IDY.54 Mobile Connect Verified MSISDN Definition and Technical Requirements

Version 1.0

06 December 2022

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

Security Classification: Non-confidential

Access to and distribution of this document is restricted to the persons permitted by the security classification. This document is 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 © 2022 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

1.1 Overview

Mobile Connect is a worldwide initiative by Mobile Operators to bring a wide portfolio of Identity services to market that enable SPs and End-Users to transact with one-another more securely through strong authentication, authorisation and exchange of attributes, subject to User consent.

The Mobile Connect architecture consists of a Core framework around which additional components can be added to support the different Mobile Connect services. The Core framework is based upon the OpenID Connect (OIDC) protocol suite [1] and allows Users to be identified by their MSISDN (or a related Pseudonymous Customer Reference) to enable authentication on their mobile device.

The serving Mobile Operator selects an appropriate Authenticator based on Mobile Operator policy, device capability and the Level of Assurance required by the SP to enable authentication.

This document details the Mobile Connect Verified MSISDN service. Mobile Connect Verified MSISDN allows SPs to verify the phone number of the device connected to the mobile data network through which a User is accessing a service provided by a SP.

Mobile Connect Verified MSISDN is defined as two service variants:

- Verified MSISDN Match: in which the Mobile Operator compares the MSISDN associated with the mobile device against that provided by the SP in the service request
- Verified MSISDN Share: in which the Mobile Operator provides the mobile device MSISDN to the SP who can then perform the check itself

This document includes a description of the Mobile Connect Verified MSISDN service, applicable use cases and the associated User experience. It also contains normative sections specifying how the service must be implemented and operated (in conjunction with requirements for the Core framework and the Resource Server).

For further information on the Mobile Connect framework please see Mobile Connect Technical Architecture and Core Requirements [7].

1.2 Scope of the document

In Scope	Out of Scope
• Mobile Connect Verified MSISDN functionality description • Mobile Connect Verified MSISDN technical specifications	• Detailed Privacy and Trust Principles • UI/UX guidelines • Mobile Connect Verified MSISDN commercial propositions • Service provider/developer implementation guidelines • Other Mobile Connect service definitions

1.3 Abbreviations

Term	Description
API	Application Programming Interface
CIBA	Client Initiated Backchannel Access
HTTP	Hyper Text Transport Protocol
ID GW	Identity Gateway
JSON	JavaScript Object Notation
KYC	Know Your Customer
LoA	Level of Assurance
MSISDN	Mobile Station International Subscriber Directory Number
OIDC	OpenID Connect
OIDF	OpenID Foundation
OTP	One Time Password
PCR	Pseudonymous Customer Reference
RCS	Rich Communication Service
RFC	Request For Comments
SIM	Subscriber Identity Module
SMS	Short Messaging Service
SP	Service Provider
UI/UX	User Interface/User Experience
URL	Uniform Resource Locator

1.4 Audience

The target audience for this document are the service managers and service/technical departments at Mobile Operators who are considering deploying the Mobile Connect Verified MSISDN service.

Readers of this document are expected to have familiarity with the Mobile Connect and some knowledge of the technical architecture and Mobile Connect Core framework technical requirements.

1.5 Relationship to Other Mobile Connect Documentation

This document details the Mobile Connect Verified MSISDN service and its usage including the relevant technical requirements (building on the Mobile Connect Core framework) and the relevant technical parameters, such as “scope” value and any service specific error codes.

The Mobile Connect Technical Overview document [6] provides a high-level description of Mobile Connect and how it works. It also includes a master list of abbreviations and terminology used within the Mobile Connect Documentation set and a map of that documentation set. It serves as a starting point for understanding how Mobile Connect works and also references the relevant documents for the reader to obtain further detail.

The Mobile Connect Technical Architecture and Core Requirements document [7] describes the Mobile Connect Architecture in more detail and also includes the core technical requirements and specification of elements for Mobile Connect that are generic to all Mobile Connect services and modes of operation.

Detailed specifications for the Mobile Connect APIs (Device-Initiated Mode [8] and Server-Initiated Mode [9]) provide details for OIDC Authorization Requests and Responses, and Token Requests (DI Mode) and Responses including examples and error codes.

The Mobile Connect Resource Server Specification [10] provides details on how to handle a Resource request and the associated response for Mobile Connect Attribute services including error codes where this approach is used by a Mobile Connect service. Service specific error codes are included within the relevant service definition and technical requirements document.

1.6 Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [4].

1.7 Terminology & Definitions

Mobile Connect specifications and related documents make use of terms that are defined by the OpenID Connect Core Specification [2] and supporting specifications and extended in the OIDF CIBA (Client Initiated Backchannel Authentication Flow) [3].

The Mobile Connect Technical Overview document [6] defines relevant terms that are used within the Mobile Connect Specifications and interprets terminology from source standards in Mobile Connect terms.

Due to potential confusion with OIDC and OAuth 2.0 terminology, the initial Mobile Connect service request which underpins Mobile Connect Authentication, Authorisation and User consent associated with attribute services, is referred to as an OIDC Authorization Request[1] (spelled with a ‘z’) throughout this document.

