Service Entitlement Configuration Test Cases

Version 1.0

15 December 2022

This is a Non-binding Permanent Reference Document of the GSMA

Security Classification: Non-confidential

Access to and distribution of this document is restricted to the persons permitted by the security classification. This document is subject to copyright protection. This document is to be used only for the purposes for which it has been supplied and information contained in it must not be disclosed or in any other way made available, in whole or in part, to persons other than those permitted under the security classification without the prior written approval of the Association.

Copyright Notice

Disclaimer

The GSM Association ("Association") makes no representation, warranty or undertaking (express or implied) with respect to and does not accept any responsibility for, and hereby disclaims liability for the accuracy or completeness or timeliness of the information contained in this document. The information contained in this document may be subject to change without prior notice.

Compliance Notice

The information contain herein is in full compliance with the GSM Association's antitrust compliance policy.

This Permanent Reference Document is classified by GSMA as an Industry Specification, as such it has been developed and is maintained by GSMA in accordance with the provisions set out in GSMA AA.35 - Procedures for Industry Specifications.

Table of Contents

1. Introduction

1.1 Overview

The aim of this GSMA standardization effort is to ensure that devices and device clients that follow the GSMA TS.43 specification behave in a similar way when communicating with the Entitlement Configuration Server (ECS) of an MNO.

Similarly it ensures that the MNO's ECS responds to entitlement requests from device clients in a consistent manner, according to the status of the service and the actions being requested.

This document provides test cases for the Service Entitlement Configuration protocol detailed in GSMA PRD TS.43 [1].

1.2 Scope

This document is intended for:

1. Parties which develop test tools and platforms
2. Test Labs / Test Houses which execute the testing
3. Entitlement Configuration Server vendors
4. Device OEMs / Manufacturers
5. Operators

The Test Book consists of a set of test cases relevant for testing a device or device client supporting Entitlement Configuration for one of the use cases covered in TS.43:

- VoLTE Entitlement
- VoWiFi Entitlement
- SMSoIP Entitlement
- Companion Devices On-Device Service Activation (ODSA)

The test cases specified within the Test Book are specified fully, step by step. For each test case specified or defined within this Test Book, there is a reference to one or more clauses or requirements from the GSMA PRD TS.43 [1].

1.3 Definition of Terms

Term	Definition
SIM	Subscriber Identity Module; a physical entity (UICC) that contains keys and ID required to authenticate and identify a user on a mobile network.
Device Client	Component/module on a device that provides a service to end-users. A device client verifies with the network’s Entitlement Configuration Server if it is entitled or not to offer that service to end-users.
Entitlement	The applicability, availability and status of a service on a device, needed by the device client before offering that service to end-users.

Entitlement Configuration	Information returned to the client by the network, providing entitlement information on a service.
Entitlement Configuration Server	The network element that provides entitlement configuration for different services to device clients.
eUICC	A removable or non-removable UICC which enables the remote and/or local management of Profiles in a secure way
Functional test case	Test cases designed to confirm compliance of main entitlement services from the user perspective to the requirements of TS.43. Functional test cases shall cover the services as a user experiences them, in realistic set up of real networks, and do not require deep inspection of traces.
Technical test case	A test case designed to verify technical implementation of devices/clients and networks and their compliance to technical requirements provided in TS.43. Technical test case require deep inspection of traces to confirm correct message syntax, formatting and sequencing
Simulator	A Network Simulator or a Test Network within a test lab.

1.4 Abbreviations

Abbreviation	Definition
AAA	Authentication, Authorization and Accounting server
CUT	Client Under Test
CDUT	Companion Device Under Test
EAP-AKA	Extensible Authentication Protocol for 3rd Generation Authentication and Key Agreement
ECS	Entitlement Configuration Server
EID	eUICC Identifier
eSIM	Embedded SIM
eUICC	Embedded Universal Integrated Circuit Card
FCM	Firebase Cloud Messaging
FFS	For Future Study
IMEI	International Mobile Equipment Identifier
HTTP	Hyper-Text Transfer Protocol
HTTPS	Hyper-Text Transfer Protocol Secure
ICCID	Integrated Circuit Card Identifier
IMEI	International Mobile Equipment Identity
IMS	IP Multimedia Subsystem
IMSI	International Mobile Subscriber Identity
JSON	JavaScript Object Notation
MSISDN	Mobile Subscriber Integrated Services Digital Network Number
ODSA	On-Device Service Activation
OIDC	OpenID Connect
OTP	One-Time Password
PRD	Permanent Reference Document
SIM	Subscriber Identity Module
SMS	Short Message Service
SMSoIP	SMS Over IP
SP	Service Provider
TAD	Technical Adaptation of Devices
TLS	Transport Layer Security
T&C	Terms & Conditions
UDH	User Data Header
UICC	Universal Integrated Circuit Card
URL	Uniform Resource Locator
VoWiFi	Voice-over-Wi-Fi

VoLTE	Voice-over-LTE
WNS	Windows Push Notification Service
XML	Extensible Markup Language

1.5 References

Ref	Document Number	Title
[1]	TS.43	GSMA PRD TS.43 - “Service Entitlement Configuration” http://www.gsma.com
[2]	RCC.14	GSMA PRD RCC.14 - “Service Provider Device Configuration” http://www.gsma.com
[3]	RFC 2119	“Key words for use in RFCs to Indicate Requirement Levels”, S. Bradner, March 1997. Available at http://www.ietf.org/rfc/rfc2119.txt
[4]	RFC 8174	Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words
[5]	RFC 4187	Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA)
[6]	RFC 7231	Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content

1.6 Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 (RFC2119) [3] (RFC8174) [4] when, and only when, they appear in all capitals, as shown here.

2 Test process

2.1 Test Environment

Two entities make up the Service Entitlement Configuration test environment:

- Client Under Test (CuT) – the device client that performs service entitlement requests towards the network for a particular service / use case
- Entitlement Configuration Server Under Test (ECSuT) – the MNO element supporting the TS.43 protocol [1], responding to entitlement requests from device clients

Test cases may be performed in a Laboratory using one or more ECS simulators or on Live Networks. In order to reach an ECS simulator, the device client shall be able to change the default ECS address configured on the client (as specified in 2.1 of TS.43 [1]) and points it to the simulator.

For some of the test cases it is necessary to log the signalling between the CuT and the ECSuT. If an ECS simulator is used, all the signalling between the CuT and the ECSuT shall be logged and shall be accessible to be used to derive the test case verdict.

If a Live Network is used, other options such as on-CUT logging or live network logging may be used. Such logging will be implementation dependent.

2.2 Pass Criteria

- A test is considered as successful only if the entire test procedure was carried out successfully and the expected results (functional test) or deep inspection (technical test) observed.
- A test is considered as failed if the tested feature shows unexpected behaviour.
- A test is considered as non-conclusive when the pass criteria cannot be evaluated due to issues during the setup of the initial conditions.

2.3 Future Study

Some of the test cases described in this Test Book are FFS (For Future Study). This means that some clarifications are expected at the requirement level to conclude on a test method, or that the test cases are low priority and have not yet been defined.

3 Test Cases

3.1 List of tests

The following table sum up the test cases

Ref	Title	page
[AUT-1]	check encoding of device GET (Accept_xxx) and ECS Response	
[AUT-2]	check percent URL encoding on all values from device GET query string	
[AUT-10]	Embedded EAP-AKA Authentication	11
[AUT-11]	Check EAP-Response EAP_AKA-Synchronization-Failure	14
[AUT-12]	EAP-AKA Fast Authentication for VoIMS procedure is successful	15
[AUT-13]	EAP-AKA Full Authentication procedure	16
[AUT-14]	EAP-AKA after SIM Swap with incompatible SIM	
[AUT-20]	OpenID Connect Authentication	

[AUT-30]	OpenID Connect Error cases	
[AUT-35]	OpenID OpenID Connect 511 response	
[AUT-36]	EAP-AKA 511 response	35
[CONF-1]	check VoIMS configuration document	
[CONF-10]	Validity period expiration	33
[ERR-01]	400 Bad Request	28
[ERR-05]	403 Forbidden	29
[ERR-07]	405 Method not Allowed	29
[ERR-10]	406 Not Acceptable	31
[ERR-15]	500 Internal Error return code	032
[ERR-20]	503 Retry-After return code	33
[ODSA-10]	ODSA on synchronous call flow	50
[ODSA-11]	ODSA on asynchronous call flow	52
[ODSA-12]	ODSA when a valid profile has been installed	53
[ODSA-20]	ODSA when eligibility rejected	54
[ODSA-30]	ODSA when no answer provided	55
[ODSA-90]	Callback management on Webviews	34
[ODSA-70]	Enable / disable profile on CDuT	56
[ODSA-75]	Mismatch profile detection	57
[ODSA-80]	Change companion device	58
[ODSA-81]	Change primary device	59
[PUSH-10]	Push notification to refresh configuration	66
[SMS-20]	SMSoIP Entitlement – status is INCOMPATIBLE	47

