|  |
| --- |
| webwxgetmsgimg (7).jpegO-RAN.WG2.TS.R1TS-R004-v02.00 |

|  |
| --- |
| Technical Specification |
|  |

|  |
| --- |
| R1 interface: Test Specification |

|  |
| --- |
|  |
| Copyright © 2025 by the O-RAN ALLIANCE e.V.  The copying or incorporation into any other work of part or all of the material available in this specification in any form without the prior written permission of O-RAN ALLIANCE e.V. is prohibited, save that you may print or download extracts of the material of this specification for your personal use, or copy the material of this specification for the purpose of sending to individual third parties for their information provided that you acknowledge O-RAN ALLIANCE as the source of the material and that you inform the third party that these conditions apply to them and that they must comply with them.  O-RAN ALLIANCE e.V., Buschkauler Weg 27, 53347 Alfter, Germany  Register of Associations, Bonn VR 11238, VAT ID DE321720189 |

Contents

Foreword 4

Modal verbs terminology 4

Introduction 4

1 Scope 5

2 References 5

2.1 Normative references 5

2.2 Informative references 5

3 Definition of terms, symbols and abbreviations 6

3.1 Terms 6

3.2 Symbols 6

3.3 Abbreviations 6

4 Test methodology 7

4.1 General 7

4.2 Conformance testing of rApps 7

4.2.1 General 7

4.2.2 Test configuration 7

4.3 Conformance testing of SMO/Non-RT RIC framework 8

4.3.1 General 8

4.3.2 Test configuration 8

5 SME Service Registration service test cases 9

5.1 Conformance test cases for rApp 9

5.1.1 General 9

5.1.2 Register service API as API Consumer test scenario 9

5.2 Conformance test cases for SMO/Non-RT RIC framework 13

5.2.1 General 13

5.2.2 Register service API as API Producer test scenario 13

6. SME Service Discovery service test cases 18

6.1 Conformance test cases for rApp 18

6.1.1 General 18

6.1.2 Discover service API as API Consumer test scenario 18

Annex (informative): Change History 20

# Foreword

This Technical Specification (TS) has been produced by WG2 of the O-RAN Alliance. It is part of a TS-family covering the WG2: R1 Interface Specifications.

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

version xx.yy.zz

where:

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

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

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

# Modal verbs terminology

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

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

# Introduction

The purpose of the present document is to specify conformance and interoperability test cases for the R1 interface. Test cases and methodology for the R1 Service Consumer and R1 Service Producer, are separately specified.

The present document contains R1 test cases for conformance testing and interoperability testing. For each specified test case, cover the entrance criteria, procedure and expected result. The test cases are defined based on the procedures, interface and functional requirements as specified in the R1 specifications series for which the test cases are used to validate conformance and interoperability.

# 1 Scope

The present document specifies test cases for conformance testing and interoperability testing of the rApps and R1 services over R1 interface.

# 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] O-RAN TS: "R1 General Aspects and Principles" ("R1GAP").

[2] O-RAN TS: "R1 Use Cases and Requirements" ("R1UCR").

[3] O-RAN TS: "Transport Protocols for R1 services"("R1TP").

[4] O-RAN TS: "Application Protocols for R1 services"("R1AP").

[5] O-RAN TS: "Non-RT RIC: Architecture"("Non-RT RIC ARCH").

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

# 3 Definition of terms, symbols and abbreviations

## 3.1 Terms

For the purposes of the present document, the terms given in R1GAP [1], R1UCR [2], R1TP [3] apply.

## 3.2 Symbols

Void

## 3.3 Abbreviations

For the purposes of the present document, the abbreviations given in R1GAP [1], R1UCR [2], R1TP [3] apply.

# 4 Test methodology

## 4.1 General

This clause describes the methodology for conformance and interoperability testing of rApps and SMO/Non-Real Time RIC framework over R1 interface.

For conformance tests, simulators are used for testing R1 procedures. These simulators will have capability of generating HTTP requests and responses. There will be flexibility in configuring URI, headers and body for these HTTP requests and responses to enable creation of various test cases.

