OAuth Working Group
Internet-Draft
Intended status: Standards Track
Expires: November 26, 2021

V. Bertocci
Auth0
May 25, 2021

# JSON Web Token (JWT) Profile for OAuth 2.0 Access Tokens
## draft-ietf-oauth-access-token-jwt-13

## Abstract

This specification defines a profile for issuing OAuth 2.0 access tokens in JSON web token (JWT) format. Authorization servers and resource servers from different vendors can leverage this profile to issue and consume access tokens in an interoperable manner.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on November 26, 2021.

## Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction . . . . . . . . . . . . . . . . . . . . . . . . 2
   1.1. Requirements Notation and Conventions . . . . . . . . . . 3
   1.2. Terminology . . . . . . . . . . . . . . . . . . . . . . . 4
2. JWT Access Token Header and Data Structure . . . . . . . . . 4
   2.1. Header . . . . . . . . . . . . . . . . . . . . . . . . . 4
   2.2. Data Structure . . . . . . . . . . . . . . . . . . . . . 4
      2.2.1. Authentication Information Claims . . . . . . . . . . 5
      2.2.2. Identity Claims . . . . . . . . . . . . . . . . . . . 5
      2.2.3. Authorization Claims . . . . . . . . . . . . . . . . 6
         2.2.3.1. Claims for Authorization Outside of Delegation Scenarios . . . . . . . . . . . . . . . . . . . . . 6
3. Requesting a JWT Access Token . . . . . . . . . . . . . . . . 7
4. Validating JWT Access Tokens . . . . . . . . . . . . . . . . 8
5. Security Considerations . . . . . . . . . . . . . . . . . . . 10
6. Privacy Considerations . . . . . . . . . . . . . . . . . . . 11
7. IANA Considerations . . . . . . . . . . . . . . . . . . . . . 12
   7.1. Media Type Registration . . . . . . . . . . . . . . . . . 12
      7.1.1. Registry Content . . . . . . . . . . . . . . . . . . 12
   7.2. Claims Registration . . . . . . . . . . . . . . . . . . . 13
      7.2.1. Registry Contents . . . . . . . . . . . . . . . . . . 13
         7.2.1.1. Roles . . . . . . . . . . . . . . . . . . . . . 13
         7.2.1.2. Groups . . . . . . . . . . . . . . . . . . . . 14
         7.2.1.3. Entitlements . . . . . . . . . . . . . . . . . 14
8. References . . . . . . . . . . . . . . . . . . . . . . . . . 14
   8.1. Normative References . . . . . . . . . . . . . . . . . . 14
   8.2. Informative References . . . . . . . . . . . . . . . . . 16
Appendix A. Acknowledgements . . . . . . . . . . . . . . . . . . 16
Appendix B. Document History . . . . . . . . . . . . . . . . . . 17
Author's Address . . . . . . . . . . . . . . . . . . . . . . . . 22

## 1. Introduction

The original OAuth 2.0 Authorization Framework [RFC6749] specification does not mandate any specific format for access tokens. While that remains perfectly appropriate for many important scenarios, in-market use has shown that many commercial OAuth 2.0 implementations elected to issue access tokens using a format that can be parsed and validated by resource servers directly, without further authorization server involvement.  The approach is particularly common in topologies where the authorization server and resource server are not co-located, are not run by the same entity, or are otherwise separated by some boundary.  At the time of writing, many commercial implementations leverage the JSON Web Tokens (JWT) [RFC7519] format.

Many vendor specific JWT access tokens share the same functional layout, using JWT claims to convey the information needed to support a common set of use cases: token validation, transporting authorization information in forms of scopes and entitlements, carrying identity information about the subject, and so on. The differences are mostly confined to the claim names and syntax used to represent the same entities, suggesting that interoperability could be easily achieved by standardizing on a common set of claims and validation rules.

The assumption that access tokens are associated to specific information doesn't appear only in commercial implementations. Various specifications in the OAuth 2.0 family (such as resource indicators [RFC8707], OAuth 2.0 bearer token usage [RFC6750] and others) postulate the presence in access tokens of scoping mechanisms, such as an audience. The family of specifications associated to introspection also indirectly suggest a fundamental set of information access tokens are expected to carry or at least be associated with.

This specification aims to provide a standardized and interoperable profile as an alternative to the proprietary JWT access token layouts going forward. Besides defining a common set of mandatory and optional claims, the profile provides clear indications on how authorization request parameters determine the content of the issued JWT access token, how an authorization server can publish metadata relevant to the JWT access tokens it issues, and how a resource server should validate incoming JWT access tokens.