1.8 References

Ref	Doc Number	Title
[1]	Open ID Connect	http://openid.net/connect/
[2]	OpenID Connect Core Specification	“An interoperable authentication protocol based on the OAuth 2.0 family of specifications” available at https://openid.net/specs/openid-connect-core-1_0.html

[1] In OAuth2.0 the initial request is referred to as an “Authorization Request”, whereas in OIDC it is referred to as an “Authentication Request”. Mobile Connect offers several services including Mobile Connect Authentication and Mobile Connect Authorisation, hence Mobile Connect specifications have adopted the term “OIDC Authorization Request” to describe this initial service request in the protocol flow.

[3]	OIDF CIBA	OpenID Connect MODRNA Client Initiated Backchannel Authentication Flow 1.0 https://openid.net/specs/openid-connect-modrna-client-initiated-backchannel-authentication-1_0.html
[4]	RFC 2119	“Keywords for use in RFCs to Indicate Requirement Levels”, S. Bradner, March 1997. Available at https://tools.ietf.org/html/rfc2119
[5]	E.164	ITU-T E.164 The international public telecommunication numbering plan https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-E.164-201011-I!!PDF-E&type=items
[6]	IDY.05	Mobile Connect Technical Overview
[7]	IDY.51	Mobile Connect Technical Architecture and Core Requirements
[8]	IDY.01	Mobile Connect Device-Initiated OIDC Profile
[9]	IDY.02	Mobile Connect Server-Initiated OIDC Profile
[10]	IDY.03	Mobile Connect Resource Server Specification
[11]	IDY.16	Mobile Connect Product Manager’s Lifecycle Handbook

2 Mobile Connect Verified MSISDN

Mobile Connect Verified MSISDN allows SPs to verify the MSISDN (phone number) of the device connected to the mobile data network through which a User is accessing an SP service. In doing so, it enables the SP to check that the device being used to access a particular SP User account belongs to the account holder. Such a service can also be used by theSP for verifying the User’s device as a complementary service when a User is being authenticated.

Mobile Connect Verified MSISDN is defined as two service variants:

- Verified MSISDN Match: in which the Mobile Operator compares the MSISDN associated with the mobile device against that provided by the SP in the service request[2]. The MSISDN can be supplied in an E164 format [5] or in a hashed form. Verified MSISDN Match ensures no data is shared by the Mobile Operator.
- Verified MSISDN Share: in which the Mobile Operator provides the mobile device MSISDN to the SP who can then perform the check itself

Note that the Verified MSISDN service only works for devices which have an active mobile data bearer. It will not work via another data connection such as WiFi.

2.1 Use Case Examples

Mobile Connect Verified MSISDN supports a range of practical use cases including:

1. Android Account Setup
2. Mobile Banking App installation and password-less login
3. Android password-less login
4. RCS client installation / configuration
5. Posting online ad or reviews on a website
6. Verified MSISDN service as the second factor authentication

Some examples are listed below in more detail to explain the use of Verified MSISDN.

2.1.1 Android device upgrade (Google)

Verified MSISDN service could be used as an additional authentication step to provide more confidence to Google when a User first logs in on a new device with their Google credentials. When upgrading to a new/different Android phone, the User needs to enter their Google credentials to personalize the device - at this point, Google need additional confirmation that the individual entering the credentials is the legitimate account owner and not an attacker.

[2] The SP submits the MSISDN value to be matched via the Resource Request rather than the initial OIDC Authorization Request.

Google typically solve this issue through the use of SMS OTP[3] but would prefer to use an Mobile Operator API such as Verified MSISDN to verify the association of device to the User without creating friction within the User experience when upgrading to a new Android phone.

2.1.2 Password-less login

A separate but related problem is that it is quite common for a User to forget their Google credentials altogether and need to establish a new set (and new identity) in order to activate their Android device - this creates friction for the User, but also presents a service continuity issue for Google.

An alternative approach for verifying a new Android device would be for Google to adopt the User's phone number as the username[4], and possession/control of the Android device associated with that phone number as an implicit factor of authentication. This approach is simple and can then be combined with a lightweight knowledge factor (e.g., related to their Google or mobile account) to sign the User in and mitigate SIM swap fraud[5].

2.1.3 Mobile Banking App & Apple/Android Pay installation

Similar to the Android upgrade use case, whenever a User wishes to install and/or configure a new SP app on their device, it is important that the SP can check that the device on which the app is being installed is associated with the User in question to mitigate against an attacker gaining access to a User's account by entering the User's credentials (username/password) via their own device. Mobile banking is a good example of this, the bank needing to ensure that the device upon which a mobile banking app is being installed is associated with the same phone number as on file for the bank account for that User.

A similar use case is where a User requests their debit/credit card to be tokenised and installed on their mobile phone (e.g., for use within Apple Pay, Samsung Pay, Android Pay etc.) and it is important that the issuing bank can verify that the User making the request is the same as the cardholder – by verifying that the phone number associated with the device upon which the debit/credit card will be installed is the same as the phone number on file for the cardholder (and matching other attributes such as name & address registered against that phone number[6]), the issuing bank can mitigate against fraud[7].

