# Mobile Connect Implementation Guidelines

# Version 1.0

# 15 November 2022

*This is a General 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



## 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 portfolio of mobile-enabled services that can be integrated into a Service Provider's application to support access to services provided by the Service Provider. Mobile Connect provides Authentication, Authorisation, and permissioned access to a User's Identity and Network Attributes.

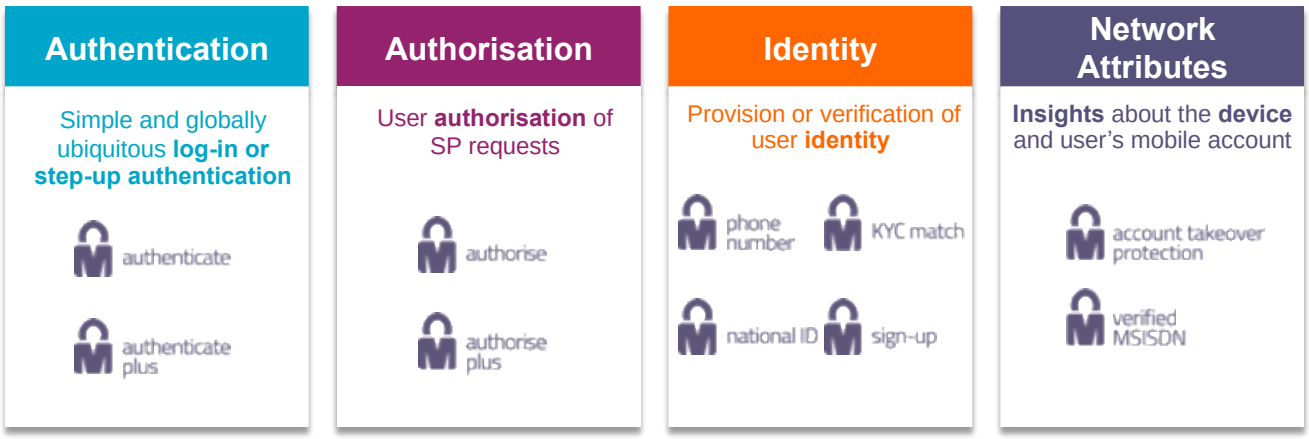


**Figure 1: Mobile Connect Portfolio of Services**

Mobile Connect is based upon the OpenID Connect (OIDC) protocol [1] which provides an identity layer on top of the OAuth 2.0 protocol [5]. It allows Users to be identified by their MSISDN (or a related Pseudonymous Customer Reference) and to be authenticated via their mobile device.

Mobile Connect defines two profiles to support Device-Initiated and Server-Initiated requests for authentication, authorisation or permissioned access to User attributes.

The serving Mobile Operator supports and selects an appropriate Authenticator to present the authentication and authorisation requests to the User on their mobile device to which the User responds. The Authenticator may also be used to seek User consent for the serving Operator to share or validate User attributes with the Service Provider. The Authenticator is selected based on Operator policy, device capability and the Level of Assurance required.

Mobile Connect also provides access to a set of User attributes[1] provided by the Mobile Operator, that can be shared or validated with a Service Provider, subject to User consent.

The Mobile Connect architecture consists of a Core framework around which additional components can be added to support different Mobile Connect services that utilise the Core.

This document provides guidance on the implementation of Mobile Connect from an Operator's ID GW perspective. It is structured in the form of a series of questions and

[1] OpenID Connect specifies a set of attributes that can be obtained from the OIDC Provider's Resource Server (e.g., the serving Operator's ID GW) also referred to as 'Protected Resources'. Mobile Connect provides an enriched set of attributes that also includes information relating to a User's mobile account and status

answers grouped around relevant topics. It seeks to build on the core documentation and should be read in conjunction with that documentation..

## 1.2 Scope

| In Scope | Out of Scope |
|---|---|
| • How to implement Mobile Connect – common questions and answers | • Detailed specification of Mobile Connect services (Refer to [16], [17], [18], [19], [20], [21], [22] and [23])<br>• Detailed specifications and documentation of the Mobile Connect Core Framework, (Refer to [8], [9] and [10]).<br>• Detailed specifications of API Exchange, Developer Portal, SDKs, Mobile Connect Interoperability Test Suite (Refer to [25], **Error! Reference source not found.**, [26], [27], [32] and [33]). |

## 1.3 Audience

The target audience for this document are mobile operators' service / technical departments who are considering deploying Mobile Connect services.

## 1.4 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 [9].

## 1.5 Terminology & Definitions

Mobile Connect technical specifications and related documentation make use of terms that are defined by the OpenID Connect Core Specification [1] and supporting specifications and extended in the OIDF CIBA (Client Initiated Backchannel Authentication Flow) [2].

The Mobile Connect Technical Overview document [7] provides a list of definitions and abbreviations that are used within the Mobile Connect Specifications. It includes terminology from source standards and interprets that terminology 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 Request2 (spelled with a 'z') throughout this document.

[2] 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 MC Authentication and MC Authorisation, hence MC specifications have adopted the term "OIDC Authorization Request" to describe this initial service request in the protocol flow.

## 1.6 References

| Ref | Doc Number | Title |
|---|---|---|
| [1] | 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 |
| [2] | 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 |
| [3] | OpenID Connect Dynamic Client Registration | https://openid.net/specs/openid-connect-registration-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] | RFC 6749 | "The OAuth 2.0 Authorization Framework", D. Hard5, Ed. October 2012 available at https://tools.ietf.org/html/rfc6749 |
| [6] | RFC 7517 | JSON Web Key (JWK) https://tools.ietf.org/html/rfc7517 |
| [7] | IDY.05 | Mobile Connect Technical Overview |
| [8] | IDY.04 | Mobile Connect Technical Architecture and Core Requirements |
| [9] | IDY.01 | Mobile Connect Device-Initiated OIDC Profile |
| [10] | IDY.02 | Mobile Connect Server-Initiated OIDC Profile |
| [11] | IDY.03 | Mobile Connect Resource Server Technical Requirements |
| [12] | IDY.16 | Mobile Connect Product Manager's Lifecycle Handbook |
| [13] | IDY.09 | Mobile Connect Authenticator Options |
| [14] | IDY.10 | Mobile Connect SIM Applet Authenticator |
| [15] | IDY.12 | Mobile Connect Smartphone Application Authenticator |
| [16] | IDY.18 | Mobile Connect Authentication Definition and Technical Requirements |
| [17] | IDY.19 | Mobile Connect Authorisation Definition and Technical Requirements |
| [18] | IDY.20 | Mobile Connect Sign-Up Definition and Technical Requirements |
| [19] | IDY.21 | Mobile Connect Phone Number Definition and Technical Requirements |
| [20] | IDY.22 | Mobile Connect National ID Definition and Technical Requirements |
| [21] | IDY.25 | Mobile Connect Verified MSISDN Definition and Technical Requirements |
| [22] | IDY.23 | Mobile Connect KYC Match Definition and Technical Requirements |
| [23] | IDY.24 | Mobile Connect ATP Definition and Technical Requirements |

| | | |
|---|---|---|
| [24] | | Mobile Connect Privacy Principles |
| [25] | IDY.33 | API Exchange Functional Description |
| [26] | IDY.35 | APIX Discovery API Specification |
| [27] | IDY.36 | APIX Request Validator API Specification |
| [28] | IDY.39 | Local Discovery Node System Architecture |
| [29] | IDY.40 | Standalone (LDN) API Exchange Setup Guide |
| [30] | IDY.41 | MNO Onboarding Form |
| [31] | | MC Deployment Options for MVNOs (PowerPoint) |
| [32] | | Mobile Connect Developer Portal: https://developer.mobileconnect.io |
| [33] | | Test Suite Portal – External User Guide |
| [34] | | Discovery UI Screens (PowerPoint) |
| [35] | OIDF Account Porting | OpenID Connect Account Porting, https://openid.net/specs/openid-connect-account-porting-1_0.html |

# 2 Mobile Connect Services

| No | Query | Response |
|---|---|---|
| [1] | How many different types of services are supported within Mobile Connect? | The Mobile Connect service portfolio covers four categories:<br>• Authentication<br>• Authorisation<br>• Identity<br>• Network Attributes<br>Specific services are defined within respective service definition and requirements documents. |
| [2] | 'Asserting the identity of the End-User' - is it implicit for all MC services? | Mobile Connect follows “Privacy by Design”, based on the Mobile Connect Privacy Principles [24]. The Identity assertion is only used when needed for the specific services, e.g. the services in the Identity category like “National ID”, subject to User Consent. For all services a request can be initiated using a pseudonymous customer reference (PCR) which ties a User to a Service Provider instead of an explicit identity assertion. PCRs are generated as part of a successful OIDC Authorization Request and can be used for subsequent requests by that SP. |
| [3] | How should the ID GW respond to an MC attribute service request when the User has instructed the Operator that they do not wish to share any data without explicit User consent? | In the case where the Operator cannot provide the requested attributes because the User has instructed to not share any attribute without explicit consent, then the “consent failure” error may be returned to the SP as specified in the relevant Mobile Connect service “Definition and Technical Requirements” document. |
| [4] | What are the guidelines for normalisation of KYC data? Is there any way an ID GW can publish normalization rules for all SPs in a region, so that SP applications can derive the normalisation rules programmatically? | Normalisation rules for the KYC claims should be negotiated offline between Operators and Service Providers within a market. Guidelines are provided in the Mobile Connect KYC Match Definition and Technical Requirements [22]. Normalisation rules supported can be included as additional metadata in the MC Provider Metadata for the SP to access. |
| **[5]** | What is the minimum viable configuration for a Mobile Connect deployment? | Mobile Connect is based around a Core Framework which must be implemented irrespective of which Mobile Connect services will be supported. This core framework supports the basic OIDC Authorization Request and responses, which are dependent upon whether Device-Initiated or Server-Initiated mode is supported. The support of specific Mobile Connect services will then place additional requirements upon the deployment. Further details can be found in the Mobile Connect Technical Overview [7]. |
| **[6]** | Which document do I need to refer to for consent capture UX flows? | The Mobile Connect Product Manager’s Lifecycle Handbook [12] contains an explanation of options for capturing user consent and also includes some examples of the User interface. Mobile Connect specific service “Definition and Technical Requirements” documents contain example user flows relating to that specific service (See “References” section 0 ). |

