Web Authorization Protocol
Internet-Draft
Intended status: Informational
Expires: 26 September 2026

A. Parecki
Okta
B. Campbell
Ping Identity
D. Liu
Alibaba Group

25 March 2026

JWT Authorization Grant Interaction Response
draft-parecki-oauth-jwt-grant-interaction-response-00

Abstract

This document defines an extension to the JWT Authorization Grant [RFC7523] that enables an authorization server to indicate that user interaction is required in order to complete an authorization request. Instead of returning an access token or an error, the authorization server returns a URI that the client launches where the user can interact with the authorization server, along with a polling interval. The client can then poll for the access token or wait for a redirect before retrying the original request.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://aaronpk.github.io/draft-parecki-oauth-jwt-grant-interaction-response/draft-parecki-oauth-jwt-grant-interaction-response.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-parecki-oauth-jwt-grant-interaction-response/.

Discussion of this document takes place on the Web Authorization Protocol Working Group mailing list (mailto:oauth@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/oauth/. Subscribe at https://www.ietf.org/mailman/listinfo/oauth/.

Source for this draft and an issue tracker can be found at https://github.com/aaronpk/draft-parecki-oauth-jwt-grant-interaction-response.

Status of This Memo

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

This Internet-Draft will expire on 26 September 2026.

Copyright Notice

Table of Contents

1. Introduction

The JWT Authorization Grant [RFC7523] and Identity Assertion Grant [I-D.draft-ietf-oauth-identity-assertion-authz-grant] enable clients to obtain access tokens without direct user approval at the authorization server. However, certain scenarios may require explicit user consent, even if the initial authorization can be obtained without user interaction:

* AI Agent Authorization: Autonomous agents acting on behalf of users need explicit consent for specific operations, not just identity verification.

* High-Risk Operations: Financial transactions, data deletion, or other sensitive operations may require step-up consent.

* Compliance Requirements: Regulatory frameworks (GDPR, etc.) may require explicit, verifiable consent for certain activities.

* Policy-Based Authorization: Fine-grained authorization policies (e.g., "allow purchases up to $50") require user understanding and approval.

Currently, if user interaction is needed, the authorization server has no standardized way to communicate this to the client within the JWT Authorization Grant flow. The client would typically receive an error response and would need to fall back to a traditional redirect-based OAuth flow, negating the benefits of using the JWT Authorization Grant.

This specification defines an interaction response that the authorization server can return in place of an access token. The interaction response contains a URI that the client opens (typically in a browser) so the user can interact with the authorization server. It also includes a polling interval, similar to the Device Authorization Grant [RFC8628], indicating how frequently the client should re-request the token.

The client's token request MAY include a redirect_uri parameter. If provided, the authorization server redirects the user's browser to this URI after the interaction is complete. Unlike the Authorization Code flow, no authorization code is included in the redirect. The redirect serves only as a signal to the client that the user interaction has completed. The client then retries its original JWT Authorization Grant request to obtain the access token.

2. 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.

3. Token Request

The client makes a token request to the authorization server's token endpoint as defined in [RFC7523], with the addition of an OPTIONAL redirect_uri parameter.

redirect_uri OPTIONAL. The URI to which the authorization server will redirect the user's browser after the interaction is complete. The redirect URI MUST be previously registered with the authorization server or otherwise validated per authorization server policy.

An example request:

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

grant_type=urn:ietf:params:oauth:grant-type:jwt-bearer
&assertion=eyJhbGciOi...
&redirect_uri=https://client.example.org/callback
```

4. Token Endpoint Responses

In addition to the error codes defined in Section 3.2.4 of [OAUTH-2.1], the following error codes are specified for use with the JWT Authorization Grant Interaction Response in token endpoint responses.

4.1. Interaction Required Response

When the authorization server receives a valid JWT Authorization Grant request but determines that user interaction is required before an access token can be issued, it responds with an OAuth error response as defined in Section 3.2.4 of [OAUTH-2.1] as defined below:

error REQUIRED. interaction_required. Indicates that user interaction is required, and the following additional parameters are defined in the response.

interaction_uri REQUIRED. The URI that the client MUST launch (typically in the user's browser) to allow the user to interact with the authorization server. The URI MUST use the "https" scheme.

interval OPTIONAL. The minimum number of seconds that the client SHOULD wait between polling requests to the token endpoint. If no value is provided, the default is 5 seconds.

expires_in OPTIONAL. The number of seconds after which the interaction URI and the associated authorization session will expire.

The response MUST include a Content-Type header field set to application/json.

