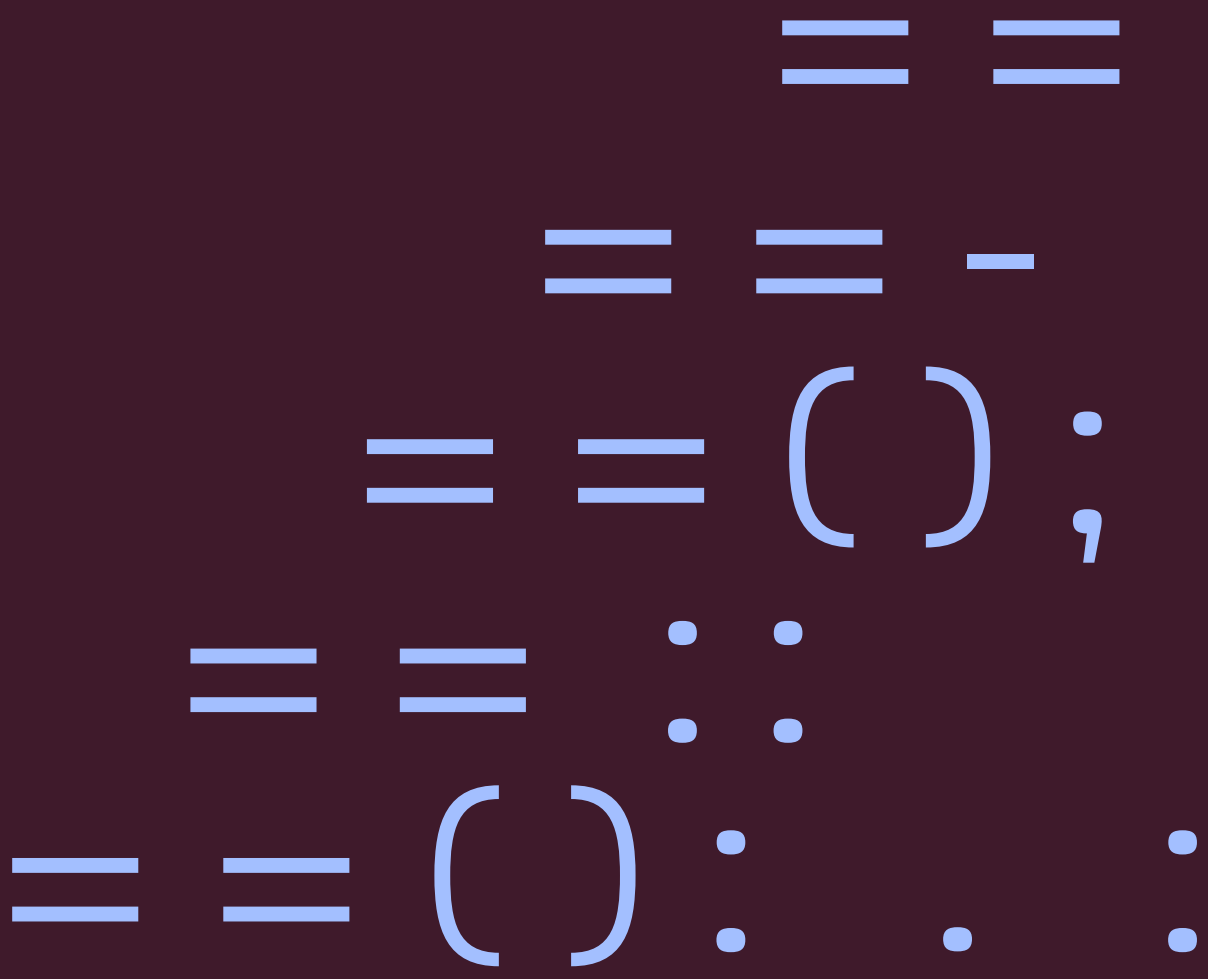
Agency for Digital Government

OIO JWT Token Profile

Version 1.0
Draft 5



18.05.2026

1. Introduction

1.1 Preface

This profile is part of a larger set of specifications aimed at supporting mobile applications (apps), agents, web clients and back-end clients based on OpenID Connect, JWT and related technologies. The specifications are written with the NemLog-in public broker in mind but can freely be used elsewhere.

