

Service Entitlement Configuration

Version 3.4 Draft

03 November 2019

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 confidential to the Association and 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

Copyright © 2019 GSM Association

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.

Antitrust Notice

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

Table of Contents

[1 Introduction 5](#_Toc23716144)

[1.1 Overview 5](#_Toc23716145)

[1.2 In Scope 5](#_Toc23716146)

[1.3 Interactions with Other GSMA Specifications 6](#_Toc23716147)

[1.3.1 Positioning of VoWiFi, VoLTE and SMSoIP entitlements with respect to TAD and MNO Provisioning 6](#_Toc23716148)

[1.3.2 Relationship with TS.32, IR.51 and IR.92 VoWiFi/VoLTE/SMSoIP Parameters 8](#_Toc23716149)

[1.3.3 Controlling Access to Network and PS Data for Entitlement Configuration 9](#_Toc23716150)

[1.4 Abbreviations 9](#_Toc23716151)

[1.5 Definitions 11](#_Toc23716152)

[1.6 References 12](#_Toc23716153)

[1.7 Conventions 12](#_Toc23716154)

[2 Entitlement Configuration Procedures 12](#_Toc23716155)

[2.1 Default Entitlement Configuration Server 12](#_Toc23716156)

[2.2 HTTP GET method Parameters 13](#_Toc23716157)

[2.3 HTTP POST Method 14](#_Toc23716158)

[2.4 Network Requested Entitlement Configuration 15](#_Toc23716159)

[2.4.1 SMS-Based Notifications 15](#_Toc23716160)

[2.4.2 Messaging Infrastructure-Based Notifications 16](#_Toc23716161)

[2.5 Roaming Conditions 17](#_Toc23716162)

[2.6 Authentication Mechanism 17](#_Toc23716163)

[2.6.1 Embedded EAP-AKA Authentication by Entitlement Configuration Server 17](#_Toc23716164)

[2.6.2 Authentication With OAuth 2.0 / OpenID Connect Procedure 19](#_Toc23716165)

[2.7 Configuration Document for Entitlements 22](#_Toc23716166)

[2.7.1 General 22](#_Toc23716167)

[2.7.2 New Characteristics for XML-Based Document 22](#_Toc23716168)

[2.7.3 Inclusion in the Complete XML document 23](#_Toc23716169)

[2.7.4 JSON-Based Configuration Document 24](#_Toc23716170)

[2.7.5 Additional Details on TOKEN 24](#_Toc23716171)

[2.8 HTTP Response Codes 25](#_Toc23716172)

[3 VoWiFi Entitlement Configuration 26](#_Toc23716173)

[3.1 VoWiFi Entitlement Parameters 26](#_Toc23716174)

[3.1.1 VoWiFi Entitlement Status 26](#_Toc23716175)

[3.1.2 VoWiFi Client’s Web Views Parameters 26](#_Toc23716176)

[3.1.3 VoWiFi Address Parameters 27](#_Toc23716177)

[3.1.4 VoWiFi T&C Status 28](#_Toc23716178)

[3.1.5 VoWiFi Provisioning Status 29](#_Toc23716179)

[3.1.6 VoWiFi Message for Incompatible Status 29](#_Toc23716180)

[3.2 Client Behaviour for VoWiFi Entitlement Configuration 30](#_Toc23716181)

[3.3 Entitlement Modes of VoWiFi Client 30](#_Toc23716182)

[3.3.1 VoWiFi Entitlement Mode - Cannot be offered 31](#_Toc23716183)

[3.3.2 VoWiFi Entitlement Mode - Can be activated 31](#_Toc23716184)

[3.3.3 VoWiFi Entitlement Mode - Service Data Missing 31](#_Toc23716185)

[3.3.4 VoWiFi Entitlement Mode - Service Data Being Updated 32](#_Toc23716186)

[3.3.5 VoWiFi Entitlement Mode - Service Being Provisioned 32](#_Toc23716187)

[3.4 VoWiFi Client Considerations Around Web View Callbacks 32](#_Toc23716188)

[3.4.1 entitlementChanged() Callback function 33](#_Toc23716189)

[3.4.2 dismissFlow() callback function 34](#_Toc23716190)

[4 VoLTE Entitlement Configuration 36](#_Toc23716191)

[4.1 VoLTE Entitlement Parameters 36](#_Toc23716192)

[4.1.1 VoLTE Entitlement Status 36](#_Toc23716193)

[4.1.2 VoLTE Message for Incompatible Status 36](#_Toc23716194)

[4.2 Client Behaviour to VoLTE Entitlement Configuration 37](#_Toc23716195)

[5 SMSoIP Entitlement Configuration 38](#_Toc23716196)

[5.1 SMSoIP Entitlement Parameters 38](#_Toc23716197)

[5.1.1 SMSoIP Entitlement Status 38](#_Toc23716198)

[5.2 Client Behaviour to SMSoIP Entitlement Configuration 38](#_Toc23716199)

[6 On-Device Service Activation (ODSA) of Companion Devices 40](#_Toc23716200)

[6.1 Companion ODSA Architecture and Operations 40](#_Toc23716201)

[6.2 Companion ODSA Request Parameters 41](#_Toc23716202)

[6.3 Examples of Companion ODSA Requests 43](#_Toc23716203)

[6.3.1 EligibilityCheck Request Example 43](#_Toc23716204)

[6.3.2 ManageSubscription Request Example 43](#_Toc23716205)

[6.3.3 ManageService Request Example 44](#_Toc23716206)

[6.3.4 AcquireConfiguration Request Example 44](#_Toc23716207)

[6.4 Companion ODSA Configuration Parameters 45](#_Toc23716208)

[6.4.1 General / Always-Present Configuration Parameters 45](#_Toc23716209)

[6.4.2 CheckEligibility Operation Configuration Parameters 45](#_Toc23716210)

[6.4.3 ManageSubscription Operation Configuration Parameters 47](#_Toc23716211)

[6.4.4 ManageService Operation Configuration Parameters 49](#_Toc23716212)

[6.4.5 Acquire Configuration Operation Configuration Parameters 50](#_Toc23716213)

[6.5 Examples of Companion ODSA Responses 51](#_Toc23716214)

[6.5.1 EligibilityCheck Response Example 51](#_Toc23716215)

[6.5.2 ManageService Response Example 52](#_Toc23716216)

[6.5.3 ManageSubscription Response Example 52](#_Toc23716217)

[6.5.4 AcquireConfiguration Response Example 55](#_Toc23716218)

[6.6 Companion ODSA Procedure Call Flows 56](#_Toc23716219)

[6.6.1 Subscription Activation via ODSA Portal – Initial Steps 56](#_Toc23716220)

[6.6.2 ODSA Portal with Immediate Download Info – Final Steps 58](#_Toc23716221)

[6.6.3 ODSA Portal with Delayed Download Info – Final Steps 59](#_Toc23716222)

[6.6.4 Subscription Activation without ODSA Portal 61](#_Toc23716225)

[6.6.5 Subscription Pre-activation via Another Channel 63](#_Toc23716226)

[6.7 ODSA Application Considerations Around Web View Callback 65](#_Toc23716227)

[Annex A Document Management 66](#_Toc23716228)

[A.1 Document History 66](#_Toc23716229)

[Other Information 67](#_Toc23716230)

# Introduction

## Overview

This document describes the procedure for configuration of a device-based service performed during the entitlement verification step of the service or during the activation of that service.

The device services covered in this document are Voice-over-Wi-Fi (VoWiFi), Voice-over-LTE (VoLTE), SMS over IP (SMSoIP) and On-Device Service Activation (ODSA) of Companion devices (associated with the requesting device).

The specification leverages the protocol and document presentation described in GSMA PRD RCC.14 [5]. In this context, the term “entitlement” refers to the applicability, availability and status of that service (or feature) on a device.

The entitlement configuration is exchanged between a VoWiFi, VoLTE, SMSoIP or Companion ODSA client on a device and a Service Provider’s Entitlement Configuration Server. It is independent from the service configuration procedure between clients and the Service Provider’s configuration server described in RCC.14 [5].

Entitlement configuration defines a mechanism for a Service Provider to inform mobile devices of the status of IP Multimedia Subsystem (IMS) network services like VoWiFi, VoLTE and SMSoIP.

In the ODSA context it defines the interaction between an ODSA client, a client application on a device that entitles and activates a companion device’s subscription, and the Service Provider.

This procedure leverages the subscription profile of the end-user, identified by the SIM card, and the network’s readiness in supporting the service. The entitlement client can then dynamically activate (or deactivate) the service according to the activation (respectively deactivation) status retrieved from the Service Provider’s Entitlement Configuration Server.

When required by the service, entitlement configuration also covers on-device service activation flow, for example to display a web page describing the service or to get end-user consent on the service’s Terms and Conditions.

Service configuration in this document deals with the configuration parameters controlling the entitlement of a service. Those parameters come in addition to the ones defined in GSMA PRD IR.51 [2] and GSMA PRD IR.92 [3] that relate to the internal settings and configuration of IMS services. IMS service configuration as defined in GSMA PRD IR.51 [2] and GSMA PRD IR.92 [3] are out of scope.

## In Scope

This document covers both the device and network aspects of the entitlement configuration for VoWiFi, VoLTE and SMSoIP services as well as for On-Device Service Activation (ODSA) of Companion devices. Service-specific aspects need to be described in documents relating to those services as in GSMA IR.51 [2] and GSMA PRD IR.92 [3] for IMS services.

The entitlement configuration can be obtained via either cellular or Wi-Fi data connectivity. In case Wi-Fi data connection is used, this document assumes that a Wi-Fi bearer is available to the device and the requirements of that Wi-Fi bearer conform to GSMA PRD TS.22 [7]. Configuration and provisioning of the Wi-Fi bearer is described in GSMA PRD TS.22 [7] Section 3.

## Interactions with Other GSMA Specifications

Entitlement configuration is an optional mechanism between applications/services on devices (like VoWiFi and VoLTE) and the SP’s core network that occurs during service activation. The procedure requires both end-user’s subscription data and network readiness information from the SP.

To support that exchange, an entitlement configuration server leverages the RCC.14 [5] protocol to carry the required entitlement data between devices’ applications and the network. The entitlement configuration procedure is separate from the service configuration procedure specified in RCC.14 [5]. A device or application shall not query for both entitlement and service configurations in the same request.

The result of entitlement configuration for a service offers the assurance that the end-user’s associated subscription and the core network’s readiness have been verified, allowing the service to be offered to the end-user.

### Positioning of VoWiFi, VoLTE and SMSoIP entitlements with respect to TAD and MNO Provisioning

The positioning of VoWiFi, VoLTE and SMSoIP entitlement configuration with respect to existing GSMA device configuration procedures (GSMA PRD TS.32 [8], GSMA PRD IR.51 [2] and GSMA PRD IR.92 [3]) is presented in Figure 1. It shows the typical timeline and triggers that would induce the procedures (note that the horizontal axis represents Time).



1. : TS.43 VoWiFi, VoLTE and SMSoIP entitlement procedure with respect to TS.32, IR.51 and IR.92

The GSMA PRD TS.32 [8] procedure of Technical Adaptation of Device (TAD) is implemented by device OEMs on a MNO-wide basis (or a range of IMSI) due to the device’s factory reset or SIM detection. General IMS, VoLTE and VoWiFi parameter values are set without taking into account end-user subscription or network related information.

The MNO provisioning procedure of GSMA PRD and also offers the possibility of setting general IMS, VoLTE and VoWiFi parameters on the device during initial service configuration. However, it is not associated with user-triggered service activation or the verification of the services’ entitlement / applicability.

The entitlement-level configuration for VoLTE and VoWiFi specified in the GSMA PRD TS.43 takes place after or outside the aforementioned GSMA’s device and service configuration procedures. It is also triggered by events not associated with GSMA PRD TS.32 [8], GSMA PRD IR.51 [2] and GSMA PRD IR.92 [3] :

* when the service needs to verify its entitlement status (during service initiation),
* when the end-user wishes to activate the service (via the service’s settings menu)

### Relationship with TS.32, IR.51 and IR.92 VoWiFi/VoLTE/SMSoIP Parameters

The VoWiFi, VoLTE and SMSoIP configuration parameters of this PRD complement the ones from GSMA PRD TS.32 [8], GSMA PRD IR.51 [2] and GSMA PRD IR.92 [3].

While those specifications define general-purpose VoWiFi, VoLTE and SMSoIP parameters to enable or disable those services on the device, the GSMA PRD TS.43 defines parameters that relate to service initiation and end-user activation (capture of Terms & Conditions, capture of physical address).

The parameters in this PRD are also based on end-user subscription’s data and on the network readiness for those services.

In case the VoWiFi, VoLTE or SMSoIP service has not been allowed and activated on the device due to a Technical Adaptation of Device (TAD) or MNO provisioning procedure, the client performing the entitlement configuration should be disabled.

The VoLTE, SMSoIP and VoWiFi configuration parameters defined in each specification are presented in Table 1.

| GSMA PRD | VoLTE Status Parameters | SMSoIP Status Parameters | VoWiFi Status Parameters |
| --- | --- | --- | --- |
| GSMA PRD TS.32[8] | * **VxLTE 1.27** Voice/Video over LTE allowed when roaming * **VxLTE 1.28** Voice/Video over LTE allowed | * **VxLTE 1.07** SMSoIP Networks Indications (not used or preferred) | * **VoWIFI 3.01** Voice and Video / Voice enabled over Wi-Fi |
| GSMA PRD IR.92[3] | As a **Media\_type\_restriction\_policy**   * Voice and/or Video over LTE allowed * Voice and/or Video over LTE allowed while roaming | **SMSoIP\_usage\_policy**  (When to use SMSoIP) | N/A |
| GSMA PRD IR.51[2] | N/A | N/A | As a **Media\_type\_restriction\_policy**   * Voice and/or Video over Wi-Fi enabled |
| TS.43 (this document) | * VoLTE entitlement status | * SMSoIP entitlement status | * VoWiFi entitlement status * VoWiFi T&Cs capture status * VoWiFi address capture status * VoWiFi provisioning status |

1. : VoLTE, SMSoIP and VoWiFi Configuration Parameters in GSMA Specifications

Note: That the configuration parameter VxLTE 1.21 - IMS Enabled (Yes/No) from and “IMS Status” from is not impacted by the GSMA PRD TS.43. The overall IMS function on the device can still be controlled by this parameter.

### Controlling Access to Network and PS Data for Entitlement Configuration

GSMA PRD IR.92 [3] defines parameters to allow device and client services to be exempt of the 3GPP PS Data Off feature. When one such parameter, **Device\_management\_over\_PS**, is set, it indicates that device management over PS is a 3GPP PS data off exempt service.