[SMS-21]	SMSoIP Entitlement – status is DISABLED	48
[SMS-22]	SMSoIP Entitlement – status is PROVISIONING	48
[SMS-23]	SMSoIP Entitlement – status is ENABLED	49
[VERS-01]	Vers parameter is aligned with the configuration document	21
[VERS-05]	Reset client	23
[VERS-06]	Reset client and stop configuration query	24
[VERS-07]	Reset client until user input and stop configuration query	26
[VLT-20]	VoLTE Entitlement – status is INCOMPATIBLE	43
[VLT-21]	VoLTE Entitlement – status is DISABLED	44
[VLT-22]	VoLTE Entitlement – status is PROVISIONING	44
[VLT-23]	VoLTE Entitlement – status is ENABLED	45
[VLT-30]	VoLTE Call	46
[WIF-10]	VoWiFi Entitlement Mode – Cannot be Offered	36
[WIF-11]	VoWiFi Entitlement Mode – Can Be Activated	37
[WIF-12]	VoWiFi Entitlement Mode – Service Data Missing	38
[WIF-13]	VoWiFi Entitlement Mode – Service Data Being Updated	39
[WIF-14]	VoWiFi Entitlement Mode – Service Being Provisioned	40
[WIF-50]	WiFi-calling indication & user actions	40

3.2 Most frequent initial conditions

This section gathers the mostly used initial condition. The description of test cases would reference this conditions, and focus on the variable conditions.

Ref	Entity	Initial conditions
[EAP_S1]	ECSuT	ECSuT ready to receive Entitlement Configuration requests from device clients. ECSuT can support EAP-AKA authentication and has access to the data and AKA procedure of a SIM on the device. VoWIFI & VoLTE services provisioned for the IMEI/IMSI used and active (version & version validity are greater or equal to 0)
[EAP_C1]	CuT	CuT can support EAP-AKA authentication with the SIM on device The SIM's subscription is known to the MNO. VoWifi and 4G or 5G network are available IMEI/IMSI eligible. SIM card is compatible with AKA mechanisms.
[EAP_S2]	ECSuT	ECSuT ready to receive Entitlement Configuration requests from device clients. ECSuT can support EAP-AKA authentication and has access to the data and AKA procedure of a SIM on the device.
[OIDC_S1]	ECSuT	ECSuT ready to receive Entitlement Configuration requests from device clients. ECSuT can support OIDC authentication and is configured to delegate authentication to the MNO's OpenID authentication server.
[OIDC_C1]	CuT	CuT is configured to perform OIDC authentication (properly handles 302 Found from ECSuT). The OIDC login / passwords are known by the tester Device is powered on. The CuT and user account are eligible to the SharedNumber service to be provisioned.
[NOES_CP1]	CDuT	No eSIM profile installed on companion device. The companion is paired with primary (CuT). The companion is eligible to the service to be provisioned.
[ES_CP1]	CDuT	eSIM profile is already installed on companion device. The companion is paired with primary (CuT).
ANY_S1	ECSuT	ECSuT ready to receive Entitlement Configuration requests from device clients. ECSuT can support EAP-AKA and OIDC authentications.
ANY_C1	CuT	CuT is configured to perform EAP-AKA or OIDC authentication. A Fast Authentication Token has already been delivered on the device. Device is powered on.

3.3 Technical test cases

This section includes generic tests which could be perform on most use-cases, such as protocol conformance. The test description can be done out of the table template, if more convenient. The description shall be kept short nevertheless.

3.3.1 [AUT-1] check encoding of device GET (Accept_xxx) and ECS Response (Content_Length+Type+Encoding)

Test Purpose

To ensure CuT use correct HTTP headers in AKA procedure.

Referenced Clauses

RCC14 [2] **HTTP Embedded EAP-AKA procedure**

Initial Conditions

See [EAP_S1]]/[EAP_C1].

Test execution

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester switch on device	CuT boot up and perform AKA procedure

Deep Inspection

check that all GET requests and ECSuT responses are compliant to the specifications.

The CuT shall send HTTP request to ESuT with at least the following headers:

Accept: application/vnd.gsma.eap-relay.v1.0+json, text/vnd.wap.connectivity-xml

Accept-encoding: gzip

and

Cookie: <cookies> whenever cookies have been set for the domain of the Entitlement Server FQDN

3.3.2 Authentication

3.3.2.1 [AUT-10] Embedded EAP-AKA Authentication

Test Purpose

To ensure a CuT with access to a SIM for authentication purposes can go through an Embedded EAP-AKA authentication exchange with the ECSuT.

Referenced Clauses

TS43 [1]: 2.8.1 Embedded EAP-AKA Authentication by Entitlement Configuration Server
RCC.14 [2]

Initial Conditions

Entity	Description of the initial conditions
ECSuT	See conditions [EAP_S1]] in section 3.2 Most frequent initial conditions
CuT	See conditions [EAP_C1] in section 3.2 Most frequent initial conditions Device is powered on. CuT does not have a token from a previous authentication exchange with the ECSuT.

Test execution

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester invokes CuT and triggers an entitlement configuration	CuT initiates entitlement configuration. Without an Auth Token from ECSuT (or an expired one), CuT get IMSI from SIM and invokes AKA function on SIM.
2	CuT → ECSuT	GET or POST request, parameters: • EAP_ID set to IMSI of SIM • No token • Other compulsory parameters shall be present: vers, app, terminal_id, entitlement_version	ECSuT detects EAP-AKA capability from client and initiates EAP procedure with MNO's authentication server. The authentication server provides EAP challenge back to the ECSuT. ECSuT creates HTTP response for client with EAP challenge.
3	ECSuT → CuT	HTTP 200 OK response with: • JSON object containing eap-relay-packet field with EAP Challenge in Base64 as per RCC.14 [2]	CuT processes EAP Challenge by sending it to SIM's AKA function. CuT obtains SIM's AKA response and creates another request to ECSuT
4	CuT → ECSuT	POST request, content: • JSON object containing eap-relay-packet field with EAP Response in Base64 as per RCC.14 [2]	ECSuT relays EAP response to MNO's authentication server. The authentication server informs ECSuT of successful authentication. ECSuT creates HTTP response for client with new Auth Token.
5	ECSuT → CuT	HTTP 200 OK response with: • TOKEN config parameter set to newly generated Auth Token. • Application data, if requested in initial request	CuT stores received Auth Token for any subsequent requests to the ECS.
6	Tester → CuT	Tester invokes CuT and triggers an entitlement configuration	

Deep Inspection

Check client headers:

Accept: application/vnd.gsma.eap-relay.v1.0+json, text/vnd.wap.connectivity-xml

3.3.2.2 [AUT-11] Check EAP-Response EAP_AKA-Synchronization-Failure

Test Purpose

To ensure CuT supports resynchronization exchanges.

Referenced Clauses

RFC 4187: `EAP-Response/AKA-Synchronization-Failure`.

Initial Conditions

Entity	Description of the initial conditions
ECSuT	See conditions [[EAP_S1]] in section 3.2 Most frequent initial conditions The SQN parameter used by the ECSuT to compute the challenge required by the EAP-AKA algorithm is different from the one expected by the SIM card of the CuT.
CuT	See conditions [[EAP_C1] in section 3.2 Most frequent initial conditions CuT is powered off. CuT does not have a token from a previous authentication exchange with the ECSuT, or the token is invalidated.

Test execution

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester switch on device	CuT boot up and attach to VoLTE network
2	Tester → CuT	Tester invokes CuT and triggers an entitlement configuration	CuT must issue a EAP_AKA-Synchronization-Failure EAP-Response
3	ECSuT → CuT	The SQN provided by the CuT is used in the new Challenge sent to the CuT	The CuT accept the new Challenge and send its own challenge.

