIC platform

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **IE/Group Name** | **Presence** | **Range** | **IE type and reference** | **Semantics description** | **Criticality** | **Assigned Criticality** |
| Message Type | M |  | 8.3.3.1 |  | YES | Reject |
| Procedure Transaction ID | M |  | 10.1.1 | Value assigned by the initiating xApp. | YES | Reject |
| xApp Description | O |  | 8.3.3.2 |  | YES | Ignore |
| Required APIs and Data Types | M |  |  |  | YES | Reject |
| > List of Required APIs |  | 0.. <maxnoofAPIs> |  | The API names and versions required by the xApp | - | - |
| >> API Name | M |  | 10.1.3 |  | - | - |
| >> API Version | M |  | 10.1.4 |  | - | - |
| > List of A1 Policy Types |  | 0.. <maxnoofA1PolicyTypes> |  | The A1 Policy Types required by the xApp | - | - |
| >> A1 Policy Type ID | M |  | 8.3.3.3 |  | - | - |
| > List of E2 Service Models |  | 0.. <maxnoofE2SMs> |  | The E2 Service Models required by the xApp | - | - |
| >> E2 Service Model Name | M |  | 8.3.3.4 |  | - | - |
| >> E2 Service Model Version | M |  | 8.3.3.5 |  | - | - |

|  |  |
| --- | --- |
| **Range bound** | **Explanation** |
| maxnoofAPIs | Maximum no. of APIs supported by the xApp. Value is 1024. |
| maxnoofA1PolicyTypes | Maximum no. of A1 Policy Types supported by the xApp. Value is 1024. |
| maxnoofE2SMs | Maximum no. of E2SMs supported by the xApp. Value is 1024. |

#### 8.3.2.2 xAPP REGISTRATION SUCCESS

This message is sent by the Near-RT RIC platform after the xApp’s registration request is successfully accepted.

Direction: Near-RT RIC platform -> xApp

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **IE/Group Name** | **Presence** | **Range** | **IE type and reference** | **Semantics description** | **Criticality** | **Assigned Criticality** |
| Message Type | M |  | 8.3.3.1 |  | YES | Reject |
| Procedure Transaction ID | M |  | 10.1.1 | Value replicated from corresponding request. | YES | Reject |
| xApp ID | M |  | 10.1.2 |  | YES | Reject |

#### 8.3.2.3 xAPP REGISTRATION FAILURE

This message is sent by the Near-RT RIC platform after the xApp’s registration request is failed.

Direction: Near-RT RIC platform -> xApp

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **IE/Group Name** | **Presence** | **Range** | **IE type and reference** | **Semantics description** | **Criticality** | **Assigned Criticality** |
| Message Type | M |  | 8.3.3.1 |  | YES | Reject |
| Procedure Transaction ID | M |  | 10.1.1 | Value replicated from corresponding request. | YES | Reject |
| Cause | M |  | 8.3.3.6 |  | YES | Reject |

#### 8.3.2.4 xAPP CONFIGURATION REQUEST

#### 8.3.2.5 xAPP CONFIGURATION RESPONSE

#### 8.3.2.6 xAPP CONFIGURATION FAILURE

### 8.3.3 Information Element Definitions

8.3.3.1 Message Type

This IE uniquely identifies the message being sent. It is mandatory for all messages.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **IE/Group Name** | **Presence** | **Range** | **IE type and reference** | **Semantics description** |
| Message Type |  |  |  |  |
| >Procedure Type | M |  | CHOICE (xApp Registration, …) |  |
| >Type of Message | M |  | CHOICE (Request, Success, Failure, …) | Refer to Table 8.1.2-3 for applicable choices for each procedure type |

8.3.3.2 xApp Description

This IE provides generic information related to the xApp such as its name, vendor and purpose.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **IE/Group Name** | **Presence** | **Range** | **IE type and reference** | **Semantics description** |
| xApp Description | M |  | STRING |  |

8.3.3.3 A1 Policy Type ID

Defined in 10.1.5.

8.3.3.4 E2 Service Model Name

This IE specifies the name of E2 Service Model supported by the xApp.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **IE/Group Name** | **Presence** | **Range** | **IE type and reference** | **Semantics description** |
| E2 Service Model Name | M |  | STRING | The E2SM short name as in Table 5-1 of [8] |

8.3.3.5 E2 Service Model Version

This IE is used to indicate the version of E2 Service Model supported by the xApp.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **IE/Group Name** | **Presence** | **Range** | **IE type and reference** | **Semantics description** |
| E2 Service Model Version | M |  | STRING |  |

8.3.3.6 Cause

This IE is used to indicate the reason for a particular event for the Management APIs.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **IE/Group Name** | **Presence** | **Range** | **IE type and reference** | **Semantics description** |
| Cause | M |  | ENUMERATED  (Unspecified, API Not Supported, A1 Policy Type Not Supported, E2 Service Model Not Supported, …) |  |

### 8.3.4 Message and Information Element Abstract Syntax

#### 8.3.4.1 Solution 1: gRPC with Protocol Buffers

##### 8.3.4.1.1 Usage of gRPC with Protocol Buffers

This solution uses Protocol Buffers as message encoding, and also as the Interface Definition Language (IDL) for gRPC [5].

Table 8.3.4.1.1-1 lists the mapping of management APIs and corresponding gRPC services.

Table 8.3.4.1.1-1: Mapping of management APIs and gRPC services

|  |  |
| --- | --- |
| API Name | gRPC service(s) |
| Mgmt\_Xapp\_Registration\_API | Mgmt\_Xapp\_Registration\_API |

A procedure of an Management API is mapped to a *gRPC method.*

Table 8.3.4.1.1-2 lists the Management API specifications included in this document.

Table 8.3.4.1.1-2: Management API specifications

|  |  |  |  |
| --- | --- | --- | --- |
| API Name / Contents | Clause | Version | File Name |
| Mgmt\_Xapp\_Registration\_API | 8.3.4.1.2.1 | 1.0.0 | mgmt\_xapp\_registration\_api\_ver\_1\_0\_0.proto |
| Management API common information elements | 8.3.4.1.3 | 1.0.0 | mgmt\_apis\_common\_ies\_ver\_1\_0\_0.proto |
| Near-RT RIC API common information elements | 10.2.1 | 1.0.0 | near\_rt\_ric\_apis\_common\_ies\_ver\_1\_0\_0.proto |

##### 8.3.4.1.2 API Definitions

###### 8.3.4.1.2.1 Mgmt\_Xapp\_Registration\_API

syntax = "proto3";

package ricapi.mgmt\_xapp\_registration\_api.v1;

import "near\_rt\_ric\_apis\_common\_ies\_ver\_1\_0\_0.proto";

import "mgmt\_apis\_common\_ies\_ver\_1\_0\_0.proto";

// gRPC service and rpc definition

service mgmt\_Xapp\_Registration\_API {

rpc XappRegistrationProc(XappRegistrationInitMsg) returns (XappRegistrationOutMsg) {};

}

message XappRegistrationInitMsg {

optional XappRegistrationRequest xapp\_registration\_request = 1;

}

message XappRegistrationOutMsg {

oneof type\_of\_message {

XappRegistrationSuccess xapp\_registration\_success = 1;

XappRegistrationFailure xapp\_registration\_failure = 2;

}

}

// ---------------- messages of Mgmt\_Xapp\_Registration\_API ----------------

// -------- messages of xApp Registration procedure --------

message XappRegistrationRequest {

optional near\_rt\_ric\_apis\_common\_ies.v1.ProcedureTransactionId procedure\_transaction\_id = 1;

optional mgmt\_apis\_common\_ies.v1.XappDescription xapp\_description = 2;

optional RequiredApisAndDataTypes required\_apis\_and\_data\_types = 3;

message RequiredApisAndDataTypes {

repeated RequiredApiItem list\_of\_required\_apis = 1;

repeated A1PolicyTypeItem list\_of\_a1\_policy\_types = 2;

repeated E2ServiceModelItem list\_of\_e2\_service\_models = 3;

message RequiredApiItem {

optional near\_rt\_ric\_apis\_common\_ies.v1.ApiName api\_name = 1;

optional near\_rt\_ric\_apis\_common\_ies.v1.ApiVersion api\_version = 2;

}

message A1PolicyTypeItem {

optional near\_rt\_ric\_apis\_common\_ies.v1.A1PolicyTypeId a1\_policy\_type\_id = 1;

}

message E2ServiceModelItem {

optional mgmt\_apis\_common\_ies.v1.E2ServiceModelName e2\_service\_model\_name = 1;

optional mgmt\_apis\_common\_ies.v1.E2ServiceModelVersion e2\_service\_model\_version = 2;

}

}

}

message XappRegistrationSuccess {

optional near\_rt\_ric\_apis\_common\_ies.v1.ProcedureTransactionId procedure\_transaction\_id = 1;

optional near\_rt\_ric\_apis\_common\_ies.v1.XappId xapp\_id = 2;

}

message XappRegistrationFailure {

optional near\_rt\_ric\_apis\_common\_ies.v1.ProcedureTransactionId procedure\_transaction\_id = 1;

optional mgmt\_apis\_common\_ies.v1.Cause cause = 2;

}

##### 8.3.4.1.3 Information Element Definitions

syntax = "proto3";

package ricapi.mgmt\_apis\_common\_ies.v1;

import "near\_rt\_ric\_apis\_common\_ies\_ver\_1\_0\_0.proto";

// ----------------------------- Information Elements -----------------------------

// --------------C--------------

enum Cause {

UNSPECIFIED = 0;

API\_NOT\_SUPPORTED = 1;

A1\_POLICY\_TYPE\_NOT\_SUPPORTED = 2;

E2\_SERVICE\_MODEL\_NOT\_SUPPORTED = 3;

}

// --------------E--------------

message E2ServiceModelName {

optional string value = 1;

}

message E2ServiceModelVersion {

optional string value = 1;

}

// --------------X--------------

message XappDescription {

optional string value = 1;

}

### 8.3.5 Message Transfer Syntax

#### 8.3.5.1 Solution 1: gRPC with Protocol Buffers

The message transfer syntax for Solution 1 is serialized Protocol Buffers [4].

