OpenID Connect for Identity Proofing

Publication Date: 09.02.2019

This specification defines an extension of OpenID Connect for providing Relying Parties with verified person data, typically used to identify a person according to a certain law.

Introduction

There are use cases where Relying Parties need to know the assurance level of the user claims attested by the OpenID Connect OP. Whereas assurance levels for authentication are a property of user account management and a certain OpenID Connect transaction, the assurance for user claims, i.e. the binding of a certain claim value to the person controlling the respective user account, might vary among the different user claims.

For example, the assurance an OP typically will be able to attest for an e-mail address will be “self-asserted”, “verified by opt-in”, or “verified by the respective e-mail provider via an attribute exchange protocol”. The family name of a user, in contrast, might have been validated in accordance with the respective Anti Money Laundering Law by showing an ID Card to a trained employee of the OP operator.

This specification aims at providing an extension to OpenID Connect to address the single use case of strong identification of a natural person in accordance with a certain laws. In Germany, for example, those would be the Anti Money Laundering Law, the Telecommunication Act and eIDAS.

Scope and Requirements

The extension specified in this document shall

- allow a OpenID Connect RP to request identity data and to express requirements regarding this data and the evidences and processes employed,
- allow an OP to attest identity data obtained and maintained in accordance with a certain law, and
- allow the RP to map the data obtained from the OP to other laws, e.g. it shall be possible to use identity data maintained in accordance with the Anti Money Laundering Law to fulfill requirements defined by eIDAS.

The scope of the extension is to define mechanisms to assert person data only. It SHALL be possible to use existing OpenID Connect claims and other extensions defined beside this extension within the same OpenID Connect transaction and especially to render the same ID token or other representations utilized to convey user claims. For example, OpenID Connect already defines additional claims to inform the RP of the verification status of the `phone_number` and `email` claims.

If necessary, this extension will introduce new user claims to cover user attributes currently uncovered by OpenID Connect, one examples is the `place_of_birth`.

The extension MUST allow the OP to attest verified and unverified claims in the same response.

This extension defines a way to assert verified claims only since self asserted claims, or better put claims without further information about the assurance, are already covered by OpenID Connect.

Even for asserting verified claims, this extension shall attempt to utilize existing OpenID Connect claims if possible and reasonable. It MUST, however, ensure developers cannot interpret none-verified claims as verified claims.

The representation defined in this extension SHALL allow RPs to obtain assertions about verified claims from different OPs using different trust frameworks.

Claims

In order to fulfill the requirements of some jurisdictions on identity proofing, this specification defines the following claims:

- `place_of_birth`: a structured claim representing the end-user's place of birth. It consists of the following fields:
    - country: ISO 3166-1 Alpha-2, e.g. DE, or ISO 3166-3
    - locality: city or other locality
- `nationality`: represents the user's nationality in format ISO 3166-1 Alpha-2, e.g. DE
- `birth_name`: family name someone has when he or she is born, or at least from the time he or she is a child. This term can be used by a man or a woman who changes a name later in life for any reason.

Verified Data Representation

This extension to OpenID Connect wants to ensure that RPs cannot mixup verified and unverified claims and incidentally process unverified claims as verified claims.

The representation proposed therefore provides the RP with the verified claims in a structure `verified_person_data` composed of the verification evidences and the corresponding claims, which were verified in the respective process.

The `verified_person_data` claim consists of the following sub-elements:

- `verification`: this element contains all data about the verification process
- `claims`: This element is a container for the verified user claims.

verification

The verification element consists of the following elements:

Fields	Type	Description
organization	String	who verified the identification document
legal_context	JSON Object	the legal/regulatory context governing the identification process of the organisation (see below)
date	String	the date when this organization verified the ID document as ISO 8601:2004 YYYY-MM-DD format
id	String	Unique reference to the verification process performed by the verifying organization. Used for backtracking in case of disputes or audits.
method	String	Method utilized for identity verification, possible values are ● "identity_document": verification of a physical document ● "eID": verification using an electronic ID Card Depending on the value of "method", there will be different additional sub-elements with the name of the method in the verification element Note: need to add further methods utilized outside of the EU

legal_context

The `legal_context` consists of the following fields:

- `country`: OPTIONAL ISO 3166-1 Alpha-2, e.g. "DE"
- `regulation`: designation of the act or set of acts or other directives, e.g. "Geldwäschegesetz"

Standardized values are:

- "DE", "Geldwäschegesetz"
- [TBD]

Note by the editor: It might be reasonable to rename this element to `trust_framework` to clearly indicate it can also be used to cover NIST SP 800-3A or eIDAS Identity Proofing including the respective assurance levels.

