Mobile Connect device initiated OIDC profile

Version 3.0

03 May 2019

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

Security Classification: Non-confidential

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

Copyright Notice

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 to provide Authentication, Authorisation, Identity Services and Network Attribute Services to be used in conjunction with services offered to a User by Service Providers.

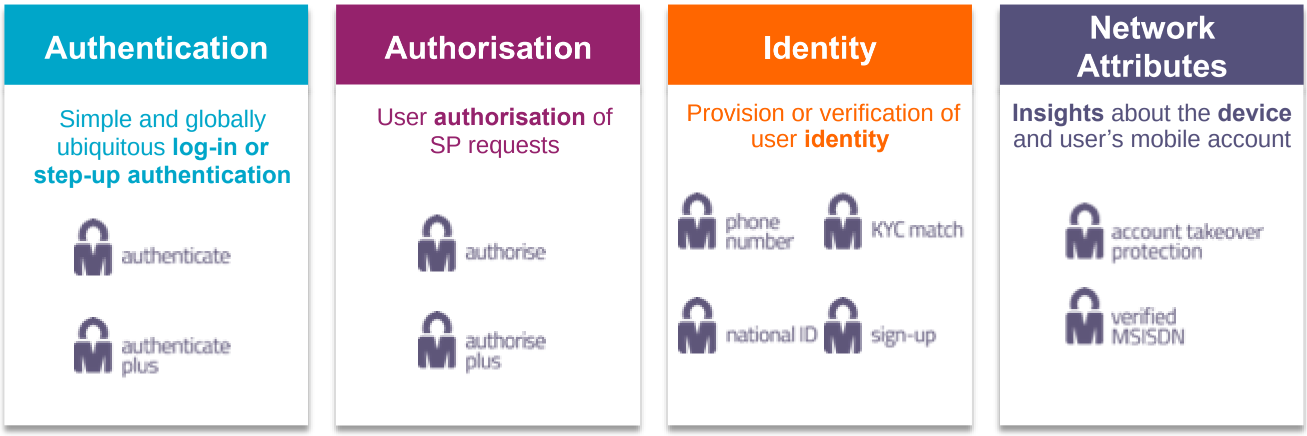
Authentication
Simple and globally ubiquitous log-in or step-up authentication
authenticate
authenticate plus
Authorisation
User authorisation of SP requests
authorise
authorise plus
Identity
Provision or verification of user identity
phone number
KYC match
national ID
sign-up
Network Attributes
Insights about the device and user's mobile account
account takeover protection
verified MSISDN

Figure 1: Mobile Connect Portfolio of Services

Mobile Connect is based upon the OpenID Connect (OIDC) protocol suite [1] and allows Users to be identified by their MSISDN (Mobile Station International subscriber Directory Number or a related Pseudonymous Customer Reference) and to provide Authentication, Authorisation and Consent via their mobile device.

The serving Mobile Operator supports and selects an appropriate Authenticator to present the Authentication, Authorisation and Consent request to the User on their mobile device to which the User responds. The Authenticator is selected based on Operator policy, device capability and the Level of Assurance required.

Mobile Connect also provides access to an enriched set of User attributes[1] provided by the Mobile Operator, that can be shared with a Service Provider (SP), 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 specification defines a Mobile Connect Device-Initiated API (Application Programming Interface) offered by an Operator's Identity Gateway (ID GW) to a Service Provider that enables a Mobile Connect service to be initiated by a User that is interacting with an SP application.

[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

1.2 Scope

In Scope	Out of Scope
• Mobile Connect Device-Initiated OIDC Profile for all Mobile Connect Services, including Error Responses and Error Codes	• Mobile Connect Server-Initiated Communication • Mobile Connect service specifications • TLS and mTLS implementation details

1.3 Abbreviations

Term	Description
API	Application Programming Interface
ASCII	American Standard code for Information Interchange
GUID	Global Unique Identifier
HTTP	Hyper Text Transport Protocol
HTTPS	Hypertext Transfer Protocol Secure
I&A	Identity & Attributes
ID GW	Identity Provider: The entity providing the authentication and Identify services, e.g. the operator
JSON	JavaScript Object Notation
JWA	JSON Web Algorithms
JWE	JSON Web Encryption
JWK	JSON Web Key
JWS	JSON Web Signature
JWT	JSON Web Token
LoA	Level of Assurance
OIDC	OpenID Connect
PCR	Pseudonymous Customer Reference
RFC	Request For Comments
RP	Relying Party: The application/service that needs the authentication and identity services
SP	Service Provider
TLS	Transport Layer Security
UI	User Interface
URI	Uniform Resource Identifier
URL	Uniform Resource Locator

1.4 Audience

The target audience for this document are mobile Operators' service / technical departments who are considering deploying Mobile Connect services using device-Initiated mode.

1.5 Relationship to Other Mobile Connect Documentation

This document describes and specifies the Mobile Connect Device-Initiated mode and API. It includes details of the OIDC Authorization Request and Response and the subsequent Token Request and Response obtaining tokens upon successful Authorization. It also includes examples and generic error codes. This specification defines the OIDC Authorization process and Token Retrieval that underpins Mobile Connect and forms part of the Core framework. All Mobile Connect Services published as user agent based services and user interacts online, make use of Device-Initiated mode.

The Mobile Connect Technical Overview document [20] provides a high-level description of Mobile Connect and how it works. It also includes a master list of abbreviations and terminology used within the Mobile Connect Documentation set and a map of that documentation set. It serves as a starting point for understanding how Mobile Connect works and also references the relevant documents for the reader to obtain further detail.

The Mobile Connect Architecture and Core Technical Requirements document [21] describes the Mobile Connect Architecture in more detail and also includes the core technical requirements and specification of elements for Mobile Connect that are generic to all Mobile Connect services and modes of operation.

The Mobile Connect Resource Server Specification [23] provides details on how to handle a Resource request and the associated response for Mobile Connect Identity and Network Attribute services including error codes where this approach is used by a Mobile Connect service.

Each individual Mobile Connect service has its own definition document which includes service specific parameters, such as scope value and any service specific error codes. It also includes technical requirements that relate to that specific Mobile Connect service.

1.6 Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [6].

1.7 Terminology & Definitions

Mobile Connect technical specifications and related documentation make use of terms that are defined by the OpenID Connect Core Specification [2] and supporting specifications and extended in the OIDF Client Initiated Backchannel Authentication Flow [5].

The Mobile Connect Technical Overview document [20] 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 (built on top of OAuth2.0) and OAuth2.0 terminology; the initial Mobile Connect service Request (OIDC Authentication Request) which underpins Mobile Connect Authentication, Authorisation and User Consent associated with Identity Services and Network Attribute Services, is referred to as an OIDC Authorization Request following the OAuth2.0 terminology (spelled with a ‘z’) throughout this document.