The profile enables infrastructure components based on OpenID Connect to support clients with token-based access to APIs offered by public and private service providers.

This profile has a close relationship with the [OIO OIDC] profile 1.0, OIOSAML 4.0 and OIO WS-Trust 1.2 profiles, and several elements from these specifications are re-used:

- Several attributes (claims) defined in OIOSAML 4.0 are used to ensure that OIO SAML and OIO JWT tokens are as similar as possible.
- Identifier types for users (UUID) and service providers (Entity IDs) are reused.
- The model for access rights and delegations as specified in OIO Basic Privilege Profile is re-used – but expressed using JSON syntax.

The goal is to facilitate a smooth transition from SAML and WS-Trust to OpenID Connect, OAuth 2.0 and JWT/JWS.

Note: readers of this document are expected to be familiar with the above-mentioned standards.

1.2 Usage Scenarios

This profile is intended for use within Danish public sector federations where information about authenticated identities is federated across service providers. The goal is to achieve standardization, interoperability, security and privacy, while enabling re-use of common implementations.

The profile is written for use with [OIO OIDC] and specifies requirements for three types of JWT tokens:

- **ID Tokens** describing an authenticated end-user including context of the authentication such as the achieved level of assurance. These tokens are issued to the client by the OIDC Token Endpoint. The requirements for these are a sub-profile of ID Tokens described in OpenID Connect Core [OIDC] (chapter 2). These correspond to OIO SAML Assertions issued by an Identity Provider when a user logs in but do not necessarily contain user attributes.
- **User Tokens describing** attributes about a user which may include privileges but are not necessarily tied to a specific authentication event. These tokens are issued to the client by the OIDC UserInfo Endpoint with some similarities to the SAML Attribute Query protocol.
- **Delegated Access Tokens** provide client access to external APIs on behalf of an end-user. These correspond to OIO IDWS tokens issued by a Security Token Service in the existing infrastructure.

The [OIO OIDC] profile further describes two other token types which are not the primary scope for this profile since they are used internally with the Authorization Server:

- Access Tokens for the endpoints in the Authorization Server
- Refresh Tokens for the above Access Tokens

Both of these MAY be opaque and if so SHOULD comply with [RFC9700] and thus contain sufficient entropy (>=128 bit) and be constructed from a cryptographically strong random or pseudo-random number sequence.

The table below summarizes the JWT tokens in scope of this profile:

Claim	Content	Usage and format
ID Token	Authentication information User attributes (optional)	JWT consumed by client and issued by OIDC Token endpoint
User Token	User attributes	JWT consumed by client and issued by OIDC UserInfo endpoint
Delegated Access Token	Representation of client authorization to external API	JWT issued by OIDC Token endpoint for External API

2. Notation and terminology

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 BCP 14 [RFC2119][RFC8174] when, and only when, they appear in all capitals, as shown here.

This specification uses the following typographical conventions in text: `<ns:Element>, Attribute`, **Datatype**, `OtherCode`. The normative requirements of this specification are individually labeled with a unique identifier in the following form: **[JTP-EXAMPLE-01]**. All information within these requirements should be considered normative unless it is set in *italic* type. Italicized text is non-normative and is intended to provide additional information that may be helpful in implementing the normative requirements.

2.1 Terminology

This specification describes flows involving the following actors:

- **Client** – a native app, agent or browser app acting in the role of client in OAuth 2.0 and OpenID Connect sense. It provides application services to the end-user, requests access tokens and consumes one or more external (REST) APIs e.g. for retrieving or updating data about the end-user. Clients are considered 'confidential' when the client application can securely store credentials (such as a client secret or private key) because it runs in a trusted environment.
- **SP API** – Service Provider API. An API offered by a Service Provider which is protected by a trusted Authorization Server – i.e. all API access requires a signed token. The API Service Provider can be the same or a different organization providing the client.
- **End-user** – a person who authorizes client access regarding defined scopes, and in case the client as a native app installs the app on a personal mobile device.
- **Authorization Server** – a central OAuth 2.0 infrastructure component
- **Token Server** – an OIDC/OAuth 2.0 infrastructure component that issues tokens which provide access to external APIs.

3. General JWT token requirements

[JTP-01]

All JWT tokens MUST comply with the core [JWT] specification.