# 3 API Exchange and Discovery

## 3.1 Discovery

| No | Query | Response |
|---|---|---|
| **[7]** | How does an Operator's (Self Service) App get Discovery credentials? | An Operator's self-care app is like any other Service Provider application. The application will need to be registered via the Mobile Connect Developer Portal [32] from which the relevant credentials (`client_id`, `client_secret`) and Discovery Endpoints can be obtained. |
| **[8]** | How does the SP get the ID GW credentials? | ID GW credentials can be obtained using the Mobile Connect Discovery Service via the API Exchange. When a Service Provider registers their application on the Mobile Connect Developer Portal, they will be issued with credentials and links to access the Discovery service. Different credentials and links are provided for access to the Sandbox environment, any Staging or Pre-Production environment and the live Mobile Connect (Production) environment. Care should be taken to ensure the correct credentials are used within the "live" SP application. ID GW credentials can be obtained by submitting a Discovery Request to the API Exchange. Note it is also possible for an SP to obtain credentials directly from an Operator without using APIX. |
| **[9]** | Is Mobile Connect implemented using OIDC protocols? What is the difference between OIDC discovery and MC discovery? | The Mobile Connect API is a specific implementation of the OIDC protocol. Mobile Connect Discovery does not use the OIDC Discovery specifications. Details on the Mobile Connect Discovery Service can be found in the API Exchange Functional Description [25] and APIX Discovery API [26]. |
| **[10]** | How does the discovery service resolve ID GW end points? | The Discovery Service utilises information submitted by an SP application in a Discovery Request to identify the serving Operator. This includes MCC and MNC, IP address (if the request is using a mobile data connection) or the User's MSISDN. The serving Operator is uniquely identified via their mobile country code (MCC) and mobile network code (MNC) which is then used to look-up the relevant information including the link to the MC Provider Metadata (which provides the services supported and associated endpoints) and associated SP credentials that should be used in a request. |
| **[11]** | Is Discovery only possible via the API Exchange? | The API Exchange seeks to offer a global service. In some markets there is a requirement for a local Discovery service or ID GW information is supplied directly to SPs. The Discovery service via the API Exchange will provide further information if a local Discovery service is required. |
| **[12]** | Is there a specification for the Discovery UI? | The Discovery UI is presented to a User in the event that information supplied in a Discovery Request by an SP application is insufficient to identify the serving Operator. The |

| No | Query | Response |
|---|---|---|
| | | Discovery allows a User to identify the serving Operator and/or enter their MSISDN without this information being exposed to the SP application. A separate specification for the Discovery UI does not exist but further information can be found in the API Exchange Functional Description [25]. Reference [34] also provides information on Discovery UI screens and branding. |
| **[13]** | Does Mobile Connect support Mobile Virtual Network Operators (MVNOs)? | MVNO is a term applied to an organisation that has a relationship with a Mobile Operator ranging from a reseller of services (under its own brand) through to owning its own core network and leasing radio network capacity from the host Operator. If an MVNO has its own core network, it will be registered and have a Mobile Network Code (MNC). If an MVNO has its own MNC then it can be registered at the API Exchange and be discoverable by the API Exchange. In this case the MVNO can implement its own Mobile Connect service (See MC Deployment Options for MVNOs [31]) |
| **[14]** | How can an MVNO offer Mobile Connect services to Service Providers for its Users if the MVNO does not have their own MNC? | If an MVNO does not have its own MNC then it will not be able to register separately at the API Exchange and therefore will not be discoverable. In this case the MVNO will have to rely on their host Operator to process Mobile Connect requests on behalf of the MVNO's End-Users. |
| **[15]** | How can an Operator's ID GW host requests for an MVNO using their network? | The host Operator will need to implement a mechanism to identify MVNO customers (Users) in order to manage Mobile Connect services appropriately on behalf of the MVNO. This will depend on how the Operator manages MVNOs and their customers (e.g. by allocating number ranges to MVNO customers). |
| **[16]** | How does Discovery handle MVNO numbers? | • If the Operator does not support the MVNO's numbers, then discovery should return an error, "Operator not found".<br>• If the MVNO number reaches the ID GW, then an error "access_denied" must be returned. |

## 3.2 Alternatives to APIX Discovery

| No | Query | Response |
|---|---|---|
| **[17]** | Is it possible to on-board Service Providers without using the Developer Portal and API Exchange? | It is possible to use local processes to on-board Service Providers and to register their applications. This should be carried out in a consistent manner within a market to ensure inter-operability. Using a local process will limit the access for a Service Provider to that particular market and will require local testing and validation of the SP application prior to commercial launch. This tends to mean that the process of local on-boarding tends to be highly manual, time-consuming and not scalable. It may be appropriate where the MC |

| No | Query | Response |
|---|---|---|
| | | services are targeted to a few potential Service Providers only.<br>The benefit of the Developer Portal and API Exchange for Service Providers is that it provides a set of tools to allow deployment of applications across markets. It provides a sandbox environment to test applications and allows the app to be approved before facilitating API access to different Operators that are registered on the API Exchange (promoting). |
| **[18]** | I do not want to use GSMA APIX, what are the possibilities available to me? Can I implement my own APIX service? | In some markets, local regulations require that data is not stored outside of the country or region and so local Discovery Services have to be implemented. In other markets Mobile Connect services are targeted at a small group of Service Providers and therefore they are onboarded directly. This process tends to be very manual and is not recommended where multiple Service Providers may require access to Mobile Connect services. Further details can be found in [28] and [29]. |
| **[19]** | Can I implement my own local APIX platform, without using the GSMA APIX? Do you have reference architecture that I can follow? | Yes, it may be possible to share the APIX codebase with an Operator to install and host locally in their own market (as their own Discovery solution). In addition, GSMA also has a reference implementation of a simple Discovery node that can also be provided. Note that in both cases, there is no support available from the GSMA. |

## 3.3 Operator On-boarding and Testing

| No | Query | Response |
|---|---|---|
| **[20]** | What documents are available to explain how Operators can be onboarded onto the API Exchange? | Operators are onboarded onto the API Exchange after completing an MNO Onboarding form [30]. This form outlines the required information that must be provided including data that is required to be added to the Mobile Connect Developer Portal to enable Developers to request access to a specific ID GW. |
| **[21]** | What are the requirements for an Operator to be listed on the Developer portal? | The Operator's OD GW implementation must pass the relevant test cases appropriate to the Mobile Connect services that it is intending to offer before they can be listed on the Developer Portal. These test cases relate to the requirements that are marked as "REQUIRED" and MUST be satisfied in the Core Framework and associated with the specific services that will be offered, as a minimum. Core requirements are included within the Mobile Connect Technical Architecture and Core Requirements document [8] , requirements specific to the Resource Server used to support MC attribute services can be found in [11] and service specific requirements are included in the relevant service "Definition and Technical Requirements" document (See "References" section 0). |

| No | Query | Response |
|---|---|---|
| **[22]** | Which documents do I need to use for testing? | The MC Test Suite Portal allows Operators and their partners to test their ID GW implementation against Mobile Connect Technical Specifications. Technical specifications include the requirements that must be satisfied by the suite of tests. For Operators the latest technical specifications can be accessed from the InfoCentre2 portal. See the Mobile Connect Technical Overview [7] for more details on Mobile Connect Specifications |
| **[23]** | Which version of MC should be used? | Deployments of Mobile Connect in new markets should use the latest version of Mobile Connect. New deployments in existing markets should use the same version that is currently deployed by other Operators within that Market. Details on Mobile Connect releases can be found in the Mobile Connect Technical Architecture and Core Requirements document [8]. |

## 3.4 SP On-boarding

| No | Query | Response |
|---|---|---|
| **[24]** | How can a Service Provider's App register with an ID GW? How does the ID GW receive the metadata of an SP service / application? | A Service Provider can register its app directly with an Operator ID GW (e.g. via the registration portal) or can register via the API Exchange. In this latter case, the Mobile Connect Developer Portal provides step-by-step guidelines to help SP's to on-board and register their applications with the API exchange and to request access from individual Operator ID GWs that are also registered on the API Exchange. |
| **[25]** | How and when does the SP get credentials? | Once an SPs application is ready for testing, they register their application on the Developer Portal. As part of their registration they are required to provide information including a `client_name` and sector_identifier_uri. A `client_id` and `client_secret` will be issued for access to test environment. Credentials to access a specific gateway (or test environment) will be obtained upon successful registration with an ID GW (either directly or via the API Exchange / Developer portal. |
| **[26]** | Why would an SP use of multiple `client_name`s? How is the `client_name` validated? | `client_name` provides a short descriptor to identify the SP application on a User's mobile device. It is REQUIRED for Mobile Connect Authorisation but is optional otherwise. If an SP registers a number of applications, then it is possible to define a set of `client_name` values through the Mobile Connect Developer Portal [32]. However, only one `client_name` should be used per application. The `client_name` value submitted in an OIDC Authorization Request must match one of the `client_name` values submitted during registration. |
| **[27]** | How is the sector_identifier_uri and its contents used? | The value of the `sector_identifier_uri` MUST be a URL using the `https` scheme that references a file with a single JSON array of `jwks_uri` values for SI |

| No | Query | Response |
|---|---|---|
| | | polling mode and / or SP notification_uri values for SI notification mode. It can also include `redirect_uri` values for DI mode. It provides a way for a group of services under common administrative control to have consistent PCR values; independent of the individual domain names. It also provides a way for SPs to change service domains without having to re-register all their Users.<br>The values MUST be included in the elements of the array, or registration MUST fail. This MUST be validated at SP on-boarding time; If it is not registered then the registration process MUST throw an error.<br><br>Mobile Connect Providers MUST utilise the `sector_identifier_uri`. |
| **[28]** | Can I have single sector identifier URI for both DI mode, SI mode – notification and SI mode - polling? What is the recommended implementation mechanism? | Yes. The `sector_identifier_uri` points to a list that contains possible `redirect_uri` values and / or notification-uri values and/or `jwks_uri` values using HTTPS.<br>Within an OIDC Authorization request the SP Client will specify the `redirect_uri` or notification_uri to be used which MUST match one of the entries in the `sector_identifier_uri` list. |
| **[29]** | I want to maintain separate sector_identifier_uri lists for DI mode and SI modes? is it allowed? If yes, what are the semantics and how I can write the implementation? What if the SP uses two modes for the same MSISDN? | This is possible - Prefix the value of the sector_identifier_uri with “si_” and “di_”. Remember for DI mode the contents must be list of `redirect_uri` values, and in SI mode, they must be either `notification-uri` values and / or `jwks_uri` for polling. |
| **[30]** | Is sector_identifier_uri content in plain text or encoded in some way? | The value of the `sector_identifier_uri` MUST be a URL using the HTTPS scheme that references a JSON file containing an array of uri values. These must be validated at SP client application registration. See [3]. |
| **[31]** | In MC specifications the `client_id` is defined as globally unique – is there a mandatory definition for generating a `client_id`? Several MC implementations do not implement this - how can interoperability be achieved? | There is no mandatory definition for the generation of `client_id` and different algorithms can be used to ensure that as far as possible the `client_id` is unique. It is recognised that this can be difficult if third parties are generating the `client_id`.<br>From an APIX perspective, the discovery service can generate a unique `client_id`, using a GUID during the generation of SP credentials. |
| **[32]** | What is the format of the SP’s `redirect_uri`. What is the maximum number of redirect URIs that can be registered with the ID GW? | The format of the `redirect_uri` values are as defined in Section 3.1.2 of RFC6749 [5]. There is no maximum number specified for `redirect_uri` values. |

| No | Query | Response |
|---|---|---|
| **[33]** | Are there any limitations on `client_name` length during registration? Can the ID GW reject if the length is exceeded? | This must be a maximum of 16 bytes.<br>The ID GW will generate an error if the `client_name` submitted in the OIDC Authorization Request does not match the value registered but will not explicitly reject based on length. |
| **[34]** | Can the ID GW accept notification endpoints using the HTTP scheme? What are the security problems with this? | All URLs must be protected with TLS. Using basic HTTP leads to security risks around leakage of tokens and manipulation of data. |
| **[35]** | What are the guidelines to integrate SP applications / clients to MC, what kind features should be provided by an ID GW? | The most straightforward mechanism for integration of SP applications / clients to MC is via the Developer Portal [32] where developers are able to test their application. This also allows them to register their clients/applications and to request access (promote) to different Operator ID GW. The ID GW will need to be able to support the required MC services. |
| **[36]** | How are the keys to verify the signed request object for an OIDC Authorization Request from an SP Client application shared in SI mode? With an example can you explain the verification procedure? | The relevant keys are provided through the `jwks_uri` which is registered during SP onboarding in SI mode. The verification is done by the IDGW based on Section 5.2 of RFC7515 **Error! Reference source not found.**, Typically implementations make use of standard libraries to validate the JWT. |
| **[37]** | With an example explain the contents of (SP) `jwks_uri` and how the keys are structured? How are they used? | The URL for the SP's JSON Web Key Set document that lists the signing keys that the ID GW uses to validate signatures from the SP. All Server-Initiated Mobile Connect service requests MUST be submitted using an OIDC signed Request Object [6].Only asymmetric signatures are supported.<br>Further information on the JSON Web Key Set (jwks) can be found at [6] including examples |
| **[38]** | Can SP register multiple notification endpoints, like multiple `redirect_uri` values in DI mode? | Yes. This is specified within the Mobile Connect Server-Initiated OIDC Profile [10] |
| **[39]** | What is the format of the ID GW public (and private) key? | RSA2048 |
| [40] | Explain the mechanism of MSISDN encryption and decryption in MC? How are the keys shared? | MSISDN Encryption is performed by the API Exchange using the public key of the serving Operator ID GW. The ID GW is then able to decrypt the MSISDN using its private key. See Mobile Connect Technical Architecture and Core Requirements for more details [8] and also the API Exchange Functional Description [25]. |