1.8 References

Ref	Doc Number	Title
[1]	Open ID Connect	http://openid.net/connect/
[2]	OpenID Connect Core Specification	"An interoperable authentication protocol based on the OAuth 2.0 family of specifications" available at https://openid.net/specs/openid-connect-core-1_0.html
[3]	OIDC Basic Client Profile	OpenID Connect Basic Client Profile 1.0 http://openid.net/specs/openid-connect-basic-1_0-28.html
[4]	OIDC Basic Client Implementer's Guide	OpenID Connect Basic Client Implementer's Guide 1.0 https://openid.net/specs/openid-connect-basic-1_0.html
[5]	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
[6]	RFC 2119	"Keywords for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997. Available at https://tools.ietf.org/html/rfc2119
[7]	RFC 6749	"The OAuth 2.0 Authorization Framework", D. Hard5, Ed. October 2012 available at http://www.ietf.org/rfc/rfc6749.txt
[8]	RFC 6750	M. Jones and D. Hardt, "The OAuth 2.0 Authorization Framework: Bearer Token Usage," RFC 6750, October 2012 https://tools.ietf.org/html/rfc6750
[9]	RFC 3339	Klyne, G., Ed. and C. Newman, "Date and Time on the Internet: Timestamps," RFC 3339, July 2002 https://www.ietf.org/rfc/rfc3339.txt
[10]	RFC 3986	"Uniform Resource Identifier (URI): Generic Syntax" http://www.ietf.org/rfc/rfc3986.txt
[11]	RFC 4627	Crockford, D., "The application/JSON Media Type for JavaScript Object Notation (JSON)," RFC 4627, July 2006 https://www.ietf.org/rfc/rfc4627.txt
[12]	RFC 5246	Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2," RFC 5246, August 2008 https://tools.ietf.org/html/rfc5322
[13]	RFC 5646	Phillips, A., and M. Davis, "Tags for Identifying Languages" BCP 47, RFC 5646, September 2009 https://tools.ietf.org/html/rfc5646
[14]	RFC 7519	M. Jones, J Bradley, N. Sakimura "JSON Web Token (JWT)", RFC 7519, MAY 2015 https://tools.ietf.org/html/rfc7519
[15]	RFC 7518	JSON Web Algorithms (JWA) https://tools.ietf.org/html/rfc7518
[16]	RFC 7517	JSON Web Key (JWK) https://tools.ietf.org/html/rfc7517

Ref	Doc Number	Title
[17]	RFC 7515	JSON Web Signature (JWS) https://tools.ietf.org/html/rfc7515
[18]	RFC 7516	Jones, M. and J. Hildebrand, "JSON Web Encryption (JWE)", RFC 7516, DOI 10.17487/RFC7516, May 2015, http://www.rfc-editor.org/info/rfc7516.
[19]	ISO 29115	International Organization for Standardization, "ISO/IEC 29115:2013 -- Information technology - Security techniques - Entity authentication assurance framework," ISO/IEC 29115, March 2013 https://www.iso.org/standard/45138.html
[20]	IDY.06)	Mobile Connect Technical Overview
[21]	IDY.04	Mobile Connect Technical Architecture and Core Requirements
[22]	IDY.02	Mobile Connect Server-Initiated OIDC Profile
[23]	IDY.03	Mobile Connect Resource Server Technical Requirements
[24]	IDY.33	API Exchange Functional Description
[25]	Dev Portal	Mobile Connect Developer Portal: https://developer.mobileconnect.io

2 OpenID Connect

OpenID Connect (OIDC) is a simple Identity layer that sits on top of the OAuth 2.0 protocol. It enables Clients to verify the identity of a User based on the authentication performed by an Authorization Server as well as to obtain basic profile information about the User in an inter-operable and REST-like manner. Figure 2 outlines the Open ID Connect Protocol Suite reproduced from OpenID.net [1]

OpenID Connect provides an additional token (an ID Token) along with the OAuth 2.0 Access Token. The ID Token is represented as a JSON[2] Web Token (JWT) [14] and contains a claim set related to the Authentication Context of the subject. The JWT can be a plain text JWT or cryptographically protected JWT – represented as a signed JWT using JSON Web Signatures (JWS) [17] or as an encrypted JWT using JSON Web Encryption (JWE)[3] [18].

OpenID Connect does not specify how users should actually be authenticated - Mobile Connect is a specific implementation of Open ID Connect (OIDC) that uses the User's MSISDN (and an associated Pseudonymous Customer Reference) as an identifier and their mobile device as the Authentication Device. Additionally, it extends the range of information about the User that can be obtained by a Client, subject to the User's consent.

[2] JSON – JavaScript Object Notation [11].

[3] Within Mobile Connect the ID Token is a signed JWT using JWS

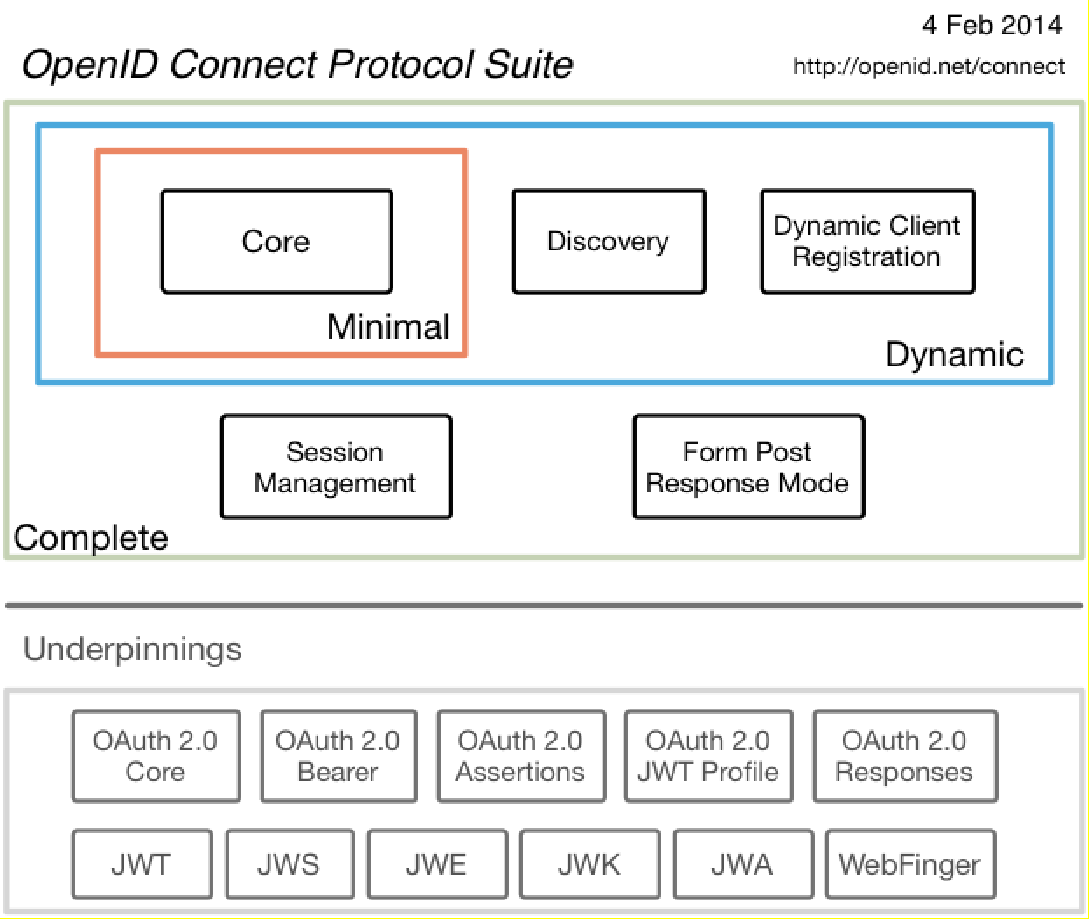

Figure 2: OpenID protocol suite

A general introduction to OIDC is provided in [1]. The Core Specification [2] defines the core OIDC functionality and that underpins this document. The Open ID Connect Basic Client Implementer's Guide [4] and the OpenID Connect Basic Client Profile 1.0 [3] contain a subset of the Core Specification that is designed to be easy to read and implement for basic web-based Service Providers using the OIDC Authorization Code Flow[4].