3.3.2.3 [AUT-12] EAP-AKA Fast Authentication for VoIMS procedure is successful

Test Purpose

To ensure CuT/ECSuT perform fast authentication procedure

Referenced Clauses

TS43 [1]: 2.8.1 Embedded EAP-AKA Authentication by Entitlement Configuration Server
TS43 [1]: 2.9.6 Additional details on TOKEN

RCC.14 [2]

Initial Conditions

Entity	Description of the initial conditions
ECSuT	See conditions [[EAP_S1]] in section 3.2 Most frequent initial conditions
CuT	See conditions [[EAP_C1] in section 3.2 Most frequent initial conditions The device has received a valid token in a a previous authencation exchange and is powered off.The token is still valid before the test.

Test execution

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester switch on device	CuT boot up and attach to VoLTE network

	Tester → CuT	Tester invokes CuT and triggers an entitlement configuration	CuT start a token-based Fast Authentication without EAP-ID but token value obtained in last configuration XML document
			and request the Entitlement services configuration for:
			app=ap2003 VoLTE
			app=ap2004 VoWIFI
			app=ap2005 SMSoIP

3	ECSuT →CuT	if authentication token is correct, ECSuT returns services configuration XML document with: possibly new authentication TOKEN generated by ES with its validity in seconds Proper services entitlement status requested in GET message	

Deep inspection:

- Check configuration document. See Appendix A.1 for document example

3.3.2.4 [AUT-13] EAP-AKA Full Authentication procedure

Test Purpose

To ensure CuT performs full authentication procedure

Referenced Clauses

RFC4187 `EAP-AKA full authentication procedure`

RFC 3986 Section 2.4 Percent-Encoding.

Initial Conditions

Entity	Description of the initial conditions
ECSuT	See conditions [[EAP_S1]] in section 3.2 Most frequent initial conditions
CuT	See conditions [[EAP_C1] in section 3.2 Most frequent initial conditions

Test execution

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester invokes CuT and triggers an entitlement configuration	CuT initiates entitlement configuration. Without an Auth Token from ECSuT. CuT creates a request to ECSuT.
2	ECSuT →CuT	The server answers an AKA challenge	The CuT is able to decode the challenge, based on SIM algorithm
3	CuT →ECSuT	The client answer its own AKAchallenge	The ECSuT is able to decode the challenge, based on AAA algorithm
4	ECSuT → CuT	The server answers to the request at step 1	The ECSuT issues a new token in the configuration document

Deep inspection:

- In the complete AKA procedure, check that all GET requests and AES response are **compliant to the percent URL encoding on all values from device GET query string (RFC 3986 Section 2.4 Percent-Encoding)**
- **The EAP_ID param is present at step 1.**
- **Check cookie presence if ECSuT use them**

-

3.3.2.5 [AUT-14] EAP-AKA after SIM Swap with incompatible SIM

Test Purpose

To ensure a token is not reused after a SIM Swap

Referenced Clauses

RFC4187 `EAP-AKA full authentication procedure`

Initial Conditions

Entity	Description of the initial conditions
ECSuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions
CuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions 2 SIM are required: one eligible to Entitlement, and the other not.

Test execution

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester invokes CuT and triggers an entitlement configuration with	
2			
3			
4			

Deep inspection:

3.3.2.6 [AUT-20] OpenID Connect Authentication

Test Purpose

To ensure a CuT without access to a SIM can go through an OpenID authentication with the MNO at the request of the ECSuT.

Referenced Clauses

TS43_2.6.2

Initial Conditions

Entity	Description of the initial conditions
ECSuT	[OIDC_S1]]
CuT	[OIDC_C1]] CuT does not have a token from a previous authentication exchange with the ECSuT.

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester invokes CuT and triggers an entitlement configuration	CuT initiates entitlement configuration. Without an Auth Token from ECSuT (or an expired one), CuT creates a request to ECSuT.
2	CuT → ECSuT	GET or POST request, parameters: • No `EAP_ID` • No `token` • Other compulsory parameters shall be present: `vers`, `app`, `terminal_id`, `entitlement_version`	ECSuT detects no identity or authentication parameters and invokes OpenID flow.
3	ECSuT → CuT	HTTP 302 Found response with: • Address of MNO's OpenID server • All parameters needed for OpenID Connect authentication (response_type, client_id, scope, redirect_uri, state, etc)	CuT processes 302 Found (redirect) and generates another GET or POST request to OpenID server with parameters as received from ECSuT
4	CuT → OpenID	GET request to OpenID server: • All parameters needed for OpenID Connect authentication	OpenID server starts the authentication exchange with the Tester
5	OpenID → Tester	OpenID server authenticates the end-user and involves the proper authenticator	Authentication is successful on OpenID and authorization code is returned to the redirect_uri

3.3.2.7 Token-based Authentication

Test Purpose

To ensure …

Referenced requirements

TS43_2.1

Initial Conditions

Entity	Description of the initial conditions
ECS	ECS simulator or ECS ready to receive Entitlement Configuration requests from device clients. …
CUT	Device is powered on. Client went through successful authentication with ECS and received an Auth Token in previous exchange. …

Test execution:

Step	Direction	Sequence	Expected Result
1	User → CUT	User invokes VoWiFi CUT and triggers an entitlement configuration	CUT initiates entitlement configuration with an Auth Token from ECS and creates request to ECS.
2	CUT → ECS	GET or POST request, parameters: • `Token` • `App` set to ap2004 • Other compulsory parameters shall be present: `vers`, `terminal_id`, `entitlement_version`	ECS …
3	ECS → CUT	HTTP 200 OK response with: •	CUT …

Deep Inspection:

3.3.2.8 [AUT-20] OpenID Connect Authentication

Test Purpose

3.3.3 Vers parameter management

3.3.3.1 [VERS-01] Vers parameter is aligned with the configuration document

Test Purpose

To ensure the CuT take into account the vers param of the configuration document.

Referenced requirements

TS43 [1] : 2.3 HTTP GET Method

TS43 [1] : 2.4 HTTP POST Method

RCC14 [2] : 2.4 Client configuration

Initial Conditions

Entity	Description of the initial conditions
ECS	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions The validity param is provisioned to 600 and the vers to 1 in the configuration documents. The validity is provisioned to 0 at the last configuration document sent.
CuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions. The client has been reseted after a SIM change or factory reset of the device. No token is available.

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app	CuT performs a request towards ECSuT. vers parameter is set to 0 (no configuration document stored)
2	ECSuT → CuT	The ECSuT and CuT performs authentication mechanism (eg: EAP-AKA or OIDC)	The user is authenticated, a configuration document is sent back with a VERS characteristic (vers=1, validity=600)
3	Tester → CuT	The tester waits for validity expiracy (600s)	The CuT performs a new configuration request. The vers parameter of the request is aligned to the one received in the VERS characteristic previously received from the ECSuT (vers=1)

Step	Direction	Sequence	Expected Result
4	Tester → device	The tester changes the SIM	The CuT performs a new configuration request, with vers param set to 0 (no previous configuration for the new SIM)
5	CuT → ECSuT	A full authentication exchange is performed (eg: EAP-AKA, OIDC…)	A configuration document is sent by the ECSuT with a valid token (vers=1 validity=600).
6	notification platform → CuT →ECSuT	A notification is sent to the CuT	The CuT performs a new configuration request, using the previously received token.
7	ECSuT → Cut	The ECSuT checks the token and send back the configuration document if the token is still valid	vers=1, validity=0 in the VERS characteristic of the document
8	Tester → device	The device is rebooted	The Cut performs a new configuration request at reboot, using the previously received token. The vers param of the request is set to 1.
9	Tester → ECSuT	The tester waits more than 600s	The CuT does not send new configuration request, as the last validity received is set to 0 (no limitation).

3.3.3.2 [VERS-05] Reset client

Test Purpose

To ensure the CuT revert to its default behaviour when required by the server.

Referenced requirements

TS43 [1] : 2.3 HTTP GET Method

TS43 [1] : 2.4 HTTP POST Method

RCC14 [2] : 2.4.1 Configuration server response

Initial Conditions

Entity	Description of the initial conditions
ECS	The ECSuT is provisioned to return a VERS characteristic with vers=0 and validity=0 in the configuration documents returned.
CuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions The CuT has already received a configuration allowing VoWifi, VoLTE and SMSoIP services. A valid token has been stored. The token validaty is 600s.

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app	CuT performs a configuration request towards ECSuT.
2	ECSuT → CuT	The ECSuT returns a configuration document requesting the reset of the device	The vers and validity parameters are set to 0 in the configuration document. The CuT disable the previously provisioned services.
3	Tester → device	The device is rebooted	The CuT performs a new configuration request at start up, without token.
4	Tester → CuT	The tester waits for 600s	No configuration request is performed by the CuT.
5	Tester → CuT	Tester starts client app and set the services (VoLTE, VoWiFi, SMSoIP) to “on”	The CuT sends a configuration request to the ECSuT.
6	Tester → device	The tester changes the SIM	The CuT sends a configuration request to the ECSuT after SIM swap detection, without token.

