Web Authorization Protocol
Internet-Draft
Intended status: Standards Track
Expires: 13 November 2026

G. Fletcher
Practical Identity LLC

12 May 2026

Transaction Token Authorization Grant Profile for OAuth Identity and Authorization Chaining
draft-fletcher-transaction-token-chaining-profile-00

Abstract

This specification defines a profile of the OAuth Identity and Authorization Chaining Across Domains [I-D.ietf-oauth-identity-chaining] mechanism that uses a Transaction Token (Txn-Token) [I-D.ietf-oauth-transaction-tokens] as the subject token in a Token Exchange [RFC8693] request to obtain a JWT Authorization Grant for crossing a trust boundary.

A Txn-Token is scoped to a single trust domain and represents the full authorization context of an in-progress transaction, regardless of whether that transaction was initiated by a human user calling an external API, by an internal system event, or by an automated workload. This profile specifies how a service operating within that trust domain can present its Txn-Token to obtain a JWT Authorization Grant that carries the necessary context across a trust boundary, enabling an access token to be issued for a partner service, without exposing internal trust-domain credentials or token formats beyond the trust boundary.

Note to Readers

RFC EDITOR: please remove this section before publication

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

Source for this draft and an issue tracker can be found at https://github.com/george-fletcher/draft-fletcher-transaction-token-chaining-profile (https://github.com/george-fletcher/draft-fletcher-transaction-token-chaining-profile).

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 13 November 2026.

Copyright Notice

Table of Contents

1. Introduction

Organizations routinely deploy services that, in fulfilling a transaction for a user or an automated process, must call one or more partner APIs that lie outside the organization’s own trust boundary. The challenge is to carry the authorization context of the original transaction — including the identity and authorization of the Initiating Principal — across that boundary in a way that is trustworthy to the partner, without leaking internal credentials or internal token formats.

Transaction Tokens (Txn-Tokens) [I-D.ietf-oauth-transaction-tokens] address the first half of this problem. A Txn-Token is a short-lived, cryptographically signed JWT scoped to a single trust domain (for example, an enterprise or a cloud service provider's internal environment). It is minted by a Transaction Token Service (TTS) at the point where a transaction enters the trust domain and captures, in immutable form, the identity of the initiating principal, the purpose of the transaction, and relevant request parameters. Every workload within the trust domain that handles the transaction receives and validates this Txn-Token, ensuring a consistent and authoritative authorization context throughout the internal call chain.

A Txn-Token may represent any of several originating contexts:

External User Request: A human user or external client calls an API exposed at the trust domain's perimeter (e.g., a financial services API that adds a stock to a watch list on behalf of the user, authenticated via an OAuth 2.0 access token). The TTS mints a Txn-Token anchored to the user's identity and the authorized scope of that external access token.

Internal System Event: An internal system triggers processing that has no direct external human caller (e.g., an SMTP server receiving an inbound message and initiating storage of that message in the recipient's mailbox). The TTS mints a Txn-Token representing the system's identity and the purpose of the transaction.

Automated Workload Request: One workload within the trust domain invokes another as part of an automated pipeline (e.g., a scheduled job triggering a data aggregation service). The Txn-Token represents the workload identity and the pipeline's authorization scope.

In all three cases, the Txn-Token provides a uniform, internal representation of the authorization context. The problem this specification addresses is what happens when a service within the trust domain, in the course of executing such a transaction, needs to call a service in a _different_ trust domain — a partner organization, a SaaS provider, or a third-party API — in order to complete the transaction.

Consider a mail service within an enterprise trust domain. Upon receiving an inbound message via SMTP, the mail service is issued a Txn-Token representing the mail delivery transaction on behalf of the recipient user. Before storing the message, the mail service must call a partner spam-rating API in the spam service's trust domain.

The mail service cannot present its internal Txn-Token to the spam service — the Txn-Token is scoped to the enterprise trust domain and carries internal context that must not be disclosed externally. Instead, the mail service must obtain a credential that is meaningful to the spam service's authorization server while preserving the relevant authorization context of the original transaction.

The OAuth Identity and Authorization Chaining Across Domains specification [I-D.ietf-oauth-identity-chaining] defines a general mechanism by which a client in Trust Domain A can obtain a JWT Authorization Grant from the Authorization Server of Trust Domain A and present it to the Authorization Server of Trust Domain B to receive an access token. The base specification deliberately leaves the choice of subject token type open, allowing profiles to constrain and specialize the mechanism for specific deployment scenarios.

This specification defines the additional details necessary to use a Txn-Token as the subject_token in the Token Exchange request described in Section 2.3 of [I-D.ietf-oauth-identity-chaining]. The Txn-Token is consumed by the Authorization Server of Trust Domain A, which validates it, applies claims transcription and minimization policy, and issues a JWT Authorization Grant targeted at the Authorization Server of Trust Domain B. The JWT Authorization Grant crosses the trust boundary carrying only the context that Trust Domain B is authorized to see. The Txn-Token itself never leaves Trust Domain A.

This profile is complementary to the Identity Assertion JWT Authorization Grant profile [I-D.ietf-oauth-identity-assertion-authz-grant], which targets deployments where the target authorization server already trusts a common IdP for SSO and subject resolution, using an OpenID Connect ID Token or SAML 2.0 assertion as the subject token. That profile is optimized for the human-user, single-sign-on scenario, where the trust relationship between AS-A and AS-B is mediated through a shared identity provider. This profile addresses scenarios where the trust relationship between AS-A and AS-B is established through a bilateral or federated Cross-Domain Trust Agreement, and where the input credential is a Txn-Token representing any authorized transaction within Trust Domain A.

A detailed structural comparison of the two profiles appears in Appendix B.

1.1. Requirements Language

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.

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.

2.1. Roles

The following roles are used in this document. They extend the OAuth 2.0 roles defined in [RFC6749] as used in [I-D.ietf-oauth-identity-chaining].