2.1 Mobile Connect Device-Initiated Mode

Device-Initiated mode is the default mode of operation where a User interacts with an SP Client application via a browser or native application on their Consumption Device and the OIDC Authorization call uses the User Agent in the device. Figure 3 illustrates the Device-Initiated mode flow. This specification details the parameters involved in the OIDC Authorization Request and Response and the Token Response and Response for Device-Initiated mode.

[4] OIDC[2] defines two more flows: Implicit Flow and Hybrid Flow. These flows are NOT used in the Mobile Connect.

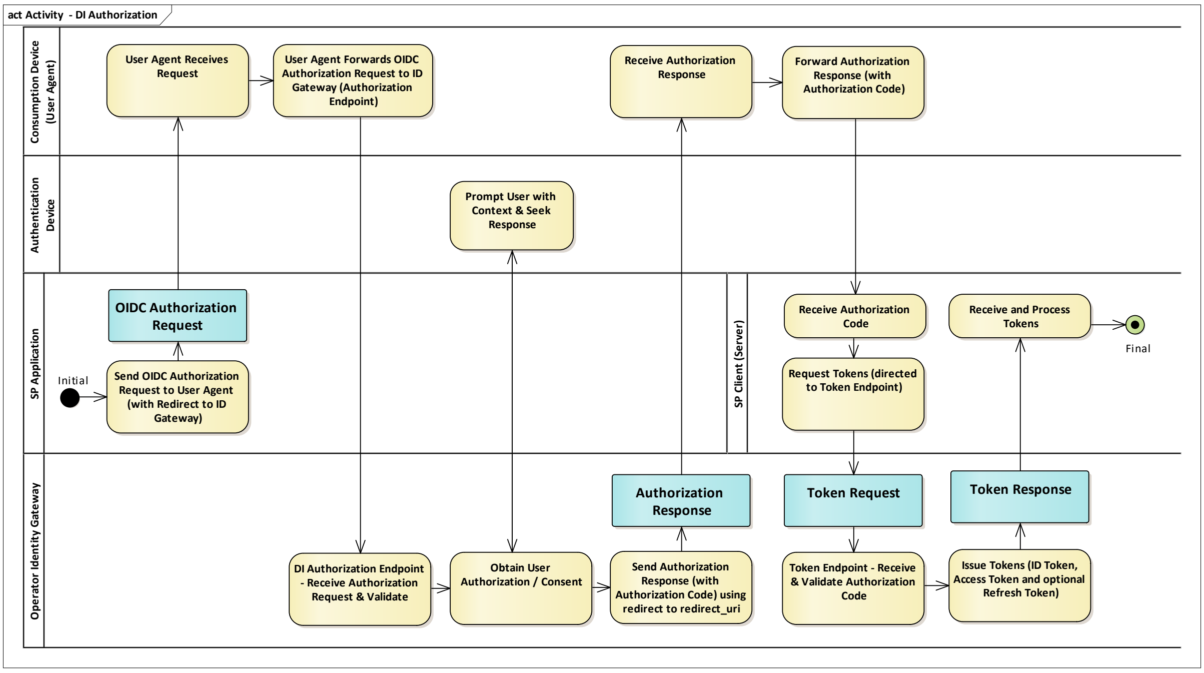
act Activity - DI Authorization
Consumption Device (User Agent)
Authentication Device
SP Application
SP Client (Server)
Operator Identity Gateway
User Agent Receives Request
User Agent Forwards OIDC Authorization Request to ID Gateway (Authorization Endpoint)
Receive Authorization Response
Forward Authorization Response (with Authorization Code)
Prompt User with Context & Seek Response
OIDC Authorization Request
Initial
Send OIDC Authorization Request to User Agent (with Redirect to ID Gateway)
Receive Authorization Code
Request Tokens (directed to Token Endpoint)
Receive and Process Tokens
Final
Authorization Response
Token Request
Token Response
DI Authorization Endpoint - Receive Authorization Request & Validate
Obtain User Authorization / Consent
Send Authorization Response (with Authorization Code) using redirect to redirect_uri
Token Endpoint - Receive & Validate Authorization Code
Issue Tokens (ID Token, Access Token and optional Refresh Token)

Figure 3: Mobile Connect Device-Initiated Mode Flow

- The User is using the service provided by the SP, and the use case requires the User to be authenticated[5]. This assumes that the SP is already registered with Mobile Connect.
- A Mobile Connect service request is initiated by the SP Client through OIDC Authorization request and this is forwarded by the User Agent (i.e. web browser or application) on the consumption device to the ID GW (Authorization Server)[6] passing the LoA (Level of Assurance) needed.
  - This enables the ID GW to interact directly with the User on the consumption device without the SP Client having any visibility.
- Mobile Connect Device-Initiated Mode utilises the Authorization Code Flow within the OIDC Core Specification [2] which determines how the request is submitted and also how the relevant tokens are returned. This is specified by setting the `response_type` value to "`code`" in the OIDC Authorization Request.
  - The Authorization Code Flow uses a 2-step process to obtain the Access Token, ID Token and optionally a Refresh Token.
- The ID GW selects the appropriate authenticator for the LoA and authenticates the User
- The OIDC Authorization Response returns an Authorization Code via re-direction of the User agent back to the SP Client

[5] The flow is similar for Authorisation and for seeking User Consent.

[6] The SP must have the credentials and service endpoints for the serving Operator / ID GW. If the SP does not have this information then the SP can optionally submit a Discovery Request to the API Exchange. Further details can be found in [25]

- The SP Client can then exchange the Authorization Code for an ID Token and Access Token by submitting an OIDC Token Request to the ID GW (Authorization Server) Token Endpoint.

Note: That this is a server to server call.

  - Within the OIDC Token Request, the SP specifies the <grant_type which for the Authorization Code Flow must be "authorization_code". The grant type value determines the flow of the returning Access Token and the ID Token to the Service Provider

- The ID GW validates the Authorization code and returns the Access Token along with the ID Token JWT (containing the authentication context), and optionally a Refresh Token.

  - The SP should validate the ID Token, validate the signature and decode it
  - The SP can then extract the PCR (Pseudonymous Customer Reference), iss and the authentication context (i.e. when and how the authentication was processed)
  - The PCR can then be used as an identifier for subsequent Mobile Connect service requests to the issued IDGW
  - Where requested (via the OIDC Authorization Request), the SP can then call the relevant resource endpoint (PremiumInfo or Mobile Connect Service-Specific Endpoint) by submitting the received Access Token to retrieve the requested attributes / claims (not shown in Figure 3).

The Authorization Code Flow provides the benefit of not exposing any tokens to the User Agent and possibly other malicious applications with access to the User Agent. The Authorization Server can also authenticate the Client before exchanging the Authorization Code for an Access Token. The Authorization Code flow is suitable for Clients that can securely maintain a Client Secret between themselves and the Authorization Server.

