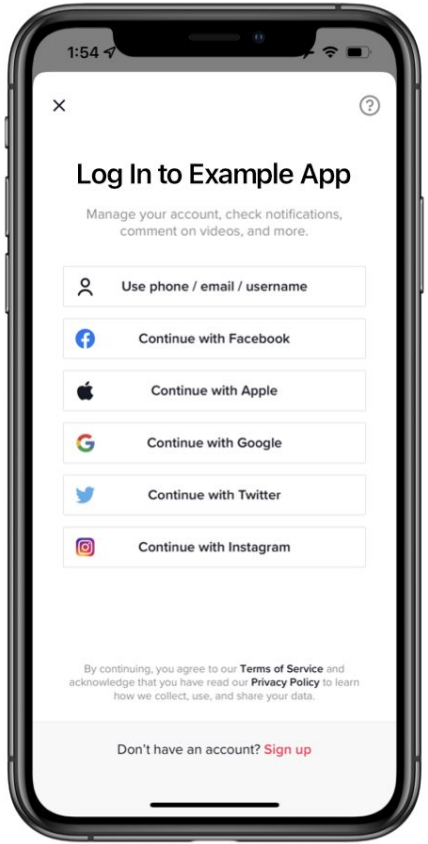
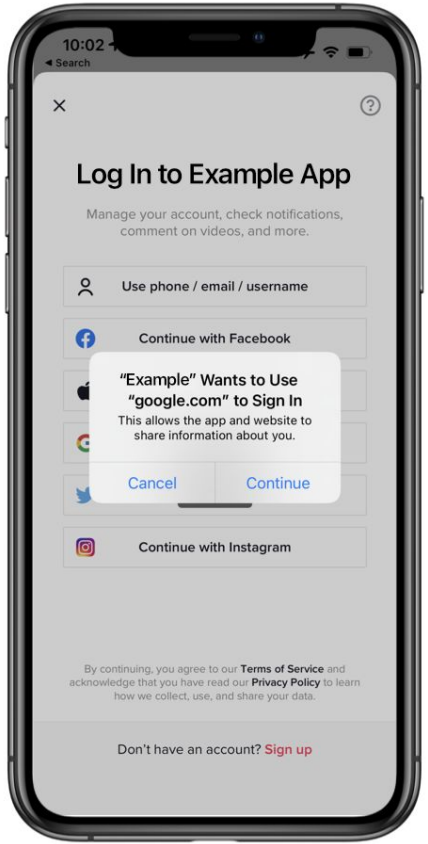
IETF 116
Yokohama
March 2023

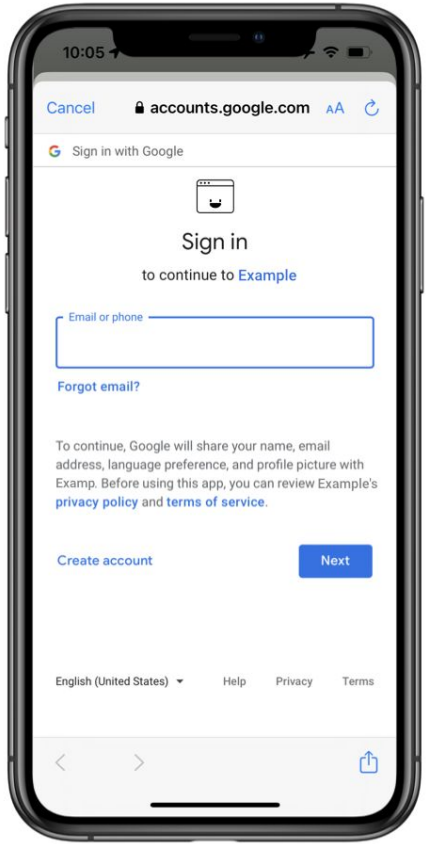
Aaron Parecki
George Fletcher
Pieter Kasselman