Initiating Principal: The entity whose authorization context is captured in the Txn-Token. The Initiating Principal may be a human user who made an external request to Trust Domain A, an internal system acting on its own behalf, or an automated workload operating within Trust Domain A. The Initiating Principal is not necessarily the same entity as the Requesting Workload that performs the cross-domain token exchange.

Requesting Workload: A service operating inside Trust Domain A that, in the course of processing a transaction, needs to call a Protected Resource in Trust Domain B. The Requesting Workload holds a Txn-Token representing the current transaction context and acts as the OAuth 2.0 client in the Token Exchange flow with AS-A.

Transaction Token Service (TTS): The service within Trust Domain A that mints and signs Txn-Tokens. The TTS is the authoritative source of transaction authorization context within Trust Domain A. In some deployments the TTS and AS-A MAY be co-located; in others they are separate services within the same Trust Domain.

Authorization Server of Trust Domain A (AS-A): The OAuth 2.0 Authorization Server within Trust Domain A that receives the Token Exchange request from the Requesting Workload, validates the presented Txn-Token, applies claims transcription and minimization policy, and issues the JWT Authorization Grant targeted at AS-B.

Authorization Server of Trust Domain B (AS-B): The OAuth 2.0 Authorization Server within Trust Domain B that receives the JWT Authorization Grant from the Requesting Workload and issues an access token for the Protected Resource.

Protected Resource: The resource server in Trust Domain B that the Requesting Workload needs to call in order to complete the transaction in progress in Trust Domain A.

2.2. Terms

Transaction: A unit of work initiated by an Initiating Principal that may span multiple workloads within Trust Domain A and that has a single, coherent authorization context. A transaction is identified by the txn claim in the Txn-Token.

Trust Domain: A deployment-specific security and administrative boundary within which services, identifiers, credentials, and policy decisions are mutually trusted. This term is used in [I-D.ietf-oauth-identity-chaining] without a formal definition; this profile formalizes it. Txn-Tokens are scoped to a single Trust Domain. In this specification, Trust Domain A is the Trust Domain in which the transaction originates and in which the Requesting Workload operates. Trust Domain B is the Trust Domain in which the Protected Resource and AS-B operate.

Cross-Domain Trust Agreement: A bilateral or federated configuration through which AS-A and AS-B establish mutual trust, permitting AS-A to issue JWT Authorization Grants that AS-B will accept, and defining the subject identifier mappings, permitted claims, and authorization policy that apply to cross-domain requests. The mechanism for establishing this trust is out of scope for this specification, but MUST be established prior to any cross-domain token exchange under this profile.

3. Overview

3.1. Transaction Token Context Within a Trust Domain

A transaction enters Trust Domain A at its perimeter. The initiating event may be:

(a) *An inbound API call from an external client*, in which case the external client presents an OAuth 2.0 access token or similar credential at the trust domain’s API gateway;

(b) *An internal system event*, such as an SMTP server receiving an inbound message, where the triggering input arrives from outside the enterprise boundary; or

(c) *An automated workload trigger*, with no direct external caller, such as a scheduled job or an event-driven pipeline invocation.

In all cases, the workload that first handles the transaction requests a Txn-Token from the TTS, presenting whatever inbound credential or context is available. The TTS validates the inbound context and mints a Txn-Token that captures the Initiating Principal’s identity (which may be a user identity, a system identity, or a workload identity), the purpose of the transaction (scope), and relevant request parameters (rctx). The Txn-Token is propagated to all downstream workloads within Trust Domain A that participate in processing the transaction.

3.2. Cross-Domain Invocation

When a Requesting Workload within Trust Domain A determines that it needs to call a Protected Resource in Trust Domain B in order to complete the transaction, it follows the flow defined in this profile. The complete end-to-end sequence is illustrated in Figure 1.