3 Service Provider Client Registration – Required Information.

The SP MUST first register the appropriate credentials for client applications with the Operator ID GW through the Mobile Connect Developer Portal [25] and API Exchange [24]. If an Operator is acting as an onboarding agent for SPs, the application developers register their Clients with the Operator, who then populates the API Exchange accordingly.

Table 1 defines the information that MUST be specified during SP registration to support Device-Initiated mode. Further details on SP registration can be found in [25].

Registration Parameter Name	Usage Category	Description
sector_identifier_uri		The value of the sector_identifier_uri[7] MUST be a URL using the https scheme that references a file with a single JSON array of redirect_uri values. It provides a way for a group of websites under common administrative control to have consistent PCR values independent of the individual domain names. It also provides a way for Clients to change redirect_uri domains without having to re-register all of their Users. The values registered in redirect_uris MUST be included in the elements of the array or the registration MUST fail. The value presented in the redirect_uri field in an OIDC Authorization Request MUST be included in the elements of the array, or the request MUST fail. This MUST be validated during SP on-boarding. If it is not registered then the registration process MUST throw an error. Mobile Connect Providers MUST utilize the sector_identifier_uri.
redirect_uris	REQUIRED	Array of Redirection URI values used by the SP. One of these registered Redirection URI values MUST exactly match the redirect_uri parameter value used in each Authorization Request, with the matching performed as described in Section 6.2.1 of [RFC 3986] (Simple String Comparison).
client_names	REQUIRED	Array of client names used by the SP. One of these registered client name values MUST exactly match the client_name parameter value used in each Authorization Request (if provided), with the matching performed as described Section 6.2.1 of [RFC 3986] (Simple String Comparison).

Table 1: SP Metadata Required for SP Client Registration (Device-Initiated Mode)

4 OIDC Authorization Request

Requirements:

[7] The previous version of MC profile was considering host part of the redirect_uri should be same for grouping SP applications to have consistent PCR value. This is deprecated since it has potential issues if SP hosts redirect URIs using multiple domain names, thus not allowing to group them.

- The communication with the ID GW for the OIDC Authorization Request MUST use HTTPS
- The request MUST use HTTP GET or POST as specified in [2]
- Query String Serialization is typically used in HTTP GET requests and Form Serialization is typically used in HTTP POST requests.

4.1 OIDC Authorization Request Parameters

Table 2 details the OIDC Authorization Request parameters that are applicable to the Mobile Connect Device-Initiated mode.

Parameter	Usage Category	Description
`response_type`	REQUIRED	• The OAuth 2.0 Response Type value that determines the authorization processing flow to use, including the parameters, returned from the endpoints used. • For device-initiated requests, the value MUST be "code", to indicate that the grant type flow to use is the Authorization Code Flow. This value indicates to the token endpoint to return both an access token and an ID Token in exchange to "code".
`client_id`	REQUIRED	OAuth 2.0 Client Identifier MUST be globally unique and is needed for an OIDC Authorization Request and be valid at the Authorization Server.
`scope`	REQUIRED	A space delimited, case-sensitive list of ASCII strings for OAuth 2.0 scope values. The Mobile Connect OIDC Authentication Request MUST contain the scope value “openid” followed by other values depending on the specific Mobile Connect products/services being requested. For Mobile Connect "`openid mc_authn`" is the default scope value. See Section 4.2 for further information.
`version`	REQUIRED	It is a plain string and value used to identify the profile version. This is always used in conjunction with the scope parameter values. Note: For backward compatibility, if the `version` parameter does not exist and scope does not contain Mobile Connect specific values, then the request MUST be considered as a first-generation authentication request only. See [21] for allowed values.
`redirect_uri`	REQUIRED	The Redirection URI where the ID GW Authorization Server sends the response. The URI MUST match one of the pre-registered redirect_uri values during client registration / provisioning. The format of the redirect_uri values are as defined in Section 3.1.2 of RFC6749 [7]. Matching is performed as described in Section 6.2.1 of RFC3986 (Simple String Comparison) [10]. Both HTTP and HTTPS schemes are allowed; The RECOMMENDED scheme is HTTPS. The redirect_uri MAY use HTTP provided that the Client Type is confidential, as defined in Section 2.1 of OAuth 2.0 [7].

Parameter	Usage Category	Description
`state`	REQUIRED	Value used by the client to maintain state between request and call-back. A security mechanism as well, if a cryptographic binding is associated with the browser cookie, to prevent Cross-Site Request Forgery.
`nonce`	REQUIRED	A string value used to associate a client session with the ID Token. It is passed unmodified from Authorization Request to ID Token. The value SHOULD be unique per session to mitigate replay attacks.
`display`	OPTIONAL	ASCII String value to specify the User interface display for the Authentication and Consent flow. The values can be: **page:** Default value, if the display parameter is not available. The UI SHOULD be consistent with a full page view of the User-Agent. **popup:** The popup window SHOULD be 450px X 500px [wide X tall]. **touch:** The Authorization Server SHOULD display the UI consistent with a "touch" based interface. **wap:** The UI SHOULD be consistent with a "feature-phone" device display.
`prompt`	OPTIONAL	Space delimited, case-sensitive ASCII string values to specify to the Authorization Server whether to prompt or not for re-authentication and consent. The value can only be: • **none :** MUST NOT display any UI for re-authentication or consent to the user. If the user is not already authenticated or authentication or consent is needed to process the Authorization Request, a login_required error is returned. This can be used as a mechanism to check existing authentication or consent. • **login**: Must prompt the user for re-authentication or consent. In case it is not possible, an error MUST be returned. • **no_seam** : To indicate that ID GW MUST prompt the User if the authentication session is not valid and to cater for the 3G/4G hotspots, tethered phone etc., scenarios. If the authentication session is not valid and is not possible to prompt the User ID GW MUST throw an error. If `scope` is "openid mc_authz" the Authorization server MUST always prompt the User for Mobile Connect Authorisation using context, binding_message and client_name parameters which must override any setting in the `prompt` parameter.
`max_age`	OPTIONAL	Specifies the maximum elapsed time in seconds since the last authentication of the User. If the elapsed time is greater than this value, a re-authentication MUST be attempted. When this parameter exists in the request, the ID Token MUST contain the `auth_time` claim value.