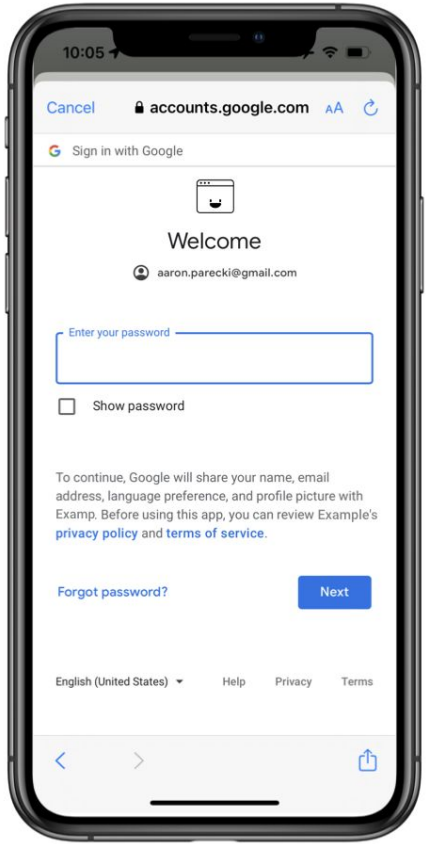
Native Apps UX

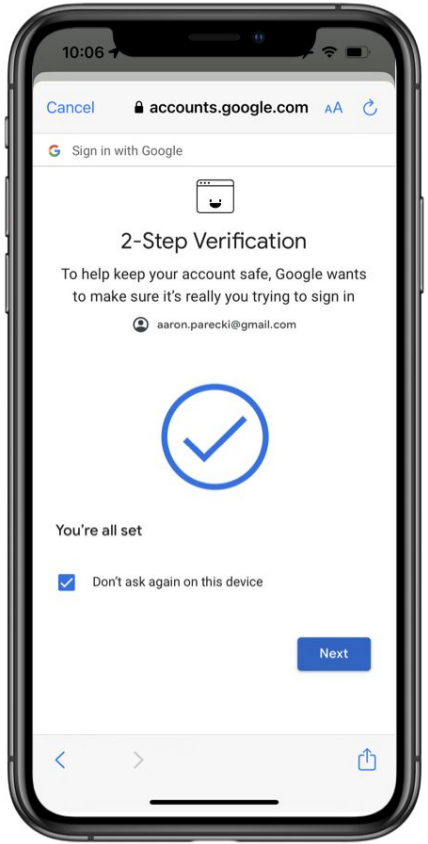
OAuth for Native Mobile Apps Today

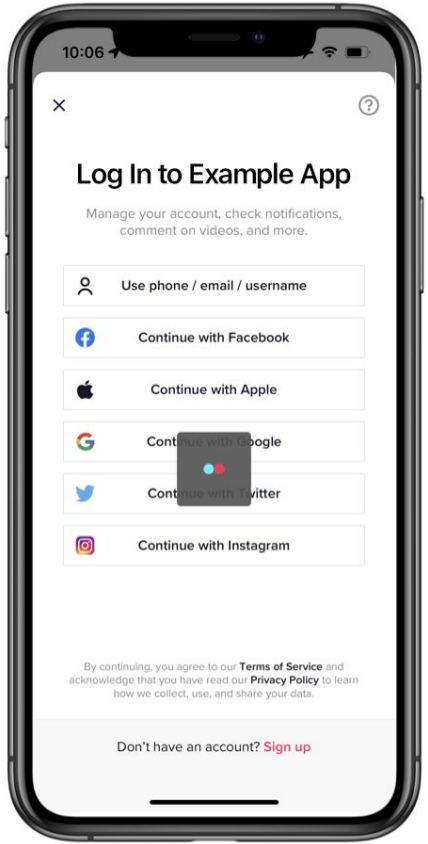

1:54
Log In to Example App
Manage your account, check notifications, comment on videos, and more.
Use phone / email / username
Continue with Facebook
Continue with Apple
Continue with Google
Continue with Twitter
Continue with Instagram
By continuing, you agree to our Terms of Service and acknowledge that you have read our Privacy Policy to learn how we collect, use, and share your data.
Don't have an account? Sign up


10:02
Search
Log In to Example App
Manage your account, check notifications, comment on videos, and more.
Use phone / email / username
Continue with Facebook
"Example" Wants to Use "google.com" to Sign In
This allows the app and website to share information about you.
Cancel
Continue
Continue with Instagram
By continuing, you agree to our Terms of Service and acknowledge that you have read our Privacy Policy to learn how we collect, use, and share your data.
Don't have an account? Sign up