## 3.5 Request Validator API

| No | Query | Response |
|---|---|---|
| **[41]** | How does the Request Validator service work? How do I get RV credentials? | In order to be able to use the Request Validator service:<br>• Developer Organisation must be on-boarded on to APIX.<br>• Serving Operator Organization must be on-boarded on to APIX.<br>• Request Validator credentials have been obtained (by registering with APIX Admin) and should have read level access to serving operator data<br>• Successful discovery should be performed to receive the ID gateway credentials(client id and secret)<br>• An active Developer Application provides API credentials (client id and `client_secret`) to the Serving Operator's endpoint based on the API signature of the Serving Operator.<br>The Request Validator then after verifying the RV credentials of the Operator's ID GW can then look-up the relevant SP application details. Full details can be found in [27] |
| **[42]** | What is a request validator API? How can it be used? | Typically, when a Service Provider or Developer registers an application on the API Exchange, they are able to requests access from registered Operator ID GW and during this process SP application credentials are provided to the ID GW and cached at the ID GW. The Request Validator API allows an ID GW to validate a Service Provider Client application request for access to Mobile Connect services. It is typically invoked if the ID GW receives a new (unknown) `client_id` & `client_secret`. Use of the Request Validator API is optional and implemented by the ID GW if required. Full details can be found in [27] |
| **[43]** | How can the ID GW use the Request Validator API? | The Request Validator request contains the ID GW credentials of the SP Client application (`client_id` & `client_secret`). If the SP Client application is successfully validated, then the API Exchange `client_id` and `client_secret` are returned along with a number of other parameters which correspond to the information that the SP provided during onboarding and application registration. Once a successful RV response is received the SP Client application data can be cached for future use. Full details can be found in [27] |
| **[44]** | What data is relevant in the Request Validator response? | Key parameters include:<br>• X_client_id<br>• X_client_secret<br>• redirectUri & redirectUriArray<br>• client_name (& array of client_names)<br>• operating mode (di_mode, si_mode or di_si_mode)<br>• sector_identifier_uri<br>• jwks_uri (for si_mode & di_si_mode) |

| No | Query | Response |
|---|---|---|
| | | • request_object_signing_alg (for si_mode & di_si_mode)<br>• notification_uri (for si_mode & di_si_mode)<br>See Request Validator API Specification [27] |

# 4 Mobile Connect Core Framework

## 4.1 User Registration

| No | Query | Response |
|---|---|---|
| **[45]** | How can End-Users register for Mobile Connect? | The registration process for End-Users is determined by Operators implementing Mobile Connect (as part of their Operational Policies) subject to any local legal and regulatory requirements. Users can register automatically as part of a new mobile subscription, on application or “on-the-fly” as part of the process for registering with or accessing a Service Provider’s application. User’s will need to understand and agree to applicable terms and conditions. Further details can be found in the Product Manager’s Lifecycle Handbook [12]. |
| **[46]** | Is it possible to register Users for Mobile Connect using offline processes? | Yes, it is possible to register Users using offline processes, subject local regulations. This might include bulk registration by an Operator as part of the introduction of Mobile Connect services within a market, in response to a marketing campaign or as part of registration for a new mobile subscription / SIM. User’s will need to understand and agree to applicable terms and conditions. |
| **[47]** | What is the best approach to manage Mobile Connect service requests to Users who do not have a Mobile Connect account? | If the Operator ID GW supports it, on-the-fly registration (e.g. MC Authenticate) can be used. This is where the User may be registered for Mobile Connect as part of the processing of a Mobile Connect service request. This will typically involve displaying a registration / Terms & Conditions page and capturing the User’s consent/agreement. On-the-fly registration will typically only enable registration for single factor (LoA2) authentication as the requirements to download and configure an Authenticator to support two-factor (LoA3) authentication will significantly interrupt the process to access or use an SP application. Registration or upgrade to two-factor authentication should then be handled via an offline process e.g. via the Operator’s self-care portal. If Operator policy does not allow “on-the-fly” registration then an “access_denied” error should be returned (See MC OIDC Profiles [9] and [10] ). |
| **[48]** | How can End-Users register for Mobile Connect? | The registration process for End-Users is determined by Operators implementing Mobile Connect (as part of their Operational Policies) subject to any local legal and regulatory requirements. Users can register automatically as part of a new mobile subscription, on application or “on-the-fly” as part of the process for registering with or accessing a Service Provider’s application. User’s will need to understand and |

| No | Query | Response |
|---|---|---|
| | | agree to applicable terms and conditions. Further details can be found in the Product Manager's Lifecycle Handbook [12]. |
| **[49]** | What are the guidelines for registration or provision of MC services to minors? | Policies for the provision of service to minors are set by local Operators, subject to local legal and regulatory requirements, Many MC services are not appropriate to minors. |
| **[50]** | What is the best practice when an End-User's account is in a suspended state? Should some MC service requests be accepted, and, if so, which services must be rejected by an ID GW? | If an End-User's mobile account is suspended, then typically the Mobile Connect account will also be suspended indicating that Mobile Connect service requests should be rejected. It may be that Operators continue to support some services related to fraud prevention e.g. check for lost or stolen device. |
| **[51]** | If a User ports from one Operator to another, is it possible to also port the Mobile Connect account for the User? | Yes, Mobile Connect supports account porting based on the Account Porting specifications from the OpenID Foundation. Account porting is User driven, so the User needs to start the process by interacting with the old Operator (similar to MSISDN porting). For Mobile Connect Account Porting to work – both the Operators (old and new for the User) need to support Mobile Connect Account Porting [35]. |
| **[52]** | Is it possible to port a Mobile Connect account to another Operator even if the User does not port the MSISDN? | Yes, it is possible to port the Mobile Connect account even if the User does not port the MSISDN. The old Operator needs to have a mechanism to authenticate the User using the old MSISDN [35]. |

## 4.2 User Identifiers

| No | Query | Response |
|---|---|---|
| **[53]** | What is a PCR [Pseudo=anonymous customer reference] in MC? What is the recommended format? Is PCR a PII? | A PCR is the pseudonymous User identifier returned in a Mobile Connect response – in the "sub" claim in the ID Token. The PCR is a PPID (Pairwise Pseudonymous ID) generated as a function of MSISDN and the Sector ID. The recommended format for the PCR is a GUID. PCR does not represent PII as its pseudonymous and also is a PPID – so it's not possible to track the User using the PCR across different SPs. |
| **[54]** | MSISDN is the End-User identifier in MC? Are there any other identifiers like passport, national ID are allowed? | Mobile Connect uses MSISDN and the derivatives (e.g. PCR, Encrypted MSISDN) as the User identifier. Other general-purpose identifiers like passport id, national id etc. are not used as User identifiers in Mobile Connect and are treated as attributes. |
| **[55]** | MC API recommends supporting plain MSISDN, ENCR MSISDN and PCR as End-User identifiers within an MC service request? Do I need to support all of them? | MSISDN needs to be supported and PCR is recommended.<br>If the ID GW registers with the API Exchange (to support Discovery and SP Registration) then Encrypted MSISDN also needs to be supported. |
| **[56]** | What is the difference between user-facing identifier (MSISDN) and | MSISDN is the User facing ID – which is known to the User and can be supplied by the User when and if needed. The system facing identifier (PCR) is generally not known to the |

| No | Query | Response |
|---|---|---|
| | system-facing identifier (PCR)? Why the distinction? | User and is used by the systems – ID GW and the SP application/server. The PCR is tied to the User and SP client and is opaque so prevents the User being tracked across different Service Providers and applications and avoids the use of the MSISDN if the User does not wish to share it. |
| **[57]** | How does GDPR realised in Mobile Connect? | Mobile Connect Privacy Principles [24] embraces “privacy by design”. No personal data is shared without the consent from the User and without the legitimate need. Mobile Connect uses the pseudonymisation and shares the PCR as the User reference. |
| **[58]** | Is there any specific algorithm that is used to generate PCR? | The recommended format for PCR is a GUID (See [8] ) |

## 4.3 ID GW Implementation

| No | Query | Response |
|---|---|---|
| **[59]** | Do you have a reference topology for Mobile Connect? Which document do I need to refer to? | MC Technical Architecture and Core Requirements document [8]. |
| **[60]** | How does MC ensure a similar integration requirement and experience for SPs by all MC providers? | All MC implementations need to implement the APIs based on the MC OIDC Profiles ([9] and [10] - See also [8]) and the interfaces are tested using the Mobile Connect Interoperability Test Suite [33]. |
| **[61]** | How many different types of deployment topologies are possible in MC? | MC deployment can be done in various ways, some popular ones are:<br>• Operator has their own on-premise or cloud hosted ID GW<br>• Operator has an ID GW from a managed service provider<br>• Operators in a country deploys a shared ID GW |
| **[62]** | We have OAuth2.0 server and would like to convert into OIDC for MC services. How different is OIDC from OAuth 2.0? | OIDC adds on an Identity layer on top of OAuth 2.0. The key difference is – OIDC includes an ID Token along with the Access Token in the token response. The ID Token is a signed JWT – which provides the Authentication Context. Mobile Connect is a specific implementation of Open ID Connect which utilises a User’s MSISDN and an associate Pseudonymous Customer Reference as an identifier. With Mobile Connect the User’s mobile device serves as an Authentication Device and make use of Authenticators to provide a mechanism to prompt a User to authenticate, authorise a transaction or provide consent based upon a context presented on that Authentication Device |
| **[63]** | Do you have a reference topology for Mobile Connect? Which document do I need to refer to? | MC Technical Architecture and Core Requirements document [8]. |
| **[64]** | How does MC ensure a similar integration requirement and experience for SPs by all MC providers? | All MC implementations need to implement the APIs based on the MC Profiles and the interfaces are tested using the Mobile Connect Interoperability Test Suite |

| No | Query | Response |
|---|---|---|
| **[65]** | How many different types of deployment topologies are possible in MC? | MC deployment can be done in various ways, some popular ones are:<br>• Operator has their own on-premise or cloud hosted ID GW<br>• Operator has an ID GW from a managed service provider<br>• Operators in a country deploys a shared ID GW |
| **[66]** | We have OAuth2.0 server and would like to convert into OIDC for MC services. How different is OIDC from OAuth 2.0? | OIDC adds on an Identity layer on top of OAuth 2.0. The key difference is – OIDC includes an ID Token along with the Access Token in the token response. The ID Token is a signed JWT – which provides the Authentication Context. Mobile Connect is a specific implementation of Open ID Connect which utilises a User's MSISDN and an associate Pseudonymous Customer Reference as an identifier. With Mobile Connect the User's mobile device serves as an Authentication Device and make use of Authenticators to provide a mechanism to prompt a User to authenticate, authorise a transaction or provide consent based upon a context presented on that Authentication Device |
| **[67]** | What are the key responsibilities for the ID GW? | The ID GW is the technical entry point for the SP to request Mobile Connect services from the Operator. The key responsibilities of the ID GW are:<br>• Exposure of OIDC interfaces based on the Mobile Connect OIDC Profiles<br>• Access control of the clients to access the OIDC endpoints<br>• Throttling management of the requests from the SPs<br>• Application credentials storage and management, with interactions with APIX/Discovery node<br>• Interactions with the Authentication sub-system<br>• Signing of the ID Token<br>• Logging and reporting |
| **[68]** | What is policy-based routing? and What is the best practice in MC for policy-based routing? | The ID GW needs to integrate with one or more Authenticator sub-systems, when implementing the ID GW, it is recommended to implement policy-based routing functionality to route requests and responses to specific authenticators, so that the most appropriate authenticator can be selected (when more than one is available) based on the current context. The context could be a fall-back scenario, e.g. when one of the authenticators fails to deliver the authentication challenge then the policy-based routing functionality can select the next possible authenticator. |
| **[69]** | What is the best practice for an ID GW to select an authenticator suitable for MC? | When multiple authenticators are available, the ID GW can select the authenticator based on Operator policy, which may be based on:<br>• Device capability/eligibility (e.g. can the device support Smartphone App Authenticator) |