Parameter	Usage Category	Description
`ui_locales`	OPTIONAL	Space separated list of User preferred languages and scripts for the UI as per RFC5646 [13]. This parameter is for guidance only and in the case of unsupported locales, ID GW Authorization Server SHOULD NOT return an error. If scope value is "`openid mc_authz`" and a value is present the Mobile Connect Provider MUST consider this value for processing the context parameter. For instance, the value "`fr-CA fr en`" represents a preference for French as spoken in Canada, and then French (without a region designation), followed by English (without a region designation). An error SHOULD NOT result if some or all of the requested locales are not supported by the OpenID Provider. For more information see Reference [2].
`claims_locales`	OPTIONAL	Space separated list of User preferred languages and scripts to return the Claims as per RFC5646 [13] . This parameter is for guidance only and in the case of unsupported locales, ID GW Authorization Server SHOULD NOT return an error.
`id_token_hint`	OPTIONAL	Used in conjunction with `prompt=none` to pass the previously issued ID Token as a hint for the current or past authentication session. If the User is logged in and the ID Token is still valid, then the server returns a positive response, otherwise, SHOULD return a login_error response. For the ID Token, the Authorization Server need not be listed as an audience of the ID Token, when included in the id_token_hint. However, the server SHOULD respond successfully when possible, even if it is not present. If scope is "`openid mc_authz`" this value MUST be ignored. ID GW Authorization Server MUST always display an authorization prompt to the User for approval.
`login_hint`	RECOMMENDED [REQUIRED if login_hint_token does not exist]	An indication to the ID GW Authorization Server on what ID to use for login. The login_hint can contain the MSISDN, encrypted MSISDN or PCR. The format MUST be as MSISDN:<Value>,ENCR_MSISDN:<Value> and PCR:<value> The usage to transport the encrypted MSISDN will be deprecated in future releases, instead login_hint_token will be used.
`login_hint_token`	OPTIONAL [REQUIRED if login_hint does not exist]	The "login_hint_token" is used to transport a User identifier (MSISDN for Mobile Connect) from the Discovery Servicer to the Operator ID GW without revealing the identifier to the SP Client [5]. The "login_hint_token" is an encrypted JSON Web Token (JWT) [14]. This token is typically created if a User has entered an MSISDN during the discovery process, and, if present, SHALL be used by the Client as login hint with the particular Operator.

Parameter	Usage Category	Description
acr_values	REQUIRED	Authentication Context Class Reference. Space separated string that specifies the Authentication Context Reference used during authentication processing. The acr_values are an indication of what level of assurance (LoA) is required and therefore which Authenticator should be selected by the ID GW. The SP Client can request Levels of Assurance in order of preference for a particular use case. Depending upon which Levels of Assurance are supported in the ID GW, the ID GW MUST consider only the first supported value in the list and ignore remaining values whilst processing the request. Possible values for acr_values are defined in the Mobile Connect Technical Architecture and Core Requirements document [21]. ID GW Authorization Server MUST return the achieved level of assurance in the acr claim in the ID Token.
binding_message	OPTIONAL [REQUIRED if scope = “openid mc_authz”]	Client provided plain text, "reference or ID" to interlock Consumption Device and Authentication Device for a better User experience and User assurance. The message will be displayed on Consumption Device and Authentication Device. Empty values are allowed. (zero length) binding_message MUST be provided by the SP Client if scope = "openid mc_authz”.
client_name	OPTIONAL [REQUIRED if scope = “openid mc_authz”]	A short name to identify the SP Client Application. It MUST be displayed on the Authentication Device. REQUIRED if scope = “openid mc_authz”. Service specific requirements can mandate the client_name. When multiple client names are registered, it is REQUIRED to submit with a valid registered value.
context	OPTIONAL [REQUIRED if scope = “openid mc_authz”]	A transaction / action based message displayed on the Authentication Device context MUST be provided by the SP Client if scope = "openid mc_authz”.
claims	OPTIONAL	Within Mobile Connect this parameter is used to specify specific claims and associated values to be returned from the relevant Resource Endpoint in the context of a requested MC service (via the scope parameter). The value is a JSON object listing the requested claims. Claim values can be in plain text or in a hashed form. This is only used for the MC KYC Match services. New services will make use of the mc_claims parameter included within a Resource Request.
correlation_id	OPTIONAL	This parameter is generated by the Service Provider only. It is used to correlate the transaction across Mobile Connect components (Discovery, MC Profile requests, ID GW internal components etc.,). It MUST be locally unique.

Table 2: OIDC Authorization Request Parameters for Mobile Connect Device-Initiated Mode

4.2 The scope Parameter

OIDC scope values determine the specific Mobile Connect services being requested by the Service Provider, subject to the SP being registered to use those services.

The Mobile Connect OIDC Authorization Request MUST contain the scope parameter which is a space delimited, case-sensitive list of ASCII strings (scope values). The scope values MUST include “openid”, to indicate that the request is an OpenID Connect request, followed by other values depending on the specific Mobile Connect services being requested. Multiple scope values can be requested simultaneously, subject to the SP being registered to use those “scopes”.

Scope values are defined for each Mobile Connect service in the relevant service “Definition and Technical Requirements” document.

5 OIDC Authorization Response

On receipt of an OIDC Authorization Request, the Operator ID GW / Authorization Server authenticates the User, obtains User consent as required (based on the requested scope), and returns an Authorization code (specified by response_type=“code” in OIDC Authorization Request) to the Service Provider Client.

The ID GW Authorization Server returns the authorization code to the redirect_uri and returns the response as query parameters using form serialization: “application/x-www-form-urlencoded”. Table 3 outlines the parameters contained within the OIDC Authorization Response.

Parameter	Required Category in Profile	Description
code	REQUIRED	Authorization code as per OAuth 2.0 section 4.1.2 [7].
state	REQUIRED	If the state parameter is present in the Client Authorization Request, it MUST equal to the value of state parameter in the Authorization Request.
correlation_id	OPTIONAL	The correlation_id submitted through the OIDC authorization request. IDGW SHOULD return this parameter with the same value; if provided in the authorization request.

Table 3: Authorization Response Parameters

In the case where the User authentication fails, or the User does not provide consent, the ID GW Authorization Server MUST return an error in the Authorization Response (See Annex A).

6 Token Request from Service Provider Client (From Server)

On receipt of a successful OIDC Authorization Response, the SP can then submit a Token Request to the ID GW Token Endpoint. Communication with the Token Endpoint MUST use TLS with form serialization: “application/x-www-form-urlencoded”.

Parameter	Required Category in Profile	Description
grant_type	REQUIRED	The value MUST be "authorization_code".
code	REQUIRED	The authorization code received from the authorization server, as a result of the successful authorization request.
redirect_uri	REQUIRED	The redirect_uri value MUST match the one sent in the authorization request.
client_id	REQUIRED	The same client identifier that is used in the authorization request for which authorization code is issued to).
client_secret	REQUIRED	The client_secret issued during registration, and MUST be utilized in HTTP Basic Authentication using the OAuth2.0 Client Password Mechanism (RFC 6749 [7].
correlation_id	OPTIONAL	The correlation_id that was submitted through the OIDC authorization request (if present).

Table 4: OIDC Token Request Parameters

6.1 Service Provider Authentication Mechanisms

Mobile Connect Providers may support one or both of the following mechanisms to cater for Service Provider applications with different capabilities. For more details refer to [7].

6.1.1 Service Provider Authentication using HTTP Basic Authentication

In Mobile Connect, all Service Providers are issued with a client_id and client_secret during registration. Service Providers in possession of a client_secret SHOULD use the HTTP Basic authentication scheme as defined in [RFC 2617] to authenticate. The client identifier is encoded using the "application/x-www-form-urlencoded" encoding algorithm and the client_secret is encoded using the same algorithm and used as the password. The authorization server MUST support the HTTP Basic authentication scheme for authenticating clients.

This is the RECOMMENDED mechanism for client authentication.

6.1.2 Service Provider Authentication with Client Credentials in The Request Body[8]

Mobile Connect Providers MAY support including the client credentials (i.e. client_id and client_secret) in the request-body. Including the client credentials in the request_body using the two parameters is NOT RECOMMENDED and SHOULD be limited to clients unable to directly utilize the HTTP Basic authentication scheme. The parameters can only be transmitted in the request_body and MUST NOT be included in the request URI. This is an OPTIONAL feature.