10:05
Cancel
accounts.google.com
AA
Sign in with Google
Sign in
to continue to Example
Email or phone
Forgot email?
To continue, Google will share your name, email address, language preference, and profile picture with Examp. Before using this app, you can review Example's privacy policy and terms of service.
Create account
Next
English (United States)
Help
Privacy
Terms


10:05
Cancel
accounts.google.com
AA
Sign in with Google
Welcome
aaron.parecki@gmail.com
Enter your password
Show password
To continue, Google will share your name, email address, language preference, and profile picture with Examp. Before using this app, you can review Example's privacy policy and terms of service.
Forgot password?
Next
English (United States)
Help
Privacy
Terms


10:06
Cancel
accounts.google.com
AA
Sign in with Google
2-Step Verification
To help keep your account safe, Google wants to make sure it's really you trying to sign in
aaron.parecki@gmail.com
You're all set
Don't ask again on this device
Next


10:06
Log In to Example App
Manage your account, check notifications, comment on videos, and more.
Use phone / email / username
Continue with Facebook
Continue with Apple
Continue with Google
Continue with Twitter
Continue with Instagram
By continuing, you agree to our Terms of Service and acknowledge that you have read our Privacy Policy to learn how we collect, use, and share your data.
Don't have an account? Sign up

Great for third party apps

- Secure isolation between app and system browser
- Leverages existing session at the OAuth server
- Supports phishing-resistant MFA

Developers want a better user experience for first-party apps

What is happening today

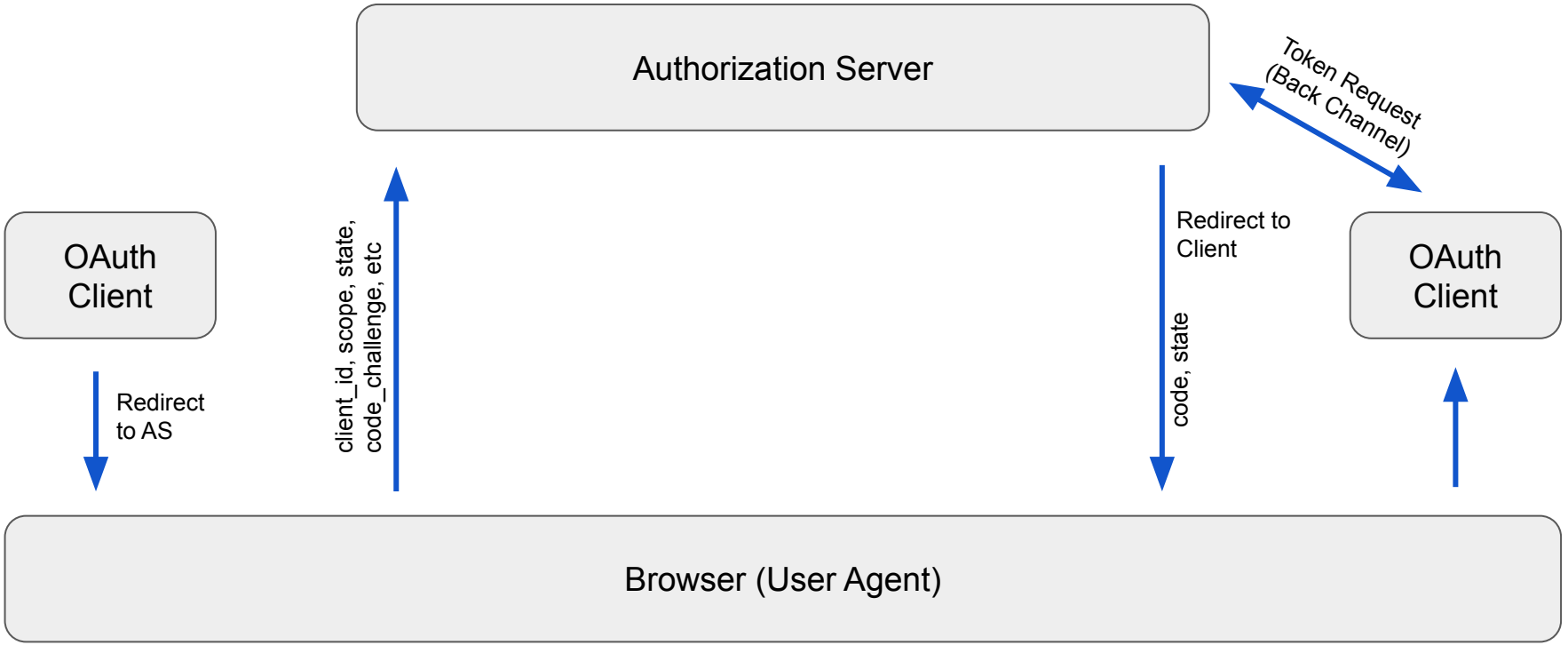
People are finding workarounds to avoid RFC8252