```
HTTP/1.1 400 Bad Request
Content-Type: application/json

{
  "error": "interaction_required",
  "interaction_uri": "https://auth.example.com/interact/abc123",
  "interval": 5,
  "expires_in": 600
}
```

4.2. Interaction Pending Response

When user interaction is pending, and the client makes a subsequent token endpoint request, the authorization server responds with an OAuth error response as defined in Section 3.2.4 of [OAUTH-2.1] with one of the values defined below:

interaction_pending The authorization request is still pending as the end user hasn't yet completed the user-interaction steps. The client SHOULD repeat the access token request to the token endpoint. Before each new request, the client MUST wait at least

the number of seconds specified by the interval parameter defined in Section 4.1, or 5 seconds if none was provided, and respect any increase in the polling interval required by the "slow_down" error.

slow_down A variant of authorization_pending, the authorization request is still pending and polling should continue, but the interval MUST be increased by 5 seconds for this and all subsequent requests.

access_denied The authorization request was denied.

4.3. Handling the Interaction Response

Upon receiving an interaction response as defined in Section 4, the client MUST:

1. Launch or redirect a browser to the interaction_uri.

2. Wait for the interaction to complete using one or both of the following mechanisms:

   * *Polling:* The client re-sends its original token request (including the same JWT assertion and parameters) to the token endpoint, waiting at least interval seconds between each request. The authorization server will continue to return the interaction response until the user interaction is complete, at which point it will return an access token response or an error response as defined in Section 3.2 of [OAUTH-2.1].

   * *Redirect notification:* If the client included a redirect_uri in its original request, it MAY wait for the authorization server to redirect the user’s browser to that URI. Upon receiving the redirect, the client SHOULD immediately retry its original token request. No authorization code or other parameters are included in the redirect; the redirect serves solely as a signal that the interaction is complete.

5. Authorization Server Behavior

When the authorization server receives a JWT Authorization Grant request and determines that user interaction is required, it MUST:

1. Generate a unique interaction_uri for the user interaction session.

2. Associate the interaction session with the original token request parameters (including the JWT assertion claims, requested scope, any redirect_uri provided by the client, and parameters from any extensions).

3. Return the interaction response as defined in Section 4.1.

While the user interaction is pending, subsequent token requests from the client with the same JWT assertion SHOULD return the interaction response or an interaction_pending error.

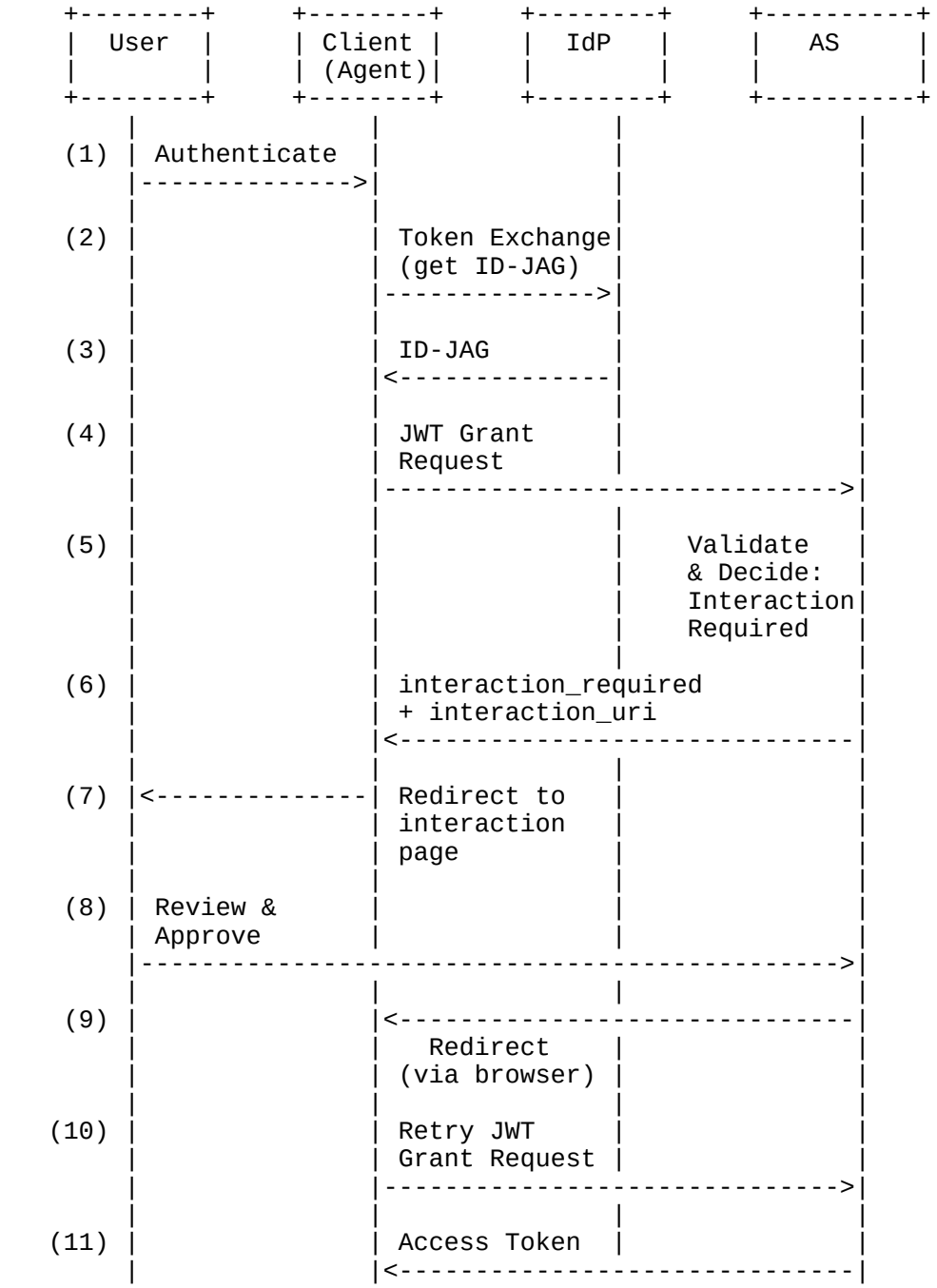
After the user has completed the required interaction, the authorization server MUST:

1. If a redirect_uri was provided, redirect the user’s browser to that URI. The redirect MUST NOT include an authorization code or access token.

2. On the next token request from the client with the same JWT assertion, return an access token response as defined in Section 3.2.3 of [OAUTH-2.1] or an error response as defined in Section 3.2.4 of [OAUTH-2.1].

6. Complete Flow Diagram

The diagram below is a non-normative example of using this specification in conjunction with the [[I-D.draft-ietf-oauth-identity-assertion-authz-grant]].


User
Client (Agent)
IdP
AS
(1) Authenticate
