IETF 121 Dublin
OAuth WG
November 2024

OAuth Identity and Authorization Chaining Across Domains

Arndt Schwenkschuster (SPIRL), Pieter Kasselman (SPIRL), Kelley Burgin (MITRE),
Michael J. Jenkins (NSA-CCSS), Brian Campbell (Ping Identity)

Agenda

- Refresher
- Confirmation key transfer

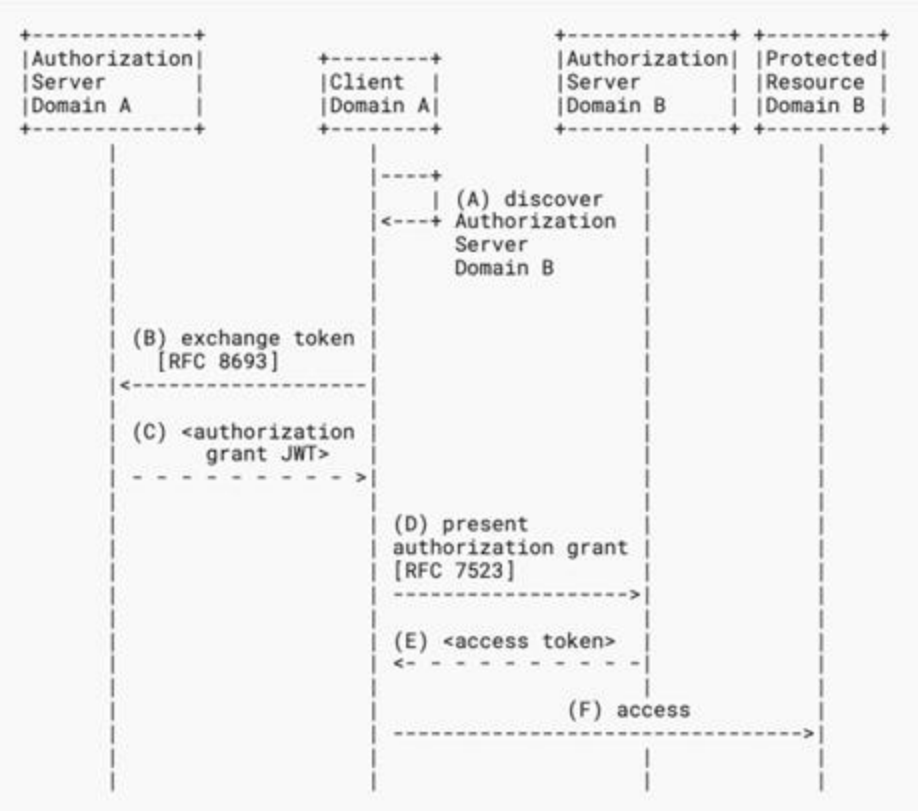
https://datatracker.ietf.org/doc/draft-ietf-oauth-identity-chaining/

