Abstract

The OpenID Connect protocol defines an identity federation system that allows a relying party to request and receive authentication and profile information about an end user.

This specification profiles the OpenID Connect protocol to increase baseline security, provide greater interoperability, and structure deployments in a manner specifically applicable to (but not limited to) government and public service domains **in the Netherlands**.

This profile builds on top of, and inherits all properties of, the OAUTH profile for iGov **in the Netherlands ("iGov-NL")**.

iGov-NL

Dutch government Assurance profile for OpenID Connect

This profile is based upon the international government assurance profile for OpenID Connect (iGov) [iGov.OIDC] as published by the OpenID Foundation (https://openid.net/foundation/). It should be considered a fork of this profile as the iGov profile is geared more towards the American situtation and in the Netherlands we have to deal with an European Union and specific Dutch context.

We have added the chapter Usecases to illustrate the specific usecase the iGov-NL profile is aimed at. Starting with chapter Introduction we follow the structure of the iGov profile. Where we do not use content from iGov we use ~~strikethrough~~ to indicate it is not part of iGov-NL. Where we have added more specific requirements for the Dutch situation this is indicated with **iGov-NL** tags or **bold** for minor changes. Note: examples in the original iGov profile are sometimes inconsistent or even invalid, these have been corrected without explicit formatting in this profile along with other changes.

As in the original iGov OpenID Connect profile, this profile focuses on a Relying Party also known as a Client. As OpenID Connect is not explicitly applicable to Resource Servers, these are left out of scope. Please refer to the iGov-NL profile for a profile dealing with interactions between Resource Servers and Authorization Servers.

Note that this profile does not detail all steps in the protocol flow, nor each option and parameter. For further details the [iGov-NL.OAuth2] profile and OAuth 2.0 and OpenID Connect specifications are applicable.

Use case

The generic use case where this profile can be applied, is very similar to the use case for the iGov-NL OAuth2 profile. A Client application wishes to identify *and authenticate* a User and may additionally want to receive User attributes from a trusted party. Authenticating the User is in addition to the use case for the OAuth2 iGov-NL profile.

Clients are restricted to web-applications in this version of the profile. Future updates may add support for native applications. Next to identification and authentication, a Client may want to receive attributes that are more reliable than those self-claimed by the User.

Flow for identification and authentication

The flow starts identical to the use case flow of the iGov-NL OAuth2 profile. As with iGov-NL OAuth2, only the authorization code flow is used in this profile. Step 1 to 5 of that profile can be applied as-is, with the distinction that the Authorization Request explicitly is an *Authentication* Request.

Step 6 will result in an Access Token and ID Token. The Access Token can be used in a UserInfo Request or in requests to a Resource Server as in regular OAuth2.

In addition to making a token Request to receive an ID Token, the Client can make a UserInfo Request. This request can be used to obtain additional information about the User.

Step 7 (OAuth2) is optional or implied. The Relying Party (= Client) can use the authentication result directly -- effectively resulting in resource server integrated in the Client -- or can make requets to a Resource Server using the obtained Access Token as in the OAuth2 use case.

/iGov-NL

1. Introduction

Government regulations for permitting users (citizens and non-citizens) online access to government resources vary greatly from region to region. There is a strong desire to leverage federated authentication and identity services for public access to government resources online to reduce 'password fatigue', increase overall account security, reduce cost**s**, and provide reliable identity assurances from established and trusted sources when applicable.

This specification aims to define an OpenID Connect profile that provides governments with a foundation for securing federated access to public services online. **This document is derived from the iGov profile by the OpenID Foundation.**

1.1. Requirements Notation and 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 .

All uses of JSON Web Signature (JWS) and JSON Web Encryption (JWE) data structures in this specification utilize the JWS Compact Serialization or the JWE Compact Serialization; the JWS JSON Serialization and the JWE JSON Serialization are not used.

1.2. Terminology

This specification uses the terms "Access Token", "Authorization Code", "Authorization Endpoint", "Authorization Grant", "Authorization Server", "Client", "Client Authentication", "Client Identifier", "Client Secret", "Grant Type", "Protected Resource", "Redirection URI", "Refresh Token", "Resource Owner", "Resource Server", "Response Type", and "Token Endpoint" defined by OAuth 2.0 , the terms "Claim Name", "Claim Value", and "JSON Web Token (JWT)" defined by JSON Web Token (JWT) , and the terms defined by OpenID Connect Core 1.0 .

1.3. Conformance

This specification defines requirements for the following components:

- OpenID Connect 1.0 relying parties (also known as OpenID Clients)
- OpenID Connect 1.0 identity providers (also known as OpenID Providers)

The specification also defines features for interaction between these components:

- Relying party to ~~identity~~ **OpenID** provider

When an iGov-**NL**-compliant component is interacting with other iGov-**NL**-compliant components, in any valid combination, all components MUST fully conform to the features and requirements of this specification. All interaction with non-iGov-**NL** components is outside the scope of this specification.

An iGov-**NL**-compliant OpenID ~~Connect IdP~~ **Provider** MUST support all features as described in this specification. A general-purpose ~~IdP~~ **OpenID Provider** MAY support additional features for use with non-iGov-**NL** clients.

An iGov-**NL**-compliant OpenID ~~Connect IdP~~ **Provider** MAY also provide iGov-compliant OAuth 2.0 authorization server functionality. In such cases, the authorization server MUST fully implement the OAuth 2.0 iGov-**NL** profile. If an iGov-compliant OpenID ~~Connect IdP~~ **Provider** does not provide iGov-compliant OAuth 2.0 authorization server services, all features related to interaction between the authorization server and protected resource**s** are therefore OPTIONAL.

An iGov-**NL**-compliant OpenID Connect client MUST ~~use~~ **implement** all ~~functions~~ **requirements** as described in this specification. A general-purpose client library MAY support additional features for use with non-iGov-**NL** IdPs.

