OAuth 2.0 Token Exchange

An exchange of ideas, issues and choices regarding token exchange

Brian Campbell & Mike Jones

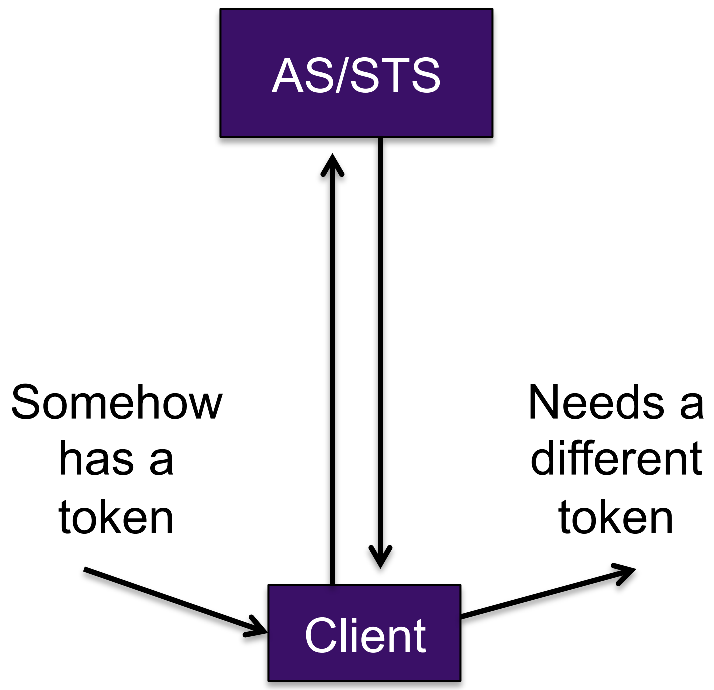
IETF 93
Prague
July 2015

Functional Goals

- Exchange one token for another token
  - Token type independence
  - Works for both OAuth tokens and security tokens
- Can describe properties of desired token, when applicable, e.g.:
  - Act-As and On-Behalf-Of capabilities
    - (kinda) like WS-Trust
  - Desired OAuth scope values
- Authenticate involved parties, when applicable
- **Keep simple things simple**

Use Cases

- Trade one token for another
  - Useful in a huge variety of circumstances
- Access to heterogeneous systems
  - Cross domain and otherwise
- Implicit/explicit impersonation/ delegation
  - Client and/or another user
- “Edge device” where client is reverse proxy or gateway
  - Chaining, validation, translation, down-scoping, etc.
- Framework should flexible but **keep simple things simple**



Drafts Referenced Herein

- draft-ietf-oauth-token-exchange-02
  - By Mike Jones & Tony Nadalin
- draft-campbell-oauth-sts-02
  - By Brian Campbell & John Bradley

Commonalities in Approaches

- Use new grant_type at Token Endpoint
- Have parameters for types of tokens
- Have parameters for act_as, on_behalf_of
- Have parameters for scope values

Issues and Decisions Needed

- The following describes issues and decisions needed
- Where existing drafts propose decisions, they are described

Issue: Bikeshedding the Title

- Options:
  - "OAuth 2.0 Token Exchange" (Jones draft)
  - "OAuth 2.0 Token Exchange: an STS for the REST of us" (Campbell draft)
- Observations:
  - Humor is good sometimes
  - The joke does convey the goal of simplicity and a modernized approach
  - This is really really important

Issue: Token Endpoint vs. New Endpoint

- Both drafts currently use the token endpoint
- Assertion Framework RFCs 7521-3 use the token endpoint, which is working in practice and proven in deployments
  - "We were able to easily add it to our existing infrastructure" – very large SaaS company
- There've been some past suggestions to define a new endpoint
- RFC 6749 defines a request/response mechanism and format for the token endpoint along with specific extension points
  - Use of the token endpoint needs to work within that framework
    - Recognizing that different grant types can define different sets of parameters and both drafts use a new grant type
      - True for request parameters. http://tools.ietf.org/html/rfc6749#section-4.5
      - Response parameters? http://tools.ietf.org/html/rfc6749#section-5
  - If that framework is too restrictive, a new endpoint should be defined