Step	Direction	Sequence	Expected Result
7	notification platform → CuT →ECSuT	A notification is sent to the CuT	The CuT performs a new configuration request, without token.

3.3.3.3 [VERS-06] Reset client and stop configuration query

Test Purpose

To ensure the CuT revert to its default behaviour when required by the server, and does not use reboot configuration at start up, neither user triggering.

Referenced requirements

TS43 [1] : 2.3 HTTP GET Method

TS43 [1] : 2.4 HTTP POST Method

RCC14 [2] : 2.4.1 Configuration server response

Initial Conditions

Entity	Description of the initial conditions
ECS	The ECSuT is provisioned to return a VERS characteristic with vers=-1 and validity=-1 in the configuration documents
CuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions The CuT has already received a configuration allowing VoWifi, VoLTE and SMSoIP services. A valid token has been stored. The token validaty is 600s.

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app	CuT performs a configuration request towards ECSuT.
2	ECSuT → CuT	The ECSuT returns a configuration document (200 OK) requesting the reset of the device	The vers and validity parameters are set to -1 in the configuration document. The CuT disable the previously provisioned services.
3	Tester → CuT	The tester waits for 600s	No configuration request is performed by the CuT.
4	Tester → device	The device is rebooted	The CuT does not performs a new configuration request at start up.

Step	Direction	Sequence	Expected Result
5	Tester → CuT	Tester starts client app and set the services (VoLTE, VoWiFi, SMSoIP) to “on”	The CuT does not performs a new configuration request at user action
6	Tester → device	The tester changes the SIM in the device	The CuT performs a new configuration request at SIM swap detection
7	ECSuT → CuT	A full authentication exchange is performed (e.g.: OIDC, EAP_AKA…)	The ECSuT returns a configuration document, vers = -1 and validity = -1 in VERS characteristic.

3.3.3.4 [VERS-07] Reset client until user input and stop configuration query

Test Purpose

To ensure the CuT revert to its default behaviour when required by the server, and does not perform query at start up, but on user triggering action.

Referenced requirements

TS43 [1] : 2.3 HTTP GET Method

TS43 [1] : 2.4 HTTP POST Method

RCC14 [2] : 2.4.1 Configuration server response

Initial Conditions

Entity	Description of the initial conditions
ECS	The ECSuT is provisioned to return a VERS characteristic with vers=-2 and validity=-2 in the configuration documents.
CuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions The CuT has already received a configuration allowing VoWifi, VoLTE and SMSoIP services. A valid token has been stored. The validity of the token is 600s.

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app	CuT performs a configuration request towards ECSuT.
2	ECSuT → CuT	The ECSuT returns a configuration document (200 OK) requesting the reset of the device	The vers and validity parameters are set to -2 in the configuration document. The CuT disable the previously provisioned services.
3	Tester → device	The device is rebooted	The CuT does not performs a new configuration request at start up.
4	Tester → CuT	The tester waits for 600s	No configuration request is performed by the CuT.
5	Tester → CuT	Tester starts client app and set the services (VoLTE, VoWiFi, SMSoIP) to “on”	A configuration request is sent without token and a full authentication exchange (eg: EAP-AKA or OIDC) is performed.

Step	Direction	Sequence	Expected Result
6	ECSuT → CuT	A configuration document is sent.	The vers parameter is set to -2, the validity parameter is set to -2.
6	Tester → device	The tester changes the SIM	The CuT performs a new configuration request, without token.

3.3.4 Error code management

3.3.4.1 [ERR-01] 400 Bad Request

Test Purpose

To ensure the CuT handles "400 Bad Request " HTTP return code properly.

Referenced requirements

TS43 [1] : 2.10 HTTP Response Codes

Initial Conditions

Entity	Description of the initial conditions
ECS	The ECSuT is provisioned to return a 400 error code.
CuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app	CuT performs a request towards ECSuT.
2	ECSuT → CuT	The ECSuT returns HTTP 400 error code.	The CuT resend the request at reboot or if the CuT is started again.

3.3.4.2 [ERR-05] 403 Forbidden

Test Purpose

To ensure the CuT handles "403 Forbidden" HTTP return code properly.

Referenced requirements

TS43 [1] : 2.10 HTTP Response Codes

RCC14 [2] : 2.4.3 Response handling

Initial Conditions

Entity	Description of the initial conditions
ECS	The ECSuT is provisioned to return a 403 error code.
CuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app	CuT performs a request towards ECSuT.
2	ECSuT → CuT	The ECSuT returns HTTP 403 error code.	The CuT resend the request at reboot or if the CuT is started again. The client shall perform in maximum 5 consecutive unsuccessful configuration requests

3.3.4.3 [ERR-07] 405 Method not Allowed

Test Purpose

To ensure the CuT handles "405 Method not Allowed " HTTP return code properly.

Referenced requirements

TS43 [1] : 2.10 HTTP Response Codes

Initial Conditions

Entity	Description of the initial conditions
ECS	The ECSuT is provisioned to return a 405 error code.
CuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app	CuT performs a POST request towards ECSuT.
2	ECSuT → CuT	The ECSuT returns HTTP 405 error code.	The CuT resend the request as GET.

3.3.4.4 [ERR-10] 406 Not Acceptable

Test Purpose

To ensure the CuT handles 406 "Not Acceptable" HTTP return code properly.

Referenced requirements

TS43 [1] : 2.10 HTTP Response Codes

Initial Conditions

Entity	Description of the initial conditions
ECS	The ECSuT is provisioned to return a 406 error code.
CuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app	CuT performs a request towards ECSuT.
2	ECSuT → CuT	The ECSuT returns HTTP 406 error code.	The CuT apply the procedure defined by the Service Provider for the case of no configuration data is available (for example silent abort or error message).

3.3.4.5 [ERR-15] 500 Internal Error return code

Test Purpose

To ensure the CuT handles "500 Internal Error" HTTP return code properly.

Referenced requirements

TS43 [1] : 2.10 HTTP Response Codes

Initial Conditions

Entity	Description of the initial conditions
ECS	The ECSuT is provisioned to return a 500 error code.
CuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app	CuT performs a request towards ECSuT.
2	ECSuT → CuT	The ECSuT returns HTTP 500 error code.	The CuT resend the request at reboot or if the CuT is started again.

3.3.4.6 [ERR-20] 503 Retry-After return code

Test Purpose

To ensure the CuT handles "503 Retry-After" HTTP return code properly.

Referenced requirements

TS43 [1] : 2.10 HTTP Response Codes

RFC 7231 [6] : 7.1.3 Retry-After

Initial Conditions

Entity	Description of the initial conditions
ECS	The ECSuT is provisioned to return a 503 error code with a date or a delay in second in the HTTP Retry-After header.
CuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app	CuT performs a request towards ECSuT.
2	ECSuT → CuT	The ECSuT returns 503+Header “retry after/service unavailable”	The CuT resend the request after the time given in the header (either a date or a delay in second)

3.3.5 [CONF-10] Validity period expiration

Test Purpose

To ensure the CuT manages the validity period of the configuration document.

Referenced requirements

RCC14 [2]:4.2 Characteristics of the Service Provider Device Configuration

Initial Conditions

Entity	Description of the initial conditions
ECS	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions The ECS is provisioned to deliver a short time validity value in the VERS characteristic

CuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app	CuT performs a request towards ECSuT.
2	ECSuT → CuT	The ECSuT deliver a configuration document with a short time validity value (VERS characteristic)	The CuT starts a timer at the reception of the document
3	Tester →CuT	Tester wait for the validity period to expire	CuT shall invoke a new configuration request to check for configuration changes and to refresh the validity.

3.3.6 [ODSA-90] Callback management on Webviews

Test Purpose

To ensure the callbacks integration by the CuT on webviews.

3.4 Functional test cases

This section includes specific tests against a complete use case.

3.4.1 Authentication

3.4.1.1 [AUT-30] OpenID Connect Error cases

Test Purpose

To ensure the CUT handles authentication error cases properly.

