OpenID Connect Mobile Connect Profile
Version 1.0
17 February 2015

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

Security Classification: Confidential - Full, Rapporteur, and Associate Members

Copyright Notice

Disclaimer

Antitrust Notice

Table of Contents

1 Introduction

The GSMA Mobile Identity programme is focused on positioning operators as trusted providers of identity services to 3rd party service providers. The programme identifies a set of propositions (authentication, identity, attribute validation, attribute brokerage) that collectively are referred to as Mobile Connect and are based on the OpenID Foundation OpenID Connect standard [OpenID Connect]. This document defines a Profile of OpenID Connect that should be used by MNOs for implementation of any of the Mobile Connect propositions.

1.1 Abbreviations

Term	Description
SP	Service Provider (In the context of Mobile Connect) : The application/service that provides end user services to the consumers and needs the authentication and identity services. And example of a Service Provider is a Ticketing website/application that needs authentication of the user.
IDP	Identity Provider : The entity providing the authentication and identity services, e.g. the MNO
OIDC	OpenID Connect
MNO	Mobile Network Operator
JSON	Javascript Object Notation
REST	Representational State Transfer
ACR	Anonymous Customer Reference
JWT	JSON Web Token
LOA	Level Of Assurance
UTC	Coordinated Universal Time
URL	Universal Resource Locator
SHA	Secure Hash Algorithm

1.2 References

Ref	Doc Number	Title
[1]	OpenID Connect Core 1.0	The Core specification of OpenID Connect: http://openid.net/specs/openid-connect-core-1_0.html
[2]	RFC 2119	“Key words for use in RFCs to Indicate Requirement Levels”, S. Bradner, March 1997. Available at: http://www.ietf.org/rfc/rfc2119.txt
[3]	RFC 6749	OAuth 2.0 Authorisation Framework. OpenID Connect is based on OAuth 2.0. Available at: http://tools.ietf.org/html/rfc6749
[4]	ACR URI Extension	ACR (Anonymous Customer Reference) as an implementation for the “sub” claim in ID Token. Available at: https://tools.ietf.org/html/draft-uri-acr-extension-04

2 OpenID Connect

OpenID Connect (OIDC) is an identity layer on top of OAuth 2.0. The functionality it provides is:

- Identity verification/Authentication of end-user
- JSON/REST-like API for authentication and basic profile sharing

2.1 Mobile Connect Propositions and OpenID Connect Profile

The key Mobile Connect propositions in the context of OpenID Connect Profile are as follows:

- Identification
- Authentication and Authentication assurance/assertion
- User consent and authorisation management
- User profile attributes assertion

Here is a chart to map the Mobile Connect propositions to the OpenID Connect Profile elements:

Proposition	OIDC Mobile Connect Element	Additional Comments
Identification	OIDC Authorisation Request, ID Token	The user identification is done through the OIDC Authorisation Request to the IDP – which MAY be the mobile network operator or a delegated entity. The IDP then challenges the user to identify himself. If the MNO is the IDP a number of options can be used to help with the Identification, utilising the MNO assets, e.g. identifying the user using network authentication, etc. if seamless identification is an option. The ID Token returns claims related to the authentication and identity.

Authentication and Authentication assurance/assertion	OIDC Authorisation Request, ID Token	The authentication is performed through the challenge sent to the user asking the user to provide credentials.
		If the IDP is the MNO, there can be a number of options to reuse the MNO assets like network, SIM etc.
		• Using HTTP Header enrichment based seamless authentication, for LOW LOA scenarios, the RP/Client requests the LOA using the acr_values request parameters.
		• Using SIM applet to authenticate the user:
		  ○ “Click OK” for LOW LOA request [using the acr_values request parameter]
		  ○ Personal PIN as the 2FA
		Personal PIN + Mobile Signature for VERY HIGH LOA request

Proposition	OIDC Mobile Connect Element	Additional Comments
User consent and authorisation management	OIDC Authorisation Request	The consent and user authorization is done through the standard OAuth 2.0 user authorization flows.
User profile attributes assertion and exchange	UserInfo Endpoint	The Profile Attributes are accessed using the UserInfo end-point through the access token as the secure/authorized token.

