OAuth Working Group
Internet-Draft
Intended status: Standards Track
Expires: 30 March 2026

JT. Turrado
FE. Escolar
Independent

26 September 2025

OAuth 2.0 External Assertion Authorization Grant
draft-external-assertion-oauth-grant-00

Abstract

This document specifies a new OAuth 2.0 authorization grant type, "external assertion", identified by urn:ietf:params:oauth:grant-type:external-assertion. It enables a client to obtain an access token by presenting a verifiable JWT assertion issued by a trusted external identity provider to an authorization server. The mechanism facilitates short-lived, auditable credentials for workloads without provisioning long-lived secrets.

Table of Contents

1. Conventions and Definitions

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.

Authorization Server (AS): The OAuth 2.0 server that issues access tokens.

External Identity Provider (External IdP): An OAuth 2.0/OpenID Connect compliant identity provider that issues JWT assertions trusted by the AS.

Assertion: A signed JWT bearing claims used by the AS to authenticate and authorize the client for token issuance.

2. Introduction

The OAuth 2.0 Authorization Framework ([RFC6749]) defines multiple grant types for obtaining access tokens. In many environments, workloads already possess identities issued by an external IdP. In order to interact with a resource server that relies on an AS different from the external IdP, a mechanism is needed to exchange the external identity for a locally-issued access token without provisioning long-lived secrets.

This document defines the "external assertion" grant type, identified by urn:ietf:params:oauth:grant-type:external-assertion, which allows a client to present a verifiable JWT assertion issued by a trusted external IdP to an AS in exchange for a short-lived access token. The grant is conceptually similar to token exchange ([RFC8693]) but is narrowly focused on a client-submitted external JWT assertion used as an authorization grant, and it does not define subject or actor token semantics beyond the validation rules herein.

3. Protocol Overview

At a high level, the client obtains a JWT assertion from an external IdP and submits it to the AS’s token endpoint using the grant type defined in this document. The AS validates the assertion according to its local trust configuration (e.g., trusted issuers and keys), and, if validation succeeds and policy permits, issues an access token to the client.

4. External Assertion Grant Type

4.1. Token Request

The client makes a request to the token endpoint with the content type application/x-www-form-urlencoded as defined in [RFC6749] Appendix B, including the following parameters:

grant_type: REQUIRED. Value MUST be urn:ietf:params:oauth:grant-type:external-assertion.

client_id: REQUIRED - unless client authentication is otherwise established by means outside of this request. The OAuth client identifier.

client_assertion: REQUIRED. A JWT assertion issued by a trusted external IdP. The assertion MUST be integrity-protected (e.g., JWS) and MUST contain the claims specified in Section 4.2.

scope: OPTIONAL. The scope of the access request as described in

Section 3.3 of [RFC6749].

Example request:

```
POST /token HTTP/1.1
Host: as.example.com
Content-Type: application/x-www-form-urlencoded

grant_type=urn:ietf:params:oauth:grant-type:external-assertion&
client_id=s6BhdRkqt3&
client_assertion=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9...
```

4.2. Assertion Validation

Upon receiving the request, the AS MUST perform all of the following validation steps:

Client Authorization Verify the client is recognized, authenticated as required by AS policy, and permitted to use this grant type (unauthorized_client if not).

Issuer Trust Ensure the assertion's iss value is configured as a trusted issuer. Trust configuration is out of scope; it may use local policy, static keys, or dynamic discovery (e.g., OpenID Provider configuration and JWKS).

Signature Verification Validate the JWS signature using keys associated with the trusted issuer (see [RFC7517] and [RFC7515]).

Expiration and Not-Before Enforce exp and, if present, nbf. The assertion MUST be unexpired at the time of processing.

Audience Ensure the assertion's aud identifies the AS (e.g., token endpoint or an AS audience value).

Subject Validate the sub according to AS policy for the trusted issuer (e.g., allowlists or mappings).

JWT ID (Replay) If a jti is present, the AS MUST prevent replay by rejecting previously seen jti values within the assertion's validity window. Use of jti is RECOMMENDED.

Issued-At (Freshness) If iat is present, the AS SHOULD enforce a maximum assertion age according to policy.

Scope and Policy Enforce that requested scopes are authorized for the client and permitted for the asserted identity.

The assertion MUST be a JWT ([RFC7519]) and MUST include the following claims:

iss Issuer identifier of the external IdP.

sub Subject identifier at the external IdP.

aud Audience for the AS (value defined by AS policy).

exp Expiration time.

iat Issued-at time (RECOMMENDED).

jti JWT ID for replay detection (RECOMMENDED).

The assertion SHOULD be signed with an algorithm acceptable to the AS (e.g., RS256 or ES256). Use of none MUST NOT be accepted. If the AS supports nested JWTs, additional confidentiality mechanisms are deployment-specific.

4.3. Token Response

If the request is valid and authorized, the AS issues an access token per Section 5.1 of [RFC6749]. A refresh token MUST NOT be issued for this grant type.

Example successful response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache

{
  "access_token": "2YotnFZFEjr1zCsicMWpAA",
  "token_type": "bearer",
  "expires_in": 3600
}
```

4.4. Error Response

On error, the AS returns an error response as defined in Section 5.2 of [RFC6749]. The following error codes are commonly applicable:

invalid_request Malformed or missing parameters.

unauthorized_client Client not authorized for this grant.

invalid_grant Assertion invalid, expired, audience mismatch, replayed, or issuer not trusted.

unsupported_grant_type Grant type not enabled at the AS.

5. Security Considerations

The security of this grant depends on the AS's trust configuration and assertion validation. Deployments MUST:

* Restrict trusted issuers and acceptable algorithms by policy.
* Validate signatures and enforce exp/nbf strictly.
* Bind assertions to the AS via aud; reject generic or missing audiences.
* Prevent replay via unique jti and server-side caching within validity windows.
* Issue short-lived access tokens and prefer least-privilege scopes.
* Use TLS for all endpoints and key retrieval.
* Log validation outcomes and consider rate limiting failed attempts.

If the AS includes an "actor" representation in issued tokens (e.g., the act claim from [RFC8693]), it SHOULD ensure that privacy and policy constraints are respected.

6. Privacy Considerations

Assertions may include identifiers that can correlate clients across services. Deployments SHOULD minimize personally identifiable information in assertions, avoid unnecessary claim propagation into issued tokens, and follow data minimization practices.

7. IANA Considerations

This document requests registration of the following value in the "OAuth URI" registry:

URN: urn:ietf:params:oauth:grant-type:external-assertion
Common Name: External Assertion Grant
Change Controller: IESG
Specification Document: This document

8. Normative References

[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>.

[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>.

[RFC7517] Jones, M., "JSON Web Key (JWK)", RFC 7517, DOI 10.17487/RFC7517, May 2015, <https://www.rfc-editor.org/info/rfc7517>.

[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>.

[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>.

[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>.

Appendix A. Acknowledgments

The authors thank the OAuth community for prior work on assertion-based flows and token exchange.

Authors' Addresses

Jorge Turrado
Independent
Email: jorge_turrado@hotmail.es

Fernando Escolar
Independent
Email: fer.escolar@gmail.com