3.4.1.2 [AUT-35] OpenID Connect 511 response

Test Purpose

To ensure the CUT handles reauthentication with OpenID

3.4.1.3 [AUT-36] EAP-AKA 511 response

Test Purpose

To ensure the CUT handles reauthentication with EAP-AKA

Referenced requirements

TS43 [1] 2.8.1 Embedded EAP-AKA Authentication by Entitlement Configuration Server

TS43 [1] 2.10 HTTP Response Codes

Initial Conditions

Entity	Description of the initial conditions
ECS	See conditions [OIDC_S1] in section 3.2 Most frequent initial conditions
CuT	See conditions [EAP_C1 in section 3.2 Most frequent initial conditions The CuT has already a token from previous EAP_AKA authentication. The token is no more valid.

Test execution

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app	CuT performs a request towards ECSuT with previous token.
2	ECSuT → CuT	The ECSuT checks the token validity. As it is no more valid, a 511 HTTP code is returned.	The CuT delete the invalidated token.
3	CuT → ECSuT	The CuT retries the request without token.	A full authentication AKA procedure is performed.

Deep Inspection

- The first request emitted by the CuT reuse the token parameter
- The second request performs a full EAP AKA authentication, without invalidated token

3.4.2 VoWiFi Entitlement Test Cases / EAP AKA authentication

3.4.2.1 [WIF-10] VoWiFi Entitlement Mode – Cannot be Offered

Test Purpose

To ensure the CUT handles "Cannot be Offered" entitlement mode properly.

Referenced requirements

TS43 [1] : 3.3.1 VoWiFi Entitlement Mode - Cannot be offered

Initial Conditions

Entity	Description of the initial conditions
ECS	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions The VoIMS services are not provisioned for the given SIM card. No token has been retrieved in a previous EAP_AKA authentication.
CuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app	CuT performs a request towards ECSuT. A full EAP AKA authentication is performed.
2	ECSuT → CuT	After full authentication, the configuration document provided does not contain a token but **EntitlementStatus=INCOMPATIBLE (2) for VoWifi.**	the device shall either display **MessageForIncompatible** when it is not void, or the default device error message (if any).

3.4.2.2 [WIF-11] VoWiFi Entitlement Mode – Can Be Activated

Test Purpose

To ensure the CUT handles "Can Be Activated" entitlement mode properly.

Referenced requirements

TS43 [1] **:** 3.3.2 VoWiFi Entitlement Mode - Can be activated

Initial Conditions

Entity	Description of the initial conditions
ECS	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions The provisioning data regarding the SIM are: • **EntitlementStatus** is ENABLED (1) • **ProvStatus** is PROVISIONED (1) or NOT REQUIRED (2) • **TC_status** and **AddrStatus** are AVAILABLE (1) or NOT REQUIRED (2)
CuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions VoWifi service setting on the device is set to on by the user.

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app	CuT performs a configuration request towards ECSuT.
2	ECSuT → CuT	The configuration document is transmitted.	VoWiFi service should be activated on the device

3.4.2.3 [WIF-12] VoWiFi Entitlement Mode – Service Data Missing

Test Purpose

To ensure the CUT handles " Service Data Missing " entitlement mode properly.

Referenced requirements

TS43 [1] **:** 3.3.3 VoWiFi Entitlement Mode - Service Data Missing

Initial Conditions

Entity	Description of the initial conditions
ECS	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions The provisioning data regarding the SIM are: • **EntitlementStatus** is DISABLED (0) • **ProvStatus** is any values • Either **TC_status** or **AddrStatus** is NOT AVAILABLE (0)
CuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions VoWifi service setting on the device is set to on by the user.

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app	CuT performs a configuration request towards ECSuT.
2	ECSuT → CuT	The configuration document is transmitted.	VoWiFi service should not be activated on the device. The CuTshall open a web view and instruct the end-user to enter the required missing VoWiFi service information (T&C or static physical address).

3.4.2.4 [WIF-13] VoWiFi Entitlement Mode – Service Data Being Updated

Test Purpose

To ensure the CUT handles " Service Data Being Updated " entitlement mode properly.

Referenced requirements

TS43 [1] **:** 3.3.4 VoWiFi Entitlement Mode - Service Data Being Updated

Initial Conditions

Entity	Description of the initial conditions
ECS	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions The provisioning data regarding the SIM are: • **EntitlementStatus** is DISABLED (0) • **ProvStatus** is any values • Either **TC_status,** or **AddrStatus** is set to IN PROGRESS (3)
CuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions VoWifi service setting on the device is set to on by the user.

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app	CuT performs a configuration request towards ECSuT.
2	ECSuT → CuT	The configuration document is transmitted.	VoWiFi service should not be activated on the device.

3.4.2.5 [WIF-14] VoWiFi Entitlement Mode – Service Being Provisioned

Test Purpose

To ensure the CUT handles " Service Being Provisioned " entitlement mode properly.

Referenced requirements

TS43 [1] **:** 3.3.5 VoWiFi Entitlement Mode - Service Being Provisioned

Initial Conditions

Entity	Description of the initial conditions
ECS	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions • The provisioning data regarding the SIM are:**EntitlementStatus** is DISABLED (0) • **TC_status** and **AddrStatus** are set to AVAILABLE (1) or NOT REQUIRED (2) • **ProvStatus** is set to NOT PROVISIONED (0) or IN PROGRESS (3) Or • **EntitlementStatus** is PROVISIONING (3) • **ProvStatus,TC_status** and **AddrStatus** are set to any values
CuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions VoWifi service setting on the device is set to on by the user.

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app	CuT performs a configuration request towards ECSuT.
2	ECSuT → CuT	The configuration document is transmitted.	VoWiFi service should not be activated on the device. After an end-user action (going into VoWiFi service settings for example), the CuTshall show that the service is pending or being provisioned.

3.4.2.6 [WIF-50] WiFi-calling indication & user actions

Test Purpose

To ensure CuT displays an indication about Wifi-calling. The way the information is provided (UI) is device dependant.

Referenced Clauses

TS43 [1]: 3 VoWiFi Entitlement Configuration

Initial Conditions

Entity	Description of the initial conditions
ECSuT	See conditions [[EAP_S1]] in section 3.2 Most frequent initial conditions
CuT	See conditions [[EAP_C1] in section 3.2 Most frequent initial conditions CuT is powered off. The device is switch on for the 1st time (factory reset procedure can be done before)

Test execution

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester invokes CuT and triggers an entitlement configuration	CuT initiates entitlement configuration
2	ECSuT → CuT	After AKA authentication procedure, a configuration document is sent to the CuT.	The tester can see on the device the WiFi calling feature is available. He can activate/deactivate the feature.

3.4.2.7 [WIF-60] WiFi-call

Test Purpose

To ensure the CuT performs WiFi calls when VoWifi is activated. The way the user is aware about WiFi call is device dependant (UI).

Referenced requirements

TS43 [1]: 3.3 Entitlement Modes of VoWiFi Client

Initial Conditions

Entity	Description of the initial conditions
ECS	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions
CuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions

Test execution

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app and enable WiFi calling	WiFi calling indication is enabled
2	Tester → device	Test perform an originating and terminating WiFi call	The calls are successful. The WiFi calling indication remains visible during the call

3.4.3 VoLTE Entitlement Test Cases

3.4.3.1 [VLT-20] VoLTE Entitlement – status is INCOMPATIBLE

Test Purpose

To ensure the CUT handles " INCOMPATIBLE " entitlement status properly.

Referenced requirements

TS43 [1] : 4.2 Client Behaviour to VoLTE Entitlement Configuration

Initial Conditions

Entity	Description of the initial conditions
ECS	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions The VoLTE service status is set to INCOMPATIBLE (2) for the CuT MessageForIncompatible is provisioned
CuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app	CuT performs a request towards ECSuT.
2	ECSuT → CuT	The configuration document provided does not contain a token but **EntitlementStatus=INCOMPATIBLE(2) for VoLTE.**	VoLTE service is not activated. The device shall either display **MessageForIncompatible** parameter when it is not void, or the default device error message (if any).

3.4.3.2 [VLT-21] VoLTE Entitlement – status is DISABLED

Test Purpose

To ensure the CUT handles " DISABLED " entitlement status properly.

Referenced requirements

TS43 [1] : 4.2 Client Behaviour to VoLTE Entitlement Configuration

Initial Conditions

Entity	Description of the initial conditions
ECS	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions The VoLTE service status is set to DISABLED (0)
CuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app	CuT performs a request towards ECSuT.
2	ECSuT → CuT	The configuration document provided contains **EntitlementStatus=DISABLED for VoLTE.**	VoLTE service is not activated

3.4.3.3 [VLT-22] VoLTE Entitlement – status is PROVISIONING

Test Purpose

To ensure the CUT handles " PROVISIONING " entitlement status properly.