2. Relying Party Profile

2.1. Requests to the Authorization Endpoint (Authentication Request)

The iGov-**NL** OAuth2 profile specifies requirements for requests to Authorization Endpoints - for example, when to use the PKCE parameters to secure token exchange.

In addition to the requirements specified in Section ~~2.1.1~~ **1.3.3.1** of the iGov-**NL** OAuth2 profile, the following describes the supported OpenID Connect Authorization Code Flow parameters for use with iGov-**NL** compatible ~~IdPs~~ **OpenID Providers**.

Request Parameters:

client_id
REQUIRED. OAuth 2.0 Client Identifier valid at the Authorization Server.

response_type
REQUIRED. MUST be set to code.

scope
REQUIRED. Indicates the attributes being requested. (See below)

redirect_uri
REQUIRED. Indicates a valid endpoint where the client will receive the authentication response. See (**OpenID** core section 3.1.2.1)

state
REQUIRED. Unguessable random string generated by the RP, used to protect against CSRF attacks. Must contain a sufficient amount of entropy to avoid guessing. Returned**, unchanged,** to the RP in the authentication response.

nonce
REQUIRED. Unguessable random string generated by the client, used to protect against CSRF attacks. Must contain a sufficient amount of entropy to avoid guessing. Returned**, unchanged,** to the client in the ID Token.

iGov-NL

Vectors of Trust are not actively used in the iGov-NL profile, as Europe uses level-of-assurance (LoA) policies which have a better fit with `acr`.

/iGov-NL

vtr
OPTIONAL. MUST be set to a value as described in Section 6.1 of Vectors of Trust vtr takes precedence over acr_values.

acr_values
OPTIONAL. Lists the acceptable LoAs for this authentication. See (below). MUST not be set if vtr is specified.

code_challenge and code_challenge_method
OPTIONAL. If the PKCE protocol is being used by the client. See OAUTH profile for iGov-**NL**.

iGov-NL

Note that a `redirect_uri` MUST be pre-registered with the OpenID Provider as described in the OAuth2 iGov-NL profile.

/iGov-NL

A sample request may look like:

```
https://op.example.nl/oidc/authorization?
  response_type=code
  &client_id=827937609728-m2mvqffo9bsefh4di90saus4n0diar2h
  &scope=openid
  &redirect_uri=https%3A%2F%2Frp.example.nl%2Foidc%2FloginResponse
  &state=2ca3359dfbfd0
  &nonce=9d3252993a38454c8a6c3a4b86997aaa
  &acr_values=http%3A%2F%2Feidas.europa.eu%2FLoA%2Fsubstantial
```

2.2. Requests to the Token Endpoint

In addition to the requirements specified in Section ~~2.1.2~~ **1.3.3.3** of the iGov-**NL** OAuth2 profile, the following ~~claims~~ **parameters** MUST be included:

The following parameters are specified:

grant_type
REQUIRED. MUST be set to authorization_code.

code
REQUIRED. The value of the code parameter returned in the authorization response.

client_assertion_type
REQUIRED. MUST be set to urn:ietf:params:oauth:client-assertion-type:jwt-bearer.

client_assertion
REQUIRED. The value of the signed client authentication JWT generated as described below. The RP **MUST** generate a new **client**_assertion JWT for each call to the token endpoint.

iGov-NL

In case the PKCE protocol was used by the Client in the relevant Authentication Request, the `code_verifier` element MUST be included.

code_verifier
~~OPTIONAL. If~~ **REQUIRED in case** the PKCE protocol was used by the Client in the relevant Authentication Request, the code_verifier MUST be present and the value MUST be the original random code, used to create the hashed challenge in `code_challenge`.

/iGov-NL

2.3. ID Tokens

All clients MUST validate the signature of an ID Token before accepting it using the public key of the issuing server, which is published in JSON Web Key (JWK) format. ID Tokens MAY be encrypted using the appropriate key of the requesting client.

Clients MUST verify the following in received ID tokens:

iss
The "issuer" field is the Uniform Resource Locat~~e~~**o**r (URL) of the ~~expected issuer~~ **OpenID Provider**

aud
The "audience" field contains the client ID(s) of the ~~client~~**Relying Party or Relying Parties. Any Relying Party MUST validate they are listed as audience for this ID-token.**

exp, iat, nbf
The "expiration", "issued at", and "not before" timestamps for the token are dates (integer number of seconds since from 1970-01-01T00:00:00Z UTC) within acceptable ranges**.**

iGov-NL

nonce
The "nonce" field MUST be used by Clients to detect/prevent CSRF, replay and other attacks and Clients MUST verify that the nonce Claim Value is equal to the value of the nonce parameter sent in the Authentication Request.

acr
Client**s** MUST validate the authentication context class reference, if present, satisfies the minimum required before authorizing access to any resource or performing any operation on behalf of the identified subject.

/iGov-NL

2.4. Request Objects

Clients MAY optionally send requests to the authorization endpoint using the request parameter as defined by OpenID Connect . Clients MAY send requests to the authorization endpoint by reference using the request_uri parameter.

Request objects MUST be signed by the client's registered key. Request objects MAY be encrypted to the authorization server's public key.

iGov-NL

In case request objects using pass by reference are used, the `request_uri` parameter value MUST be pre-registered. The URI MUST be using either a URN or an absolute HTTPS URI, and MUST contain a sufficient amount of entropy to avoid guessing. To avoid attacks by `request_uri` manipulation, the base value of the URI MUST be registered prior to usage and SHOULD be unambiguously distinct from other URIs in use. Since the `request_uri` should at the same time differ for each unique request due to caching considerations, a HTTPS URI MUST use a parameter identifying the request object. The `request_uri` value MUST be reachable or retrievable by the OP. The server hosting the `request_uri` MAY require authentication of the OP.