Issue: How to Authenticate the Requester

- Options:
  - Signature on a request JWT (in Jones draft)
  - OAuth client authentication (in Campbell draft)
- Observations:
  - RFC 6749 already provides a framework for client authentication
    - Including RFC 7523 JWT Assertion Client Authentication, which allows for a signature to be used for client authentication
  - Also, sometimes authentication not needed
    - OAuth client authentication allows anonymous
    - JWT “none” alg
  - Key question: Is the requester always an OAuth client?
  - The approaches aren’t necessarily mutually exclusive
  - OAuth Bearer or PoP tokens

Issue: Format of Request

- Options:
  - Primary content of the request is in a JWT that is a request parameter (in Jones draft)
  - Form request parameters (in Campbell draft)
  - JSON request body (like RFC 7591 Dynamic Client Registration)
- Observations:
  - JWT approach requires client to have JWT capabilities and will often result in double base64url encoding
  - Request parameters are simple and efficient
  - JSON request body at token endpoint not supported by RFC 6749 so would necessitate a new endpoint

Issue: Way to Pass Input Token

- Options:
  - Encode as request JWT (in current Jones draft)
  - Pass as a separate request parameter and type (in Campbell draft and planned as option for Jones draft)
- Observation:
  - To be token type independent, a separate token input parameter is required in the request
    - rather than the input token always being the JWT encoding the request (as in the current Jones draft)

Issue: Format of Response

- Options (both drafts use JSON):
  - security_token & security_token_type members (in Jones draft)
  - Standard RFC 6749 OAuth token endpoint response + security_token_type member (in Campbell draft)
- Observations:
  - Reuse of RFC 6749 response parameters is confusing to some while perfectly natural to others
    - token_type & expires_in & scope can provide client with useful info about the token when it's opaque
    - token_type & expires_in often unnecessary, since this information is typically encoded in the token itself when it's not opaque
  - In one interpretation of RFC 6749, the Jones draft style would necessitate a new endpoint because it departs from RFC 6749's token endpoint response definition
  - In another interpretation, the same endpoint can be used because the parameters are grant type specific

Issue: Indicating the Target of the Requested Token

- Providing requester the ability to indicate where it intends to use the requested token allows the server to apply policy
  - Campbell draft uses “aud” parameter akin to PoP Key Distribution (draft-ietf-oauth-pop-key-distribution)
    - (currently required but should be optional)
  - For Jones draft, an “aud” claim would be used
- Observations:
  - Use cases exist where this is needed and “aud” seems to fit
  - … “aud” has applicability beyond POP

Issue: Act-As, On-Behalf-Of Terminology

- Some find the WS-Trust based act-as and on-behalf-of terminology confusing
  - Even confusion around John Bradley’s confusion
- Proposed solution:
  - Add examples showing how act-as, on-behalf-of are used in practice
  - Evaluate specific editorial suggestions on how to make the meanings clearer
- Other solution:
  - Use new terminology

Issue: Names for OAuth Token Types

- Options:
  - urn:ietf:params:oauth:token-type:access-token & urn:ietf:params:oauth:token-type:refresh-token URIs
  - “access_token” & “refresh_token” names from RFC 6749
  - Default a “urn:ietf:params:oauth:token-type:” prefix when a simple name is used
- Observations:
  - Neither draft specifies this currently but some identifiers are needed

Issue: Defining actor claim

- Should we define a way of making a claim that a party can act for the issuing party?
  - Useful for evaluating act-as requests
  - This would be a JWT claim
    - Similar claims could be defined for other token types
  - Present in Jones draft – not in Campbell draft
- Observations:
  - Potentially useful though may need refinement
  - Need to maintain token type independence of overall framework

Issue: Proof-of-Possession Support

- Mechanisms to handle PoP tokens are needed/desirable
  - For both input and output tokens (independently)
  - For output tokens and key negotiation, consistent use of token endpoint syntax and semantics allows straightforward incorporation and reuse of PoP Key Distribution
  - For input tokens, consider existing proof-of-possession proposals inflight
    - Others?
  - Some use-cases get rather complicated quickly (i.e. the “edge device” case)
  - Concern over introducing inter-spec dependencies?

Way Forward

- Discuss issues and determine resolutions
- Produce new draft incorporating decisions
- Combine editors & produce common draft
  - Maybe also invite Chuck Mortimore to be an editor