For interoperability tests, devices under tests are rApps and SMO/Non-RT RIC framework that are defined in the Non-RT RIC architecture specification [5], these devices are brought to operation by connecting to appropriate real or simulated devices.

## 4.2 Conformance testing of rApps

### 4.2.1 General

For conformance testing of rApps, rApp is the device under test and SMO/Non-RT RIC framework is the test simulator.

The present document specifies conformance tests for API Consumer and API Producer functionality as specified in R1AP [4].

### 4.2.2 Test configuration

#### 4.2.2.1 Overview

The test configuration for R1 conformance testing of rApp is illustrated in figure 4.2.2.1-1. For testing of rApp over R1 interface, the rApp is onboarded and instantiated in the cloud environment of the test simulator.



Figure 4.2.2.1-1: Illustration of R1 conformance testing of rApps.

#### 4.2.2.2 Device under test rApps

For enabling conformance testing, rApp has implemented API Consumer and/or API Producer functionality and the procedures as specified in R1AP [4] that are required to perform testing of the applicable R1 service test cases.

#### 4.2.2.3 Test simulator

For enabling conformance testing, the test simulator has implemented API Producer and/or API Consumer functionality, have HTTP Client and HTTP Server capabilities, and have flexibility to generate, receive, and validate HTTP messages for all the R1 procedures. The test simulator logs all message content during the testing.

As illustrated in figure 4.2.2.1-1, test simulator has all the capabilities required to execute the conformance testing of an rApp. For example, the test simulator has the capabilities to simulate the functionality of A1, O1, and O2 in a cloud environment.

## 4.3 Conformance testing of SMO/Non-RT RIC framework

### 4.3.1 General

For conformance testing of SMO/Non-RT RIC framework, SMO/Non-RT RIC framework is the device under test and rApp is the test simulator.

The present document specifies conformance tests for API Consumer and API Producer functionality as specified in R1AP [4].

### 4.3.2 Test configuration

#### 4.3.2.1 Overview

The test configuration for R1 conformance testing of SMO/Non-RT RIC framework is illustrated in figure 4.3.2.1-1.



Figure 4.3.2.1-1: Illustration of R1 conformance testing of Non-RT RIC framework.

#### 4.3.2.2 Device under test (SMO/Non-RT RIC Framework)

For enabling conformance testing, the SMO/Non-RT RIC framework has implemented API Producer and/or API Consumer functionality and the procedures specified in R1AP [4] that are required to perform testing of the applicable test cases.

#### 4.3.2.3 Test simulator

For enabling conformance testing, the test simulator rApp(s) has both API Producer and/or API Consumer and HTTP Server capabilities and have flexibility to generate, receive and validate HTTP messages for all the R1 procedures.

As illustrated in figure 4.3.2.1-1, test simulator has all the capabilities required to execute the conformance testing of an SMO/Non-RT RIC framework. For example, the test simulator has the capabilities to simulate the functionality of A1, O1, and O2.

# 5 SME Service Registration service test cases

## 5.1 Conformance test cases for rApp

### 5.1.1 General

#### 5.1.1.1 Device under test requirements

The rApp that acts as Device Under Test (DUT) in the test scenarios of this clause can be a function that is under development, or it can be a finalised commercial product. The requirements on the DUT for these tests are that it can handle the SME service registration service, and the purpose of the test scenarios is to validate that it confirms API Consumer functionality as specified in R1AP [4].

#### 5.1.1.2 Test simulator capabilities

The test simulator has the capabilities as required for a SMO/Non-RT RIC framework. In addition, it has the following capabilities:

* Recording of received HTTP requests and responses and analyzing them regarding conformance to the R1 service definitions.
* Controlled initiation of procedures with configurable URIs and payload formulated and modified.

Validating messages and issuing of verdicts related to the procedures in the test cases and thereby enabling determination of the DUT’s conformance API Producer functionality as specified in R1AP [4].