What’s in the Draft

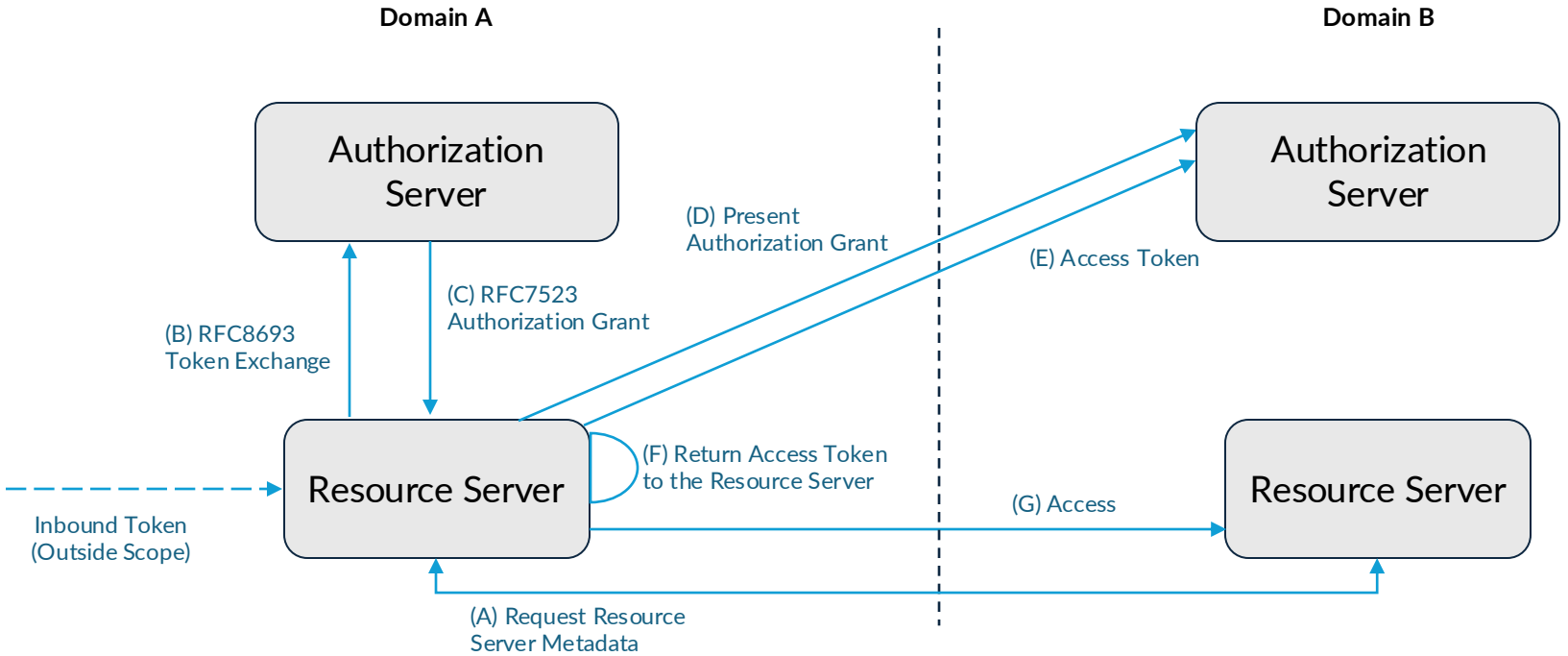
- The draft describes a mechanism to
  - preserve identity information and
  - federate authorization

  across trust domains that use the OAuth 2.0 Framework
- Open issue from IETF 120
  - How to enable sender constrained tokens throughout the flow
- Proposed solution to propagate confirmation key data