/iGov-NL

2.5. OpenID Provider Discovery

iGov-NL

Clients and Resource Servers can use OpenID configuration published by OpenID Providers to obtain information about OpenID Provider configurations.

/iGov-NL

Clients and protected resources SHOULD cache OpenID Provider metadata once an OP has been discovered and used by the client **or resource server**.

iGov-NL

In case the OP uses the recommendations in this profile, caching information is provided through HTTP headers with the OpenID Provider metadata. It is therefor RECOMMENDED for clients and resource servers to use the HTTPS headers when determining validity of cached OP metadata.

2.6 Client Authentication and Registration

Clients SHOULD be able to perform dynamic registration. If dynamic registration is not supported, it is up to the discretion of the OP whether and up to what level of detail an alternative process is provided. Pre-conditions to registration of Clients with the OP MAY be applicable, for various reasons such as organizational, legal, administrative, policy, security or technical reasons. Any such pre-conditions are out of scope of this profile.

Clients MUST be registered using the name of the responsible Service Provider and not any intermediary or supplier related names. The name MUST be familiar for the User and MUST be clearly and visibly shown to the User both at the RP's website and at the OP.

Authentication of the client using the private_key_jwt client authentication method MUST be supported. Alternative client authentication methods MAY be supported, provided they offer at least equivalent security. NOTE: the client_secret_jwt method is not considered of equivalent security and the methods client_secret_basic and client_secret_post are obviously less secure. These three methods therefor MUST NOT be used.

PKIoverheid

In case the Relying Party and OpenID Provider are not operated under responsibility of the same organization, each party MUST use PKIoverheid certificates with OIN.

PKIoverheid certificates MUST be included as `x5c` parameters in JWKs. The `x5c` parameter MUST be included as a list (array) of X.509 certificate(s), as Base64 DER encoded PKIoverheid certificate(s). The first certificate MUST be the Client's certificate, optionally followed by the rest of that certificate's chain. The jwks structure MUST include the public key parameters with the same values of the corresponding X.509 certificate included as `x5c` , as per [RFC7517] §4.7.

/iGov-NL

3. OpenID Provider Profile

3.1. ID Tokens

All ID Tokens MUST be signed by the OpenID Provider's private signature key. ID Tokens MAY be encrypted using the appropriate key of the requesting client.

~~The~~ ID Token**s** MUST expire and SHOULD have an active lifetime no longer than five minutes. Since the ID token is consumed by the client and not presented to remote systems, much shorter expiration times are RECOMMENDED where possible.

The token response includes an access token (which can be used to make a UserInfo request) and ID token (a signed and optionally encrypted JSON Web Token). ID Token values have the following meanings:

iss
REQUIRED. The "issuer" field is the Uniform Resource Locat**eo**r (URL) of the ~~expected issuer~~ **OpenID Provider**.

aud
REQUIRED. The "audience" field contains the client ID(s) of the ~~client~~**Relying Party or Relying Parties**.

sub
REQUIRED. The identifier of the user. OpenID Providers MUST support a pairwise identifier in accordance with OpenID Connect section 8.1. See Pairwise Identifiers below on when it may be useful to relax this requirement.

iGov-NL

The parameters `vot` and `vtm` are not actively used in the iGov-NL profile, as Europe uses level-of-assurance (LoA) policies which fit best in `acr` .

/iGov-NL

vot
OPTIONAL. The vector value as specified in Vectors of Trust . See Vectors of Trust for more details. vot takes precedence over acr.

vtm
REQUIRED if vot is provided. The trustmark URI as specified in Vectors of Trust . See Vectors of Trust for more details.

acr
OPTIONAL. The LoA the user was authenticated at. MUST be a member of the acr_values list from the authentication request **or that was agreed upon through other means**. The OpenID Provider MUST NOT include this field if vot is provided. See Authentication Context for more details.

nonce
REQUIRED. MUST match the nonce value that was provided in the authentication request.

jti
REQUIRED. A unique identifier for the token, which can be used to prevent reuse of the token.

auth_time
RECOMMENDED. This SHOULD be included if the **OpenID** Provider can assert an end-user's authentication intent was demonstrated. For example, a login event where the user took some action to authenticate.

exp, iat, nbf
REQUIRED. The "expiration", "issued at", and "not before" timestamps for the token are dates (integer number of seconds since from 1970-01-01T00:00:00Z UTC) within acceptable ranges.

This example ID token has been signed using the server's RSA key:


```
eyJhbGciOiJSUzI1NiJ9.eyJhdXRoX3RpbWUiOjE0
MTg2OTg3ODIsImV4cCI6MTQxODY5OTQxMiwic3ViI
joiNldaUVBwblF4ViIsIm5vbmNlIjoiMTg4NjM3Yj
NhZjE0YSIsImF1ZCI6WyJjMWJjODRlNC00N2VlLTR
iNjQtYmI1Mi01Y2RhNmM4MWY3ODgiXSwiaXNzIjoi
aHR0cHM6XC9cL2lkcC1wLmV4YW1wbGUuY29tXC8iL
CJpYXQiOjE0MTg2OTg4MTJ9mQc0rtL56dnJ7_zO_f
x8-qObsQhXcn-qN-FC3JIDBuNmP8i11LRA_sgh_om
RRfQAUhZD5qTRPAKbLuCD451lf7ALAUwoGg8zAASI
5QNGXoBVVn7buxPd2SElbSnHxu0o8ZsUZZwNpircW
NUlYLje6APJf0kre9ztTj-5J1hRKFbbHodR2I1m5q
8zQR0ql-FoFlOfPhvfurXxCRGqP1xpvLLBUi0JAw3
F8hZt_i1RUYWMqLQZV4VU3eVNeIPAD38qD1fxTXGV
```

```
Ed2XDJpmlcxjrWxzJ8fGfJrbsiHCzmCjflhv34O22
zb0lJpC0d0VScqxXjNTa2-ULyCoehLcezmssg
```