Table 1: Mobile Connect proposition to OIDC mapping

2.2 OIDC Protocol Suite

As specified in the OpenID Connect specification suite [OpenID Connect], here is a protocol map for OpenID Connect:

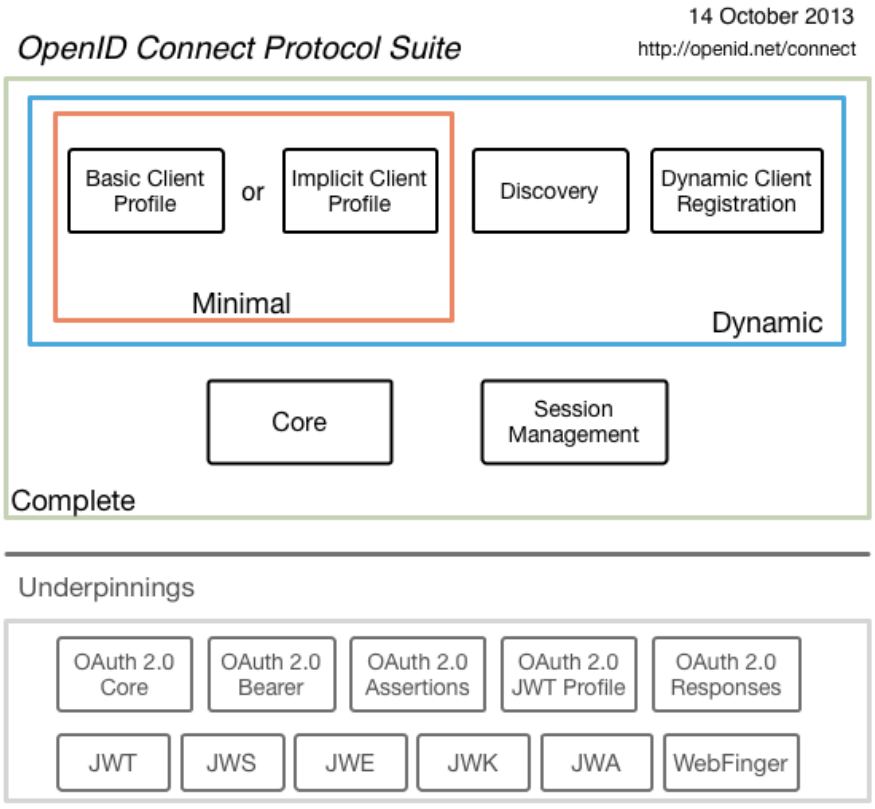

Figure 1: OIDC Protocol Suite

2.3 OIDC Abstract Protocol Flow

OIDC reuses the OAuth 2.0 protocol and parameters, and extends on OAuth 2.0 to introduce an Identity Layer through the following additions::

- Along with access token, an ID token is returned, which is a JSON Web Token [JWT] (http://tools.ietf.org/html/draft-ietf-oauth-json-web-token-14) with identity claims
- A UserInfo endpoint is introduced, which returns basic profile attributes against the access token