Referenced requirements

TS43 [1] : 4.2 Client Behaviour to VoLTE Entitlement Configuration

Initial Conditions

Entity	Description of the initial conditions
ECS	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions The VoLTE entitlement status is set to PROVISIONING (3)
CuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app	CuT performs a request towards ECSuT.
2	ECSuT → CuT	The configuration document contains **EntitlementStatus=** PROVISIONING(3) **for VoLTE.**	VoLTE service is not activated. After an end-user action (going into VoLTE service settings for example), the client shall show that the service is pending or being provisioned.

3.4.3.4 [VLT-23] VoLTE Entitlement – status is ENABLED

Test Purpose

To ensure the CUT handles " ENABLED " entitlement status properly.

Referenced requirements

TS43 [1] : 4.2 Client Behaviour to VoLTE Entitlement Configuration

Initial Conditions

Entity	Description of the initial conditions
ECS	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions The VoLTE entitlement status is set to ENABLED (1)
CuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions The VoLTE's service setting on the device is equivalent to ON

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app	CuT performs a request towards ECSuT.
2	ECSuT → CuT	The configuration document contains **EntitlementStatus=** ENABLED **(1) for VoLTE.**	VoLTE service is activated.

3.4.3.5 [VLT-30] VoLTE Call

Test Purpose

To ensure the CuT allows 4G calls when VoWifi is deactivated

Referenced requirements

TS43 [1] Client Behaviour to VoLTE Entitlement Configuration

Initial Conditions

Entity	Description of the initial conditions
ECS	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions
CuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions

Test execution

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app and disable WiFi calling	WiFi calling icon should be disabled
2	Tester → device	Test perform an originating and terminating VoLTE call	The calls are successful. 4G icon is visible during the call

3.4.4 SMSoIP Entitlement Test Cases

3.4.4.1 [SMS-20] SMSoIP Entitlement – status is INCOMPATIBLE

Test Purpose

To ensure the CUT handles " INCOMPATIBLE " entitlement status properly.

Referenced requirements

TS43 [1] : 5.2 Client Behaviour to SMSoIP Entitlement Configuration

Initial Conditions

Entity	Description of the initial conditions
ECS	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions The SMSoIP service status is set to INCOMPATIBLE
CuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app	CuT performs a request towards ECSuT.
2	ECSuT → CuT	The configuration document provided does not contain a token but **EntitlementStatus=INCOMPATIBLE for** SMSoIP**.**	SMSoIP service is not activated.

3.4.4.2 [SMS-21] SMSoIP Entitlement – status is DISABLED

Test Purpose

To ensure the CUT handles " DISABLED " entitlement status properly.

Referenced requirements

TS43 [1] : 5.2 Client Behaviour to SMSoIP Entitlement Configuration**Initial Conditions**

Entity	Description of the initial conditions
ECS	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions The SMSoIP service status is set to DISABLED
CuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app	CuT performs a request towards ECSuT.
2	ECSuT → CuT	The configuration document provided contains **EntitlementStatus=DISABLED for** SMSoIP**.**	SMSoIP service is not activated

3.4.4.3 [SMS-22] SMSoIP Entitlement – status is PROVISIONING

Test Purpose

To ensure the CUT handles " PROVISIONING " entitlement status properly.

Referenced requirements

TS43 [1] : 5.2 Client Behaviour to SMSoIP Entitlement Configuration**Initial Conditions**

Entity	Description of the initial conditions
ECS	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions The SMSoIP service status is set to PROVISIONING
CuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app	CuT performs a request towards ECSuT.

Step	Direction	Sequence	Expected Result
2	ECSuT → CuT	The configuration document contains **EntitlementStatus=** PROVISIONING **for** SMSoIP**.**	SMSoIP service is not activated. After an end-user action (going into SMSoIP's service settings for example), the client shall show that the service is pending or being provisioned.

3.4.4.4 [SMS-23] SMSoIP Entitlement – status is ENABLED

Test Purpose

To ensure the CUT handles " ENABLED " entitlement status properly.

Referenced requirements

TS43 [1] : 5.2 Client Behaviour to SMSoIP Entitlement Configuration

Initial Conditions

Entity	Description of the initial conditions
ECS	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions The SMSoIP service status is set to PROVISIONING
CuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions The SMSoIP's service setting on the device is equivalent to ON

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app	CuT performs a request towards ECSuT.
2	ECSuT → CuT	The configuration document contains **EntitlementStatus=** ENABLED **for** SMSoIP**.**	SMSoIP service is activated.

3.4.5 Companion ODSA Entitlement Test Cases/OIDC authentication

3.4.5.1 [ODSA-10] ODSA on synchronous call flow

Test Purpose

To ensure CuT installs an eSIM profile on companion when no notification is required

Referenced requirements

TS43 – 6 "On-Device Service Activation (ODSA) Entitlement and Configuration"

TS43 – 7.2 "ODSA Portal with Immediate Download Info – Final Steps"

Initial Conditions

Entity	Description of the initial conditions
ECS	See conditions [OIDC_S1] in section 3.2 Most frequent initial conditions
CuT	See conditions [OIDC_C1] in section 3.2 Most frequent initial conditions. No previous authentication has been performed.
CDuT	See conditions [NOES_CP1] in section 3.2 Most frequent initial conditions

Test execution

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app	The app displays a page inviting the user to subscribe an offer to connect the companion device to his/her MNO
2	CuT → ECSuT	A TS43 request is issued without authentication token	The ECSuT issue HTTP 302 redirect toward MNO's OpenID authentication server; the user is invited to authenticate
3	Tester → CuT	Tester goes through several websheet to authenticate with his MSISDN and password	A token is returned by the ECSuT to identify the user
4	ECSuT → CuT	the CuT is provided with information to download an eSIM profile	A page allowing the user to activate the MNO mobile service on the companion is displayed
5	Tester → CuT	Tester activate the service	Some notifications are displayed on CuT / CDuT while the eSIM profile is downloading.
6	Tester → CuT	Tester finalize the set up and close the client app.	The eSIM profile is downloaded and an ICCID is visible on CDuT and/or CuT menus.

Step	Direction	Sequence	Expected Result
7	Tester → CuT	Tester starts client to display the activation state of the service	A page displaying an “activated” profile status is displayed
8	Tester → CuT	Tester disable Bluetooth	The CDuT is isolated from CuT
9	Tester → CDuT	Tester performs outgoing & incoming cellular voice calls	Voice calls are performed without the primary device being involved.

3.4.5.2 [ODSA-11] ODSA on asynchronous call flow

Test Purpose

To ensure CuT installs a profile on companion eSIM after a notification mechanism. This test requires the ECS to delay the activation state.

Referenced requirements

TS43 – 6 “On-Device Service Activation (ODSA) Entitlement and Configuration”

TS43 – 7.3 “ODSA Portal with Delayed Download Info – Final Steps”

Initial Conditions

Entity	Description of the initial conditions
ECS	See conditions [OIDC_S1] in section 3.2 Most frequent initial conditions The ECS delays the activation state.
CuT	See conditions [OIDC_C1] in section 3.2 Most frequent initial conditions. A previous authentication has been performed.
CDuT	See conditions [NOES_CP1] in section 3.2 Most frequent initial conditions

Test execution

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app	The app displays a page inviting the user to subscribe an offer to connect the companion device to his/her MNO
2	CuT → ECSuT	A TS43 request is issued with authentication token	No authentication phase is required
4	ECSuT → CuT	the CuT is provided with information to download an eSIM profile	A page allowing the user to activate the MNO mobile service on the companion is displayed
5	Tester → CuT	Tester activate the service	Some notifications are displayed on CuT / CDuT while the eSIM profile is downloading.
6	Tester → CuT	Tester finalize the set up and close the client app.	The eSIM profile is downloaded and an ICCID is visible on CDuT and/or CuT menus.
7	Tester → CuT	Tester starts client to display the activation state of the service	A page displaying an “activating” profile status is displayed.
8	ECSuT → CuT	A notification is issued when the profile state changes from ACTIVATING to ACTIVATED	
9	CuT → ECSuT	CuT requests to update the eSIM profile	A notification warns the user that he can use the service
10	Tester → CuT	Tester disable Bluetooth	The CDuT is isolated from CuT

Step	Direction	Sequence	Expected Result
11	Tester → CDuT	Tester performs outgoing & incoming cellular voice calls	Voice calls are performed without the primary device being involved.

