DIGITALISERINGSSTYRELSEN

OIO JWT Token Profile
Version 0.91

Status: Draft
Date: 06.09.2021

1 Introduction

1.1 Preface

This profile is part of a larger set of specifications aimed at supporting mobile applications (apps) and web clients based on OpenID Connect, JWT and related technologies. The specifications are written with NemLog-in3 in mind but can freely be used elsewhere, and they are exclusively focused on app-scenarios.

The Danish Agency for Digitisations plans to establish infrastructure components based on OpenID Connect to support native apps and web clients with token-based access to REST APIs offered by public and private service providers. The future NemLog-in3 components will include an authorization server, a token service and web portals for registration and management of apps and APIs. The infrastructure will ensure user authentication based on [NSIS] and subsequent authorization of apps with user consent by issuing and managing security tokens – corresponding to the current infrastructure for web applications and SOAP-services based on SAML and WS-Trust.

This profile has a close relationship with the OIOSAML 3.0 and OIO WS-Trust 1.1 profiles, and several elements from these specifications are re-used[1]:

- Several attributes (claims) defined in OIOSAML 3.0 are used to ensure that SAML and JWT tokens are as similar as possible.
- Identifier types for users (UUID) and service providers (Entity IDs) are re-used.
- The model for access rights and delegations as specified in OIO Basic Privilege Profile are re-used – but expressed using JSON syntax.

The goal is to ensure the same user experience in apps and web applications, and that back-end infrastructure can easily be re-used for the two scenarios.

Note: readers of this document are expected to be familiar with OAuth, JWT and OpenID Connect.

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 targeted for use with the OpenID Connect protocol (also currently being profiled) for use with native apps as well as web clients.

[1] A merge of specifications can be considered at a later stage, but for now they are kept separate as they have different status (draft vs final specification).

The profile specifies requirements for two types JWT tokens issued by NemLog-in infrastructure in app scenarios:

- ID tokens describing an authenticated end-user (OIDC).
- Access tokens providing an app to access a REST API on behalf of an end-user.

The NemLog-in OIDC flow uses two other tokens:

- Access tokens for the NemLog-in Token Server endpoint
- Refresh tokens for the above access tokens

Both are opaque and thus not meaningful outside NemLog-in and not detailed in this specification. They should however comply with [RFC6819] and thus contain entropy >=128 bits and be constructed from a cryptographically strong random or pseudo-random number sequence.

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: **[OIO-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 or browser app acting in the role of client in OAuth 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.
- **SP API** – Service Provider API. An API offered by a Service Provider which is protected by a trusted Authorization- and Token Server – i.e. all API access requires a signed token from these. The API Service Provider can be the same or a different organization providing the client.
- **End-user** – a person authorizing client access on his/her behalf regarding defined scopes, and in case the client as a native app installs the app on his/her personal mobile device.
- **Authorization Server** – a central OAuth 2.0 infrastructure component (in the future delivered by NemLog-in).
- **Token Server** – OIDC/OAuth 2.0 infrastructure component (in the future delivered by NemLog-in) that issues tokens which provide access to external APIs.

3 JWT token requirements

This chapter contains requirements for ID tokens. Two variants of claims sets are defined:

- Tokens issued to natural persons (e.g. citizens)
- Tokens issued to professionals (persons acting on behalf of an organization)

Later other variant may be added, including claim sets for non-person entities (e.g. software robots) and for EU-citizens logging in via the Danish eID-gateway.

3.1 General requirements

[JTP-01]

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

3.2 Required claims for all token types

[JTP-02]

Tokens issued to persons and professionals MUST include all the claims listed below with non-empty values as specified:

Claim	Value
iss	Identifier for the issuer as an URL using https scheme. Example: 'https://nemlog-in.dk'
jti	A unique identifier for the token, which can be used to prevent reuse of the token.
sub	Subject identifier containing a persistent and service-provider specific UUID in the format specified in OIOSAML 3.0 requirement [OIO-IDP-15]. Example: 'https://data.gov.dk/model/core/eid/**person**/uuid/123e4567-e89b-12d3-a456-426655440000' Note that the subject will indicate the type of identity (person or professional).
aud	Audience for the token as an EntityID following OIOSAML 3.0 [OIO-IDP-18]. Example: 'https://digitalpost.dk'.
exp	Expiration time. JSON number with the same clock skew tolerance as defined in OIOSAML 3.0 [OIO-GE-01].
iat	Time at which the token was issued as a JSON number.
auth_time	Time when the end-user authentication occurred as a JSON number.