GSMA PRD TS.43 extends the **Device\_management\_over the\_PS** parameter to include Entitlement Configuration as a type of “device management” service that can be exempt of 3GPP PS Data Off.

The home operator can also configure a policy on the Entitlement Client around the access type used during entitlement configuration. This is done with the **AccessForEntitlement** parameter with values listed in Table 2.

| AccessForEntitlement Value | Description |
| --- | --- |
| 0 | any access type |
| 1 | 3GPP accesses only |
| 2 | WLAN/Wi-Fi only |
| 3 | 3GPP accesses preferred, WLAN/Wi-Fi as secondary |
| 4 | WLAN/Wi-Fi preferred, 3GPP accesses as secondary |
| 5-255 | not assigned |

1. : AccessForEntitlement Parameter

A "not assigned" value is interpreted as "any access type" value.

When not preconfigured by the home operator with the **AccessForEntitlement** parameter, the Entitlement Client shall perform entitlement configuration requests over Wi-Fi if available. When there is no Wi-Fi connectivity, the Entitlement Client shall perform requests over cellular if it is not forbidden (i.e. PS data off and not exempt).

## Abbreviations

|  |  |
| --- | --- |
| **Abbreviation** | **Definition** |
| CP AC | Client Provisioning Application Characteristic |
| DNS | Domain Name Server |
| EAP-AKA | Extensible Authentication Protocol for 3rd Generation Authentication and Key Agreement |
| EID | eUICC Identifier |
| eUICC | Embedded Universal Integrated Circuit Card |
| FCM | Firebase Cloud Messaging |
| FQDN | Fully Qualified Domain Name |
| GCM | Google Cloud Messaging |
| 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 |
| LPA | Local Profile Assistant |
| LTE | Long-Term Evolution |
| MCC | Mobile Country Code (As defined in E.212) |
| MNC | Mobile Network Code (As defined in E.212) |
| MO | Management Object |
| MSISDN | Mobile Subscriber Integrated Services Digital Network Number |
| ODSA | On-Device Service Activation |
| OIDC | OpenID Connect |
| OMNA | Open Mobile Naming Authority, registry available at: <http://www.openmobilealliance.org> |
| OTP | One-Time Password |
| PRD | Permanent Reference Document |
| RCS | Rich Communication Services |
| 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 |
| URL | Uniform Resource Locator |
| VoWiFi | Voice-over-Wi-Fi |
| VoLTE | Voice-over-LTE |
| WNS | Windows Push Notification Service |
| XML | Extensible Markup Language |
| XSD | Extensible Markup Language Schema Definition |

## Definitions

| **Definition** | **Meaning** |
| --- | --- |
| Client | Component/module on a device that provides the VoLTE or VoWiFi service. A 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, needed by the 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 clients. |

## References

| **Ref** | **Document Number** | **Title** |
| --- | --- | --- |
|  | OMA-APPIDREG | OMA Registry of Application Identifiers (AppID) <http://www.openmobilealliance.org/wp/OMNA/dm/dm_ac_registry.html> |
|  | IR.51 | GSMA PRD IR.51 - “IMS Profile for Voice, Video and SMS over untrusted Wi-Fi access” Version 5.0, 23 May 2017. <http://www.gsma.com> |
|  | IR.92 | GSMA PRD IR.92 - “IMS Profile for Voice and SMS” Version 11.0, 23 June 2017. <http://www.gsma.com> |
|  | NG.102 | GSMA PRD NG.102 - “IMS Profile for Converged IP Communications”  Version 4.0 13 June 2017. <http://www.gsma.com> |
|  | RCC.14 | GSMA PRD RCC.14 “Service Provider Device Configuration”, Version 5.0, 28 June 2017. <http://www.gsma.com> |
|  | RFC2119 | “Key words for use in RFCs to Indicate Requirement Levels”, S. Bradner, March 1997. <http://www.ietf.org/rfc/rfc2119.txt> |
|  | TS.22 | Recommendations for Minimum Wi-Fi Capabilities of Terminals, Version 5.0, 07 April 2016. <http://www.gsma.com> |
|  | TS.32 | Technical Adaptation of Devices through Late Customisation, Version 3.0, 20 September 2017. <http://www.gsma.com> |
|  | E.212 | Mobile network codes (MNC) for the international Identification plan for public networks and subscriptions(according to recommendation ITU-T E.212 (05/2008)) |
|  | SGP.21 | Remote SIM Provisioning Architecture. <http://www.gsma.com> |
|  | SGP.22 | Remote SIM Provisioning Technical Specification. <http://www.gsma.com> |

## Conventions

“The key words “must”, “must not”, “required”, “shall”, “shall not”, “should”, “should not”, “recommended”, “may”, and “optional” in this document are to be interpreted as described in [6].”

# Entitlement Configuration Procedures

## Default Entitlement Configuration Server

The client shall follow a discovery procedure to obtain the address of the entitlement configuration server, with a resulting FQDN based on the following format:

* + aes.mnc<MNC>.mcc<MCC>.pub.3gppnetwork.org

Whereby <MNC> (Mobile Network Code) and <MCC> (Mobile Country Code) shall be replaced by the respective values of the home network in decimal format and with a 2-digit MNC padded out to 3 digits by inserting a 0 at the beginning.

## HTTP GET method Parameters

A client supporting service entitlement configuration shall indicate the support by inclusion of an "app" HTTP GET request parameter as defined in with the proper identifiers for the targeted entitlement.

The Open Mobile Naming Authority (OMNA) maintains a registry of values for Application Characteristic Identifier (AppID) and the range ap2001-ap5999 is used for externally defined Application entities. The following AppIDs are used for VoWiFi, VoLTE and SMSoIP entitlement applications, and for the ODSA for Companions application:

* VoLTE Entitlement - AppID of “ap2003”
* VoWiFi Entitlement - AppID of “ap2004”
* SMSoIP Entitlement – AppID of “ap2005”
* ODSA for Companion, Entitlement and Activation – AppID of “ap2006”

The GET parameters from RCC.14 [5] are used for entitlement configuration requests (“IMSI”, “token”, “vers”, “app”, “terminal\_vendor”, “terminal\_model”, “terminal\_sw\_version”). In addition new parameters are introduced specific for the entitlement purposes, as described in Table 3.

| HTTP GET parameter | Type | Description | Usage |
| --- | --- | --- | --- |
| terminal\_id | String | The unique identifier for the device, for example IMEI (preferred) or a UUID. This identifier must be persistent. | Required. |
| entitlement\_version | String | Current version of the entitlement specification. Set to “2.0”. | Required. |
| app\_name | String | The name of the device application making the request. | Optional. |
| app\_version | String | The version of the device application making the request. | Optional. |
| notif\_token | String | The registration token to be used when notifications are transmitted to the device over a cloud-based messaging infrastructure (refer to 2.3). | Optional, required each time the device obtains or disables a registration token from the notification service.  Sent at the same time as “notif\_action” parameter. |
| notif\_action | Integer | The action associated with the registration token “notif\_token” parameter.  Possible values are:   * 0 - disable notification token * 1 - enable GCM notification token * 2 - enable FCM notification token * 3 - enable WNS push notification | Optional, required if the “notif\_token” parameter is present. |

1. : GET Parameters for Entitlement Configuration Request

Table 4 presents a sample HTTP GET request for VoWiFi entitlement with the parameters located in the HTTP query string.

|  |
| --- |
| GET ? terminal\_id = 013787006099944&  token = es7w1erXjh%2FEC%2FP8BV44SBmVipg&  terminal\_vendor = TVENDOR&  terminal\_model = TMODEL&  terminal\_sw\_version = TSWVERS&  app = ap2004&  vers = 1 HTTP/1.1  Host: entitlement.telco.net:9014  User-Agent: Mozilla/5.0 (Windows NT 6.1; Gecko/20100101 Firefox/43.0  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8  Accept-Language: en-US,en;q=0.5  Accept-Encoding: gzip, deflate  Connection: keep-alive |

1. : Example of an HTTP GET Entitlement Configuration Request

## HTTP POST Method

In addition to the HTTP GET, the HTTP POST method can be used by the client for entitlement configuration request. In this case, the parameters are located in the HTTP message body and should follow the JSON object value format. The same parameters defined in Section 2.2 are used for the POST request.

If a client supports the POST method, it shall use it instead of the GET method for entitlement configuration requests. The Entitlement Configuration Server should be able to process both GET and POST methods. In case the server does not support POST, it shall return an HTTP response with 405 “Method Not Allowed”. In that case, the client should resend the request using the GET method.

Table 5 presents a sample HTTP POST request for VoWiFi entitlement with the parameters located in the HTTP message body.

|  |
| --- |
| POST / HTTP/1.1  Host: entitlement.telco.net:9014  User-Agent: Mozilla/5.0 (Windows NT 6.1; Gecko/20100101 Firefox/43.0  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8  Accept-Language: en-US,en;q=0.5  Accept-Encoding: gzip, deflate  Connection: keep-alive  Content-Type: application/json  {  "terminal\_id" : "013787006099944",  "token" : "es7w1erXjh%2FEC%2FP8BV44SBmVipg",  "terminal\_vendor" : "TVENDOR",  "terminal\_model" : "TMODEL",  "terminal\_sw\_version" : "TSWVERS",  "app" : "ap2004",  "vers" : "1"  } |

1. : Example of an HTTP POST Entitlement Configuration Request

## Network Requested Entitlement Configuration

Two mechanisms are available to operators to trigger an entitlement configuration request from a device application, either:

* by sending a Short Message Service (SMS) message to the target device, or
* by sending a notification message to the device over a cloud-based messaging infrastructure (FCM, GSM or WNS)

When an application is notified in this manner, it shall generate the proper GET Entitlement request to the entitlement configuration server:

* For applications “ap2003”, “ap2004” and “ap2005” (VoLTE, VoWiFi or SMSoIP entitlement) a GET for the corresponding app is generated.
* For application “ap2006” (ODSA for Companion), a GET for app of ap2006 and the operation of AcquireConfiguration is generated.

### SMS-Based Notifications

To notify the target device of a change in the entitlement configuration, the entitlement configuration server can use the same method described in Chapter 3 of RCC.14 [5] and generate a Short Message Service (SMS) message towards the target device via application-port addressed SMS with a User Data Header (UDH).

The User Data Header (UDH) contains the following Information Elements:

* **UDH length** : 6 (six octets)
* **Information-Element-Identifier** (IEI) : x05, message is using "application port addressing scheme, 16-bit address"
* **Destination application port** : by default set to 8095 or 0x1F9F
* **Source application port** : set to 0

The content of the message is different from RCC.14 [5], in order to differentiate a network-triggered notification coming from a configuration server and one coming from an entitlement configuration server:

* + - Instead of the SMS user-data set to: user-id "-**rcs**cfg" [ "," parm ]
    - The following is used: user-id "-**aes**cfg" [ "," parm ]

The “parm” parameter contains the application(s) notified with this SMS. An example of the SMS content is:

214011001388741-aescfg,ap2003

This message would trigger (or wake up) the VoLTE application on the device to create and send a request (HTTP GET with service parameters) to the Entitlement Configuration Server. If several applications are targeted, they would appear as a comma-separated list, for example:

214011001388741-aescfg,ap2003,ap2004,ap2005

### Messaging Infrastructure-Based Notifications

A notification message can also be sent by the Entitlement Configuration Server to the device over a cloud-based messaging infrastructure that devices registered with in order to receive network-initiated messages. The device’s application is reached and identified via the notif\_token present in the original GET request received by the entitlement configuration server.  
  
The details of the cloud-based messaging technology is implementation dependent and not covered in this specification. The payload of the notification message is a JSON object value that should contain a “**data**” element with at least two key-value pairs:

* + - "**app**" : the application targeted for re-configuration, with value of either "ap2003", “ap2004”, “ap2005” or “ap2006”.   
      If multiple applications are targeted, the value is a JSON array of strings.
    - "**timestamp**" : the time of the notification, in ISO 8601 format, of the form YYYY-MM-DDThh:mm:ssTZD, where TZD is time zone designator (Z or +hh:mm or -hh:mm)
    - An example of the notification payload for VoLTE follows:

"data":

{

"app": "ap2003",

"timestamp": "2019-01-29T13:15:31-08:00"

}

* + - An example of the notification payload for multiple applications follows:

"data":

{

"app": ["ap2003", "ap2004", "ap2005"],

"timestamp": "2019-01-29T13:15:31-08:00"

}

## Roaming Conditions

The fact that the device is roaming does not impact the ability of a client to request an entitlement configuration. The client can send the HTTP-based entitlement configuration request over an available data connection, either Wi-Fi or a cellular data APN. Refer to for the configuration and usage of those connections as related to operator traffic.

The device can therefore be in a roaming situation when requesting for an entitlement configuration on VoLTE and/or VoWiFi.

## Authentication Mechanism

The different authentication procedures described in of RCC.14 [5] shall be followed during the entitlement configuration exchange.

Entitlement configuration is usually triggered by the device or client and the user is not aware of an entitlement configuration process taking place. It is then preferable for the entitlement configuration server to rely on authentication mechanisms like “User Authentication via HTTP Embedded EAP-AKA” which does not involve user interactions.

In case access to the device’s SIM data is not possible (which would prevent authentication based on EAP-AKA), authentication following the OpenID or OAuth 2.0 procedure is the preferred alternative.

Both authentication methods are detailed in the following two sections.

### Embedded EAP-AKA Authentication by Entitlement Configuration Server

The Embedded EAP-AKA procedure of RCC.14 [5] involves a separate authentication server included in the flow as part of an HTTP Redirect response (as per OpenID Connect). In case an operator does not carry such OpenID Connect authentication server with EAP relay capabilities and its entitlement configuration server supports the EAP relay function, it is possible for the server to omit the HTTP Redirect and exchange the EAP payloads directly with the client.

This flow is shown in Figure 2. Note that the EAP payload specification along with the GET and POST headers and parameters defined in for the HTTP Embedded EAP-AKA procedure of RCC.14 [5] are kept. The only difference is the omission of the HTTP 302 Found responses (HTTP redirects).