Finally, this specification provides security and privacy considerations meant to prevent common mistakes and anti patterns that are likely to occur in naive use of the JWT format to represent access tokens.

Please note: although both this document and [RFC7523] use JSON Web Tokens in the context of the OAuth2 framework, the two specifications differ in both intent and mechanics. Whereas [RFC7523] defines how a JWT Bearer Token can be used to request an access token, this documents describes how to encode access tokens in JWT format.

## 1.1. Requirements Notation and Conventions

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.

### 1.2. Terminology

JWT access token: An OAuth 2.0 access token encoded in JWT format and complying with the requirements described in this specification.

This specification uses the terms "access token", "refresh token", "authorization server", "resource server", "authorization endpoint", "authorization request", "authorization response", "token endpoint", "grant type", "access token request", "access token response", and "client" defined by The OAuth 2.0 Authorization Framework [RFC6749].

## 2. JWT Access Token Header and Data Structure

### 2.1. Header

JWT access tokens MUST be signed. Although JWT access tokens can use any signing algorithm, use of asymmetric cryptography is RECOMMENDED as it simplifies the process of acquiring validation information for resource servers (see Section 4). JWT access tokens MUST NOT use "none" as the signing algorithm. See Section 4 for more details.

Authorization servers and resource servers conforming to this specification MUST include RS256 (as defined in [RFC7518]) among their supported signature algorithms.

This specification registers the "application/at+jwt" media type, which can be used to indicate that the content is a JWT access token. JWT access tokens MUST include this media type in the "typ" header parameter to explicitly declare that the JWT represents an access token complying with this profile. Per the definition of "typ" in Section 4.1.9 of [RFC7515], it is RECOMMENDED that the "application/" prefix be omitted. Therefore, the "typ" value used SHOULD be "at+jwt". See the security considerations section for details on the importance of preventing OpenID Connect ID Tokens (as defined by Section 2 of [OpenID.Core]) from being accepted as access tokens by resource servers implementing this profile.

### 2.2. Data Structure

The following claims are used in the JWT access token data structure.

iss  REQUIRED - as defined in Section 4.1.1 of [RFC7519].

exp  REQUIRED - as defined in Section 4.1.4 of [RFC7519].

aud  REQUIRED - as defined in Section 4.1.3 of [RFC7519]. See Section 3 for indications on how an authorization server should determine the value of "aud" depending on the request.

sub  REQUIRED - as defined in Section 4.1.2 of [RFC7519]. In case of access tokens obtained through grants where a resource owner is involved, such as the authorization code grant, the value of "sub" SHOULD correspond to the subject identifier of the resource owner. In case of access tokens obtained through grants where no resource owner is involved, such as the client credentials grant, the value of "sub" SHOULD correspond to an identifier the authorization server uses to indicate the client application. See Section 5 for more details on this scenario. Also, see Section 6 for a discussion about how different choices in assigning "sub" values can impact privacy.

client_id  REQUIRED - as defined in Section 4.3 of [RFC8693].

iat  REQUIRED - as defined in Section 4.1.6 of [RFC7519]. This claim identifies the time at which the JWT access token was issued.

jti  REQUIRED - as defined in Section 4.1.7 of [RFC7519].

### 2.2.1. Authentication Information Claims

The claims listed in this section MAY be issued in the context of authorization grants involving the resource owner, and reflect in the access token the types and strength of authentication that the authentication server enforced prior to returning the authorization response to the client. Their values are fixed, and remain the same across all access tokens that derive from a given authorization response, whether the access token was obtained directly in the response (e.g., via the implicit flow) or after one or more token exchanges (e.g., obtaining a fresh access token using a refresh token, or exchanging one access token for another via [RFC8693] procedures).

auth_time  OPTIONAL - as defined in Section 2 of [OpenID.Core].

acr  OPTIONAL - as defined in Section 2 of [OpenID.Core].

amr  OPTIONAL - as defined in Section 2 of [OpenID.Core].

### 2.2.2. Identity Claims

In the context of authorization grants involving the resource owner, commercial authorization servers will often include resource owner attributes directly in access tokens, so that resource servers can

consume them directly for authorization or other purposes without any further round trips to introspection ( [RFC7662]) or UserInfo ( [OpenID.Core]) endpoints. This is particularly common in scenarios where the client and the resource server belong to the same entity and are part of the same solution, as is the case for first party clients invoking their own backend API.

This profile does not introduce any mechanism for a client to directly request the presence of specific claims in JWT access tokens, as the authorization server can determine what additional claims are required by a particular resource server by taking in consideration the client_id of the client, the "scope" and the "resource" parameters included in the request.