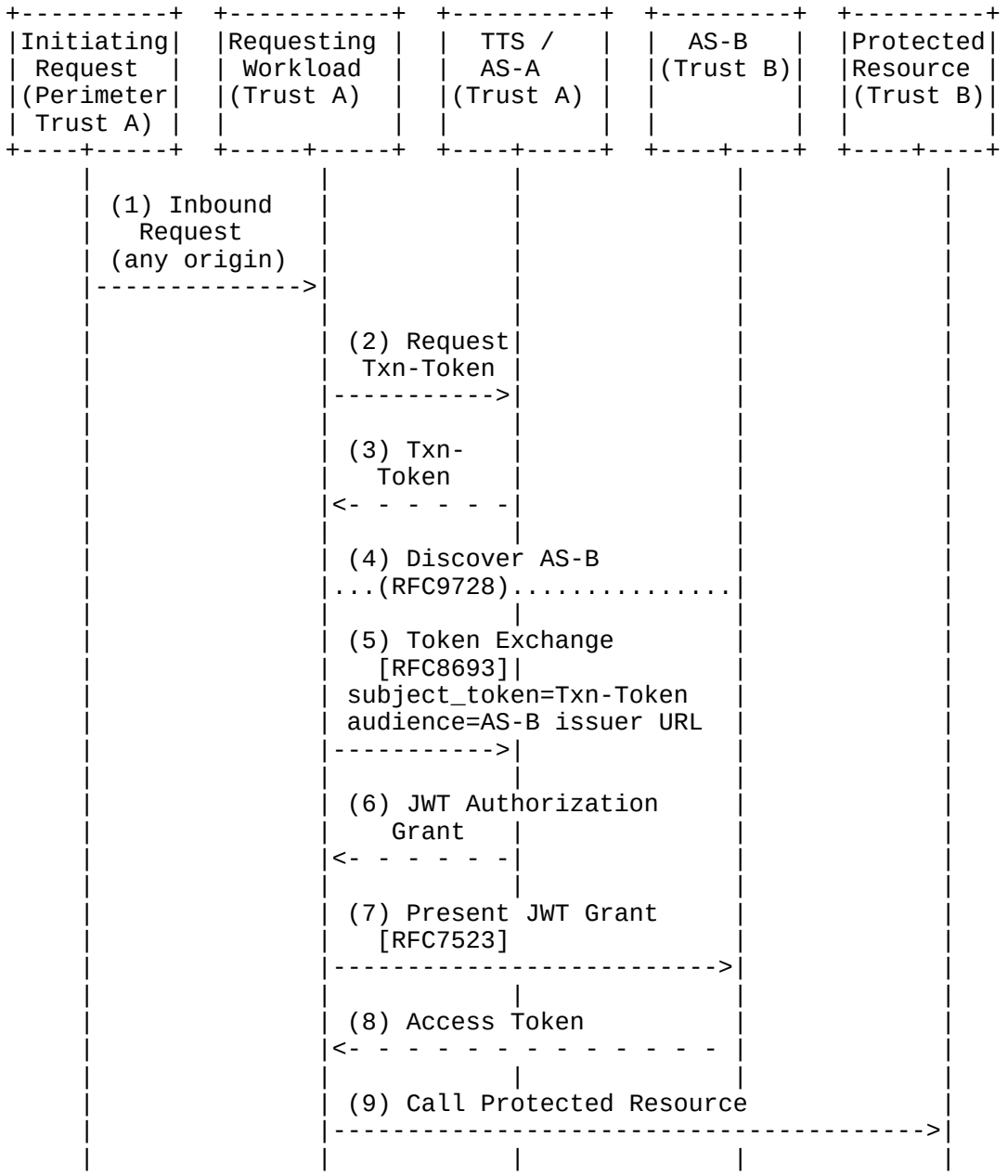

Figure 1: Transaction Token Chaining Flow

The steps are as follows:

1. An inbound request arrives at the Requesting Workload's perimeter. This may be an OAuth 2.0-protected API call from an external user or client, an SMTP message delivery, a scheduled job trigger, or any other initiating event.

2. The Requesting Workload (or the first workload within Trust Domain A that receives the transaction) requests a Txn-Token from the TTS, presenting the available inbound context (e.g., the external OAuth 2.0 access token, an internal system credential, or a workload identity), following the Txn-Token issuance procedures defined in [I-D.ietf-oauth-transaction-tokens]. The TTS records the Initiating Principal's identity in the sub claim and the transaction purpose in the scope claim of the Txn-Token.

3. The TTS issues a Txn-Token to the Requesting Workload. The Txn-Token is scoped to Trust Domain A and MUST NOT be presented to any entity outside Trust Domain A.

4. The Requesting Workload discovers AS-B using the mechanisms defined in Section 2.2 of [I-D.ietf-oauth-identity-chaining], such as the authorization_servers metadata property in the Protected Resource Metadata [RFC9728] published by the Protected Resource. The Requesting Workload obtains AS-B's issuer URL for use as the audience parameter in the Token Exchange request.

5. The Requesting Workload presents the Txn-Token as the subject_token in an OAuth 2.0 Token Exchange [RFC8693] request to AS-A, identifying AS-B in the audience parameter and optionally specifying the target Protected Resource in the resource parameter. See Section 4.3 for the full parameter specification.

6. AS-A validates the Txn-Token, verifies that a Cross-Domain Trust Agreement exists with the indicated AS-B, applies subject identifier mapping (Section 7.3) and claims minimization (Section 7), and issues a signed JWT Authorization Grant. The Txn-Token is consumed entirely within Trust Domain A and is not forwarded.

7. The Requesting Workload presents the JWT Authorization Grant to AS-B using the JWT Profile for OAuth 2.0 Authorization Grants [RFC7523].

8. AS-B validates the JWT Authorization Grant and issues an access token for the Protected Resource.

9. The Requesting Workload calls the Protected Resource with the access token, completing the cross-domain portion of the transaction.

4. Transaction Token as Subject Token

4.1. Subject Token Requirements

When this profile is used, the subject_token in the Token Exchange request (Step 5 of Figure 1) MUST be a Txn-Token as defined in [I-D.ietf-oauth-transaction-tokens].

The subject_token_type parameter MUST be:

```
subject_token_type =
    "urn:ietf:params:oauth:token-type:txn_token"
```

This value is defined in [I-D.ietf-oauth-transaction-tokens].

The Txn-Token presented as the subject_token MUST satisfy all of the validity requirements specified in [I-D.ietf-oauth-transaction-tokens], including:

* The Txn-Token MUST NOT be expired.

* The Txn-Token MUST be signed and verifiable by AS-A using keys published by the TTS.

* The Txn-Token's aud claim MUST identify AS-A (or a value that AS-A accepts as a valid audience for presented subject tokens).

A Txn-Token failing any of the above checks MUST be rejected per Section 2.2.2 of [RFC8693].

4.2. Txn-Token Initiating Principal Context

The Txn-Token's sub claim identifies the Initiating Principal of the transaction. The Initiating Principal type is not constrained by this profile; a Txn-Token may represent any originating context defined by the Transaction Token specification [I-D.ietf-oauth-transaction-tokens]. The following are common examples:

Human User Identity: The sub claim identifies a human user whose identity was established when the transaction entered Trust Domain A via an OAuth 2.0-protected API call. In this case the sub value is typically derived from the user's identity in the external access token presented at the API gateway, and the Txn-Token's rctx claim captures relevant attributes of the external request (such as the OAuth client identifier and originating IP address).

System Identity: The sub claim identifies an internal system

component (such as an SMTP server or a messaging gateway) acting in its own right, with no external user as the Initiating Principal. The scope claim is particularly significant in this case, as it conveys the reason for the transaction in the absence of a user-facing authorization context.

Workload Identity: The sub claim identifies an automated workload (such as a scheduled job or pipeline service). Workload identifiers MAY take the form of SPIFFE URIs [I-D.ietf-wimse-arch] when WIMSE-compatible infrastructure is in use within Trust Domain A.

The above examples are illustrative; other Initiating Principal types are possible. The claims transcription rules in Section 7 and the subject identifier mapping rules in Section 7.3 apply regardless of which Initiating Principal type the Txn-Token represents. AS-A MUST map the sub claim to an identifier appropriate for Trust Domain B, applying the mapping logic defined in the Cross-Domain Trust Agreement for the Initiating Principal type in question.