### 5.1.2 Register service API as API Consumer test scenario

#### 5.1.2.1 Register service (positive case)

##### 5.1.2.1.1 Test description and applicability

The purpose of this test case is to test the Register service API as specified in R1AP [4] clause 6.1.4.1. The expected outcome is successful validation of the request from the DUT.

##### 5.1.2.1.2 Test entrance criteria

The DUT has functionality to initiate the Register service procedure as defined in R1GAP [1].

NOTE: The DUT provides the service API description as defined in R1AP [4]. The service being registered can be a standard or non-standard R1 service.

##### 5.1.2.1.3 Test methodology

###### 5.1.2.1.3.1 Initial conditions

The test simulator as API Producer is ready and available to receive HTTP requests from the DUT.

###### 5.1.2.1.3.2 Procedure

Step 1. The DUT as an API Consumer initiates a HTTP request to register a service that it produces.

Step 2. At the test simulator the contents of the received HTTP request are recorded.

Step 3. The test simulator does the following validation:

a) The URI confirms to the format specified in R1AP [4] clause 6.1.5.2.

b) The HTTP request is a POST operation.

c) The HTTP request message content includes rApp identifier and the service API description conforms to the schema as specified in R1AP [4].

Step 4. The test simulator generates the service API identifier and constructs the URI for the created resource and sends the appropriate HTTP response as specified in R1AP [4] clause 6.1.4.1.1.

##### Expected result

The test is considered passed if Step 3 validation has passed.

#### 5.1.2.2 Update registered service (positive case)

##### 5.1.2.2.1 Test description and applicability

The purpose of this test case is to test the Update registered service API as specified in R1AP [4] clause 6.1.4.2. The expected outcome is successful validation of the request from the DUT.

##### 5.1.2.2.2 Test entrance criteria

1. The DUT has functionality to initiate the Update registered service procedure as defined in R1GAP [1].
2. A service API registration exists in test simulator and the serviceApiId and the apfId are known to DUT
3. The schema of the ServiceAPIDescription used for this test are available and used in DUT to formulate the Update registered service request, and in test simulator to validate the request.

NOTE: The DUT provides the ServiceApiDescription as defined in R1AP [4]. The service being updated can be a standard or non-standard R1 service.

##### 5.1.2.2.3 Test methodology

###### 5.1.2.2.3.1 Initial conditions

The test simulator as API Producer is ready and available to receive HTTP requests from the DUT.

###### 5.1.2.2.3.2 Procedure

Step 1. The DUT as an API Consumer initiates a HTTP request to update a registered service API identified by the apfId and the serviceApiId.

Step 2. At the test simulator the contents of the received HTTP request are recorded.

Step 3. The test simulator does the following validation:

a) The URI confirms to the format specified in R1AP [4] clause 6.1.5.3.

b) The HTTP request is a PUT operation.

c) The apfId and the serviceApiId in the URI match the service API being updated

d) The HTTP request message body contains the ServiceAPIDescription of the service API to be updated and conforms to the schema as specified in R1AP [4]

Step 4. The test simulator updates the resource, and a representation of the updated resource shall be returned in the response body the appropriate HTTP response as specified in R1AP [4] clause 6.1.5.3.1.1.

##### 5.1.2.2.4 Expected result

The test is considered passed if Step 3 validation has passed.

#### 5.1.2.3 Deregister service (positive case)

##### 5.1.2.3.1 Test description and applicability

The purpose of this test case is to test the Deregister service API as specified in R1AP [4] clause 6.1.4.3. The expected outcome is successful validation of the request from the DUT.

##### 5.1.2.3.2 Test entrance criteria

1. The DUT has functionality to initiate the Deregister service procedure as defined in R1GAP [1].
2. A service API registration exists in test simulator and the serviceApiId and the apfId are known to DUT.

##### 5.1.2.3.3 Test methodology

###### 5.1.2.3.3.1 Initial conditions

The test simulator as API Producer is ready and available to receive HTTP requests from the DUT.

###### 5.1.2.3.3.2 Procedure

Step 1. The DUT as an API Consumer initiates a HTTP request to deregister a service API identified by the apfID and the serviceApiId.