1. – Embedded EAP-AKA Authentication Flow with Entitlement Configuration Server Supporting EAP Relay Function

If the Entitlement Configuration Server is handling the EAP-AKA relay to an operator’s Authentication server (a 3GPP AAA for example), Table 6 shows the mapping between the response codes from the 3GPP AAA and the corresponding HTTP GET response. The response code are coming from AVP « Result-Code » or AVP « Experimental-Result » sent by the 3GPP AAA in the Diameter EAP Response (DER).

| DER Result Code | HTTP GET Response | Reason |
| --- | --- | --- |
| 1001 | 200 OK | Waiting for AKA challenge response from device |
| 2001 | 200 OK | Successfully authenticated by AAA |
| 4001,  5001, 5003 | 503 Retry After / Service Unavailable | 4001 DIAMETER\_AUTHENTICATION\_REJECTED,  5001 DIAMETER\_ERROR\_USER\_UNKNOWN, and  5003 DIAMETER\_ERROR\_IDENTITY\_NOT\_REGISTERED are considered errors that can be resolved at the device |
| 3001-3010,  5002, 5004-5017 | 503 Retry After / Service Unavailable | Those are error codes for protocol errors, and miscellaneous AAA errors, that could be resolved in subsequent requests from devices |
| Connection failure to 3GPP AAA | 500 Internal Server Error | Lack of connectivity to 3GPP AAA is presented as an internal error for the Entitlement Configuration Server |

1. : Mapping Between 3GPP AAA’s DER Result Code and HTTP Response Code

### Authentication With OAuth 2.0 / OpenID Connect Procedure

The OpenID Connect (OIDC) authentication method is available for clients that cannot access the AKA function of the SIM on the device and the Service Provider decides not to use other Authentication methods like SMS-OTP. The end-user must instead go through an authentication procedure managed by the Service Provider’s OAuth 2.0 / OIDC authentication server. The invocation of OIDC-based authentication by the entitlement configuration server follows the procedure defined in Section 2.8 of RCC.14 [5].

Figure 3 presents an overview of the steps for the OIDC-based authentication procedure, shown here for informational purposes.

* After deciding that OIDC procedure is needed (lack of token or invalid token, no EAP\_ID in GET request, other authentication methods not available), the entitlement configuration server redirects (with 302 Found) the GET request from the device’s client to the Service Provider’s OIDC authentication endpoint.
* The OIDC authentication endpoint can offer different types of authenticators, some of which involve actual user actions.
* When the end-user is authenticated, the entitlement configuration server will receive an OAuth 2.0 “auth code” from the authentication server (via the client or user agent on the device, again using a 302 Found).
* The entitlement configuration server requests for both an access token and an ID Token from the Service Provider’s OIDC Token endpoint.
* After validating the OAuth 2.0 access token and the OpenID token, the entitlement configuration server can identify the end-user subscription and resumes processing of the original GET resource request.



1. – OAuth 2.0 / OpenID Authentication Flow with Entitlement Configuration Server

## Configuration Document for Entitlements

### General

The attributes for the entitlement of VoWiFi, VoLTE or SMSoIP and for the requested operations of the “ODSA for Companion devices” application are conveyed between the entitlement configuration server and the client via a configuration document. This document is located in the returned 200 OK response message and can follow two formats:

* An XML document similar to the one defined in section 4 of RCC.14 [5], composed of a set of characteristic types, each with a number of parameters
* A JSON object composed of a number of structured values (a set of fields presented as name-value pairs) corresponding to the characteristic types of the XML document

### New Characteristics for XML-Based Document

Extending the XML definition from RCC.14 [5], new APPLICATION characteristics are defined for the entitlements of VoWiFi, VoLTE, SMSoIP and for the operations of the “ODSA for Companion devices” application, with a unique Application Identifier (AppID) assigned to each.

Refer to 2.2 for the AppID assigned to the entitlement applications for VoWiFi, VoLTE, SMSoIP and to the “ODSA for Companion devices” application.

An example configuration document containing the combined entitlement parameters for the VoWiFi, VoLTE and SMSoIP services is shown in Table 7. This is an example and as such non-normative. The example presents all those entitlements, but only the requested service entitlements shall be included in the document (based on the received “app” request parameter).

For the application “ODSA for Companion devices”, refer to 6.5 for the XML document examples defined for each operation of that application.

|  |
| --- |
| <characteristic type="APPLICATION">  <parm name=”AppID” value=”ap2004”/>  <parm name=”Name” value=”VoWiFi Entitlement settings”/>  <parm name=”EntitlementStatus” value=”X”/>  <parm name=”ServiceFlow\_URL” value=”X”/>  <parm name=”ServiceFlow\_UserData” value=”X”/>  <parm name=”MessageForIncompatible” value=”X”/>  <parm name=”AddrStatus” value=”X”/>  <parm name=”TC\_Status” value=”X”/>  <parm name=”ProvStatus” value=”X”/>  </characteristic>  <characteristic type="APPLICATION">  <parm name=”AppID” value=”ap2003”/>  <parm name=”Name” value=”VoLTE Entitlement settings”/>  <parm name=”EntitlementStatus” value=”X”/>  <parm name=”MessageForIncompatible” value=”X”/>  </characteristic>  <characteristic type="APPLICATION">  <parm name=”AppID” value=”ap2005”/>  <parm name=”Name” value=”SMSoIP Entitlement settings”/>  <parm name=”EntitlementStatus” value=”X”/>  </characteristic> |

1. : VoWiFi, VoLTE and SMSoIP entitlement document structure (non-normative)

### Inclusion in the Complete XML document

The complete XML document with combined VoWiFi, VoLTE and SMSoIP entitlement configurations is illustrated in Table 8. This is an example and as such non-normative. The example presents all those entitlements, but only the requested service entitlements shall be included in the document (based on the received “app” request parameter).

|  |
| --- |
| <?xml version="1.0"?>  <wap-provisioningdoc version="1.1">  <characteristic type="VERS">  <parm name="version" value="X"/>  <parm name="validity" value="Y"/>  </characteristic>  <characteristic type="TOKEN"> <!-- This section is OPTIONAL -->  <parm name="token" value="U"/>  <parm name="validity" value="V"/> <!-- Optional parameter -->  </characteristic>  <!-- Potentially additional optional characteristics such as MSG, User and Access Control -->  <!-- see [PRD-RCC.14] -->  <characteristic type="APPLICATION">  <parm name=”AppID” value=”ap2004”/>  <parm name=”Name” value=”VoWiFi Entitlement settings”/>  <parm name=”EntitlementStatus” value=”X”/>  <parm name=”ServiceFlow\_URL” value=”X”/>  <parm name=”ServiceFlow\_ UserData” value=”X”/>  <parm name=”MessageForIncompatible” value=”X”/>  <parm name=”AddrStatus” value=”X”/>  <parm name=”TC\_Status” value=”X”/>  <parm name=”ProvStatus” value=”X”/>  </characteristic>  <characteristic type="APPLICATION">  <parm name=”AppID” value=”ap2003”/>  <parm name=”Name” value=”VoLTE Entitlement settings”/>  <parm name=”EntitlementStatus” value=”X”/>  <parm name=”MessageForIncompatible” value=”X”/>  </characteristic>  <characteristic type="APPLICATION">  <parm name=”AppID” value=”ap2005”/>  <parm name=”Name” value=”SMSoIP Entitlement settings”/>  <parm name=”EntitlementStatus” value=”X”/>  </characteristic>  </wap-provisioningdoc> |

1. : Complete XML-based entitlement document structure (non-normative)

### JSON-Based Configuration Document

The JSON object value returned as part of an entitlement configuration request for the entitlements of VoWiFi, VoLTE and SMSoIP is presented in Table 9. Each characteristic type of the XML document is mapped to the JSON document as a structured object with several fields.

For the application “ODSA for Companion devices”, refer to 6.5 for a description of the JSON-based document defined for each operation of that application.

|  |
| --- |
| {  "Vers" : {  "version" : "X",  "validity" : "Y"  }, "Token" : { // Optional  "token" : "U",  "validity" : "V"  }, "ap2004": { // VoWiFi Entitlement settings  "EntitlementStatus" : "X",  "ServiceFlow\_URL" : "X",  "ServiceFlow\_ UserData" : "X",  "MessageForIncompatible" : "X",  "AddrStatus" : "X",  "TC\_Status" : "X",  "ProvStatus" : "X"  }, "ap2003" : { // VoLTE Entitlement settings  "EntitlementStatus" : "X",  "MessageForIncompatible" : "X"  }, "ap2005" : { // SMSoIP Entitlement settings  "EntitlementStatus" : "X"  }  } |

1. : JSON-based entitlement document for VoWiFi, VoLTE and SMSoIP (non-normative)

### Additional Details on TOKEN

As seen in Table 8 and Table 9, the document for entitlement configuration contains the VERS and TOKEN attributes, as defined by RCC.14 [5]. In addition to the definition of TOKEN from RCC.14, the following rules apply to the entitlement configuration’s TOKEN:

* TOKEN is not restricted to entitlement configuration requests made from non-3GPP access networks access types
* A “validity” attribute is allowed and indicates the lifetime of the provided token
* The token shall be kept by clients during reboot cycles
* The token is of variable length

## HTTP Response Codes

Table 10 presents the possible entitlement configuration server response codes (including associated reasons) at the HTTP level.

| GET Response Code | Reason | Device’s Action |
| --- | --- | --- |
| **200** OK + with application data | New or updated application data sent to the device | Process the returned application data |
| **302** Found | OAuth 2.0 / OpenID Connect authentication should be followed. Refer to Section 2.5.2 for details on the procedure and its initiation. | Redirect the GET request to the OIDC AuthN endpoint specified by the **Location**: field of the 302 Found response |
| **400** Bad Request | Invalid or missing GET parameters or wrong format | Retry on next reboot/the next time the client app starts |
| **403** Forbidden | Invalid identities (device id, primary or companion) | Retry on next reboot/the next time the client app starts |
| **405** Method not Allowed | The server does not support the HTTP POST method used by the client | Retry the request using GET method |
| **500** Internal Server error | Internal error during processing of GET request | Retry on next reboot/the next time the client app starts |
| **503** Retry after / Service Unavailable | The server does not have access to external resources (temporary error) | Retry after the time specified in the “**Retry- After**” header |
| **511** Network Authentication Required | To initiate authentication with the server, when proper AuthN parameters are missing, the OTP is invalid, or the token obtained through a previous authentication exercise expired | Client app should go through an authentication procedure with the entitlement configuration server and get a new token |
| The server is unreachable | Entitlement configuration server is missing or down | Retry on next reboot, the next time the client starts |

1. : HTTP Response Codes from Entitlement Configuration Server

# VoWiFi Entitlement Configuration

The following sections describe the different configuration parameters associated with the VoWiFi entitlement as well as the expected behaviour of the VoWiFi client based on the entitlement configuration document received by the client.

## VoWiFi Entitlement Parameters

Parameters for the VoWiFi entitlement provide the overall status of the VoWiFi service to the client, as well as the different sub-status associated with the activation procedure of the service.

The VoWiFi entitlement parameters also include information associated with the web views presented to users by the VoWiFi client during activation and management of the service.

### VoWiFi Entitlement Status

* Parameter Name: EntitlementStatus
* Presence: Mandatory

This parameter indicates the overall status of the VoWiFi entitlement, stating if the service can be offered on the device, and if it can be activated or not by the end-user.

The different values for the VoWiFi entitlement status are provided in Table 11.

| VoWiFi Entitlement parameter | Type | Values | Description |
| --- | --- | --- | --- |
| EntitlementStatus (Mandatory) | Integer | 0 - DISABLED | VoWiFi service allowed, but not yet provisioned and activated on the network side |
| 1 - ENABLED | VoWiFi service allowed, provisioned and activated on the network side |
| 2 - INCOMPATIBLE | VoWiFi service cannot be offered |
| 3 - PROVISIONING | VoWiFi service being provisioned on the network side |

1. : Entitlement Parameter - VoWiFi Overall Status

### VoWiFi Client’s Web Views Parameters

* Parameter Names: ServiceFlow\_URL and ServiceFlow\_UserData
* Presence: Mandatory

During the activation procedure of the VoWiFi service, end-users can be presented with web views specific to the Service Provider. VoWiFi web views allow end-users to change user-specific attributes of the VoWiFi service, like the acceptance of the service’s Terms and Conditions (T&C) and the end-user’s physical address (needed in some regions for VoWiFi emergency calling purposes).

The entitlement parameters associated with the VoWiFi service’s web views are described in Table 12.

| VoWiFi Entitlement parameter | Type | Description |
| --- | --- | --- |
| ServiceFlow\_URL (Mandatory) | String | The URL of web views to be used by VoWiFi client to present the user with VoWiFi service activation and service management options, which may include entering physical address and agreeing to the T&C of the VoWiFi service. |
| ServiceFlow\_UserData (Mandatory) | String | User data associated with the HTTP web request towards the ServiceFlow URL. It can contain user-specific attributes to ease the flow of VoWiFi service activation and management.  See below for details on the content. |

1. : Entitlement Parameters - VoWiFi Web Views Information

The content of the ServiceFlow\_UserData parameter is defined by the requirements of the Service Provider’s VoWiFi web views. In a typical case, the web view is presented when VoWiFi service is activated by the end-user. At such time the VoWiFi client connects the user to the ServiceFlow\_URL and includes the ServiceFlow\_UserData in the HTTP web request.

In order to improve user experience, this parameter should include user and service-specific information that would allow the VoWiFi’s web views to identify the requestor and be aware of the latest VoWiFi entitlement status values.

An example of the ServiceFlow\_UserData string is:

"imsi=XXXXXXXXX&amp;msisdn=XXXXXXXX&amp;tnc=X&ampaddr=X&amp;prov=X&amp;device\_id=XXXXXXXX&amp;entitlement\_name=VoWiFi&amp;signature=Xl%2F1tT23C0dNI32hiVZZS”

This example contains elements associated with the device and user identities as well as service-related information like the current T&C, address and provisioning status of the VoWiFi service. Note the use of “&amp;” is required to allow the ‘&’ character to be used in a string value within an XML document.

### VoWiFi Address Parameters

* Parameter Name: AddrStatus, AddrExpiry, AddrIdentifier
* Presence:
  + - * AddrStatus: Mandatory
      * AddrExpiry, AddrIdentifier: Optional

In some regions, end-users must provide their static physical address before being allowed to use the VoWiFi service. Those entitlement parameters indicates if that condition must be met before offering the VoWiFi service and provide additional information on the captured location (expiration and identifier).

Also, if a physical address from the end-user is indeed needed for the VoWiFi service, this parameter indicates the state of the “address capture” process.

The different values for the VoWiFi address status are provided in Table 13.

| VoWiFi Entitlement parameter | Type | Values | Description |
| --- | --- | --- | --- |
| AddrStatus (Mandatory) | Integer | 0 - NOT AVAILABLE | Address has not yet been captured from the end-user |
| 1 - AVAILABLE | Address has been entered by the end-user |
| 2 - NOT REQUIRED | Address is not required to offer VoWiFi service |
| 3 - IN PROGRESS | Address capture from end-user is on-going |
| AddrExpiry (Optional) | Time | in ISO 8601 format, of the form YYYY-MM-DDThh:mm:ssTZD | The time/date when the address expires and should be recaptured from the user |
| AddrIdentifier (Optional) | String | Generated by emergency system | Associated identifier of the location, to be used during an IMS emergency session by the device, as defined in . |

1. : Entitlement Parameters - VoWiFi Address

The absence of the AddrExpiry parameter indicates that there is no expiration date for the address.