Any additional identity attribute whose semantic is well described by an entry in the JSON Web Token (JWT) IANA registry introduced in [RFC7519] SHOULD be encoded using the corresponding claim name, if that attribute is to be included in the JWT access token. Note that the JWT IANA registry includes the claims found in Section 5.1 of [OpenID.Core].

Authorization servers MAY return arbitrary attributes not defined in any existing specification, as long as the corresponding claim names are collision resistant or the access tokens are meant to be used only within a private subsystem. Please refer to Sections 4.2 and 4.3 of [RFC7519] for details.

Authorization servers including resource owner attributes in JWT access tokens need to exercise care and verify that all privacy requirements are met, as discussed in Section 6.

### 2.2.3. Authorization Claims

If an authorization request includes a scope parameter, the corresponding issued JWT access token SHOULD include a "scope" claim as defined in Section 4.2 of [RFC8693].

All the individual scope strings in the "scope" claim MUST have meaning for the resources indicated in the "aud" claim. See Section 5 for more considerations about the relationship between scope strings and resources indicated by the "aud" claim.

#### 2.2.3.1. Claims for Authorization Outside of Delegation Scenarios

Many authorization servers embed in the access tokens they issue authorization attributes that go beyond the delegated scenarios described by [RFC7519]. Typical examples include resource owner memberships in roles and groups that are relevant to the resource

being accessed, entitlements assigned to the resource owner for the targeted resource that the authorization server knows about, and so on.

An authorization server wanting to include such attributes in a JWT access token SHOULD use as claim types the "groups","roles" and "entitlements" attributes of the "User" resource schema defined by Section 4.1.2 of [RFC7643]).

Authorization servers SHOULD encode the corresponding claim values according to the guidance defined in [RFC7643]. In particular, a non-normative example of "groups" attribute can be found in Section 8.2 of [RFC7643]. No specific vocabulary is provided for "roles" and "entitlements".

Section 7 of this document provides entries for registering "groups", "roles" and "entitlements" attributes from [RFC7643] as claim types to be used in this profile.

## 3. Requesting a JWT Access Token

An authorization server can issue a JWT access token in response to any authorization grant defined by [RFC6749] and subsequent extensions meant to result in an access token.

If the request includes a "resource" parameter (as defined in [RFC8707]), the resulting JWT access token "aud" claim SHOULD have the same value as the "resource" parameter in the request.

Example request below:

```
GET /as/authorization.oauth2?response_type=code
        &client_id=s6BhdRkqt3
        &state=xyz
        &scope=openid%20profile%20reademail
        &redirect_uri=https%3A%2F%2Fclient%2Eexample%2Ecom%2Fcb
        &resource=https%3A%2F%2Frs.example.com%2F HTTP/1.1
     Host: authorization-server.example.com
```

Figure 1: Authorization Request with Resource and Scope Parameters

Once redeemed, the code obtained from the request above will result in a JWT access token in the form shown below:

Header:

```
{"typ":"at+JWT","alg":"RS256","kid":"RjEwOwOA"}
```

Claims:

```
{
  "iss": "https://authorization-server.example.com/",
  "sub": "5ba552d67",
  "aud":   "https://rs.example.com/",
  "exp": 1639528912,
  "iat": 1618354090,
  "jti" : "dbe39bf3a3ba4238a513f51d6e1691c4",
  "client_id": "s6BhdRkqt3",
  "scope": "openid profile reademail"
}
```

Figure 2: The Header and JWT Claims Set of a JWT Access Token

The authorization server MUST NOT issue a JWT access token if the authorization granted by the token would be ambiguous. See Section 5 for more details about common cases that might lead to ambiguity and strategies an authorization server can enact to prevent them.

If the request does not include a "resource" parameter, the authorization server MUST use in the "aud" claim a default resource indicator. If a "scope" parameter is present in the request, the authorization server SHOULD use it to infer the value of the default resource indicator to be used in the "aud" claim. The mechanism through which scopes are associated to default resource indicator values is outside the scope of this specification. If the values in the "scope" parameter refer to different default resource indicator values, the authorization server SHOULD reject the request with "invalid_scope" as described in Section 4.1.2.1 of [RFC6749].

## 4. Validating JWT Access Tokens

For the purpose of facilitating validation data retrieval, it is here RECOMMENDED that authorization servers sign JWT access tokens with an asymmetric algorithm.

Authorization servers SHOULD use OAuth 2.0 Authorization Server Metadata [RFC8414] to advertise to resource servers their signing keys via "jwks_uri" and what "iss" claim value to expect via the "issuer" metadata value. Alternatively, authorization servers implementing OpenID Connect MAY use the "OpenID Connect discovery [OpenID.Discovery] document for the same purpose. If an

authorization server supports both OAuth 2.0 Authorization Server Metadata and OpenID Connect discovery, the values provided MUST be consistent across the two publication methods.