3.1 Required claims for all JWT tokens

[JTP-02]

Issued JWT tokens MUST include all the claims listed below with non-empty values as specified:

Claim	Value
iss	Identifier for the OIDC issuer as an HTTP URI. Example: 'https://nemlog-in.dk'
aud	Audience for the token as an EntityID following OIOSAML 4.0 [OIO-IDP-18]. Example: 'https://digitalpost.dk'. The aud claim be aither be a string or an array strings. For ID Tokens, aud must identify the client; for Delegated Access Tokens, aud should identify the external API.
exp	Expiration time. JSON number with 5 minutes clock skew tolerance.
iat	Time at which the token was issued as a JSON number.

3.2 Signature and validation requirements

[JTP-03]

JWT tokens MUST be signed using [JWS] with one of the following algorithms from [JWA]:

- PS256, PS384, PS512 (RSA)
- ES256, ES384, ES512 (ECDSA)

Note: compliance frameworks such as [NSIS] may pose additional requirements for protection of the private key using cryptographic hardware modules.

[JTP-04]

Token signatures MUST be verified against a pinned certificate provided as part of the secure configuration (e.g. token signing certificate). Tokens with invalid signatures or algorithms MUST be rejected. Revocation checks of pinned token signing certificates are not required, but emergency rollover procedures SHOULD be in place.

[JTP-05]

A key ID (kid) header SHOULD be used to indicate the version of signing key in order to support key-rollover schemes.

[JTP-06]

The following JWS header fields for tokens issued under this profile[1] MUST NOT be used: x5u, x5c, jku, or jwk.

[1] DPoP proof JWTs are different.

4. ID token requirements

ID tokens are consumed by client applications to establish a local session. This chapter describes ID-token specific requirements in addition to the generic token requirements described in chapter 3.

[JTP-07]

ID Tokens MUST comply with requirements for ID tokens in [OIDC] Core (chapter 2) and additional requirements stated below.

[JTP-08]

ID Tokens MUST contain the following claims:

Claim	Value
sub	Subject identifier containing a persistent and service-provider specific UUID in the format specified in OIOSAML 4[2] requirement [OIO-IDP-15]. Example: 'https://data.gov.dk/model/core/eid/**person**/uuid/123e4567-e89b-12d3-a456-426655440000'
nonce	Client nonce received in the request which is relayed back to allow detection of man-in-the-middle and replay attacks. Note that requirements for entropy is defined in the OIO OIDC Profile.
at_hash	Hash value of the Access Token issued together with the ID token, as a cryptographic proof that both tokens belong to the same authentication response. See [OIO OIDC] for details.
auth_time	Time when the end-user authentication occurred as a JSON number.

[JTP-09]

ID tokens MUST contain claims describing the level of assurance (LoA) regarding the user authentication. The LoA claims are defined in chapter 5.

ID tokens MAY contain claims about user attributes as defined in chapter 5.

[2] Note that an example is currently missing in OIOSAML 4 for legalpersons. Here the prefix …/legalperson/uuid can be used.

5. Attribute profile requirements

This chapter specifies attributes for User tokens and optionally ID tokens (if they contain user attributes).

The desired attribute profile(s) and claims can be requested by the client as scope values in the authorization request, see [OIO OIDC].

[JTP-10]

Tokens with user attributes MUST contain an attribute_profile claim with a declaration of the attribute profile (type of identity) the token was issued according to. The allowed set of claim values are:

- person_dk
- person_dk_withoutcpr
- person_dk_anonymous
- professional_dk
- professional_dk_anonymous
- person_eu
- professional_eu
- legalperson_eu

[JTP-11]

Tokens with user attributes MUST contain claims defined by the attribute profile (found in the attribute_profile claim) as specified in the tables below. The markup in the tables should be interpreted as follows:

- 'M' – the claim is Mandatory and must always be present in all tokens issued according to the attribute profile. To enable data minimization, the set of mandatory attributes is kept small.
- 'S' – the claim must be supported by token issuers offering the attribute profile and included in a token if requested by the client.
- 'O' – the attribute is Optional for token issuer to support, and it may be present in issued tokens. A token issuer may choose to include optional claims according to agreement with the client.
- Blank – the attribute cannot be expected in token of the specified type.