4.3. Token Exchange Request Parameters

In addition to the subject token requirements in Section 4.1, the Token Exchange request ([RFC8693] Section 2.1) MUST include the following parameters when this profile is in use.

4.3.1. Identifying the Target Authorization Server and Resource

This profile uses the audience and resource parameters following the convention in [I-D.ietf-oauth-identity-assertion-authz-grant] Section 4.3. The two parameters serve distinct purposes and MUST NOT be conflated.

audience: REQUIRED. The issuer identifier of AS-B ([RFC8414] Section 2). Becomes the aud claim of the JWT Authorization Grant. Implementations MUST use this parameter to identify AS-B and MUST NOT pass the AS-B issuer URL as resource.

resource: OPTIONAL. A URI identifying the Protected Resource (resource server) in Trust Domain B, as defined in [RFC8707] Section 2. When present, AS-A SHOULD propagate this value into the resource claim of the JWT Authorization Grant.

4.3.2. Remaining Parameters

grant_type: REQUIRED. The value MUST be urn:ietf:params:oauth:grant-type:token-exchange.

subject_token: REQUIRED. The Txn-Token as described in Section 4.1.

subject_token_type: REQUIRED. The value MUST be urn:ietf:params:oauth:token-type:txn_token.

requested_token_type: OPTIONAL. When present, the value MUST be urn:ietf:params:oauth:token-type:jwt. If absent, AS-A MUST still produce a JWT Authorization Grant conforming to this profile when the other parameters conform to this profile.

scope: OPTIONAL. Space-separated list of scopes requested for the JWT Authorization Grant. AS-A MUST NOT issue a grant with scope exceeding the scope claim of the presented Txn-Token (see Section 7).

The actor_token and actor_token_type parameters defined in [RFC8693] are not used in this profile.

4.3.3. Example Token Exchange Request

The following is a non-normative example conforming to this profile. A mail service workload in an enterprise (Trust Domain A) has received an SMTP message and holds a Txn-Token representing a mail-delivery transaction. The mail service needs to call a spam-rating API operated by a partner spam service whose Authorization Server is https://as.spamsvc.example and whose spam-rating API is https://api.spamsvc.example/spam-rating.

```
POST /token HTTP/1.1
Host: as.enterprise.example
Content-Type: application/x-www-form-urlencoded
Authorization: Bearer <mail-service-client-credential>

grant_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Atoken-exchange
&subject_token=<txn-token>
&subject_token_type=urn%3Aietf%3Aparams%3Aoauth%3Atoken-type%3Atxn_token
&audience=https%3A%2F%2Fas.spamsvc.example
&resource=https%3A%2F%2Fapi.spamsvc.example%2Fspam-rating
&scope=spam.rating.read
```

4.3.4. Token Exchange Response

If the request is valid and the Requesting Workload is authorized to receive a JWT Authorization Grant for the indicated audience, AS-A returns a Token Exchange response as defined in Section 2.2 of [RFC8693].

access_token: REQUIRED. The JWT Authorization Grant. (Token

Exchange uses the access_token field for the returned token for historical compatibility reasons; this is not an OAuth access token.)

issued_token_type: REQUIRED. The value MUST be urn:ietf:params:oauth:token-type:jwt.

token_type: REQUIRED. The value MUST be N_A.

expires_in: RECOMMENDED. The lifetime of the JWT Authorization Grant in seconds. This value SHOULD reflect the exp claim of the returned grant JWT and SHOULD be short (see Section 6.1.2).

refresh_token: This parameter SHOULD NOT be present.

On error, AS-A returns an error response as defined in Section 5.2 of [RFC6749] and Section 2.2.2 of [RFC8693].