An authorization server MAY elect to use different keys to sign OpenID Connect ID Tokens and JWT access tokens. This specification does not provide a mechanism for identifying a specific key as the one used to sign JWT access tokens. An authorization server can sign JWT access tokens with any of the keys advertised via AS metadata or OpenID Connect discovery. See Section 5 for further guidance on security implications.

Resource servers receiving a JWT access token MUST validate it in the following manner.

- The resource server MUST verify that the typ header value is "at+jwt" or "application/at+jwt" and reject tokens carrying any other value.

- If the JWT access token is encrypted, decrypt it using the keys and algorithms that the resource server specified during registration. If encryption was negotiated with the authorization server at registration time and the incoming JWT access token is not encrypted, the resource server SHOULD reject it.

- The Issuer Identifier for the authorization server (which is typically obtained during discovery) MUST exactly match the value of the "iss" claim.

- The resource server MUST validate that the "aud" claim contains a resource indicator value corresponding to an identifier the resource server expects for itself. The JWT access token MUST be rejected if "aud" does not contain a resource indicator of the current resource server as a valid audience.

- The resource server MUST validate the signature of all incoming JWT access tokens according to [RFC7515] using the algorithm specified in the JWT alg Header Parameter. The resource server MUST reject any JWT in which the value of "alg" is "none". The resource server MUST use the keys provided by the authorization server.

- The current time MUST be before the time represented by the "exp" claim. Implementers MAY provide for some small leeway, usually no more than a few minutes, to account for clock skew.

The resource server MUST handle errors as described in Section 3.1 of [RFC6750]. In particular, in case of any failure in the validation

checks listed above the authorization server response MUST include the error code "invalid_token". Please note that this requirement does not prevent JWT access tokens to use authentications schemes other than Bearer.

If the JWT access token includes authorization claims as described in Section 2.2.3, the resource server SHOULD use them in combination with any other contextual information available to determine whether the current call should be authorized or rejected. Details about how a resource server performs those checks is beyond the scope of this profile specification.

## 5. Security Considerations

The JWT access token data layout described here is very similar to the one of the id_token as defined by [OpenID.Core]. The explicit typing required in this profile, in line with the recommendations in [RFC8725] helps the resource server to distinguish between JWT access tokens and OpenID Connect ID Tokens.

Authorization servers should prevent scenarios where clients can affect the value of the "sub" claim in ways that could confuse resource servers. For example, if the authorization server elects to use the client_id as the "sub" value for access tokens issued using the client credentials grant, the authorization server should prevent clients from registering an arbitrary client_id value, as this would allow malicious clients to select the sub of a high privilege resource owner and confuse any authorization logic on the resource server relying on the "sub" value. For more details please refer to Section 4.14 of [OAuth2.Security.BestPractices].

To prevent cross-JWT confusion, authorization servers MUST use a distinct identifier as "aud" claim value to uniquely identify access tokens issued by the same issuer for distinct resources. For more details on cross-JWT confusion please refer to Section 2.8 of [RFC8725].

Authorization servers should use particular care when handling requests that might lead to ambiguous authorization grants. For example: if a request includes multiple resource indicators, the authorization server should ensure that each scope string included in the resulting JWT access token, if any, can be unambiguously correlated to a specific resource among the ones listed in the "aud" claim. The details on how to recognize and mitigate this and other ambiguous situations is highly scenario-dependent, hence out of scope for this profile.

Authorization servers cannot rely on the use of different keys for signing OpenID Connect ID Tokens and JWT tokens as a method to safeguard against the consequences of leaking specific keys. Given that resource servers have no way of knowing what key should be used to validate JWT access tokens in particular, they have to accept signatures performed with any of the keys published in AS metadata or OpenID Connect discovery: consequently, an attacker just needs to compromise any key among the ones published to be able to generate and sign JWTs that will be accepted as valid by the resource server.

## 6. Privacy Considerations

As JWT access tokens carry information by value, it now becomes possible for clients and potentially even end users to directly peek inside the token claims collection of unencrypted tokens.

The client MUST NOT inspect the content of the access token: the authorization server and the resource server might decide to change token format at any time (for example by switching from this profile to opaque tokens) hence any logic in the client relying on the ability to read the access token content would break without recourse. The OAuth 2.0 framework assumes that access tokens are treated as opaque by clients. Administrators of authorization servers should also take into account that the content of an access token is visible to the client. Whenever client access to the access token content presents privacy issues for a given scenario, the authorization server needs to take explicit steps to prevent it.

