Structured Access Token for Sharing Authorization Grant between a Resource Server and an Authorization Server
draft-sakimura-oidc-structured-token-01

Abstract

This specification defines a format of structured access tokens that are issued by the authorization server and received by the resource server. By using this format, the authorization server and the resource server can be easily separated and the validation of the access token can be performed offline.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

Copyright and License Notice

Copyright (c) 2013 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 (http://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 ................................................................. 3
   1.1. Terminology ......................................................... 3
2. Assumptions ................................................................. 4
3. Structured Access Token Format .......................................... 5
   3.1. "iss" (Issuer) Claim .............................................. 5
   3.2. "sub" (Subject) Claim .............................................. 5
   3.3. "aud" (Audience) Claim ........................................... 5
   3.4. "exp" (Expiration Time) Claim ................................... 5
   3.5. "nbf" (Not Before) Claim ......................................... 5
   3.6. "iat" (Issued At) Claim ......................................... 5
   3.7. "claims" Claim .................................................... 5
   3.8. "azp" Claim ....................................................... 6
4. Structured Access Token Usages ........................................... 7
5. Examples of the structured access token usage ........................... 8
   5.1. UserInfo Request .................................................. 8
   5.2. Aggregated/Distributed Claims in OpenID Connect ................ 8
   5.3. Other Usages ...................................................... 9
6. Security Considerations .................................................. 10
7. Privacy Considerations .................................................. 10
8. IANA Considerations ..................................................... 10
9. References ................................................................. 10
   9.1. Normative References .......................................... 10
   9.2. Informative References ....................................... 11
Appendix A. Document History .............................................. 12
Authors’ Addresses .......................................................... 12
1. Introduction

The OAuth 2.0 authorization framework ([RFC6749]) defines a method for third-party client authorization using access tokens. In OAuth, an authorization server issues an access token to a client and the client requests resources from a resource server sending the access token.

In OAuth, an authorization server and a resource server need neither be the same server nor be closely-coupled. This means that a resource server may have to validate access tokens without information that is implicitly passed from authorization servers. However, the OAuth 2.0 ([RFC6749]) framework does not define methods for resource servers to validate access tokens, leading to implementation specific non-interoperable validation methods.

To achieve better interoperability, this specification defines a format of structured access token that can be examined by the recipient. By using this format, the authorization server and the resource server can be easily separated and the validation of the access token can be performed offline.

The format of structured access tokens specified in this document is based on the JSON Web Token (JWT) format ([I-D.ietf-oauth-json-web-token]).

[[Note: this document should be harmonized with the jwt-bearer draft.]]

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].
2. Assumptions

In this specification, it is assumed that the account at the protected server is linked to the subject at the issuer in advance.

The exact method of how this account linking is done is out of scope of this document. A few possible ways includes the pre-existing common identifier between the protected resource and the issuer, and the explicit linking by the user providing the protected resource credential together with the ID Token from the issuer.
3. Structured Access Token Format

A structured access token will be a JWT ([I-D.ietf-oauth-json-web-token]) that contains claims defined in the following sections and other claims. The JWT MUST be signed unless otherwise the authenticity of the JWT is assured. The JWT MAY be encrypted by the issuer.

3.1. "iss" (Issuer) Claim

The "iss" (issuer) claim contains the identifier of the authorization server that issues the access token. Use of this claim is REQUIRED.

3.2. "sub" (Subject) Claim

The "sub" (subject) claim contains the identifier of the entity whose claims are provided in exchange for the access token. The value of this claims needs to be shared between the authorization server and the resource server. Use of this claim is REQUIRED.

3.3. "aud" (Audience) Claim

The "aud" (audience) claim contains the identifier of the resource server that receives the access token and returns claims. Use of this claim is REQUIRED.

3.4. "exp" (Expiration Time) Claim

The "exp" (expiration time) claim contains the expiration time on or after which the JWT MUST NOT be accepted. The value is an integer of seconds from 1970-01-01T00:00:00Z. Use of this claim is REQUIRED.

3.5. "nbf" (Not Before) Claim

The "nbf" (not before) claim contains the time before which the JWT MUST NOT be accepted. The value is an integer of seconds from 1970-01-01T00:00:00Z. Use of this claim is OPTIONAL.

3.6. "iat" (Issued At) Claim

The "iat" (issued at) claim contains the time at which the JWT was issued. The value is an integer of seconds from 1970-01-01T00:00:00Z. Use of this claim is OPTIONAL.

3.7. "claims" Claim

The "claims" claim contains the required claims about the entity identified by the "sub" claim. The value is an array of strings. Use
of this claim is OPTIONAL.

3.8. "azp" Claim

The "azp" (Authorized Presenter) claim contains the identifier of the party which is intended to use the access token and to request resources. Use of this claim is OPTIONAL.
4. Structured Access Token Usages

Structured access tokens can be used for accessing protected resources like end-users’ claims. The resource server and the authorization server need to share entities’ identifiers beforehand in order to validate access tokens and provide resources by the "sub" claim.