### VoWiFi T&C Status

* Parameter Name: TC\_Status
* Presence: Mandatory

In some regions, end-users must agree to the Terms and Conditions (T&C) of the VoWiFi service before being allowed to use it. This entitlement parameter indicates if that condition must be met before offering the VoWiFi service.

Also, if acceptance of the VoWiFi’s T&C is indeed needed from the end-user, this parameter indicates the state of the “T&C acceptance” process.

The different values for the VoWiFi T&C status are provided in Table 14.

| VoWiFi Entitlement parameter | Type | Values | Description |
| --- | --- | --- | --- |
| TC\_Status (Mandatory) | Integer | 0 - NOT AVAILABLE | T&C have not yet been accepted by the end-user |
| 1 - AVAILABLE | T&C have been accepted by the end-user |
| 2 - NOT REQUIRED | T&C acceptance is not required to offer VoWiFi service |
| 3 - IN PROGRESS | T&C capture and acceptance is on-going |

1. : Entitlement Parameter - VoWiFi T&C Status

### VoWiFi Provisioning Status

* Parameter Name: ProvStatus
* Presence: Mandatory

In some cases, the network is not provisioned by default to support VoWiFi service for all end-users. Some type of network-side provisioning must then take place before offering the VoWiFi service to the end-user. This entitlement parameter indicates the progress of VoWiFi provisioning on the network for the requesting client.

The different values for the VoWiFi provisioning status are provided in Table 15.

| VoWiFi Entitlement parameter | Type | Values | Description |
| --- | --- | --- | --- |
| ProvStatus (Mandatory) | Integer | 0 - NOT PROVISIONED | VoWiFi service not provisioned yet on network side |
| 1 - PROVISIONED | VoWiFi service fully provisioned on network |
| 2 - NOT REQUIRED | Provisioning progress of VoWiFi is not tracked / not required |
| 3 - IN PROGRESS | VoWiFi provisioning is still in progress |

1. : Entitlement Parameter - VoWiFi Provisioning Status

### VoWiFi Message for Incompatible Status

* Parameter Name: MessageForIncompatible
* Presence: Mandatory

When the status for the VoWiFi entitlement is INCOMPATIBLE (see 3.1.1) and the end-user tries to activate VoWiFi, the VoWiFi client should show a message to the end-user indicating why activation was refused.

This entitlement parameter provides the content of that message, as decided by the Service Provider. Table 16 describes this VoWiFi entitlement parameter.

| VoWiFi Entitlement parameter | Type | Description |
| --- | --- | --- |
| MessageForIncompatible (Mandatory) | String | A message to be displayed to the end-user when activation fails due to an incompatible VoWiFi Entitlement Status |

1. : Entitlement Parameter - VoWiFi Message for Incompatible Status

## Client Behaviour for VoWiFi Entitlement Configuration

The entitlement parameters for VoWiFi provides an overall status for the service as well as additional information associated with the activation procedure and provisioning of the service.

As such, the entitlement configuration for VoWiFi carries information that impacts the behaviour of the VoWiFi client.

The client shall then activate (or deactivate) the VoWiFi service according to the combination of the VoWiFi’s general setting on the device (controlled by the end-user) and the received VoWiFi entitlement configuration.

The client shall also use the VoWiFi entitlement parameters to decide if VoWiFi web views for activation and service management should be presented to the end-user. This include country-specific details on the need for VoWiFi’s Terms & Conditions acceptance and the requirement to capture or not the user’s physical address - a country’s regulations may require users to enter their physical address as well as agree to the Terms & Conditions of the service when activating VoWiFi.

## Entitlement Modes of VoWiFi Client

To simplify the description of the client’s behavior with respect to the VoWiFi entitlement configuration, a set of “VoWiFi entitlement modes” for the client is defined, each with specific expectations on the client side.

The relationship between the values of the VoWiFi entitlement parameters and the VoWiFi entitlement modes are shown in Table 17.

| VoWiFi Entitlement Parameters | | | | VoWiFi Entitlement mode |
| --- | --- | --- | --- | --- |
| Entitlement  Status | ProvStatus | TC\_status | AddrStatus |
| INCOMPATIBLE | Any | | | Cannot be offered |
| DISABLED | Any | At least one is NOT AVAILABLE | | Service Data Missing |
| At least one is IN PROGRESS | | Service Data being Updated |
| DISABLED | NOT PROVISIONED, IN PROGRESS | AVAILABLE or NOT REQUIRED | | Service being Provisioned |
| PROVISIONING | Any | | |
| ENABLED | PROVISIONED or NOT REQUIRED | AVAILABLE or NOT REQUIRED | | Can be activated |

1. : VoWiFi Entitlement Modes

The description of each VoWiFi entitlement mode follows.

### VoWiFi Entitlement Mode - Cannot be offered

The Client shall stay in this mode when:

* **EntitlementStatus** is INCOMPATIBLE

The Client shall not activate the VoWiFi service.

Due to end-user’s action, the client may send a request to the Entitlement Configuration Server to refresh the VoWiFi entitlement status. If the received status is still INCOMPATIBLE, the device shall either display **MessageForIncompatible** when it is not void, or the default device error message (if any).

### VoWiFi Entitlement Mode - Can be activated

The Client shall stay in this mode when all the following conditions are met:

* **EntitlementStatus** is ENABLED
* **ProvStatus** is PROVISIONED or NOT REQUIRED
* **TC\_status** and **AddrStatus** are AVAILABLE or NOT REQUIRED

When entering this mode, the client shall activate the VoWiFi service if the VoWiFi’s service setting on the device is equivalent to ON (may require end-user action).

### VoWiFi Entitlement Mode - Service Data Missing

The Client shall stay in this mode when all the following conditions are met:

* **EntitlementStatus** is DISABLED
* **ProvStatus** is any values
* Either **TC\_status** or **AddrStatus** is NOT AVAILABLE

In that mode the Client shall not activate the VoWiFi service.

Due to end-user’s action, the Client may send a request to the Entitlement Configuration Server to refresh the VoWiFi entitlement status. If the received status lead to the same mode, the Client shall open a web view and instruct the end-user to enter the required missing VoWiFi service information (T&C or static physical address).

### VoWiFi Entitlement Mode - Service Data Being Updated

The Client shall stay in this mode when all the following conditions are met:

* **EntitlementStatus** is DISABLED
* **ProvStatus** is any values
* Either **TC\_status,** or **AddrStatus** is set to IN PROGRESS

In that mode the Client shall not activate the VoWiFi service.

### VoWiFi Entitlement Mode - Service Being Provisioned

The Client shall stay in this mode when all the following conditions are met:

* **EntitlementStatus** is DISABLED
* **TC\_status** and **AddrStatus** are set to AVAILABLE or NOT REQUIRED
* **ProvStatus** is set to NOT PROVISIONED or IN PROGRESS

Or

* **EntitlementStatus** is PROVISIONING
* **ProvStatus,TC\_status** and **AddrStatus** are set to any values

The Client shall not activate the VoWiFi service. After an end-user action (going into VoWiFi service settings for example), the client shall show that the service is pending or being provisioned.

## VoWiFi Client Considerations Around Web View Callbacks

During the activation procedure of the VoWiFi service, end-users can be presented with web views specific to the Service Provider (hosted by a VoWiFi portal web server). To support this feature, the VoWiFi entitlement parameters **ServiceFlow\_URL** and **ServiceFlow\_UserData** associated with the invocation of VoWiFi service’s web views by the VoWiFi client are defined in section 3.1.2.

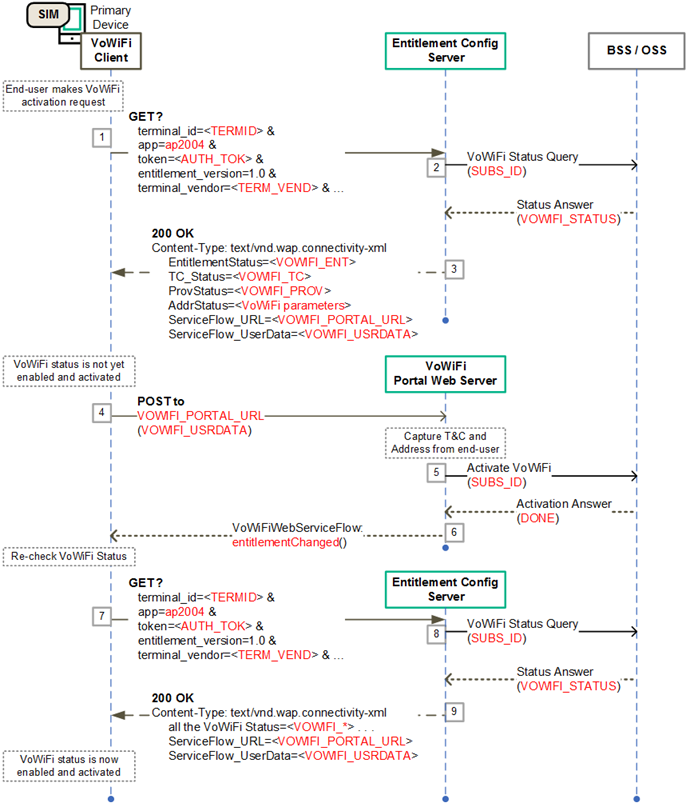
At the completion of the web service flow by the VoWiFi portal web server, the web page shall invoke a specific JavaScript (JS) callback function associated with the VoWiFi client. The callback functions shall provide the overall state of the web flow to the VoWiFi client and indicate that the VoWiFi web view on the device needs to be closed.

The object associated with the callback functions is **VoWiFiWebServiceFlow** and two different callback functions are defined to reflect the state of the web logic.

### entitlementChanged() Callback function

The **entitlementChanged**() callback function indicates that the VoWiFi service flow ended properly between the device and and VoWiFi portal web server.  
  
The web view to the end-user should be closed and the VoWiFi client shall make a request for the latest VoWiFi entitlement configuration status, via the proper TS.43 entitlement configuration request.  
  
Based on the returned set of status parameters, the VoWiFi client shall behave as specified in 3.3.

The following call flow presents how the **entitlementChanged**() callback function fits into the typical steps involved with VoWiFi entitlement configuration. At the end of the VoWiFi service flow the callback function (step 7) is invoked by the web server and the VoWiFi client acts accordingly by requesting for the latest VoWiFi entitlement configuration.



1. - VoWiFi Entitlement Configuration Flow with entitlementChanged() Callback

### dismissFlow() callback function

The **dismissFlow**() callback function indicates that the VoWiFi service flow ends prematurely, either caused by user action (DISMISS button for example) or by an error in the web sheet logic or from the network side.

As a result of the dismissal of the service flow, the VoWiFi entitlement status has not been updated by the VoWiFi portal.

The web view to the end-user should be closed and the VoWiFi client should not make a request for the latest VoWiFi entitlement configuration status.

The call flow in Figure 5 presents how the **dismissFlow**() callback function fits into the typical steps involved with VoWiFi Entitlement Configuration. Due to an error or user action the callback function (step 6) is invoked by the web server and the VoWiFi client acts accordingly.



1. - VoWiFi Entitlement Configuration Flow with dismissFlow() Callback

# VoLTE Entitlement Configuration

The following sections describe the different configuration parameters associated with the VoLTE entitlement as well as the expected behaviour of the VoLTE client based on the entitlement configuration document received by the client.

## VoLTE Entitlement Parameters

Parameters for the VoLTE entitlement provide the overall status of the VoLTE service to the client and other client-related information.

### VoLTE Entitlement Status

* Parameter Name: EntitlementStatus
* Presence: Mandatory

This parameter indicates the overall status of the VoLTE entitlement, stating if the service can be offered on the device, and if it can be activated or not by the end-user.

The different values for the VoLTE entitlement status are provided in Table 18.

| VoLTE Entitlement parameter | Type | Values | Description |
| --- | --- | --- | --- |
| EntitlementStatus (Mandatory) | Integer | 0 - DISABLED | VoLTE service allowed, but not yet provisioned and activated on the network side |
| 1 - ENABLED | VoLTE service allowed, provisioned and activated on the network side |
| 2 - INCOMPATIBLE | VoLTE service cannot be offered |
| 3 - PROVISIONING | VoLTE service being provisioned on the network side |

1. : Entitlement Parameter - VoLTE Overall Status

### VoLTE Message for Incompatible Status

* Parameter Name: MessageForIncompatible
* Presence: Mandatory

When the status for the VoLTE entitlement is INCOMPATIBLE and the end-user tries to activate VoLTE, the client should show a message to the end-user indicating why activation was refused.

This entitlement parameter provides the content of that message, as decided by the Service Provider. Table 19 describes this VoLTE entitlement parameter.

| VoLTE Entitlement parameter | Type | Description |
| --- | --- | --- |
| MessageForIncompatible (Mandatory) | String | A message to be displayed to the end-user when activation fails due to an incompatible VoLTE Entitlement Status |

1. : Entitlement Parameter - VoLTE Message for Incompatible Status

## Client Behaviour to VoLTE Entitlement Configuration

The client shall activate (or deactivate) the VoLTE service according to the combination of the VoLTE settings on the device (controlled by the end-user) and the received VoLTE Entitlement status described in this document. This is presented in Table 20.

| VoLTE Entitlement Status | VoLTE Client Behavior |
| --- | --- |
| INCOMPATIBLE | The Client shall not activate the VoLTE service.  The client may send a request to the Entitlement Configuration Server to refresh the VoLTE entitlement status. If the received status is still INCOMPATIBLE, the device shall either display **MessageForIncompatible** parameter when it is not void, or the default device error message (if any). |
| DISABLED | The Client shall not activate the VoLTE service.  After an end-user action (going into VoLTE service settings for example), the client may send a request to the Entitlement Configuration Server to refresh the VoLTE entitlement status. |
| PROVISIONING | The Client shall not activate the VoLTE service.  After an end-user action (going into VoLTE service settings for example), the client shall show that the service is pending or being provisioned. |
| ENABLED | The client shall activate the VoLTE service if the VoLTE’s service setting on the device is equivalent to ON (may require end-user action). |

1. : VoLTE Client Behaviour

# SMSoIP Entitlement Configuration

The following sections describe the different configuration parameters associated with the SMSoIP entitlement as well as the expected behaviour of the SMSoIP client based on the entitlement configuration document received by the client.