In scenarios in which JWT access tokens are accessible to the end user, it should be evaluated whether the information can be accessed without privacy violations (for example, if an end user would simply access his or her own personal information) or if steps must be taken to enforce confidentiality.

Possible measures to prevent leakage of information to clients and end users include: encrypting the access token, encrypting the sensitive claims, omitting the sensitive claims or not using this profile, falling back on opaque access tokens.

In every scenario, the content of the JWT access token will eventually be accessible to the resource server. It's important to evaluate whether the resource server gained the proper entitlement to have access to any content received in form of claims, for example through user consent in some form, policies and agreements with the organization running the authorization servers, and so on. For example, a user might not wish to consent to granting a given resource server information about any of the non-mandatory claims enumerated in Section 2 (and subsections)."

This profile mandates the presence of the "sub" claim in every JWT access token, making it possible for resource servers to rely on that information for correlating incoming requests with data stored locally for the authenticated principal.  Although the ability to correlate requests might be required by design in many scenarios, there are scenarios where the authorization server might want to prevent correlation.  The "sub" claim should be populated by the authorization servers according to a privacy impact assessment.  For instance, if a solution requires preventing tracking principal activities across multiple resource servers, the authorization server should ensure that JWT access tokens meant for different resource servers have distinct "sub" values that cannot be correlated in the event of resource servers collusion.  Similarly, if a solution requires preventing a resource server from correlating the principal's activity within the resource itself, the authorization server should assign different "sub" values for every JWT access token issued.  In turn, the client should obtain a new JWT access token for every call to the resource server, to ensure that the resource server receives different "sub" and "jti" values at every call, thus preventing correlation between distinct requests.

## 7. IANA Considerations

### 7.1. Media Type Registration

#### 7.1.1. Registry Content

This section registers the "application/at+jwt" media type [RFC2046] in the "Media Types" registry [IANA.MediaTypes] in the manner described in [RFC6838], which can be used to indicate that the content is an access token encoded in JWT format.

- Type name: application

- Subtype name: at+jwt

- Required parameters: N/A

- Optional parameters: N/A

- Encoding considerations: binary; JWT values are encoded as a series of base64url-encoded values (with trailing '=' characters removed), some of which may be the empty string, separated by period ('.') characters.

- Security considerations: See the Security Considerations Section of [[TODO: update once there's a RFC number for the JWT AT profile]]

- Interoperability considerations: N/A

- Published specification: [[TODO: update once there's a RFC number for the JWT AT profile]]

- Applications that use this media type: Applications that access resource servers using OAuth 2.0 access tokens encoded in JWT format

- Fragment identifier considerations: N/A

- Additional information: Magic number(s): N/A File extension(s): N/A Macintosh file type code(s): N/A

- Person and email address to contact for further information: Vittorio Bertocci, vittorio@auth0.com

- Intended usage: COMMON

- Restrictions on usage: none

- Author: Vittorio Bertocci, vittorio@auth0.com

- Change controller: IESG

- Provisional registration?  No

## 7.2. Claims Registration

Section 2.2.3.1 of this specification refers to the attributes "roles", "groups", "entitlements" defined in [RFC7643] to express authorization information in JWT access tokens.  This section registers those attributes as claims in the JSON Web Token (JWT) IANA registry introduced in [RFC7519].

### 7.2.1. Registry Contents

#### 7.2.1.1. Roles

- Claim Name: "roles"

- Claim Description: Roles

- Change Controller: IESG

- Specification Document(s): Section 4.1.2 of [RFC7643] and Section 2.2.3.1 of [[this specification]]

#### 7.2.1.2. Groups

- Claim Name: "groups"
- Claim Description: Groups
- Change Controller: IESG
- Specification Document(s): Section 4.1.2 of [RFC7643] and Section 2.2.3.1 of [[this specification]]

#### 7.2.1.3. Entitlements

- Claim Name: "entitlements"
- Claim Description: Entitlements
- Change Controller: IESG
- Specification Document(s): Section 4.1.2 of [RFC7643] and Section 2.2.3.1 of [[this specification]]

## 8. References

### 8.1. Normative References

[OpenID.Core]
Sakimura, N., Bradley, J., Jones, M., Medeiros, B., and C. Mortimore, "OpenID Connect Core 1.0", November 2014, <https://openid.net/specs/openid-connect-core-1_0.html>.

[OpenID.Discovery]
Sakimura, N., Bradley, J., Jones, M., and J. Jay, "OpenID Connect Discovery 1.0", November 2014, <https://openid.net/specs/openid-connect-discovery-1_0.html>.