Step 2. At the test simulator the contents of the received HTTP request are recorded.

Step 3. The test simulator does the following validation:

a) The URI confirms to the format specified in R1AP [4] clause 6.1.5.3.

b) The HTTP request is a DELETE operation.

c) The serviceApiId and apfId in the URI match the service being deleted.

d) The message body is empty.

Step 4. The test simulator generated the appropriate HTTP response as specified in R1AP [4] clause 6.1.5.3.1.2.

##### 5.1.2.3.4 Expected result

The test is considered passed if Step 3 validation has passed.

#### 5.1.2.4 Query registered service (positive case)

##### 5.1.2.4.1 Test description and applicability

The purpose of this test case is to test the Query service APIs as specified in R1AP [4] clause 6.1.4.5. The expected outcome is successful validation of the request from the DUT.

##### 5.1.2.4.2 Test entrance criteria

1. The DUT has functionality to initiate Query registered service procedure.
2. A set of service API registrations exist in the test simulator and the apfId is known to DUT.

##### 5.1.2.4.3 Test methodology

###### 5.1.2.4.3.1 Initial conditions

The test simulator as API Producer is ready and available to receive HTTP requests from the DUT.

###### 5.1.2.4.3.2 Procedure

Step 1. The DUT as an API Consumer initiates a HTTP request to query service APIs that are registered by the apfId.

Step 2. At the test simulator the contents of the received HTTP request are recorded.

Step 3. The test simulator does the following validation:

a) The URI confirms to the format specified in R1AP [4] clause 6.1.5.3.

b) The HTTP request is a GET operation.

c) The apfId in the URI match the service APIs being queried.

d) The message body is empty.

Step 4. The test simulator generated the appropriate HTTP response as specified in R1AP [4] clause 6.1.5.2.1.2

##### 5.1.2.4.4 Expected result

The test is considered passed if Step 3 validation has passed.

#### 5.1.5.5 Partially updated registered service (positive case)

##### 5.1.5.5.1 Test description and applicability

The purpose of this test case is to test the partially update registered service API as specified in R1AP [4] clause 6.1.4.4. The expected outcome is successful validation of the request from the DUT.

##### 5.1.5.5.2 Test entrance criteria

1. The DUT has functionality to initiate the Partially update registered service procedure as defined in R1GAP [1].
2. A service API registration exists in test simulator and the serviceApiId and the apfId are known to DUT
3. The schema of the ServiceAPIDescriptionPatch used for this test are available and used in DUT to formulate the Partially update registered service request, and in test simulator to validate the request.

NOTE: The DUT provides the ServiceApiDescriptionPatch as defined in R1AP [4]. The service being updated can be a standard or non-standard R1 service.

##### 5.1.5.5.3 Test methodology

###### 5.1.5.5.3.1 Initial conditions

The test simulator as API Producer is ready and available to receive HTTP requests from the DUT.

###### 5.1.5.5.3.2 Procedure

Step 1. The DUT as an API Consumer initiates a HTTP request to partially update a registered service API identified by the apfId and the serviceApiId.

Step 2. At the test simulator the contents of the received HTTP request are recorded.

Step 3. The test simulator does the following validation:

a) The URI confirms to the format specified in R1AP [4] clause 6.1.5.3.

b) The HTTP request is a PATCH operation.

c) The apfId and the serviceApiId in the URI match the service API being updated

d) The HTTP request message body contains the ServiceAPIDescriptionPatch of the service API to be updated and conforms to the schema as specified in R1AP [4]

Step 4. The test simulator updates the resource, and a representation of the partially updated resource shall be returned in the response body the appropriate HTTP response as specified in R1AP [4] clause 6.1.5.3.1.3.

##### 5.1.5.5.4 Expected result

The test is considered passed if Step 3 validation has passed.