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

{
  "access_token": "eyJ...<JWT Authorization Grant>...",
  "issued_token_type": "urn:ietf:params:oauth:token-type:jwt",
  "token_type": "N_A",
  "expires_in": 60
}
```

5. Processing Rules

5.1. AS-A Processing Rules

Upon receipt of a Token Exchange request conforming to this profile, AS-A MUST perform the following steps:

1. Authenticate the client (Requesting Workload) using the mechanisms specified in Section 5.1 of [I-D.ietf-oauth-identity-chaining] and Section 2.5 of [RFC9700].

2. Validate the Txn-Token signature using the public keys of the TTS that issued it. AS-A MUST be configured with the TTS's jwks_uri or equivalent key material.

3. Validate that the Txn-Token is not expired.

4. Validate that the aud claim of the Txn-Token identifies AS-A or a value AS-A is configured to accept as a valid audience for presented subject tokens.

5. Verify that the audience value identifies a known AS-B for which a Cross-Domain Trust Agreement has been established. AS-A MUST NOT accept a resource server URI in audience in place of an AS-B issuer identifier. If the audience is unknown or disallowed by policy, AS-A MUST return an error per Section 2.2.2 of [RFC8693].

6. If the resource parameter is present, validate that it identifies a Protected Resource within Trust Domain B consistent with the indicated AS-B. AS-A SHOULD propagate the resource value into the resource claim of the JWT Authorization Grant.

7. Validate that the requested scope, if present, does not exceed the scope claim of the Txn-Token. AS-A MUST NOT issue a JWT Authorization Grant with broader scope than the Txn-Token asserts.

8. Determine the Initiating Principal type from the Txn-Token and apply the appropriate subject identifier mapping as described in Section 7.3. If no mapping can be determined, AS-A MUST return an error.

9. Apply claims transcription and minimization policy as described in Section 7.

10. Construct and sign the JWT Authorization Grant as described in Section 6, setting the aud claim to the AS-B issuer identifier resolved in step 5.

11. Return the JWT Authorization Grant in the Token Exchange response as described in Section 4.3.

5.2. AS-B Processing Rules

Upon receipt of a JWT Bearer grant request ([RFC7523]) conforming to this profile, AS-B MUST perform the following steps in addition to the processing rules specified in Section 2.4.2 of [I-D.ietf-oauth-identity-chaining]:

1. Validate the typ header of the JWT Authorization Grant. The value MUST be txn-chain+jwt as defined in Section 6.

2. Validate that the aud claim matches AS-B’s own issuer identifier.

3. Validate that the iss claim identifies an AS-A with which a Cross-Domain Trust Agreement has been established, and validate the JWT signature using the public keys advertised by that AS-A.

4. Validate that the JWT is not expired and that the jti value has not been previously presented (single-use enforcement).

5. Resolve the subject from the sub claim according to the mapping rules defined in the Cross-Domain Trust Agreement. AS-B SHOULD evaluate the sub claim against its configured cross-domain access policy; supplementary identifiers in txn_claims (e.g., email) MAY also be used for subject resolution where the Cross-Domain Trust Agreement permits. If subject resolution fails and the Cross-Domain Trust Agreement does not permit Just-In-Time provisioning, AS-B MUST return an error.

6. If present, evaluate the txn_claims claim to apply context-aware authorization policy (see Section 7), for example verifying that the scope value is consistent with the requested scope.

7. Issue an access token constrained by the scope and, if present, the resource claim in the JWT Authorization Grant. AS-B SHOULD NOT issue refresh tokens, consistent with Section 5.4 of [I-D.ietf-oauth-identity-chaining].

6. JWT Authorization Grant

6.1. Grant Format

The JWT Authorization Grant produced by AS-A in response to a Token Exchange request conforming to this profile is a JWT [RFC7519] that MUST conform to the JWT Authorization Grant requirements specified in Section 2.3.3 of [I-D.ietf-oauth-identity-chaining].

6.1.1. JWT Header

typ: REQUIRED. The value MUST be txn-chain+jwt ([RFC8725]).

alg: REQUIRED. An asymmetric signing algorithm. Deployments SHOULD use PS256, PS384, PS512, ES256, ES384, or ES512 as defined in [RFC7519]. The none algorithm and symmetric algorithms are prohibited.

kid: RECOMMENDED. The key identifier corresponding to the signing key.

6.1.2. JWT Claims Requirements

The following claims MUST be present:

iss: REQUIRED. The Issuer identifier of AS-A ([RFC8414] Section 2).

sub: REQUIRED. The Initiating Principal's identity as mapped by AS-A according to Section 7.3. The value MUST be meaningful to AS-B within the context of the Cross-Domain Trust Agreement.

aud: REQUIRED. The Issuer URL of AS-B, derived from the audience parameter of the Token Exchange request. MUST be a single value to prevent grant replay at an unintended authorization server.

iat: REQUIRED. Issuance time ([RFC7519] Section 4.1.6).

exp: REQUIRED. Expiration time ([RFC7519] Section 4.1.4). The lifetime SHOULD be short. Deployments SHOULD use a value no greater than 300 seconds and SHOULD prefer values of 60 seconds or less, consistent with the short-lived nature of Txn-Tokens.

jti: REQUIRED. A unique identifier for this JWT ([RFC7519] Section 4.1.7). AS-B MUST enforce single-use semantics by tracking presented jti values within the grant's validity window.

scope: RECOMMENDED. The authorized scope ([RFC6749] Section 3.3). MUST NOT be wider than the scope claim of the source Txn-Token.

The following claims SHOULD be present:

txn: The unique transaction identifier from the originating Txn-Token ([I-D.ietf-oauth-transaction-tokens]). AS-B SHOULD record this value in its audit logs.

The following claims MAY be present:

resource: A URI or array of URIs identifying the Protected Resource(s) in Trust Domain B ([RFC8707] Section 2), derived from the resource parameter of the Token Exchange request. AS-B SHOULD use this to issue a resource-bound access token.

txn_claims: A JSON object containing a curated subset of Txn-Token claims, selected and minimized per the policy in Section 7. AS-B MAY use these claims for context-aware authorization decisions.

cnf: If sender-constraining is in use (see Section 9.2), the confirmation method claim conveying the Requesting Workload's public key, as defined in [RFC7800].

6.1.3. Example JWT Authorization Grant

The following is a non-normative example corresponding to the mail service scenario in Section 4.3. The Initiating Principal is the mail service's system identity (mail-gateway@enterprise.example) and the Txn-Token's rctx carries the SMTP envelope sender. The scope and a minimized rctx are transcribed into txn_claims.

Header:

```
{
  "typ": "txn-chain+jwt",
  "alg": "ES256",
  "kid": "as-enterprise-2026-01"
}
```

Claims:

```
{
  "iss": "https://as.enterprise.example",
  "sub": "mail-gateway@enterprise.example",
  "aud": "https://as.spamsvc.example",
  "iat": 1746700000,
  "exp": 1746700060,
  "jti": "8f14e45f-ceee-467a-a19e-ab8f290a1f30",
  "scope": "spam.rating.read",
  "resource": "https://api.spamsvc.example/spam-rating",
  "txn": "a9b2c3d4-e5f6-7890-abcd-ef1234567890",
  "txn_claims": {
    "scope": "mail-delivery",
    "rctx": {
      "smtp_from": "sender@external.example"
    }
  }
}
```

The aud claim (https://as.spamsvc.example) identifies the authorization server of Trust Domain B. The resource claim (https://api.spamsvc.example/spam-rating) identifies the specific API endpoint. These are distinct values serving distinct purposes.

7. Claims Transcription

This profile constrains and extends the claims transcription rules of Section 2.5 of [I-D.ietf-oauth-identity-chaining] as follows.

7.1. Mandatory Transcriptions

AS-A MUST derive the sub claim of the JWT Authorization Grant from the sub claim of the Txn-Token, applying the subject identifier mapping defined in Section 7.3.

AS-A MUST include the txn claim from the Txn-Token as the txn claim in the JWT Authorization Grant, preserving the transaction correlation identifier across the domain boundary.

7.2. Constrained Scope Transcription

The scope in the JWT Authorization Grant MUST be the intersection of the Txn-Token's scope claim and the scope parameter of the Token Exchange request (if present). AS-A MUST NOT expand scope beyond the Txn-Token's scope under any circumstances.

7.3. Subject Identifier Mapping

The sub claim of the Txn-Token identifies the Initiating Principal within Trust Domain A's namespace. AS-A MUST translate this identifier to a form that is both meaningful and authorized for use in Trust Domain B, according to the mapping rules defined in the Cross-Domain Trust Agreement. The Cross-Domain Trust Agreement MUST define mapping rules for every Initiating Principal type that may appear in Txn-Tokens exchanged under this profile. If no mapping can be determined for the Initiating Principal presented, AS-A MUST deny the Token Exchange request.

7.4. Claims Minimization

Txn-Tokens MUST NOT be forwarded across trust boundaries. The JWT Authorization Grant is the only artifact that crosses the boundary, and AS-A MUST apply strict claims minimization.

The optional txn_claims object in the JWT Authorization Grant MAY carry a curated subset of Txn-Token claims that are relevant to AS-B's authorization policy. AS-A MUST apply the following minimization rules:

Purpose Claim (scope): SHOULD be included when it is meaningful to AS-B's authorization policy (e.g., to enable the Protected Resource to apply different handling based on transaction type).

Requester Context (rctx): MAY be included in a minimized form. Information relevant to the cross-domain request (e.g., the originating client IP address for a user-initiated transaction, or the SMTP envelope sender address for a mail delivery transaction)

MAY be included. Internal network addresses, intermediate workload identifiers, and internal infrastructure topology details MUST be omitted.

Internal Call Chain: Claims that record intermediate workloads or the internal call chain within Trust Domain A MUST NOT be included in txn_claims.

Supplementary Identity Claims: For human user Initiating Principals, claims such as email MAY be included in txn_claims if the Cross-Domain Trust Agreement explicitly permits their disclosure and AS-B requires them for subject resolution.

The Cross-Domain Trust Agreement SHOULD define the set of claims permitted to appear in txn_claims and their expected semantics, to ensure that AS-A and AS-B have a shared, normative understanding of each transcribed claim.

8. Authorization Server Metadata

This profile adds to the Authorization Server Metadata framework defined in [RFC8414] and Section 3 of [I-D.ietf-oauth-identity-chaining].

An Authorization Server that supports this profile MUST include the value urn:ietf:params:oauth:token-type:txn_token in its identity_chaining_requested_token_types_supported metadata parameter.

9. Security Considerations

9.1. Client Authentication

The Requesting Workload MUST authenticate to AS-A when performing the Token Exchange request. The use of asymmetric key-based client authentication (e.g., a JWT client assertion per [RFC7523]) is RECOMMENDED. Static shared secrets SHOULD NOT be used. AS-A SHOULD follow the client authentication guidance in Section 2.5 of [RFC9700].

9.2. Sender Constraining Tokens

AS-B SHOULD issue sender-constrained access tokens. Both DPoP (OAuth 2.0 Demonstrating Proof of Possession) and Mutual-TLS ([RFC9700] Section 2.3) are RECOMMENDED mechanisms.

When AS-A acts as the client toward AS-B (the authorization-server-as-client topology described in Appendix B.2 of [I-D.ietf-oauth-identity-chaining]), the delegated key binding

mechanism described in Appendix B.3 of that document SHOULD be used. AS-A MUST verify proof of possession of the Requesting Workload's key and convey it to AS-B using the cnf claim in the JWT Authorization Grant.

9.3. Txn-Token Confidentiality

A Txn-Token MUST NOT be forwarded to any entity outside Trust Domain A. All communication between the Requesting Workload and AS-A MUST be encrypted and the Requesting Workload MUST be authenticated (e.g., via mutual TLS; see also Section 9.1). Txn-Token lifetimes SHOULD be short.

9.4. JWT Authorization Grant Replay Prevention

The JWT Authorization Grant is a bearer token. AS-B MUST enforce single-use semantics on the jti claim. AS-A SHOULD set a short validity lifetime (see Section 6.1.2). Additional guidance is provided in Section 5.5 of [I-D.ietf-oauth-identity-chaining].

9.5. Scope Boundary Enforcement

AS-A MUST enforce that the JWT Authorization Grant scope does not exceed the Txn-Token's scope. AS-B MUST independently enforce that the access token it issues does not convey scope exceeding the JWT Authorization Grant. These controls together prevent the chaining mechanism from being used to escalate privileges beyond the originating transaction's authorized scope.

9.6. Cross-Domain Trust Agreement Integrity

Operators MUST ensure that:

* AS-A issues JWT Authorization Grants only for AS-B instances with which a bilateral Cross-Domain Trust Agreement has been explicitly established and is actively maintained.

* AS-B accepts JWT Authorization Grants only from AS-A instances listed in its trusted issuers configuration.

* The Cross-Domain Trust Agreement, including subject identifier mappings and permitted txn_claims, is reviewed whenever the participating services or their authorization policies change.

9.7. Refresh Tokens

AS-B SHOULD NOT issue refresh tokens. Because Txn-Tokens are short-lived and transaction-specific, re-obtaining a new Txn-Token and repeating the chaining flow is the correct renewal mechanism. Issuing a refresh token would decouple the access lifetime from the originating transaction's authorization context and create a persistent credential outside the control of Trust Domain A.