[3] Note that the approach of issuing an SMS OTP is really gaining User confirmation for the action rather than verifying the underlying device as the OTP could still be returned if the MSISDN on file at Google pointed to a different device

[4] the MSISDN makes a good identifier given that it is typically retained whenever the User changes their device or their Mobile Operator

[5] e.g., where the User's SIM card has been stolen and inserted in an attacker's phone

[6] e.g., using the Mobile Connect KYC Match service

[7] Note that whilst this mitigates against an attacker trying to access the User's bank account on the attacker's phone (e.g., where account credentials were obtained through phishing or a data breach) it does not address the scenario of a rogue app acting on the User's phone itself hence an additional step of User confirmation or app attestation (provided through the underlying app OS/store) is ideally needed.

3 Verified MSISDN Functional Description

3.1 Verified MSISDN Service Flow

The SP must initiate the Verified MSISDN request via the User's mobile device which is directly connected to the mobile data network. The Mobile Operator ID GW must support HTTP Header enrichment or a similar mechanism to be able to extract the User's MSISDN from the mobile data network. Verified MSIDN will not work if the device is on Wi-Fi or on a mobile hotspot.

SPs must pre-register for the Mobile Connect Verified MSISDN service (and, in particular, whichever service variant they intend to use, based on what is supported by the Mobile Operator ID GW).

Because Mobile Connect Verified MSISDN requires the use of the mobile data network it may be that the SP app includes a feature to allow switching the data bearer to ensure that the mobile data network is used.

Figure 1 shows a high-level flow for a Verified MSISDN service request. The use case shown is one of registering a mobile app to allow password-less log-in.

- This User is installing an SP's mobile app on their device and in response to the prompt, the User enters their phone number.
- The SP educates the User about the need to perform number verification whenever the User attempts to login in the future. The SP provides further details such as revocation mechanism in a linked page (shown as "Learn more").
- The User is asked to provide long-lived consent to the SP. The use of data needs to be clearly stated in terms of the service and the privacy policy.

Mobile Connect Technical Architecture and Core Requirements [7] provides a more detailed sequence diagram illustrating the flow for Device-Initiated mode. The Mobile Connect Device-Initiated OIDC Profile [8] defines the API calls and responses for each mode.

The service flow is as follows:

- The SP's app initiates a Mobile Connect Verified MSISDN service request towards the Mobile Operator's ID GW Authorization endpoint using Mobile Connect Device-Initiated mode [8]. The SP specifies the service required using the `scope` parameter to indicate whether Verified MSISDN Share, Verified MSISDN Match (Plain) or Verified MSISDN Match (Hashed) is required.
  - In this example, it is assumed that the SP already has the relevant Mobile Operator ID GW metadata and credentials to be able to initiate a service request to the correct Mobile Operator ID GW for that User. If not available, then this can be obtained by using the API Exchange Discovery service or by obtaining the details directly from the Mobile Operator.
- The Mobile Operator's ID GW validates the request (i.e. that the SP has been registered with the Mobile Operator for the Mobile Connect Verified MSISDN service requested)