Note: In the tables below, the claim name is written in bold along with the corresponding attribute name from OIOSAML 4 in square brackets. Using the OIOSAML attribute name, the definition/content of the claim/attribute value can be looked up in the OIOSAML 4 specification.

Note: Claims regarding level of assurance are mandatory in ID tokens but optional in User Tokens.

Common attributes:

Claim	DK person	DK Person without CPR	*DK person (anonymized)*	*DK professional*	*DK professional (anonymized)*	*eIDAS natural person*	*eIDAS legal person*	*eIDAS professional*
spec_ver *) [https://data.gov.dk/model/core/specVersion]	M	M	M	M	M	M	M	M
priv **) [https://data.gov.dk/model/core/eid/privilegesIntermediate]	O	O	O	S	S			
nsis_loa [https://data.gov.dk/concept/core/nsis/loa]	M	M	M	M	M			
nsis_ial [https://data.gov.dk/concept/core/nsis/ial]	O	O	O	O	O			
nsis_aal [https://data.gov.dk/concept/core/nsis/aal]	O	O	O	O	O			
full_name [https://data.gov.dk/model/core/eid/fullName]	S	S		S				

Claim	DK person	DK Person without CPR	*DK person (anonymized)*	*DK professional*	*DK professional (anonymized)*	*eIDAS natural person*	*eIDAS legal person*	*eIDAS professional*
given_name [https://data.gov.dk/model/core/eid/firstName]	S	O		S				
family_name [https://data.gov.dk/model/core/eid/lastName]	S	O		S				
alias [https://data.gov.dk/model/core/eid/alias]			M		M			
email [https://data.gov.dk/model/core/eid/email]	O	O		O				
cpr [https://data.gov.dk/model/core/eid/cprNumber]	S		O	O		O		O
age [https://data.gov.dk/model/core/eid/age]	S	S	O	O				
cpr_uuid [https://data.gov.dk/model/core/eid/cprUuid]	S		O	O		O		O
date_of_birth [https://data.gov.dk/model/core/eid/dateOfBirth]	S	S	O	O				
pid [https://data.gov.dk/model/core/eid/person/pid]	O		O					
persistent_id [https://data.gov.dk/model/core/eid/professional/uuid/persistent]				S	S			
rid [https://data.gov.dk/model/core/eid/professional/rid]				O	O			
cvr [https://data.gov.dk/model/core/eid/professional/cvr]				M	M			
org_name [https://data.gov.dk/model/core/eid/professional/orgName]				M	M			
p_number [https://data.gov.dk/model/core/eid/professional/productionUnit]				O	O			
se_number [https://data.gov.dk/model/core/eid/professional/seNumber]				O	O			
auth_to_repr [https://data.gov.dk/model/core/eid/professional/authorizedToRepresent]				O				

*) The value of the ‘spec_ver’ claim should be ‘1.0’ for this specification.

**) The encoding of the 'priv' claim is described in chapter 6 with a slightly different syntax than OIOSAML 4 (but equivalent semantics).

Table 2.1: New attributes introduced in OIOSAML 4

Claim	DK person	DK Person without CPR	DK person (anonymized)	DK professional	DK professional (anonymized)	eIDAS natural person	eIDAS legal person	eIDAS professional
attribute_profile [https://data.gov.dk/concept/core/eid/profile]	M	M	M	M	M	M	M	M
id_provider [https://data.gov.dk/concept/core/eid/provider]	O	O	O	O	O	O	O	O
acr ***) [https://data.gov.dk/model/core/loa]	M	M	M	M	M	M	M	M
eidas_loa [https://data.gov.dk/model/core/eidas/loa]	O	O				M	M	M
eidas_member_state [https://data.gov.dk/model/core/eid/eidas/memberState]	O	O				S	S	S
cpr_ial [https://data.gov.dk/model/core/nsis/cpr_ial]	O					O	O	O
is_robot [https://data.gov.dk/model/core/eid/professional/isRobot]				O	O			
allow_qualified_signing [https://data.gov.dk/model/core/eid/allowQualifiedSigning]	O	O	O	O	O	O	O	O

***) The 'acr' claim now specifies the generic loa level whose value will correspond to the eIDAS or NSIS level of assurance for the current authentication. In version 0.91 of the profile, the claim contained the NSIS loa which is now specified in the nsis_loa claim.