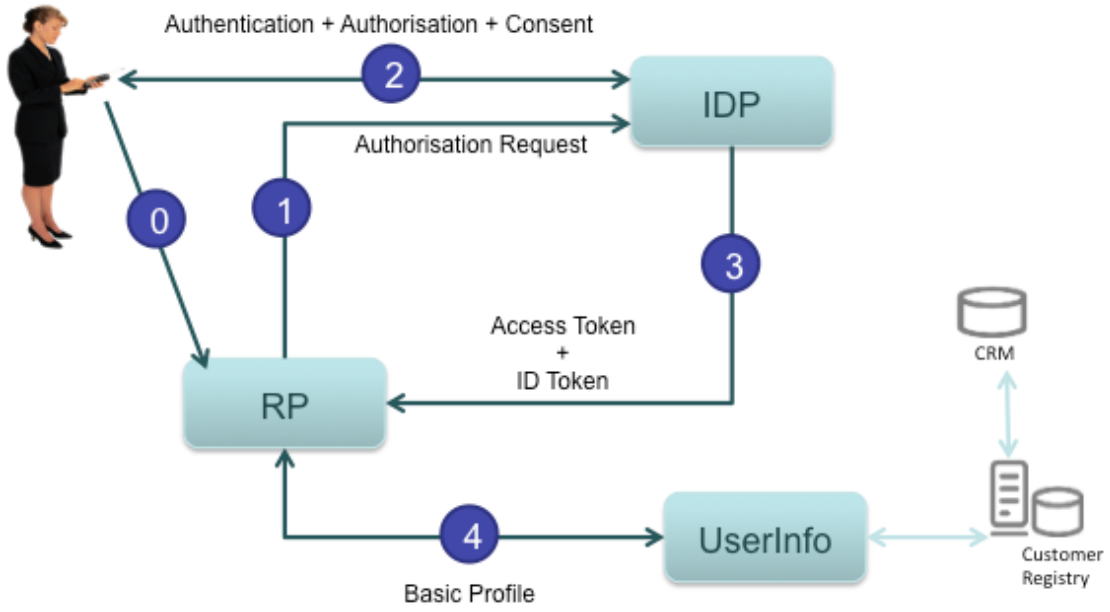

Figure 2: OIDC Abstract Protocol Flow

The above diagram illustrates an abstract flow; the actual flow may have additional steps depending on the authorisation grant model used, e.g. the Authorisation Code grant model needs an additional flow to get the access token using the authorisation code.

Here is a description of the flows:

The user is using the service from the SP and the use case needs to authenticate the user

1. The SP prepares the OIDC Authorisation request and sends that to the Authorisation end-point at the IDP (e.g. MNO acting as the IDP), passing the LoA needed in the Request Object
    - The entry-point to the IDP can be the ID Gateway
2. The IDP selects the appropriate authenticator for the LoA and authenticates the user
3. The IDP returns the response – depending on the grant-type used, e.g. for Authorisation Code grant-type, the Authorisation Code is returned, or the access token along with the ID Token is returned to the SP
    - The SP gets the anonymised identifier and the authentication context [when, how the authentication was performed]
4. If needed, the SP can call the UserInfo end-point at the IDP to get the basic attributes, passing the access token

2.4 Registration/Provisioning of RP/Client

It is required that application developers first register their client applications with the IDP/Authorisation Server. This step is required to improve end user information security. Registration involves at least the following:

- The client MUST provide one or more redirect_uri to be used for responding back for Authorisation Requests through redirect
- The client will receive client_id, client_secret to be used for request authentication and authorisation. The client_secret is not used in the Implicit Flow.

2.5 Grant Types and Flows

The OpenID Connect authentication requests can use 3 flow types, as specified in the OpenID Connect specification:

- Authorisation Code Flow
- Implicit Flow
- Hybrid Flow

For the Mobile Connect Profile, the Authorisation Code Flow is the recommended option (and is in scope for this document).

The grant type flows determine how the Access Token and the ID Token are returned to the Relying Party. The flow to be used is decided by the RP/Client and is determined by the response_type parameter in the Authorisation Request initiated by the SP/Client towards the IDP/Authorisation Server.

Feature	Authorisation Code Flow	Implicit Flow
Tokens returned from Token EndPoint	Yes	No
Tokens not revealed to User Agent	Yes	No
Client can be Authenticated	Yes	No
Refresh Token Possible	Yes	No
Communication in one round-trip	No	Yes
Server-to-server communication	Yes	No

Table 2: Differences between the grant-type flows

Mobile Applications, with a server side support SHOULD use the Authorisation Code flow.