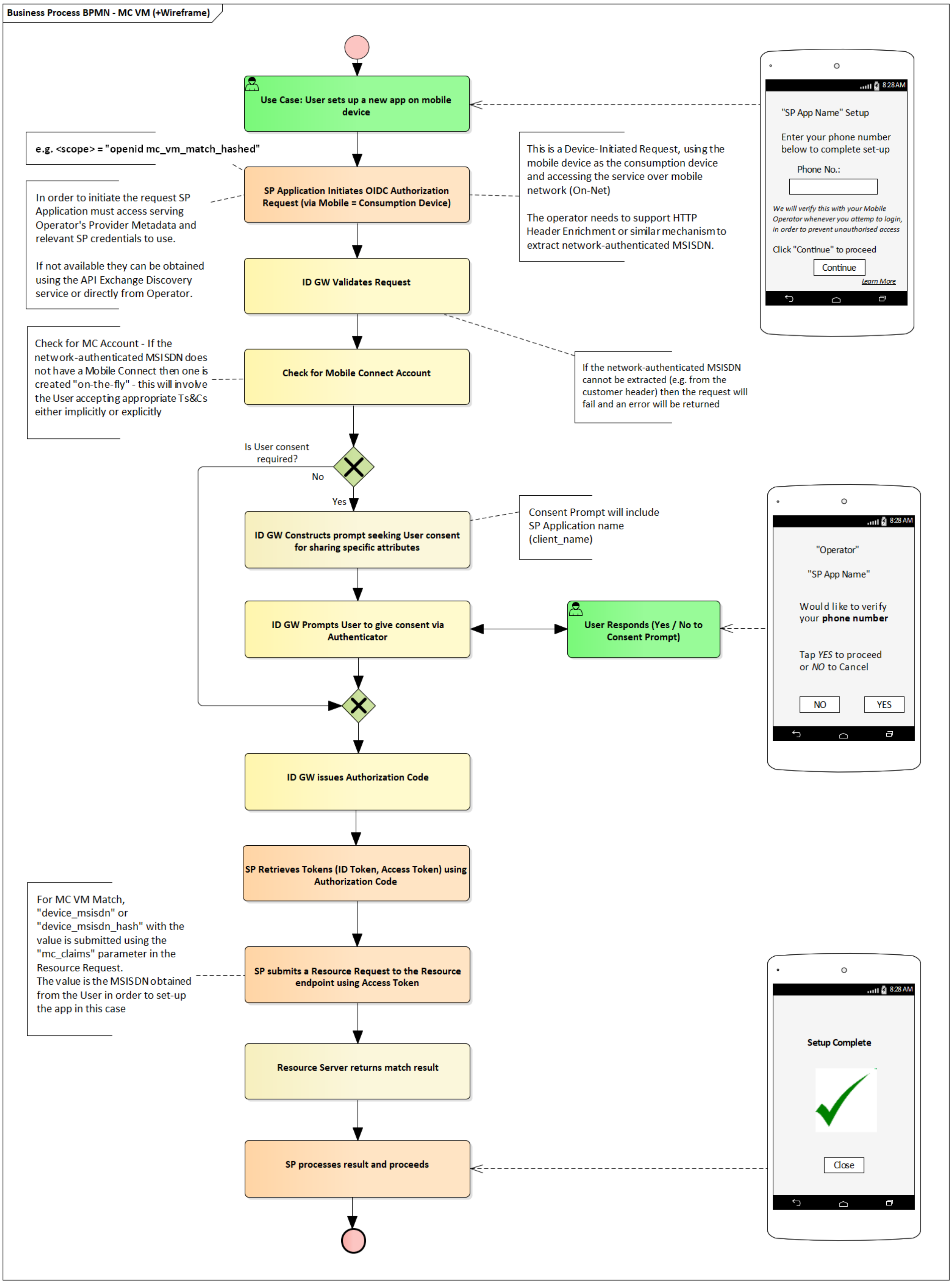
Business Process BPMN - MC VM (+Wireframe)
Use Case: User sets up a new app on mobile device
e.g. <scope> = "openid mc_vm_match_hashed"
SP Application Initiates OIDC Authorization Request (via Mobile = Consumption Device)
In order to initiate the request SP Application must access serving Operator's Provider Metadata and relevant SP credentials to use.
If not available they can be obtained using the API Exchange Discovery service or directly from Operator.
This is a Device-Initiated Request, using the mobile device as the consumption device and accessing the service over mobile network (On-Net)
The operator needs to support HTTP Header Enrichment or similar mechanism to extract network-authenticated MSISDN.
"SP App Name" Setup
Enter your phone number below to complete set-up
Phone No.:
We will verify this with your Mobile Operator whenever you attemp to login, in order to prevent unauthorised access
Click "Continue" to proceed
Continue
Learn More
8:28AM
ID GW Validates Request
Check for Mobile Connect Account
Check for MC Account - If the network-authenticated MSISDN does not have a Mobile Connect then one is created "on-the-fly" - this will involve the User accepting appropriate Ts&Cs either implicitly or explicitly
If the network-authenticated MSISDN cannot be extracted (e.g. from the customer header) then the request will fail and an error will be returned
Is User consent required?
No
Yes
ID GW Constructs prompt seeking User consent for sharing specific attributes
Consent Prompt will include SP Application name (client_name)
ID GW Prompts User to give consent via Authenticator
User Responds (Yes / No to Consent Prompt)
8:28 AM
"Operator"
"SP App Name"
Would like to verify your phone number
Tap YES to proceed or NO to Cancel
NO
YES
ID GW issues Authorization Code
SP Retrieves Tokens (ID Token, Access Token) using Authorization Code
For MC VM Match, "device_msisdn" or "device_msisdn_hash" with the value is submitted using the "mc_claims" parameter in the Resource Request.
The value is the MSISDN obtained from the User in order to set-up the app in this case
SP submits a Resource Request to the Resource endpoint using Access Token
Resource Server returns match result
SP processes result and proceeds
8:28 AM
Setup Complete
Close

Figure 1: Mobile Connect Verified MSISDN Service Flow

- The ID GW optionally can check that the source is from a known IP range (white listed)
- The Mobile Operator ID GW checks whether the User is registered for Mobile Connect
- If the MSISDN is not yet registered for Mobile Connect, based on ID GW policies a new Mobile Connect account may be created “on-the-fly” for that MSISDN.
- The Mobile Operator retrieves the MSISDN from the IP Header (or uses an alternate mechanism to obtain the network-authenticated MSISDN for that device)

  In order to be able to share (or match) User information with a SP, the User must give their consent.

  Figure 1 illustrates the option where the Mobile Operator ID GW captures User consent which, within the context of this use case, might be at initial registration for the SP service.

  User consent can also be captured by the SP, subject to the Mobile Operator’s ID GW consent policies and the contractual agreement with the SP.

  Consent can also be long-lived where it exists for an extended period of time to provide a smoother User experience. User consent is discussed in more detail in Section 3.4.

  If consent is not required (i.e. the User has already given their consent) then this step can be omitted.