[8] In real-life scenario, few Operators have service providers who uses request body to authenticate to the authorization server. This is not recommended feature and Mobile Connect does not use this method for compliance, whereas HTTP Basic authentication mechanism must be supported by the IDGW.

7 Token Response to Service Provider Client

On receipt of a valid Authorization Code at the ID GW Token Endpoint, the ID GW responds with ID Token, Access Token and optionally a Refresh Token back to the SP. The Token response MUST comply with OAuth 2.0, and the encoding scheme SHOULD be in UTF-8. Table 5 details the parameters within the Token Response.

Parameter	Required Category in Profile	Description
access_token	REQUIRED	OAuth 2.0 access_token used to get the PremiumInfo service specific resource object from the PremiumInfo/service specific resource end point and can be reused for accessing other protected resources, if required. This parameter MUST be utilized using either the Authorization header field or a form-encoded POST body parameter.
token_type	REQUIRED	MUST be "bearer" as defined in RFC 6749 section 7.1 [7] and RFC6750 [8], unless another token_type value as agreed between the SP Client and ID GW Authorization Server. For the Mobile Connect Profile, token_type=bearer is the RECOMMENDED value.
id_token	REQUIRED	This the additional token used in OIDC to provide the identity token claim, a security token that contains Claims about the Authentication of an User by an Authorization Server when using a Client, and potentially other requested Claims. The format of the ID Token is a JSON Web Token (JWT).
expires_in	REQUIRED	Expiration time of the Access Token in seconds since the response was generated.
refresh_token	OPTIONAL	OAuth 2.0 refresh token to get the new access tokens using the same authorization grant through a grant_type parameter.
correlation_id	OPTIONAL	The correlation_id submitted through the OIDC Authorization request. If present. IDGW should return the same value.

Table 5: OIDC Token Response Parameters

7.1 ID Token (JWT – Mobile Connect Service Identity & Proof Token)

The primary extension that OpenID Connect makes to OAuth 2.0 to enable Users to be Authenticated is the ID Token data structure. The ID Token is a security token that contains Claims about the Authentication of a User by an Authorization Server when using a Client, and potentially other requested Claims. It is returned along with the OAuth 2.0 Access Token

The ID Token is represented as a JSON Web Token (JWT) [14] and is created and returned by the ID GW. The JWT is signed by the ID GW using JSON Web Signatures (JWS) [17]. See also the OIDC Core Specification [2].

Table 6 describes the contents of the ID Token and whether the Parameter or Claim is Mandatory or Optional for the Mobile Connect Device-Initiated OIDC Profile.

Parameter	Required Category in Profile	Description
iss	REQUIRED	Issuer Identifier - a case-sensitive HTTPS based URL, with the host. It MAY contain the port and path element (OPTIONAL) but no query parameters.
sub	REQUIRED	Subject identifier - A globally unique identifier for the User to work in a federated environment. It is a case-sensitive ASCII string with a maximum length of 255. The sub value MUST not contain the MSISDN. Within Mobile Connect this is populated by a Pseudonymous Customer Reference (PCR) which is a system-facing unique identifier that links a User's MSISDN to a Service Provider (SP). It is generated by the ID GW. This is defined in Mobile Connect Technical Architecture and Core Requirements [21].
aud	REQUIRED	The intended audience for the ID Token. It is an array of case-sensitive strings. It MUST contain the client_id of the SP Client, and MAY contain identifiers of other OPTIONAL audiences. If there is one audience, the aud value MAY be a single case-sensitive string OR an array of case-sensitive strings with only one element. An implementation MUST support both scenarios.
exp	REQUIRED	The expiration time after which the ID Token MUST NOT be accepted for processing. The format is the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time specified. If scope = "openid mc_authz", the implementor MUST give the lowest possible time but no more than a few minutes. The processing of this parameter requires that the current date/time MUST be before the expiration date/time listed in the value. See RFC 3339 [9] for details regarding date/times in general and UTC in particular.
iat	REQUIRED	The time at which the ID Token JWT was issued. The format is the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time specified.
auth_time	REQUIRED	Time of User authentication or authorization. The format is the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time specified. See RFC 3339 [9] for details regarding date/times in general and UTC in particular.

Parameter	Required Category in Profile	Description
nonce	REQUIRED	Opaque string value to associate the SP Client session with the ID Token, to avoid the replay attacks. The nonce value MUST be same as the nonce used in the Authorization request.
at_hash	REQUIRED	A base64url encoded, the value of the hash of the access_token The OIDC Core Specification, Section 3.1.3.6 [2] describes the process for generating the Access Token hash value (at_hash).
acr	REQUIRED	A REQUIRED authentication context class reference. It is a case-sensitive string, representing the achieved authentication by the ID GW Authorization Server. The values MUST meet the requirements of ISO/IEC29115 [19].
amr	REQUIRED	Authentication Methods References. An array of case-sensitive strings to indicate the authentication method used. The possible amr values are defined in Mobile Connect Technical Architecture and Core Requirements [21].
azp	OPTIONAL [REQUIRED if the audience to the ID Token is different to the Authorised Party]	Authorised Party – the party to which the ID Token is issued. If present, it MUST contain the Client ID. This Claim MUST be present when the ID Token has a single audience value, and that audience is different than the authorized party. It MAY be included even when the authorized party is the same as the sole audience. The azp value is a case sensitive string containing a String Or URI.
displayed_data	OPTIONAL [REQUIRED for MC Authorisation services]	Displayed data on the Authentication Device. Value is derived by combining client_name, context and binding_message. Service specific requirements can mandate this parameter [i.e. when consent is captured]
dts	OPTIONAL [only applicable for future LoA4 Mobile Connect services]	Signature. The data signed includes displayed_data and dts_time. Optional extension that can be used in conjunction with LoA4 enabled Mobile Connect services that require a User-signed response to be returned to the SP.
upk	OPTIONAL [only applicable for future LoA4 Mobile Connect services]	User Public Key or User certificate / reference to the certificate. Optional extension that can be used in conjunction with LoA4 enabled Mobile Connect services that require a User-signed response to be returned to the SP.

Parameter	Required Category in Profile	Description
`dts_time`	OPTIONAL [only applicable for future LoA4 Mobile Connect services]	The time of signing the text. The format is the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time specified. Optional extension that can be used in conjunction with LoA4 enabled Mobile Connect services that require a User-signed response to be returned to the SP.
`hashed_login_hint`	REQUIRED	Hashed login_hint or login_hint_token value to mitigate security threats. If the request is made using login_hint, then it MUST return the hashed login_hint value and if the request is made using login_hint_token then it MUST return the hashed login_hint_token value. See Mobile Connect Technical Architecture and Core Requirements [21] for more information on hashing algorithms. The SHA256 algorithm SHOULD only be used for backwards compatibility only.

Table 6: ID Token Claims (Device-Initiated Mode)

7.2 Access Token

Access tokens are credentials used to access protected resources (User attributes). An Access Token is a string representing an authorization issued to the SP Client. The string is opaque to the SP Client. Tokens represent specific scopes and durations of access, granted by the User (through appropriate User consent), and enforced by the ID GW Resource Server and ID GW Authorization Server. Further information on the Access Token can be found in RFC6749 [7] and RFC6750 [8].