Deep Inspection:

The profile downloaded by the CDuT has the status “ACTIVATING” and not “ACTIVATED”.

3.4.5.3 [ODSA-12] ODSA when a valid profile has been installed

Test Purpose

To ensure CuT does not propose a new subscription after a companion factory reset for the same user.

3.4.5.4 [ODSA-20] ODSA when eligibility rejected

Test Purpose

To ensure CuT handles CheckEligibility operation rejected answer from the ECSuT

Referenced requirements

TS43 – 6.5.2 “CheckEligibility Operation Configuration Parameters”

Initial Conditions

Entity	Description of the initial conditions
ECS	See conditions [OIDC_S1] in section 3.2 Most frequent initial conditions
CuT	CuT is configured to perform OIDC authentication (properly handles 302 Found from ECSuT). The OIDC login / passwords are known by the tester Device is powered on. The CuT or user account are NOT eligible to the service to be provisioned.
CDuT	See conditions [NOES_CP1] in section 3.2 Most frequent initial conditions

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester launch CuT app to start ODSA journey	The CuT does not allow to download an eSIM profile on CDuT. The reason should be displayed.

Deep Inspection:
Several mechanism can be used to display the reject. E.g.: use the “notEnabledURL” parameter, or an inner redirect to an ECS page. Checks the mechanism according to ECSuT used.

3.4.5.5 [ODSA-30] ODSA when no answer provided (or 4xx response code)

Test Purpose

To ensure CuT handles a error answers from ECSuT

Referenced requirements

TS43[1] – 2.10 “HTTP Response Codes”

Initial Conditions

Entity	Description of the initial conditions
ECS	See conditions [OIDC_S1] in section 3.2 Most frequent initial conditions. The test requires the ability to provision specific error codes and no answer from ECS.
CuT	See conditions [OIDC_C1] in section 3.2 Most frequent initial conditions. No previous authentication has been performed.
CDuT	See conditions [NOES_CP1] in section 3.2 Most frequent initial conditions

Test execution

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app	The app displays a page inviting the user to subscribe an offer to connect the companion device to his/her MNO
2	CuT → ECSuT	A TS43 request is issued without authentication token	
3	ECSuT → CuT	The ECS returns an error code 4xx or 5xx, or does not answer	According to the error code, the CuT displays a dedicated or a generic error message.

Deep inspection:

- After a successful authentication, a 511 return should lead next use of the CuT to perform requests without any token.

3.4.5.6 [ODSA-70] Enable / disable profile on CDuT

Test Purpose

To ensure the change of service status performed from the CuT is acknowledged by the CDuT.

Referenced requirements

TS43[1] – 6 “On-Device Service Activation (ODSA) Entitlement and Configuration”

TS43[1] – 6.5.5 “Acquire Configuration Operation Configuration Parameters”

Initial Conditions

Entity	Description of the initial conditions
ECS	See conditions [OIDC_S1] in section 3.2 Most frequent initial conditions
CuT	See conditions [OIDC_C1] in section 3.2 Most frequent initial conditions. A previous authentication has been performed.
CDuT	See conditions [ES_CP1] in section 3.2 Most frequent initial conditions. An active eSIM profile has been downloaded on CDuT.

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT, CDuT	Tester open on CuT app a page that should allow to activate/deactivate the selected profile, and perform a deactivation action, then exit from the app.	The CDuT should display a notification indicating a service deactivation.
2	Tester → CuT	Tester open on CuT app the page that display the service status.	The page should indicate the status deactivated
3	Tester → CuT, CDuT	Tester open the page and perform an activation action, then exit from the app.	The CDuT should display a notification indicating a service activation.
4	Tester → CuT	Tester open on CuT app the page that display the service status.	The page should indicate the status activated.

Deep inspection:
Check the service status changes are reflected through the result of an AcquireConfiguration request.

3.4.5.7 [ODSA-75] Mismatch profile detection

Test Purpose

To ensure a mismatch profile is detected by the CuT after a SIM SWAP for eligible users.

Referenced requirements

TS43[1] – 6 “On-Device Service Activation (ODSA) Entitlement and Configuration”

TS43[1] – 6.5.5 “Acquire Configuration Operation Configuration Parameters”

Initial Conditions

Entity	Description of the initial conditions
ECS	See conditions [OIDC_S1] in section 3.2 Most frequent initial conditions
CuT	See conditions [OIDC_C1] in section 3.2 Most frequent initial conditions. A previous authentication has been performed. 2 eligible SIM are required for the test.
CDuT	See conditions [ES_CP1] in section 3.2 Most frequent initial conditions. An eSIM profile has been downloaded on CDuT with SIM#1 in primary device.

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT, CDuT	Tester disable Bluetooth on CuT and CDuT, and perform an incoming call to SIM#1	Both devices (primary and companon) rings.
2	Tester → CuT	Tester switch off the smartphone and insert new SIM (SIM#2, also eligible to SharedNumber service) inside it. Switch on the smartphone. Tester Launch CuT app and use menu to manage CDuT service subscription	A mismatch profile is detected. The user is allowed only to delete the current profile and download a new one.
3	Tester → CuT	Tester delete current profile.	The CDuT has no profile
4	Tester → CuT	Tester downloads a new profile (a new authentication is performed)	The new profile is downloaded to CDuT
5	Tester → CuT, CDuT	Tester disable Bluetooth on CuT and CDuT, and perform an incoming call to SIM#2	Both devices (primary and companion) rings.

3.4.5.8 [ODSA-80] Change companion device

Test Purpose

To ensure an existing service subscription is reused when a user changes a companion for a given primary device.

Referenced requirements

TS43[1] – 6 “On-Device Service Activation (ODSA) Entitlement and Configuration”

TS43[1] – 6.5.3 “ManageSubscription Operation Configuration Parameters”

Initial Conditions

Entity	Description of the initial conditions
ECS	See conditions [OIDC_S1] in section 3.2 Most frequent initial conditions
CuT	See conditions [OIDC_C1] in section 3.2 Most frequent initial conditions. A previous authentication has been performed.
CDuT	See conditions [ES_CP1] in section 3.2 Most frequent initial conditions. 2 CDuT are needed for this test. An eSIM profile has been downloaded and is active on CDuT#1.

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT, CDuT#1	Tester delete existing eSIM profile on CDuT#1.	The subscription to the service SharedNumber is not cancelled.
2	Tester → CuT, CDuT#2	Tester pairs the CDuT#2 with the primary device, and launch the CuT app.	The user is proposed to download a new profile (he is not proposed to subscribe to SharedNumber service a 2nd time).
3	ECSuT → CuT,CDuT #2	Once the downloading is accepted by the tester, the ECSuT returns download information to the CuT.	A profile is downloaded to the CDuT#2
4	Tester → CuT, CDuT#2	Tester disable Bluetooth on CDuT#2 and primary device. Tester performs incoming call.	Both CDuT#2 and primary rings.

3.4.5.9 [ODSA-81] Change primary device

Test Purpose

To ensure a existing service subscription is reused when a user changes a primary device for a given companion.

Referenced requirements

TS43[1] – 6 “On-Device Service Activation (ODSA) Entitlement and Configuration”

TS43[1] – 6.5.3 “ManageSubscription Operation Configuration Parameters”

Initial Conditions

Entity	Description of the initial conditions
ECS	See conditions [OIDC_S1] in section 3.2 Most frequent initial conditions
CuT	See conditions [OIDC_C1] in section 3.2 Most frequent initial conditions. 2 primary devices identified by CuT#1 & CuT#2 are needed for this test. They are both eligible to SharedNumber service.
CDuT	See conditions [ES_CP1] in section 3.2 Most frequent initial conditions. An eSIM profile has been downloaded with CuT#1 and is active on CDuT.

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT#1, CDuT	Tester unpairs the CDuT from CuT#1	The subscription to the service SharedNumber is not cancelled.
2	Tester → CuT#1, CuT#2	Tester removes SIM cards from CuT#1 and inserts it in CuT#2.	
3	Tester → CuT#2, CDuT	Tester pairs the CDuT with CuT#2	No new subscription is proposed to the user. No other eSIM profile is downloaded to CDuT.
4	Tester → CuT#2, CDuT	Tester disables Bluetooth on CDuT and CuT#2. Tester performs incoming call.	Both CuT#2 and CDuT rings.

Deep Inspection:

Check no new ICCID is returned from the AcquireConfiguration request, and no new download is performed by the CDuT.

3.4.6 Primary ODSA Entitlement Test Cases

3.4.7 DCB Entitlement Test Cases / any authentication

3.4.7.1 [DCB-10] DCB Entitlement Mode - Cannot purchase

Test Purpose

To ensure the CUT handles "Cannot purchase" entitlement mode properly.