nonce	Client session ID received in the request which is relayed back to prevent man-in-the-middle attacks. Note that requirements for entropy is defined in the OIO OIDC Profile.
acr	Authentication methods reference using the NSIS assurance level achieved during authentication. MUST be one of the following: https://data.gov.dk/concept/core/nsis/loa/Low https://data.gov.dk/concept/core/nsis/loa/Substantial https://data.gov.dk/concept/core/nsis/loa/High
spec_ver	Version of this token specification, currently '1.0'.

3.3 Optional claims for persons

[JTP-03]

Tokens issued for persons MAY include the claims specified below with values as specified. The desired claims set SHOULD be agreed in advance via out-of-band mechanisms.

Claim	Value
priv	Privileges according to OIO Basic Privilege Profile 1.1 encoded as JSON (see chapter 4 for details).
ial	Identity Assurance Level as specified on OIOSAML 3.0 section 6.2.5
aal	Authenticator Assurance Level as specified on OIOSAML 3.0 section 6.2.6
name	Full name as specified in OIOSAML 3.0 section 6.2.7
given_name	First name as specified in OIOSAML 3.0 section 6.2.8
family_name	Last name as specified in OIOSAML 3.0 section 6.2.9
email	Email as specified in OIOSAML 3.0 section 6.2.11
cpr	CPR number as specified in OIOSAML 3.0 section 6.2.12
cpr_uuid	CPR UUID as specified in OIOSAML 3.0 section 6.2.11

3.4 Mandatory claims for professionals

[JTP-04]

Tokens issued for professionals MUST (in addition to claims specified in [JTP-02]) include the claims specified below with values as specified.

Claim	Value

cvr	CVR number as specified in OIOSAML 3.0 section 6.4.3
org_name	Name of organization as specified in OIOSAML 3.0 section 6.4.4

3.5 Optional claims for professionals

[JTP-05]

Tokens issued for professionals MAY include the claims specified below with values as specified. The desired claims set SHOULD be agreed in advance via out-of-band mechanisms.

Claim	Value
priv	Privileges according to OIO Basic Privilege Profile 1.1 encoded as JSON (see chapter 4 for details).
ial	Identity Assurance Level as specified on OIOSAML 3.0 section 6.2.5
aal	Authenticator Assurance Level as specified on OIOSAML 3.0 section 6.2.6
name	Full name as specified in OIOSAML 3.0 section 6.2.7
given_name	First name as specified in OIOSAML 3.0 section 6.2.8
family_name	Last name as specified in OIOSAML 3.0 section 6.2.9
email	Email as specified in OIOSAML 3.0 section 6.2.11
cpr	CPR number as specified in OIOSAML 3.0 section 6.2.12
cpr_uuid	CPR UUID as specified in OIOSAML 3.0 section 6.2.11
auth_to_repr	Set when the person is authorized to represent the organization as specified in OIOSAML 3.0 section 6.4.7
p_number	Production number as specified in OIOSAML 3.0 section 6.4.5
se_number	SE number as specified in OIOSAML 3.0 section 6.4.6
persistent_id	Persistent identifier across service providers as specified in OIOSAML 3.0 section 6.4.1

3.6 Signature and validation requirements

[JTP-06]

Tokens MUST be signed using [JWS] using one of the following algorithms from [JWA]:

- PS256, PS384, PS512 (RSA)

- ES256, ES384, ES512 (ECDSA)

[JTP-07]

Token signatures MUST be verified against a pinned certificate provided as part of the secure configuration (e.g. NemLog-in token signing certificate). Tokens with invalid signatures or algorithms MUST be rejected. Revocation checks of pinned token signing certificates is not required.