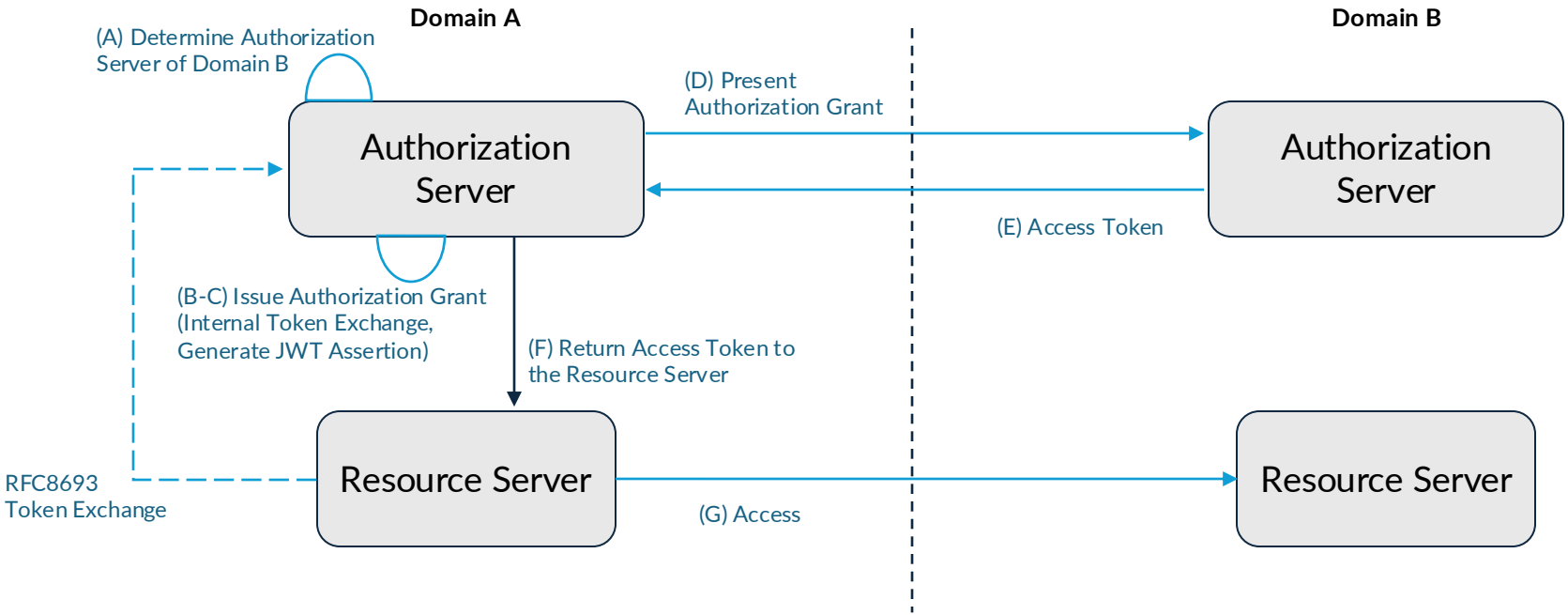
General flow



Resource Server acting as the Client



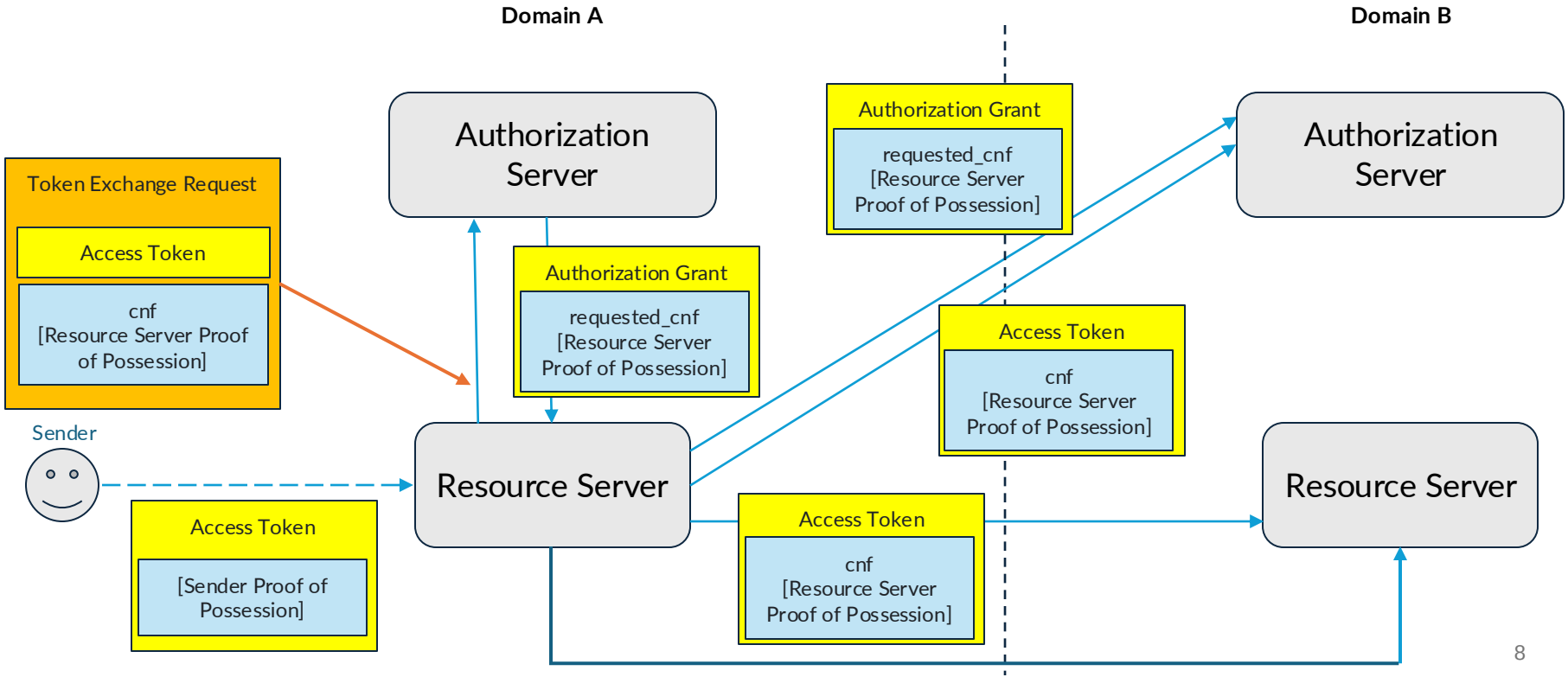
Authorization Server acting as the Client