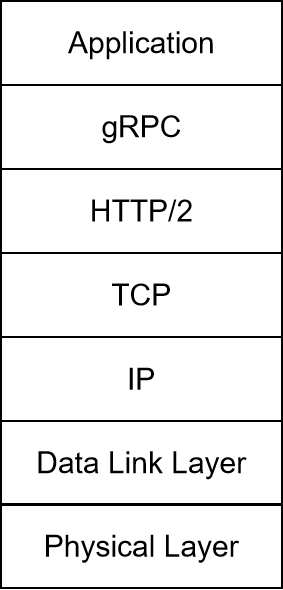
# 9 Enablement API

## 9.1 Overview

### 9.1.1 Protocol Stack

In this version of specification, the default protocol stack for the Enablement APIs is shown on Figure 9.1.1-1.

Note: The security protocols are specified in [6].



**Figure 9.1.1-1: Protocol stack for Enablement APIs**

The Enablement APIs use gRPC [5] with Protocol Buffers [4] as the application layer serialization protocol.

### 9.1.2 List of Enablement APIs, procedures and messages

The enablement API procedures introduced in section 9.5 of [2] are categorized under the enablement APIs. These APIs include:

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

These APIs are summarized in Table 9.1.2-1.

Table 9.1.2-1: List of Enablement APIs

|  |  |
| --- | --- |
| API Name | Description |
| Enabl\_Discovery\_API | Near-RT RIC platform service for API discovery |
| Enabl\_Events\_API | Near-RT RIC platform service for API related events |

The table 9.1.2-2 lists the API procedures for each enablement API.

Table 9.1.2-2: List of enablement API procedures

|  |  |  |
| --- | --- | --- |
| API Name | API procedure | Communication type |
| Enabl\_Discovery\_API | API Discovery | Request/Response |
| Enabl\_Events\_API | API Event Subcription | Request/Response |
| API Event Subcription Delete | Request/Response |
| API Event Notification | Notify |

The table 9.1.2-3 lists the messages for each enablement API procedure.

Table 9.1.2-3: List of enablement API messages

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **API Procedure** | **Initiated by** | **Initiating Message** | **Outcome Message** | |
| **Successful Outcome** | **Unsuccessful Outcome** |
| API Discovery | xApp | API DISCOVERY REQUEST | API DISCOVERY RESPONSE | API DISCOVERY FAILURE |
| API Event Subscription | xApp | API Event Subscription REQUEST | API Event Subscription RESPONSE | API Event Subscription FAILURE |
| API Event Subscription Delete | xApp | API Event Subscription DELETE REQUEST | API Event Subscription DELETE RESPONSE | API Event Subscription DELETE FAILURE |
| API Event Notification | Near-RT RIC platform | API Event NOTIFICATION | API Event NOTIFICATION ACK | - |

## 9.2 Enablement API Procedures

### 9.2.1 Enabl\_Discovery\_API Procedures

#### 9.2.1.1 General

The Enabl\_Discovery\_API allows an xApp to discover service APIs available at the Near-RT RIC platform.

The API procedure defined for Enabl\_Discovery\_API is shown in table 9.2.1.1-1.

Table 9.2.1.1-1: Procedure of the Enabl\_Discovery\_API

| **API Procedure** | **Description** | **Initiated by** |
| --- | --- | --- |
| API Discovery | This procedure is used by an xApp to discover Near-RT RIC APIs available at the Near-RT RIC platform. | xApp |

#### 9.2.1.2 API Discovery procedure

##### 9.2.1.2.1 Successful Operation

@startuml

skin rose

skinparam ParticipantPadding 50

skinparam BoxPadding 10

skinparam defaultFontSize 12

skinparam lifelineStrategy solid

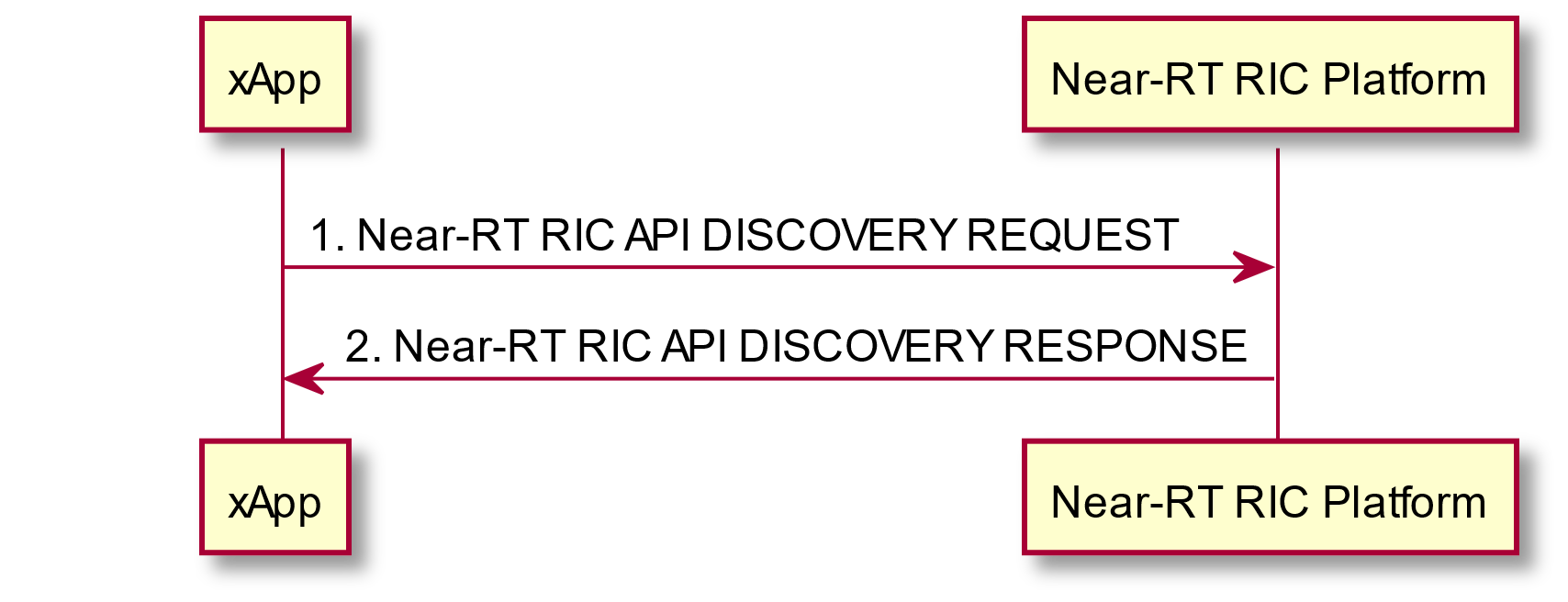
Participant xApp as "xApp"

Participant RICP as “Near-RT RIC Platform”

xApp -> RICP : 1. API DISCOVERY REQUEST

RICP -> xApp : 2. API DISCOVERY RESPONSE

@enduml



**Figure 9.2.1.2.1-1: API Discovery procedure, successful operation**

To discover Near-RT RIC APIs available at the API Enablement function, the xApp shall send an API DISCOVERY REQUEST message with the *xApp ID* IE and the *Query Parameters* IE to the Near-RT RIC platform.

Upon receiving the message, the Near-RT RIC platform shall:

1. verify the identity of the xApp and check if it is authorized to discover the Near-RT RIC APIs;

2. if the xApp is authorized to discover the Near-RT RIC APIs, the Near-RT RIC platform shall:

a. search the API registry for APIs matching the query criteria in the *Query Parameters* IE;

b. apply the discovery policy, if any, on the search results and filter the search results to obtain the list of service API description;

c. return the filtered search results in the API DISCOVERY RESPONSE message.

##### 9.2.1.2.2 Unsuccessful Operation

@startuml

skin rose

skinparam ParticipantPadding 50

skinparam BoxPadding 10

skinparam defaultFontSize 12

skinparam lifelineStrategy solid

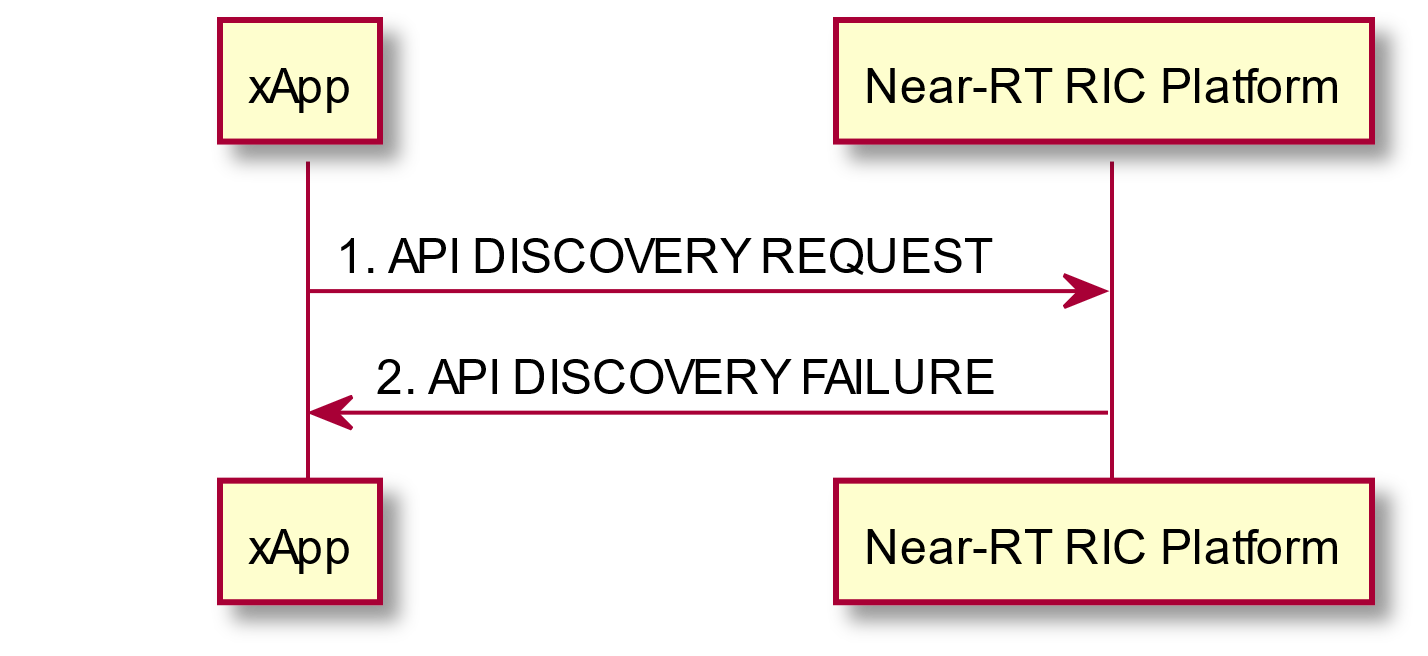
Participant xApp as "xApp"