10. Privacy Considerations

Txn-Tokens may contain claims that relate to the Initiating Principal, including personal identity information for human-user-initiated transactions (e.g., user identifier, email address, IP address) that may be subject to applicable privacy regulations.

AS-A MUST apply claims minimization (Section 7) before issuing a JWT Authorization Grant. Specifically:

* Only identity claims necessary for AS-B to resolve the subject and apply authorization policy SHOULD be included in txn_claims.

* Claims that could be used to reconstruct internal activity patterns within Trust Domain A MUST NOT be included.

* The Cross-Domain Trust Agreement MUST specify which identity claims AS-A is permitted to disclose to AS-B, consistent with the data handling and privacy policies of both organizations.

The txn claim enables end-to-end transaction correlation across the domain boundary. Operators SHOULD evaluate whether the auditability benefits outweigh the privacy implications for their specific deployment, particularly for human-user-initiated transactions.

11. IANA Considerations

11.1. JWT Typ Registration

This specification requests registration of the following value in the "JSON Web Signature and Encryption Header Parameters" registry (maintained by IANA):

* Header Parameter Name: txn-chain+jwt

* Header Parameter Description: JWT type for a Transaction Token Chaining Authorization Grant as defined in this document

* Change Controller: IETF

* Specification Document(s): Section 6 of this document

11.2. JWT Claims Registry

This specification requests registration of the following claim name in the "JSON Web Token Claims" registry (maintained by IANA):

* Claim Name: txn_claims

* Claim Description: Transcribed claims from a Transaction Token, included in a JWT Authorization Grant to convey cross-domain authorization context

* Change Controller: IETF

* Specification Document(s): Section 7 of this document

12. References

12.1. Normative References

[I-D.ietf-oauth-identity-chaining]
Schwenkschuster, A., Kasselman, P., Burgin, K., Jenkins, M., Campbell, B., and A. Parecki, "OAuth Identity and Authorization Chaining Across Domains", Work in Progress, Internet-Draft, draft-ietf-oauth-identity-chaining-11, 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-oauth-identity-chaining-11>.

[I-D.ietf-oauth-transaction-tokens]
Tulshibagwale, A., Fletcher, G., and P. Kasselman, "Transaction Tokens", Work in Progress, Internet-Draft, draft-ietf-oauth-transaction-tokens-08, 2026, <https://datatracker.ietf.org/doc/draft-ietf-oauth-transaction-tokens/>.

[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>.

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

[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/rfc/rfc7519>.

[RFC7521] Campbell, B., Mortimore, C., Jones, M., and Y. Goland, "Assertion Framework for OAuth 2.0 Client Authentication and Authorization Grants", RFC 7521, DOI 10.17487/RFC7521, May 2015, <https://www.rfc-editor.org/rfc/rfc7521>.

[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>.

[RFC7800] Jones, M., Bradley, J., and H. Tschofenig, "Proof-of-Possession Key Semantics for JSON Web Tokens (JWTs)", RFC 7800, DOI 10.17487/RFC7800, April 2016, <https://www.rfc-editor.org/rfc/rfc7800>.

[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>.

[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/rfc/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/rfc/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/rfc/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/rfc/rfc8725>.

[RFC9700] Lodderstedt, T., Bradley, J., Labunets, A., and D. Fett, "Best Current Practice for OAuth 2.0 Security", BCP 240, RFC 9700, DOI 10.17487/RFC9700, January 2025, <https://www.rfc-editor.org/rfc/rfc9700>.

[RFC9728] Jones, M.B., Hunt, P., and A. Parecki, "OAuth 2.0 Protected Resource Metadata", RFC 9728, DOI 10.17487/RFC9728, April 2025, <https://www.rfc-editor.org/rfc/rfc9728>.

12.2. Informative References

[I-D.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-03, 2026, <https://datatracker.ietf.org/doc/draft-ietf-oauth-identity-assertion-authz-grant/>.

[I-D.ietf-wimse-arch]
Salowey, J., Rosomakho, Y., and H. Tschofenig, "Workload Identity in a Multi System Environment (WIMSE) Architecture", Work in Progress, Internet-Draft, draft-ietf-wimse-arch-07, 2026, <https://datatracker.ietf.org/doc/draft-ietf-wimse-arch/>.

[I-D.ietf-wimse-workload-creds]
Campbell, B., Salowey, J., Schwenkschuster, A., Sheffer, Y., and Y. Rosomakho, "WIMSE Workload Credentials", Work in Progress, Internet-Draft, draft-ietf-wimse-workload-creds-00, 2025, <https://datatracker.ietf.org/doc/draft-ietf-wimse-workload-creds/>.

[RFC9068] Bertocci, V., "JSON Web Token (JWT) Profile for OAuth 2.0 Access Tokens", RFC 9068, DOI 10.17487/RFC9068, October 2021, <https://www.rfc-editor.org/rfc/rfc9068>.

[RFC9396] Lodderstedt, T., Richer, J., and B. Campbell, "OAuth 2.0 Rich Authorization Requests", RFC 9396, DOI 10.17487/RFC9396, May 2023, <https://www.rfc-editor.org/rfc/rfc9396>.

Appendix A. Use Cases

The following use cases illustrate the three Initiating Principal types described in Section 4.2, each demonstrating a scenario where a workload within a Trust Domain must call a partner service in a separate Trust Domain to complete the transaction.

A.1. User-Initiated External API Call Requiring a Partner Service

A financial services enterprise exposes a portfolio management API to its customers. A customer uses a mobile application to add a stock to their watch list, calling POST /watchlist at the enterprise’s API gateway with an OAuth 2.0 access token.

The API gateway workload requests a Txn-Token from the TTS, presenting the user's access token as the inbound credential. The TTS mints a Txn-Token with sub set to the user's enterprise identifier, scope set to watchlist-update, and rctx capturing the mobile client's OAuth client identifier and IP address. This Txn-Token propagates through the internal portfolio service call chain.

To enrich the watch list entry with current market data, the portfolio service must call a market-data API operated by a partner financial data provider in Trust Domain B. The portfolio service exchanges the Txn-Token for a JWT Authorization Grant using this profile. AS-A maps the user's enterprise identifier to a cross-domain user identifier agreed with the partner (e.g., the user's email address or a pairwise identifier), and includes a minimized txn_claims carrying scope: watchlist-update.

The partner's authorization server issues an access token that identifies the user (enabling per-user rate limiting and audit logging at the partner) without receiving the enterprise's internal Txn-Token, internal access token, or internal user database identifiers.

A.2. System-Initiated Event Requiring a Partner Service