[RFC2046] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, DOI 10.17487/RFC2046, November 1996, <https://www.rfc-editor.org/info/rfc2046>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC6749] Hardt, D., Ed., "The OAuth 2.0 Authorization Framework", RFC 6749, DOI 10.17487/RFC6749, October 2012, <https://www.rfc-editor.org/info/rfc6749>.

[RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <https://www.rfc-editor.org/info/rfc6838>.

[RFC7515] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015, <https://www.rfc-editor.org/info/rfc7515>.

[RFC7518] Jones, M., "JSON Web Algorithms (JWA)", RFC 7518, DOI 10.17487/RFC7518, May 2015, <https://www.rfc-editor.org/info/rfc7518>.

[RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <https://www.rfc-editor.org/info/rfc7519>.

[RFC7643] Hunt, P., Ed., Grizzle, K., Wahlstroem, E., and C. Mortimore, "System for Cross-domain Identity Management: Core Schema", RFC 7643, DOI 10.17487/RFC7643, September 2015, <https://www.rfc-editor.org/info/rfc7643>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8414] Jones, M., Sakimura, N., and J. Bradley, "OAuth 2.0 Authorization Server Metadata", RFC 8414, DOI 10.17487/RFC8414, June 2018, <https://www.rfc-editor.org/info/rfc8414>.

[RFC8693] Jones, M., Nadalin, A., Campbell, B., Ed., Bradley, J., and C. Mortimore, "OAuth 2.0 Token Exchange", RFC 8693, DOI 10.17487/RFC8693, January 2020, <https://www.rfc-editor.org/info/rfc8693>.

[RFC8707] Campbell, B., Bradley, J., and H. Tschofenig, "Resource Indicators for OAuth 2.0", RFC 8707, DOI 10.17487/RFC8707, February 2020, <https://www.rfc-editor.org/info/rfc8707>.

[RFC8725] Sheffer, Y., Hardt, D., and M. Jones, "JSON Web Token Best Current Practices", BCP 225, RFC 8725, DOI 10.17487/RFC8725, February 2020, <https://www.rfc-editor.org/info/rfc8725>.

## 8.2. Informative References

[OAuth2.Security.BestPractices]
Lodderstedt, T., Bradley, J., Labunets, A., and D. Fett, "OAuth 2.0 Security Best Current Practice", July 2019, <https://datatracker.ietf.org/doc/html/draft-ietf-oauth-security-topics-18>.

[RFC6750] Jones, M. and D. Hardt, "The OAuth 2.0 Authorization Framework: Bearer Token Usage", RFC 6750, DOI 10.17487/RFC6750, October 2012, <https://www.rfc-editor.org/info/rfc6750>.

[RFC7523] Jones, M., Campbell, B., and C. Mortimore, "JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants", RFC 7523, DOI 10.17487/RFC7523, May 2015, <https://www.rfc-editor.org/info/rfc7523>.

[RFC7662] Richer, J., Ed., "OAuth 2.0 Token Introspection", RFC 7662, DOI 10.17487/RFC7662, October 2015, <https://www.rfc-editor.org/info/rfc7662>.

## Appendix A. Acknowledgements

The initial set of requirements informing this specification was extracted by numerous examples of access tokens issued in JWT format by production systems. Thanks to Dominick Baier (IdentityServer), Brian Campbell (Ping Identity), Daniel Dobalian (Microsoft), Karl Guinness (Okta) for providing sample tokens issued by their products and services. Brian Campbell and Filip Skokan provided early feedback that shaped the direction of the specification. This profile was discussed at length during the OAuth Security Workshop 2019, with several individuals contributing ideas and feedback. The author would like to acknowledge the contributions of:

John Bradley, Brian Campbell, Vladimir Dzhuvinov, Torsten Lodderstedt, Nat Sakimura, Hannes Tschofenig and everyone who actively participated in the unconference discussions.

The following individuals contributed useful feedback and insights on the drafts, both on the IETF OAuth 2.0 WG DL and during the IIW28 conference:

Dale Olds, George Fletcher, David Waite, Michael Engan, Mike Jones, Hans Zandbelt, Vladimir Dzhuvinov, Martin Schanzenbach , Aaron Parecki, Annabelle Richard Backman, Dick Hardt, Denis Pinkas, Benjamin Kaduk, Dominick Baier, Andrii Deinega, Mike Jones and everyone who actively participated in the IIW28 unconference

discussions and the IETF OAuth 2.0 WG DL discussions. Thanks to Roman Danyliw for the AD review, Joseph Salowey and Roni Even for the SECDIR/GENART reviews, Francesca Palomini, Lars Eggert, Murray Kucherawy, Roberto Polli, Martin Duke, Benjamin Kaduk for the IESG reviews.