[JTP-08]

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

[JTP-09]

The following JWS header fields MUST not be used: x5u, x5c, jku, or jwk.

4 Privileges in JSON encoding

This section describes how to encode a set of assigned of 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 of these are simple text strings.[2]

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"
                 }
          ]
       }
    ]
}
```

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

5 Example (not normative)

Below is show an example of claims sections for a person JWT:

```
{
  "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" : 1317281970,
  "iat" : 1311280970,
  "auth_time" : 1311280969,
  "nonce" : "n-0S6_WzA2Mj",
  "acr" : "https://data.gov.dk/concept/core/nsis/loa/Substantial",
  “specver” : ”1.0”,
  “cpr” : “2611771023”
}
```

6 Other considerations (not normative)

6.1 Encryption

Since tokens are not transported via the user agent and are always sent over TLS 1.23 (or higher) with pinned server certificates, application-level encryption of tokens is not deemed necessary.

6.2 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 (see requirement OIDC-63).

6.3 Differences between ID- and access tokens

As mentioned in the introduction, this profile will be used with two types of tokens:

- ID tokens describing an authenticated end-user.
- Access tokens providing access to a REST API on behalf of an end-user - for example using privileges described in chapter 4.

Even though the claims are the same, a few differences will exist in claim values:

Claim	ID tokens	Access Tokens
aud	The app will be the audience. The structure of entityID for apps will be defined in the NemLog-in registration model (separate document.)	The entityID of the API will be the audience.
priv	Contains user's rights in the app.	Contains the access rights granted by the user to the app with the API.
exp	ID tokens are typically short-lived (minutes). Token lifetime is defined according to implementation policy and not defined here.	Access tokens are typically longer-lived in order to last a typical user session (hours).
auth_time	User authentication typically takes place just before ID tokens are issued.	User authentication can be much longer back in time especially if the access token is refreshed multiple times using a refresh token. By including the original time of user authentication, the API can enforce its own policies.

[3] Requirements for transport protocols are defined in the OIO OIDC profile to be used with this token profile.

6.4 Holder-of-key tokens

Access tokens (not ID tokens) according to this profile MAY be issued as holder-of-key tokens by including a `cnf` confirmation claim that contains a SHA-256 thumbprint of the client certificate (via the `x5t#S256` element). Holder-of-key tokens provide additional security against attacks where a stolen token is presented by an illegitimate client, since usage of a holder-of-key token requires proof of possession of a private key corresponding to the (pinned) certificate. See [HOK] and [OIO OIDC] for additional details.

Example:

```
{
  "cnf": {
      "x5t#S256" : "w5cK0ebwmCZUYDB2Y5SlESsXE8o9yZg05O89jdNidgI"
  }
}
```

Note: holder-of-key tokens are only relevant for confidential clients which can adequately protect a private key (e.g. 'confidential clients').

7 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 Proposed 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.
- [HOK] Campbell, Bradley, Sakimura: "OAuth 2.0 Mutual-TLS Client Authentication and Certificate-Bound Access Tokens", https://datatracker.ietf.org/doc/html/rfc8705.
- [NSIS] ”National Standard for Identiteters Sikringsniveauer 2.0.1”. https://digst.dk/it-loesninger/nemlog-in/det-kommende-nemlog-in/vejledninger-og-standarder/nsis-standarden/
- [OIOSAML] ”OIOSAML Web SSO Profile 3.0.2”. https://digst.dk/it-loesninger/nemlog-in/det-kommende-nemlog-in/vejledninger-og-standarder/oiosaml-302/
- [OIO-BPP] ”OIO Basic Privilege Profile 1.2”. https://digst.dk/media/20999/oiosaml-basic-privilege-profile-1_2.pdf
- [OIO OIDC] "OIO OIDC Profile V0.9", Danish Agency for Digitisation. https://digst.dk/it-loesninger/nemlog-in/det-kommende-nemlog-in/vejledninger-og-standarder/openid-connect-profiler/
- [RFC6819] “OAuth 2.0 Threat Model and Security Considerations”, IETF. https://tools.ietf.org/html/rfc6819