## SMSoIP Entitlement Parameters

Parameters for the SMSoIP entitlement provide the overall status of the SMSoIP service to the client and other client-related information.

### SMSoIP Entitlement Status

* Parameter Name: EntitlementStatus
* Presence: Mandatory

This parameter indicates the overall status of the SMSoIP entitlement, stating if the service can be offered on the device, and if it can be activated or not by the end-user.

The different values for the SMSoIP entitlement status are provided in Table 21.

| SMSoIP Entitlement parameter | Type | Values | Description |
| --- | --- | --- | --- |
| EntitlementStatus (Mandatory) | Integer | 0 - DISABLED | SMSoIP service allowed, but not yet provisioned and activated on the network side |
| 1 - ENABLED | SMSoIP service allowed, provisioned and activated on the network side |
| 2 - INCOMPATIBLE | SMSoIP service cannot be offered |
| 3 - PROVISIONING | SMSoIP service being provisioned on the network side |

1. : Entitlement Parameter - SMSoIP Overall Status

## Client Behaviour to SMSoIP Entitlement Configuration

The client shall activate (or deactivate) the SMSoIP service according to the combination of the SMSoIP settings on the device (controlled by the end-user) and the received SMSoIP Entitlement status described in this document. This is presented in Table 22.

| SMSoIP Entitlement Status | SMSoIP Client Behavior |
| --- | --- |
| INCOMPATIBLE | The Client shall not activate the SMSoIP service.  The client may send a request to the Entitlement Configuration Server to refresh the SMSoIP entitlement status. |
| DISABLED | The Client shall not activate the SMSoIP service.  After an end-user action (going into SMSoIP’s service settings for example), the client may send a request to the Entitlement Configuration Server to refresh the SMSoIP entitlement status. |
| PROVISIONING | The Client shall not activate the SMSoIP service.  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. |
| ENABLED | The client shall activate the SMSoIP service if the SMSoIP’s service setting on the device is equivalent to ON (may require end-user action). |

1. : SMSoIP Client Behaviour

# On-Device Service Activation (ODSA) of Companion Devices

The ODSA procedure for companion devices is initiated by a client application on a requesting device (called the “primary” device). The ODSA application requires entitlement and configuration information from the Service Provider in order to complete the procedure. The following sections present the different operations associated with ODSA of Companion devices and the resulting configuration documents.

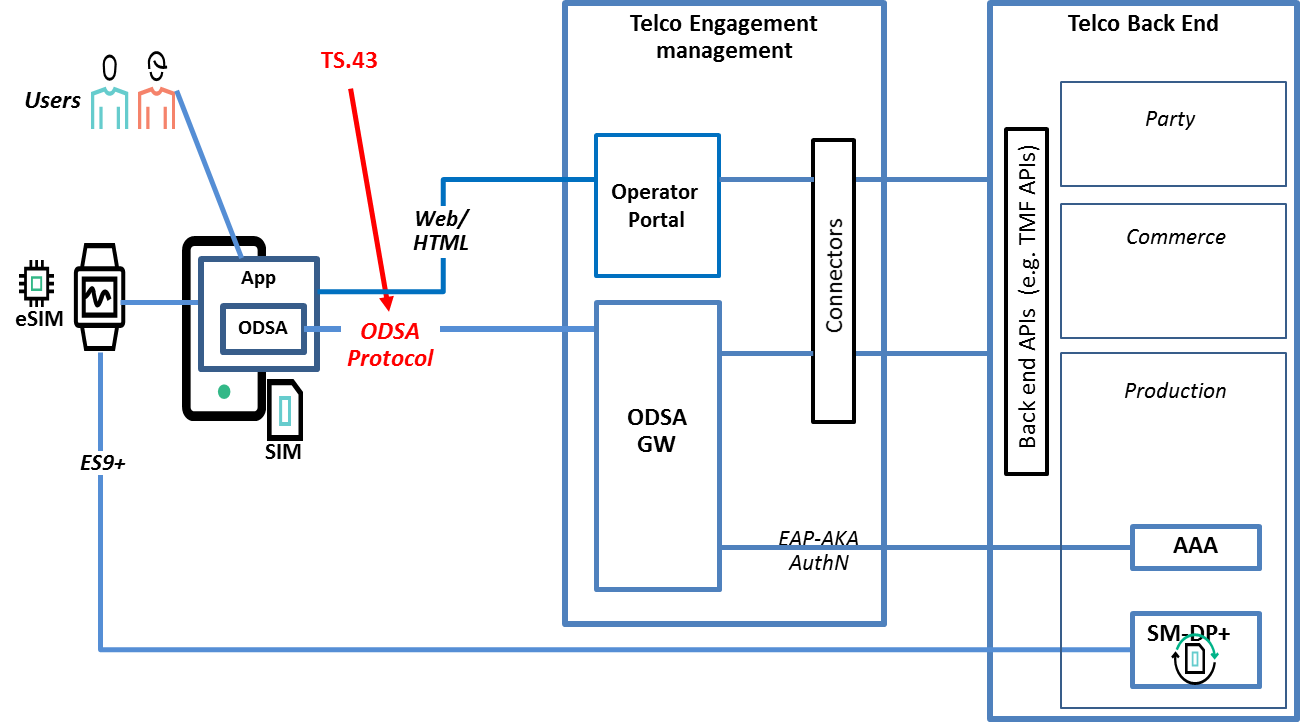
## Companion ODSA Architecture and Operations

The Companion ODSA Client application runs on a primary device and allows the end-user to perform a seamless activation of the subscription and associated services on the eSIM of the companion device, without involvement of Service Provider’s customer or support personnel.

In order to have access to the companion device, the ODSA application on the primary device shall be invoked at the request of the end-user and shall capture proper interactions (e.g. user consent) as described in SGP.21 [10] and SGP.22 [11].

The architecture is shown in Figure 6. The Entitlement Configuration Server acts as the Service Provider’s Device Gateway for the ODSA procedure (labelled as the “ODSA GW” in Figure 6), providing entitlement and configuration data to the “ODSA for Companion devices” application.

This specification does not cover the HTML-based interactions between the ODSA application and the Service Provider’s portal web server (labelled as the “Operator Portal” in Figure 6). The ODSA web server can be used to present different subscription options to the end-user and capture Terms & Conditions agreements.



1. – ODSA for Companion devices architecture and TS.43 positioning

The product implementations including for the Entitlement Configuration Server and the Service Provider’s portal web server shall protect the privacy of the subscriber and of the end-user on all data that could be used for tracking such as ICCID, MSISDN, EID.

Instead of just one entitlement configuration request, the Companion ODSA application requires several exchanges with the Entitlement Configuration Server. Each exchange is associated with an operation, resulting in the need of a new string-based operation GET request parameter.

Table 23 presents the allowed operations for the Companion ODSA procedure.

| Companion ODSA Operation | Description |
| --- | --- |
| CheckEligibility | To verify if end-user is allowed to invoke the Companion ODSA application |
| ManageSubscription | To request for subscription-related action on a companion device. |
| ManageService | To activate / deactivate the service on the companion device. This is an optional operation. |
| AcquireConfiguration | To provide service-related data on a companion device. |

1. : Companion ODSA Operations

## Companion ODSA Request Parameters

The ODSA procedure for Companion devices requires additional parameters in the GET requests, outside of the RCC.14 [5] specification. Table 24 presents the new parameters and their associated ODSA operations.

| New GET parameters for Companion ODSA | Type | Values | Description |
| --- | --- | --- | --- |
| operation | String | CheckEligibility ,  ManageSubscription,  ManageService,  AcquireConfiguration | Indicates the operation requested by the “ODSA for Companion devices” application |
| operation\_type | Integer | Used by the ManageSubscription operation. | |
| 0 - SUBSCRIBE | Indicates this is a request to activate a subscription for the companion device. |
| 1 - UNSUBSCRIBE | Indicates this is a request to cancel a subscription for the companion device. |
| 2 – CHANGE SUBSCRIPTION | Indicates this is a request to manage an existing subscription on the companion device. |
| Used by the ManageService operation. | |
| 10 – ACTIVATE SERVICE | Indicates this is a request to activate a service on the companion device. |
| 11 – DEACTIVATE SERVICE | Indicates this is a request to deactivate a service on the companion device. |
| companion\_terminal\_id | String | Used by **all** the Companion ODSA operations. | |
| Any string value | A unique identifier for the companion device. Suggested source is the IMEI of the device. |
| companion\_terminal\_ vendor  (Optional) | String | Used by the operations CheckEligibility, ManageSubscription and ManageService. | |
| Any string value | Manufacturer of the companion device. |
| companion\_terminal\_ model  (Optional) | String | Used by the operations CheckEligibility, ManageSubscription and ManageService. | |
| Any string value | Model of the companion device. |
| companion\_terminal\_ sw\_version  (Optional) | String | Used by the operations CheckEligibility, ManageSubscription and ManageService. | |
| Any string value | Software version of the companion device. |
| companion\_terminal\_ friendly\_name (Optional) | String | Used by the operations CheckEligibility, ManageSubscription and ManageService. | |
| Any string value | User-friendly identification for the companion device which can be used by the Service Provider in Web Views. |
| companion\_terminal\_ service | String | Used by the ManageService operation | |
| SharedNumber | Indicates that the service being managed is “Shared Number”, where the companion device carries the same MSISDN as the primary device |
| DiffNumber | Indicates that the service being managed is “Different Number”, where the companion device carries a different MSISDN from the primary device |
| companion\_terminal\_ iccid  (Optional) | String | Used by the ManageSubscription and AcquireConfiguration operations. | |
| Value following the ICCID format | The ICCID of the companion device being managed, provided only if there is a communication profile on the companion’s eUICC |
| companion\_terminal\_eid (Optional) | String | Used by the ManageSubscription and AcquireConfiguration operations. | |
| Value following eUICC format | eUICC identifier (EID) of the companion device being managed |

1. : New GET parameters for Companion ODSA application

## Examples of Companion ODSA Requests

This section presents samples of Companion ODSA requests using the GET method. It is also possible to use the POST method as indicated in Section 2.3. In the POST case, the parameters would be located in the message body as a JSON object instead of being in the HTTP query string.

### EligibilityCheck Request Example

Table 25 presents an example for the Eligibility Check operation for a Companion ODSA application.

|  |
| --- |
| GET ? terminal\_id = 013787006099944&  token = es7w1erXjh%2FEC%2FP8BV44SBmVipg&  terminal\_vendor = TVENDOR&  terminal\_model = TMODEL&  terminal\_sw\_version = TSWVERS&  app = ap2006&  operation = CheckEligibility&  companion\_terminal\_id = 98112687006099944&  vers = 1 HTTP/1.1  Host: entitlement.telco.net:9014  User-Agent: Mozilla/5.0 (Windows NT 6.1; Gecko/20100101 Firefox/43.0  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8  Accept-Language: en-US,en;q=0.5  Accept-Encoding: gzip, deflate  Connection: keep-alive |

1. : Example of a CheckEligibility Companion ODSA Request

### ManageSubscription Request Example

Table 26 presents an example for the Manage Subscription operation for a Companion ODSA application.

|  |
| --- |
| GET ? terminal\_id = 013787006099944&  token = es7w1erXjh%2FEC%2FP8BV44SBmVipg&  app = ap2006&  operation = ManageSubscription&  operation\_type = 0& ! subscribe  companion\_terminal\_id = 98112687006099944&  companion\_terminal\_eid = JHSDHljhsdfy763hh&  vers = 1 HTTP/1.1  Host: entitlement.telco.net:9014  User-Agent: Mozilla/5.0 (Windows NT 6.1; Gecko/20100101 Firefox/43.0  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8  Accept-Language: en-US,en;q=0.5  Accept-Encoding: gzip, deflate  Connection: keep-alive |

1. : Example of a ManageSubscription Companion ODSA Request

### ManageService Request Example

Table 27 presents an example for the Manage Service operation for a Companion ODSA application.

|  |
| --- |
| GET ? terminal\_id = 013787006099944&  token = es7w1erXjh%2FEC%2FP8BV44SBmVipg&  terminal\_vendor = TVENDOR&  terminal\_model = TMODEL&  terminal\_sw\_version = TSWVERS&  app = ap2006&  operation = ManageService&  operation\_type = 10& ! activate service  companion\_terminal\_id = 98112687006099944&  companion\_terminal\_service = DiffNumber&  vers = 1 HTTP/1.1  Host: entitlement.telco.net:9014  User-Agent: Mozilla/5.0 (Windows NT 6.1; Gecko/20100101 Firefox/43.0  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8  Accept-Language: en-US,en;q=0.5  Accept-Encoding: gzip, deflate  Connection: keep-alive |

1. : Example of a ManageService Companion ODSA Request

### AcquireConfiguration Request Example

Table 28 presents an example for the Acquire Configuration operation for a Companion ODSA application.

|  |
| --- |
| GET ? terminal\_id = 013787006099944&  token = es7w1erXjh%2FEC%2FP8BV44SBmVipg&  terminal\_vendor = TVENDOR&  terminal\_model = TMODEL&  terminal\_sw\_version = TSWVERS&  app = ap2006&  operation = AcquireConfiguration&  companion\_terminal\_id = 98112687006099944&  vers = 1 HTTP/1.1  Host: entitlement.telco.net:9014  User-Agent: Mozilla/5.0 (Windows NT 6.1; Gecko/20100101 Firefox/43.0  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8  Accept-Language: en-US,en;q=0.5  Accept-Encoding: gzip, deflate  Connection: keep-alive |

1. : Example of a AcquireConfiguration Companion ODSA Request

## Companion ODSA Configuration Parameters

### General / Always-Present Configuration Parameters

* Parameter names:
  + - * OperationResult: Mandatory

This parameter provides the result of the requested operation as described in Table 29.

| General Configuration Parameter | Type | Values | Description |
| --- | --- | --- | --- |
| OperationResult (Mandatory) | Integer | 1 - SUCCESS | Operation was a success |
| 100 - ERROR, GENERAL | There was a general error during processing |
| 101 - ERROR, INVALID OPERATION | An invalid operation value was provided in request |
| 102 - ERROR, INVALID PARAMETER | An invalid parameter name or value was provided in request |

1. : General Configuration Parameters for ODSA Operation

### CheckEligibility Operation Configuration Parameters

* Parameter names and presence:
  + - * CompanionAppEligibility: Mandatory
      * CompanionDeviceServices: Mandatory
      * NotEnabledURL: Optional
      * NotEnabledUserData: Optional
      * NotEnabledContentsType: Optional

Those parameters are associated with the eligibility of offering the Companion ODSA application on the requesting device and for the end-user. The application usually runs on the primary device (with SIM).The eligibility value can be based on factors like the type of end-user’s subscription/plans and the device details.