Mobile Applications, without server side support SHOULD use the Implicit flow.

2.5.1 Authorisation Code Flow

The Authorisation Code flow uses a 2 step process to obtain the Access Token + ID Token

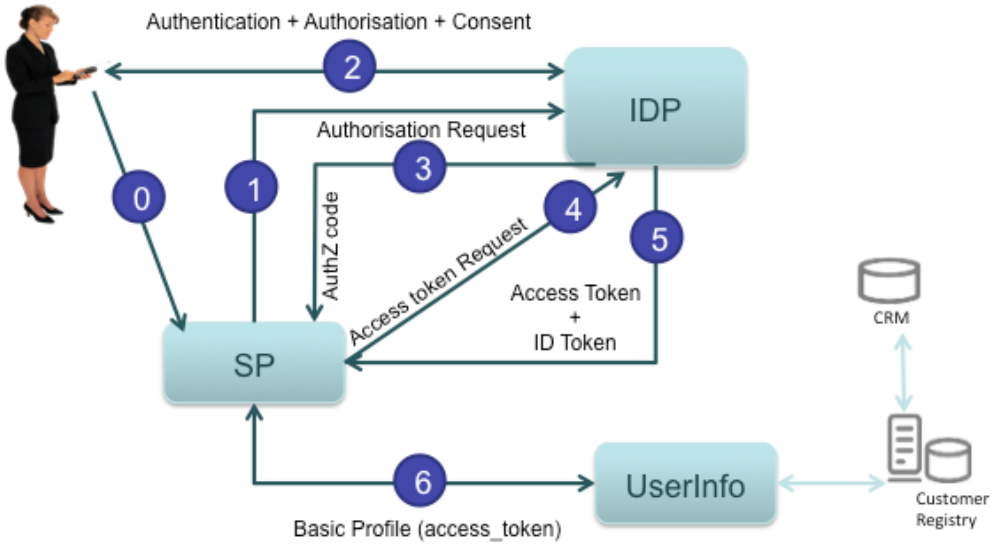

Figure 3: Authorisation Code Flow

Here is a description of the flows:

The user is using the service from the SP and the use case needs to authenticate the user

1. The SP prepares the OIDC Authorisation request and sends that to the Authorisation end-point at the IDP (e.g. MNO acting as the IDP), passing the LoA needed in the Request Object
    - The entry-point to the IDP can be the ID Gateway
2. The IDP selects the appropriate authenticator for the LoA and authenticates the user
    - This is an indicative step – this step can happen at a later stage – at Step 4, when tokens are requested
3. The IDP returns the Authorisation Code to the SP
4. The SP sends the token request to the token end-point at the IDP, passing the Authorisation code
5. The IDP validates the Authorisation code and returns the access token along with the ID Token JWT – containing the authorisation context
    - The SP gets the anonymised identifier and the authentication context [when, how the authentication was performed]
6. If needed, the SP can call the UserInfo end-point at the IDP to get the basic attributes, passing the access token

2.5.1.1 RP/Client prepares the Authorisation Request

Requirements:

- The communication with the IDP for the Authorisation MUST use TLS
- The request MUST use HTTP GET
- The request parameters are added using Query String serialisation

2.5.1.2 Request Parameters

Note: The changes in the profile for the “Mandatory” tag from the OIDC core spec are highlighted in green in the table below.