7.3 Refresh Token (Optional)

Refresh Tokens are credentials used to obtain Access Tokens. Refresh Tokens are issued to the client by the authorization server and are used to obtain a new Access Token when the current Access Token becomes invalid or expires.

A Refresh Token is a string representing the authorization granted to the client by the resource owner. The string is usually opaque to the client. The token denotes an identifier used to retrieve the authorization information. Unlike Access Tokens, Refresh Tokens are intended for use only with the Authorization Server and are never sent to Resource Servers.

Further information can be found in RFC6749 [7]

8 Security Considerations

The security considerations listed in the OIDF specifications SHALL be considered in the Mobile Connect implementation:

- Section 16, OIDC Core Specification [2].
- Section 7, OIDC MODRNA Client-Initiated Backchannel Authentication Flow [5].

Annex A Generic Error Codes and Descriptions for Device-Initiated Mode

A.1 OIDC Authorization Response – Error Codes and Descriptions

Mobile Connect follows the OIDC error handling mechanism to send any errors back to the Service Provider (See OIDC Core Specification [2]). These errors can be returned in a query string using an HTTP redirect 302 status code back to the Service Provider if there is a valid value for the `redirect_uri` parameter.

If the OIDC Authorization Request contained a `state` parameter, then the error response MUST contain that `state` parameter value. The value is set to the value received from the Service Provider. If the `correlation_id` parameter is provided in the OIDC Authorization Request, then it must be included in the OIDC Authorization Response, and the value must be set to the value received from the Service Provider.

The following is an example of an error response.

```
HTTP/1.1 302 Found
Location: https://sp.example.org/redirct_here?
 error=invalid_request
 &error_description=Invalid%20response_type%20value
 &state=af0ifjsldkj
 &correlation_id=example correlationid
```

Table 7 lists the generic[9] error codes and error descriptions from the ID GW Authorization Server. Error responses are passed via the `redirect_uri`. If the `redirect_uri` is missing or the `redirect_uri` value is invalid, then the Operator ID GW always treats this as a high priority error and MUST return the appropriate error response (400 Bad Request) as described in Table 7.

[9] i.e. not specific to a particular Mobile Connect service

Error Scenario	HTTP mode	Error code	Error Description [RECOMMENDED text]
MSISDN/ENCR MSISDN/ PCR provided does not belong to the Operator	Redirect 302	access_denied	"Unknown User" [OR] "User is not recognized"
MSISDN/ENCR_MSDISDN belongs to the Operator, but MC services are not enabled Note: This applies if ID GW policy does not allow "on-the-fly" User registration	Redirect 302	access_denied	"User is not registered" [OR] "Unknown User
System connection problems [internal to ID GW] (or) Authenticator unreachable (or) Expiration in server (or) Any Unexpected error [internal to ID GW]	Redirect 302	server_error (or) temporarily_unavailable	Internal Server Error
Multiple requests for the same MSISDN sent at the same time	Redirect 302	access_denied	The User is busy with another transaction
redirect_uri value is invalid or not a registered URI or does not exist in the request.	Bad Request 400	invalid_request	redirect_uri is invalid
response_type parameter is missing	Redirect 302	invalid_request	REQUIRED parameter response_type is missing or value is invalid
response_type parameter exists but the value is invalid (or) response_type parameter exists and the value is valid as defined in the specs, but value is not supported by the IDGW.	Redirect 302	invalid_requset (or) unsupported_response_type	Invalid response_type value
client_id parameter is missing	Bad Request 400	invalid_request (or) access_denied	REQUIRED parameter client_id is missing
client_id parameter value is invalid	Bad Request 400	invalid_client (or) access_denied	The client is not authorized to request an authorization code
client_id is valid, but not allowed to make MC service requests and redirect_uri is valid.	Redirect 302	unauthorized_client(or) access_denied	The client is not allowed to make MC service request.
client_id is valid, but not allowed to make MC service requests and redirect uri is invalid.	Bad Request 400	unauthorized_client(or) access_denied	The client is not allowed to make MC service request.

Error Scenario	HTTP mode	Error code	Error Description [RECOMMENDED text]
scope parameter is missing	Redirect 302	invalid_request	REQUIRED parameter scope is missing (or) invalid scope value
scope parameter exists value is invalid (or) "openid" is missing	Redirect 302	invalid_scope	Malformed scope value
scope parameter exists, IDGW has published the scope value in the provider metadata, but IDGW does not support the requested scope temporarily	Redirect 302	temporarily_unavailable	Requested service is not available temporarily
version parameter is missing (or) value is invalid	Redirect 302	invalid_request	REQUIRED parameter version is missing / invalid
state parameter exists, but the value is invalid	Redirect 302	invalid_request	RECOMMENDED parameter state is invalid
nonce parameter is missing or the value is empty	Redirect 302	invalid_request	REQUIRED parameter nonce is missing (or) invalid
login_hint and login_hint_token parameters are missing Note: In Device-Initiated mode, the ID GW SHOULD prompt the User to input their MSISDN instead of throwing an error, where: -- MC service is for stand-alone authentication only, and, -- If ID GW policy allows the capture of the MSISDN directly from the use	Redirect 302	invalid_request	REQUIRED parameters login_hint_token (or) login_hint does not exist
login_hint and login_hint_token both exist	Redirect 302	invalid_request	Malformed request, duplicate parameter entries
login_hint (or) login_hint_token value is invalid	Redirect 302	invalid_request	Invalid value for login_hint or login_hint_token
acr_values parameter is missing (or) contains an invalid value, other than supported values	Redirect 302	invalid_request	REQUIRED parameter acr_values are missing or invalid values

Error Scenario	HTTP mode	Error code	Error Description [RECOMMENDED text]
`display` parameter exists and it has an invalid value (or) the ID GW does not support the requested value	Redirect 302	`invalid_request`	Invalid display value. / not supported
The same parameter exists multiple times	Redirect 302	`invalid_request`	Multiple parameter names in the authorization request. Malformed request
`prompt` value exists, and it has an invalid value	Redirect 302	`invalid_request`	`prompt` value is invalid
`claims` parameter exists, but it does not contain any values (or) contains invalid values.	Redirect 302	`invalid_request`	`claims` value is invalid
GET request is used, and the request parameter is NOT serialized using URI string serialization, ID GW able to validate the `redirect_uri`.	Redirect 302	`invalid_request`	GET request invalid serialization
POST request is used, the request parameters are NOT serialized using form serialization, ID GW can validate the `redirect_uri`	Redirect 302	`invalid_request`	POST request Invalid serialization
`max_age` parameter exists, and it has an invalid value	Redirect 302	`invalid_request`	Invalid `max_age` value
Multiple problems in authorization request [redirect URI is valid]	Redirect 302	`invalid_request`	Malformed request multiple problems exist
`correlation_id` exists, but it has an empty value	Redirect 302	`invalid_request`	Invalid correlation_id value
`client_name` exists but it has empty value (or) `client_name` parameter exists, but the provided value is not a registered `client_name` with Operator ID GW and invalid	Redirect 302	`invalid_request`	Invalid `client_name` value

