KOMBIT
JWT Token Profile
Version 0.9

Status: Draft
Date: 12.04.2021

1 Introduction

This document contains a profile of JSON Web Tokens (JWT) tokens prepared by KOMBIT, which builds on the [OIO-JWT] profile from the Danish Agency for Digitisation.

The goal of the profile is to add support for system user scenarios. Scenarios with 'pure' system clients are missing from the OIO OIDC + JWT profiles documents from DIGST which only covers clients acting on behalf of human users.

Note: this profile has a companion profile 'KOMBIT OAuth Token Request Profile' [KOMBIT-TRP] that explains the scenario and specifies protocols for requesting JWT tokens and how to use these to invoke REST APIs offered by a Service Provider.

2 System user JWT token requirements

This chapter contains requirements for JWT tokens issued to clients acting as system users.

2.1 General requirements

[JTP-01]

All tokens MUST comply with the [JWT] specification.

2.2 Required claims

[JTP-02]

Tokens issued for clients acting as system users MUST include all the claims listed below with values as specified:

Claim	Value
iss	Identifier for the token issuer as an URI. Example: 'https://sts.kombit.dk'
jti	A unique identifier for the token, which can be used to prevent playback / reuse of the token. A UUID v4 SHOULD be used.
sub	Subject identifier specifying the client as an UUID or EntityID.
aud	Audience specifies the Service Provider for which the token is issued as an EntityID following OIOSAML 3.0 [OIO-IDP-18]. Example: http://entityid.kombit.dk/service/sp/eindkomst/3.
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.
spec_ver	Version of this token specification, currently '1.0'.
x5t#S256	Contains the SHA-256 thumbprint of the client certificate which is used to bind the Access Token to the client via the 'Holder-of-key' property. In other words, only a client that can establish a TLS session with the referenced client certificate towards a Service Provider can successfully present the token. The parameter is a base64url-encoded SHA-256 thumbprint (a.k.a. digest) of the DER encoding of the X.509 certificate [RFC5280] used as client certificate. See the [JWS] specification for details.
cvr	Specifies the organization(s) the client is acting on behalf of (e.g. 'anvenderkontekst'). In KOMBIT's setting it shall be either a single 8 digit CVR-number or a short hand string / wildcard denoting a group of CVR-numbers[1].

2.3 Optional claims

[JTP-03]

Tokens MAY further 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 3 for details).

2.4 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. Authorization Server 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 MUST 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.

[1] The list of allowed short-hand / wildcard strings representing multiple CVR numbers is handled outside this profile document.

3 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.

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"
                 }
          ]
       }
    ]
}
```

4 Example (not normative)

Below is shown an example of the claims sections of a system user JWT token:

```
{
  "iss" : "https://sts.kombit.dk",
  "jti" : "423e4567-e81b-12d3-a456-426614174000",
  "sub" : "89b580f7-5fec-4614-b83b-8b1bf4a9d32b",
  "aud"  :  ”https://kombit.dk/organisation",
  "exp" : 1317281970,
  "iat" : 1311280970,
  “specver” : ”1.0”,
  “cvr” : “28182838”,
  "x5t#S256" : "w5cK0ebwmCZUYDB2Y5SlESsXE8o9yZg05O89jdNidgI"
}
```

5 References

- [OIO-JWT] "OIO JWT Token Profile V0.3", Danish Digitisation Agency.
- [KOMBIT-TRP] "KOMBIT OAuth Token Request Profile 0.1", KOMBIT.
- [JWA] Jones, M., “JSON Web Algorithms (JWA),” draft-ietf-jose-json-web-algorithms (work in progress), July 2014.
- [JWE] Jones, M., Rescorla, E., and J. Hildebrand, “JSON Web Encryption (JWE),” draft-ietf-jose-json-web-encryption (work in progress).
- [JWK] Jones, M., “JSON Web Key (JWK),” draft-ietf-jose-json-web-key (work in progress), July 2014.
- [JWS] Jones, M., Bradley, J., and N. Sakimura, “JSON Web Signature (JWS),” draft-ietf-jose-json-web-signature (work in progress), July 2014.
- [JWT] Jones, M., Bradley, J., and N. Sakimura, “JSON Web Token (JWT),” draft-ietf-oauth-json-web-token (work in progress), July 2014.
- [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”. https://digst.dk/it-loesninger/nemlog-in/det-kommende-nemlog-in/vejledninger-og-standarder/oiosaml-30/
- [OIO-BPP] ”OIO Basic Privilege Profile 1.1”. https://www.digitaliser.dk/resource/2377872
- [RFC6819] “OAuth 2.0 Threat Model and Security Considerations”, IETF.
- https://tools.ietf.org/html/rfc6819