Its claims are as follows:

```
{
  "iss": "https://op.example.nl/",
  "aud": [
    "c1bc84e4-47ee-4b64-bb52-5cda6c81f788"
  ],
  "sub": "6WZQPpnQxV",
  "nonce": "188637b3af14a",
  "acr":"http://eidas.europa.eu/LoA/substantial",
  "auth_time": 1418698782,
  "iat": 1418698812,
  "nbf": 1418698812,
  "exp": 1418699412,
          "jti": "b42e57f8-4cfa-474a-afed-f0e9a77880c9"
}
```

3.2. Pairwise Identifiers *and Public Identifiers*

Pairwise identifiers specified in OpenID Connect Core section 8 help protect an end user's privacy by allowing an OpenID Provider to represent a single user with a different subject identifier (sub) for every client the user connects to. This technique can help mitigate correlation of a user between multiple clients by preventing the clients from using the subject identifier (the sub claim) to track a user between different sites and applications. Use of pairwise identifiers does not prevent clients from correlating data based on other identifying attributes such as names, phone numbers, email addresses, document numbers, or other attributes. However, since not all transactions require access to these attributes, but a subject identifier is always required, a pairwise identifier will aid in protecting the privacy of end users as they navigate the system.

OpenID Providers MUST support pairwise identifiers for cases where clients require this functionality. OpenID Providers MAY support public identifiers for frameworks where public identifiers are required, or for cases where public identifiers are shared as attributes and the framework does not have a requirement for subject anonymity.

iGov-NL

The Netherlands has standardized on using a citizen identification number (*BurgerServiceNummer* or BSN), for citizen to government related interactions. Usage of the BSN is restricted by Dutch law. The BSN therefor SHOULD be protected by additional security controls. For example, a BSN in the `sub` parameter can be encrypted to the Relying Party.

A `sub` containing a BSN SHOULD be handled using `subject_type=public`. For use cases where the BSN is not explicitly applicable, alternative identifiers SHOULD be used. Such subject identifiers can be either public or pairwise, depending on the identifier and use case. An OP conforming to this profile SHOULD support public identifiers.

/iGov-NL

3.3. UserInfo Endpoint

OpenID Providers MUST support the UserInfo Endpoint and, at a minimum, the sub (subject) claim. It is expected that the sub claim will remain pseudonymous in use cases where obtaining personal information is not needed.

Support for a UserInfo Endpoint is important for maximum client implementation interoperability even if no additional user information is returned. Clients are not required to call the UserInfo Endpoint, but should not receive an error if they do.

In an example transaction, the client sends a request to the UserInfo Endpoint like the following:

```
GET /userinfo HTTP/1.1
Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJleHAiOjE0MTg3MDI0MTIsIm
 F1ZCI6WyJjMWJjODRlNC00N2VlLTRiNjQtYmI1Mi01Y2RhNmM4MWY3ODgiXSwiaXNzIjo
 iaHR0cHM6XC9cL2lkcC1wLmV4YW1wbGUuY29tXC8iLCJqdGkiOiJkM2Y3YjQ4Zi1iYzgx
 LTQwZWMtYTE0MC05NzRhZjc0YzRkZTMiLCJpYXQiOjE0MTg2OTg4MTJ9i.HMz_tzZ90_b
 0QZS-AXtQtvclZ7M4uDAs1WxCFxpgBfBanolW37X8h1ECrUJexbXMD6rrj_uuWEqPD738
 oWRo0rOnoKJAgbF1GhXPAYnN5pZRygWSD1a6RcmN85SxUig0H0e7drmdmRkPQgbl2wMhu
 -6h2Oqw-ize4dKmykN9UX_2drXrooSxpRZqFVYX8PkCvCCBuFy2O-HPRov_SwtJMk5qjU
 WMyn2I4Nu2s-R20aCA-7T5dunr0iWCkLQnVnaXMfA22RlRiU87nl21zappYb1_EHF9ePy
 q3Q353cDUY7vje8m2kKXYTgc_bUAYuW-W3SMSw5UlKaHtSZ6PQICoA
```

```
Accept: text/plain, application/json, application/*+json, */*
Host: op.example.nl
Connection: Keep-Alive
```

And receives a document in response like the following:

```
HTTP/1.1 200 OK
Date: Tue, 16 Dec 2014 03:00:12 GMT
Access-Control-Allow-Origin: *
Content-Type: application/json;charset=ISO-8859-1
Content-Language: en-US
Content-Length: 333
Connection: close

{
   "sub": "6WZQPpnQxV",
   "iss": "https://op.example.nl"
   "given_name": "Stephen",
   "family_name": "Emeritus",
}
```

OpenID Providers MUST support the generation of JWT encoded responses from the UserInfo Endpoint in addition to unsigned JSON objects. Signed responses MUST be signed by the OpenID Provider's key, and encrypted responses MUST be encrypted with the authorized client's public key. The OpenID Provider MUST support the RS256 signature method (the Rivest, Shamir, and Adleman (RSA) signature algorithm with a 256-bit hash) and MAY use other asymmetric signature and encryption methods**, which provide equivalent or better security,** listed in the JSON Web Algorithms (JWA) specification.

iGov-NL

In addition to RS256, OpenID Providers SHOULD support using the PS256 (RSASSA-PSS using SHA-256 and MGF1 with SHA-256) signature algorithm.

/iGov-NL

3.4. Request Objects

OpenID Providers MUST accept requests containing a request object signed by the client's private key. Servers MUST validate the signature on such requests against the client's registered public key. OpenID Connect Providers MUST accept request objects encrypted with the server's public key.