## Appendix B. Document History

[[ to be removed by the RFC Editor before publication as an RFC ]]

draft-ietf-oauth-access-token-jwt-13

- Added an explicit requirement for the JWT access tokens to be signed in Section 2.1.
- Added sections in Section 7.2 for improving readability.
- In Figure 2, added jti and iat claims.
- Editorial fixes in Section 2.2.1 and Section 4.
- Clarified that additional claims are to be added only if necessary - in Section 2.2.2. Editorial changes in the same section.
- Updates sample values in Section 3.
- Added leeway language to the exp validation step in Section 4.
- In Section 4, clarified that the use of RFC6750 error codes does not imply that JWT ATs can only be Bearer tokens.
- Various edits and nits in Section 5 and Section 6.
- Added clarifying language in Section 6 regarding what claims in the AT might require explicit user consent.
- Added URLs for the OpenID references and the oauth security BCP. Moved RFC7523 to the informative references section.

draft-ietf-oauth-access-token-jwt-12

- Editorial changes in the abstract.
- Updated the client_id value in the figures in Section 3.
- Added clarifying language in Section 1 positioning this specification vs [RFC7523].
- In section Section 2.1 added a sentence that makes RS256 support mandatory for AS and RS.

draft-ietf-oauth-access-token-jwt-11

- Editorial updates and typo fixes in Section 1.2, Section 2.2, Section 2.2.1, Section 2.2.2, Section 3, Section 4, Section 5, Appendix A
- Updated language in Section 2.2.2 to explicitly refer to the JWT IANA registry.
- Citation added in in Section 2.1
- Added a normative reference entry for OpenID Connect Discovery 1.0 and referenced it from Section 4

draft-ietf-oauth-access-token-jwt-10

- Updated Figure 2 in Section 3 to clarify that the intent of that snippet is to describe the content rather than exact JWT AT format.
- Updated registry references in Section 7.2.1 to point to Section 2.2.3.1
- Modified Section 2.2.3.1 to make it easier for the reader to understand what values and format is expected for the groups, roles and entitlement claims. Minor formatting issues fixed.

draft-ietf-oauth-access-token-jwt-09

- Removed unused reference to http://www.iana.org/assignments/oauth-parameters; moved the OAuth2 security BCP to the informative references section.
- Restructured opening paragraphs in Section 6 for clarity.

draft-ietf-oauth-access-token-jwt-08

- Numerous edits for correcting typos, improving clarity and precision of language.
- Moved RFC7519 to the normative section; eliminated unused references RFC7644 and RFC3986.

draft-ietf-oauth-access-token-jwt-07

- In Section 2.1, added language that forbids use of none as alg value, and references Section 4 where the same prohibition was already expressed from the RS perspective.
- In the sub definition in Section 2.2, added a sentence that clarifies what goes in the sub in the case of grants where a resource owner is involved.
- Updated acknowledgements.
- Updated Section 2.2.1 to clarify that acr, amr and auth_type can occur if the AT has been obtained by grants where the resource owner is involved.
- Updated Section 2.2.2 to clarify that identity claims can occur if the AT has been obtained by grants where the resource owner is involved.
- In Section 2.2.3.1 eliminated the claim that SCIM doesn't provide a vocabulary for the attributes listed there.
- In Section 5 added reference to 8725.
- In Section 4 added application/jwt+at as accepted typ value.
- Various typos and formatting issues fixed.

draft-ietf-oauth-access-token-jwt-06

- In Section 2.2 and Section 6 added a discussion about how different sub values affect the privacy properties of a solution.
- In Section 2.2.3 and Section 3 eliminated language prohibiting JWT AT requests featuring multiple resources, substituting it with the prohibition for the AS to emit JWT ATs expressing ambiguous authorization grants. In Section 5, added language warning against scope confusion and mentioned the existence of other ambiguous authorization grant.
- In Section 2.2 promoted claims iat and jti from RECOMMENDED to REQUIRED.
- In Section 2.1 eliminated temporary note on the lack of authenticated encryption methods specifications.
- Updated acknowledgements.