- An Authorization Code is then returned to the SP as part of a standard Device-Initiated flow and the SP can then retrieve the ID Token and Access Token by make a Token Request to the ID GW Token endpoint.
- For Verified MSISDN Share the SP can then make a Resource Request to the Resource endpoint with the Access Token to fetch the MSISDN of the device.
- For Verified MSISDN Match the SP can fetch the MSISDN match result by making a Resource Request with the Access Token and including the appropriate attribute identifier and value within the `mc_claims` parameter:
  - The attribute identifier is “`device_msisdn`” with a value of the MSISDN in plain text if `scope=”mc_vm_match”` was specified in the service request.
  - The attribute identifier is “`device_msisdn_hash`” with a value of the hash of the MSISDN if `scope=”mc_vm_match_hash”` was specified in the service request.

- The Resource Server matches the attribute value with the previously extracted[8] MSISDN or hash of the MSISDN extracted by the Authorization Server[9]. If they match, the service returns a Boolean value “true” to the SP.
- If the verification fails, then it returns a Boolean value “false” to the SP.

3.2 Mobile Connect Account Setup

If the User is not already registered for Mobile Connect, a new User account will be generated – this may either be done implicitly, via notification to the User (e.g., when asking the User to consent to sharing or matching of their MSISDN) or explicitly as required. Account registration is discussed in more detail in the Mobile Connect Product Manager’s Lifecycle Handbook [11].

3.3 Overcoming the Mobile Data Access Point Limitation

The Verified MSISDN service requires the User to be on the mobile data network when a SPs app, for example, submits a Verified MSISDN service request to the Mobile Operator ID GW. This may create some friction for Users if they are not accessing the service via their mobile device in the mobile data network. The SP can manage the User experience by:

- Only requesting the Verified MSISDN service when the User is connected to the mobile data network. In many “mobile first” markets, this limitation is not a show stopper as the majority of internet consumption is on mobile data.
- Users are becoming aware of risks of hacking and are increasingly willing to take action to protect themselves. The SPs may present an option to the User to switch to mobile data for protection as mobile data access is inherently safer than Wi-Fi hotspots. Both Android and iOS allow switching data access point – Android can be done programmatically by the SP app whilst with iOS, the User must make this switch manually.
- The User will require a working mobile data connection and the service will not work, for example, if the User is a pre-paid customer who does not have enough credit. The Mobile Operator may consider white listing the SP’s URL to minimise this limitation and at the same time the SP may consider white listing the ID GW endpoints.

3.4 User Consent Management

Where a Mobile Connect service requires the sharing or matching of personal information relating to the User or their mobile subscription, the User’s consent must be obtained before any data is shared. Depending upon the Mobile Connect service or the specific use case within a service, User consent can be captured by the Mobile Operator’s ID GW or by the SP. The Mobile Connect Verified MSISDN service is designed to be used as a fraud check so typically, User consent will be captured by the SP (at registration and /or included within standard terms and

[8] How to acquire previously extracted MSISDN from AS is implementation specific and out of scope of this document. One way of doing is through introspection end point or directly accessing an intermediate storage area. Why is footnote 8 and 9 referenced twice ?

[9] How to acquire previously extracted MSISDN from AS is implementation specific and out of scope of this document. One way of doing is through introspection end point or directly accessing an intermediate storage area.

conditions) and the service will be processed in the background without any explicit User consent.

Further discussion on User Consent can be found in the Mobile Connect Product Manager's Lifecycle Handbook [11].

4 Verified MSISDN Service Specification

This Section is normative and contains the relevant information required by Mobile Operators to implement and support Mobile Connect Verified MSISDN services.

4.1 OIDC Authorization Request Parameters - scope

The SP requests Mobile Connect Verified MSISDN services via the scope parameter in the Mobile Connect OIDC Authorization Request as per Table 1.

Mobile Connect Verified MSISDN	OIDC Authorization Request scope Parameter[10]
Mobile Connect Verified MSISDN Share [plain]	"mc_vm_share"
Mobile Connect Verified MSISDN Match [plain]	"mc_vm_match"
Mobile Connect Verified MSISDN Match [hashed]	"mc_vm_match_hash"

Table 1: Mobile Connect Verified MSISDN scope Values

Note that the Level of Assurance required for Mobile Connect Verified MSISDN is set by Mobile Operator policy based upon the sensitivity of the data being shared. The default LoA is LoA2 (single factor authentication). Any value submitted within the acr_values parameter in the OIDC Authorization Request will be ignored. Note that the ID Token must return acr=2 and amr="SEAM_OK".

4.2 API Modes Supported

Mobile Connect Verified MSISDN is only supported in Device-Initiated mode [8] where the User is accessing an online SP service via their mobile device using a mobile data connection.

4.2.1 Mobile Connect Verified MSISDN Share

For Mobile Connect Verified MSISDN Share, a single MSISDN value is returned in a successful Resource Response from the applicable Resource endpoint as specified in Table 2.

[10] Note that in all service requests "openid" must be included in the scope value followed by the specific Mobile Connect service scope value(s) separated by a space so for a Mobile Connect Verified MSISDN Share service request (without any other Mobile Connect services being requested the scope value would be "openid mc_vm_share".