| No | Query | Response |
|---|---|---|
| | | • User eligibility (e.g. is the User allowed to get the SIM Applet)<br>• SIM eligibility (e.g. Is the SIM eligible to deploy the SIM applet)<br>• Fall-back scenario (e.g. when one authenticator fails to deliver the authentication challenge)<br>• LoA requested<br>• Product criteria (e.g. some services may need the usage of seamless authentication only) |
| **[70]** | ID GW would like to use different hashing algorithm than recommended for an MC service, what are the guidelines? how the agreed hashing algorithm can be shared between SP application and ID GW for a specific MC service? [ex. kyc match hashed, vm hash etc]. | The hashing algorithm used in specific services can be published in the MC Provider Metadata, so that the SP client can programmatically identify the hashing mechanism in a predictive way and use that in the service requests. See Mobile Connect Technical Architecture and Core Requirements [8]. |
| **[71]** | What are the mandatory features MC ID GW must implement complying to OIDC for MC? | The ID GW must implement the OIDC interfaces based on the Mobile Connect OIDC Profiles (Device Initiated [9], Server Initiated [10]). See also Mobile Connect Technical Architecture and Core Requirements [8]. |
| **[72]** | We have deployed an ID GW based on Mobile Connect v1.1. Do we need to do anything to improve compatibility for SP's using a higher MC version? | Operators that have deployed an ID GW against Mobile Connect v1.1 can improve compatibility with SPs that are utilising the latest version of Mobile Connect by:<br>• Adding the `hashed_login_hint` parameter to the ID Token<br>If an SP wishes to use Mobile Connect across several applications using the same PCR then the ID GW would have to be upgraded to at least v2.3 in order to support the `sector_identifier_uri` that the SP must specify and that is used as the basis for generating the PCR. |
| **[73]** | We have deployed an ID GW based on Mobile Connect v2. Do we need to do anything to improve compatibility for SP's using a higher MC version? | Operators that have deployed an ID GW against Mobile Connect v2.0 can upgrade to v2.3 with a minor upgrade, including:<br>• Addition of the `hashed_login_hint` parameter to the ID Token<br>• Use of `state` and `at_hash` parameters in the ID Token which are now mandatory<br>• Support for the `sector_identifier_uri`<br>• The PCR should be generated based upon the `sector_identifier_uri` submitted by the SP.<br>• Use of the Mobile Connect Provider Metadata is also mandatory from v2.3 onwards.<br>Full details on the structure of the OIDC Authorization Request and the ID Token that is returned are provided in the |

| No | Query | Response |
|---|---|---|
| | | relevant MC OIDC Profile (Device-Initiated OIDC Profile [9] or Server-Initiated OIDC Profile [10]. |

## 4.4 Authenticators and Level of Assurance

| No | Query | Response |
|---|---|---|
| **[74]** | What is the significance of "Levels of Assurance" (LoA) used in Mobile Connect? | Level of Assurance represents the degree of confidence that the person authenticating is the same person that registered for Mobile Connect and reflects the number of factors associated with the authentication process. These can be mapped to other definitions of Level of Assurance in different standards. LoA values are:<br>• LoA1: Not relevant for Mobile Connect (not used)<br>• LoA2: Single-factor authentication<br>• LoA3: Two-factor authentication<br>• LoA4: Two-factor with PKI |
| **[75]** | What are the different types of authenticators commonly used in Mobile Connect? | • Seamless Authenticator: LoA2<br>• SMS+URL: LoA2<br>• USSD-based Authenticator: LoA2<br>• SIM Applet: LoA2, LoA3, LoA4<br>• Smartphone App Authenticator: LoA2, LoA3, LoA4<br>See Authenticator Options [13]. |
| **[76]** | Is SMS+OTP considered as an authenticator in MC? | SMS+OTP is a possible authenticator in Mobile Connect although it is not recommended as a secure authenticator. The User experience is not optimal if the User needs to type back the OTP from one device to another. |
| **[77]** | How do we implement a Smartphone App Authenticator? | There is a Smartphone App Authenticator specification for Mobile Connect, which can be used as a guidance to implement the Smartphone App Authenticator [15]. |
| **[78]** | How do we implement SIM Applet Authenticator? | There is a specification for SIM Applet Authenticator for Mobile Connect, which provides the details including the SIM requirements, the detailed messaging specification between the SIM Applet Authentication Server (MSSP) and the SIM applet, deployment architecture and other elements needed to implement the SIM Applet Authenticator [14]. |
| **[79]** | What are the encoding schemes recommended by MC for different authenticators? Are there any guidelines? | UCS2, GSM-7 and UTF-8. The API parameters generally use UTF-8. The Authenticators may use different encoding schemes depending on the messaging interface used for the Authenticator, these are defined in the Authenticator specifications (e.g. the SIM Applet specification) |
| **[80]** | Is it possible to use biometrics-based authentication in any of the authenticators in Mobile Connect? | The Smartphone App Authenticator can use biometrics based on the smartphone devices that are being supported. This can be used as a second factor in place of the PIN. |

| No | Query | Response |
|---|---|---|
| **[81]** | Does Mobile Connect authenticators support multi-language? | The Authenticator implementation is carried out by the Operator and can support multiple languages. The Authenticator specifications describe mechanisms to support multi-language (e.g. the SIM Applet specification). The ID GW Provider Metadata will indicate what languages are supported. |
| **[82]** | How is the Seamless Authentication supported? | The most popular way for supporting Seamless Authentication is through HTTP Header Enrichment (HHE). There are different ways this can be done:<br>• The core network adds the authenticated MSISDN as a HTTP header targeted to the ID GW endpoint.<br>• The ID GW uses the client_ip to get the authenticated MSISDN from the network |
| **[83]** | Can Seamless Authentication be requested by a Service Provider? | An SP can request single factor authentication (LoA2) and if the User's device is using the mobile data connection and the ID GW supports header enrichment or similar to be able to extract the MSISDN then the Operator may select seamless authentication. The SP cannot directly request it. The SP is able to specify not to use seamless authentication, if available, by setting `prompt` to "no_seam". This might be used where the User is in a 3G/4G hotspot or is using a personal hotspot. Note that Seamless Authentication can only be used in DI mode when the mobile device is also the Consumption Device and is accessing the service over the mobile network. |
| **[84]** | Are there any guidelines on implementation and selection of Authenticators? | • Authenticators for Mobile Connect might use existing systems and network components in order to present a prompt and seek a response on a User's mobile device (Authentication Device) and an ID GW may support multiple Authenticators, possibly including new Authenticators that have yet to be specified.<br>• On this basis it is recommended to implement different Authenticators as pluggable components, as far as possible, by using an Authenticator adaptor that provides a consistent interface to the ID GW and abstracts the specific implementation of the Authenticator mechanism and the interaction that might be required.<br>• The specifics of how an authenticator is implemented and the routing of requests to specific network elements should be confined within the authenticator adaptor as far as possible. This provides a loose coupling between the authenticator implementation and the Mobile Connect system.<br>  ○ The policy routing engine within policy management should route the Authenticator request to the appropriate authenticator adaptor. |

| No | Query | Response |
|---|---|---|
| | | ○ The adaptor may then use an authenticator specific interaction model to invoke and communicate with the User.<br>○ The adaptor may use more than one call to invoke the authenticator (composition).<br>• The various factors (inputs) used for the authentication selection should be stored in a configuration database. It should be possible to manage and administer the lifecycle of these factors.<br>• The authenticator selection mechanism should validate and ensure that the request contains enough information to invoke the authenticator and all authenticator specific details are available.<br>• The authenticator selection mechanism should convert and add any specific information needed for the authenticator.<br>• A fall-back mechanism should be configured such that authenticators can be selected based on the LoA order. The Service Provider may pass multiple LoA values in `acr_values` parameter in order of preference. |

## 4.5 API – General

| No | Query | Response |
|---|---|---|
| **[85]** | Are there any guidelines for general session management? | Mobile Connect interactions are stateless. In a DI mode – the Authorization call can be correlated back with the response through the redirect using the state parameter. In SI mode – the auth_req_id is used to correlate back the Token response with the SI Authorization call. |
| **[86]** | Which version of the TLS must be supported in MC? | The ID GW should always use the latest version of TLS. This is currently TLS v1.3. |
| **[87]** | What is the minimum duration that an SSL certificate should be valid for an ID GW server? | This depends on the Operators' operational policy, but it is recommended that this should be 3 months at least. |
| **[88]** | What is form serialization and in which MC requests is form serialization used? | Form serialization is used to output URL encoded or hashed output for objects, when a POST request is used to make the call, e.g. whenever a request is made from a server. |
| **[89]** | What is the form urlencoded scheme and in which MC requests is it used? | "Form urlencoded" is used as the Content-Type when POST is used for the requests, e.g. for Authorization and Token calls.<br>Content-Type: application/x-www-form-urlencoded |
| **[90]** | In which MC requests I need to use JSON serialization? | In DI mode: The Token response uses JSON serialization.<br>In SI mode: The Authorization response (error or acknowledgement) and also the Token response uses JSON serialization. |

| No | Query | Response |
|---|---|---|
| **[91]** | What is query string serialization and in which MC requests is this used? | Query string serialization is used in GET requests to the Authorization Server. The parameters and values are added in the query string using `application/x-www-form-urlencoded` format. See [1], Section 13 for more information. |
| **[92]** | How are string operations carried out in MC? How are JSON strings and other unicode strings compared with each other? | String operations and comparisons between JSON strings and other Unicode strings MUST be performed as specified below:<br>• Remove any JSON applied escaping to produce an array of Unicode code points.<br>• Unicode Normalization MUST NOT be applied at any point to either the JSON string or to the string it is to be compared against.<br>• Comparisons between the two strings MUST be performed as a Unicode code point to code point equality comparison. |
| **[93]** | What are symmetric signatures and asymmetric signatures? in MC what kind of signature types are used. | Symmetric or asymmetric signature depends on the usage of symmetric or asymmetric keys. Mobile Connect supports both types of signatures in the ID Token. |
| **[94]** | What is the best practice for using the asymmetric signing keys, if they are used in MC? | The asymmetric keys should be rotated at regular intervals, dependent on the Operator operational policies. |
| **[95]** | What client authentication methods are allowed in MC - client_secret_basic or `client_secret`_post? | Mobile Connect uses client_secret_basic, where the `client_secret` is passed using HTTP Basic Authentication scheme in DI mode. |
| **[96]** | Are confidential clients allowed in MC? What is the best practice? | All Mobile Connect clients are confidential clients, they need to keep the `client_secret` secure and secret. The Token call must be initiated from a server, so that the `client_secret` can be stored as a secret and secure. |
| **[97]** | Are multiple `response_type` combinations allowed in MC? if yes, how to use them? [ex. code token, code id_token, code id_token token etc. | Mobile Connect uses the response_type as “code” only for DI mode. Refer to the Mobile Connect Device-Initiated OIDC Profile [9]. |
| **[98]** | What are the differences between OAuth2.0 authorization request and a Mobile Connect OIDC Authorization Request? | OIDC provides a simple identity layer on top of OAuth 2 which results in the generation of an ID Token that provides an authentication context as well as an Access Token. The ID Token also acts as a security token allowing the Access Token to be validated. Mobile Connect is based on OIDC. MC services are defined using the `scope` parameter and does not allow the use of the `claims` parameter to specify a subset of attributes – these are defined in MC service definitions based upon the use of a specific `scope`. The `claims` parameter in the OIDC Authorization Request is supported for MC attribute match services to allow attributes and their values to be asserted. However, this has been |

| No | Query | Response |
|---|---|---|
| | | superseded with the move to a split architecture and the use of the `mc_claims` parameter in the Resource call. |
| **[99]** | Are there any other redirect codes other than 302 supported in MC? | It is strongly recommended to use the 302-redirect code in the Mobile Connect DI mode authorization response, as the other redirect codes have implications on Authorization Code leakage through User Agent caching. |
| **[100]** | We believe more error codes are needed for our implementation, what is the process to add more error codes without affecting interoperability? | It is recommended that any requirements for additional error codes should be raised within the Mobile Connect technical forums, so that they can be included as part of the standard specifications. |
| **[101]** | Is it allowed to publish SI mode Authorization Endpoints through HTTP instead of HTTPS? | The SI mode Authorization Endpoint must use HTTPS. |
| **[102]** | Can the ID GW publish polling end points through HTTP rather than HTTPS? | Polling endpoints must use HTTPS. |