OpenID Providers MAY accept request objects by reference using the request_uri parameter.

Both of these methods allow for clients to create a request that is protected from tampering through the browser, allowing for a higher security mode of operation for clients and applications that require it. Clients are not required to use request objects~~., but~~ OpenID Providers ~~are required to~~ **MUST** support requests using ~~them~~ **signed request objects passed by value, MAY support requests using signed request objects pass by reference and MAY support encryption of (signed) request objects**.

iGov-NL

In case request objects using pass by reference are used, the `request_uri` parameter value MUST be pre-registered. The URI MUST be using either a URN or an absolute HTTPS URI and MUST contain a sufficient amount of entropy to avoid guessing. To avoid attacks by `request_uri` manipulation, the base value of the URI MUST be registered prior to usage and SHOULD be unambiguously distinct from other URIs in use. The `request_uri` value MUST be reachable or retrievable by the OP. The OP MUST support authenticating to the server hosting the `request_uri`.

/iGov-NL

3.5. Vectors of Trust

iGov-NL

Vectors of Trust are not actively used in the iGov-NL profile, as Europe has standardized on level-of-assurance (LoA) policies under eIDAS which fit best in acr. Although vectors of trust provide more granularity and support for some anonymous use cases, these are less common in citizen-to-government use cases and therefor not applicable.

/iGov-NL

If the vtr (Vectors of Trust Request) value is present in the authorization request as defined in the Vectors of Trust standard, the OpenID Provider SHOULD respond with a valid vot value as defined in [section 3.1]. Both the vtr and vot MUST contain values in accordance with the Vectors of Trust standard. These values MAY be those defined in the Vectors of Trust standard directly or MAY be from a compatible standard. The OpenID Provider MAY require the user to re-authenticate, provide a second factor, or perform another action in order to fulfill the state requested in the vtr.

For backwards compatibility clients MAY send an acr_values parameter. If both the vtr and acr_values are in the request, the vtr MUST take precedence and the acr_values MUST be ignored.

It is out of scope of this document to determine how an organization maps their digital identity practices to valid VOT component values.

3.6. Authentication Context

OpenID Providers ~~MAY~~ **SHOULD** provide acr (authentication context class reference, equivalent to the Security Assertion Markup Language (SAML) element of the same name) and **MAY provide** amr (authentication methods reference) values in ID tokens only if vtr is not used.

iGov-NL

As Europe has standardized on level-of-assurance (LoA) policies under eIDAS, the `acr` element is RECOMMENDED to be used. Valid values vary depending on context, use case and OpenID Providers in use. RECOMMENDED is to apply the three eIDAS LoAs (low, substantial and high), using the respective URIs defined by eIDAS as values.

/iGov-NL

3.7. Discovery

OpenID Connect Discovery ~~standard~~ **specifications** provides a standard, program**m**atic way for clients to obtain configuration details for communicating with OpenID Providers. Discovery is an important part of building scalable federation ecosyst~~s~~ems.

Exposing a Discovery endpoint does NOT inherently put the OpenID Provider at risk to attack. Endpoints and parameters specified in the Discovery document SHOULD be considered public information regardless of the existence of the Discovery document.

Access to the Discovery document MAY be protected with existing web authentication methods if required by the **OpenID** Provider. Credentials for the Discovery document are then managed by the **OpenID** Provider. Support for these authentication methods is outside the scope of this specification.

Endpoints described in the Discovery document MUST be secured in accordance with this specification and MAY have addit**i**onal controls the Provider wishes to support.

All OpenID Providers are uniquely identified by a URL known as the issuer. This URL serves as the prefix of a service discovery endpoint as specified in the OpenID Connect Discovery ~~standard~~ **specification**. The discovery document MUST contain at minimum the following fields:

issuer
REQUIRED. The fully qualified issuer URL of the OpenID Provider.

authorization_endpoint
REQUIRED. The fully qualified URL of the OpenID Provider's authorization endpoint defined by [RFC6749].

token_endpoint
REQUIRED. The fully qualified URL of the server's token endpoint **as** defined by [RFC6749].

iGov-NL

userinfo_endpoint
REQUIRED. The fully qualified URL of the server's user info endpoint **as** defined by [OpenID.Discovery].

/iGov-NL

introspection_endpoint
OPTIONAL. The fully qualified URL of the server's introspection endpoint **as** defined by OAuth Token Introspection.

revocation_endpoint
OPTIONAL. The fully qualified URL of the server's revocation endpoint **as** defined by OAuth Token Revocation.

jwks_uri
REQUIRED. The fully qualified URI of the server's public key in JWK Set format. For verifying the signatures on the id_token.

scopes_supported
REQUIRED. The list of scopes, including iGov-**NL** scopes, the server supports.

response_types_supported
REQUIRED. MUST be set to code, since only authorization code flow is supported by this profile.

claims_supported
REQUIRED. The list of claims available in the supported scopes. See below.

vot
OPTIONAL. The vectors supported.

acr_values_supported
OPTIONAL. The acr values supported.

iGov-NL

In case the request objects using pass by reference are supported, registration of `request_uri` is required to avoid attacks using e.g. request rewrites or abuse of open redirectors.

require_request_uri_registration
REQUIRED. MUST be set to `true`. Either a URN or absolute HTTPS URI is required to be registered in case passing request objects by reference will be used.