When a resource server received a structured access token, the resource server is RECOMMENDED to validate the token in the following criteria. Note that the validation criteria MAY vary by the contents of tokens.

- The "aud" claim MUST be the identifier of the resource server or the value that the resource server can accept as the audience of the token.
- The value of the "exp" claim MUST be more than the current time.
- If the "nbf" claim exists, the value MUST be less than or equal to the current time.
- If the "iat" claim exists, the value can be used to reject tokens that are issued at an unacceptable time (e.g., future or too far past). The acceptable range is dependent on the actual protocol and implementation used and is out of scope for this document.
- The token MUST be a valid JWT and signed by the authorization server identified by the "iss" claim.
- The "iss" claim MUST be an identifier of an acceptable authorization server. The acceptable values MAY vary by the "sub" claim or other claims.
- The "sub" claim MUST be a valid identifier of an entity. The value will be authorization-server specific, so the resource server needs to evaluate the "sub" claim together with the "iss" claim.
- The authorization server specified by the "iss" claim MUST be granted to manage authorization by the entity specified by the "sub" claim.
- If "azp" claim exists and the protected resource policy requires the identification of the client, the resource server MUST compare the "azp" and the client identifier. The exact method to achieve it is dependent on the actual protocol used and is out of scope for this specification.

After validating a structured access token, a resource server SHOULD provide resources specified by the "claims" claim or other claims.
5. Examples of the structured access token usage

5.1. UserInfo Request

When using a structured access token at UserInfo Endpoint of OpenID Connect, the "claims" claim will be used to specify requested claim names.

The following is a non-normative example of a JSON object (RFC4627) that could be encoded into a JWT (I-D.ietf-oauth-json-web-token) for a UserInfo request:

```json
{
  "aud":"https://op.example.com/",
  "iss":"https://op.example.com/",
  "sub":"alice",
  "exp":1360387733,
  "iat":1360386833,
  "nbf":1360385933,
  "claims": [
    "gender",
    "picture"
  ],
  "azp":"https://rp.example.com/
}
```

5.2. Aggregated/Distributed Claims in OpenID Connect

In OIDF.Connect.Standard, when using a structured access token for requesting claims to a claims provider in aggregated or distributed claims, the "claims" claim is used in the request. Typically the claims provider and the OpenID provider are separated parties, so they need to share the identifier of the entity specified by the "sub" claim beforehand. This may be done, for example, by the claims provider performing as a relying party to the OpenID provider.

The following is a non-normative example of a JSON object (RFC4627) that could be encoded into JWT:

```json
{
  "aud":"https://cp.example.com/",
  "iss":"https://op.example.com/",
  "sub":"82FB7D5C-C6F1-4505-B230-796F31DF683E",
  // PPID from the OpenID provider to the claims provider
  "exp":1360847733,
  "iat":1360846833,
  "nbf":1360845933,
  "claims": [
```
"gender",
"picture"
},
"azp":"https://rp.example.com/
}

5.3. Other Usages

This document specifies methods for relying parties to obtain claims using structured access tokens. Furthermore, the format may be used for access grant transfer between entities; for example, Alice grants Bob to download her picture and the authorization server issues an access token for Bob to access the protected picture. The protocols to issue and transfer access tokens are out of scope of this document.
6. Security Considerations

If an access token is a "Bearer" access token, the token MUST be sent over a protected channel such as TLS ([RFC5246]).

Resource servers MUST verify the "iss" claim in the access tokens and confirm that the issuer is trusted to issue access tokens for the subject. The process in which the issuer is trusted is out of scope of this document; several examples are described in the Assumption section.

Resource servers are responsible to protect resources from invalid access tokens. Resource servers MUST deny requests when requested claims contains unacceptable values. Resource servers SHOULD provide end-users with methods to discard trust on authorization servers.

7. Privacy Considerations

In the distributed claims model, the access token issued to the client might contain information that should not be disclosed to the client. To protect such information, JWE ([I-D.ietf-jose-json-web-encryption]) MAY be used to encrypt the contents of the token.

In the distributed claims model, resource servers might know which client is requesting claims via the "azp" claim. The "azp" claim is OPTIONAL and authorization servers MAY omit the claim.

8. IANA Considerations

<IANA considerations text>

9. References

9.1. Normative References


9.2. Informative References


Appendix A. Document History

-01 in 5
  o Renamed draft’s title.
  o Inserted Assumption section.
  o Inserted Other Usages section.
  o Inserted Security Considerations and Privacy Consideration sections.
  o Removed OIDC dependency from the normative text.

-00
  o Initial draft.

Authors’ Addresses

Nat Sakimura
Nomura Research Institute
EMail: n-sakimura@nri.co.jp

Boku Kihara (editor)
Lepidum Co. Ltd.
EMail: kihara@lepidum.co.jp

Kazuki Shimizu
Lepidum Co. Ltd.