## 4.6 Authorization Server

| No | Query | Response |
|---|---|---|
| **[103]** | What is the difference between response_type and grant_type how are they used in DI mode and SI mode? | The response_type is used in the Authorization call to indicate which authorization process flow needs to be used. For Mobile Connect DI mode, the response_type value must be “code”. For Mobile Connect SI mode the response_type value must be “mc_si_async_code” where Token retrieval is via Notification or “mc_si_polling” where Token retrieval is via Polling<br><br>The grant_type is used in the Token Request to indicate which grant to use for the Token call. For Mobile Connect, the grant_type value must be `“authorization_code”` for DI mode or "urn:openid:params:mc:grant-type:server_initiated" for SI mode using Polling. |
| **[104]** | How is redirect URI matching performed? | The `redirect_uri` passed in the Authorization request in DI mode must match one of the `redirect_uri`s registered for the application during application onboarding |
| **[105]** | Does MC allow HTTP and HTTPS schemes for `redirect_uri`? | HTTPS must be used for the SPs `redirect_uri` for security. |
| **[106]** | Are there any cases in MC where an http scheme can be used for a redirect? | When header enrichment is used, HTTP is used as the first call (as header enrichment happens in HTTP and not HTTPS, unless an alternate approach is used for capturing the authenticated MSISDN from the core network). Redirects must use HTTPS. |

| No | Query | Response |
|---|---|---|
| **[107]** | Can POST be used for DI mode OIDC Authorization Requests? Is there an example? | Yes, DI mode OIDC Authorization Requests support both GET and POST.<br><br>Example request:<br><br>`POST /authorize HTTP/1.1`<br>`Host: mno1.example.com`<br>`Content-Type:application/x-www-form-urlencoded`<br><br>`response_type=code`<br>`&scope=openid`<br>`&client_id=s6BhdRkqt3`<br><br>`&redirect_uri=https%3A%2F%2Fclient.example.org%2Fcb` |
| **[108]** | Is it possible to use the same Authorization Endpoint for both DI mode Authorization Endpoint and SI mode authorization end point? | It is possible to reuse the same Authorization Endpoint for both DI and SI mode; the semantics are slightly different – so the implementation behind the endpoint for DI and SI mode needs to consider that. E.g., for DI mode, a holding page is displayed to the User and the response is returned as a redirect at the registered `redirect_uri` with the `authorization_code`, whereas the response in SI mode is an acknowledgement and no holding page is generated and no `authorization_code` is generated. |
| **[109]** | Does the asynchronous SI mode protocol offer better performance compared to the DI mode synchronous protocol? | SI mode allows the SP to implement using a non-blocking design approach, so that the SP gets an acknowledgement immediately after the authorization call is been sent to the ID GW and can reuse the resources (thread, port etc.) without blocking them. At the same time, the SI mode uses fewer calls without any redirects involved. |
| **[110]** | We are implementing Authenticate services only, but we plan to implement the support for 'SP provided text' through the “context” parameter for a better End-User experience. Is that possible? | Yes, the context parameter is optional but can be used to allow the SP application to provide contextual information to be displayed in the authentication prompt for a better User experience. |
| **[111]** | We would like to combine authentication/authorisation `scope`s with attribute services; What is the best practice to do that? | The `scope` parameter values are space separated. It is possible to include multiple `scope` values in the authorization request and then the Access Token generated can be associated with the attributes in the requested `scope`, so that the SP can then call the resource endpoints for the attributes service associated with the requested `scope`s, passing the Access Token. |

### 4.6.1 OIDC Authorization Request and Parameters

| No | Query | Response |
|---|---|---|
| **[112]** | How are the `scope` values used in Mobile Connect? | Mobile Connect defines specific `scope` values for each Mobile Connect service. Based on the `scope` value(s) the ID GW Authorization Server interprets how the request is handled, e.g. SP entitlement for the service, consent requirement, SLA, requests for attributes from the Resource Server, etc. |
| **[113]** | Previous version of MC profiles allows "openid" stand alone string for MC authentication. When implementing profiles with the latest version, is it mandatory to keep backwards compatibility? | Yes. This is required to maintain backward compatibility with v1.1 and is required for supporting SPs who are still using v1.1 API calls. |
| **[114]** | What are the version values for different profiles? I am using new profiles, without version I would like to reject the request? Is it allowed? | Refer to the Mobile Connect Technical Architecture and Core Requirements document [8] for the version parameter values. When the version parameter is not included in the request, then the request must be treated as Mobile Connect Authentication request for backward compatibility – refer to Mobile Connect OIDC DI Profile [9]. |
| **[115]** | How does the `state` parameter help in maintaining the state between different calls involved in the DI mode? | The SP sends the mandatory `state` parameter in the OIDC Authorization Request. The ID GW, after successful authentication, returns the `authorization_code` to the registered `redirect_uri` of the SP and includes the `state` parameter. The SP uses the `state` parameter to correlate the authorization request sent to the ID GW with the `authorization_code` received through redirect. |
| **[116]** | How is `state` used to prevent cross-site request forgery? | XSRF (Cross Site Request Forgery) is done by forcing a client to follow a misleading link that can allow an `authorization_code` of an attacker to be injected instead. The state parameter acts as a session_id for the client, which the client must check when receiving the `authorization_code` back from the ID GW as a redirect. |
| **[117]** | How are `state` and `nonce` parameters used to mitigate replay attacks? Are there any best practices? | The `state` + `nonce` parameter uniquely identifies a transaction and any replay of the same `state` + `nonce` value must be rejected at the ID GW to prevent any replay attack. |
| **[118]** | `max_age` : are there any scenarios in which `max_age` is used in an MC request? With this parameter there are two possibilities:<br>[1] always authenticate the User by prompting each time - it works well<br>[2] do not authenticate the User – since he has already been authenticated. This might be a security loop-hole. Can MC define the best practice with examples? | It is recommended to authenticate the User each time. |

| No | Query | Response |
|---|---|---|
| **[119]** | As of today,the `id_token_hint` parameter is NOT used in any MC service - Why do we require this parameter, and how will it be used in MC? | The `id_token_hint` will be used in the future to pass the ID Token (including PCR and iss) as a User identifier. Currently the `login_hint` is used to pass the PCR. |
| **[120]** | Are there any situations for which the `id_token_hint` parameter would be ignored? | If `scope` contains "mc_authz", the `id_token_hint` value MUST be ignored. ID GW/Authorization server MUST always display an authorization prompt to the User for approval. |
| **[121]** | The `login_hint` parameter allows the encrypted MSISDN to be used and the encryption is carried out by the APIX and returned in the Discovery response. Is it possible for the MSISDN to be encrypted by the SP? | Currently, it is strongly recommended that the encryption is done by the APIX. In the future, SP encrypted MSISDN will be enabled as well – with a process for distributing keys to the SP and also identifying the source for the encryption (APIX or SP) so that the ID GW can apply the policies. |
| **[122]** | If the SP requested `acr_values` are not supported by the ID GW, what should happen? | The ID GW can authenticate the User to the maximum LoA (`acr_values`) that is supported. The achieved LoA is returned in the `acr` claim and Authenticator used in the `amr` claim in the ID Token. |
| **[123]** | The `binding_message` is optional. If the SP does not provide this parameter and the Operator ID GW also does not create one, is it allowed in MC? | The binding message helps the User to get assurance that the consumption and the authentication devices are part of the same transaction. Although it is not mandatory for all services to use the `binding_message`, it is strongly recommended to provide assurance to the User. Note the `binding_message` parameter is required when the `scope` contains “mc_authz” i.e. for MC Authorisation services. |
| **[124]** | Are there any limitations on the binding_message maximum length? Can the ID GW reject it if the length is exceeded? | There are practical limitations on the length of the `binding_message` parameter that arise from the available space of 220 bytes to ensure global interoperability (based on a SIM-Applet Authenticator). This needs to accommodate `client_name`, `context`, if present, and `binding_message`, if present. Guidance on practical `binding_message` length should be provided to Service Providers. There is no length restriction on the `binding_message` but an Operator can truncate the message or can reject the `binding_message` based on local policies |
| **[125]** | The `claims` parameter is used in KYC specifications, are there any plans to move this claims parameter to mc_claims in the resource request? How can interoperability be achieved? What is the best practice? | KYC Match will continue to use the `claims` parameter. Mobile Connect is now using the `mc_claims` parameter in the Resource Request and all new services are likely to use this approach. |
| **[126]** | What is the best practice to implement `correlation_id`? GUID is the best option for | The `correlation_id` can be generated as a random number with sufficient entropy. The `correlation_id` is |

| No | Query | Response |
|---|---|---|
| | uniqueness, but, due to some browser's length limitations this may not be appropriate - What is the best practice? | used for correlating transactions between the APIX and the ID GW, and it should not be used as a security parameter. |
| **[127]** | How is the `correlation_id` processed at the ID GW? | The `correlation_id` is used to correlate the transactions between the APIX and the ID GW – for debugging, auditing and other consolidation purposes. The `correlation_id` is passed by the SP in the request and needs to be included in the responses from the ID GW. |
| **[128]** | How is the request object created in SI mode? | The request object in SI mode must be a signed JWT. |
| **[129]** | How should the `nonce` parameter be implemented? | The `nonce` should be created as a random number with enough entropy, so that it is unique, difficult to guess and also mitigates against replay attacks. |

### 4.6.2 Device-Initiated Mode

| No | Query | Response |
|---|---|---|
| **[130]** | What are the endpoints that I need to publish to support DI mode? | Authorization, Token and the Resource endpoints (PremiumInfo and / or service specific Resource endpoints) need to be published via the Mobile Connect Provider Metadata (openid-configuration URL) – See Mobile Connect Technical Architecture and Core Requirements [8] |
| **[131]** | When supporting DI mode authentication only, is there a need to publish any resource end points like the PremiumInfo? | No, Resource Endpoints are not needed. |
| **[132]** | What are the supported HTTP methods for DI mode OIDC Authorization Requests in MC? | Both GET and POST are supported. |
| **[133]** | What are the grant types supported in DI Mode? Why is the Authorization Code grant type necessary? Why is the implicit grant type, as defined in OIDC, NOT supported? | Mobile Connect only supports the "`authorization_code`" grant type for DI mode.<br>The "`authorization_code`" grant type enables the initiation of the request from a device and at the same time sharing of secure tokens (Access Token, ID Token) between the SP server and the ID GW.<br>The Implicit grant type is not suitable for mobile device originated secure calls. |
| **[134]** | What are the security risks associated with DI mode redirect mechanism and how are they mitigated? | The DI mode uses the "Authorization Code Flow" , so that the Authorization Code is returned to the SP through redirect at the `redirect_uri` of the SP. As a MITM injection – there is a risk that the `redirect_uri` can be changed in the authorization request – to mitigate this – the ID GW must verify that the `redirect_uri` used in the authorization request is one of the registered `redirect_uri` values for the SP app. |