## 5.2 Conformance test cases for SMO/Non-RT RIC framework

### 5.2.1 General

#### 5.2.1.1 Device under test requirements

The SMO/Non-RT RIC framework that acts as Device Under Test (DUT) in the test scenarios of this clause can be a product that is under development, or it can be a finalised commercial product. The requirements on the DUT for these tests are that it can handle the SME service registration service, and the purpose of the test scenarios is to validate that it confirms API Producer functionality as specified in R1AP [4].

#### 5.2.1.2 Test simulator capabilities

The test simulator has the capabilities as specified in section 4.3.2. In addition, it has the following capabilities:

1. Recording of received HTTP requests and responses and analyzing them regarding conformance to the R1 service definitions.
2. Controlled initiation of procedures with configurable URIs and payload formulated and modified.

Validating messages and issuing of verdicts related to the procedures in the test cases and thereby enabling determination of the DUT’s conformance to API Producer functionality as specified in R1AP [4].

### Register service API as API Producer test scenario

#### 5.2.2.1 Register service (positive case)

##### 5.2.2.1.1 Test description and applicability

The purpose of this test case is to test the Register service API as specified in R1AP [4] clause 6.1.4.1. The expected outcome is successful validation of the response from the DUT.

##### 5.2.2.1.2 Test entrance criteria

The test simulator has functionality to initiate the Register service procedure as defined in R1GAP [1].

NOTE: The test simulator provides the service API description as defined in R1AP [4]. The service being registered can be a standard or non-standard R1 service.

##### 5.2.2.1.3 Test methodology

###### 5.2.2.1.3.1 Initial conditions

The test simulator as API Consumer is ready and available to receive HTTP responses from the DUT.

###### 5.2.2.1.3.2 Procedure

Step 1. The test simulator as an API Consumer initiates a HTTP request to register a service that it produces.

Step 2. The DUT receives the HTTP request.

Step 3. The DUT does the following validation:

1. The HTTP request is a POST request.
2. The URI conforms to the format specified in R1AP [4] clause 6.1.5.2.1.1.
3. The HTTP request message content includes the information as specified in R1AP [4]

Step 4. The DUT generates the service API identifier and constructs the URI for the created resource.

Step 5. The DUT sends the appropriate HTTP response as specified in R1AP [4] clause 6.1.4.1.1.

Step 6. At the test simulator the contents of the received HTTP response are recorded.

Step 7. The test simulator does the following validation:

a) The HTTP response message content includes the information as specified in R1AP [4], and the Location header will be present.

b) The URI confirms to the format specified in R1AP [4] clause 6.1.5.2.

##### 5.2.2.1.4 Expected result

The test is considered passed if Step 7 validation has passed.

#### 5.2.2.2 Update registered service (positive case)

##### 5.2.2.2.1 Test description and applicability

The purpose of this test case is to test the Update registered service API as specified in R1AP [4] clause 6.1.4.2. The expected outcome is successful validation of the response from the DUT.

##### 5.2.2.2.2 Test entrance criteria

1. The test simulator has functionality to initiate the Update registered service procedure as defined in R1GAP [1].
2. A service API registration exists in DUT and the serviceApiId and the apfId are known to test simulator
3. The schema of the ServiceAPIDescription used for this test are available and used in test simulator to formulate the Update registered service request, and in the test simulator to validate the request.

NOTE: The test simulator provides the ServiceApiDescription as defined in R1AP [4]. The service being updated can be a standard or non-standard R1 service.

##### Test methodology

###### 5.2.2.2.3.1 Initial conditions

The DUT as API Producer is ready and available to receive HTTP requests from the test simulator.

###### 5.2.2.2.3.2 Procedure

Step 1. The test simulator as an API Consumer initiates a HTTP request to update a registered service API identified by the apfId and the serviceApiId.

Step 2. The DUT receives the HTTP request.

Step 3. The DUT does the following validation:

a) The URI conforms to the format specified in R1AP [4] clause 6.1.5.3.

b) The HTTP request is a PUT operation.

c) The apfId and the serviceApiId in the URI match the service API being updated

d) The HTTP request message content includes the information as specified in R1AP [4]

Step 4. The DUT sends the appropriate HTTP response as specified in R1AP [4] clause 6.1.4.2.1.