Participant RICP as “Near-RT RIC Platform”

xApp -> RICP : 1. API DISCOVERY REQUEST

RICP -> xApp : 2. API DISCOVERY FAILURE

@enduml



**Figure 9.2.1.2.2-1: API Discovery procedure, unsuccessful operation**

If the Near-RT RIC platform rejects the request it shall respond with the API DISCOVERY FAILURE message with a proper cause. If the message contains the *Time to Wait* IE the xApp shall wait at least the indicated time before reinitiating the request to the Near-RT RIC platform.

##### 9.2.1.2.3 Abnormal Conditions

Not applied.

### 9.2.2 Enabl\_Events\_API Procedures

#### 9.2.2.1 General

The Enabl\_Events\_API allows an xApp to subscribe to and unsubscribe from events related to Near-RT RIC APIs and to receive notifications of the events.

Such events includes:

* Availability of Near-RT RIC API: event related to availability of Near-RT RIC APIs (e.g. active, inactive)
* Update of Near-RT RIC API: event related to change in Near-RT RIC API information

The API procedures defined for the Enabl\_Events\_API are shown in table 9.2.2.1-1.

Table 9.2.2.1-1: Operations of the Enabl\_Events\_API

| **API Procedure** | **Description** | **Initiated by** |
| --- | --- | --- |
| API Event Subscription | This procedure is used by xApp to subscribe for receiving Near-RT RIC API related events. | xApp |
| API Event Subscription Delete | This procedure is used by xApp to unsubscribe from receiving Near-RT RIC API related events. | xApp |
| API Event Notification | This procedure is used by Near-RT RIC to send a notification of Near-RT RIC API related event to an xApp. | Near-RT RIC platform |

#### 9.2.2.2 API Event Subscription procedure

##### 9.2.2.2.1 Successful Operation

@startuml

skin rose

skinparam ParticipantPadding 50

skinparam BoxPadding 10

skinparam defaultFontSize 12

skinparam lifelineStrategy solid

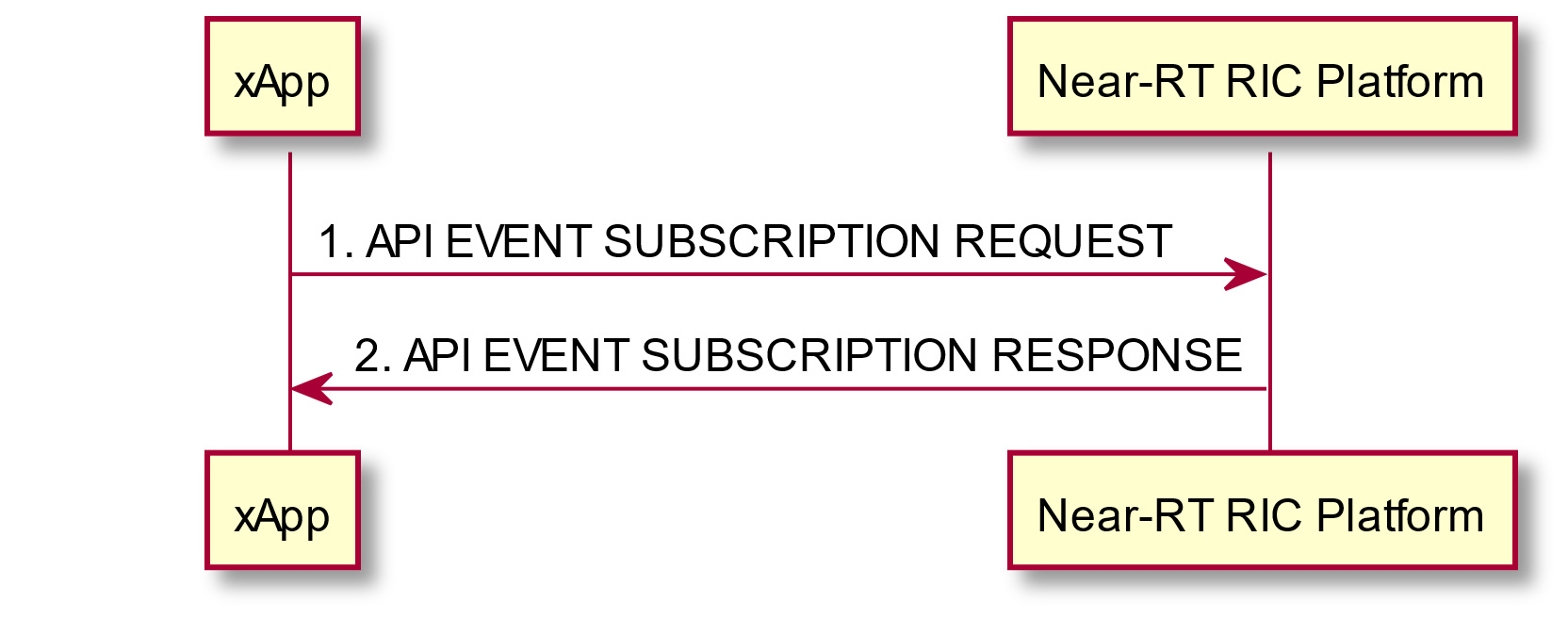
Participant xApp as "xApp"

Participant RICP as “Near-RT RIC Platform”

xApp -> RICP : 1. API EVENT SUBSCRIPTION REQUEST

RICP -> xApp : 2. API EVENT SUBSCRIPTION RESPONSE

@enduml



**Figure 9.2.2.2.1-1: API Event Subscription procedure, successful operation**

To subscribe to Near-RT RIC API related events, the xApp shall send an API EVENT SUBSCRIPTION REQUEST message to the Near-RT RIC platform. The message shall include the *xApp ID* IE, the *List of Events* IE, the *Event Notification Requirements* IE and the *Notification Destination* IE.

The xApp may also include additional event filter(s) in the *List of Event Filters* IE. Such event filters include:

- API ID(s) in the *API ID Filter* IE, if the event type is NEAR\_RT\_RIC\_API\_AVAILABLE, NEAR\_RT\_RIC\_API\_UNAVAILABLE or NEAR\_RT\_RIC\_API\_UPDATE;

Upon receiving the message, the API Enablement function shall:

1. verify the identity of the xApp and check if the xApp is authorized to subscribe to the Near-RT RIC API related events specified in the request message;

2. if the xApp is authorized to subscribe to the Near-RT RIC API related events, the Near-RT RIC platform shall prepare necessary resource for the subscription and associated notifications, according to the contents in the *List of Events* IE and the *Event Notification Requirements* IE, and send an API EVENT SUBSCRIPTION RESPONSE message back to the xApp. The response message shall include the assigned subscription ID in the *Subscription ID* IE.

##### 9.2.2.2.2 Unsuccessful Operation

@startuml

skin rose

skinparam ParticipantPadding 50

skinparam BoxPadding 10

skinparam defaultFontSize 12

skinparam lifelineStrategy solid

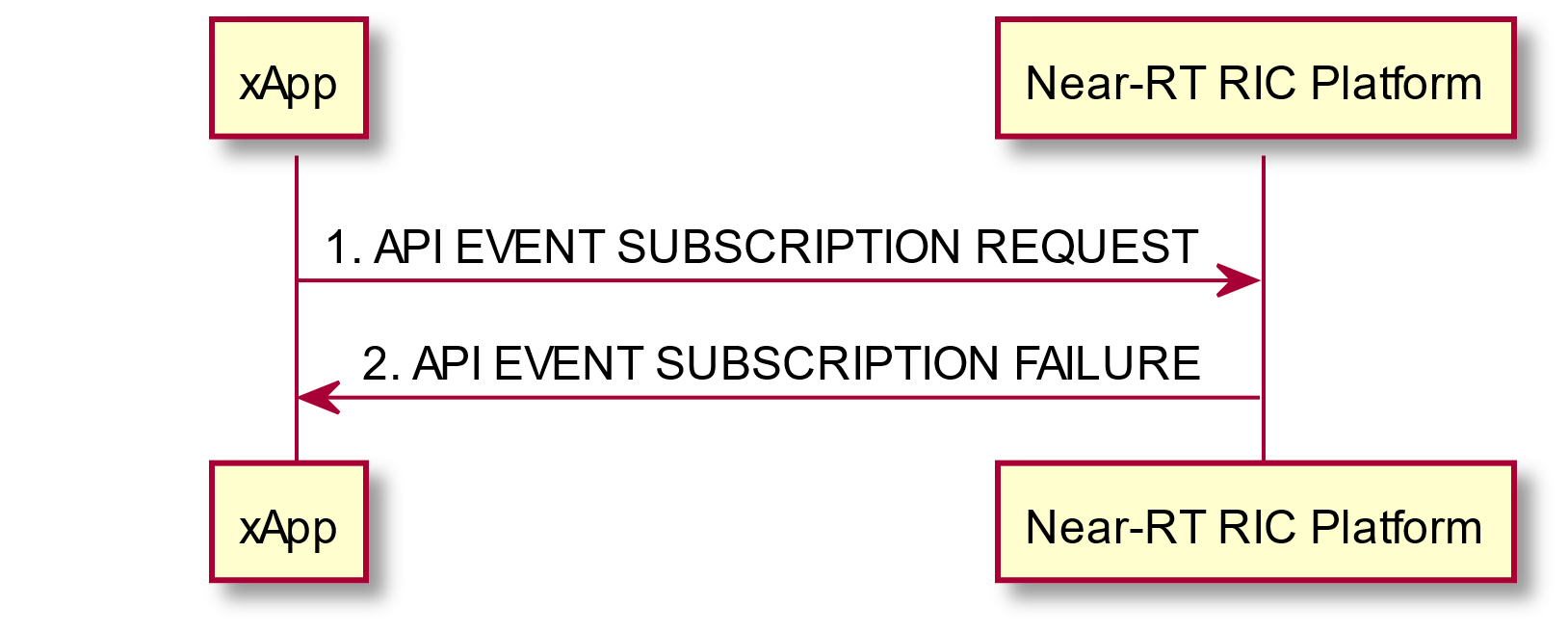
Participant xApp as "xApp"