| No | Query | Response |
|---|---|---|
| **[135]** | Which HTTP method (GET or POST) is used for a Token request in DI mode and why? | POST is used for the Token request, as this is a server-initiated call and also a secure call protected using `client_secret`. |
| **[136]** | In MC DI mode, why is the Token request always from the server? Why can't we initiate the request from the SP application instead of server? | Mobile Connect DI mode uses the Authorization Code flow, where the Token request needs to use the `client_secret` so that the Tokens (Access Token, ID Token) is shared with authenticated clients. The Consumption Device may not be sufficiently secure to store the `client_secret`, hence the Token request is always initiated from the server – where the `client_secret` can be stored securely. |
| **[137]** | In DI mode for the MC Token Request which client authentication method is used? How can I pass `client_id` and `client_secret` in the header or request body? | The Mobile Connect Token request uses "client_secret_basic" as the client authentication method, where the `client_secret` is passed using the HTTP Basic Authorization scheme. |
| **[138]** | Which encoding scheme is used for MC authorization requests in DI mode? what should happen if request is NOT encoded? | For Authorization Requests, the "application/x-www-form-urlencoded" encoding scheme is used. When the request parameters are not encoded – an error should be thrown as per the Mobile Connect Technical Architecture and Core Requirements Document [8]. |
| **[139]** | What are the implementation steps to validate a DI mode Token request | Here are the implementation steps for validating the Token request at the ID GW for DI mode:<br>Check that the "client_secret_basic" client authentication mode is used<br>Validate the `client_secret`<br>Validate the `authorization_code`<br>Validate that the mandatory parameters are present |
| **[140]** | Why are short-lived authorization codes recommended rather than long lived authorization codes? | The authorization code should be a single-use code and an attempted second use must result in revoking the issued tokens. The `authorization_code` is returned to the SP using redirect at the `redirect_uri` which involves the UA/device, where there is a potential for leakage of the `authorization_code`. |
| **[141]** | DI mode errors codes must be returned through redirect 302 from the Authorization Endpoint. Is it possible to return errors with other 3xx responses? | Mobile Connect strongly recommends the usage of redirect code 302, as other redirect codes may be susceptible to `authorization_code` leakage through caching at the UA. |
| **[142]** | In DI mode, are there any types of error that the ID GW Authorization Endpoint must return in a non-redirect mode (i.e. not using 302)? | When the `client_id` is not recognised, the registered `redirect_uri` cannot be used, in that case the errors must be returned using 4xx error codes. |
| **[143]** | When an IP address-based Discovery request is made (i.e. the Consumption Device and Authentication Device are the same mobile and it is using the mobile | This can happen if, and only if two Operators shared this MSISDN, this is a registration problem and rarely or never occurs and is outside of the `scope` of Mobile Connect. |

| No | Query | Response |
|---|---|---|
| | data channel), the wrong operator metadata is returned | |
| **[144]** | When an IP address-based request is made (i.e. the Consumption Device and Authentication Device are the same mobile and it is using the mobile data channel and header enrichment or similar is enabled), the core network does not introduce any MSISDN | If the MSISDN is not available to the ID GW, then the proper implementation will be that the ID GW presents a page to the User in order to capture MSISDN when an OIDC Authorization Request is made. |
| **[145]** | When an IP address-based request is made (i.e. the Consumption Device and Authentication Device are the same mobile and is using the mobile data channel and header enrichment or similar is enabled), the core network introduces a wrong MSISDN | This is rarely happens, but as defined in the MC Device-Initiated OIDC Profile [9], the IDGW should return an error (“access_denied”). |
| **[146]** | How should the ID GW handle Corporate numbers (MSISDNs) | Where an MSISDN is not associated with an individual (because it is assigned to a Corporate account), the ID GW should treat it as an invalid number and reject the request (“access_denied”). |

### 4.6.3 Server Initiated Mode

| No | Query | Response |
|---|---|---|
| **[147]** | What Endpoints need to be published for SI Mode – Notification? | SI Authorization Endpoint needs to be published (si-authorize) along with any Resource Endpoints for the ID GW. An SP will need to publish the notification endpoint(s) where SI mode uses notification for token retrieval. Refer to the Mobile Connect OIDC SI Profile [10]. |
| **[148]** | What are the endpoints needed to publish for SI Mode – Polling? | SI Authorization Endpoint (si-authorize) and the Polling endpoint along with any Resource Endpoints. Refer to the Mobile Connect OIDC SI Profile [10] |
| **[149]** | What are the supported HTTP methods among GET and POST used in SI mode authorization request in MC? | Only POST is used for an SI mode OIDC Authorization Request. |
| **[150]** | What values of `response_type` are supported in SI mode notification and polling? | • For Notification: “mc_si_async_code”<br>• For Polling: “mc_si_polling” |
| **[151]** | What are the advantages and disadvantages of SI mode notification? | Advantages:<br>• Resource optimised for SP, as the SP does not need to actively request for the Tokens<br>• SP needs to call just one endpoint – SI Authorization Endpoint |

| No | Query | Response |
|---|---|---|
| | | Disadvantages:<br>• SP needs to have a listener implemented<br>• SP needs to have a mechanism to open up their infrastructure securely for the ID GW to send the notification |
| **[152]** | What are the advantages and disadvantages of SI mode polling? | Advantages:<br>• Better control for the SP, as they call the Polling endpoint to get the Tokens<br>• SP does not need to implement a listener<br>Disadvantages:<br>• SP needs to implement a scheduler-based Polling mechanism<br>• Ties-up Resources as SP needs to manage Polling at certain intervals until the Tokens are received |
| **[153]** | What are the semantics of the Polling request to polling end point? | Refer to the Mobile Connect OIDC SI Profile [10] |
| **[154]** | What should happen if the ID GW receives an error as notification acknowledgement? | • ID GW should never respond back to SP after receiving an error in the notification acknowledgement.<br>• ID GW should log the error<br>• It must not redirect if it receives 3xx error codes – this represents a potential, security loophole.<br>• It must REVOKE the already issued tokens, other than for a “server error” (i.e. the issued tokens will become invalid). |
| **[155]** | What is the best practice in mitigating the race conditions between SI Authorization Endpoint acknowledgement and notification? | It is recommended that a queue is implemented to process the SI requests / acknowledgement / notifications at the ID GW. |
| **[156]** | How is a Polling request authenticated? | The Polling request uses the “private_key_jwt” method for client authentication. The signature must be verified using the public key retrieved from the `jwks_uri` of the SP. |
| **[157]** | How is the acknowledgement for the Notification returned by the SP to the ID GW? | The SP acknowledges successful receipt of the Notification using HTTP 200 OK or 204 No Content (recommended). In the case of an error an appropriate error code will be returned (See [10] ) |
| **[158]** | What are the implementation steps to validate a SI mode polling request? | The validation steps for the SI Polling request are:<br>• Check that the SP is registered for Polling, otherwise return an error<br>• Validate the signature of the request, using the public key from the `jwks_uri`<br>• Validate that the mandatory parameters are included in the Polling request |

| No | Query | Response |
|---|---|---|
| **[159]** | What are the mandatory parameters that a polling response header should contain? | Refer to the Mobile Connect SI OIDC Profile [10]. |

### 4.6.4 Token Retrieval

| No | Query | Response |
|---|---|---|
| **[160]** | Is it mandatory for the ID GW to return an Access Token in the case where MC Authenticate services are only supported - where the Access Token is never used? | Yes, this is needed to comply with the OIDC specifications. |
| **[161]** | What is the difference between the MC Access Token (based on OIDC) and the Access Token referenced in OAuth2.0 specifications? | The purpose of the Access Token is exactly the same – it is an opaque Bearer token that grants access to protected resources (attributes) from a Resource Server. In line with OIDC, the SP can validate the Access Token against the at_hash claim within the ID Token. |
| **[162]** | What is the difference between one-time valid authentication, time bound authentication? Are they available in MC? What is the best practice? | The ID Token includes an `exp` (expiration time) after which the ID Token is no longer valid. This allows the ID Token to be valid for a period of time during which if an OIDC Authorization Request is submitted a User authentication or authorisation is not required. In practice it is best to keep the lifetime of the ID Token to as short a time as possible (this is REQUIRED for MC Authorisation) and to re-authenticate if required. |
| **[163]** | Can the ID Token be signed and encrypted in MC? | ID Token must always be signed (as ID Token is a JWT), an encrypted ID Token is optional. The ID Token must be signed first before encryption |
| **[164]** | How is the Token response validated by the SP application? | The Token Request is a server-to-server call and the response is received by the SP server. The Token response uses JSON encoding, so the SP should validate the response by extracting the JSON payload, checking the mandatory parameters as per the Mobile Connect OIDC Profiles. |
| **[165]** | How is the ID Token validated in DI mode? | The steps for validating the ID Token are:<br>• Check the ID Token is formatted as a JWT<br>• Extract the individual parts of the ID Token: the header, body and the signature<br>• Validate the signature using the algorithm mentioned in the header and retrieving the key from the jwks_uri in the MC Provider Metadata<br>• Extract the JSON payload from the body<br>• Check the mandatory claims are included in the ID Token as per the Mobile Connect OIDC Profiles |

#### 4.6.4.1 ID Token

| No | Query | Response |
|---|---|---|
| **[166]** | When authentication is not performed by the ID GW, what should the values for the `acr` and `amr` claims be? | `acr`=0 and `amr`="NO_AUTHN" |
| **[167]** | What is the best practice format for the `sub` identifier in MC.? | The recommended format for PCR is a GUID (See [8] ) |
| **[168]** | The `aud` claim is defined as a single case sensitive string or an array of strings [with one or many] - what format is recommended for interoperability? | An ID GW implementation must support both options |
| **[169]** | If the `exp` value in the ID Token is set for one hour to expiry, does this mean, that for the same MSISDN from same SP application/service, the ID GW may not perform authentication? | Yes, the ID GW may not perform explicit authentication, if the ID Token for the context is still valid – depending on the Operator's policy. |
| **[170]** | The `iat` and `auth_time` values in the ID Token show a small difference. Is it possible to set them to the same time or does MC always expect them to be different? | The `iat` is the time that the ID Token was generated whereas `auth_time` is the time of authentication; they will be different if the time of authentication is known through the Authenticator sub-system, otherwise they can be set as the same value. (Note that `iat` must never be earlier than `auth_time`). |
| **[171]** | If the Access Token is null what should be the value of the `at_hash` value? | Access Token cannot be set to null, even if the `scope` used is Authentication and Authorisation service categories. |
| **[172]** | What is the process to register a new `amr` value to reflect a different authenticator – for example my authenticator is registered as an approved authenticator which uses biometrics and I would like to register an amr value for this authenticator. | Any requirement for adding new `amr` claims should be raised with the Mobile Connect technical forums so that it can be included in the Mobile Connect technical specifications, enabling interoperability. |
| **[173]** | The `azp` value is mandatory if the audience of the ID Token is different to the authenticated client. What is the best practice for interoperability and compliance? | The `azp` may be included in all instances. |
| **[174]** | For the `hashed_login_hint` claim, the mandatory algorithm is SHA256. Are other hashing algorithms allowed? | Other hashing algorithms are allowed – these will be published in the ID GW MC Provider Metadata. Refer to the Mobile Connect Technical Architecture and Core Requirements [8] for guidance on hashing algorithm usage. |

| No | Query | Response |
|---|---|---|
| **[175]** | How is the ID Token signature verified? how the signature keys are shared between SP application and ID GW? | The ID Token signature can be verified by the SP app using the keys published via the `jwks_uri` in the MC Provider Metadata. |
| **[176]** | What is the significance of `recipient` claim in the ID Token in SI mode? | The `recipient` claim returns the notification_uri used for returning the tokens and it acts as an integrity measure to re-confirm that the correct notification_uri was used and that the tokens are intended for that notification_uri only. |
| **[177]** | What is the significance of the `iss` claim in the ID Token? | The `iss` claim identifies the Operator as the Mobile Connect provider and this must match the issuer as received from the Discovery response. The PCR (`sub` claim) along with the `iss` claim should be used as the unique User reference for the User. |
| **[178]** | What is the significance of `sub` (Subject - PCR) and `iss` (issuer) claims – are they reliable? | Yes. The PCR (`sub`) provides a consistent User reference for the SP app – which is privacy protected through pseudonymisation. `iss` provides the mechanism for the SP to check if the response was actually generated by the Operator. |