Step 5. At the test simulator the contents of the received HTTP response are recorded.

Step 6. The test simulator does the following validation:

a) The HTTP response message content includes the information as specified in R1AP [4], and the Location header will be present.

##### 5.2.2.2.4 Expected result

The test is considered passed if Step 6 validation has passed.

#### 5.2.2.3 Deregister service (positive case)

##### 5.2.2.3.1 Test description and applicability

The purpose of this test case is to test the Deregister service API as specified in R1AP [4] clause 6.1.4.3. The expected outcome is successful validation of the response from the DUT.

##### 5.2.2.3.2 Test entrance criteria

1. The test simulator has functionality to initiate the Deregister service procedure as defined in R1GAP [1].
2. A service API registration exists in DUT and the serviceApiId and the apfId are known to test simulator.

##### 5.2.2.3.3 Test methodology

###### 5.2.2.3.3.1 Initial conditions

The test simulator as API Consumer is ready and available to receive HTTP requests from the test simulator.

###### 5.2.2.3.3.2 Procedure

Step 1. The test simulator as an API Consumer initiates a HTTP request to deregister a service API identified by the apfID and the serviceApiId.

Step 2. The DUT receives the HTTP request.

Step 3. The DUT does the following validation:

1. The HTTP request is a DELETE request.
2. The URI format as specified in R1AP [4] clause 6.1.5.3.1.2.
3. The HTTP request content includes the information as specified in R1AP [4].

Step 4. The DUT sends the appropriate HTTP response as specified in R1AP [4] clause 6.1.4.3.1.

Step 5. At the test simulator the contents of the received HTTP response are recorded.

Step 6. The test simulator does the following validation:

1. The URI confirms to the format specified in R1AP [4] clause 6.1.5.3.
2. The HTTP response message content includes the information as specified in R1AP [4].

c) The message body is empty.

##### 5.2.2.3.4 Expected result

The test is considered passed if Step 6 validation has passed.

#### 5.2.2.4 Query registered service (positive case)

##### 5.2.2.4.1 Test description and applicability

The purpose of this test case is to test the Query service APIs as specified in R1AP [4] clause 6.1.4.5. The expected outcome is successful validation of the response from the DUT.

##### 5.2.2.4.2 Test entrance criteria

1. The test simulator has functionality to initiate Query registered service procedure.
2. A set of service API registrations exist in the DUT and the apfId is known to test simulator.

##### 5.2.2.4.3 Test methodology

###### 5.2.2.4.3.1 Initial conditions

The test simulator as API Consumer is ready and available to receive HTTP responses from the DUT.

###### 5.2.2.4.3.2 Procedure

Step 1. The test simulator as an API Consumer initiates a HTTP request to query service APIs that are registered by the apfId.

Step 2. The DUT receives the HTTP request.

Step 3. The DUT does the following validation:

1. The HTTP request is a GET request.
2. The URI format as specified in R1AP [4] clause 6.1.5.2.1.2.
3. The HTTP request message content includes the information as specified in R1AP [4]

Step 4. The DUT initiates a HTTP Response.

Step 5. At the test simulator the contents of the received HTTP response are recorded.

Step 6. The test simulator does the following validation:

a) The URI conforms to the format specified in R1AP [4] clause 6.1.5.2.

b) The HTTP response message content includes the information as specified in R1AP [4].

##### 5.2.2.4.4 Expected result

The test is considered passed if Step 6 validation has passed.

#### 5.2.5.5 Partially updated registered service (positive case)

##### 5.2.5.5.1 Test description and applicability

The purpose of this test case is to test the partially update registered service API as specified in R1AP [4] clause 6.1.4.4. The expected outcome is successful validation of the response from the DUT.

##### 5.2.5.5.2 Test entrance criteria

1. The test simulator has functionality to initiate the Partially update registered service procedure as defined in R1GAP [1].
2. A service API registration exists in DUT and the serviceApiId and the apfId are known to test simulator
3. The schema of the ServiceAPIDescriptionPatch used for this test are available and used in test simulator to formulate the Partially update registered service request, and in test simulator to validate the request.