Participant RICP as “Near-RT RIC Platform”

xApp -> RICP : 1. API EVENT SUBSCRIPTION REQUEST

RICP -> xApp : 2. API EVENT SUBSCRIPTION FAILURE

@enduml



**Figure 9.2.2.2.2-1: API Event Subscription procedure, unsuccessful operation**

If the Near-RT RIC platform rejects the request it shall respond with the API EVENT SUBSCRIPTION FAILURE message with proper cause. If the message contains *Time to Wait* IE the xApp shall wait at least the indicated time before reinitiating the request to the Near-RT RIC platform.

##### 9.2.2.2.3 Abnormal conditions

If the Near-RT RIC platform receives an API EVENT SUBSCRIPTION REQUEST message containing *Event Type* IE that is not supported by the Near-RT RIC platform, or the optional filter definition is invalid, the Near-RT RIC platform shall send the API EVENT SUBSCRIPTION FAILURE message to the xApp with an appropriate cause value.

#### 9.2.2.3 API Event Subscription Delete procedure

##### 9.2.2.3.1 Successful Operation

@startuml

skin rose

skinparam ParticipantPadding 50

skinparam BoxPadding 10

skinparam defaultFontSize 12

skinparam lifelineStrategy solid

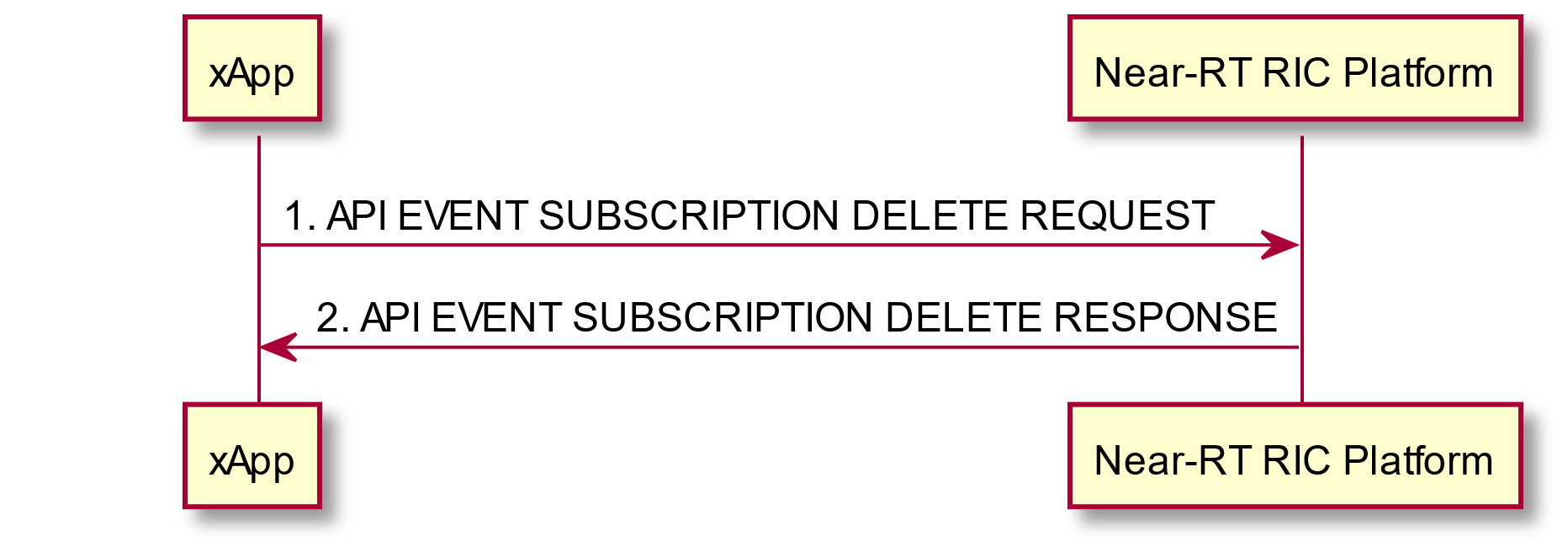
Participant xApp as "xApp"

Participant RICP as “Near-RT RIC Platform”

xApp -> RICP : 1. API EVENT SUBSCRIPTION DELETE REQUEST

RICP -> xApp : 2. API EVENT SUBSCRIPTION DELETE RESPONSE

@enduml



**Figure 9.2.2.3.1-1: API Event Subscription Delete procedure, successful operation**

To unsubscribe from Near-RT RIC API related events, the xApp shall send an API EVENT SUBSCRIPTION DELETE REQUEST message to the Near-RT RIC platform with the *xApp ID* IE and the *Subscription ID* IE.

Upon receiving the message, the Near-RT RIC platform shall:

1. verify the identity of the xApp and check if the xApp is authorized to unsubscribe from the Near-RT RIC API related events associated with the subscription ID; and

2. if the xApp is authorized to unsubscribe from the Near-RT RIC API related events, the Near-RT RIC platform shall delete the resource for the subscription and reply API EVENT SUBSCRIPTION DELETE RESPONSE to the xApp.

##### 9.2.2.3.2 Unsuccessful Operation

@startuml

skin rose

skinparam ParticipantPadding 50

skinparam BoxPadding 10

skinparam defaultFontSize 12

skinparam lifelineStrategy solid

Participant xApp as "xApp"

Participant RICP as “Near-RT RIC Platform”

xApp -> RICP : 1. API EVENT SUBSCRIPTION DELETE REQUEST

RICP -> xApp : 2. API EVENT SUBSCRIPTION DELETE FAILURE

@enduml

Graphical user interface, text

Description automatically generated with medium confidence

**Figure 9.2.2.3.2-1: API Event Subscription Delete procedure, unsuccessful operation**

If the Near-RT RIC platform rejects the request it shall respond with the API EVENT SUBSCRIPTION DELETE FAILURE message with proper cause. If the message contains *Time To Wait* IE the xApp shall wait at least the indicated time before reinitiating the request to the Near-RT RIC platform.

##### 9.2.2.3.3 Abnormal conditions

Not applied.

#### 9.2.2.4 API Event Notification procedure

##### 9.2.2.4.1 Successful Operation

@startuml

skin rose

skinparam ParticipantPadding 50

skinparam BoxPadding 10

skinparam defaultFontSize 12

skinparam lifelineStrategy solid

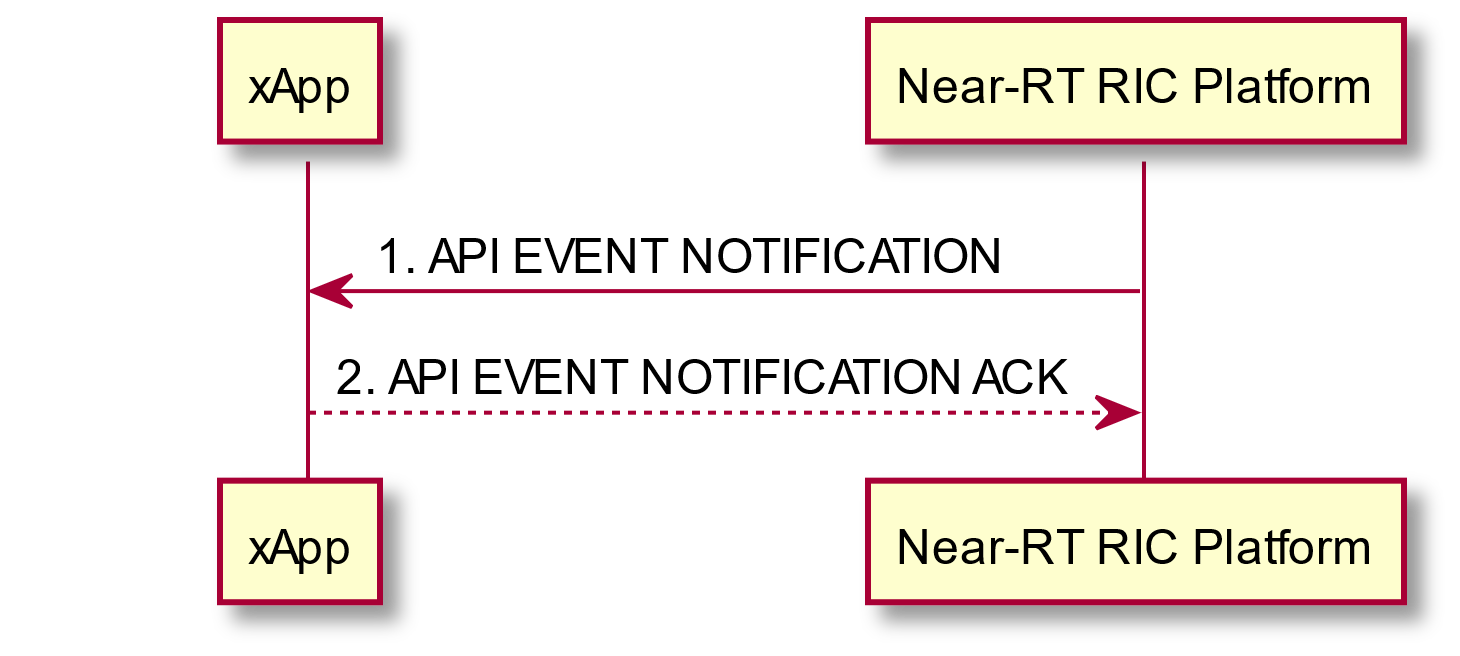
Participant xApp as "xApp"

Participant RICP as “Near-RT RIC Platform”

RICP -> xApp : 1. API EVENT NOTIFICATION

RICP <-- xApp : 2. API EVENT NOTIFICATION ACK

@enduml



**Figure 9.2.2.4.1-1: API Event Notification procedure, successful operation**

To notify a Near-RT RIC API related event, the Near-RT RIC platform shall send an API EVENT NOTIFICATION message using the *Notification Destination* IE received in the API Event Subscription procedure.