Parameter	Mandatory in Spec	Mandatory in Profile	Description
response_type	Mandatory	Mandatory	The value MUST be “code”, to indicate that the grant type flow to be used is Authorisation Code. It also indicates that the access_token (and id_token) be returned in exchange of “code”.
client_id	Mandatory	Mandatory	Needed for OAuth 2.0 authorisation request.
Scope	Mandatory	Mandatory	Space delimited and case-sensitive list of ASCII strings for OAuth 2.0 scope values. OIDC Authorisation request MUST contain the scope value “openid”. The other optional values for scope in OIDC are: “profile”, “email”, “address”, “phone” and “offline_access”.
redirect_uri	Mandatory	Mandatory	The URI where the response will be sent through redirection. The URI MUST match one of the pre-registered redirect_uris at client registration/provisioning.
state	Recommended	Mandatory	Value used by the client to maintain state between request and callback. A security mechanism as well, if a cryptographic binding is done with the browser cookie, to prevent Cross-Site Request Forgery.
nonce	Optional	Mandatory	String value used to associate a client session with the ID Token. It is passed unmodified from Authorisation Request to ID Token. The value SHOULD be unique per session to mitigate replay attacks.
display	Optional	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 added. 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].

Parameter	Mandatory in Spec	Mandatory in Profile	Description
			touch: The Authorisation 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	Recommended	Space delimited, case-sensitive ASCII string values to specify to the Authorisation Server whether to prompt or not for reauthentication and consent. The values can be: **none:** MUST NOT display any UI for reauthentication or consent to the user. If the user is not authenticated already, or authentication or consent is needed to process the Authorisation Request, a login_required error is returned. This can be used as a mechanism to check existing authentication or consent. **login:** SHOULD prompt the user for reauthentication or consent. In case it cannot be done, an error MUST be returned. **consent:** SHOULD display a UI to get consent from the user. **select_account:** In the situations, where the user has multiple accounts with the IDP/Authorisation Server, this SHOULD prompt the user to select the account. If it cannot be done, an error MUST be returned.
max_age	Optional	Recommended	Specifies the maximum elapsed time in seconds since last authentication of the user. If the elapsed time is greater than this value, a reauthentication MUST be done. When this parameter is used in the request, the ID Token MUST contain the auth_time claim value.
ui_locales	Optional	Optional	Space separated list of user preferred languages and scripts for the UI as per RFC5646. This parameter is for guidance only and in case the locales are not supported, error SHOULD NOT be returned.
claims_locales	Optional	Optional	Space separated list of user preferred languages and scripts for the Claims

Parameter	Mandatory in Spec	Mandatory in Profile	Description
			being returned as per RFC5646. This parameter is for guidance only and in case the locales are not supported, error SHOULD NOT be returned.
id_token_hint	Optional	Optional	Generally 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 ID Token is still valid and the user is logged in then the server returns a positive response, otherwise SHOULD return a login_error response. For the ID Token, the server need not be listed as audience, when included in the id_token_hint.
login_hint	Optional	Optional	An indication to the IDP/Authorisation Server on what ID to use for login, e.g. emailid, MSISDN (phone_number) etc. It is Recommended that the value matches the value used in Discovery.
acr_values	Optional	Mandatory	Authentication Context class Reference. Space separated string that specifies the Authentication Context Reference to be used during authentication processing. The LOA required by the RP/Client for the use case can be used here. The values appear as order of preference. The acr satisfied during authentication is returned as acr claim value. The recommended values are the LOAs as specified in ISO/IEC 29115 Clause 6 – 1, 2, 3, 4 – representing the LOAs of LOW, MEDIUM, HIGH and VERY HIGH. The acr_values are indication of what authentication method to used by the IDP. The authentication methods to be used are linked to the LOA value passed in the acr_values. The IDP configures the authentication method selection logic based on the acr_values.

Table 3: Request Parameters for Authorisation Request

2.5.1.3 Additional Request Parameters for the Mobile Connect Profile

Parameter	Mandatory in Spec	Description
dtbs	Optional [Mandatory for LoA = 4 use cases]	Data To Be signed. The Data/String to be signed by the private key owned by the end-user. The signed data is returned in the ID Claim, as private JWT claims for this profile.

Table 4: Additional Request parameters for Authorisation Request

2.5.1.4 RP/Client sends the Authorisation Request to the IDP/Authorisation server

The request is sent using HTTPS / TLS to the IDP/Authorisation Server using GET or POST.