The CompanionDeviceServices parameter represents the different services that can be activated on the companion device.

The URL, User Data and Contents Type parameters offer the option of using operator-specific web views when the end-user attempts to invoke the Companion ODSA application when it is not enabled. If absent, the device presents instead an internally-generated message to the end-user.

The different values for the configuration parameters of the CheckEligibility operation are provided in Table 30.

| “Check Eligibility” Configuration parameter | Type | Values | Description |
| --- | --- | --- | --- |
| CompanionAppEligibility (Mandatory) | Integer | 0 - DISABLED | Companion ODSA app cannot be offered and invoked by the end-user |
| 1 - ENABLED | Companion ODSA app can be invoked by end-user to activate companion device subscription |
| 2 - INCOMPATIBLE | Companion ODSA app is not compatible with the device |
| CompanionDeviceServices (Mandatory) | String | Comma-separated list with all services available on the companion device | |
| SharedNumber | Indicates that the “Shared Number” service is active on the companion device (where the device carries the same MSISDN as the primary one) |
| DiffNumber | Indicates that the “Diff Number” service is active on the companion device (where the device carries a different MSISDN from the primary one) |
| NotEnabledURL (Optional) | String | URL to a Service Provider site or portal | The provided URL shall present a Web view to user on the reason(s) why the Companion ODSA app cannot be used/invoked |
| NotEnabledUserData (Optional) | String | Parameters or content to insert when invoking URL provided in the NotEnabledURL parameter | User data sent to the Service Provider when requesting the NotEnabledURL web view.  It should contain user-specific attributes to improve user experience.  The format must follow the NotEnabledContentsType parameter. |
| NotEnabledContentsType (Optional) | String | Specifies content and HTTP method to use when reaching out to the web server specified in NotEnabledURL. | |
| NOT present | Method to NotEnabledURL is HTTP GET request with query parameters from NotEnabledUserData. |
| json | Method to NotEnabledURL is HTTP POST request with JSON content from NotEnabledUserData. |
| xml | Method to NotEnabledURL is HTTP POST request with XML content from NotEnabledUserData. |

1. : Configuration Parameters – Check Eligibility ODSA Operation

### ManageSubscription Operation Configuration Parameters

* Parameter names and presence:
  + - * SubscriptionResult: Mandatory
      * SubscriptionServiceURL: Conditional
      * SubscriptionServiceUserData: Conditional
      * SubscriptionServiceContentsType: Conditional
      * DownloadInfo: Conditional

Those parameter provide the result of a companion ODSA subscription request, including any additional data needed to complete the subscription (URL to send users to, or communication profile download information for the companion eSIM).

The different values for the configuration parameters of the ManageSubscription operation are provided in Table 31.

| “Manage Subscription” Configuration parameters | Type | Values | Description |
| --- | --- | --- | --- |
| SubscriptionResult (Mandatory) | Integer | 1 - CONTINUE TO WEBSHEET | Indicates that end-user must go through the companion subscription web view procedure, using information included below. |
| 2 - DOWNLOAD PROFILE | Indicates that a communication profile must be downloaded by companion device, with further information included in response |
| SubscriptionServiceURL (Conditional) | String | URL to a Service Provider site or portal | Present only if SubscriptionResult is “1”. URL refers to web views responsible for a certain action on the companion device subscription.  The Service Provider can provide different URL based on the operation\_type input parameter (subscribe, unsubscribe, change subscription). |
| SubscriptionServiceUserData (Conditional) | String | Parameters to insert when invoking URL provided in Subscription ServiceURL | Present only if SubscriptionResult is “1”, and also optional. User data sent to the Service Provider when requesting the SubscriptionServiceURL web view.  It should contain user-specific attributes to improve user experience.  The format must follow SubscriptionServiceContentsType. |
| SubscriptionService ContentsType (Conditional) | String | Specifies content and HTTP method to use when reaching out to the web server specified by SubscriptionServiceURL | |
| NOT present | Method to SubscriptionServiceURL is HTTP GET request with query parameters from SubscriptionServiceUserData. |
| “json” | Method to SubscriptionServiceURL is HTTP POST request with JSON content from SubscriptionServiceUserData. |
| “xml” | Method to SubscriptionServiceURL is HTTP POST request with XML content from SubscriptionServiceUserData. |
| DownloadInfo (Conditional) | Structure | multi-parameter value - see next table for details | Present if SubscriptionResult is “2”. Specifies how and where to download the communication profile associated with the companion device. |

1. : Configuration Parameters – Manage Subscription ODSA Operation

The DownloadInfo configuration parameter is defined as a structure with several parameters as shown in Table 32.

| “Download Info” parameters | Type | Values | Description |
| --- | --- | --- | --- |
| ProfileIccid (Optional) | String | ICCID of the profile to download from SM-DP+ | Can be a new ICCID or the re-usable ICCID that was provided in the request parameter companion\_terminal\_iccid. |
| ProfileSmdpAddress (Conditional) | String | URL to SM-DP+ platform of MNO | Address(es) of SM-DP+ to obtain communication profile. If more than one, they must be comma-separated.  It is not needed if ProfileActivationCode is present. |
| ProfileActivationCode (Conditional) | String | Encoded in Base64. Must follow the activation code format from GSMA SGP.22 | Activation code as defined in SGP.22 to permit the download of a communication profile from an SM-DP+.  It is not needed if ProfileSmdpAddress is present. |

1. : Configuration Parameters – Download Info for Manage Subscription

### ManageService Operation Configuration Parameters

* Parameter names and presence:
  + - * ServiceStatus: Mandatory

The parameter provide the result of a Companion ODSA service request.

The different values for the configuration parameters of the ManageService operation are provided in Table 33.

| “Manage Service” Configuration parameters | Type | Values | Description |
| --- | --- | --- | --- |
| ServiceStatus (M) | Integer | 1 - ACTIVATED | Companion device’s service is activated. |
| 2 - ACTIVATING | Companion device’s service is being activated. |
| 3 - DEACTIVATED | Companion device’s service is not activated. |
| 4 - DEACTIVATED, NO REUSE | Companion device’s service is not activated and the associated ICCID should not be reused. |

1. : Configuration Parameters – Manage Service ODSA Operation

### Acquire Configuration Operation Configuration Parameters

* Parameter names and presence:
  + - * CompanionConfigurations: Conditional,  
        Top level, present if there is one or more companion device(s) associated with the primary device
      * CompanionConfiguration: Mandatory,   
        Within CompanionConfigurations, one or more

CompanionConfiguration is a multi-parameter structure that provides the configuration settings of the subscription and service running on a companion device.

If the companion device subscription was just activated by the Service Provider and the requesting AcquireConfiguration operation was the first one received since the activation, CompanionConfiguration shall contain a DownloadInfo element. As with the ManageSubscription operation, DownloadInfo specifies how to obtain the communication profile for the companion device from the Service Provider.

The CompanionConfiguration structure has the parameters listed in Table 34.

| “Acquire Configuration” configuration parameters | Type | Values | Description |
| --- | --- | --- | --- |
| ICCID (Conditional) | String | a valid ICCID, encoded as a 10 octet string | Integrated Circuit Card Identification - Identifier of the communication profile on the companion device’s eSIM.  Present if a profile exists for the companion eSIM. |
| CompanionDeviceService (Mandatory) | String | SharedNumber | Indicates that the configuration is for the “Shared Number” service (where the device carries the same MSISDN as the primary one) |
| DiffNumber | Indicates that the configuration is for the “Different Number” service (where the device carries a different MSISDN from the primary one) |
| ServiceStatus (Mandatory) | Integer | 1 to 4 | Refer to Table 33 for a description of the allowed values for ServiceStatus. |
| DownloadInfo (Conditional) | Structure | multi-parameter value - see Table 30 for details | Specifies how and where to download the communication profile associated with the companion device.  Present in case the profile is to be downloaded at this stage. |

1. : Configuration Parameters – Acquire Configuration ODSA Operation

## Examples of Companion ODSA Responses

### EligibilityCheck Response Example

Table 35 presents an example for the EligibilityCheck response to a Companion ODSA application.

|  |
| --- |
| <?xml version="1.0"?>  <wap-provisioningdoc version="1.1">  <characteristic type="**VERS**">  <parm name="**version**" value="1"/>  <parm name="**validity**" value="172800"/>  </characteristic>  <characteristic type="**TOKEN**">  <parm name="**token**" value="ASH127AHHA88SF"/>  </characteristic>  <characteristic type="**APPLICATION**">  <parm name="**AppID**" value="ap2006"/>  <parm name="**CompanionAppEligibility**" value="1"/>  <parm name="**CompanionDeviceServices**" value="SharedNumber"/>  <parm name="**NotEnabledURL**" value="/www.MNO.org/AppNotAllowed"/>  <parm name="**NotEnabledUserData**" value="msisdn=XX&amp;device\_id=XX"/>  <parm name="**OperationResult**" value="1"/>  </characteristic>  </wap-provisioningdoc> |

1. : Example of a CheckEligibility ODSA Response in XML format

Table 36 presents an example for the EligibilityCheck response to a Companion ODSA application in JSON format.

|  |
| --- |
| {  "Vers" : {  "**version**" : "1",  "**validity**" : "172800"  }, "Token" : { // Optional  "**token**" : "ASH127AHHA88SF"  }, "ap2006" : { // ODSA for Companion Device app  "**CompanionAppEligibility**" : "1",  "**CompanionDeviceServices**" : "SharedNumber",  "**NotEnabledURL**" : "/www.MNO.org/AppNotAllowed",  "**NotEnabledUserData**" : "msisdn=XX&amp;device\_id=XX",  "**OperationResult**" : "1"  }  } |

1. : Example of a CheckEligibility ODSA Response in JSON format

### ManageService Response Example

Table 37 presents an example for the ManageService response to a Companion ODSA application.

|  |
| --- |
| <?xml version="1.0"?>  <wap-provisioningdoc version="1.1">  <characteristic type="**VERS**">  <parm name="**version**" value="1"/>  <parm name="**validity**" value="172800"/>  </characteristic>  <characteristic type="**TOKEN**">  <parm name="**token**" value="ASH127AHHA88SF"/>  </characteristic>  <characteristic type="**APPLICATION**">  <parm name="**AppID**" value="ap2006"/>  <parm name="**ServiceStatus**" value="3"/>  <parm name="**OperationResult**" value="1"/>  </characteristic>  </wap-provisioningdoc> |

1. : Example of a ManageService ODSA Response

Table 38 presents an example for the ManageService response to a Companion ODSA application in JSON format.

|  |
| --- |
| {  "Vers" : {  "**version**" : "1",  "**validity**" : "172800"  }, "Token" : { // Optional  "**token**" : "ASH127AHHA88SF"  }, "ap2006" : { // ODSA for Companion Device app  "**ServiceStatus**" : "3",  "**OperationResult**" : "1"  }  } |

1. : Example of a ManageService ODSA Response in JSON format

### ManageSubscription Response Example

Table 39 presents an example for the ManageSubscription response in XML format to a Companion ODSA application. This response indicates that the end-user is to be sent to an ODSA portal web server.

|  |
| --- |
| <?xml version="1.0"?>  <wap-provisioningdoc version="1.1">  <characteristic type="**VERS**"  <parm name="**version**" value="1"/>  <parm name="**validity**" value="172800"/>  </characteristic>  <characteristic type="**TOKEN**">  <parm name="**token**" value="ASH127AHHA88SF"/>  </characteristic>  <characteristic type="**APPLICATION**">  <parm name="**AppID**" value="ap2006"/>  <parm name="**SubscriptionServiceURL**" value="http://www.MNO.org/CDSubs"/>  <parm name="**SubscriptionServiceUserData**" value="imsi=XX&amp;msisdn=XX"/>  <parm name="**SubscriptionResult**" value="1"/> <!-- continue to websheet -->  <parm name="**OperationResult**" value="1"/>  </characteristic>  </wap-provisioningdoc> |

1. : Example of a ManageSubscription ODSA Response in XML format to send user to ODSA portal

Table 40 presents an example for the ManageSubscription response in XML format to a Companion ODSA application. This response provides information on the profile to download.

|  |
| --- |
| <?xml version="1.0"?>  <wap-provisioningdoc version="1.1">  <characteristic type="**VERS**"  <parm name="**version**" value="1"/>  <parm name="**validity**" value="172800"/>  </characteristic>  <characteristic type="**TOKEN**">  <parm name="**token**" value="ASH127AHHA88SF"/>  </characteristic>  <characteristic type="**APPLICATION**">  <parm name="**AppID**" value="ap2006"/>  <characteristic type="**DownloadInfo**">  <parm name="**ProfileSmdpAddress**" value="SMDP+ ADDR"/>  <parm name="**ProfileActivationCode**" value="COMM PROFILE CODE"/>  </characteristic>  <parm name="**SubscriptionResult**" value="2"/> <!—download profile -->  <parm name="**OperationResult**" value="1"/>  </characteristic>  </wap-provisioningdoc> |

1. : Example of a ManageSubscription ODSA Response in XML format with profile download information

Table 41 presents an example for the ManageSubscription response in JSON format to a Companion ODSA application. This response indicates that the end-user is to be sent to an ODSA portal web server.

|  |
| --- |
| {  "Vers" : {  "**version**" : "1",  "**validity**" : "172800"  }, "Token" : { // Optional  "**token**" : "ASH127AHHA88SF"  }, "ap2006" : { // ODSA for Companion Device app  "**SubscriptionServiceURL**" : "http://www.MNO.org/CDSubs",  "**SubscriptionServiceUserData**" : "imsi=XX&amp;msisdn=XX", "**SubscriptionResult**" : "1", // continue to websheet  "**OperationResult**" : "1"  }  } |

1. : Example of a ManageSubscription ODSA Response in JSON format to send user to ODSA portal

Table 42 presents an example for the ManageSubscription response in JSON format to a Companion ODSA application. This response provides information on the profile to download.

|  |
| --- |
| {  "Vers" : {  "**version**" : "1",  "**validity**" : "172800"  }, "Token" : { // Optional  "**token**" : "ASH127AHHA88SF"  }, "ap2006" : { // ODSA for Companion Device app  "**DownloadInfo**" : {  "**SubscriptionServiceURL**" : "SMDP+ ADDR",  " **ProfileActivationCode**" : "COMM PROFILE CODE"  },  "**SubscriptionResult**" : "2", // download profile  "**OperationResult**" : "1"  }  } |

1. : Example of a ManageSubscription ODSA Response in JSON format with profile download information

### AcquireConfiguration Response Example

Table 43 presents an example for the AcquireConfiguration operation in XML format for a Companion ODSA application.

|  |
| --- |
| <?xml version="1.0"?>  <wap-provisioningdoc version="1.1">  <characteristic type="**VERS**">  <parm name="**version**" value="1"/>  <parm name="**validity**" value="172800"/>  </characteristic>  <characteristic type="**TOKEN**">  <parm name="**token**" value="ASH127AHHA88SF"/>  </characteristic>  <characteristic type="**APPLICATION**">  <parm name="**AppID**" value="ap2006"/>  <characteristic type="**CompanionConfigurations**">  <characteristic type="**CompanionConfiguration**">  <parm name="**ICCID**" value="8991101200003204510"/>  <parm name="**CompanionDeviceService**" value="SharedNumber"/>  <parm name="**ServiceStatus**" value="1"/>  </characteristic>  </characteristic>  <parm name="**OperationResult**" value="1"/>  </characteristic>  </wap-provisioningdoc> |

1. : Example of an AcquireConfiguration ODSA Response in XML format

Table 44 presents an example for the AcquireConfiguration operation in JSON format for a Companion ODSA application.

|  |
| --- |
| {  "Vers" : {  "**version**" : "1",  "**validity**" : "172800"  }, "Token" : { // Optional  "**token**" : "ASH127AHHA88SF"  }, "ap2006" : { // ODSA for Companion Device app  "**CompanionConfigurations**" : [  "**CompanionConfiguration**" : {  "**ICCID**" : "8991101200003204510",  "**CompanionDeviceService**" : "SharedNumber",  "**ServiceStatus**" : "1"  }  ],  "**OperationResult**" : "1"  }  } |

1. : Example of an AcquireConfiguration ODSA Response in JSON format

## Companion ODSA Procedure Call Flows

The following sections present a number of informational call flows for the different user experiences and use cases of the Companion ODSA procedure. The ODSA application on the primary device is invoked at the request of the end-user and should capture proper user consent in order to have access to the companion device.