An enterprise mail service receives an inbound email message via SMTP. The SMTP server is an internal system component operating under its own system credential; no external OAuth client is involved. The SMTP server requests a Txn-Token from the TTS with sub set to its system identity (system:mail-gateway@enterprise.example), scope set to mail-delivery, and rctx carrying the SMTP envelope sender address and the recipient user's internal identifier. This Txn-Token propagates to the mail storage service workload.

Before storing the message in the recipient's mailbox, the mail storage service must call a spam-rating API operated by a partner spam service in Trust Domain B (whose Authorization Server is https://as.spamsvc.example and whose spam-rating API is https://api.spamsvc.example/spam-rating).

The mail storage service exchanges the Txn-Token for a JWT Authorization Grant using this profile. AS-A maps the system identity to the cross-domain service identifier agreed with the spam service, and includes a minimized txn_claims carrying scope: mail-delivery and rctx.smtp_from (the envelope sender address, stripped of internal routing metadata).

The spam service's authorization server issues an access token for the spam-rating API. The spam service can apply per-sender and per-recipient policy based on txn_claims, enabling personalized spam filtering without requiring the enterprise to expose internal user tokens or the Txn-Token outside its trust boundary.

A.3. Automated Workload Requiring a Partner Service

An enterprise data platform runs a nightly telemetry aggregation job. The job is an automated workload with no direct external caller, triggered by an internal scheduler. The scheduler requests a Txn-Token from the TTS with sub set to the job's SPIFFE workload URI (spiffe://enterprise.example/telemetry/nightly-agg), scope set to telemetry-aggregation, and no user context in rctx.

To complete the aggregation, the job must query a third-party analytics API in Trust Domain B. The job exchanges the Txn-Token for a JWT Authorization Grant using this profile. AS-A maps the SPIFFE workload URI to a cross-domain workload identifier agreed with the analytics provider, and includes scope: telemetry-aggregation in txn_claims.

The analytics provider's authorization server issues a scoped access token. The txn claim in the JWT Authorization Grant allows the analytics provider to correlate API calls to the originating job run for billing and audit purposes, without receiving the internal SPIFFE URI or other Trust Domain A infrastructure details.

Appendix B. Relationship to Related Specifications

This specification is one of a family of profiles of [I-D.ietf-oauth-identity-chaining].

B.1. Identity Assertion JWT Authorization Grant

[I-D.ietf-oauth-identity-assertion-authz-grant] (the "ID-JAG" specification, adopted by the OAuth Working Group as of April 2026) targets deployments where AS-B already trusts AS-A (acting as an IdP) for Single Sign-On (SSO) and subject resolution, using an OpenID Connect ID Token or SAML 2.0 assertion as the subject token.

The key structural differences between the two profiles are:

Subject Token Type: The ID-JAG profile uses an OpenID Connect ID Token or SAML 2.0 assertion as the subject_token. This profile uses a Txn-Token (urn:ietf:params:oauth:token-type:txn_token).

Initiating Principal Scope: The ID-JAG profile is exclusively

centered on a human End-User whose authenticated session at the IdP drives the cross-domain access. This profile supports all three Initiating Principal types — human user, internal system, and automated workload — uniformly, because Txn-Tokens capture all three.

Trust Relationship Basis: The ID-JAG profile relies on a pre-existing SSO trust relationship between AS-A (the IdP) and AS-B (the Resource AS) for the same user population. This profile relies on a bilateral Cross-Domain Trust Agreement between AS-A and AS-B, which may exist independently of any shared identity provider.

audience and resource Parameters: Both profiles use audience to identify AS-B (the target authorization server) and resource ([RFC8707]) optionally to identify the target Protected Resource. These parameters serve the same distinct purposes in both profiles: audience → AS-B issuer URL → aud in the grant; resource → resource server URI → resource claim in the grant.

client_id Requirement: The ID-JAG includes a REQUIRED client_id claim identifying the OAuth 2.0 client at AS-B acting on behalf of the resource owner. This is appropriate where the application has a pre-registered client relationship with AS-B. This profile does not require a pre-registered client_id at AS-B; the Requesting Workload's identity is conveyed through client authentication to AS-A and the subject mapping in the JWT Authorization Grant.

Multi-Tenancy: The ID-JAG profile defines tenant, aud_tenant, and aud_sub claims for multi-tenant SaaS deployments. This profile does not define equivalent tenant-scoping claims, as Trust Domain boundaries are typically organizational or service-provider boundaries rather than tenant partitions within a shared platform.

Rich Authorization Requests (RAR): The ID-JAG profile supports the optional authorization_details claim ([RFC9396]) in the grant. This profile does not currently define RAR integration; a future revision MAY define how authorization_details from a Txn-Token are transcribed into the JWT Authorization Grant.

SAML 2.0 Interoperability: The ID-JAG profile includes SAML 2.0 identity assertion interoperability. This profile addresses only JWT-based Txn-Tokens.

Sender Constraining: Both profiles use the cnf claim to convey a

sender-constraining key to AS-B. The ID-JAG profile embeds cnf in the ID-JAG itself; this profile includes cnf in the JWT Authorization Grant, derived from the Requesting Workload's client credential presented to AS-A (see Section 9.2).

The two profiles are complementary. A deployment MAY support both: the ID-JAG profile for human-user cross-domain access coordinated through a shared identity provider, and this profile for any transaction-driven cross-domain access (user-initiated, system-initiated, or workload-initiated) where the trust relationship is established through a bilateral Cross-Domain Trust Agreement. An Authorization Server implementing both MUST distinguish between them by inspecting the JWT typ header: oauth-id-jag+jwt for the ID-JAG profile and txn-chain+jwt for this profile.

Acknowledgements

The author would like to thank Atul Tulshibagwale, Pieter Kasselman, Aaron Parecki, Brian Campbell, Arndt Schwenkschuster, Kelley Burgin, Karl McGuinness, and the members of the IETF OAuth Working Group for their foundational work on the specifications that this profile depends on.

The Transaction Tokens concept was originally developed by Atul Tulshibagwale, George Fletcher, and Pieter Kasselman. The OAuth Identity and Authorization Chaining Across Domains specification was authored by Arndt Schwenkschuster, Pieter Kasselman, Kelley Burgin, Michael Jenkins, Brian Campbell, and Aaron Parecki.

Author's Address

George Fletcher
Practical Identity LLC
Email: george@practicalidentity.com