- 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

What is happening today

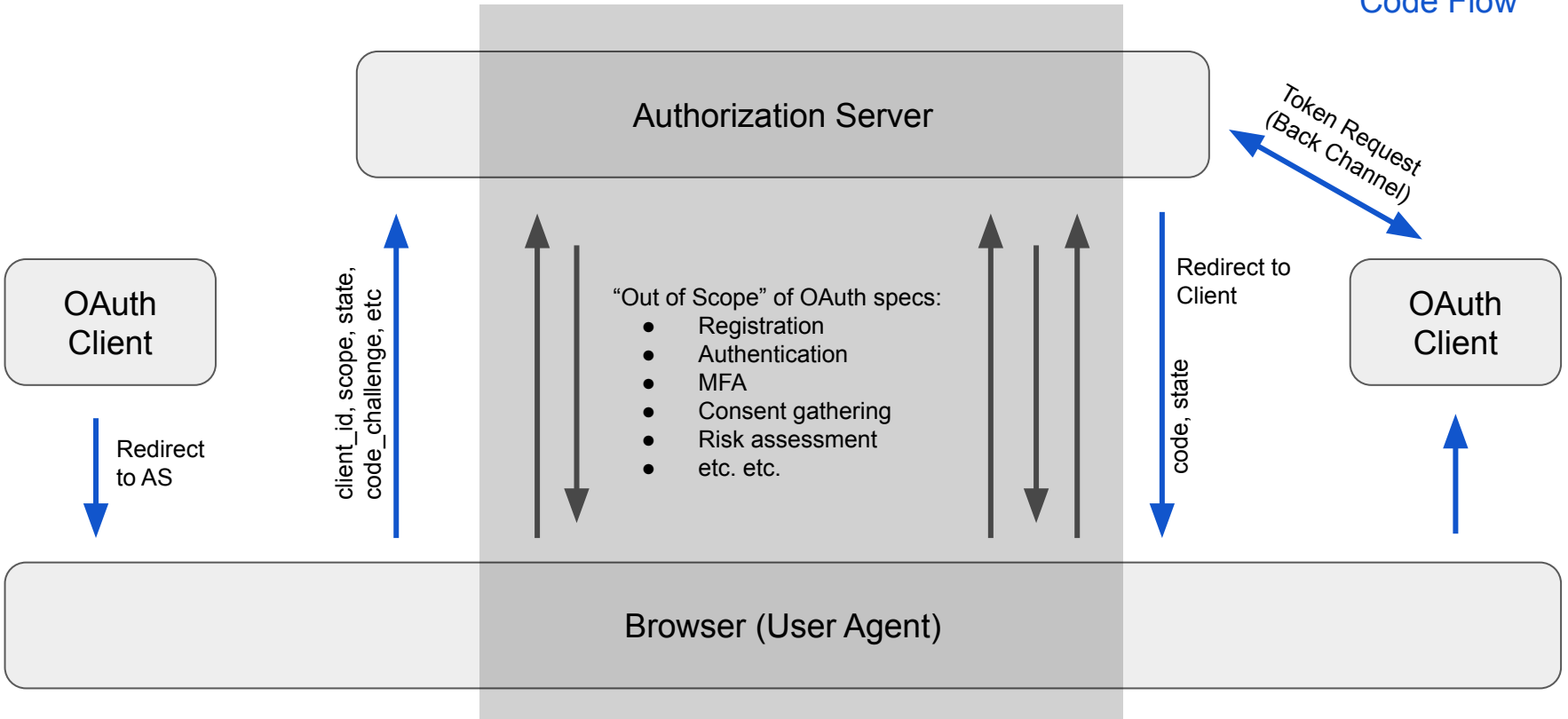
All of these lead to worse outcomes

Authorization Code Flow for Web Apps



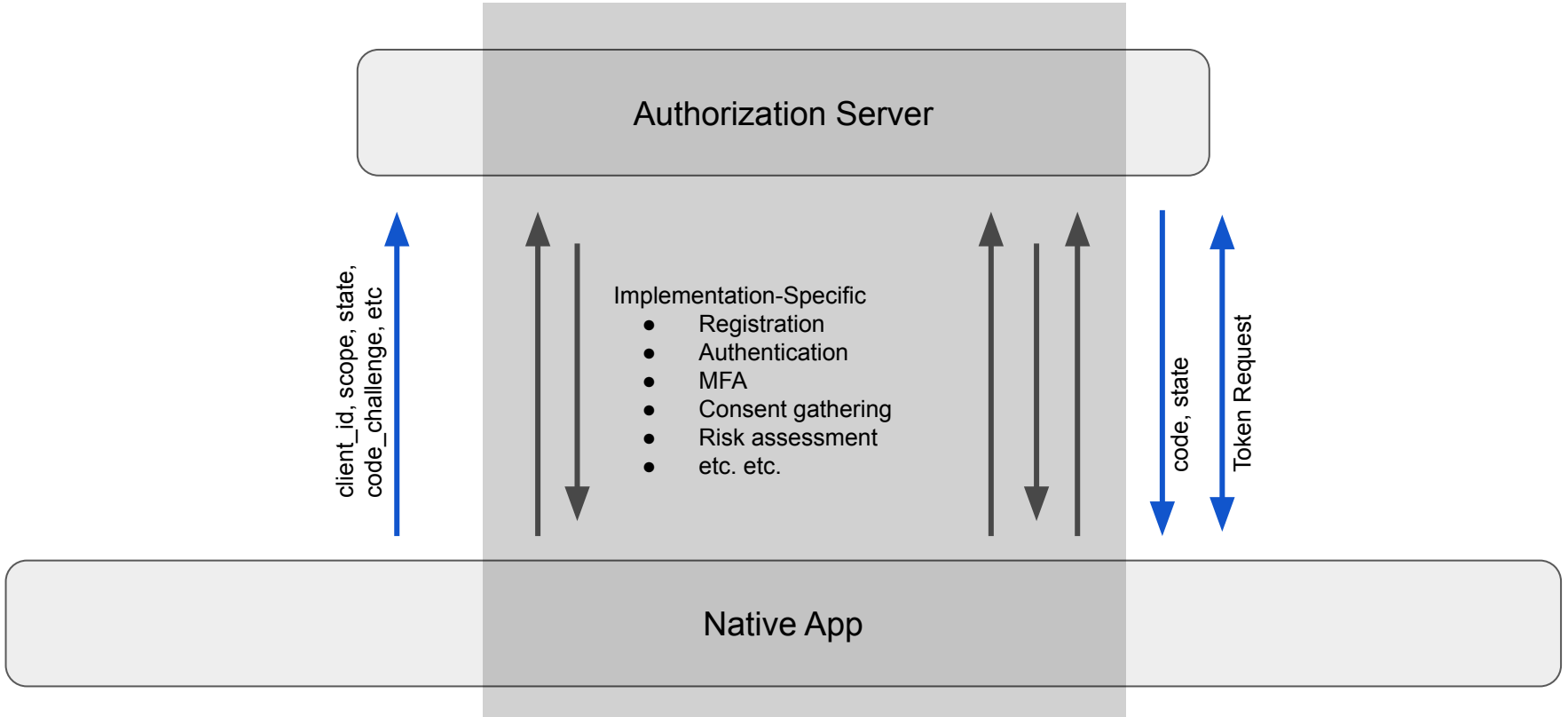
Authorization Code Flow for Web Apps

Blue arrows are the OAuth Authorization Code Flow



Native Apps Flow

Blue arrows are the new flow



Questions

- Is OAuth WG the right place for this?
  - The artifacts issued will still be access tokens and refresh tokens, and likely ID tokens as well
- Should the flow leave the details of the app-AS exchange unspecified to mirror the Authorization Code flow?
  - Should the flow define a “challenge-response” format that can be extended with provider-specific logic?