Attribute Identifier	Type	Description
device_msisdn	string	MSISDN returned to the SP (from the Resource Server). E.164 format [5] is RECOMMENDED

Table 2: Mobile Connect Verified MSISDN Share – Returned Attributes in the Resource Response

4.2.2 Mobile Connect Verified MSISDN Match

For Mobile Connect Verified MSISDN Match, the MSISDN that the SP wishes to match against the device MSISDN captured by the Mobile Operator ID GW is submitted as part of the Resource Request as described in the Mobile Connect Resource Server Specification [10].

For Mobile Connect Verified MSISDN Match one of the attribute identifiers and associated value shown in Table 3 must be included in the Resource Request depending upon the scope value specified in the OIDC Authorization Request.

Attribute Identifier	Type	Description
device_msisdn	String	The value is the MSISDN to be verified. E.164 format [5] is RECOMMENDED
device_msisdn_hash	String	Hashed value of MSISDN to be verified. Hashing algorithms such as PBKDF2, SHA256_crypt and Argon2 can be used. The SHA256 algorithm should only be used for interoperability with current deployments of Mobile Connect[11]. The hashing algorithm can be negotiated between the SP and Mobile Operator ID GW offline[12]

Table 3: Mobile Connect Verified MSISDN Match – Attribute Identifiers and Values for the Resource Request

The Resource Request contains a JSON Payload with the mc_claims parameter which contains the requested service scope value (e.g. “mc_vm_match_hash”) followed by the appropriate attribute identifier, as defined in Table 3, and the value (hashed value in this case) to be matched. For example, based on an Mobile Connect Verified MSISDN Match (Hashed) service request, the Resource Request would be as follows:

[11] SHA256 is fast but to mitigate brute force attacks on the hash, the hashing algorithms should be slow, like, for example, PBKDF2

[12] Currently identified algorithms are PBKDF2, SHA256_crypt, Argon2 and SHA256. An Mobile Operator and SP can negotiate any of these algorithms offline.

```
POST /connect/mc_vm HTTP/1.1.
user-Agent: XXXXXXXXXX
Host: mc-idgw-Operator.example.com.
Authorization: Bearer LTRjZDMtNDUyYi1iNjk.
Content-Type: application/json.
Accept: application/json.
Content-Length: 73.
.
{
"mc_claims" : {

        "device_msisdn_hash":
"3d84a3838599719df7deacc7fb91903bde5430a8c0e007c3eba93bce0c69c5a2"
                         }

}
```

Table 4 shows the attribute identifier and associated value that is returned in the Resource Response for Mobile Connect Verified MSISDN Match. The response is the same irrespective of whether plain text or hashed values were submitted in the Resource Request.

Attribute Identifier	Type	Usage Category	Description
`device_msisdn_verified`	Boolean	REQUIRED	Match result: “true” / “false”

Table 4: Mobile Connect Verified MSISDN Match – Returned Attributes in the Resource Response

5 Service-Specific Requirements

Table 5 provides service-specific requirements relating to Mobile Connect Verified MSISDN services.

These should be used in conjunction with the following requirements in the implementation of this Mobile Connect service:

- Core Requirements specified in the Mobile Connect Technical Architecture and Core Requirements [7]. Note that these are common to all Mobile Connect Services.
- Resource Server and Attribute Services Requirements specified in the Mobile Connect Resource Server Specification [10]. Note that these are common to all Mobile Connect attribute services. Service specific requirements may further refine or qualify the more general requirements for attribute services.

For terminology and associated specifications please refer to the Mobile Connect Technical Overview [6]

No	Relating To	Requirement
Mobile Connect _Verified MSISDN _01	Service Registration	The ID GW must be able to allow a SP (client application / service) to register for the relevant Mobile Connect Verified MSISDN service variant (Share, Match (plain text MSISDN) or Match (hashed MSISDN)), based on which services the Mobile Operator supports. Mobile Connect Verified MSISDN is only supported in Device-Initiated mode as defined in the Mobile Connect Device-Initiated OIDC Profile[8]. Server-Initiated requests are not allowed.
Mobile Connect _Verified MSISDN _02	Service Invocation	A SP must be able to request either the Mobile Connect Verified MSISDN Share, Verified MSISDN Match (plain text MSISDN) or the Verified MSISDN Match (hashed MSISDN) variant through use of the appropriate scope parameter value in the OIDC Authorization Request as specified in Section 4 of this document. Note that the Mobile Operator is free to decide which variants of the service they wish to support.
Mobile Connect _Verified MSISDN _03	Service Request	For Mobile Connect Verified MSISDN services, the ID GW must be able to extract the network-authenticated MSISDN of the mobile device in use that is connected via the mobile network (This requires the Mobile Operator to support HTTP Header Enrichment or a similar mechanism). The ID GW must reject the request if the network-authenticated MSISDN is not available.
Mobile Connect _Verified MSISDN _04	Service Request	The ID GW must determine whether the network-authenticated MSISDN supplied via HTTP Header Enrichment (or similar) is registered for Mobile Connect. If not, then a Mobile Connect account must be created "on-the-fly" for Mobile Connect Verified MSISDN services if ID GW policies allow it
Mobile Connect _Verified MSISDN _05	Service Request	As an option, for the Mobile Connect Verified MSISDN service, the ID GW Authorization Server may also check the source IP address to identify that it is from the core network.
Mobile Connect _Verified MSISDN _06	Tokens	The ID GW Authorization Server must not issue a Refresh Token for the Verified MSISDN service

