draft-ietf-oauth-browser-based-apps-03

OAuth 2.0 for Browser-Based Apps

Aaron Parecki

IETF 105 • Montreal
July 26, 2019

OAuth 2.0 for Browser Based Apps

- Includes recommendations for implementors building browser-based apps using OAuth 2.0
- "Browser-based apps" are defined as applications running in a browser, also called a "SPA" or "single-page apps"

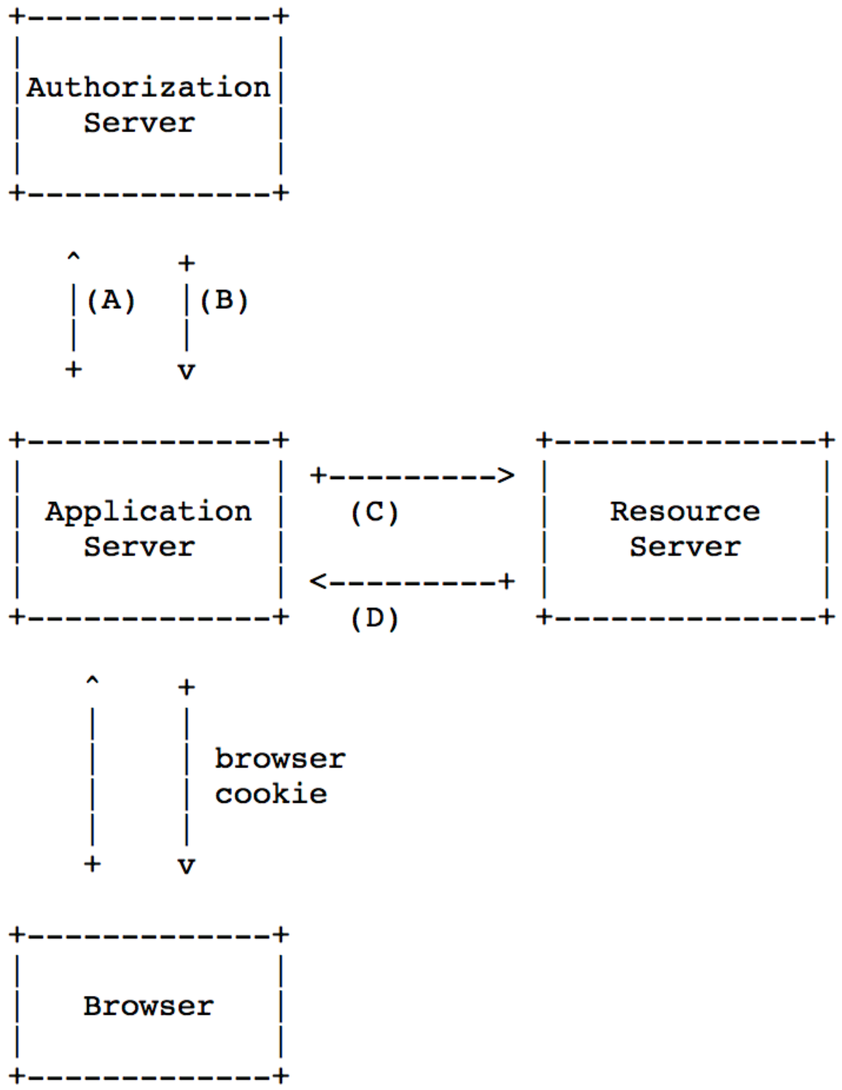
OAuth 2.0 for Browser Based Apps

- **SHOULD** use the OAuth 2.0 authorization code flow with the PKCE extension
- **MUST NOT** return access tokens in the front channel (e.g. no Implicit flow)
- **MUST** use the OAuth 2.0 state parameter to carry one-time use CSRF tokens
- The AS **MUST** require an exact match of the redirect URI
- The AS **SHOULD NOT** issue refresh tokens to browser-based apps

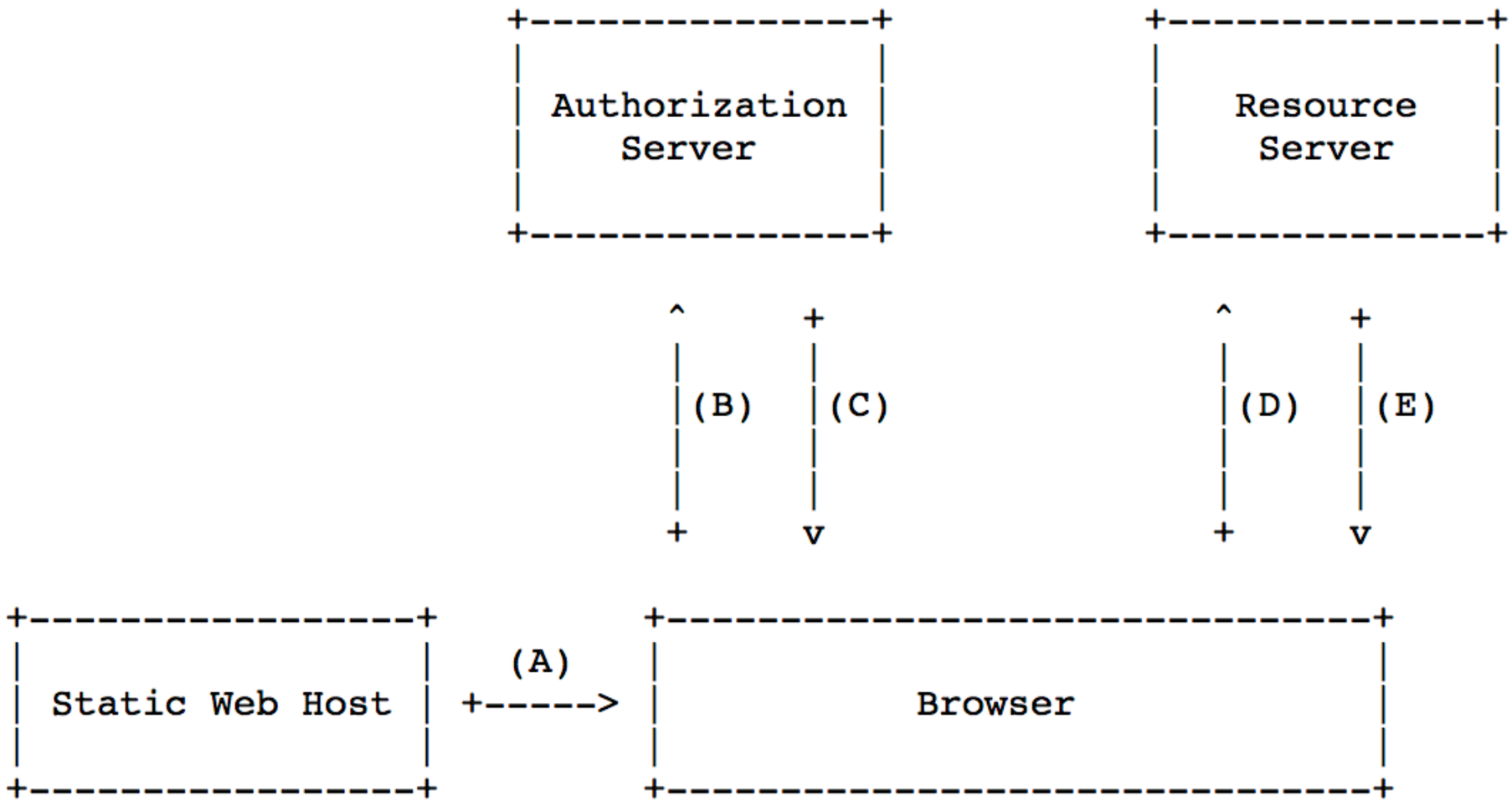
What's New Since IETF104?

- Exact redirect URI matching - no partial matching allowed
- Split doc into three architectural patterns (with diagrams)
- Expanded reasoning behind same-domain architecture recommendations
- Editorial clarifications

SPA with Backend



SPA without Backend



Same-Domain Applications

6.1. Apps Served from a Common Domain as the Resource Server

- Traditional OAuth redirect flows are not needed if the client and AS and RS can share cookies, and OAuth introduces problems that could be avoided otherwise
- But the AS/RS separation is still useful - enables MFA, avoids apps handling passwords, etc
- What should we recommend for these apps?
- Should we limit these recommendations to same-domain apps that *do* use OAuth? If so, what are those recommendations?

Open Questions

- Confirm that we want to require "state" be used for CSRF protection even if PKCE is used
- With the potential for DPoP or similar, should the document avoid saying "SHOULD NOT issue refresh tokens" to leave open this possibility?
- Can we recommend that browser-based apps MUST NOT use the password grant?
- Section 9.8 - a list of security issues with the implicit flow - keep a summary and refer to Security BCP?
- SPA w/backend - Should we have some indication that the AT may be sent to the browser?

Refresh Tokens in SPAs

Pros:

- Refresh tokens w/rotation provide the AS more opportunities to detect problems
- Refresh tokens mean shorter lifetime access tokens can be used

Cons:

- Refresh tokens are bearer tokens and can be used if stolen
- RTs typically have a longer lifetime than ATs so are riskier

Refresh Tokens in SPAs

Potential Solutions:

- No bearer refresh tokens? (Require client auth or PoP of some sort)
- Require that refresh tokens have a limited lifetime?
  - Some time-based value? Tied to AS authentication session?
- Require refresh token rotation? (as mentioned in Security BCP)
- If refresh tokens are rotated, should the new one extend the lifetime or keep the same total lifetime?
- Not mention anything about refresh tokens?

Fin