Table 7 : Generic Errors from the Device-Initiated Authorize Endpoint

A.2 Token Response – Error Codes and Descriptions

The Token Request is always a server-initiated request. It must be a POST request. A SP makes a token request by presenting the parameters using form serialization to the Token Endpoint. In the event of an error in processing a Token Request, the Token Endpoint must return errors in the following format:

```
HTTP/1.1 400 Bad Request
  Content-Type: application/json
  Cache-Control: no-store
```

```
Pragma: no-cache

{
 "correlation_id": "<example correlation id value>",
 "error": "invalid_request",
 "error_description": "mandatory parameter is missing"
}
```

Where the correlation_id parameter is the same value as included within the original OIDC Authorization Request (if it was present). Table 7 above lists the generic error codes and error descriptions that may be returned by the ID GW Token Endpoint

Error Scenario	HTTP mode	Error code	Error Description [RECOMMENDED text]
grant_type parameter is missing	Bad Request 400	invalid_request	REQUIRED parameter grant_type is missing (or) invalid
grant_type parameter exists, but value is invalid or not supported	Bad Request 400	unsupported_grant_type	Invalid grant type
Authorization code parameter is missing (or) the value is invalid (or) the value has already been used or has expired (or) the Authorization code is valid, but has not been issued to the authenticated client (or) the Authorization code is valid, but it is related to an MC OIDC Request	Bad Request 400	invalid_grant (or) invalid_request	REQUIRED parameter code is missing (or) invalid (or) expired
redirect_uri parameter is missing (or) it has a value that DOES NOT match the one sent in the authorization request (or) it has an unregistered value (or) where Operator ID GW has multiple redirect URI values registered for a given client_id; the redirect_uri parameter exists and it has a value that matches one of the redirect URI registered with Operator ID GW, but the value DOES NOT match the one sent in the previous authorization request	Bad request 400	invalid_request	REQUIRED parameter redirect_uri is missing (or) is invalid
client_id parameter does not exist (or) it has a value that is not registered at Operator ID GW	Bad Request 400 (or) 401	access_denied (or) invalid_client	Invalid client credentials
client_secret parameter does not exist (or) it has invalid value	Bad Request 400 (or) 401	access_denied (or) invalid_client	Invalid client credentials

Error Scenario	HTTP mode	Error code	Error Description [RECOMMENDED text]
correlation_id does not exist but previous authorization request and response has correlation ID (or) correlation_id exists but it has empty value (or) correlation_id exists but it has value that DOES NOT match the one sent in the previous authorization request	Bad Request 400	invalid_request	Missing REQUIRED parameter correlation ID (or) invalid
Same parameter exists multiple times	Bad Request 400	invalid_request	Malformed request, the same parameter exists multiple times
Unexpected error	Internal server Problem 500	server_error	Internal error
System connection problem	Service Unavailable 503	server_error	Service is not available
SP sends token request through POST, but without form serialization	Bad Request 400	invalid_request	No form serialization exists
Multiple problems in token request	Bad Request 400	access_denied	Multiple problems were in the token request

Table 8: Generic Errors from the Device-Initiated Token Endpoint

Annex B Example Requests and Responses

B.1 OIDC Authorization Request

The request is sent using HTTPS / TLS to the ID GW Authorization Server using GET or POST.

B.1.1 Mobile Connect Authentication – default scope:

The following is an example of an OIDC Authorization Request where only the “openid” scope value is used – in this case it defaults to a Mobile Connect Authentication:

```
GET /authorize?
response_type=code&
client_id=s6BhdRkqt3
&redirect_uri=https%3A%2F%2Fclient.example.org
&scope=openid
&state=af0ifjsldkj
&nonce=n-0S6_WzA2Mj
&correlation_id=42da5b19-457a-4d30-a5c4-038c62dccbb0

HTTP/1.1

Host: operator1.example.com
```

B.1.2 Mobile Connect Authentication – scope “mc_authn”

The following is an example of an OIDC Authorization Request where the “mc_authn” scope value is used (along with the “openid” value which must be included) to explicitly request a Mobile Connect Authentication:

```
GET /authorize?
response_type=code&
client_id=s6BhdRkqt3
&redirect_uri=https%3A%2F%2Fclient.example.org
&scope=openid%20mc_authn
&state=af0ifjsldkj
&nonce=n-0S6_WzA2Mj
&version=mc_v2.3
&correlation_id=142ab373-0764-4c0a-ae25-ed1d00101f63

HTTP/1.1

Host: operator1.example.com
```

B.1.3 Mobile Connect Authorisation – scope “mc_authz”

The following example illustrates the OIDC Authorization Request for a Mobile Connect Authorisation using the “mc_authz” scope value:

```
GET /authorize?
response_type=code&
client_id=s6BhdRkqt3
&redirect_uri=https%3A%2F%2Fclient.example.org
&scope=openid%20mc_authz
&state=af0ifjsldkj
&nonce=n-0S6_WzA2Mj
&version=mc_v2.3


HTTP/1.1
Host: operator1.example.com
```

B.2 OIDC Authorization Response

The following is an example of a successful OIDC Authorization Response which returns the Authorization Code:

```
HTTP/1.1 302 Found
Location: https://client.example.org?code=SplxlOBeZQQYbYS6WxSbIA
&state=af0ifjsldkj
&correlation_id=42da5b19-457a-4d30-a5c4-038c62dccbb0
```

B.3 OIDC Token Request

The following example shows the Token Request to exchange the Authorization Code for ID Token, Access Token and optionally a Refresh Token:

```
POST /token HTTP/1.1
Host: operator1.example.com
Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW
Content-Type: application/x-www-form-urlencoded

grant_type=authorization_code&code=SplxlOBeZQQYbYS6WxSbIA
&redirect_uri=https%3A%2F%2Fclient%2Eexample%2Eorg
&correlation_id=42da5b19-457a-4d30-a5c4-038c62dccbb0
```

B.4 OIDC Token Response

The following illustrates the Token Response after a successful Token Request:

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache

{
“correlation_id”:”42da5b19-457a-4d30-a5c4-038c62dccbb0”,
"access_token":"SlAV32hkKG",
"token_type”: Bearer",
"expires_in":3600,
"refresh_token":"tGzv3JOkF0XG5Qx2TlKWIA",
"id_token":"eyJ0 ... NiJ9.eyJ1c ... I6IjIifX0.DeWt4Qu ... ZXso"
}
```

Annex C Document Management

C.1 Document History

Version	Date	Brief Description of Change	Approval Authority	Editor / Company
1.0	18/02/2015	PRD published	PDATA/ PSMC	Gautam Hazari, GSMA
2.0	12/02/2016	Document updated to be consistent with the latest OpenID Connect specs and to support the Mobile Connect Authentication, Mobile Connect Authorisation and PremiumInfo concepts.	PDATA/ PSMC	Venkatasivakumar Boyalakuntla (Siva), GSMA
2.1	13/03/17	Document updated with extra requests and parameters information.	PDATA/PSMC	Venkatasivakumar Boyalakuntla (Siva), GSMA
2.2	12/05/2017	Transfer of PRD from Personal Data		Nick Cheung, GSMA
2.3	11/08/2017	Document updated: addition of new parameters and removal of server-initiated references	TG	Venkatasivakumar Boyalakuntla, GSMA
V3.0	03/05/2019	Merged David & DQrt (Donna's) review comments. Major change, error handling is added, whole document is restructured. After TG approval this document is considered as v3.0 DI profile.	TG	Venkatasivakumar Boyalakuntla (Siva)/GSMA

C.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.