Figure 4: Sample Request for Authorisation

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


HTTP/1.1

Host: mid.example.com
Accept: application/json
```

2.5.1.5 IDP/Authorisation Server authenticates user, gets user consent, returns "code" to the RP/Client

The code is returned to the URI value specified in the redirect_uri, response parameters are included as query parameters encoded using application/x-www-form-urlencoded.

Parameter	Mandatory in Spec	Mandatory in Profile	Description
code	Mandatory	Mandatory	Authorisation Code as per OAuth 2.0
state	Mandatory [if state was added in the request]	Mandatory	MUST be same as the state value added in the request.

Table 5: Response Parameters for Authorisation

```
HTTP/1.1 302 Found
Location: https://client.mid.org?code=SplxlOBeZQQYbYS6WxSbIA
&state=af0ifjsldkj
```

Figure 5: Sample Repsonse

In case the user authentication fails or user does not provide consent, error Authorisation

Response as per OAuth 2.0 SHOULD be returned.

2.5.1.6 RP/Client requests for Access Token + ID Token

Communication with the Token End-Point MUST use TLS. The request encoding used is application/x-www-form-urlencoded.

Parameter	Mandatory in Spec	Mandatory in Profile	Description
grant_type	Mandatory	Mandatory	The value MUST be set to authorization_code
code	Mandatory	Mandatory	The authorisation code received from the authorisation server, from the authorisation request
redirect_uri	Mandatory	Mandatory	The redirect_uri value MUST match the one sent in the authorisation request
client credential	Mandatory	Mandatory	The client_secret used in HTTP Basic Authentication using the OAuth 2.0 Client Password mechanism [RFC 6749 Section 2.3.1]

Table 6: Request Parameters for Token Request

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

grant_type=authorization_code&code=SplxlOBeZQQYbYS6WxSbIA
&redirect_uri=https%3A%2F%2Fclient%2Emid%2Ecom
```

Figure 6: Sample Request for Token Request

2.5.1.7 RP/Client gets the Tokens (Access Token + ID Token)

The response is in accordance with OAuth 2.0 and SHOULD be encoded in UTF-8.

Note: The changes in the profile for the “Mandatory” tag from the OIDC core spec are highlighted in green in the table below.

Parameter	Mandatory in Spec	Mandatory in Profile	Description
access_token	Mandatory	Mandatory	OAuth 2.0 access_token, used to get the UserInfo object from the UserInfo end-point and can be reused for accessing other protected resources, if required.
token_type	Mandatory	Mandatory	MUST be “bearer” unless another token_type value as agreed between the RP/Client and the IDP/Authorisation Server. For the Mobile Connect Profile, token_type=bearer is the recommended value.

Parameter	Mandatory in Spec	Mandatory in Profile	Description
id_token	Mandatory	Mandatory	This is the additional token used in OIDC to provide the Identity token claim.
expires_in	Optional	Recommended	Expiration time in seconds from the time of generation of the response.
refresh_token	Optional	Optional	OAuth 2.0 referesh token to get the access_token when the access_token expires.