No	Relating To	Requirement
Mobile Connect _Verified MSISDN _07	Resource Request	The Resource Server must be able to serve Verified MSISDN Match Resource Requests through the use of a valid Access Token and the mc_claims parameter in the Resource Request as specified in this document. If any mandatory parameters or values are missing or attribute identifiers are mismatched, then the Resource Server must return an error as specified in the Mobile Connect Resource Server Specification [10] and this document.
Mobile Connect _Verified MSISDN _08	Resource Response	For Mobile Connect Verified MSISDN Share, the service will return the mobile device MSISDN upon receiving a Resource Request at the appropriate Resource endpoint with a valid Access Token as defined in Section 4 of this document. If the Resource Request fails, an error must be returned.
Mobile Connect _Verified MSISDN _09	Resource Response	The Verified MSISDN Match service must return a "true" or "false" indication of whether the MSISDN supplied by the SP matches the MSISDN of the target mobile device, if the User grants consent. The SP can provide the MSISDN in plain text or hashed form.
Mobile Connect _Verified MSISDN _10	Resource Server	The Resource Server is recommended to expose a service-specific Resource endpoint for supporting the Mobile Connect Verified MSISDN Match service OR premiumInfo endpoint should be used. This must be published as part of the Mobile Connect Provider's Metadata for the Mobile Operator's ID GW as described in Mobile Connect Technical Architecture and Core Requirements [7].
Mobile Connect_ Verified MSISDN_ 11	Resource Server	The Resource Server should try to return the matched response for Mobile Connect Verified MSISDN to the SP promptly upon receiving the Resource Reques[13]t.
Mobile Connect_ Verified MSISDN_ 12	Error Responses	Error Responses may be returned at different stages of the processing of the service request as specified in the Mobile Connect Device-Initiated OIDC Profile [8] and must be supported for the Mobile Connect Verified MSISDN service. Errors may also be generated as a result of processing of the Resource Request at the Resource Server as specified in the Mobile Connect Resource Server document [10] and must be supported for the Mobile Connect Verified MSISDN service.

[13] It depends on the network delay, infrastructure, kind of network etc., which are implementation specific. However Mobile Operators should be able to return the response in a reasonable timeframe.

No	Relating To	Requirement
		These errors are generic to Mobile Connect services and Mobile Connect attribute services, respectively. Service Specific Error Responses are specified in Annex A of this document and must be supported for the Mobile Connect Verified MSISDN service.
Mobile Connect_ Verified MSISDN_ 13	Transaction Logs	A complete Mobile Connect transaction log must be maintained, archived and accessible to resolve any disputes in line with local data protection laws and the Mobile Operator's data retention policy. For Mobile Connect Verified MSISDN, this should include: • Phone number (MSISDN) • Date & Time • Mobile Connect Verified MSISDN service scope • Attributes(s) matched / returned [i.e. MSISDN] • PCR • Consent State (active, revoked) • Status (Complete, in-process, Error) • Any Errors (error codes and error description) • Time of consent capture (if the Mobile Operator captures the consent) • Evidence of consent.

Table 5: Service Requirements

Annex A Mobile Connect Verified MSISDN - Service Specific Error Codes and Descriptions

This Annex lists the service-specific error codes and associated descriptions that are REQUIRED for the Mobile Connect Verified MSISDN services in addition to the generic error codes and descriptions that are specified in the relevant OIDC Profiles (Mobile Connect Device-Initiated OIDC Profile [8] and Mobile Connect Server-Initiated OIDC Profile [9]) and the Resource Server Specification [10].

A.1 Single Page and Two Page Environments

Certain error codes are generated depending on whether the implementation of Mobile Connect Verified MSISDN requires a single page to be displayed or two pages to be displayed on the User's Authentication Device. The default is for a single page to be displayed but there may be a requirement in certain regulatory environments to use a two-page approach. A two-page environment involves authenticating the User on the first page and presenting attributes related information and seeking User consent on the second page.

A.2 Error Responses for Device-Initiated Mode

Table 6 lists the additional error codes and descriptions for Mobile Connect Verified MSISDN that are returned from the Authorize Endpoint. These errors are applicable if the Mobile Operator captures consent.

Error Scenario	HTTP mode	Error code	Error Description [RECOMMENDED text]

In a single-page environment, the User failed to give consent (or) the ID GW was unable to authenticate the User (authentication failure).	Redirect 302	`consent_failure`
		(or)
		`access_denied`
			User failed to give consent (or) was not authenticated.

In a single-page environment, the User denied the request for consent.	Redirect 302	`consent_denied`
		(or)
		`consent_failure`
		(or)
		`access_denied`
			User has not given consent (or) consent failure.

The User was unable to give consent – a timeout occurred.	Redirect 302	`consent_failure`
		(or)
		`access_denied`
			Timeout occurred during consent capture.

In a two-page environment, the ID GW failed to authenticate the User on the first page.	Redirect 302	`consent_failure`
		(or)
		`access_denied`
			User was not authenticated.