/iGov-NL

The following example shows the JSON document found at a discovery endpoint for an ~~authorization server~~ **OpenID Provider**:

```
{
  "issuer": "https://op.example.nl/",
  "authorization_endpoint": "https://op.example.nl/authorize",
  "request_parameter_supported": true,
  "claims_parameter_supported": true,
  "request_uri_parameter_supported": true,
  "require_request_uri_registration": true,
  "request_object_encryption_enc_values_supported": [
        "A192GCM", "A128GCM", "A256GCM"
  ],
  "request_object_encryption_alg_values_supported": [
        "RSA-OAEP", "RSA-OAEP-256"
  ],
  "request_object_signing_alg_values_supported": [
        "RS256", "RS384", "RS512", "PS256", "PS384", "PS512"
  ],
  "token_endpoint": "https://op.example.nl/token",
  "token_endpoint_auth_methods_supported": [
        "private_key_jwt",
  ],
  "token_endpoint_auth_signing_alg_values_supported": [
        "RS256", "RS384", "RS512", "PS256", "PS384", "PS512"
  ],
  "id_token_signing_alg_values_supported": [
        "RS256", "RS384", "RS512", "PS256", "PS384", "PS512"
  ],
  "id_token_encryption_alg_values_supported": [
        "RSA-OAEP", "RSA-OAEP-256"
  ],
  "id_token_encryption_enc_values_supported": [
        "A192GCM", "A128GCM", "A256GCM"
  ],
  "userinfo_endpoint": "https://op.example.nl/userinfo",
  "userinfo_signing_alg_values_supported": [
```

```
        "RS256", "RS384", "RS512", "PS256", "PS384", "PS512"
    ],
    "userinfo_encryption_alg_values_supported": [
        "RSA-OAEP", "RSA-OAEP-256"
    ],
    "userinfo_encryption_enc_values_supported": [
        "A192GCM", "A128GCM", "A256GCM"
    ],
    "introspection_endpoint": "https://op.example.nl/introspect",
    "registration_endpoint": "https://op.example.nl/register",
    "subject_types_supported": [
        "public", "pairwise"
    ],
    "jwks_uri": "https://op.example.nl/jwk",
    "service_documentation": "https://op.example.nl/about",
    "response_types_supported": [
        "code"
    ],
    "revocation_endpoint": "https://op.example.nl/revoke",
    "claim_types_supported": [
        "normal"
    ],
    "grant_types_supported": [
        "authorization_code",
    ],
    "scopes_supported": [
        "profile", "openid"
    ],
    "op_tos_uri": "https://op.example.nl/about",
    "op_policy_uri": "https://op.example.nl/about",
    "claims_supported": [
        "sub", "name", "acr"
    ],
    "acr_values_supported": [
          "http://eidas.europa.eu/LoA/substantial",
          "http://eidas.europa.eu/LoA/high"
    ]
}
```

It is RECOMMENDED that servers provide cache information through HTTP headers and make the cache valid for at least one week.

iGov-NL

PKIoverheid

In case the Relying Party and OpenID Provider are not operated under responsibility of the same organization, each party MUST use PKIoverheid certificates with OIN.

The PKIoverheid certificate MUST be included as a `x5c` parameter. The `x5c` parameter MUST be included as a list (array) of X.509 certificate(s), as Base64 DER encoded PKIoverheid certificate(s). The first certificate MUST be the Client's certificate, optionally followed by the rest of that certificate's chain. The jwks structure MUST include the public key parameters with the same values of the corresponding X.509 certificate included as `x5c` , as per [RFC7517] §4.7.

JWKS

/iGov-NL

The server MUST provide its public key in JWK Set format, such as the following 2048-bit RSA key:

```
{
  "keys": [
        {
          "alg": "RS256",
          "e": "AQAB",
          "n": "o80vbR0ZfMhjZWfqwPUGNkcIeUcweFyzB2S2T-hje83IOVct8gVg9Fx
                        vHPK1ReEW3-p7-A8GNcLAuFP_8jPhiL6LyJC3F10aV9KPQFF-w6Eq6V
                        tpEgYSfzvFegNiPtpMWd7C43EDwjQ-GrXMVCLrBYxZC-P1ShyxVBOze
                        R_5MTC0JGiDTecr_2YT6o_3aE2SIJu4iNPgGh9MnyxdBo0Uf0TmrqEI
                        abquXA1-V8iUihwfI8qjf3EujkYi7gXXelIo4_gipQYNjr4DBNl
```

```
                        E0__RI0kDU-27mb6esswnP2WgHZQPsk779fTcNDBIcYgyLujlcUATEq
                        fCaPDNp00J6AbY6w",
            "kty": "RSA",
            "kid": "rsa1",
            "x5c": "MIIFdDCCA1ygAwIBAgIEAJiiOTANBgkqhkiG9w0BAQsFADBaMQswCQYDVQQGEwJO
                          ...
                        QFH1T/U67cjF68IeHRaVesd+QnGTbksVtzDfqu1XhUisHWrdOWnk4Xl4vs4Fv6EM
                        94B7IWcnMFk="
          }
      ]
    }
```

3.8. Dynamic Client Registration

If the OP is acting as an iGov-**NL** OAuth Authorization Server (iGov-**NL** OAuth2 profile), then Dynamic Registration MUST be supported in accordance with that specification (see section ~~3.13~~ **1.4.1.3**).

4. User Info

The availability, quality, and reliability of an individual's identity attributes will vary greatly across jurisdictions and **OpenID** Provider systems. The following recommendations **aims to** ensure maximum cross-jurisdictional interoperability, while setting Client expectations on the type of data they may acquire.

4.1. Claims Supported

Discovery mandates the inclusion of the claims_supported field that defines the claims a client MAY expect to receive for the supported scopes. OpenID Providers MUST return claims on a best effort basis. However, a **OpenID** Provider asserting it can provide a user claim does not imply that this data is available for all its users: clients MUST be prepared to receive partial data. Providers MAY return claims outside of the claims_supported list, but they MUST still ensure that the extra claims to not violate the privacy policies set out by the federation.

4.2. Scope Profiles

In the interests of data minimization**,** balanced with the requirement to successfully identify the individual signing in to a service, the default OpenID Connect profiles may not be appropriate.

Matching of the identity assertion based on claims to a local identifier or 'account' related to the individual identity at a level of assurance is a requirement where the government in question is not able to provide a single identifier for all citizens based on an authoritative register of citizens.

The requirement for matching is also of importance where a cross-border or cross-jurisdiction authentication is required and therefore the availability of a single identifier (e.g. social security number) cannot be guaranteed for the individual wishing to authenticate.

This standard defines a set of common scope values that aim to provide maximum cross-jurisdictional identity matching while not being p**re**scriptive on the exact attributes shared, as every jurisdiction will likely have vari**o**us levels of information available, and different policies for sharing personal data even if it is on file.

profile

The OpenID Connect Core defined profile provides baseline identity information. It is HIGHLY RECOMMENDED that the attributes for given_name, family_name, address, birthdate be supported by iGov Providers if possible, unless data quality requirements or privacy considerations prevent it. It is left to the OpenID Provider and the trust framework to set any further limitations on this profile data - see Privacy Considerations below.

iGov-NL

As the Netherlands has standardized on using a citizen identification number (*BurgerServiceNummer* or BSN), directly referencing document numbers is discouraged in the iGov-NL profile. Usage of `doc` claims in the Netherlands is NOT RECOMMENDED.

/iGov-NL

doc

The identity document type(s) and associated "number" the Provider is capable of providing. Mu**l**tiple document types

MAY be returned. This could be passport, driver's license, national ID card, health insurance no., and so on. See the Claims below for the document claims available.

4.3. Claims Request

[OpenID.Core] section 5.5 defines a method for a client to request specific claims in the UserInfo object. OpenID Providers MUST support this claims parameter in the interest of data minimization - that is, the **OpenID** Provider only returns information on the subject the client specifically asks for, and does not volunteer additon**i**al information about the subject.

Clients requesting the profile scope MAY provide a claims request parameter. If the claims request is omitted, the OpenID Provider SHOULD provide a default claims set that it has available for the subject, in accordance with any policies set out by the trust framework the Provider supports.

iGov-NL

As the Netherlands has standardized on using a citizen identification number (*BurgerServiceNummer* or BSN), directly referencing document numbers is discouraged in the iGov-NL profile. Usage of the `doc` claims in the Netherlands is NOT RECOMMENDED.

/iGov-NL

Client**s** requesting the doc scope MUST provide a claims request parameter indicating the document type (or types) and fields they wish to receive, or they will receive none. The OpenID Provider MUST NOT return any doc related claims not included in the claims request. Client requests that include the doc scope but no claims request MUST NOT be rejected by the OpenID Provider, but simply no doc related claims are returned in the UserInfo object.

4.4. ~~Claims~~ UserInfo Response

Response to a UserInfo request MUST match the scope and claims requested to avoid having a OpenID Provider over-expose a user's identity information.

iGov-NL

As the Netherlands has standardized on using a citizen identification number (*BurgerServiceNummer* or BSN), directly referencing document numbers is discouraged in the iGov-NL profile. Usage of the `doc` claims in the Netherlands is NOT RECOMMENDED.

/iGov-NL

The document doc claims response include:

type
The document type.

num
The value of the document identifier. Note that not all values are technically numeric.

issued
The date the document was issued. Timestamp format.

expires
The date the document expires (or expired). Timestamp format.

issuer_location
The location/address of the issuing agency.

Claims response MAY also make use of the agg**r**egated and/or distributed claims structure to refer to the original source of the subject's claims.

4.5. Claims Metadata

Claims Metadata (such as locale or the confidence level the OpenID Provider has in the claim for the user) can be expressed as attributes within the UserInfo object, but are outside the scope of this document. These types of claims are best described by the trust framework the clients and OpenID Providers operate within.

5. Privacy Considerations

iGov-NL

All parties implementing this specification MUST ensure that due care is taken to protect user privacy.

/iGov-NL

Data minimization is an essential concept in trust frameworks and federations exchanging user identity information for government applications. The design of this specification takes into consideration mechanisms to protect the user's government identity information and activity from unintentional exposure.

Pairwise anonymous identifiers MUST be supported by the OpenID Providers for ~~frameworks~~ **cases** where subjects should not be traceable across clients by their subject ~~ID~~ **identifiers**. This prevents a situation where a user may inadvertently be assigned a universal government identifier.

Request claims MUST be supported by the OpenID Providers to ensure that only the data the client explicitly requests is provided in the UserInfo response. This prevents situations where a client may only require a partial set of claims, but receives (and is therefore exposed to) a full set of claims. ~~For example, if a client only needs a single government document type and number, the OpenID Provider MUST NOT send the client the full document information, possibly from multiple documents.~~

Despite the mechanisms enforced by this specification, the operational circumstances of a federation may allow these controls to be relaxed. For example, if a framework always requires clients to request a national ID number, then the pairwise anonymous identifier requirement may be relaxed. ~~In cases where all clients are entitled to all government document claims associated to a subject at an OpenID Provider, the claims request requirement may be relaxed.~~

The reasons for relaxing the controls that support data minimalization are outside the scope of this specification.

6. Security Considerations

All transactions MUST be protected in transit ~~by~~**using** TLS as described in BCP195.

iGov-NL

In addition to the Best Current Practice for TLS, it is highly RECOMMENDED for all conforming implementations to incorporate the TLS guidelines from the Dutch NCSC [NCSC.TLS] into their implementations. If these guidelines are applied:

- For back-channel communication, the guidelines categorized as "good" MUST be applied.
- For front-channel communication, the guidelines for "good" MUST be applied and the guidelines for "sufficient" MAY be applied, depending target audience and support requirements.
- Guidelines categorized as "insufficient" MUST NOT be applied and those categorized as "deprecated" SHOULD NOT be used.