Table 7: Response parameters for Token

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

{
"access_token":"SlAV32hkKG",
"token_type":"Bearer",
"expires_in":3600,
"refresh_token":"tGzv3JOkF0XG5Qx2TlKWIA",
"id_token":"eyJ0 ... NiJ9.eyJ1c ... I6IjIifX0.DeWt4Qu ... ZXso"
}
```

Figure 7: Sample Response for Token

2.6 ID Token

ID Token is an extension to the OAuth 2.0 token (Access Token) to provide the claims for Authentication Context/Event, represented as a JWT (http://tools.ietf.org/html/draft-ietf-oauth-json-web-token-14).

The ID Token is created and returned by the IDP (e.g. MNO).

Claims used in an ID Token:

Note: The changes in the profile for the “Mandatory” tag from the OIDC core spec are highlighted in green in the table below.

Parameter	Mandatory in Spec	Mandatory in Profile	Description
iss	Mandatory	Mandatory	Issuer Identifier. It is a case-sensitive HTTPS based URL, with host. It MAY contain the port and path element (Optional) but no query parameters.
sub	Mandatory	Mandatory	Subject Identifier. A unique (locally) identifier of the end-user. It is a case-sensitive ASCII string with a maximum length of 255. The ACR [Anonymous Customer Reference -

Parameter	Mandatory in Spec	Mandatory in Profile	Description
			https://tools.ietf.org/html/draft-uri-acr-extension-04] can be used as the “sub” value, if possible.
aud	Mandatory	Mandatory	The intended audience for the ID Token. It is an array of case-sensitive strings. It MUST contain the client_id of the RP/Client and MAY contain identifiers of other optional audiences.
exp	Mandatory	Mandatory	The expiration time after which the ID Token MUST NOT be accepted for processing. Its represented as the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time specified.
iat	Mandatory	Mandatory	The time of issue of the ID Token. Its represented as the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time specified.
auth_time	Mandatory [if max_age was used in the Request], Optional otherwise.	Mandatory	Time of end-user authentication. Its represented as the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time specified.
nonce	Mandatory [If nonce was used in the Authorisation Request], Optional otherwise.	Mandatory	Opaque string value to associate the RP/Client session with the ID Token, to avoid the replay attacks. The nonce value MUST be same as the nonce used in the Authorisation request. For the Mobile Connect Profile it’s a recommended parameter.
at_hash	Optional	Recommended [SHA-256 is the recommended hash algorithm]	A base64url encoded value of the hash of the access_token [the hash algorithm is negotiated during registration].
acr	Optional	Mandatory	Authentication Context Class Reference. It’s a case sensitive string, representing the fact that the authentication process followed the acr [e.g. LOA] requested or not.
amr	Optional	Mandatory [The values are: OK, DEV_PIN, SIM_PIN, UID_PWD,	Authentication Methods References. An array of case-sensitive strings to indicate the authentication method used. The values need to be negotiated offline..

Parameter	Mandatory in Spec	Mandatory in Profile	Description
		BIOM, HDR, OTP]	
azp	Mandatory [if the audience to the ID Token is different to the Authorised Party], Optional otherwise.	Mandatory [if the audience to the ID Token is different to the Authorised Party], Optional otherwise.	Authorised Party – the party to which the ID Token is issued. Represented as the client_id of the party.

Table 8: ID Token Claims

2.6.1 Additional ID Token Claims for the Profile

Parameter	Mandatory in Spec	Description
dts	Optional [Mandatory when dtbs is passed in the Request for Authorisation, for LoA = 4 use cases]	Data Signed. The signed data with the user's private key]
upk	Optional [Mandatory when dts is returned]	User Public Key.
dts_time	Optional [Mandatory when dts is returned]	The time of signing. Its represented as the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time specified.

Table 9: Additional Claims for Signed Response

2.7 Scope parameter

OIDC "scope" values determine the specific set of claim values to return in the response.

Scope values definition:

The scope value "openid" is mandatory to indicate that the request is an OpenID Connect request. The other scope values are related to the UserInfo.

Note: The changes in the profile for the "Mandatory" tag from the OIDC core spec are highlighted in green in the table below.

Parameter	Mandatory in Spec	Mandatory in Profile	Description
openid	Mandatory	Mandatory	This value indicates the Request is an OpenID Connect request.
profile	Optional	Optional	Requests access to the default basic profile claims: name, family_name, given_name, middle_name, nickname, preferred_username, profile, picture,