The exchanges between the Entitlement Configuration Server (ECS) (aka ODSA Device Gateway) and the Service Provider’s (SP) back-end systems are shown for informational purposes only. This applies as well for the exchanges that involve the ODSA Portal Web Server.

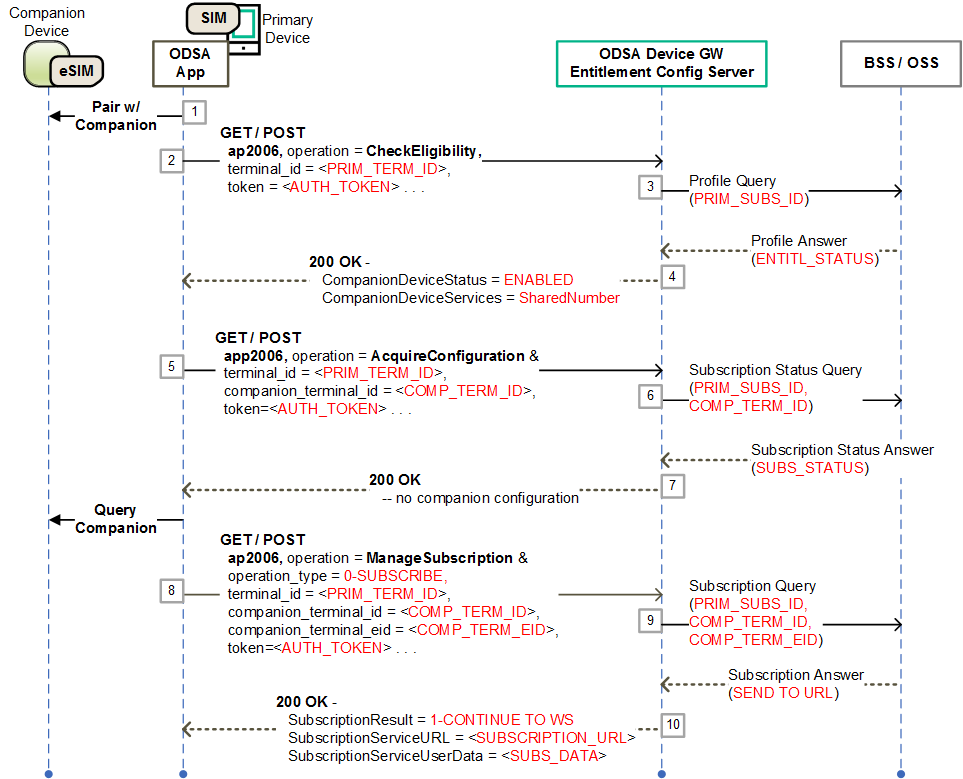
### Subscription Activation via ODSA Portal – Initial Steps

The following presents the case where:

* The companion ODSA application is allowed for the type of primary device and enabled for the end-user (entitled);
* the companion device does not have an active subscription and communication profile from the Service Provider;
* the SP's ODSA portal web server is responsible for completing the subscription activation for the companion device

Figure 7 shows the initial steps of the flow involving the SP's ODSA portal, where the Companion ODSA application acquires proper entitlement and subscription data from the SP's ECS. The steps are:

1. End-user invokes the Companion ODSA application on the primary device which connects with the companion device to initiate the ODSA procedure (over a protocol outside the scope of this specification).
2. The companion ODSA application makes a **CheckEligibility** request to the ECS.
3. The ECS queries the SP's back-end system managing the end-user’s entitlements and services.
4. The ECS processes the answer from the SP's back-end system and generates the proper 200 OK response containing CompanionDevice set to ENABLED and allowed services in the CompanionDeviceServices field set to SharedNumber.
5. Since the CompanionDevice value is correct and target service is allowed, the companion ODSA application sends an **AcquireConfiguration** request to the ECS to obtain information on any communication profiles associated with the companion device.
6. The ECS queries the SP's back-end system managing the subscriptions and active profiles.
7. The ECS processes the response from the SP's back-end system and generates the proper 200 OK response containing CompanionDeviceConfigurations without any CompanionConfiguration (no profile/subscription is associated with the companion device).
8. The companion ODSA application makes a **ManageSubscription** request to the ECS with an operation\_type set to SUBSCRIBE (value of 0) to initiate the subscription procedure for the companion device.
9. The ECS queries the SP's back-end system to determine the next step and method to use for the companion device's subscription request.
10. The ECS processes the response from the SP's back-end system and generates the proper 200 OK response to send the application and end-user to the SP's ODSA portal. The response contains a SubscriptionResult set to CONTINUE\_TO\_WS (value of 1), and SubscriptionServiceURL along with SubscriptionServiceUserData presenting the URL of the ODSA Portal web server and any user-specific data that would be useful to the Portal.



1. – Initial steps for companion ODSA procedure involving ODSA portal

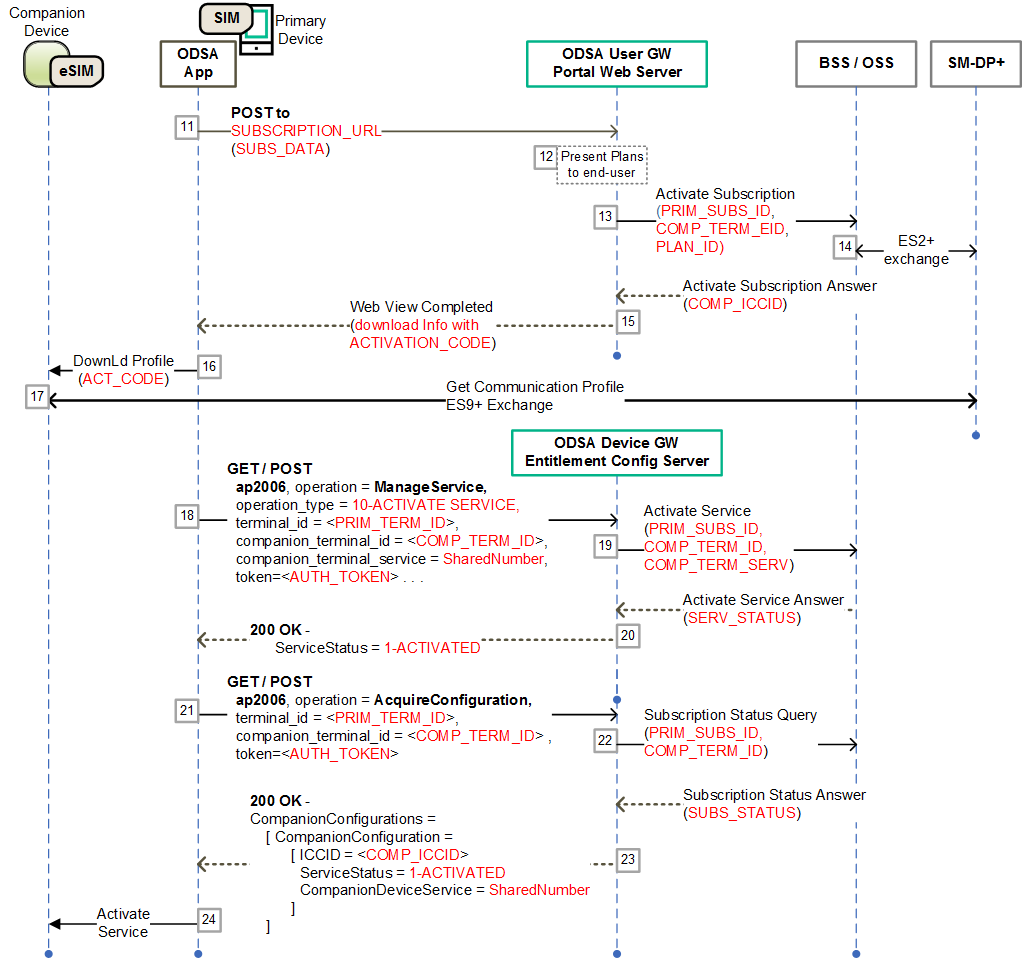
### ODSA Portal with Immediate Download Info – Final Steps

The following presents the case where:

* The companion ODSA application was already informed to use the SP's ODSA portal to complete the subscription procedure (refer to 6.6.1);
* The ODSA portal is able to generate the communication profile download information as a result of the exchanges with the end-user

Figure 8 shows the final steps of the Companion ODSA procedure in the case where the ODSA portal provides the profile download information back to the application (immediate delivery). The steps are:

1. The ODSA application connects with the ODSA portal web server using the URL provided in the **ManageSubscription** operation, allowing the web pages from the portal to be displayed to the end-user
2. The ODSA portal web server presents a set of plan offers to the end-user and captures the selection from the end-user
3. The ODSA portal makes a request towards the SP's back-end system to activate the selected plan and subscription
4. The SP's back-end system interacts with the SM-DP+ over the ES2+ interface to make the required profile requests associated with the new subscription (for example, DownloadOrder, ConfirmOrder and ReleaseProfile) resulting in an activation code and ICCID for the companion device
5. The ODSA portal provides the communication profile download information (activation code) to the ODSA application using a JavaScript call back function
6. The ODSA application informs the companion device to download the profile
7. The companion device downloads the communication profile from the SM-DP+
8. Optional - The ODSA application makes a **ManageService** request to the ECS with an operation\_type set to ACTIVATE SERVICE (value of 10) to have the network activate and provision the NumberShare service on the companion device
9. The ECS makes the appropriate requests to the SP's back-end system for service activation on the companion device's subscription
10. The SP's back-end system replies back with service status and the ECS generates the proper response with service status to the ODSA application.
11. The ODSA application makes an **AcquireConfiguration** request to the ECS to verify that the subscription and service for the companion device are in the proper states
12. The ECS queries the SP's back-end system managing the subscriptions and profiles
13. The ECS processes the response from the SP's back-end system and generates the proper 200 OK response containing CompanionDeviceConfigurations with a CompanionDeviceConfiguration entry for the newly active subscription bearing the ACTIVATED status (value of 1).
14. As the companion device’s subscription and service are in the right states, the ODSA application informs the companion device to initiate cellular service



1. – Final steps for companion ODSA procedure with profile download info from ODSA portal

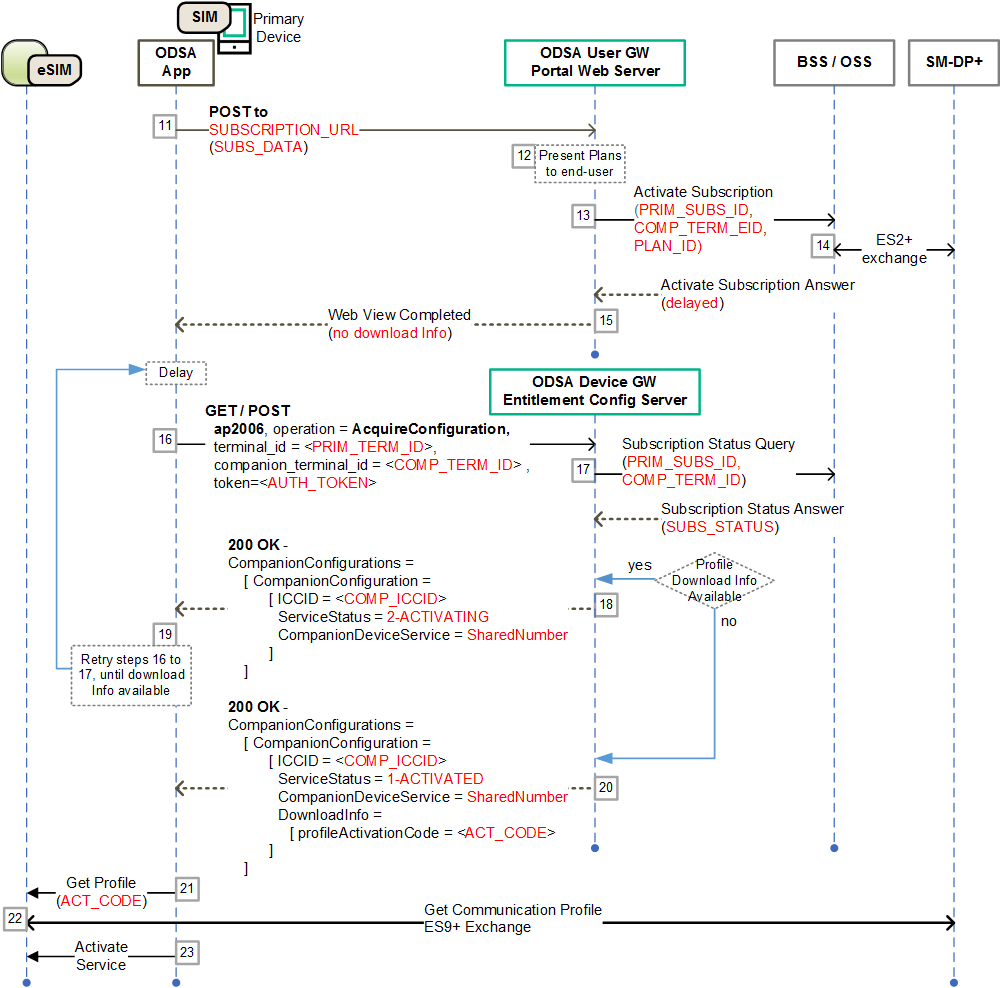
### ODSA Portal with Delayed Download Info – Final Steps

