

draft-lombardo-oauth- client-extension-claims



Alex Babeanu
Lead Product Manager@Indykite
(Former CTO@3Edges)



in/ababeanu
@babssy.bsky.social



Jeff Lombardo
Prin. Identity Specialist@AWS



in/jflombardo
@authnopuz.xyz



Subject details? OK so far

ACR provides information about the strength of the authentication mechanism or process

Normalized by URI that identifies the authentication context class reference

See **ISO/IEC 29115**

See **OpenID Provider Authentication Policy Extension 1.0**

AMR provides JSON array of strings that are identifiers for authentication methods used in the authentication

Normalized by IANA

See **RFC 8176 - Authentication Method Reference Values**

■ And Pam Dingle / Dean Saxe *Stop thinking in FACTORS* - Authenticate 2024

Framework implementation requirements

(1/2)

- FAPI 2.0 Security Profile / 5.3.1.1 General Requirements
 - shall support confidential clients as defined in [RFC6749]
 - shall only issue sender-constrained access tokens,
 - shall use one of the following methods for sender-constrained access tokens:
 - MTLS as described in [[RFC8705](#) / OAuth 2.0 Mutual-TLS Client Authentication and Certificate-Bound Access Tokens]
 - DPOP as described in [[RFC9449](#) / OAuth 2.0 Demonstrating Proof-of-Possession at the Application Layer (DPOP)]
 - shall authenticate clients using one of the following methods:
 - MTLS as specified in section 2 of [[RFC8705](#) / OAuth 2.0 Mutual-TLS Client Authentication and Certificate-Bound Access Tokens]
 - private_key_jwt as specified in section 9 of [[OIDC](#) Core]

Framework implementation requirements

(2/2)

- SMART on FHIR /
[section 4.1.5 Authenticating to the Token Endpoint](#)

After generating an authentication JWT, the client requests an access token following either the [SMART App Launch](#) or the [SMART Backend Services](#) specification. Authentication details are conveyed using the following additional properties on the token request:

Parameters

<code>client_assertion_type</code>	required	Fixed value: <code>urn:ietf:params:oauth:client-assertion-type:jwt-bearer</code>
<code>client_assertion</code>	required	Signed authentication JWT value (see above)

Client Details? How to get assurance...

1. Type of token? (Identity or Access)

Identity Token specifics:

- does not contain a `scope` claim
- does contain a `nonce` claim
- does contain a `c_hash` claim representing the authorization code, if applicable
- does contain a `at_hash` claim representing the hash of the access token issued along
- does contain a `sub` claim different from `aud` claim value
- might contain a `auth_time` claim representing the authentication time if explicitly requested

Access Token specifics:

- does contain a `scope` claim (sometimes: a `scp` claim and non spaced delimited values)
- does not contain a `nonce` claim.
- does not contain a `c_hash` claim
- does not contain a `at_hash` claim
- does contain a `sub` claim different from `client_id` (if authorization code grant flow)
 - or a `sub` claim equal to the `client_id` (if client credentials grant flow)
- might contain a header with a `typ` claim equal to `application/at+jwt` (if RFC 9068)
- might contain a `auth_time` claim representing the authentication time if explicitly requested

2. Type of Authorization flow? (Device code, Client credentials, CIBA,...)
3. Binding to extensions? (PKCE, RAR, PAR, JAR, DPOP, SD-JWT)
4. Type of client authentication? (private JWT, mTLS, Software assertion)

We MIGHT be able to use header field, but...

\$1,000

Which is a correct token header `typ` field?

- ◆ **A:** application/authorization-grant+dpop+jwt
- ◆ **B:** application/authorization-code+jwt+dpop
- ◆ **C:** application/jwt+dpop+authorization-code
- ◆ **D:** application/authorization-grant+jwt+dpop

“+” ??

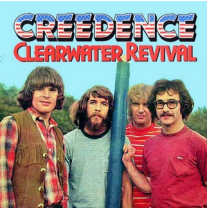
typ explosion !

Missing client authentication methods

PDPs may not be able to access token header

So we beg for differentiated

New proposed claims for JWT profiled Access Token

Claim name	Client focus objective	Format	Expected values	Note
gty	Describes the grant flow type used	Monovalued String	$ac^{(1)}, cc^{(2)}, dc^{(3)}, ciba$ <i>¹⁾ authorization code grant, ²⁾ client credentials grant, ³⁾ device code grant, including imp (implicit grant), rop (resource owner password grant) for legacy support</i>	
cxt	Describes extensions used	Space separated value list	$pkce, dpop, rar, par, jar$	typ header field is not sufficient
ccr	Describes <u>Client</u> authentication context	Monovalued URI	Custom defined namespace* <i>(*) client authentication assurance is currently looked at by WIMSE</i>	
cmr	Describes <u>Client</u> authentication context	Monovalued String	$csec^{(1)}, jwt, mtl, spiffe, htasm^{(2)}, ssa^{(3)}$	Some claims exist like x5t but are specific

Proposal -02

- Description of the four new claims
- No introduction of new Authorization Request parameters
 - AS supporting this proposal would return the values automatically
- Support declaration of capability in Authorization Server Metadata
- Proposition of the registration of 20 new IANA entries
 - OAuth Grant Type: 10
 - OAuth Grant Extension Type: 6
 - OAuth Token Endpoint Authentication Methods: 4
 - Leaving the ability for implementers to register URI as client authentication context class reference

11 Issues processed

- Allowed to sharpen the Security Section
- Made clear connections with:
 - FAPI 2.0 Security Profiles
 - SMART on FHIR (HL7)

Thanks to our Personal Draft reviewers: George Fletcher (Practical Identity), Christopher Langton (Vulnetix)

What are we looking for today?

- Support to move this work as an IETF Draft
- More reviewers from this Working Group to reinforce the proposal