Parameter	Mandatory in Spec	Mandatory in Profile	Description
			website, gender, birthdate, zoneinfo, locale, and updated_at.
email	Optional	Optional	Request access to the claims : email and email_verified
address	Optional	Optional	Requests access to the claim : address
phone	Optional	Optional	Requests access to the claims : phone_number and phone_number_verified
offline_access	Optional	Optional	Requests that the Refresh Token to obtain the Access Token to get the UserInfo in case of the user is not logged in [no user present]

Table 10: Scope parameters

2.8 UserInfo

The UserInfo is an OAuth 2.0 protected resource that returns claimed identity attributes about the authenticated user.

The UserInfo resource is represented by a HTTPS URL and MAY have port, path and query parameters.

The UserInfo response is returned as a JSON object.

In the UserInfo response, if a claimed attribute cannot be returned, the name MUST be removed from the JSON object. Null or blank values are not allowed in the UserInfo response JSON object.

UserInfo claimed attributes:

Note: The changes in the profile for the “Mandatory” tag from the OIDC core spec are highlighted in green in the table below.

Attribute	Mandatory in Spec	Type	Description
sub	Mandatory	String	Subject Identifier of the user.
name	Optional	String	User’s full name, in a form that it can be displayed.
given_name	Optional	String	First name(s) of the user, separated by space.
family_name	Optional	String	Last name(s) of the user, separated by space.
middle_name	Optional	String	Middle name(s) of the user [if used], separated by space.
nickname	Optional	String	Casual name used by the user. MAY or MAY NOT be the given_name.

Attribute	Mandatory in Spec	Type	Description
preferred_username	Optional	String	Shortname that the user prefers to be referred to be at the RP/Client. The value does not need to be unique at the RP. It MAY be a valid JSON string.
profile	Optional	String	URL for the user's profile page.
picture	Optional	String	URL for the user's profile image. The URL MUST refer to an image file and not a page.
website	Optional	String	URL for user's information, content page like a blog etc.
email	Optional	String	Preferred email address of the user. MUST follow the RFC5322 syntax. MUST NOT be considered as unique at the RP/Client.
email_verified	Optional	Boolean	TRUE if the email is verified that it is controlled and owned by the user, otherwise false.
gender	Optional	String	Values used are: female; male
birthdate	Optional	String	User's birthdate, represented as per ISO 8601:2004 YYYY-MM-DD format. The year can be omitted using YYYY = 0000, if that's what is preferred by the user.
zoneinfo	Optional	String	String from the zoneinfo TimeZone database [http://www.twinsun.com/tz/tz-link.htm], representing the user's timezone.
locale	Optional	String	User's locale as per RFC 5646. The value is ISO 639-1 Alpha-2 language code in lower case and ISO 3166-1 Alpha-2 country code in upper case, the 2 values separated by a dash [e.g. en-GB].
phone_number	Optional	String	User's preferred phone number in E.164 format including the international prefix e.g. +1 for the USA
phone_number_verfied	Optional	Boolean	TRUE if the phone number is verified, FALSE otherwise.
address	Optional	JSON Object	User's preferred address as a JSON object.
updated_at	Mandatory	Number	Time at which the user's profile data was last updated. Its represented as the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time.

Table 11: UserInfo Attributes

The UserInfo endpoint MUST return a content-type header:

Content-Type	Format
Application/json	JSON object in plain-text

Table 12: Header – content-type

2.8.1 Address Format

The Address attribute represents a physical mailing address. The IDP MAY return a subset of the fields, depending on the data available for the end-user and also taking into account the user's privacy instructions and preferences. The Address Fields:

Field	Description
formatted	Full mailing address, formatted for display. MAY contain multiple lines, separated by newline characters.
street_address	MAY contain house number, street name, PO Box number. If using multiple lines, the lines are separated by newline characters.
locality	City, Town
region	State, Province, County
postal_code	Post Code, ZIP code
country	Country name

Table 13: Address Format

Annex A Document Management

A.1 Document History

Version	Date	Brief Description of Change	Approval Authority	Editor / Company
1.0	08-05-2014	New PRD	PDATA PET, PSMC	Gautam Hazari / GSMA

Other Information

Type	Description
Document Owner	PDATA
Editor / Company	Gautam Hazari / 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.