The Near-RT RIC platform may include details of the event in the *Event Detail* IE. Such details include:

- if the event type is Near-RT\_RIC\_API\_AVAILABLE or Near-RT\_RIC\_API\_UNAVAILABLE, the API IDs in the *List of API IDs* IE;

- if the event is Near-RT\_RIC\_API\_UPDATE, the updated API information in the *List of API Information* IE;

Upon receiving the message, the xApp may reply an API EVENT NOTIFICATION ACK message.

##### 9.2.2.4.2 Unsuccessful Operation

Not applied.

##### 9.2.2.4.3 Abnormal conditions

Not applied.

## 9.3 Elements for Enablement API Communication

### 9.3.1 General

### 9.3.2 Message Functional Definition and Content

#### 9.3.2.1 API DISCOVERY REQUEST

This message is sent by an xApp to the Near-RT RIC platform to discover the available Near-RT RIC APIs.

Direction: xApp -> Near-RT RIC platform

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description | Criticality | Assigned Criticality |
| Message Type | M |  | 9.3.3.1 |  |  |  |
| xApp ID | M |  | 10.1.2 |  |  |  |
| Query Parameters |  | 0..1 |  |  |  |  |
| >API Name | O |  | 9.3.3.6 |  |  |  |
| >API Version | O |  | 9.3.3.7 |  |  |  |
| >Protocol | O |  | 9.3.3.8 |  |  |  |
| >Data Format | O |  | 9.3.3.9 |  |  |  |
| >API Category | O |  | 9.3.3.10 |  |  |  |

#### 9.3.2.2 API DISCOVERY RESPONSE

This message is sent by Near-RT RIC platform to the xApp to carry the information of discovered Near-RT RIC APIs.

Direction: Near-RT RIC platform -> xApp

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description | Criticality | Assigned Criticality |
| Message Type | M |  | 9.3.3.1 |  |  |  |
| xApp ID | M |  | 10.1.2 |  |  |  |
| List of API Information |  | 0.. <maxnoofAPIDescriptions> |  | Description of the Near-RT RIC APIs as published by the provider(s). |  |  |
| >API Information | M |  | 9.3.3.2 |  |  |  |

#### 9.3.2.3 API DISCOVERY FAILURE

This message is sent by Near-RT RIC platform to the xApp for the case of failure of the discovery of the available near-RT RIC APIs.

Direction: Near-RT RIC platform -> xApp

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description | Criticality | Assigned Criticality |
| Message Type | M |  | 9.3.3.1 |  |  |  |
| xApp ID | M |  | 10.1.2 |  |  |  |
| Cause | M |  | 9.3.3.4 |  |  |  |
| Time to Wait | O |  | 9.3.3.5 |  |  |  |

#### 9.3.2.4 API EVENT SUBSCRIPTION REQUEST

This message is sent by an xApp to the Near-RT RIC platform to subscribe for events related to Near-RT RIC APIs.

Direction: xApp -> Near-RT RIC platform

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description | Criticality | Assigned Criticality |
| Message Type | M |  | 9.3.3.1 |  |  |  |
| xApp ID | M |  | 10.1.2 |  |  |  |
| List of Events |  | 1.. <maxnoofEventFilter> |  |  |  |  |
| >Event Type | M |  | 9.3.3.13 |  |  |  |
| >API ID Filter |  | 0.. <maxnoofAPIIDFiliters> |  | API identifiers that the event subscriber wants to know in the interested event. |  |  |
| >>API ID | M |  | 9.3.3.11 |  |  |  |
| Event Notification Requirements |  | 1 |  | Represents the reporting requirements of the event subscription. |  |  |
| >Notification Method | M |  | ENUMERATED (On Event, Periodic, ...) |  |  |  |
| >Notification Period | O |  | INTEGER (1.. 300, ...) | The duration between notifications expressed in unit of second. Shall be present when the *Notification Method* IE is set to " Periodic". |  |  |
| Notification Destination | M |  | 9.3.3.12 |  |  |  |

#### 9.3.2.5 API EVENT SUBSCRIPTION RESPONSE

This message is sent by Near-RT RIC platform to the xApp as successful response to API event subscription request.

Direction: Near-RT RIC platform -> xApp

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description | Criticality | Assigned Criticality |
| Message Type | M |  | 9.3.3.1 |  |  |  |
| xApp ID | M |  | 10.1.2 |  |  |  |
| Subscription ID | M |  | 9.3.3.3 |  |  |  |

#### 9.3.2.6 API EVENT SUBSCRIPTION FAILURE

This message is sent by Near-RT RIC platform to the xApp for the case of failure of the API event subscription request.

Direction: Near-RT RIC platform -> xApp

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description | Criticality | Assigned Criticality |
| Message Type | M |  | 9.3.3.1 |  |  |  |
| xApp ID | M |  | 10.1.2 |  |  |  |
| Cause | M |  | 9.3.3.4 |  |  |  |
| Time to Wait | O |  | 9.3.3.5 |  |  |  |

#### 9.3.2.7 API EVENT SUBSCRIPTION DELETE REQUEST

This message is sent by an xApp to the Near-RT RIC platform to unsubscribe for events related to Near-RT RIC APIs.

Direction: xApp -> Near-RT RIC platform

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description | Criticality | Assigned Criticality |
| Message Type | M |  | 9.3.3.1 |  |  |  |
| xApp ID | M |  | 10.1.2 |  |  |  |
| Subscription ID | M |  | 9.3.3.3 |  |  |  |

#### 9.3.2.8 API EVENT SUBSCRIPTION DELETE RESPONSE

This message is sent by Near-RT RIC platform to the xApp as successful response to API event subscription delete request.

Direction: Near-RT RIC platform -> xApp

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description | Criticality | Assigned Criticality |
| Message Type | M |  | 9.3.3.1 |  |  |  |
| xApp ID | M |  | 10.1.2 |  |  |  |
| Subscription ID | M |  | 9.3.3.3 |  |  |  |

#### 9.3.2.9 API EVENT SUBSCRIPTION DELETE FAILURE

This message is sent by Near-RT RIC platform to the xApp for the case of failure of the API event subscription delete request.

Direction: Near-RT RIC platform -> xApp

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description | Criticality | Assigned Criticality |
| Message Type | M |  | 9.3.3.1 |  |  |  |
| xApp ID | M |  | 10.1.2 |  |  |  |
| Cause | M |  | 9.3.3.4 |  |  |  |
| Time to Wait | O |  | 9.3.3.5 |  |  |  |

#### 9.3.2.10 API EVENT NOTIFICATION

This message is sent to a subscriber xApp from the Near-RT RIC platform to notify on an API event (based on subscription).

Direction: Near-RT RIC platform -> xApp

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description | Criticality | Assigned Criticality |
| Message Type | M |  | 9.3.3.1 |  |  |  |
| Subscription ID | M |  | 9.3.3.4 |  |  |  |
| Event Type | M |  | 9.3.3.13 |  |  |  |
| Event Detail |  | 0.. 1 |  |  |  |  |
| >List of API IDs |  | 0.. <maxnoofAPIIDs> |  | The API IDs which are available/unavailable in the relevant event. |  |  |
| >>API ID | M |  | 9.3.3.11 |  |  |  |
| >List of API Information |  | 0.. <maxnoofAPIInformation> |  | Description of the Near-RT RIC APIs which are to be updated. |  |  |
| >>API Information | M |  | 9.3.3.2 |  |  |  |

#### 9.3.2.11 API EVENT NOTIFICATION ACK

This message is optionally sent by the xApp to Near-RT RIC platform as acknowledgement to an API event notification.

Direction: xApp -> Near-RT RIC platform

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description | Criticality | Assigned Criticality |
| Message Type | M |  | 9.3.3.1 |  |  |  |
| Subscription ID | M |  | 9.3.3.3 |  |  |  |

### 9.3.3 Information Element Definitions

#### 9.3.3.1 Message Type

This IE identifies the message being sent. It is mandatory for all messages.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE Type and Reference | Semantics Description |
| Message Type |  |  |  |  |
| >Procedure Type | M |  | CHOICE (API Discovery, API Event Subscription, API Event Subscription Delete, API Event Notification, …) |  |
| >Type of Message | M |  | CHOICE (Request, Response, Failure, …) | Refer to Table 9.1.2-3 for applicable choices for each procedure type |

#### 9.3.3.2 API Information

This IE indicates the information of a Near-RT RIC API.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description |
| API Name | M |  | 9.3.3.6 |  |
| API ID | M |  | 9.3.3.11 |  |
| API Category | M |  | 9.3.3.10 |  |
| Description | O |  | STRING | Text description of the API |
| List of API Profiles |  | 1.. <maxnoofAPIProfiles> |  |  |
| >API Version | M |  | 9.3.3.7 |  |
| >Protocol | O |  | 9.3.3.8 |  |
| >Data Format | O |  | 9.3.3.9 |  |
| >Endpoint Information | O |  | 9.3.3.10 |  |

#### 9.3.3.3 Subscription ID

The IE uniquely identifies a subscription to API-related event(s).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description |
| Subscription ID | M |  | STRING | Identifier of the subscription resource to which the notification is related |

#### 9.3.3.4 Cause

This IE is used to indicate the reason for a particular event for the Enablement APIs.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description |
| Cause | M |  | ENUMERATED (Unspecified, ...) |  |

#### 9.3.3.5 Time to Wait

This IE specifies the time that xApp shall wait before reinitiating a request.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description |
| Time to Wait | M |  | INTEGER (1.. 300, ...) | Unit in second |

#### 9.3.3.6 API Name

Defined in 10.1.3.