Table 2.3: eIDAS natural person

Claim	DK person	DK Person without CPR	*DK person (anonymized)*	*DK professional*	*DK professional (anonymized)*	*eIDAS natural person*	*eIDAS legal person*	*eIDAS professional*
eidas_npi [http://eidas.europa.eu/attributes/naturalperson/PersonIdentifier]						M		M*)
eidas_family_name [http://eidas.europa.eu/attributes/naturalperson/CurrentFamilyName]						M		M*)
eidas_given_name [http://eidas.europa.eu/attributes/naturalperson/CurrentGivenName]						M		M*)
eidas_date_of_birth [http://eidas.europa.eu/attributes/naturalperson/DateOfBirth]						M		M*)
eidas_birth_name [http://eidas.europa.eu/attributes/naturalperson/BirthName]						O		O*)
eidas_place_of_birth [http://eidas.europa.eu/attributes/naturalperson/PlaceOfBirth]						O		O*)
eidas_address [http://eidas.europa.eu/attributes/naturalperson/CurrentAddress]						O		O*)
eidas_gender [http://eidas.europa.eu/attributes/naturalperson/Gender]						O		O*)
eidas_nationality [http://eidas.europa.eu/attributes/naturalperson/Nationality]						O		O*)
eidas_country_of_birth [http://eidas.europa.eu/attributes/naturalperson/CountryOfBirth]						O		O*)
eidas_town_of_birth [http://eidas.europa.eu/attributes/naturalperson/TownOfBirth]						O		O*)
eidas_country_of_residence [http://eidas.europa.eu/attributes/naturalperson/CountryOfResidence]						O		O*)
eidas_phone_number [http://eidas.europa.eu/attributes/naturalperson/PhoneNumber]						O		O*)
eidas_email_address [http://eidas.europa.eu/attributes/naturalperson/EmailAddress]						O		O*)
eidas_power_of_repr_scope [http://data.europa.eu/p4s/attributes/PowerOfRepresentationScope]								O

Table 2.4: eIDAS legal person

Attribute ID	DK person	DK Person without CPR	*DK person (anonymized)*	*DK professional*	*DK professional (anonymized)*	*eIDAS natural person*	*eIDAS legal person*	*eIDAS professional*
eidas_lpi [http://eidas.europa.eu/attributes/legalperson/LegalPersonIdentifier]							M	M
eidas_legal_name [http://eidas.europa.eu/attributes/legalperson/LegalName]							M	M
eidas_legal_address [http://eidas.europa.eu/attributes/legalperson/LegalPersonAddress]							O	O
eidas_vat [http://eidas.europa.eu/attributes/legalperson/VATRegistrationNumber]							O	O
eidas_tax [http://eidas.europa.eu/attributes/legalperson/TaxReference]							O	O
eidas_d2012_17 [http://eidas.europa.eu/attributes/legalperson/D-2012-17-EUIdentifier]							O	O
eidas_lei [http://eidas.europa.eu/attributes/legalperson/LEI]							O	O
eidas_eori [http://eidas.europa.eu/attributes/legalperson/EORI]							O	O
eidas_seed [http://eidas.europa.eu/attributes/legalperson/SEED]							O	O
eidas_sic [http://eidas.europa.eu/attributes/legalperson/SIC]							O	O
eidas_legal_phone_number [http://eidas.europa.eu/attributes/legalperson/LegalPhoneNumber]							O	O
eidas_legal_email [http://eidas.europa.eu/attributes/legalperson/LegalEmailAddress]							O	O

Note:

- The ‘eIDAS professional’ profile corresponds in eIDAS terms to a ‘natural person representing a legal person’.
- The ‘sub’ claim provides an additional, unique identifier (not part of the attribute profiles). This identifier is generally more privacy-friendly than most of the identifying attributes in the table above.
- Token issuers may include additional attributes if required by the client. In this case, impact on interoperability and privacy should be considered.

Note regarding the eIDAS professional profile *)

The ‘eIDAS professional’ attribute profile is modelled as ‘a natural person representing a legal’ person in the eIDAS SAML Attribute Profile specification [ESAML-AP]. This implies two things: firstly, the professional attribute profile contains attributes of both an eIDAS natural person (the representative) and an eIDAS legal person (the represented).