(2) Token Exchange (get ID-JAG)
(3) ID-JAG
(4) JWT Grant Request
(5) Validate & Decide: Interaction Required
(6) interaction_required + interaction_uri
(7) Redirect to interaction page
(8) Review & Approve
(9) Redirect (via browser)
(10) Retry JWT Grant Request
(11) Access Token

1. User authenticates to the Client through the IdP, typically via OpenID Connect

2. The Client exchanges the previously-obtained ID Token for an ID-JAG

3. The IdP validates the request against the configured policy and returns an ID-JAG

4. The Client presents the ID-JAG to the AS in a JWT Authorization Request

5. The AS validates the ID-JAG, and determines that further interaction is required

6. The AS returns the interaction_required response

7. The client redirects the browser to the specified page

8. The user visits the URL and confirms the request

9. The AS redirects to the Client’s redirect_uri

10. The Client retries the JWT Authorization Request

11. The AS validates the request, sees the user has confirmed, and issues an access token

7. Security Considerations

7.1. Interaction URI Security

The interaction_uri MUST be an "https" URI. The URI SHOULD contain sufficient entropy to prevent guessing or brute-force attacks. The authorization server SHOULD bind the interaction session to the client identity from the original token request.

7.2. Redirect URI Validation

If the client provides a redirect_uri, the authorization server MUST validate it against the client’s registered redirect URIs, consistent with Section 2.3.1 of [OAUTH-2.1].

8. IANA Considerations

TBD

9. References

9.1. Normative References

[OAUTH-2.1] Hardt, D., Parecki, A., and T. Lodderstedt, "The OAuth 2.1 Authorization Framework", Work in Progress, Internet-Draft, draft-ietf-oauth-v2-1-15, 2 March 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-oauth-v2-1-15>.

[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/rfc/rfc2119>.

[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/rfc/rfc7523>.

[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/rfc/rfc8174>.

9.2. Informative References

[I-D.draft-ietf-oauth-identity-assertion-authz-grant] Parecki, A., McGuinness, K., and B. Campbell, "Identity Assertion JWT Authorization Grant", Work in Progress, Internet-Draft, draft-ietf-oauth-identity-assertion-authz-grant-02, 2 March 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-oauth-identity-assertion-authz-grant-02>.

[RFC8628] Denniss, W., Bradley, J., Jones, M., and H. Tschofenig, "OAuth 2.0 Device Authorization Grant", RFC 8628, DOI 10.17487/RFC8628, August 2019, <https://www.rfc-editor.org/rfc/rfc8628>.

Appendix A. Appendix

A.1. Use Cases

A.1.1. AI Agent with Browser Access

An AI agent needs to perform operations on behalf of a user at a third-party service. The agent can redirect the user’s browser to a consent page, then receive a callback when consent is complete.

This differs from the Device Authorization Grant (RFC 8628) in that no polling is required - the redirect/callback model provides lower latency.

A.1.2. High-Risk Transaction Approval

A client presents a valid JWT assertion but requests authorization for a high-value financial transaction. The AS determines that step-up consent is required before issuing the token.

A.1.3. Regulatory Compliance

A client requests access to sensitive data. Regulatory requirements mandate that explicit, auditable consent must be obtained and recorded before access is granted.

A.2. Example Sequence

Acknowledgments

TODO acknowledge.

Authors' Addresses

Aaron Parecki
Okta
Email: aaron@parecki.com

Brian Campbell
Ping Identity
Email: bcampbell@pingidentity.com

Dapeng Liu
Alibaba Group
Email: max.ldp@alibaba-inc.com