#### 9.3.3.7 API Version

Defined in 10.1.4.

#### 9.3.3.8 Protocol

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description |
| Protocol | M |  | ENUMERATED (gRPC, SCTP, HTTP REST, …) | Protocol used by the Near-RT RIC API |

#### 9.3.3.9 Data Format

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description |
| Data Format | M |  | ENUMERATED (Proto3, JSON, ASN.1, …) | Data format used by the Near-RT RIC API |

#### 9.3.3.10 API Category

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description |
| API Category | M |  | ENUMERATED (E2 related APIs, SDL APIs, A1 related APIs, …) | The Near-RT RIC API category to which the API belongs |

#### 9.3.3.11 API ID

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description |
| API ID | M |  | STRING | API identifier assigned by the Near-RT RIC platform to the published Near-RT RIC API |

#### 9.3.3.12 Endpoint Information

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description |
| Domain Name | M (NOTE) |  | STRING |  |
| IPv4 Address | M (NOTE) |  | STRING |  |
| IPv6 Address | M (NOTE) |  | STRING |  |
| Port | O |  | INTEGER |  |
| NOTE: At least one of these IEs need to be present | | | | |

#### 9.3.3.13 Event Type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description |
| Event Type | M |  | ENUMERATED (NEAR\_RT\_RIC\_API\_AVAILABLE, NEAR\_RT\_RIC\_API\_UNAVAILABLE, NEAR\_RT\_RIC\_API\_API\_UPDATE, ...) |  |

Table 9.3.3.13-1: Meaning of Event Types

|  |  |
| --- | --- |
| value | Description |
| NEAR\_RT\_RIC\_API\_AVAILABLE | Events related to the availability of APIs after the APIs are published. |
| NEAR\_RT\_RIC\_API\_UNAVAILABLE | Events related to the unavailability of APIs after the APIs are unpublished. |
| NEAR\_RT\_RIC\_API\_API\_UPDATE | Events related to change in API information |

### 9.3.4 Message and Information Element Abstract Syntax

#### 9.3.4.1 Solution 1: gRPC with Protocol Buffers

##### 9.3.4.1.1 Usage of gRPC with Protocol Buffers

This solution uses Protocol Buffers as message encoding, and also as the Interface Definition Language (IDL) for gRPC [5].

Table 9.3.4.1.1-1 lists the mapping of enablement APIs and corresponding gRPC services.

Table 9.3.4.1.1-1: Mapping of enablement APIs and gRPC services

|  |  |
| --- | --- |
| API Name | gRPC service(s) |
| Enabl\_Discovery\_API | Enabl\_Discovery\_API |
| Enabl\_Events\_API | Enabl\_Events\_API, Enabl\_Events\_API\_Notification |

A procedure of an Enablement API is mapped to a *gRPC method.*

Table 9.3.4.1.1-2 lists the Enablement API specifications included in this document.

|  |  |  |  |
| --- | --- | --- | --- |
| API Name / Contents | Clause | Version | File Name |
| Enabl\_Discovery\_API | 9.3.4.1.2.1 | 1.0.0 | enabl\_discovery\_api\_ver\_1\_0\_0.proto |
| Enabl\_Events\_API | 9.3.4.1.2.2 | 1.0.0 | enabl\_events\_api\_ver\_1\_0\_0.proto |
| Enablement API common information elements | 9.3.4.1.3 | 1.0.0 | enabl\_apis\_common\_ies\_ver\_1\_0\_0.proto |
| Near-RT RIC API common information elements | 10.2.1 | 1.0.0 | near\_rt\_ric\_apis\_common\_ies\_ver\_1\_0\_0.proto |

##### 9.3.4.1.2 API Definitions

###### 9.3.4.1.2.1 Enabl\_Discovery\_API

syntax = "proto3";

package ricapi.enabl\_discovery\_api.v1;

import "near\_rt\_ric\_apis\_common\_ies\_ver\_1\_0\_0.proto";

import "enabl\_apis\_common\_ies\_ver\_1\_0\_0.proto";

// gRPC service and rpc definition

service Enabl\_Discovery\_API {

rpc ApiDiscoveryProc(ApiDiscoveryInitMsg) returns (ApiDiscoveryOutMsg) {};

}

message ApiDiscoveryInitMsg {

optional ApiDiscoveryRequest api\_discovery\_request = 1;

}

message ApiDiscoveryOutMsg {

oneof type\_of\_message {

ApiDiscoveryResponse api\_discovery\_response = 1;

ApiDiscoveryFailure api\_discovery\_failure = 2;

}

}

// ---------------- messages of Enabl\_Discovery\_API ----------------

// -------- messages of API Discovery procedure --------

message ApiDiscoveryRequest {

optional near\_rt\_ric\_apis\_common\_ies.v1.XappId xapp\_id = 1;

optional QueryParameters qps = 2;

message QueryParameters {

optional near\_rt\_ric\_apis\_common\_ies.v1.ApiName api\_name = 1;

optional near\_rt\_ric\_apis\_common\_ies.v1.ApiVersion api\_version = 2;

optional enabl\_apis\_common\_ies.v1.Protocol protocol = 3;

optional enabl\_apis\_common\_ies.v1.DataFormat data\_format = 4;

optional enabl\_apis\_common\_ies.v1.ApiCategory api\_category = 5;

};

};

message ApiDiscoveryResponse {

optional near\_rt\_ric\_apis\_common\_ies.v1.XappId xapp\_id = 1;

repeated enabl\_apis\_common\_ies.v1.ApiInformation list\_of\_api\_information = 2;

};

message ApiDiscoveryFailure {

optional near\_rt\_ric\_apis\_common\_ies.v1.XappId xapp\_id = 1;

optional enabl\_apis\_common\_ies.v1.Cause cause = 2;

optional uint32 time\_to\_wait = 3;

};

###### 9.3.4.1.2.2 Enabl\_Events\_API

syntax = "proto3";

package ricapi.enabl\_events\_api.v1;

import "near\_rt\_ric\_apis\_common\_ies\_ver\_1\_0\_0.proto";

import "enabl\_apis\_common\_ies\_ver\_1\_0\_0.proto";

// gRPC service and rpc definition

service Enabl\_Events\_API {

rpc ApiEventSubscriptionProc(ApiEventSubscriptionInitMsg) returns (ApiEventSubscriptionOutMsg) {};

rpc ApiEventSubscriptionDeleteProc(ApiEventSubscriptionDeleteInitMsg) returns (ApiEventSubscriptionDeleteOutMsg) {};

}

message ApiEventSubscriptionInitMsg {

optional ApiEventSubscriptionRequest api\_event\_subscription\_request = 1;

}

message ApiEventSubscriptionOutMsg {

oneof type\_of\_message {

ApiEventSubscriptionResponse api\_event\_subscription\_response = 1;

ApiEventSubscriptionFailure api\_event\_subscription\_failure = 2;

}

}

message ApiEventSubscriptionDeleteInitMsg {

optional ApiEventSubscriptionDeleteRequest api\_event\_subscription\_delete\_request = 1;

}

message ApiEventSubscriptionDeleteOutMsg {

oneof type\_of\_message {

ApiEventSubscriptionDeleteResponse api\_event\_subscription\_delete\_response = 1;

ApiEventSubscriptionDeleteFailure api\_event\_subscription\_delete\_failure = 2;

}

}

service Enabl\_Events\_API\_Notification {

rpc ApiEventNotificationProc(ApiEventNotificationInitMsg) returns (ApiEventNotificationOutMsg) {};

}

message ApiEventNotificationInitMsg {

optional ApiEventNotification api\_event\_notification = 1;

}

message ApiEventNotificationOutMsg {

oneof type\_of\_message {

ApiEventNotificationACK api\_event\_notification\_ack = 1;

}

}

// ---------------- messages of Enabl\_Events\_API ----------------

// -------- messages of API Event Subscription procedure --------

message ApiEventSubscriptionRequest {

optional near\_rt\_ric\_apis\_common\_ies.v1.XappId xapp\_id = 1;

repeated EventItem list\_of\_events = 2;

optional EventNotificationRequirements event\_notification\_requirements = 3;

optional enabl\_apis\_common\_ies.v1.EndpointInformation notification\_destination = 4;

message EventItem {

optional enabl\_apis\_common\_ies.v1.EventType event\_type = 1;

repeated ApiIdFilterItem api\_id\_filter = 2;

message ApiIdFilterItem {

optional enabl\_apis\_common\_ies.v1.ApiId api\_id = 1;

}

}

message EventNotificationRequirements {

optional NotificationMethod notification\_method = 1;

optional uint32 notification\_period = 2;

enum NotificationMethod {

NOTIFICATION\_METHOD\_ON\_EVENT = 0;

NOTIFICATION\_METHOD\_PERIODIC = 1;

}

}

};

message ApiEventSubscriptionResponse {

optional near\_rt\_ric\_apis\_common\_ies.v1.XappId xapp\_id = 1;

optional enabl\_apis\_common\_ies.v1.SubscriptionId subscription\_id = 2;

}

message ApiEventSubscriptionFailure {

optional near\_rt\_ric\_apis\_common\_ies.v1.XappId xapp\_id = 1;

optional enabl\_apis\_common\_ies.v1.Cause cause = 2;

optional uint32 time\_to\_wait = 3;