/iGov-NL

All clients MUST conform to applicable recommendations found in the Security Considerations sections of [RFC6749] and those found in the OAuth 2.0 Threat Model and Security Considerations document.

7. Normative References

[BCP195]	**Sheffer, Y., Holz, R. and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015.**
~~**[HEART.OAuth2]**~~	~~Richer, J.~~, "~~Health Relationship Trust Profile for OAuth 2.0~~", ~~April 2017.~~
~~**[I-D.ietf-oauth-pop-architecture]**~~	~~Hunt, P., Richer, J., Mills, W., Mishra, P. and H. Tschofenig~~, "~~OAuth 2.0 Proof-of-Possession (PoP) Security Architecture~~", ~~Internet-Draft draft-ietf-oauth-pop-architecture-08~~, ~~July 2016~~.
[iGov.OAuth2]	Richer, J., "iGov Profile for OAuth 2.0", October 2018.
[iGov-NL.OAuth2]	**Terpstra, F., "NL GOV Assurance profile for OAuth 2.0", March 2019.__**

[BCP195]	**Sheffer, Y., Holz, R. and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015.**
[iGov.OIDC]	Varley, M., "iGov Profile for OpenID Connect 1.0", October 2018.__
[NCSC.TLS]	**NCSC of the Netherlands, "ICT-beveiligingsrichtlijnen voor Transport Layer Security (TLS)", version *2.0, April 2019*.__
[OpenID.Core]	Sakimura, N., Bradley, J., Jones, M., de Medeiros, B. and C. Mortimore, "OpenID Connect Core 1.0", August 2015.
[OpenID.Discovery]	Sakimura, N., Bradley, J., Jones, M. and E. Jay, "OpenID Connect Discovery 1.0", August 2015.
[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997.
[RFC2246]	Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, DOI 10.17487/RFC2246, January 1999.
[RFC3986]	Berners-Lee, T., Fielding, R. and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005.
[RFC5246]	Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008.
[RFC5322]	Resnick, P., "Internet Message Format", RFC 5322, DOI 10.17487/RFC5322, October 2008.
[RFC5646]	Phillips, A. and M. Davis, "Tags for Identifying Languages", BCP 47, RFC 5646, DOI 10.17487/RFC5646, September 2009.
[RFC5785]	Nottingham, M. and E. Hammer-Lahav, "Defining Well-Known Uniform Resource Identifiers (URIs)", RFC 5785, DOI 10.17487/RFC5785, April 2010.
[RFC6125]	Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, DOI 10.17487/RFC6125, March 2011.
[RFC6749]	Hardt, D., "The OAuth 2.0 Authorization Framework", RFC 6749, DOI 10.17487/RFC6749, October 2012.
[RFC6750]	Jones, M. and D. Hardt, "The OAuth 2.0 Authorization Framework: Bearer Token Usage", RFC 6750, DOI 10.17487/RFC6750, October 2012.
[RFC6819]	Lodderstedt, T., McGloin, M. and P. Hunt, "OAuth 2.0 Threat Model and Security Considerations", RFC 6819, DOI 10.17487/RFC6819, January 2013.
[RFC7009]	Lodderstedt, T., Dronia, S. and M. Scurtescu, "OAuth 2.0 Token Revocation", RFC 7009, DOI 10.17487/RFC7009, August 2013.
[RFC7033]	Jones, P., Salgueiro, G., Jones, M. and J. Smarr, "WebFinger", RFC 7033, DOI 10.17487/RFC7033, September 2013.
[RFC7515]	Jones, M., Bradley, J. and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015.
[RFC7516]	Jones, M. and J. Hildebrand, "JSON Web Encryption (JWE)", RFC 7516, DOI 10.17487/RFC7516, May 2015.
[RFC7517]	Jones, M., "JSON Web Key (JWK)", RFC 7517, DOI 10.17487/RFC7517, May 2015.
[RFC7518]	Jones, M., "JSON Web Algorithms (JWA)", RFC 7518, DOI 10.17487/RFC7518, May 2015.
[RFC7519]	Jones, M., Bradley, J. and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015.
[RFC7591]	**Jones, M., Bradley, J., Machulak, M. and hunt, P., "OAuth 2.0 Dynamic Client Registration Protocol", RFC 7591, July 2015.__**

[BCP195]	**Sheffer, Y., Holz, R. and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015.**
[RFC7636]	Sakimura, N., Bradley, J. and N. Agarwal, "Proof Key for Code Exchange by OAuth Public Clients", RFC 7636, DOI 10.17487/RFC7636, September 2015.
[RFC7662]	Richer, J., "OAuth 2.0 Token Introspection", RFC 7662, DOI 10.17487/RFC7662, October 2015.
[RFC7800]	Jones, M., Bradley, J. and H. Tschofenig, "Proof-of-Possession Key Semantics for JSON Web Tokens (JWTs)", RFC 7800, DOI 10.17487/RFC7800, April 2016.
[RFC8485]	Richer, J. and L. Johansson, "Vectors of Trust", RFC 8485, DOI 10.17487/RFC8485, October 2018.

Appendix A. Future updates

The following updates are foreseen to be made in future updates to this profile. These are not yet part of the profile.

Client Authentication using Mutual TLS

Instead of, or in addition to, the `private_key_jwt` client authentication method, mutual authenticated TLS MAY be used. This currently is specified in draft by IETF, see https://tools.ietf.org/html/draft-ietf-oauth-mtls-13. As result of this addition, the following changes SHOULD be anticipated:

- Addition of a `client_id` parameter in token Requests, instead of a `client_assertion` and `client_assertion_type` .
- Option for Clients to support `tls_client_auth` instead of or in addition to `private_key_jwt` .
- Requirement on OPs to support for `tls_client_auth` client authentication method next to `private_key_jwt` .