Referenced requirements

TS43 [1] : 11.4.1 DCB Entitlement Mode - Cannot purchase

Initial Conditions

Entity	Description of the initial conditions
ECS	See conditions [ANY_S1 in section 3.2 Most frequent initial conditions The DCB is not provisioned for the given SIM card. No token has been retrieved in a previous OIDC authentication.
CuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app	CuT performs a request towards ECSuT. An OIDC authentication is performed.
2	ECSuT → CuT	After authentication, the configuration document provided does not contain a token but **EntitlementStatus=INCOMPATIBLE (2) for DCB.**	the device shall either display **MessageForIncompatible** when it is not void, or the default device error message (if any).

3.4.7.2 [DCB-11] DCB Entitlement Mode - Can purchase

Test Purpose

To ensure the CUT handles "Can purchase" entitlement mode properly.

Referenced requirements

TS43 [1] **:** 11.4.4 DCB Entitlement Mode - Can purchase

Initial Conditions

Entity	Description of the initial conditions
ECS	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions The provisioning data regarding the SIM are: • **EntitlementStatus** is ENABLED (1) • **TC_status** is AVAILABLE (1) or NOT REQUIRED (2)
CuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions DCB setting on the device is set to on by the user.

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app	CuT performs a configuration request towards ECSuT.
2	ECSuT → CuT	The configuration document is transmitted.	DCB should be activated on the device

3.4.7.3 [DCB-12] DCB Entitlement Mode – Service Data Missing

Test Purpose

To ensure the CUT handles "Service Data Missing" entitlement mode properly.

Referenced requirements

TS43 [1] **:** 11.4.3 DCB Entitlement Mode - Service Data Missing

Initial Conditions

Entity	Description of the initial conditions
ECS	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions The provisioning data regarding the SIM are: • **EntitlementStatus** is DISABLED (0) • **TC_status** is NOT AVAILABLE (0)
CuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions DCB setting on the device is set to on by the user.

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app	CuT performs a configuration request towards ECSuT.
2	ECSuT → CuT	The configuration document is transmitted.	DCB service should not be activated on the device. The CuT shall open a web view or display a message to instruct the end-user to enter the required missing DCB service information (T&C).

3.4.7.4 [DCB-14] DCB Entitlement Mode – Service Being Provisioned

Test Purpose

To ensure the CUT handles "Service Being Provisioned" entitlement mode properly.

Referenced requirements

TS43 [1] **:** 11.4.3 DCB Entitlement Mode - Service Being Provisioned

Initial Conditions

Entity	Description of the initial conditions
ECS	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions The provisioning data regarding the SIM are: • **EntitlementStatus** is DISABLED (0) • **TC_status** is set to AVAILABLE (1) or NOT REQUIRED (2) Or • **EntitlementStatus** is PROVISIONING (3) • **TC_status** is set to any values
CuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions DCB service setting on the device is set to on by the user.

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app	CuT performs a configuration request towards ECSuT.
2	ECSuT → CuT	The configuration document is transmitted.	DCB service should not be activated on the device. After an end-user action (going into DCB service settings for example), the CuT shall show that the service is pending or being provisioned.

3.4.7.5 [DCB-50] DCB-status indication & user actions

Test Purpose

To ensure CuT displays an indication about DCB-status. The way the information is provided (UI) is device dependant.

Referenced Clauses

TS43 [1]: 3 DCB Entitlement Configuration

Initial Conditions

Entity	Description of the initial conditions
ECSuT	See conditions [[EAP_S1]] in section 3.2 Most frequent initial conditions
CuT	See conditions [[EAP_C1] in section 3.2 Most frequent initial conditions CuT is powered off. The device is switch on for the 1st time (factory reset procedure can be done before)

Test execution

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester invokes CuT and triggers an entitlement configuration	CuT initiates entitlement configuration
2	ECSuT → CuT	After OIDC authentication procedure, a configuration document is sent to the CuT.	The tester can see on the store app the DCB feature is available. He can activate/deactivate the feature.

3.4.7.6 [DCB-60] DCB-payment

Test Purpose

To ensure the CuT performs DCB purchase when DCB is activated. The way the user is aware about DCB purchase is store dependant (UI).

Referenced requirements

TS43 [1]: 3.3 Entitlement Modes of DCB Client

Initial Conditions

Entity	Description of the initial conditions
ECS	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions
CuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions

Test execution

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app and enable DCB payment	DCB indication is enabled on the store.
2	Tester → device	Test perform DCB payment	Purchase is successful.

3.4.8 Message based Notifications

3.4.8.1 [PUSH-10] Push notification to refresh VoIMS services configuration

Test Purpose

To ensure configuration refresh on message notification is supported by CuT.

Referenced requirements

TS43 [1] : 2.6.2 Messaging Infrastructure-Based Notifications

Initial Conditions

Entity	Description of the initial conditions
ECS	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions
CuT	See conditions [EAP_S1] in section 3.2 Most frequent initial conditions The CuT is registered on a messaging service and provides a notification token to be addressed.

Test execution:

Step	Direction	Sequence	Expected Result
1	Tester → CuT	Tester starts client app	CuT performs a request towards ECSuT.
2	ECSuT → CuT	The configuration document provided by the ECSuT enables (EntitlementStatus=ENABLED (1) VoWifi,VoLTE and SMSoIP services	The services are activated on the device. Depending on the device, the UI reflects the service activation.
3	Tester →ECSuT/ messaging service	Tester disable the services for the CuT on the provisioning of the server side and emits a notification message towards the messaging service	A message is received by the CuT
4	CuT → ECSuT	The CuT request the new configuration document. The document now contains EntitlementStatus=DISABLED (0) for the services	The services are deactivated on the device. Depending on the device, the UI reflects the service deactivation.

Deep Inspection:

The CuT uses the token previously acquired to query the ECSuT at step 4.

3.4.9 SMS based notification

TBC

Annex A Exchanges examples

A.1 Configuration document

Possibly new authentication TOKEN generated by ECSuT with its validity in seconds

Proper services entitlement status requested in GET message

Exemple:

```
HTTP/1.1 200 OK
Content-Type: text/xml

<?xml version='1.0' encoding='UTF-8'?>
<wap-provisioningdoc version="1.1">
<characteristic type="VERS">
  <parm name="version" value="1"/>
  <parm name="validity" value="43200"/>
</characteristic>
<characteristic type="TOKEN">
  <parm name="token" value="MjA4MDE3NTAzNTExOTYxMTUyNzY2MjAwMDE4MQ=="/>
  <parm name="validity" value="86400"/>
</characteristic>
<characteristic type="APPLICATION">
  <parm name="AppID" value="ap2004"/>
  <parm name="Name" value="VoWiFi Entitlement settings"/>
  <parm name=”AppRef" value="VoWiFi-Settings"/>
  <parm name="EntitlementStatus" value=”1"/>
  <parm name="ServiceFlow_URL" value=""/>
  <parm name="ServiceFlow_UserData" value=""/>
  <parm name=”MessageForIncompatible” value=””/>
  <parm name="TC_Status" value="2"/>
  <parm name="AddrStatus" value="2"/>

  <parm name="ProvStatus" value="2"/>

</characteristic>

<characteristic type="APPLICATION">

  <parm name="AppID" value="ap2003"/>

  <parm name="Name" value="VoLTE Entitlement settings"/>

  <parm name=”AppRef" value="VoLTE-Settings"/>

  <parm name="EntitlementStatus" value=”1"/>

  <parm name=”MessageForIncompatible” value=””/>

</characteristic>

<characteristic type="APPLICATION">

  <parm name="AppID" value="ap2005"/>

  <parm name="Name" value=" SMSoIP Entitlement settings"/>

  <parm name=”AppRef" value=?SMSoIP-Settings"/>

  <parm name="EntitlementStatus" value=”1"/>

</characteristic>

</wap-provisioningdoc>
```

Annex B Document Management

Document History

Version	Date	Brief Description of Change	Approval Authority	Editor / Company
1.0	Dec 2022	New PRD TS.56	TSG#50 ISAG#26	François de Turenne / Orange

Other Information

Type	Description
Document Owner	Terminal Steering Group
Editor / Company	François de Turenne / Orange

It is our intention to provide a quality product for your use. If you find any errors or omissions, please contact us with your comments. You may notify us at prd@gsma.com

Your comments or suggestions & questions are always welcome.