Secondly, the natural person attribute names (under eIDAS namespace) have the prefix ‘eidas_representative’ instead of just ‘eidas’. Thus, the attribute

- `eidas_npi`

becomes

- `eidas_representative_npi`

6. Privileges in JSON encoding

This section describes how to encode a set of assigned privileges defined in OIO Basic Privilege Profile [OIO-BPP] as a JSON structure with exactly the same semantics. Thus, all names of privileges, scopes and constraints are URIs and values are simple text strings.[3]

The intermediate version of [OIO-BPP] uses a structure like the one below (with white spaces inserted for readability):

```
<?xml version="1.0" encoding="UTF-8"?>
<bpp:PrivilegeList
   xmlns:bpp="http://digst.dk/oiosaml/basic_privilege_profile"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
   <PrivilegeGroup Scope="urn:dk:gov:saml:cvrNumberIdentifier:12345678">
       <Constraint Name="http://sts.kombit.dk/constraints/KLE/1">25.*</Constraint>
       <Constraint Name="http://sts.kombit.dk/constraints/foelsomhed/1">
                 31c09910-e011-46a5-86fb-254374421fe8
       </Constraint>
       <Privilege>
         http://serviceplatformen.prod-serviceplatformen.dk/roles/servicesystemrole/dummy/1
       </Privilege>
   </PrivilegeGroup>
</bpp:PrivilegeList>
```

The corresponding JSON structure for the `priv` claim is formatted as shown below (which should not be base64 encoded when included in a JWT):

```
{
    "privilegegroups" : [
       {
           "privilege" : ["http://serviceplatformen.prod-serviceplatformen.dk/roles/servicesystemrole/dummy/1"],
           "scope" : "urn:dk:gov:saml:cvrNumberIdentifier:12345678",
           "constraints" : [
                 {
                    "name" : "http://sts.kombit.dk/constraints/KLE/1",
                     "value" : "25.*"
                 },
                 {
                     "name" : "http://sts.kombit.dk/constraints/foelsomhed/1",
                     "value" : "31c09910-e011-46a5-86fb-254374421fe8"
                 }
          ]
       }
    ]
}
```

[3] A more formal syntax definition will be given at a later stage.

7. Delegated Access Token requirements

This chapter defines requirements for Delegated Access Tokens issued for a client accessing an external API on behalf of an end-user. These tokens are issued using OAuth 2.0 Token Exchange in accordance with [OIO OIDC].

[JTP-12]

Delegated Access Tokens MUST comply with generic requirements for JWT tokens found in chapter 3 and thereby include `iss, aud, exp` and `iat` claims.

The `aud` claim MUST contain the EntityID of the external API.

[JTP-13]

Delegated Access Tokens MUST contain a `sub` claim representing the end-user (subject) who granted the client access on their behalf.

Delegated Access Tokens SHOULD contain claims describing the level of assurance of the end user authentication (e.g. `generic_loa` or `nsis_loa`).

Delegated Access Tokens MUST contain an `act` claim (actor) with an embedded `sub` containing the ID of the client acting on behalf of the end user (`act.sub`).

Delegated Access Tokens MUST contain `priv` claims specifying the scope of access granted to the client by the end user.

7.1 Sender-constrained tokens

Sender-constrained tokens provide additional security against attacks where a stolen token is presented by an illegitimate client, since usage of a sender-constrained token requires proof of possession of a private key corresponding public JWK. See [DPoP] and [OIO OIDC] for additional details.

[JTP-14]

Delegated Access tokens for confidential clients SHOULD be issued as sender-constrained tokens by including a `cnf` confirmation claim containing the SHA-256 JWK thumbprint via a `jkt` element.