NOTE: The test simulator provides the ServiceApiDescriptionPatch as defined in R1AP [4]. The service being updated can be a standard or non-standard R1 service.

##### 5.2.5.5.3 Test methodology

###### 5.2.5.5.3.1 Initial conditions

The test simulator as API Consumer is ready and available to receive HTTP responses from the DUT.

###### 5.2.5.5.3.2 Procedure

Step 1. The test simulator as an API Consumer initiates a HTTP request to partially update a registered service API identified by the apfId and the serviceApiId.

Step 2. The DUT receives the HTTP request.

Step 3. The DUT does the following validation:

1. The HTTP request is a PATCH request.
2. The URI format as specified in R1AP [4] clause 6.1.5.3.1.3.
3. The HTTP request message content includes the information as specified in R1AP [4].

Step 4. The DUT initiates a HTTP Response.

Step 5. At the test simulator the contents of the received HTTP request are recorded.

Step 6. The test simulator does the following validation:

a) The HTTP response message content includes the information as specified in R1AP [4], and the Location header will be present.

##### 5.2.5.5.4 Expected result

The test is considered passed if Step 6 validation has passed.

# 6. SME Service Discovery service test cases

## 6.1 Conformance test cases for rApp

### 6.1.1 General

#### 6.1.1.1 Device under test requirements

The rApp that acts as Device Under Test (DUT) in the test scenarios of this clause can be a function that is under development, or it can be a finalised commercial product. The requirements on the DUT for these tests are that it can handle the SME service discovery service, and the purpose of the test scenarios is to validate that it conforms to the Service discovery API specified in R1AP [4].

#### 6.1.1.2 Test simulator capabilities

The test simulator has the capabilities as required for a SMO/Non-RT RIC framework. In addition, it has the following capabilities:

* Recording of received HTTP requests and responses and analyzing them regarding conformance to the R1 service definitions.
* Controlled initiation of procedures with configurable URIs and payload formulated and modified.

Validating messages and issuing of verdicts related to the procedures in the test cases and thereby enabling determination of the DUT’s conformance API Producer functionality as specified in R1AP [4].

### 6.1.2 Discover service API as API Consumer test scenario

#### 6.1.2.1 Query service (positive case)

##### 6.1.2.1.1 Test description and applicability

The purpose of this test case is to test the Query service APIs operation as specified in R1AP [4] clause 6.2.4.1. The expected outcome is successful validation of the request from the DUT.

##### 6.1.2.1.2 Test entrance criteria

1) The DUT has functionality to initiate the Discover services procedure.

2) A set of service API registrations exist in the test simulator.

##### 6.1.2.1.3 Test methodology

###### 6.1.2.1.3.1 Initial conditions

1) The test simulator as API Producer is ready and available to receive HTTP requests from the DUT.

###### 6.1.2.1.3.2 Procedure

Step 1. The DUT as an API Consumer initiates a HTTP request to query service APIs.

Step 2. At the test simulator the contents of the received HTTP request are recorded.

Step 3. The test simulator does the following validation:

a) The URI confirms to the format specified in R1AP [4] clause 6.2.5.2.

b) The HTTP request is a GET operation.

c) The message body is empty.

Step 4. The test simulator generated the appropriate HTTP response as specified in R1AP [4] clause 6.2.5.2.1.1

NOTE: Presence or validation of optional filter parameters is not used to determine validation on this test.

##### 6.1.2.1.4 Expected result

The test is considered passed if Step 3 validation has passed.

Annex (informative):   
Change History

|  |  |  |
| --- | --- | --- |
| Date | Revision | Description |
| 2024-11-21 | 02.00 | Published with Conformance test of Non-RT RIC as DUT for Service registration test case |
| 2024-07-11 | 01.00 | Published with conformance testing of rApps and SMO/Non-RT RIC Framework and with one Service registration test case |