Method-specific elements

The following elements are contained in an `identity_document` sub-element. All elements not explicitly marked as optional are mandatory.

Fields	Type	Description
country	String	OPTIONAL. country where the document was issued: ISO 3166-1 Alpha-2, e.g. DE
type	String	Type of the id document, standardized values are: ● “ID Card”: national ID Card ● “Passport”: Passport It is also possible to set type to country-specific types (free text). RPs will either just store this value for audit purposes or apply bespoken business logic to it.
issuer	String	Issuer of the identity document
number	String	Number of the identity document
date_of_issuance	String	CONDITIONALLY REQUIRED. The date the document was issued as ISO 8601:2004 YYYY-MM-DD format, if this attribute exists for the particular type of document.
date_of_expiry	String	CONDITIONALLY REQUIRED. The date the document will expire as ISO 8601:2004 YYYY-MM-DD format, if this attribute exists for the particular type of document.
method	String	The method used to verify the document, standardized values are: ● “Physical In-Person Proofing (bank)” ● “Physical In-Person Proofing (shop)” ● “Physical In-Person Proofing (courier)” ● “Supervised remote In-Person Proofing” Note: if the IDP cannot determine the method used to verify the identity document it MUST omit this field.
organization	String	CONDITIONALLY REQUIRED. Organization which performed the verification. Can be omitted if this is the same organisation as in the enclosing `verification`

		element.
agent	String	OPTIONAL. Agent (person) who conducted the verification. The agent may be identified by Name or an identifier which can be resolved into the agent's name during an audit.

The following elements are contained in an `eID` sub-element:

Fields	Type	Description
country	String	country where the eID was issued: ISO 3166-1 Alpha-2, e.g. DE
type	String	type of the eID standardized value is: ● "ID Card": national ID Card It is also possible to set type to country-specific designations
identifier	String	person identifier obtained by the verifying organization from the eID system

claims

The `claims` element contains the user claims whose binding to the actual user account was verified by the `verification/organization` and is attested by the OP. This element may contain at most the claims listed in the following.

This specification uses the following existing OpenID Connect user claims as defined in [https://openid.net/specs/openid-connect-core-1_0.html#StandardClaims]:

- `name`
- `given_name`
- `middle_name`
- `family_name`
- `birthdate`
- `address`

Additionally, the following claims defined in this document can be used:

- `place_of_birth`:

- `nationality`
- `birth_name`

Examples

The following shows an example of `verified_person_data` objects for all three verification methods.

identity_document


```
{
   "verified_person_data":{
      "verification":{
         "organization":"Bank Y",
         "legal_context":{
            "country":"DE",
            "regulation":"Geldwäschegesetz"
         }
         "date":"2013-02-21",
         "id":"676q3636461467647q8498785747q487",
         "method":"identity_document",
         "identity_document":{
            "country":"DE",
            "type":"ID Card",
            "issuer":"Stadt Augsburg",
            "number":"53554554",
            "date_of_issuance":"2012-04-23",
            "date_of_expiry":"2022-04-22",
            "method":"Physical In-Person Proofing (shop)",
            "organization":"Deutsche Post AG",
            "agent":"Steffen Schuster"
         }
      },
      "claims":{
         "given_name":"Max",
         "family_name":"Meier",
         "birthdate":"1956-01-28",
         "place_of_birth":{
            "country":"DE",
            "city":"Musterstadt"
```

```
        },
        "nationality":"DE",
        "address":{
           "locality":"Maxstadt",
           "postal_code":"12344",
           "country":"DE",
           "street":"An der Sanddüne 22"
        }
     }
  }
}
```


eID


```
{
   "verified_person_data":{
      "verification":{
         "organization":"Bank Y",
         "legal_context":{
            "country":"DE",
            "regulation":"Geldwäschegesetz"
         }
         "date":"2013-02-21",
         "id":"676q3636461467647q8498785747q487",
         "method":"eID",
         "eID":{
            "country":"DE",
            "type":"ID Card",
            "identifier":"???"
         }
      },
      "claims":{
         "given_name":"Max",
         "family_name":"Meier",
         "birthdate":"1956-01-28",
         "place_of_birth":{
            "country":"DE",
            "city":"Musterstadt"
         },
```

```
            "nationality":"DE",
            "address":{
               "locality":"Maxstadt",
               "postal_code":"12344",
               "country":"DE",
               "street":"An der Sanddüne 22"
            }
         }
      }
}
```

Requesting verified person data

The way verified person data can be requested (on a claim level) follows the convention introduced by the OpenID Connect specification. Basically, the claim `verified_person_data` is added to the respective section of the claims parameter.