#### 4.6.4.2 Access Token and Refresh Token

| No | Query | Response |
|---|---|---|
| **[179]** | Do I need to revoke the Access Token after receiving an error through notification acknowledgement in SI mode? Do I need return an error that invalidates the Access Token? | The Access Token should be valid for the duration it is created for or for a number of uses as per the policy (e.g. if the Access Token is valid for a single usage) then it should be discarded after the usage. In the event of an error response in the Notification Acknowledgement from the SP, then for all errors other than SP “server_error”, the tokens (ID Token, Access Token, Refresh Token) must be discarded. |
| **[180]** | How is the Access Token validated? How is the `at_hash` value in the ID Token generated? | The OIDC Core Specification, Section 3.1.3.6 [1] describes the process of generating the Access Token hash value (at_hash). The Access Token can be hashed using the same procedure and compared with the value in the ID Token `at_hash` claim. |
| **[181]** | Can the Access Token be a JWT? | The Access Token is an opaque token and the implementation is decided by the Operator. It can be created as a JWT as well. |
| **[182]** | What is the advantage of issuing short-lived Access Tokens? | Short lived Access Tokens helps to mitigate against token leakage and also mitigates against risks when the consent is revoked, and the Access Token needs to be invalidated. The Access Token can be renewed using the refresh token anyway – if supported. |
| **[183]** | Can I use `scope`d Access Tokens? is it allowed? | In Mobile Connect, the Access Tokens are always associated with a specific `scope` and the Resource Servers must check that the `scope` associated with the resource is associated |

| No | Query | Response |
|---|---|---|
| | | with the Access Token used in the resource call as bearer token. |
| **[184]** | The use of a Refresh Token is optional, and no MC service requires a Refresh Token. Our ID GW does not return a Refresh Token as defined in the specification - is this OK? | This is ok, although in the design – it should be considered how to generate a refresh token and renew short-lived Access Tokens using the refresh token. Note that a number of MC services must not use a Refresh Token due to the nature of the service. |

# 5 Resource Server and Attribute Services

| No | Query | Response |
|---|---|---|
| **[185]** | What is the difference between the MC Resource Request / Response and the OIDC Resource Request Response? | The MC Resource Request / Response is based upon the OIDC Resource Request / Response (UserInfo call). The MC Resource Request may include the `mc_claims` parameter to allow the SP to assert attributes and their values for match services which is Mobile Connect specific. MC also supports the provision of different Resource endpoints including the MC PremiumInfo endpoint. |
| **[186]** | Which endpoints are mandatory? E.g. PremiumInfo | To support attribute services, it is necessary to specify appropriate Resource Endpoints. The PremiumInfo Endpoint is a Mobile Connect Specific Endpoint that can support all attribute services or service specific Endpoints can be specified. The Endpoints are published via the Operator's ID GW Provider Metadata (See Mobile Connect Technical Architecture and Core Requirements [8]) |
| **[187]** | How can SP accept the resource response as a valid response? what is the significance of 'sub' value provided in the resource response when OIDC provided Access Token is used? | In Mobile Connect, the resources (when PII is used in the response) need to include the "sub" as one of the attributes, so that the resource response is tied with the ID Token and the User in this context. |
| **[188]** | What is JSON serialization is it mandatory in resource response? | Yes, JSON Serialization is the serialization approach for structured data. Mobile Connect uses JSON serialization for the Resource Responses. |
| **[189]** | My current OIDC implementation returns all attributes through the ID Token, to comply with MC do I need to migrate to return all attributes through resource end point? Why? What are the advantages? | Mobile Connect recommends returning User attributes using resource endpoints. |
| **[190]** | OpenID Connect has a feature to return attributes through ID Token, why is it not allowed in MC? Is | Mobile Connect is based around a split architecture where the Authorization Server manages DI and SI OIDC Authorization Requests and Responses and Token Retrieval and a Resource Server manages Resource Requests and |

| No | Query | Response |
|---|---|---|
| | there any advantage or security benefit for not allowing it? | Responses – this leads to a more flexible architecture that can support a number of different deployments within a market (See Mobile Connect Resource Server Specification [11]). |
| **[191]** | What are the guidelines for optional attributes, if the values are not available? | The support for optional attributes is up to the Operator implementing the ID GW and supported optional attributes can be returned in response to an MC service request on this basis. If data is not available for a User, then the attribute should be returned with an empty value. |
| **[192]** | What is the best practice for customizing the attribute list by the SP application, using the claims parameter? | The `scope` parameter value specifies the MC services required and, therefore, the attributes that are available, based on what is supported by the ID GW (REQUIRED attributes plus supported OPTIONAL attributes). For attribute share services all the supported attributes will be returned, subject to User consent.<br>For attribute match services and where the use of the `claims` parameter is specified, an SP includes an assertion for the attributes for which it has data, and which are supported by the Operator (i.e. REQUIRED plus supported OPTIONAL attributes).<br>If an attribute is not supported in a specific implementation (match service), it will be ignored. If an attribute is supported but the data is not available for that User then the Resource Server will return a response “match failed, data not available” |
| **[193]** | Using the MC profile `claims` parameter, if the SP requests a sub-set of attributes, how should the ID GW should respond? | For match services that support the use of the `claims` parameter, the SP submits attribute names and values for the attributes for which it has a value and the match service will respond with the match result.<br>For share services, the use of the `claims` parameter to specify a sub-set of attributes to be returned is not allowed in Mobile Connect. |
| **[194]** | Mandatory attributes values are not available, do I need to return empty values, OR does the ID GW omit those attributes? | The ID GW must support the REQUIRED attributes in order to be able to offer the MC service as defined in the relevant service “Definition and Technical Requirements” document. If for some reason a Resource Response cannot return a supported attribute then for a User then the attribute should be returned with an empty value.<br>At a practical level, if an Operator is not to able to support the REQUIRED attributes at service launch, they must add the support for REQUIRED attributes to their development roadmap and can return empty values temporarily. |
| **[195]** | What are the logging guidelines, if any, where mandatory attribute values are not available? | If a supported attribute is not available, then this should be logged. Service-specific “Definition and Technical Requirements” documents provide guidance on transaction logging. |

| No | Query | Response |
|---|---|---|
| **[196]** | Which claims are supported in MC? Normal claims, Aggregated claims or Distributed claims? | Only Normal Claims are supported in Mobile Connect – i.e. Claims that can be asserted directly by the ID GW. Aggregated or distributed claims may be supported at an Operator' |
| **[197]** | What is consent? Why it is required? What are the different ways to capture the consent? | The Mobile Connect Privacy Principles [24] require that no personal information relating to the User or their mobile subscription is shared with a Service Provider without the User's consent. The attributes to be shared are specified within the relevant Mobile Connect service "Definition and Technical Requirements document. Consent can be captured by the ID GW or by a Service Provider depending upon the MC service requested and the contractual terms between the SP and the Operator ID GW. Based upon the Operators' policy the ID GW can enforce capturing User consent by the ID GW if required. |
| **[198]** | What are the guidelines when Operator acquires the consent? | The Mobile Connect Product Manager's Lifecycle Handbook [12] contains guidance on capturing User consent |
| **[199]** | Is it possible to provide time bound consent? i.e. consent is valid for one day etc.? What are the security draw backs? Why has MC mandated transaction-based consent [if captured]? What are the advantages of it? | It is possible to provide long-lived consent. This is established by the Operator's ID GW policy depending upon the specific MC service being requested and the contractual arrangements in place with a Service Provider. It can be implemented by returning a long-lived Access Token or using a Refresh Token in conjunction with the Access Token. An SP will be required to keep the Access Token secure during its lifetime. It is important to ensure that a User is aware of long-lived consent being in place and that the facility for the User to revoke consent is in place. The advantages of consent on a per transaction basis makes this more straightforward. (See Mobile Connect Product Manager's Lifecycle Handbook [12] for more details) |
| **[200]** | How can a revocation mechanism be implemented to allow the User to withdraw consent for a specific service in an ID GW? What is the best practice? For which MC services revoke mechanism is possible? | User consent is relevant for all Mobile Connect services where User attributes are shared or validated with a Service Provider. If User consent is captured by the Service Provider, then the Service Provider should provide the ability for the User to revoke consent. This should ideally be communicated with the ID GW. If the ID GW is capturing consent, then it is recommended that the facility to revoke consent is provided via the Operator's self-service portal or similar. The status of consent for specific MC services with specific Service Providers should be recorded and available when validating SP requests. See the Mobile Connect Product Manager's Lifecycle Handbook [12] for more details. |
| **[201]** | ID GW would like to capture consent on mobile device only, What is the best practise to capture this consent for all attributes? How do the space limitations of authenticators impact my | The ID GW policy can be set to ensure that User consent is only captured on the mobile device. There are two options:<br>• Use the Authenticator to display the relevant information for which consent is requested and to gain that consent<br>• Authenticate the User via the Authenticator on the mobile device and then display the relevant information for which consent is requested and to gain that consent via the SP |

| No | Query | Response |
|---|---|---|
| | implementation and what are the guidelines? | application or the web browser (Device-Initiated mode only). This potentially provides more scope to display sufficient detail at the expense of additional steps in the consent process.<br>• The choice of Authenticator puts limitations on what information can be displayed on a single screen page which will affect the level of detail that can be displayed to enable informed consent and to avoid excessive page scrolling. |

# 6 Implementation of Non-Functional Requirements

| No | Query | Response |
|---|---|---|
| **[202]** | Is it always necessary to keep the transaction logs at an ID GW? How long do they need to be kept? | It is strongly recommended to log the transactions at the ID GW and also at other required components (e.g. Authenticator sub-systems like SMSC, USSD GW, MSSP etc.). This is needed for consolidation, auditing, dispute resolution, debugging, regulatory requirements and also invoicing (where applicable). The period of retention for transaction records depends on the Operators' operational policies, subject to local data retention laws. |
| **[203]** | What is the best practice for logging and auditing of transactions in MC? | The transaction logs should at least include the state, nonce, `correlation_id`, `client_id`, `scope`, PCR, timestamp of the transaction events, and status of transaction. Further details can be found in specific service "Definition and Technical Requirements" documents (See "References" section 0 ). |
| **[204]** | What is the best practice from MC for high availability and scalability? | The ID GW should be implemented in a way that it is horizontally scalable to accommodate the increase in load and performance requirements. Scalability should not be purely based on hardware upgrades only. The ID GW should also be deployed in a high availability mode, with active/passive standby systems. |
| **[205]** | What is the best practice for protection against man in the middle attack? | The use of TLS protection for the endpoints is key. Along with that, the usage of the hashed_login_hint claim in the ID Token helps mitigate MITM attacks from the User agent. |
| **[206]** | How can brute force attacks against the ID GW be prevented? | The ID GW must always authenticate SP clients to verify their entitlement to requests an MC service. Multiple client authentication failures should be considered in fraud management processes (e.g. after a configured number of failed client authentication failures from the same source IP, in a particular time-window in quick succession etc. the requests can be throttled). |
| **[207]** | What is the best practice for session fixation attack prevention? | Mobile Connect interfaces do not use web sessions and the authentication response is included in a signed JWT (ID Token), it is recommended that the tokens in the response are not included in session ids/cookies. |