The following presents the case where:

* The companion ODSA application was already informed to use the SP's ODSA portal to complete the subscription procedure (refer to 6.6.1);
* The ODSA portal interacts with the end-user for plan selection and subscription activation but does not return the profile download information to the application
* The companion ODSA application subsequently obtains the profile information by querying the ECS

Figure 9 shows the final steps of the Companion ODSA procedure in the case where the profile download information is obtained by the application after the end-user interastions with the ODSA portal (delayed delivery). The steps are:

1. The ODSA application connects with the ODSA portal web server using the URL provided in the **ManageSubscription** operation, allowing the web pages from the portal to be displayed to the end-user
2. The ODSA portal web server presents a set of plan offers to the end-user and captures the selection from the end-user
3. The ODSA portal makes a request towards the SP's back-end system to activate the selected plan and subscription
4. The SP's back-end system interacts with the SM-DP+ over the ES2+ interface to make the required profile requests associated with the new subscription (for example, DownloadOrder, ConfirmOrder and ReleaseProfile), and indicates to the ODSA portal that the final response with the download info is delayed (asynchronous)
5. The ODSA portal indicates to the ODSA application the end of the end-user flow via a JavaScript call back function without providing the profile download information (activation code)
6. After a delay, the ODSA application makes an **AcquireConfiguration** request to the ECS to verify that the subscription and service for the companion device are in the proper states
7. The ECS queries the SP's back-end system managing the subscriptions and profiles.  
   If the subscription is not yet ready and profile info is not yet available, go to step 18.  
   If the subscription is ready, as well as profile download info, go to step 20
8. The ECS generates a 200 OK response with a CompanionDeviceConfiguration entry bearing the ACTIVATING status (value of 2).
9. The ODSA application repeats the **AcquireConfiguration,** going to step 16
10. The ECS generates a 200 OK response with a CompanionDeviceConfiguration entry for the newly active subscription bearing the ACTIVATED status (value of 1) and a filled in DownloadInfo structure.
11. As the companion device’s subscription and service are in the right states, the ODSA application informs the companion device to download the profile
12. The companion device downloads the communication profile from the SM-DP+
13. The ODSA application informs the companion device to initiate cellular service



1. – Final steps for companion ODSA procedure with ODSA portal and delayed profile download info

### Subscription Activation without ODSA Portal

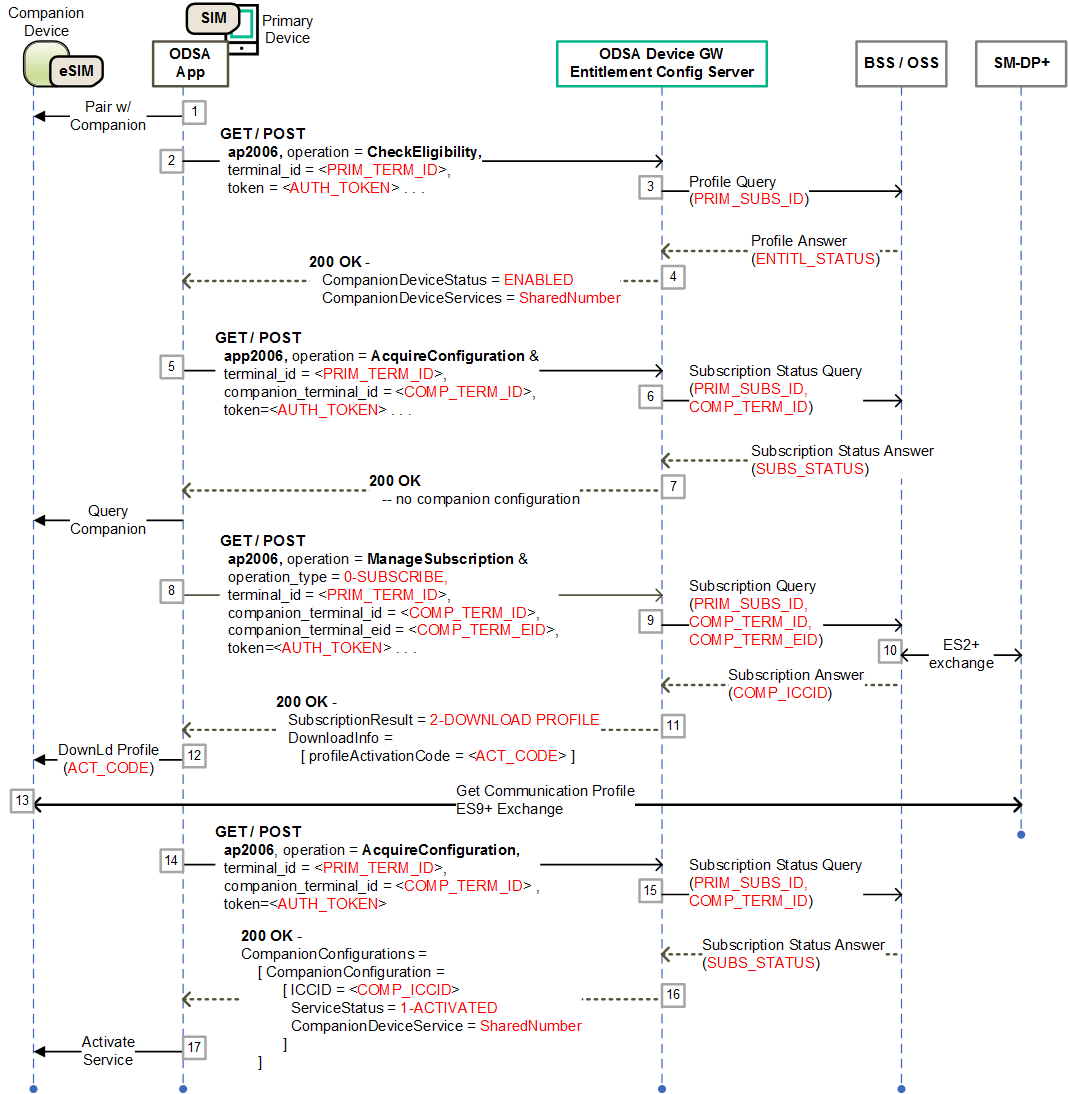
The following presents the case where:

* The companion ODSA application is allowed for the type of primary device and enabled for the end-user (entitled);
* the companion device does not have an active subscription and communication profile from the Service Provider;
* the SP is able to activate a subscription and create a communication profile without involving the ODSA portal web server

Figure 10 presents a call flow where the profile download information for the companion device is made available by the SP at the time of the ManageSubscription request. There is no need to send the end-user to an ODSA portal web server.

The steps 1 to 8 are the same as in 6.6.1:

1. The ECS queries the SP's back-end system to determine the next step and method to use for the companion device's subscription request (no need for ODSA portal)
2. The SP's back-end system interacts with the SM-DP+ over the ES2+ interface to make the required profile requests associated with the new subscription (for example, DownloadOrder, ConfirmOrder and ReleaseProfile) resulting in an activation code and ICCID for the companion device returned to the ECS
3. The ECS processes the response from the SP's back-end system and generates the proper **ManageSubscription** 200 OK response with a SubscriptionResult set to DOWNLOAD\_PROFILE (value of 2), and a filled in DownloadInfo structure.
4. The ODSA application informs the companion device to download the profile
5. The companion device downloads the communication profile from the SM-DP+
6. The ODSA application makes an **AcquireConfiguration** request to the ECS to verify that the subscription and service for the companion device are in the proper states
7. The ECS queries the SP's back-end system managing the subscriptions and profiles
8. The ECS processes the response from the SP's back-end system and generates the proper 200 OK response containing CompanionDeviceConfigurations with a CompanionDeviceConfiguration entry for the newly active subscription bearing the ACTIVATED status (value of 1).
9. The ODSA application informs the companion device to initiate cellular service



1. – Call flow for Companion ODSA procedure without ODSA Portal

### Subscription Pre-activation via Another Channel

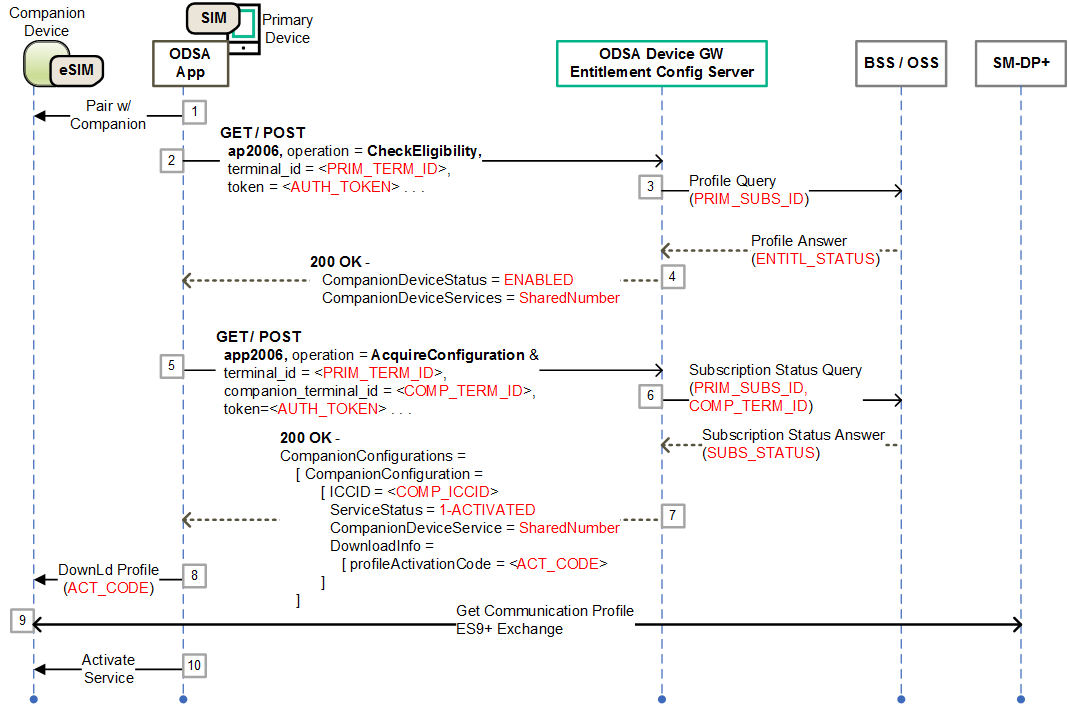
The following presents the case where:

* The companion ODSA application is allowed for the type of primary device and enabled for the end-user (entitled);
* the companion device has an active subscription and communication profile from the Service Provider, created beforehand through another channel (for example point of sale or call to a SP's representative)

Figure 10 presents a call flow where the profile download information for the companion device is made available by the SP at the time of the ManageSubscription request. There is no need to send the end-user to an ODSA portal web server.

The steps 1 to 4 are the same as in 6.6.1:

1. The ODSA application makes an **AcquireConfiguration** request to the ECS to verify that the subscription and service for the companion device are in the proper states
2. The ECS queries the SP's back-end system managing the subscriptions and profiles, which shows that the companion device already has a subscription and associated communication profile
3. The ECS processes the response from the SP's back-end system and generates the proper 200 OK response containing CompanionDeviceConfigurations with a CompanionDeviceConfiguration entry for the newly active subscription bearing the ACTIVATED status (value of 1) and a filled in DownloadInfo structure.
4. The ODSA application informs the companion device to download the profile
5. The companion device downloads the communication profile from the SM-DP+
6. The ODSA application informs the companion device to initiate cellular service



1. Call flow for Companion ODSA procedure with pre-activated subscription

## ODSA Application Considerations Around Web View Callback

During the procedure for ODSA on Companion devices, end-users can be presented with a set of web views specific to the Operator. The web views are hosted by an Operator portal web server as shown in Figure 6.

To support proper communication between web views and the ODSA application, the application should support JS callbacks to allow for the portal to share the following events and corresponding data elements:

* + **Event**: Communication Profile Ready for Download, do not finish web flow  
    **Data**: download method and corresponding download info parameter(s), for example Activation Code or SM-DP+ URL, see Table 32 for details
  + **Event**: Finish the web flow, and follow with the retrieval of the status of the communication profile and associated service via an AcquireConfiguration request
  + **Event**: Dismiss the web flow, without communication profile download

1. Document Management
   1. Document History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | Brief Description of Change | Approval Authority | Editor / Company |
| V1.0 | July 2018 | First version | TG#11 | J. Sicard / HPE |
| V2.0 | October 2018 | Updated with changes detailed in CR1002 | TSG | J. Sicard / HPE |
| V3.0 | May 2019 | Added the Companion On-Device Service Activation (ODSA) Use Case | TSG | J. Sicard / HPE |
| V3.1 | June 2019 | Added:   * Clarifications on notifications to clients, * New app\_name and app\_version parameters, * OpenID authentication flow, * JSON-based configuration document, * Description of OperationResult * Clarifications on ODSA response parameters, * Updated call flow in Section 6.6 for ODSA |  |  |
| V3.2 | August 2019 | Corrections and clarifications for the Companion ODSA use case and OpenID Authentication | TSG | J. Sicard / HPE |
| V3.3 | September 2019  (Draft) | Included CR1004 for the VoWiFi callback functions definition (section 3.4).  Added section 2.3 on allowing POST method in addition to GET method. Updated Figure 6 around the ODSA architecture. | TSG | J. Sicard / HPE |
| October 2019 (Draft B) | Inserted statements in Section 6.1 from eSIM WG's SGP.21 / SGP.22 alignment proposal.  Removed IMSI from example requests. Used proper straight double quotes in the request and response examples. | TSG | J. Sicard / HPE |
| V3.4 | November 2019 (Draft) | Adjusted call flows for Companion ODSA (clause 6.6) and inserted detailed step-by-step descriptions and new call flows (Clauses 6.6.3 to 6.6.5). Added new clause 6.7 on Callback functions for Companion ODSA | TSG | J. Sicard / HPE |

Other Information

|  |  |
| --- | --- |
| Type | Description |
| Document Owner | Terminal Steering Group (TSG) |
| Editor / Company | Jerome Sicard / HPE |

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](mailto:prd@gsma.com)

Your comments or suggestions & questions are always welcome.