In a two-page environment, the User was authenticated in the first step, but denied the request for consent	Redirect 302	`consent_denied`
		(or)
		`consent_failure`
		(or)
		`access_denied`
			User has not given consent (or) consent failure.

The Network MSISDN is not available or unable to retrieve network MSISDN.	Redirect 302	`access_denied`	Device MSISDN is not available.

Table 6: Mobile Connect Verified MSISDN: Errors – Device-Initiated Authorization Response

A.3 Error Responses from the Resource Endpoint

Table 7 defines the service-specific error codes and descriptions that are REQUIRED for error responses from the Resource Endpoint, returned in the Resource Response. These service specific error responses are in addition to and returned in the same format as the generic error codes specified in [10].

Error Scenario	HTTP mode	Error code	Error Description [RECOMMENDED text]
`mc_claims` parameter does not exist	Bad Request 400	`invalid_request`	REQUIRED parameter mc_claims are missing.
`mc_claims` parameter exists but the REQUIRED parameter `device_msisdn` is missing (applicable only to Mobile Connect Verified MSISDN Match)	Bad Request 400	`invalid_request`	REQUIRED parameter is missing from mc_claims.
`mc_claims` parameter exists, but has no entries.	Bad Request 400	`invalid_request`	REQUIRED parameter values from mc_claims are missing.

Table 7: Mobile Connect Verified MSISDN - Errors Returned from the Resource Endpoint

Annex B Example Resource Requests and Responses

B.1 Resource Request

The following example show a Resource Request for the Mobile Connect Verified MSISDN Share service:

```
GET /connect/mc_vm HTTP/1.1.
User-Agent: XXXXXXXXXX.
Host: mc-idgw-Operator.example.com.
Authorization: Bearer LTRjZDMtNDUyYi1iNjk.
Accept: application/json.
```

The following example shows the Resource Request for the Mobile Connect Verified MSISDN Match service using the `mc_claims` parameter in the Resource Request:

```
POST /connect/mc_vm HTTP/1.1.
user-Agent: XXXXXXXXXX
Host: mc-idgw-Operator.example.com.
Authorization: Bearer LTRjZDMtNDUyYi1iNjk.
Content-Type: application/json.
Accept: application/json.
Content-Length: 73.
.
{
"mc_claims" : {
               "device_msisdn" : "+44123456789"
               }
}
```

The following example shows the Resource Request for the Mobile Connect Verified MSISDN Match service using the `mc_claims` parameter with the hashed MSISDN in the Resource Request:

```
POST /connect/mc_vm HTTP/1.1.
user-Agent: XXXXXXXXXX
Host: mc-idgw-Operator.example.com.
Authorization: Bearer LTRjZDMtNDUyYi1iNjk.
Content-Type: application/json.
Accept: application/json.
Content-Length: 73.
.
{
"mc_claims" : {

        "device_msisdn_hash":
"3d84a3838599719df7deacc7fb91903bde5430a8c0e007c3eba93bce0c69c5a2"
                                     }
}
```

B.2 Resource Response

The following example shows the Resource Response for the Mobile Connect Verified MSISDN Share service:

```
HTTP/1.1 200 OK.
Date: Tue, 03 Oct 2017 09:37:43 GMT.
Server: XXXXXXX
Expires: Thu, 19 Nov 1981 08:52:00 GMT.
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-
check=0.
Pragma: no-cache.
Content-Length: xx.
Content-Type: application/json.
.
{
  "sub": "cd45a691-d311-4134-9a0c-2747e5110d22 "
  "device_msisdn": "+44123456789"
}
```

The following example shows the Resource Response for the Mobile Connect Verified MSISDN Match service. Note the same result is returned for both plain text and hashed variants.

```
HTTP/1.1 200 OK.
Date: Tue, 03 Oct 2017 09:37:43 GMT.
Server: XXXXXXX
Expires: Thu, 19 Nov 1981 08:52:00 GMT.
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-
check=0.
Pragma: no-cache.
Content-Length: xx.
Content-Type: application/json.
.
{
  "sub": "cd45a691-d311-4134-9a0c-2747e5110d22 "
  "device_msisdn_verified": true
}
```

Annex C Document Management

C.1 Document History

Version	Date	Brief Description of Change	Approval Authority	Editor / Company
0.1	14/11/2018	First draft of integrated version	David Pollington / GSMA	Nick Spencer
0.2	30/11/2018	Updated template and minor tweeks after review	David Pollington / GSMA	Nick Spencer
0.3	10/12/2018	Version number, doc number, prd content	David Pollington / GSMA	Siva (Venkatasivakumar Boyalakuntla) / GSMA
0.4	04/03/2019	Modified the specs based on TEF and Orange review comments. • code snippets are corrected. • removed scope value from mc_claims. • some of the requirements are reworded mentioning idgw policies. • cosmetic changes.	David Pollington / GSMA	Siva [Venkatasivakumar Boyalakuntla]/GSMA
1.0	06/12/2022	Go throught TG approval	TG	Yolanda Sanz/GSMA

C.1 Other Information

Type	Description
Document Owner	IDG
Editor/Company	Yolanda Sanz/GSMA

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

Your comments or suggestions & questions are always welcome.