| No | Query | Response |
|---|---|---|
| **[208]** | OAuth2.0 vulnerabilities lead to the risk of account takeover i.e. an attacker being able to sign into a victim's mobile app account and take control of it - How does MC (based on OIDC) prevent this? | Mobile Connect uses out-of-band authentication, so that the authentication challenge is delivered to the User using a different channel than the consumption channel (even if the physical device for service consumption and authentication is same) - |
| **[209]** | In DI mode, what is the best practice to prevent attacks on Authorization Code grant flow? | The following steps represent “best practice” for securing DI mode implementations:<br>• TLS protection for all endpoints<br>• Client authentication and access control<br>• Authorization code delivery at the registered `redirect_uri` only<br>• Authorization code usage control (one-time usage, time bound usage)<br>• Token (Access Token, ID Token) delivery through TLS protected and `client_secret` protected server-side Token call<br>• Inclusion of `hashed_login_hint` claim in ID Token to mitigate MITM attack on login_hint<br>• Inclusion of the hash of the Access Token in the `at_hash` claim in ID Token<br>• Access control around the Access Token |
| **[210]** | What is the best practice to prevent the authorization server acting as an open redirector? | The ID GW must check that the `redirect_uri` passed in the Authorization request matches one of the registered `redirect_uri` values. |
| **[211]** | How should the ID GW mitigate against clients acting as open redirectors? | Mobile Connect ID GW must always return the Authorization Code to the registered `redirect_uri` and the `redirect_uri` should use HTTPS. Also, the final response with the tokens (Access Token, ID Token) are only shared in the Token call – protected using a `client_secret` and is a server-side TLS protected call. |
| **[212]** | What is the best practice for TLS terminating reverse proxies? i.e. application server sitting behind a reverse proxy, which terminates the TLS connection and dispatches the incoming requests to the respective application nodes? | If the TLS is terminated in a proxy before the request hits the ID GW, then the TLS terminating proxy and the ID GW must be in the same security domain – so that the ID GW can trust the request parameters arriving at the endpoints for processing. |
| **[213]** | `state` and `nonce` in DI mode are used to mitigate some security attacks.In SI mode `state` does not exist - how is `nonce` used to mitigate security attacks? | The “state” parameter is used for correlating back the Authorization Code via the `redirect_uri` with the original Authorization request in DI mode and it is also used to protect against XSRF (Cross Site Request Forgery). In the SI mode – the request is initiated from a server and redirect is not used – so the requirement for the state parameter is not there. The nonce is still used for protecting against replay attacks. |

| No | Query | Response |
|---|---|---|
| **[214]** | Does MC support native applications where a User-agent is not involved? | Mobile Connect can be used from native apps, to handle the redirects – a WebView is generally needed. |
| **[215]** | How can the ID GW mitigate the risk of client impersonation attacks? | To protect against client impersonation attacks - the ID GW must authenticate the client using the client credentials and also the Authorization Code must only be returned to the registered `redirect_uri`. |
| **[216]** | What is the best practice to prevent man in the middle attacks directed towards the authorization and token Endpoints? | TLS must be used for all endpoints. For DI mode, the `hashed_login_hint` claim must be included in the ID Token to mitigate any MITM injection in the UA to manipulate the `login_hint` SP clients must be authenticated at the Token Endpoint with the `client_secret`. |
| **[217]** | What is the best practice for Access Token and Authorization Code generation to prevent confidentiality guessing attacks? | The Authorization Code and Access Token should be generated as opaque tokens with a large entropy to make the guessing by brute force difficult. |
| **[218]** | In DI mode, what is the best practice to prevent Cross Site Request forgery? | The `state` parameter must be used in the OIDC Authorization Request and this must also be returned in the Authorization Response (via the redirect) to prevent XSRF (Cross Site Request Forgery). |
| **[219]** | What is the best practice to prevent clickjacking attacks? | Mobile Connect uses APIs to interface with the SPs, in cases where a User interaction is needed from a web page – it is recommended to use X-Frame-Options to prevent clickjacking attacks. |
| **[220]** | What is the best practice for an authorization server to prevent code injection and input validation? | The ID GW (Authorization Server) should ignore any request parameters that it does not understand, also it should check that the parameter values are as specified in the Mobile Connect OIDC Profiles. |
| **[221]** | In attribute services, what is the best practice to prevent token manufacture/modification? | The Authorization code and Access Token generation process should consider large entropy to ensure randomness so that these cannot be guessed or regenerated. After the usage or the expiry of the Authorization Code – it should be discarded for any further use. |
| **[222]** | How can bearer tokens leakage through token disclosure best be avoided? | Access Tokens should be discarded after the delivery of the resource. Also, the Access Token validity should always be checked at the Resource Server |
| **[223]** | What is the best practice, to prevent token replay? | Resource Endpoints must always be TLS protected and velocity throttling should be in place at the Resource server. |
| **[224]** | Why is it good to validate the TLS certificate chain? | It is important to check the TLS chain to identify the root CA and validate if it can be trusted. |
| **[225]** | Why is it not good practice to store Access Tokens in cookies? | Access Tokens are bearer tokens, so anyone obtaining the Access Token can claim the authorization to access protected resources (User attributes). Therefore, the Access Tokens should be stored in secure storage and cookies are not secure storage. |

| No | Query | Response |
|---|---|---|
| **[226]** | How can covert redirect be managed in MC? What is the best practice? | Covert redirect is mitigated in Mobile Connect using the Authorization Code Flow, where the possible compromise of the Authorization Code does not allow an attacker to get access to the protected resources (User attributes) as the resources are protected using the Access Token. Access Tokens are provided using server-side calls using the client credentials for client authentication (not using redirects). |
| **[227]** | How does MC architecture prevent third parties from obtaining client secrets? | The `client_secret` for the ID GW is delivered to the SP client during the Discovery process, also the period of validity of the `client_secret` can be configured. |
| **[228]** | How does MC prevent third parties from obtaining Access Tokens? | The Token Request to obtain Access Tokens needs to be called using `client_secret` and also the Authorization Code needs to be passed. |
| **[229]** | How can phishing by Counterfeit Authorization Servers be prevented? | Mobile Connect offers the Discovery service to provide the provisioned Authorization Server endpoints (ID GW) and also the client credentials, it is recommended that other sources should not be used to obtain the endpoints and client credentials for the Authorization Server. |
| **[230]** | How does MC prevent a malicious client from obtaining existing authorization by fraud (especially when an End-User is not authenticated)? | Mobile Connect recommends that the ID GW checks the entitlement of a client to use specific MC services and that flows are checked for each client. This would be managed through the ID GW Policy Management engine. In addition, it is recommended that the Access Token be short lived and, if required, that is can be renewed using a refresh token. |
| **[231]** | How are eavesdropping access attacks prevented? | Mobile Connect interfaces and endpoints use TLS. Also, no personal data, including security tokens - is shared through weaker mediums like User Agents. All resources are protected using an Access Token and Access Tokens are bound to individual Users. |
| **[232]** | How does MC prevent a malicious client from obtaining authorization? | The ID GW authenticates the client using client credentials and also the Authorization Code is returned to the registered `redirect_uri` |
| **[233]** | How can Authorization Code phishing attacks be prevented? | The ID GW authenticates the client using client credentials and also the Authorization Code is returned to the registered `redirect_uri` |
| **[234]** | What is the best practice to prevent User session impersonation? | Mobile Connect uses a stateless interaction. |
| **[235]** | How is the risk of Authorization Code leakage through a counterfeit client mitigated? | The Authorization Code is returned through the registered `redirect_uri` only, so that if the client is compromised – still the Authorization Code will be returned to the registered source. The Authorization Code is a also temporary code, the final responses – including the Access Token and ID Token are returned through the Token call – which is a server side call and requires the complete client credential (including `client_secret`) to be used for the call. |

| No | Query | Response |
|---|---|---|
| **[236]** | What is the best practice to mitigate resource owner impersonation? | The User should be authenticated before generating the Access Token which is then used to provide access to the resources. |
| **[237]** | How can DoS attacks that exhaust resources be prevented? | The ID GW should have velocity throttling to prevent DoS attacks. Also, the use of a firewall in the deployment architecture is recommended. |
| **[238]** | How can DoS attacks using manufactured Authorization Codes be preven? | The ID GW should have velocity throttling and also policy control to check for multiple invalid Authorization Code usage from a single source or within a small time-window |
| **[239]** | How can Access Token phishing by counterfeit resource servers be prevented? | The SPs must use the Resource Endpoints published in the MC Provider Metadata and must ensure that the MC Provider Metadata URL (OpenID Config URL) is constructed from the Discovery response. |
| **[240]** | What is the best practice for preventing leakage of confidential data in HTTP proxies? | HTTPS must be used for all endpoints |
| **[241]** | What is the best practice when implementing MC to prevent spam authentication requests to the End-User by providing random MSISDNs? | A “spam code” can be used to protect against spamming. The User can be asked to register a “spam code”, which can then be requested when the User initiates the request – preventing random MSISDNs from being used. |
| **[242]** | What actions need to be taken to mitigate security risks in Mobile Connect? | While Mobile Connect has been designed bottom-up to be secure there are a number of potential vulnerabilities that need to be taken into consideration during implementation:<br>• Operator vulnerabilities may include various threats such as DDoS attack or data leaks.<br>• A user may suffer social media attacks, OS incompatible bugs, malware, spam, etc.<br>• A Service Provider may also suffer from DDoS and phishing attacks.<br>The following Operator Side Security and Fraud Mitigation steps should be considered:<br>• DDoS Attack - The Operator should provide anti-DDoS solutions, such as an IPS system, to protect the ID GW and other components. The Operator should also consider DDoS mitigation solutions to ensure resilience of service. This should be applied on all the in-scopes systems and should be coupled with effective incident response capabilities.<br>• Data Leak - The Operator should provide data protection such as data encryption and access control mechanisms to keep the user’s personal information safe. Tools such as an Intrusion Detection System and/or Intrusion Prevention |

| No | Query | Response |
|---|---|---|
| | | should also be considered along with effective monitoring, detection, and incident management.<br><br>• Mass Spam and Target Spam - The Operator should provide antispam solutions by using second attributes such as location or using alias input such as MSISDN. Meanwhile the Identity Gateway should have the ability to detect abnormal patterns.<br><br>• SIM Cloning - A SIM card may be cloned and despite off network countermeasures this may allow for fraudulent registration of the service. The Operator should implement SIM cloning detection capabilities such as the use of volume, value, and velocity checking within their fraud management system.<br><br>• MSISDN Recycling - An abuse of MSISDN recycle/purge processes may create an opportunity for fraudulent registration services. The Operator should implement an internal audit process to tackle such issues.<br><br>• OTA - The Operator may use OTA campaigns to distribute updated SIM applications. A specific fraud risk includes the possibility to download the application to SIM profiles with known OTA vulnerabilities. OTA campaigns must be constructed to identify and reject downloads to SIMs with known vulnerable profiles.<br><br>• SMS Gateway and SMSC - An attacker sends spoofed SMS to customers using SMS Gateways or SMSCs as part of fake authentication processes to socially engineer the customer to believe they have authenticated to legitimate sites. The Operator should take precautions to unambiguously identify the SMS source. |

# Annex A Document Management

## A.1 Document History

| Version | Date | Brief Description of Change | Approval Authority | Editor / Company |
|---|---|---|---|---|
| V0.1 | 26/06/2018 | Initial Draft | David Pollington / GSMA | Siva (Venkatasivakumar Boyalakuntla) / GSMA |
| V0.2 | 11/09/2018 | Amended questions for various topics | David Pollington /GSMA | Siva (Venkatasivakumar Boyalakuntla) / GSMA |
| V0.3 | 17/09/2018 | Amended service specific queries for all services | David Pollington / GSMA | Siva (Venkatasivakumar Boyalakuntla)/GSMA |
| V0.4 | 09/11/2018 | Updated with additional questions and completion of answers by NS and GH | David Pollington / GSMA | Nick Spencer |
| V0.5 | 19/11/2018 | Updated following review by Siva and additional updates | David Pollington / GSMA | Nick Spencer |
| V0.6 | 29/11/2018 | Consolidated with DP review and changes based on refinement of how attribute services are handled | David Pollington / GSMA | Nick Spencer |
| V0.7 | 11/12/2018 | Changed version numbers PRD information. | David Pollington / GSMA | Siva (Venkatasivakumar Boyalakuntla) / GSMA |
| V1.0 | 15/11/2022 | Changes on the coversheet to go for publication | TG | Yolanda Sanz/GSMA |

## A.2 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.