Since `verified_person_data` contains the effective user claims in a nested claims element, this sub element is also used to denote the desired user claims as keys with a null value. An example is shown in the following.

```
{
   "userinfo":{
      "verified_person_data":{
         "claims":{
            "given_name":null,
            "family_name":null,
            "birthdate":null,
            "place_of_birth":null,
            "nationality":null,
            "address":null
         }
      }
   }
}
```

Use of the claims parameters allows the RP to exactly select the user claims needed for their use case and fulfil the requirement for data minimization.

Note: A `claims` sub-element with value `null` is interpreted as a request for all possible claims. An example is shown in the following:

```
{
   "userinfo":{
      "verified_person_data":{
         "claims":null
      }
   }
}
```

Note: If the `claims` sub-element is empty or contains a claim other than the claims listed in claims, the OP will abort the transaction with an `invalid_request` error.

Note: The `claims` sub-element can be omitted, which is equivalent to a `claims` element whose value is `null`.

The RP may also express a requirement regarding the age of the verification data, i.e., the time elapsed since the verification process asserted in the `verification` element has taken place. This requirement is expressed by defining `max_age` field for the sub-element `date` of the `verification` sub-claim that defines the maximum time elapsed since the verification process took place in seconds. The following is an example claims parameter:

```
{
   "userinfo":{
      "verified_person_data":{
         "verification":{
            "date":{
               "max_age":"63113852"
            }
         },
         "claims":null
      }
   }
}
```

The IDP is supposed to try its best to fulfill this requirement. If the verification data of the user is older than the requested `max_age`, it SHOULD attempt to refresh the user's verification by sending her through a video id process or utilize other digital identity means (e.g. electronic ID card).

If the IDP is unable to fulfill the requirement, there are two possible outcomes of the transaction:

- If the RP did not request `date` as essential claim, the IDP will provide the RP with the data available and the RP may decide how to use the data.
- If the RP requested `date` as essential claim, the IDP will abort the transaction and respond with the error code `unable_to_meet_requirement`.

The following is an example of a claims parameter requesting `date` as essential claim.

```
{
   "userinfo":{
      "verified_person_data":{
         "verification":{
            "date":{
               "max_age":"63113852",
               "essential":true
            }
         }
      }
   }
}
```

transaction_id

The OP MUST maintain an audit trail of any transaction which leads to the attestation of verified person data and it MUST provide the RP with a reference to this transaction for potential dispute resolution if requested. The claim `transaction_id` is used for this purpose. The RP requests this claim like any other claim. `transaction_id` is of type string. It MUST be unique within the context of the respective IDP, uniquely identify a certain transaction at the IDP, and allow an RP to obtain transaction details. Those details consist of

- the transaction id
- the authentication methods employed
- the transaction type (e.g. scope values)

The OP MUST store the transaction data as long as it is required to store transaction data for auditing purposes by its respective regulation. The RP might use for dispute resolution and in the context of an audit at the RP.

OP Capabilities

The OP advertises its capabilities with respect to verified person data in the standard `openid-configuration` using the following elements.

`verified_person_data_supported` JSON array containing all claims supported within `verified_person_data`.

`verification_methods_supported` This element is a JSON array containing at the top level the verification methods as defined in section verification. `qes` and `eID` are simple JSON strings whereas `identity_document` is contained in a JSON object and is itself an array containing all identity document proofing methods employed by the respective organisation.

This is an example:

```
{
...
   "verified_person_data_supported":[
      "given_name",
      "family_name",
      "birthdate",
      "place_of_birth",
      "nationality",
      "address"
   ],
   "verification_methods_supported":[
      {
         "identity_document":[
            "Physical In-Person Proofing (bank)",
            "Physical In-Person Proofing (shop)",
            "Physical In-Person Proofing (courier)",
            "Supervised remote In-Person Proofing"
         ]
      },
      "eID"
   ]
...
}
```

Change Log

- 03.02.2019
    - Added distinct section for defining the additional user claims
    - Reworked introduction and added new section on scope, and requirements
    - Removed QES method
- 09.02.2019
    - Added `birth_name` to claims section

Acknowledgement

The following people contributed to the concept described in this document:

- Sven Manz, Waldemar Zimpfer, Willi Wiedergold, Fabian Hoffmann, Daniel Keijsers, Ralf Wagner, Sebastian Ebling, Daniel Fett, Peter Eisenhofer (yes.com)

Authors' Addresses

Dr. Torsten Lodderstedt
yes.com AG
Email: torsten@lodderstedt.net
URI: http://www.yes.com/