Example:

```
{
  "cnf": {
      "jkt": "base64url-thumbprint-of-client-public-jwk"
  }
}
```

8. Example (not normative)

8.1 ID token issued to person

Below is an example of an ID token issued to a person:


```
{
  "iss": "https://nemlog-in.dk",
  "sub": "https://data.gov.dk/model/core/eid/person/uuid/123e4567-e89b-12d3-66554400...",
  "aud": "https://digitalpost.dk/postapi",
  "exp": 3317281970,
  "iat": 1311280970,
  "auth_time": 1311280969,
  "nonce": "n-0S6_WzA2Mj",
  "nsis_loa": "https://data.gov.dk/concept/core/nsis/loa/Substantial",
  "acr": "https://data.gov.dk/concept/core/loa/Substantial",
  "spec_ver": "1.0",
  "given_name": "Hans",
  "family_name": "Jensen",
  "cpr": "2611779999",
  "attribute_profile": "person_dk"
}
```

8.2 ID token issued to a professional

Below is an example of claims section for an ID token issued for a professional:


```
{
  "iss": "https://nemlog-in.dk",
  "sub": "https://data.gov.dk/model/core/eid/professional/uuid/987e4567-...",
  "aud": "https://serviceplatformen.dk/",
  "exp": 1317281970,
  "iat": 1311280970,
  "auth_time": 1311280969,
  "nonce": "n-0S6_WzA2Mj",
  "nsis_loa": "https://data.gov.dk/concept/core/nsis/loa/Substantial",
  "acr": "https://data.gov.dk/concept/core/loa/Substantial",
  "spec_ver": "1.0",
  "attribute_profile": "professional_dk",
  "priv": {
    "privilegegroups" : [
      {
        "privileges" : ["http://serviceplatformen.dk/roles/servicesystemrole/dummy/1"],
        "scope" : "urn:dk:gov:saml:cvrNumberIdentifier:12345678",
        "constraints" : [
          {
            "name" : "http://sts.kombit.dk/constraints/KLE/1",
            "value" : "25.*"
          },
          {
            "name" : "http://sts.kombit.dk/constraints/foelsomhed/1",
            "value" : "31c09910-e011-46a5-86fb-254374421fe8"
          }
        ]
      }
    ]
  }
}
```

9. Other considerations (not normative)

9.1 Encryption

Since tokens are not transported via the user agent and are always sent over TLS 1.2 (or higher) with pinned server certificates, application-level encryption of tokens is not deemed necessary. Deployments are free to add it however if deemed necessary in special scenarios.

9.2 Token validity period

The maximum validity period of tokens (including refresh tokens) is defined in the [OIO OIDC] profile based on client and token type.

10. References

- [JWA] Jones, M., “JSON Web Algorithms (JWA), IETF Proposed Standard", RFC7518, May 2015. https://datatracker.ietf.org/doc/html/rfc7518
- [JWE] Jones, M., and J. Hildebrand, “JSON Web Encryption (JWE), IETF Pro-posed Standard” https://tools.ietf.org/html/rfc7516
- [JWK] Jones, M., “JSON Web Key (JWK),” IETF Proposed Standard, https://tools.ietf.org/html/rfc7517
- [JWS] Jones, M., Bradley, J., and N. Sakimura, “JSON Web Signature (JWS),” IETF Proposed Standard, https://tools.ietf.org/html/rfc7515
- [JWT] Jones, M., Bradley, J., and N. Sakimura, “JSON Web Token (JWT),” IETF Proposed Standard, https://tools.ietf.org/html/rfc7519.
- [NSIS] ”National Standard for Identiteters Sikringsniveauer 2.1.0”. https://digst.dk/it-loesninger/nemlog-in/det-kommende-nemlog-in/vejledninger-og-standarder/nsis-standarden/
- [OIOSAML] ”OIOSAML Web SSO Profile 4”. https://digst.dk/it-loesninger/standarder/oiosaml-profiler/
- [OIO-BPP] ”OIO Basic Privilege Profile 1.2”. https://digst.dk/it-loesninger/standarder/oiosaml-profiler/)
- [OIO OIDC] "OIO OIDC Profile V1.0", Danish Agency for Digitisation. https://digst.dk/it-loesninger/standarder/openid-connect-profiler/
- [OIDC] “OpenID Connect Core 1.0 incorporating errata set 2”, https://openid.net/specs/openid-connect-core-1_0.html
- [RFC6819] “OAuth 2.0 Threat Model and Security Considerations”, IETF. https://tools.ietf.org/html/rfc6819
- [DPoP] “RFC 9449 OAuth 2.0 Demonstrating Proof of Possession (DPoP)”, https://www.rfc-editor.org/rfc/rfc9449.html

Agency for Digital Government