Transaction Tokens

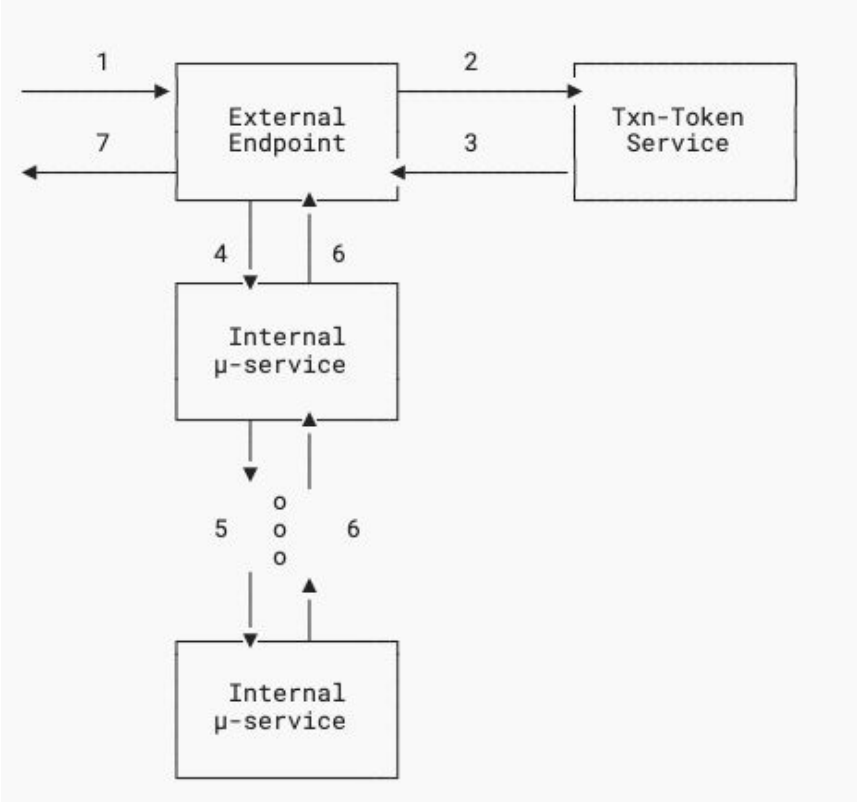
IETF 118
November 10, 2023

Atul Tulshibagwale
George Fletcher
Pieter Kasselman

What are Transaction Tokens

Transaction Tokens

- Token internal to a given trust boundary
- Maintains the immutable context of a transaction
  - Subject
  - Context
  - Authorization Details
- Shared across multiple workloads
- Allows for “down-scoping” a transaction at the edge
- Supports finer-grained authorization



Updates to the Draft

Updates to the existing draft

Based on conversations at IETF 117 and beyond

1. Removed Nested Transaction Tokens from the specification
2. Updated claims

Nested Tokens

Removed due to

- Added complexity
- Lack of compelling use cases

New Claims

Audience - ‘aud’ - string

- Identify the trust domain in which the transaction token is valid

Purpose - ‘purp’ - string

- Identifies the purpose of the transaction

New Claims

Authorization details - ‘azd’ - JSON Object

- Additional details for the transaction that must remain immutable throughout the call chain

New Claims

Request Context - `req_ctx` - JSON Object

- Contains the claims about the context of the transactions

The following sub claims are defined and can be extended

Requesting IP address - `req_ip` - string

Authentication method - `authn` - URN

Requesting workload - `req_wl` - URN

Updated Example

```
{
   "iss": "https://trust-domain.example/txn-token-service",
   "iat": "1686536226000",
   "aud": "trust-domain.example",
   "exp": "1686536526000",
   "txn": "97053963-771d-49cc-a4e3-20aad399c312",
   "sub_id": {
      "format": "email",
      "email": "user@trust-domain.example"
   },
   "req_ctx": {
     "req_ip": "69.151.72.123", // env context of external call
     "authn": "urn:ietf:rfc:6749", // env context of the external call
     "req_wl": "apigateway.trust-domain.example" // the internal entity that requested the Txn-Token
   },
   "purp" : "trade.stocks",
   "azd": {
      "action": "BUY", // parameter of external call
      "ticker": "MSFT", // parameter of external call
      "quantity": "100", // parameter of external call
      "user_level": "vip" // computed value not present in external call
   }
}
```

Open Questions

Special header to transmit Transaction Tokens?

Is a transaction token an authorization token?

Should it use the HTTP Authorization header?

Should we define a new HTTP header to carry the token?

Is there other work that an help here?

Subject of the transaction token

Should we allow both a default `sub` claim or stick with the SET defined `sub_id` claim?

Is the issuer of the transaction token really the same as the issuer of a simple `sub` claim?

Should the ‘iss’ claim be required or optional?

Given that transaction tokens are only valid within a single trust domain, is it necessary to include the ‘iss’ claim as it just makes the token bigger?

If the ‘iss’ claim is not there…

- How is key rotation handled?
- How is the token verified to be issued by the correct issuer?

Is the ‘aud’ claim sufficient for trust domain verification with local key management policies?

Sender constrained and Transaction Tokens

Given a transaction token is separate from client authentication and is issued for the purpose of completing a transaction, should they be sender constrained…

Or should there be a way to layer client authentication when presenting a transaction token but not binding them together?

Or would a different call chain path suffice?

Thank You