draft-ietf-oauth-access-token-jwt-05

- Varios typos, grammar issues and improper abbreviations fixed.
- Reworded the definition of at+jwt in Section 2.1.
- In Section 2.2, clarified that iat refers to the issuance time of the JWT itself.
- In Section 2.2.2, added a reference to public/private claims definitions (Sections 4.2, 4.3) of [RFC7519].
- In Section 3, removed the paragrah stating that every JWT AT MUST have an aud, as it is already defined in Section 2.2.
- Reworded description of the JWT AT adoption landscape in Section 1.
- Simplified the individual descriptions of the claims list in Section 2.2.1.
- Updated Section 4 and Section 5 to clarify that the AS can use any of the published keys to sign JWT access tokens, and that the AS should not rely on use of different signing keys per token type as a security mechanism.
- In Section 2.2 promoted claims iat and jti from OPTIONAL to RECOMMENDED
- In Section 4, switched the validation steps list type from numbers to bullets.
- In Section 4, eliminated the auth_time instructions from the validation steps list.
- In Section 2.2.2, added a reference to the JWT claims registry as source of claims for JWT ATs
- In Section 4, clarified that failures in JWT AT validation checks will result in invalid_token.

draft-ietf-oauth-access-token-jwt-04

- Eliminated reference to resource aliases list from the aud claim description in Section 2.

- Eliminated references to resource aliases list from the aud validation guidance in Section 4.
- Introduced a new subsection Section 2.2.1, moved the definitions of auth_time, acr and amr there and incorporated the language proposed by Annabelle and Brian on the WG mailing list.
- In section Section 3 softened (from MUST to SHOULD) the requirement that ties the resource identifier in the request to the value in the aud claim of the issued access token.
- Updated acknowledgements.
- In the section Section 3, the example request now has response_type=code.
- Updated text in the Privacy Consideration section to clarify what protection steps the text refers to.
- Updated the typ header discussion in Section 2.1 to clarify that it helps preventing resources from accepting OpenID Connect ID Tokens as JWT access tokens.
- Updated refrences to token exchange, resource indicators and JWT best practices to reflect their RFC status (8693,8707,8725).

draft-ietf-oauth-access-token-jwt-03

- Varios typos fixed.
- In the security considerations section, relaxed the claim that the typ header value "at+jwt" will prevent RS from misinterpreting JWT ATs as idtokens.
- In the "Requesting JWT Access Tokens" section, added "invalid_target" as a possible error returned for the multiple resources request case.
- In the Validating JWT Access Tokens" section, disallowed JWTs with "alg":"none"
- in the IANA registration entries for the SCIM claim types, complemented the reference to the SCIM spec with a reference to this spec so that the eventual registration entries have better context.
- Updated acknowledgements.
- In the section Section 3, the example request now has response_type=code.
- Updated text in the Privacy Consideration section to clarify what protection steps the text refers to.

draft-ietf-oauth-access-token-jwt-02

- In 2.2.1, opened the sources of identity attributes to any identity related specification.
- In 2.2.2, relaxed from MUST to SHOULD the requirement that requests including a scope always result in access tkens containing a corresponding scope claim.

- In the security considerations setting, added a requirement for the authorization server to assing unique identifiers for different resources- to prevent cross JWT confusion.
- Added IANA registration for the authorization attributes borrowed from SCIM CORE

draft-ietf-oauth-access-token-jwt-01

- Added note on authenticated encryption.
- Added a mention to the 1st party clients scenarios in the identity claims section.
- Changed the definition reference for the iss, exp, aud, sub, iat claims from OpenID.Core to RFC7519.
- Added a mention of the client_id==sub case in the security considerations section, added a reference to draft-ietf-oauth-security-topics-13. Added a reference to the security considerations from the sub claim definition section.
- Specified invalid_request as the error code the authorization server should return in case of multiple resources in the access token request.
- Specified invalid_scope as the error code the authorization server should return in case it isn;t possible to determine to which resource the requested scopes refers to.
- In the identity claims section, added a reference to introspection as possible source of claim types and added language explicitly stating that the AS can add arbitrary attributes as long as they are collision resistant or private.
- Updated language for the auth_time claim to include the case in which the AS reauthenticates the user mid-session (e.g. during step up auth).
- Removed note about adding a mechanism for extablishing whether the token was obtained on behalf or the resource owner or of the client itself (client credentials grant).
- Removed note about adding a mechanism for indicating whether the authorization server sent the resource owner to authenticate with a federated identity provider, and the identity of that federated provider.
- Removed the note in the security consideration sections about discussing the purpose of aud, iss, exp validation (redundant).
- In the authorization claims section, stated intent to register roles, groups and entitlements as claim types in IANA
- Clarified in the privacy considerations that clients should not inspect access tokens.
- Expanded the privacy considerations with more explicit guidance about privacy preserving approaches.
- Added IANA registry content for the at+JWT MIME type.
- Updated acknowledgements.

draft-ietf-oauth-access-token-jwt-00

- Initial draft to define a JWTt profile for OAuth 2.0 access tokens.

## Author's Address

Vittorio Bertocci
Auth0

Email: vittorio@auth0.com