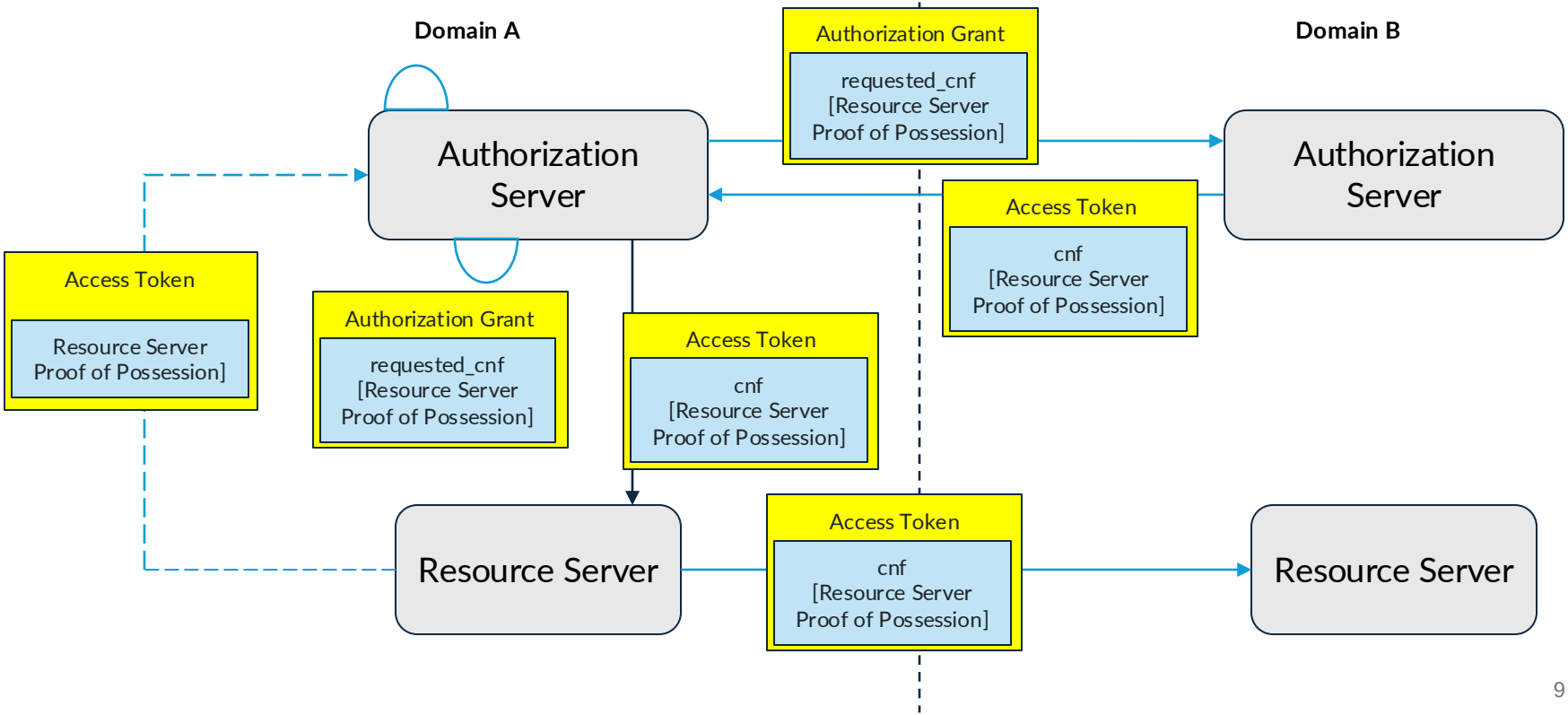
Confirmation Key Transfer: Additional Processing Rules

- Token Exchange Request
  - If the Client requests a JWT assertion in return from Token Exchange (indicates intended use across trust domains), the authorization server in Domain A includes in the returned JWT Assertion a "requested_cnf" claim that contains the “cnf” value.
  - “requested_cnf” {“cnf”}
- Authorization Grant
  - If the assertion contains a ”requested_cnf {“cnf”}” claim, the authorization server in Domain B includes the “cnf” claim in the returned access token.

Confirmation Key Transfer: Resource Server acting as the Client



Confirmation Key Transfer: Authorization Server acting as the Client



Open Issues

- Issue #98 Security Considerations on Client authentication
- Issue #99 Using different keys for sender constrained tokens
- Issue #100 Clarify client terminology
  - Sender of the inbound token vs abstract client
- Issue #101 Remove need for additional metadata

Discussion - “requested_cnf”

Does the WG have other alternative solutions to carry confirmation key data across trust domains (through token exchange and assertions)?

What’s Next

- Read the PR in GitHub
  - Provide Feedback
- WGLC before IETF 122

General Flow