}

// -------- messages of API Event Subscription Delete procedure --------

message ApiEventSubscriptionDeleteRequest {

optional near\_rt\_ric\_apis\_common\_ies.v1.XappId xapp\_id = 1;

optional enabl\_apis\_common\_ies.v1.SubscriptionId subscription\_id = 2;

};

message ApiEventSubscriptionDeleteResponse {

optional near\_rt\_ric\_apis\_common\_ies.v1.XappId xapp\_id = 1;

optional enabl\_apis\_common\_ies.v1.SubscriptionId subscription\_id = 2;

}

message ApiEventSubscriptionDeleteFailure {

optional near\_rt\_ric\_apis\_common\_ies.v1.XappId xapp\_id = 1;

optional enabl\_apis\_common\_ies.v1.Cause cause = 2;

optional uint32 time\_to\_wait = 3;

}

// -------- messages of API Event Notification procedure --------

message ApiEventNotification {

optional enabl\_apis\_common\_ies.v1.SubscriptionId subscription\_id = 1;

optional enabl\_apis\_common\_ies.v1.EventType event\_type = 2;

optional EventDetail event\_detail = 3;

message EventDetail {

repeated enabl\_apis\_common\_ies.v1.ApiId list\_of\_api\_ids = 1;

repeated enabl\_apis\_common\_ies.v1.ApiInformation list\_of\_api\_information = 2;

};

};

message ApiEventNotificationACK {

optional enabl\_apis\_common\_ies.v1.SubscriptionId subscription\_id = 1;

};

##### 9.3.4.1.3 Information Element Definitions

syntax = "proto3";

package ricapi.enabl\_apis\_common\_ies.v1;

import "near\_rt\_ric\_apis\_common\_ies\_ver\_1\_0\_0.proto";

// ----------------------------- Information Elements -----------------------------

// --------------A--------------

enum ApiCategory {

APICATEGORY\_E2\_RELATED = 0;

APICATEGORY\_SDL = 1;

APICATEGORY\_A1\_RELATED = 2;

};

message ApiId {

optional string value = 1;

}

message ApiInformation {

optional near\_rt\_ric\_apis\_common\_ies.v1.ApiName api\_name = 1;

optional ApiId api\_id = 2;

optional ApiCategory api\_category = 3;

optional string description = 4;

repeated ApiProfile list\_of\_api\_profiles = 5;

message ApiProfile {

optional near\_rt\_ric\_apis\_common\_ies.v1.ApiVersion api\_version = 1;

optional Protocol protocol = 2;

optional DataFormat data\_format = 3;

optional EndpointInformation endpoint\_information = 4;

};

};

// --------------C--------------

enum Cause {

UNSPECIFIED = 0;

}

// --------------D--------------

enum DataFormat {

DATA\_FORMAT\_PROTO3 = 0;

DATA\_FORMAT\_JSON = 1;

DATA\_FORMAT\_ASN1 = 2;

};

// --------------E--------------

message EndpointInformation {

optional string domain\_name = 1;

optional string ipv4\_address = 2;

optional string ipv6\_address = 3;

optional uint32 port = 4;

};

enum EventType {

EVENT\_TYPE\_NEAR\_RT\_RIC\_API\_AVAILABLE = 0;

EVENT\_TYPE\_NEAR\_RT\_RIC\_API\_UNAVAILABLE = 1;

EVENT\_TYPE\_NEAR\_RT\_RIC\_API\_UPDATE = 2;

}

// --------------P--------------

enum Protocol {

PROTOCOL\_GRPC = 0;

PROTOCOL\_SCTP = 1;

PROTOCOL\_HTTP = 2;

};

// --------------S--------------

message SubscriptionId {

optional string value = 1;

}

### 9.3.5 Message Transfer Syntax

#### 9.3.5.1 Solution 1: gRPC with Protocol Buffers

The message transfer syntax for Solution 1 is serialized Protocol Buffers [4].

# 10 Common definitions across APIs

## 10.1 Information element definitions

### 10.1.1 PTI

The Procedure Transaction Identity (PTI) IE allows identifing a procedure among all ongoing parallel procedures of the same type initiated by the same RICAPI peer. The PTI value is assigned to the initiating message by the procedure initiator, and subsequent messages shall include the same PTI value as in the initiating message, if any. PTI is released when the procedure is completed.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE Type and Reference | Semantics Description |
| PTI | M |  | INTEGER (0..232-1) |  |

### 10.1.2 xApp ID

The xApp ID IE uniquely identifies an xApp..

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **IE/Group Name** | **Presence** | **Range** | **IE type and reference** | **Semantics description** |
| xApp ID | M |  | STRING | The format of the xApp ID shall be a Universally Unique Identifier (UUID) version 4, as described in [9]. |

### 10.1.3 API Name

The API Name IE indicates the Name of each Near-RT RIC API.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description |
| API Name | M |  | ENUMERATED (E2\_Subscription\_API, E2\_Indication\_API, E2\_Control\_API, E2\_Guidance\_API, A1\_Policy \_API, A1\_EI\_API, SDL\_Information\_API, ...) | Contains the Near-RT RIC API name |

The table 10.1.3-1 lists the API Names.

Table 10.1.3-1: List of API Names

|  |  |
| --- | --- |
| API Category | API Name |
| E2 related APIs | E2\_Subscription\_API, E2\_Indication\_API, E2\_Control\_API, E2\_Guidance\_API |
| A1 related APIs | A1\_Policy\_API, A1\_EI\_API |
| SDL APIs | SDL\_Information\_API |

### 10.1.4 API Version

The API Version IE indicates the Version of each Near-RT RIC API.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE type and reference | Semantics description |
| API Version | M |  | STRING | Contains the Near-RT RIC API Version |

### 10.1.5 A1 Policy Type ID

The A1 Policy Type ID IE identifies the type of one A1 policy.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE Type and Reference | Semantics Description |
| A1 Policy Type ID | M |  | STRING | Applicable values defined in [7]. |

### 10.1.6 Global E2 Node ID

Defined in 9.2.6 of [3].

### 10.1.7 RAN Function ID

Defined in 9.2.8 of [3].

### 10.1.8 SCTP API PDU

The SCTP API PDU IE serves as the outer layer encapsulation for the payload when SCTP is used as the transport protocol.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IE/Group Name | Presence | Range | IE Type and Reference | Semantics Description |
| API Name | M |  | 10.1.3 |  |
| API Version | M |  | 10.1.4 |  |
| API Payload | M |  | OCTET STRING |  |

## 10.2 Information element abstract syntax

### 10.2.1 Abstact syntax in Protocol Buffers

syntax = "proto3";

package ricapi.near\_rt\_ric\_apis\_common\_ies.v1;

// Information elements used across different API categories

// ----------------------------- Information Elements -----------------------------

// --------------A--------------

// From A1TD Clause 5.2

message A1PolicyTypeId {

optional string value = 1;

}

enum ApiName {

APINAME\_E2\_SUBSCRIPTION\_API = 0;

APINAME\_E2\_INDICATION\_API = 1;

APINAME\_E2\_CONTROL\_API = 2;

APINAME\_E2\_GUIDANCE\_API = 3;

APINAME\_A1\_POLICY\_API = 4;

APINAME\_A1\_EI\_API = 5;

APINAME\_SDL\_INFORMATION\_API = 6;

};

message ApiVersion {

optional string value = 1;

}

// --------------B--------------

// BitString definition

message BitString {

optional string value = 1;

optional uint32 len = 2;

}

// --------------G--------------

// From E2AP Clause 9.2.6, "Global E2 Node ID"

message GlobalE2NodeId {

oneof node\_type {

GlobalE2NodeGnbId gnb = 1;

GlobalE2NodeEnGnbId en\_gnb = 2;

GlobalE2NodeNgEnbId ng\_enb = 3;

GlobalE2NodeEnbId enb = 4;

}

message GlobalE2NodeGnbId {

optional GlobalGnbId global\_gnb\_id = 1;

optional GlobalEnGnbId global\_en\_gnb\_id = 2;

optional GnbCuUpId gnb\_cu\_up\_id = 3;

optional GnbDuId gnb\_du\_id = 4;

}

message GlobalE2NodeEnGnbId {

optional GlobalEnGnbId global\_en\_gnb\_id = 1;

optional GnbCuUpId en\_gnb\_cu\_up\_id = 2;

optional GnbDuId en\_gnb\_du\_id = 3;

}

message GlobalE2NodeNgEnbId {

optional GlobalNgEnbId global\_ng\_enb\_id = 1;

optional GlobalEnbId global\_enb\_id = 2;

optional NgEnbDuId ng\_enb\_du\_id = 3;

}

message GlobalE2NodeEnbId {

optional GlobalEnbId global\_enb\_id = 1;

}

}

// From E2AP Clause 9.2.6, as part of "Global E2 Node ID"

message GlobalEnbId {

optional PlmnIdentity plmn\_identity = 1;

oneof enb\_id\_choice {

BitString macro\_enb\_id = 2;

BitString home\_enb\_id = 3;

BitString short\_macro\_enb\_id = 4;

BitString long\_macro\_enb\_id = 5;

}

}

// From E2AP Clause 9.2.6, as part of "Global E2 Node ID"

message GlobalEnGnbId {

optional PlmnIdentity plmn\_identity = 1;

oneof en\_gnb\_id\_choice {

BitString en\_gnb\_id = 2;

}

}

// From E2AP Clause 9.2.6, as part of "Global E2 Node ID"

message GlobalGnbId {

optional PlmnIdentity plmn\_identity = 1;

oneof gnb\_id\_choice {

BitString gnb\_id = 2;

}

}

// From E2AP Clause 9.2.6, as part of "Global E2 Node ID"

message GlobalNgEnbId {

optional PlmnIdentity plmn\_identity = 1;

oneof ng\_enb\_id\_choice {

BitString macro\_ng\_enb\_id = 2;

BitString short\_macro\_ng\_enb\_id = 3;

BitString long\_macro\_ng\_enb\_id = 4;

}

}

// From E2AP Clause 9.2.6, as part of "Global E2 Node ID"

message GnbCuUpId {

optional uint64 value = 1;

