

IETF 122
Bangkok
March 2025

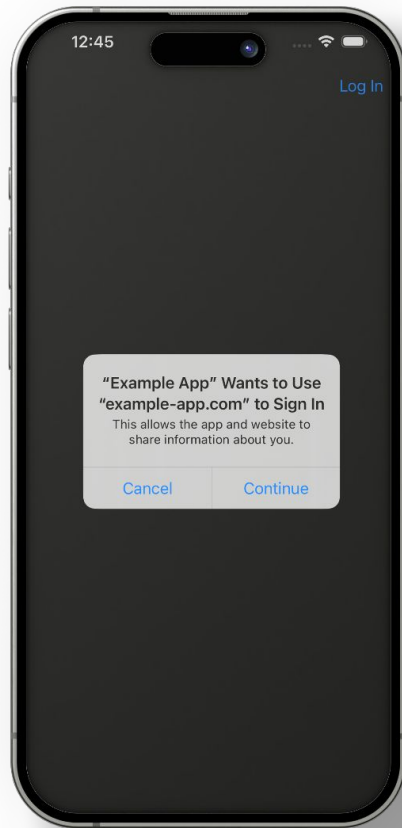
Aaron Parecki
George Fletcher
Pieter Kasselmann

OAuth for First-Party Apps

[https://datatracker.ietf.org/doc/draft-ietf-oauth-first-party-apps/
draft -00](https://datatracker.ietf.org/doc/draft-ietf-oauth-first-party-apps/draft-00)

Why?

Developers want a
better user experience
for first-party apps

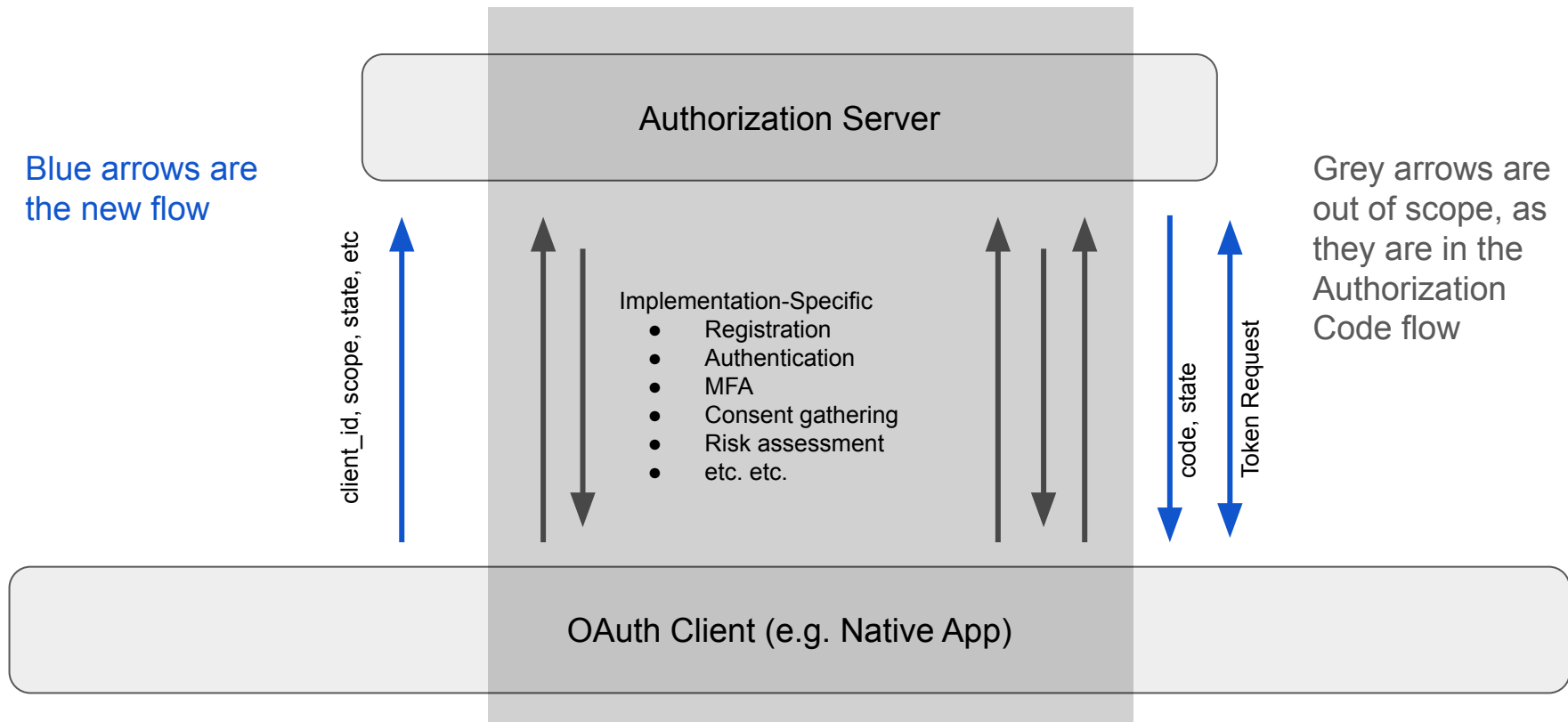


What is happening today

People are finding workarounds to avoid RFC8252

- Custom DIY solutions for native apps
- Using Resource Owner Password Grant
 - (Unable to add MFA)
- OAuth servers creating proprietary APIs to facilitate direct interaction with native apps
- Scripting hidden web views to emulate user interaction with the AS

First-Party Apps Flow



Authorization Challenge Endpoint

- New endpoint
- Accepts parameters that would have been included in the query string to the authorization endpoint
 - including any extensions such as Resource Indicators, OpenID Connect, JAR, etc
- Accepts POST from client to start and continue an authorization
 - The AS defines what the client sends in the requests and defines its own error responses
- Response is an authorization code, error, “`redirect_to_web`”, or custom
 - The AS may want to interact with the user directly, e.g. based on risk assessment, new authentication method not implemented in the app, or exceptions like account recovery
- Further interaction with the user can happen at custom endpoints

Implementations

- **WSO2 Identity Server**
 - <https://github.com/wso2/product-is/issues/21696>
- **Microsoft**
 - <https://devblogs.microsoft.com/identity/introducing-native-auth/>
 - (based on an earlier version of this draft)
- **Yahoo**
 - <https://identiverse.com/idv24/session/2089642/>
- **Auth0**
 - (in progress, no public docs yet)
- **Keycloak**
 - <https://github.com/keycloak/keycloak/discussions/25014>

Previous WG Decision

Create an extension to this draft for passkeys

- In progress work from Tim Cappalli
<https://github.com/oauth-wg/oauth-first-party-apps/pull/93>
- Needs to be pulled out into its own document

Questions

Should this draft be defined as a PAR extension or not?

Would defining this as a PAR extension affect your plans to implement it today?

What are the pros and cons of defining this as a PAR extension?

FIPA

This draft is already 80% PAR, but with different names for things

Implementing this draft on its own would get you 80% of the way to implementing PAR

Main differences if extending PAR:

- “redirect_to_web” error would be removed in favor of the current PAR request_uri response

Pros and Cons of FIPA Extending PAR

Pros

Trims down the amount of text in this draft because fewer new things are being defined

Cleaner spec architecture/layering

Avoids risk of reinventing things

May help drive adoption of PAR

Cons

Creates the perception of more work to implement, since you first have to implement PAR

”Happy path” of PAR is web redirect and “happy path” of FIPA is to stay in the native app

Possible conflation of first-party and third-party apps