}

// From E2AP Clause 9.2.6, as part of "Global E2 Node ID"

message GnbDuId {

optional uint64 value = 1;

}

// --------------N--------------

// From E2AP Clause 9.2.6, as part of "Global E2 Node ID"

message NgEnbDuId {

optional uint64 value = 1;

};

// --------------P--------------

// From E2AP Clause 9.2.6, as part of "Global E2 Node ID"

message PlmnIdentity {

optional bytes value = 1;

};

message ProcedureTransactionId {

optional uint32 value = 1;

};

// --------------R--------------

// From E2AP Clause 9.2.8, "RAN Function ID"

message RanFunctionId {

optional uint32 value = 1;

};

//---------------S--------------

// Outer layer encapsulation for SCTP based APIs

message SctpApiPdu {

optional ApiName api\_name = 1;

optional ApiVersion api\_version = 2;

optional bytes api\_payload = 3;

}

// --------------X--------------

message XappId {

optional string value = 1;

}

# Revision history

|  |  |  |
| --- | --- | --- |
| Date | Revision | Description |
| 2022.11.20 | 01.00 | Version inceremented for TSC approval. |
| 2022.11.16 | 00.06.00 | Enclosure of CR:  [VMW.AO-2022.11.11-WG3-CR-0003-RICAPI-Improvements on Proto3 definitions-v2.zip](https://oranalliance.atlassian.net/wiki/download/attachments/2523464076/VMW.AO-2022.11.11-WG3-CR-0003-RICAPI-Improvements%20on%20Proto3%20definitions-v2.zip?api=v2) agreed at RICARCH #79.  Editorial updates. |
| 2022.11.12 | 00.05.00 | Enclosure of CR:  [CMCC.AO-2022.11.09-WG3-CR-0010-RICAPI-Collection of Common IEs across API categories under Section 10-v1.zip](https://oranalliance.atlassian.net/wiki/download/attachments/2523464076/CMCC.AO-2022.11.09-WG3-CR-0010-RICAPI-Collection%20of%20Common%20IEs%20across%20API%20categories%20under%20Section%2010-v1.zip?api=v2) agreed at RICARCH TG 20221111 Ad-hoc meeting.  Editorial updates. |
| 2022.11.09 | 00.04.00 | Copyright statement update in accordance to O-RAN alliance rules.  Enclosure of CRs:  [CMCC.AO-2022.05.13-WG3-CR-0003-RICAPI-E2 related APIs overview\_v04.docx](https://oranalliance.atlassian.net/wiki/download/attachments/2523464076/CMCC.AO-2022.05.13-WG3-CR-0003-RICAPI-E2%20related%20APIs%20overview_v04.docx?api=v2) agreed at RICARCH #76.  [AsiaInfo.AO-2022.08.05-WG3-CR-0004-RICAPI-Cause and Criticality Diagnostics IE Definition-v04.docx](https://oranalliance.atlassian.net/wiki/download/attachments/2523464076/AsiaInfo.AO-2022.08.05-WG3-CR-0004-RICAPI-Cause%20and%20Criticality%20Diagnostics%20IE%20Definition-v04.docx?api=v2), and  [CMCC.AO-2022.09.16-WG3-CR-0006-RICAPI-A1 related APIs overview\_v01.docx](https://oranalliance.atlassian.net/wiki/download/attachments/2523464076/CMCC.AO-2022.09.16-WG3-CR-0006-RICAPI-A1%20related%20APIs%20overview_v01.docx?api=v2) agreed at RICARCH #77.  [CMCC.AO-2022.10.10-WG3-CR-0005-RICAPI-Proto3 Definitions for E2 related APIs-v1.zip](https://oranalliance.atlassian.net/wiki/download/attachments/2523464076/CMCC.AO-2022.10.10-WG3-CR-0005-RICAPI-Proto3%20Definitions%20for%20E2%20related%20APIs-v1.zip?api=v2) agreed at 202210 Madrid F2F RICARCH stream.  [CMCC-2022.08.17-WG3-CR-0004-RICAPI-Basic contents in Section 4-v3.docx](https://oranalliance.atlassian.net/wiki/download/attachments/2523464076/CMCC-2022.08.17-WG3-CR-0004-RICAPI-Basic%20contents%20in%20Section%204-v3.docx?api=v2),  [CMCC-2022.09.16-WG3-CR-0007-RICAPI-Enablement APIs overview for Section 9.1-v2.docx](https://oranalliance.atlassian.net/wiki/download/attachments/2523464076/CMCC-2022.09.16-WG3-CR-0007-RICAPI-Enablement%20APIs%20overview%20for%20Section%209.1-v2.docx?api=v2), and  [CMCC-2022.11.01-WG3-CR-0009-RICAPI-Alignment on Message Type IEs for all API categories-v1.docx](https://oranalliance.atlassian.net/wiki/download/attachments/2523464076/CMCC-2022.11.01-WG3-CR-0009-RICAPI-Alignment%20on%20Message%20Type%20IEs%20for%20all%20API%20categories-v1.docx?api=v2) agreed at RICARCH #78.  [CICT.AO-2021.06.01-WG3-CR-0004-procedural descriptions and msgs applied to A1-related API\_v9.zip](https://oranalliance.atlassian.net/wiki/download/attachments/2523464076/CICT.AO-2021.06.01-WG3-CR-0004-procedural%20descriptions%20and%20msgs%20applied%20to%20A1-related%20API_v9.zip?api=v2),  [CMCC.AO-2022.10.25-WG3-CR-0008-SDL APIs overview and Proto3 Definitions-v3.zip](https://oranalliance.atlassian.net/wiki/download/attachments/2523464076/CMCC.AO-2022.10.25-WG3-CR-0008-SDL%20APIs%20overview%20and%20Proto3%20Definitions-v3.zip?api=v2),  [AsiaInfo.AO-2022.10.11-WG3-CR-0005-RICAPI-xApp Registration Procedure Description and Messages-v3.docx](https://oranalliance.atlassian.net/wiki/download/attachments/2523464076/AsiaInfo.AO-2022.10.11-WG3-CR-0005-RICAPI-xApp%20Registration%20Procedure%20Description%20and%20Messages-v3.docx?api=v2),  [AsiaInfo.AO-2022.10.20-WG3-CR-0006-RICAPI-Proto3 Definitions for Management APIs-v1.zip](https://oranalliance.atlassian.net/wiki/download/attachments/2523464076/AsiaInfo.AO-2022.10.20-WG3-CR-0006-RICAPI-Proto3%20Definitions%20for%20Management%20APIs-v1.zip?api=v2), and  [CMCC.AO-2022.08.22-WG3-CR-0005-RICAPI-Proto3 Definitions for Enablement APIs-v4.zip](https://oranalliance.atlassian.net/wiki/download/attachments/2523464076/CMCC.AO-2022.08.22-WG3-CR-0005-RICAPI-Proto3%20Definitions%20for%20Enablement%20APIs-v4.zip?api=v2) agreed at RICARCH TG 20221108 Ad-hoc meeting.  Editorial updates. |
| 2022.08.10 | 00.03.00 | Clause assigned per API message.  Enclosure of CR:  [AsiaInfo.AO-2021.11.17-WG3-CR-0001-RICAPI-E2 Subscription API Procedure descriptions and messages\_v09.docx](https://oranalliance.atlassian.net/wiki/download/attachments/2210201622/AsiaInfo.AO-2021.11.17-WG3-CR-0001-RICAPI-E2%20Subscription%20API%20Procedure%20descriptions%20and%20messages_v09.docx?api=v2) agreed at RICARCH #74.  Minor editorial update. |
| 2022.06.06 | 00.02.00 | Adoption of new O-RAN TS template.  Enclosure of CRs:  [NOK-2022.03.21-WG3-CR-0001-RICAPI-E2NodeState-v02.docx](https://oranalliance.atlassian.net/wiki/download/attachments/2210201622/NOK-2022.03.21-WG3-CR-0001-RICAPI-E2NodeState-v02.docx?api=v2) agreed at RICARCH #67.  [Lenovo.AO-2022.05.13-WG3-RICAPI-Enablement API procedure descriptions and definitions\_v04.docx](https://oranalliance.atlassian.net/wiki/download/attachments/2210201622/Lenovo.AO-2022.05.13-WG3-RICAPI-Enablement%20API%20procedure%20descriptions%20and%20definitions_v04.docx?api=v2),  [CICT-2022.04.22-WG3-CR-0009-Procedure Transaction Identity\_v3.docx](https://oranalliance.atlassian.net/wiki/download/attachments/2210201622/CICT-2022.04.22-WG3-CR-0009-Procedure%20Transaction%20Identity_v3.docx?api=v2) ,  [AsiaInfo.AO-2022.01.04-WG3-CR-0002-RICAPI-E2 Indication API Procedure descriptions and messages\_v04.docx](https://oranalliance.atlassian.net/wiki/download/attachments/2210201622/AsiaInfo.AO-2022.01.04-WG3-CR-0002-RICAPI-E2%20Indication%20API%20Procedure%20descriptions%20and%20messages_v04.docx?api=v2), and  [AsiaInfo.AO-2022.03.04-WG3-CR-0003-RICAPI-E2 Control API Procedure descriptions and messages\_v04.docx](https://oranalliance.atlassian.net/wiki/download/attachments/2210201622/AsiaInfo.AO-2022.03.04-WG3-CR-0003-RICAPI-E2%20Control%20API%20Procedure%20descriptions%20and%20messages_v04.docx?api=v2) agreed at RICARCH #70.  Minor editorial update. |
| 2021.10.12 | 00.01.00 | Enclosure of CR [VMW.AO-2021.07.14-WG3-CR-0002-SDLAPI-v04.docx](https://oranalliance.atlassian.net/wiki/download/attachments/1641513213/VMW.AO-2021.07.14-WG3-CR-0002-SDLAPI-v04.docx?api=v2) agreed at RICARCH #56. |
| 2021.05.18 | 00.00.03 | Approved TS skeleton. |

# History

|  |  |  |
| --- | --- | --- |
| Date | Revision | Description |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |