OAuth in Native Apps:

It's worse than we thought.

OAuth Security Workshop
April 2024 • Rome

Aaron Parecki
aaronparecki.com

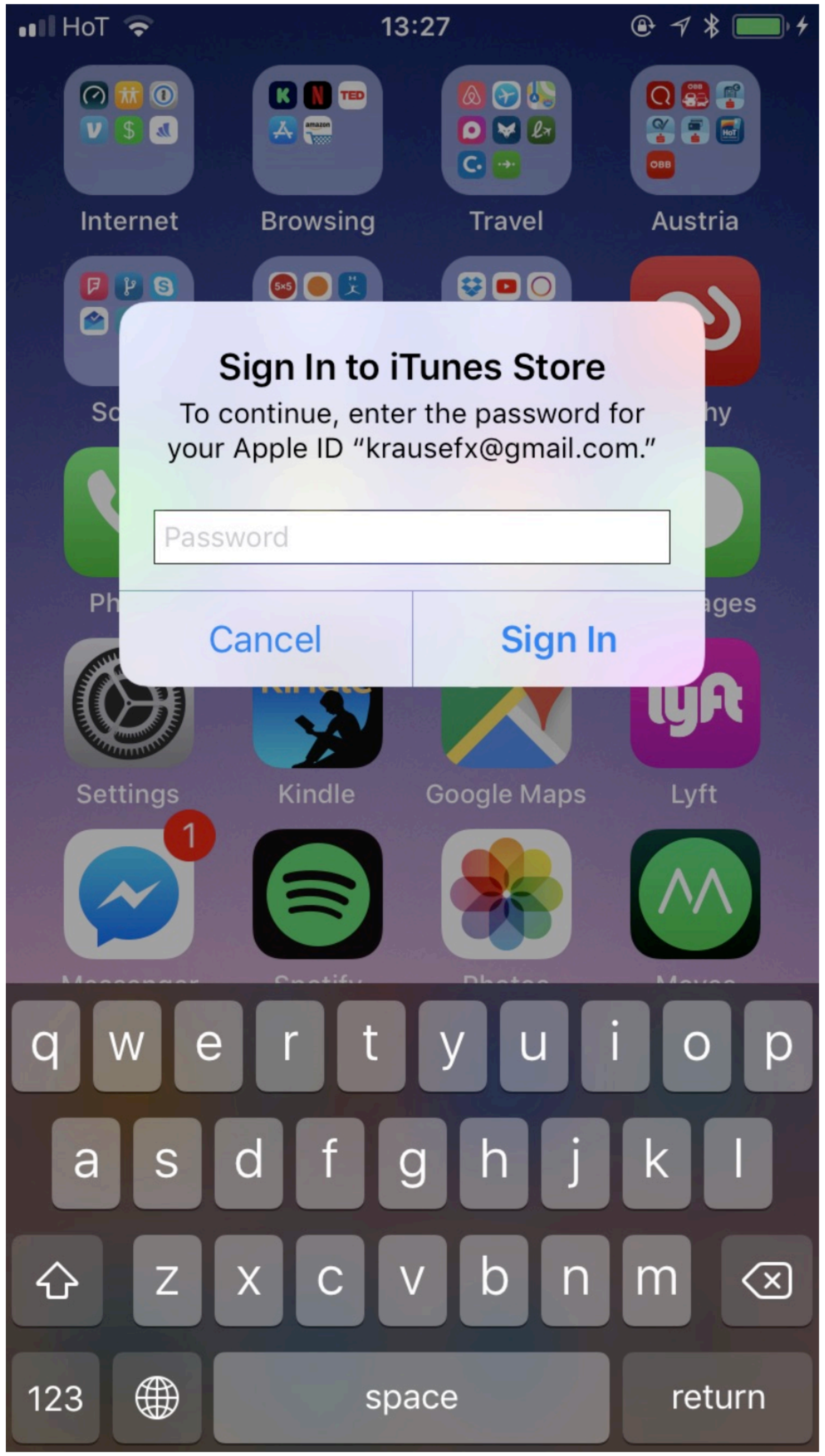
OAuth 2.0 for Native Apps (RFC 8252) Summary

- The client MUST use the system browser, not embedded web views
- MUST be treated as a public client
- The client MUST use PKCE
- Redirect URLs can be:
  - custom URI scheme (`com.example-app://redirect`)
  - app-claimed https URL (`https://example-app.com/redirect`)
  - Loopback address with custom port (`http://127.0.0.1:5192/redirect`)
- The AS SHOULD NOT automatically redirect without user consent
  - Unless the identity of the client can be assured (e.g. using app-claimed https URLs)

https://datatracker.ietf.org/doc/html/rfc8252

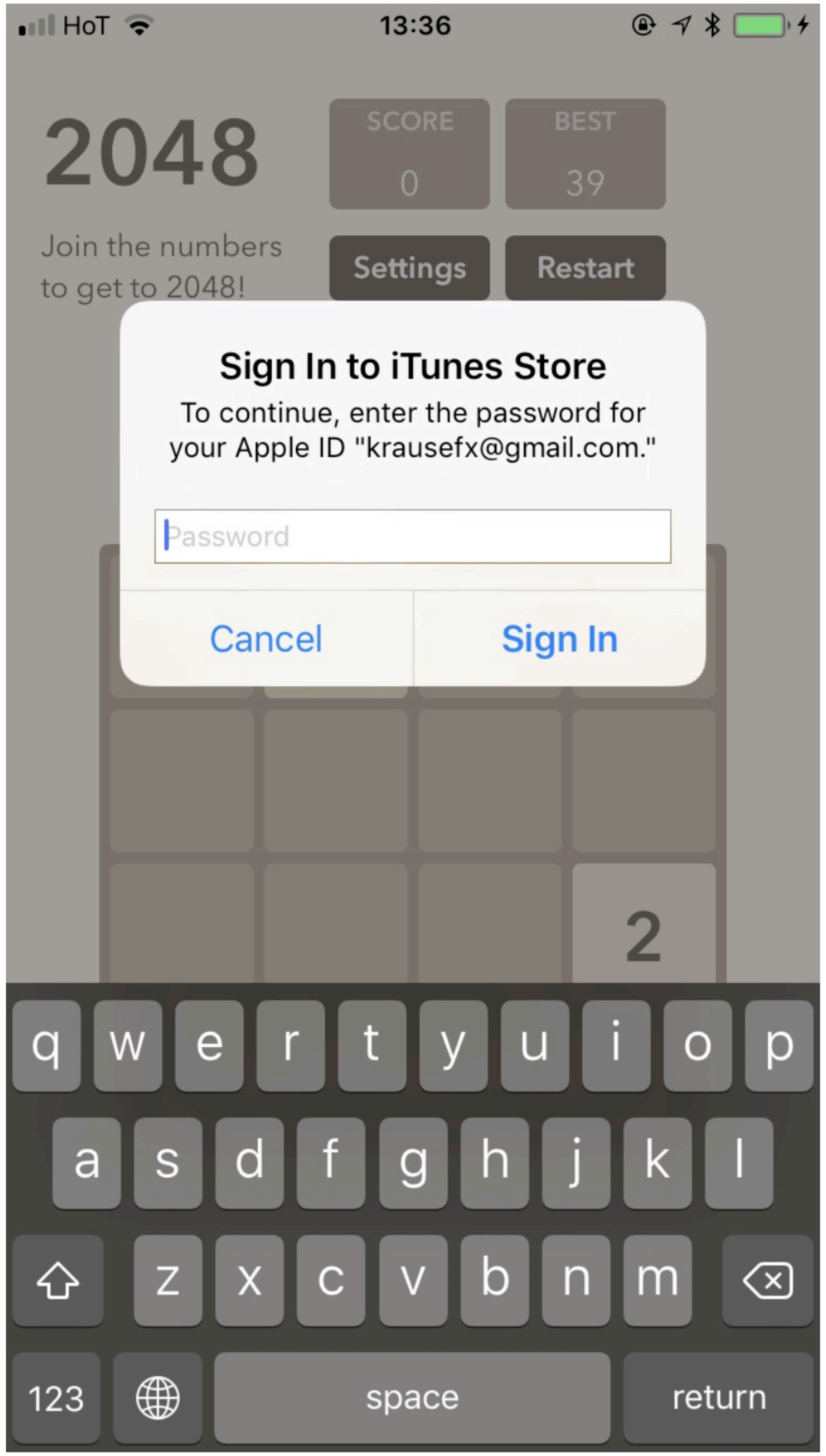
Use the System Browser

```
To conform to this best practice, native apps
MUST use an external user-agent to perform OAuth
authorization requests.
```



Sign In to iTunes Store

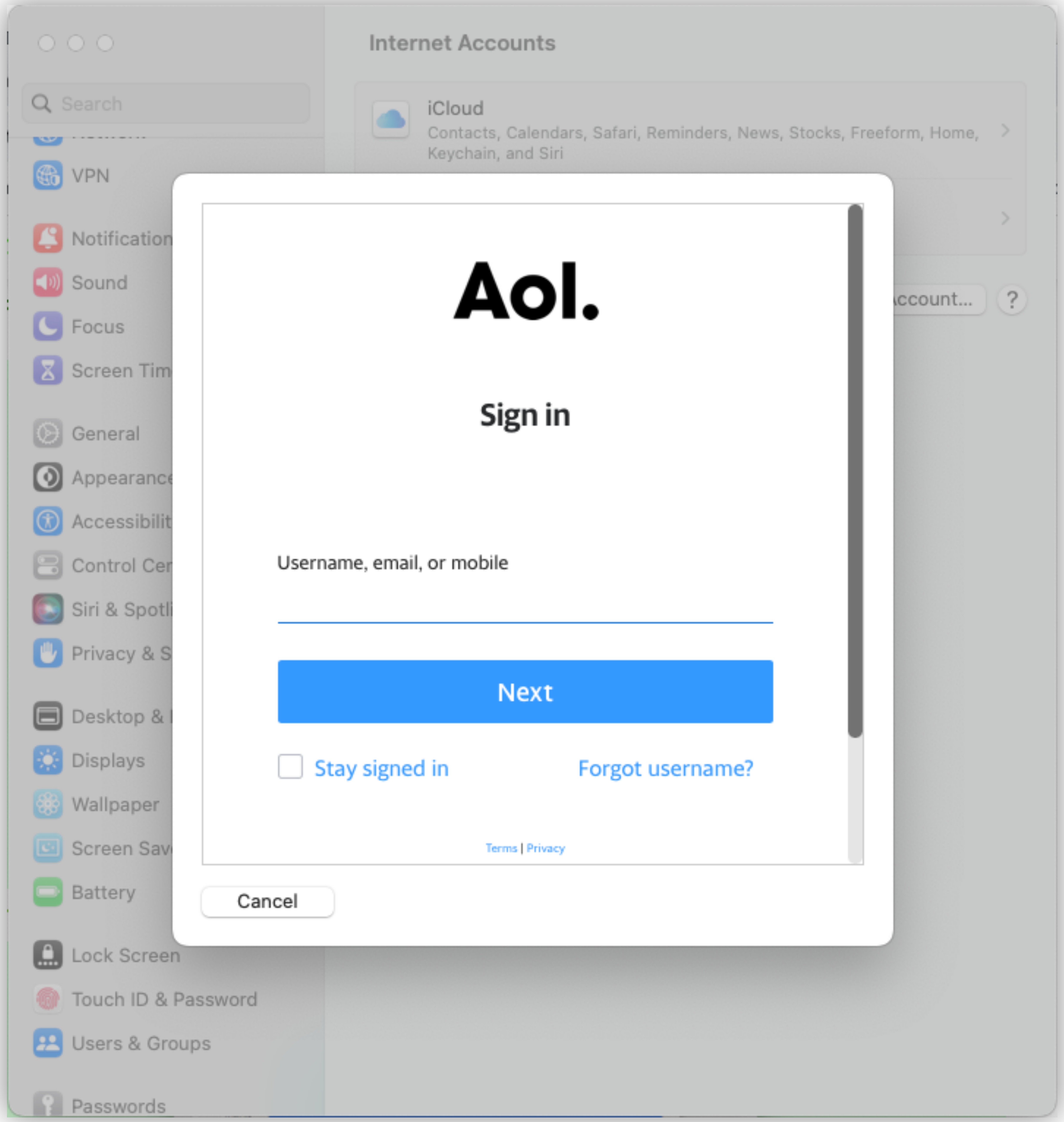
To continue, enter the password for your Apple ID "krausefx@gmail.com."



Sign In to iTunes Store

To continue, enter the password for your Apple ID "krausefx@gmail.com."

https://krausefx.com/blog/ios-privacy-stealpassword-easily-get-the-users-apple-id-password-just-by-asking



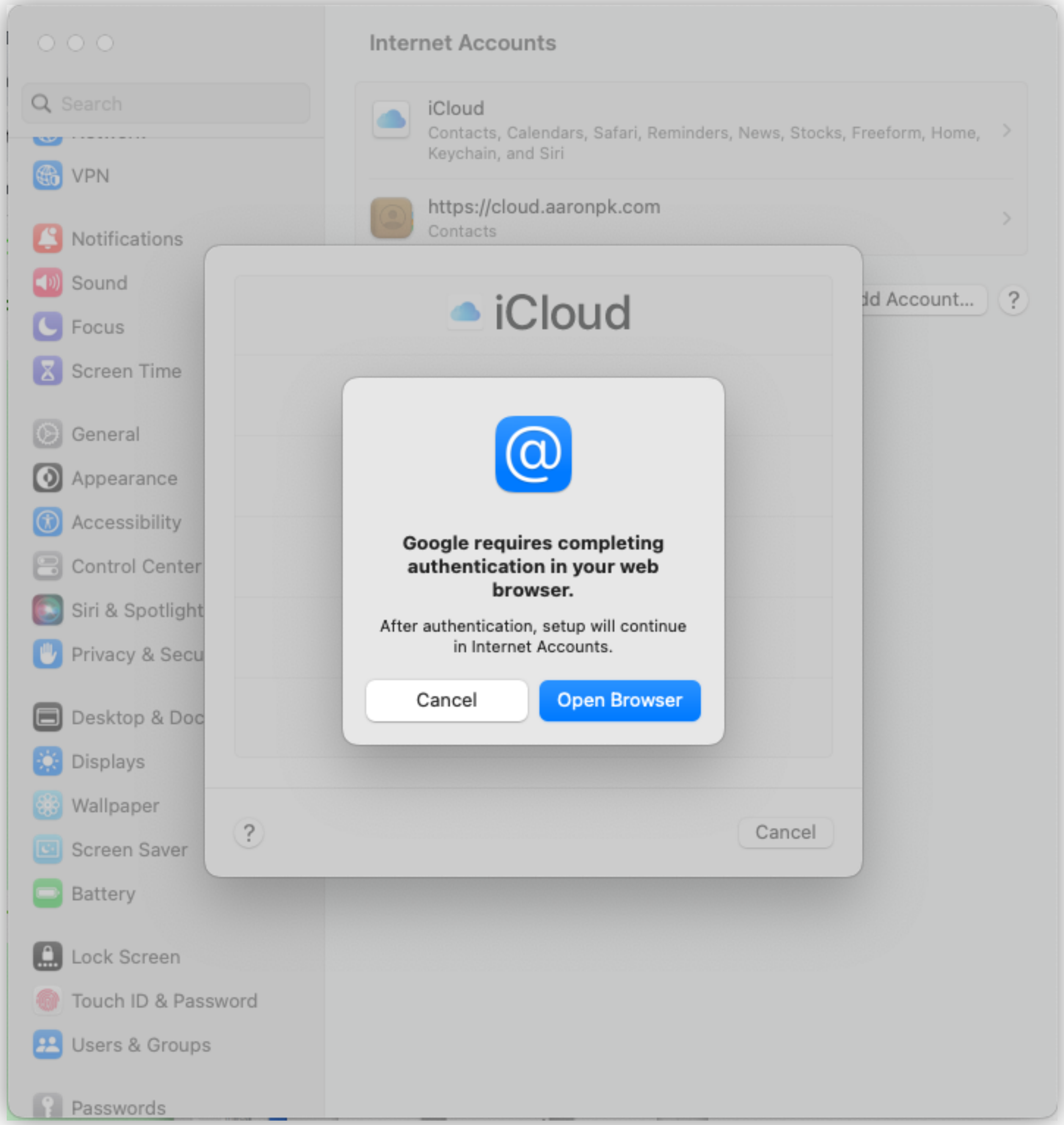
Internet Accounts

iCloud
Contacts, Calendars, Safari, Reminders, News, Stocks, Freeform, Home, Keychain, and Siri

Aol.

Sign in

Username, email, or mobile



Internet Accounts

iCloud
Contacts, Calendars, Safari, Reminders, News, Stocks, Freeform, Home, Keychain, and Siri

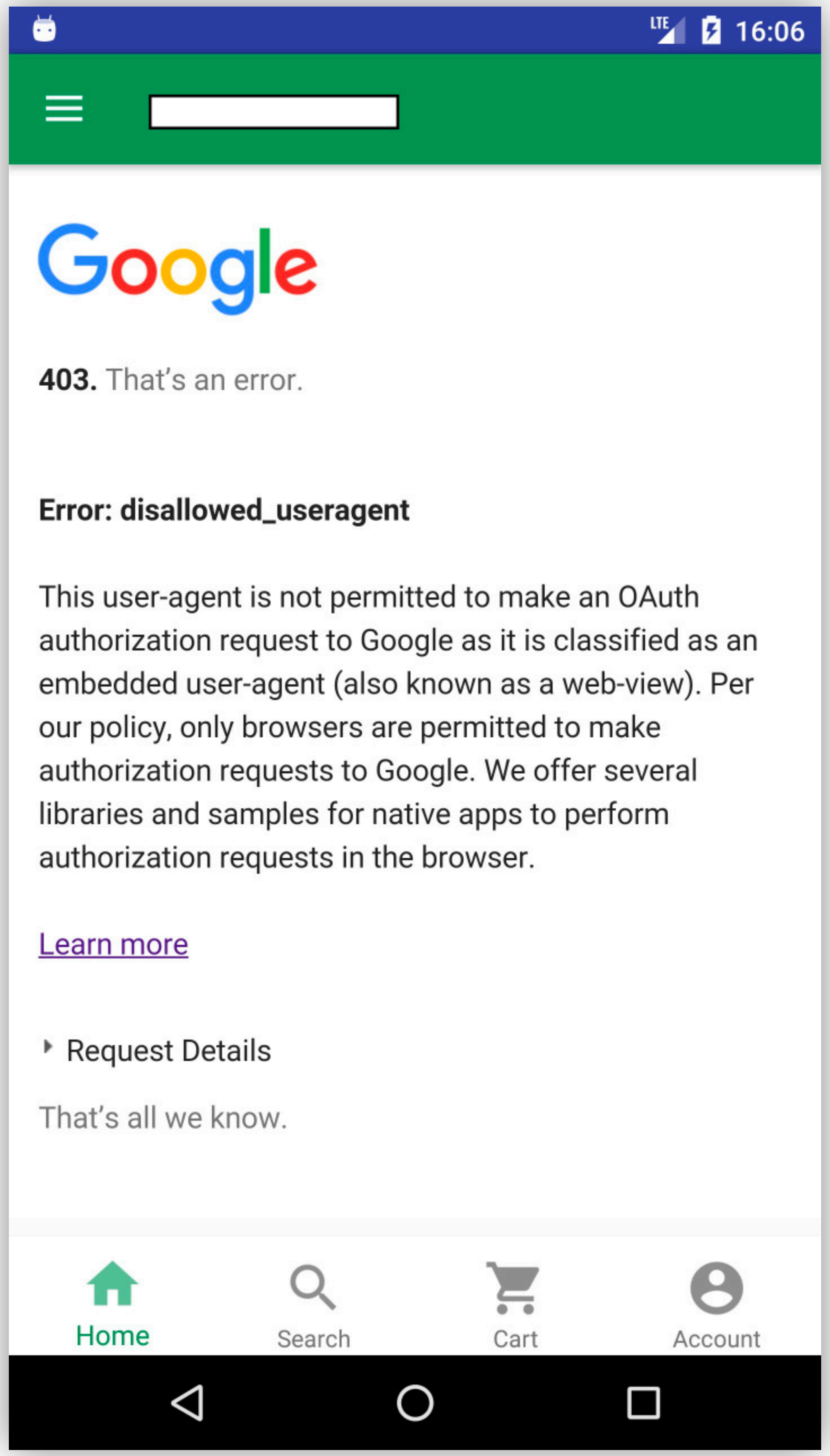
https://cloud.aaronpk.com
Contacts

iCloud

Google requires completing authentication in your web browser.

After authentication, setup will continue in Internet Accounts.

Cancel | Open Browser



Google

403. That's an error.

Error: disallowed_useragent

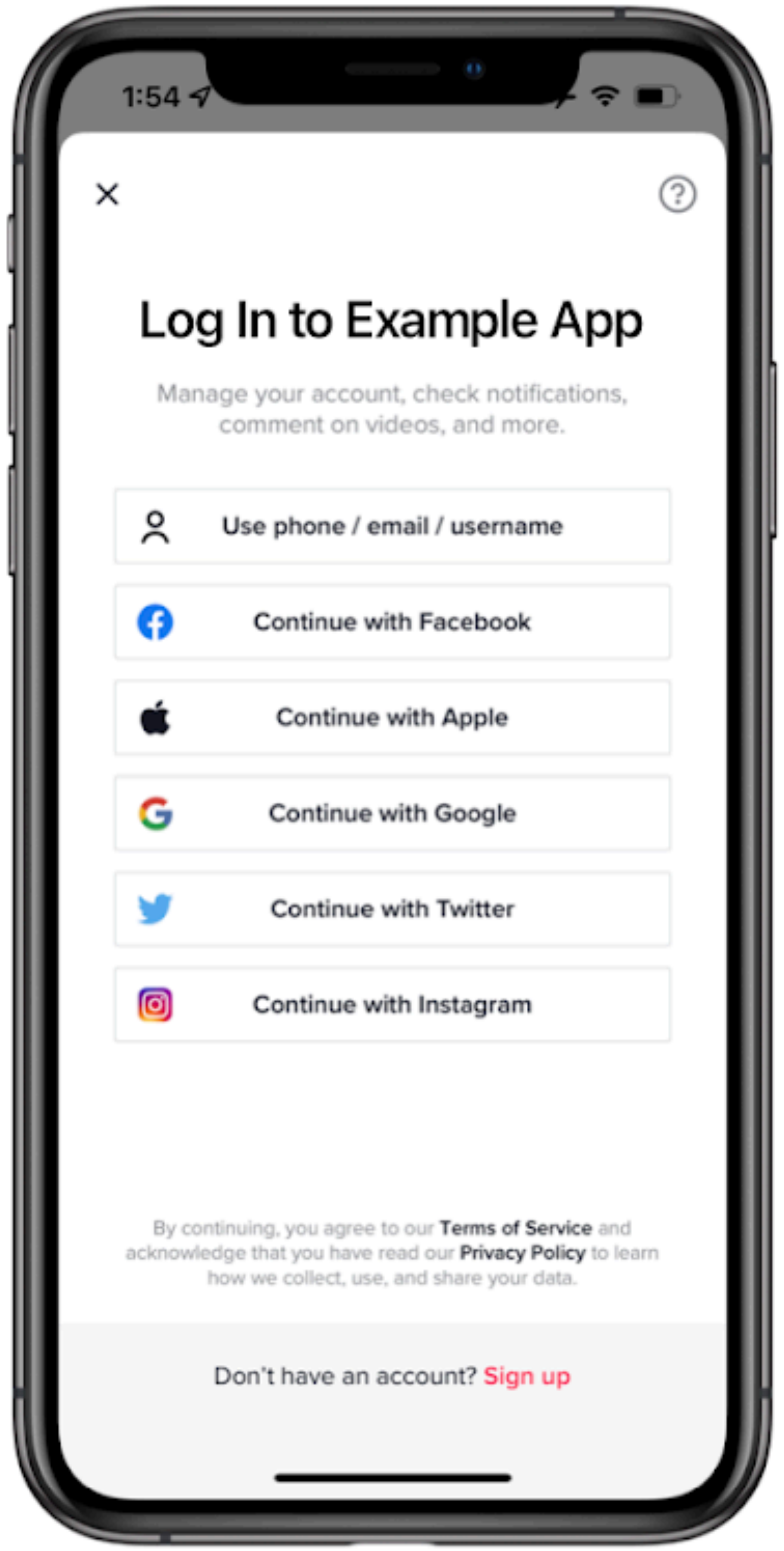
This user-agent is not permitted to make an OAuth authorization request to Google as it is classified as an embedded user-agent (also known as a web-view). Per our policy, only browsers are permitted to make authorization requests to Google. We offer several libraries and samples for native apps to perform authorization requests in the browser.

Learn more

▸ Request Details

That's all we know.

https://auth0.com/blog/google-blocks-oauth-requests-from-embedded-browsers/

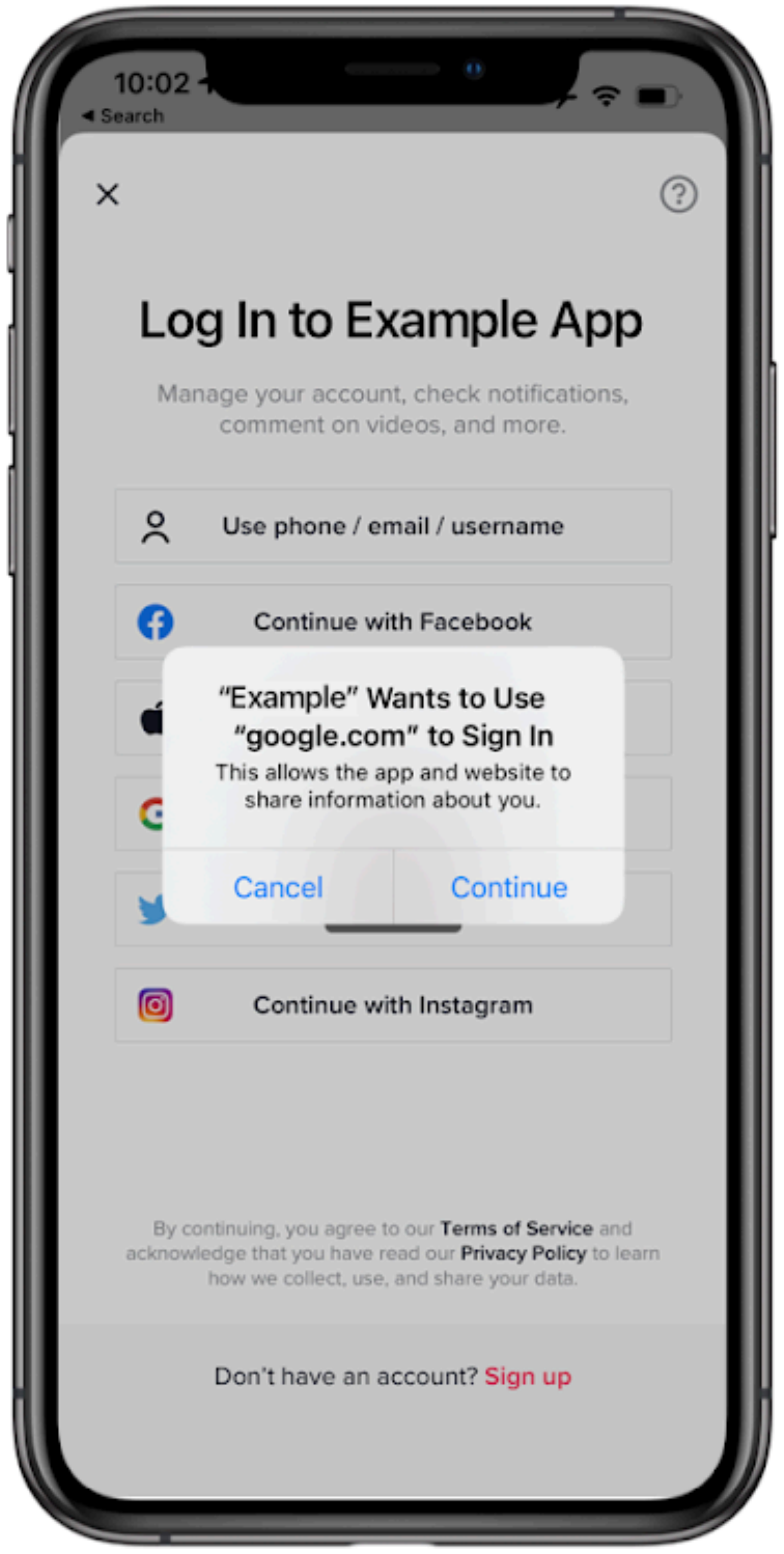


Log In to Example App

Manage your account, check notifications, comment on videos, and more.

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

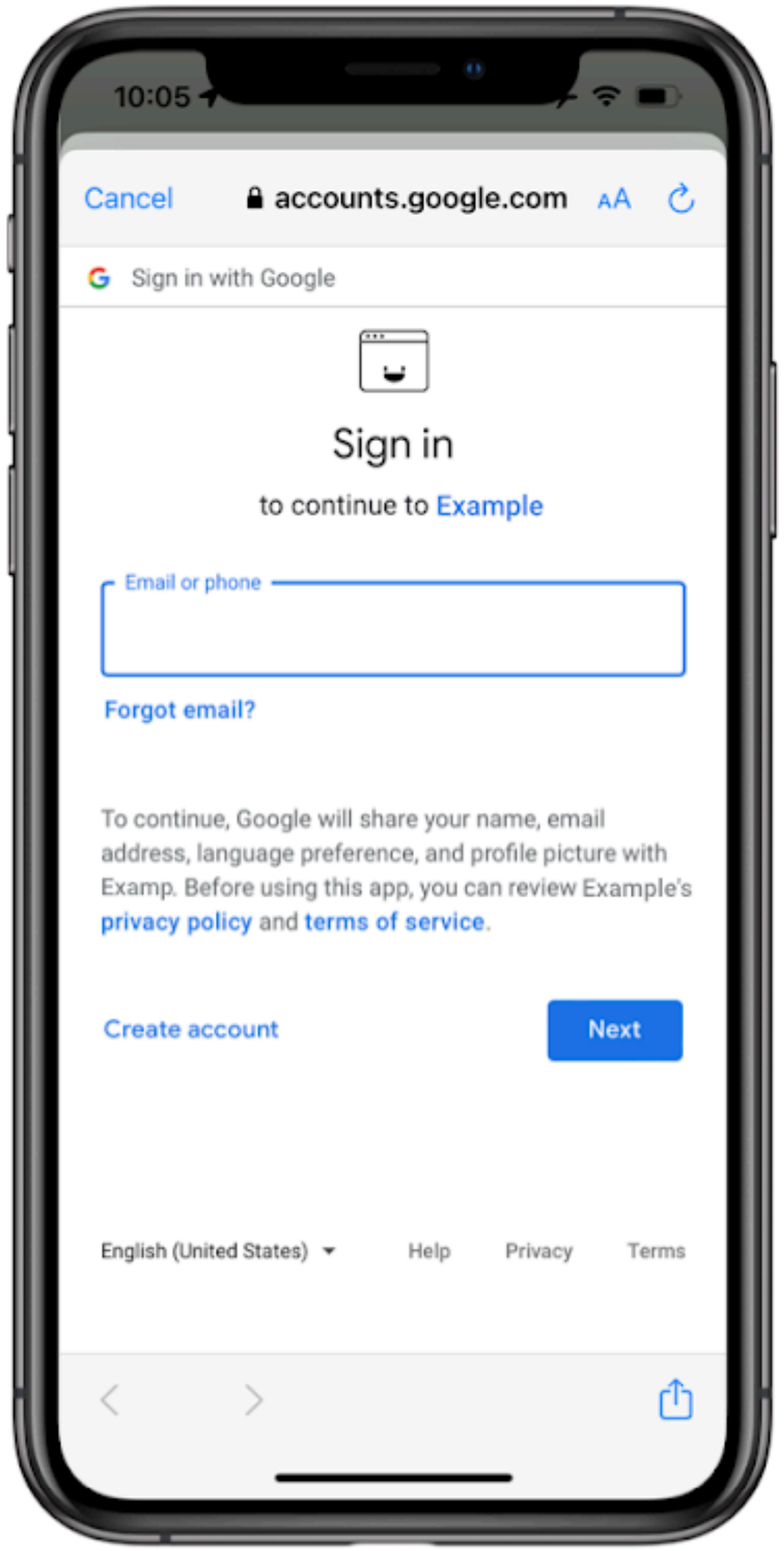
This allows the app and website to share information about you.

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

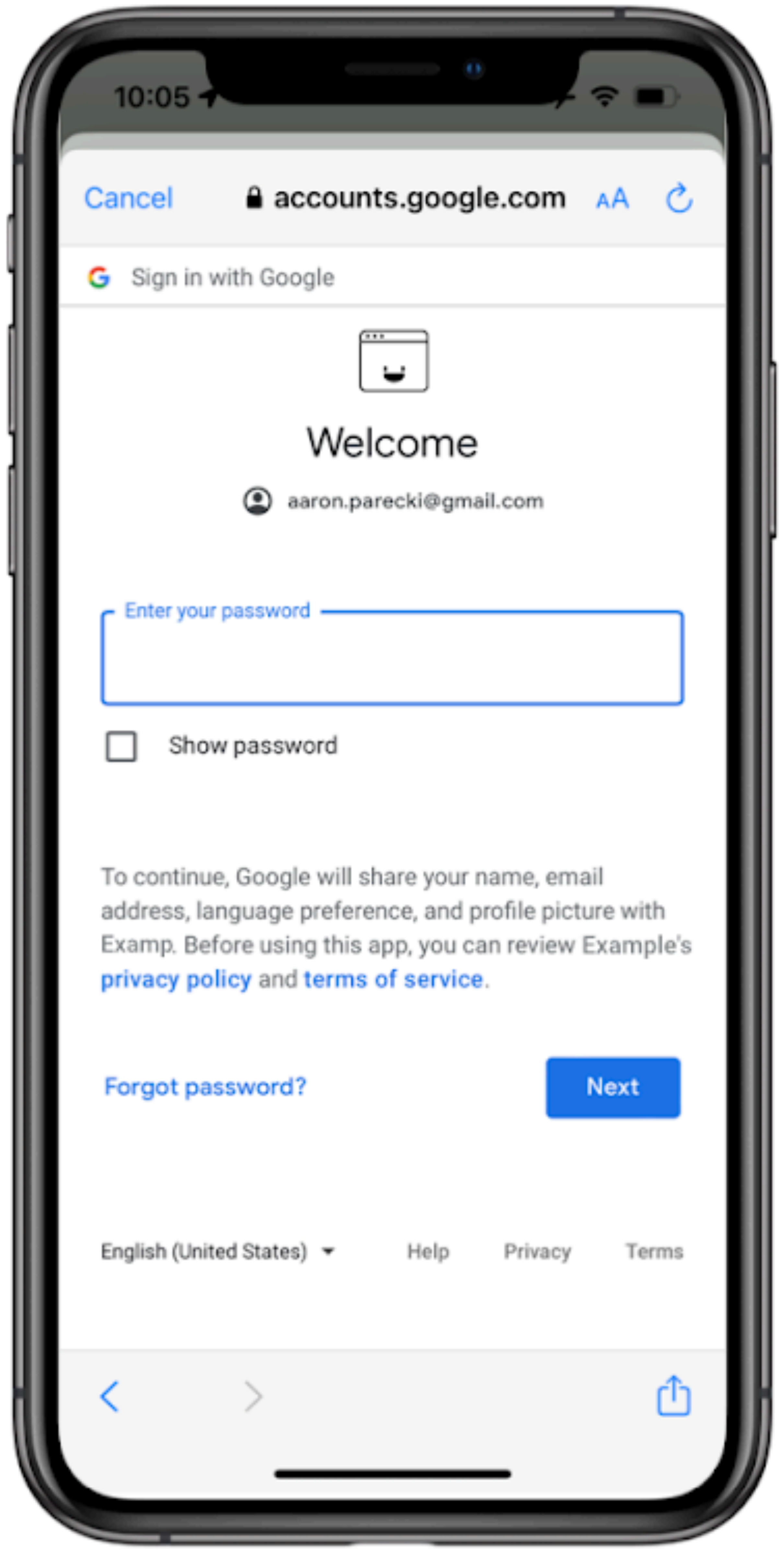
Forgot email?

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

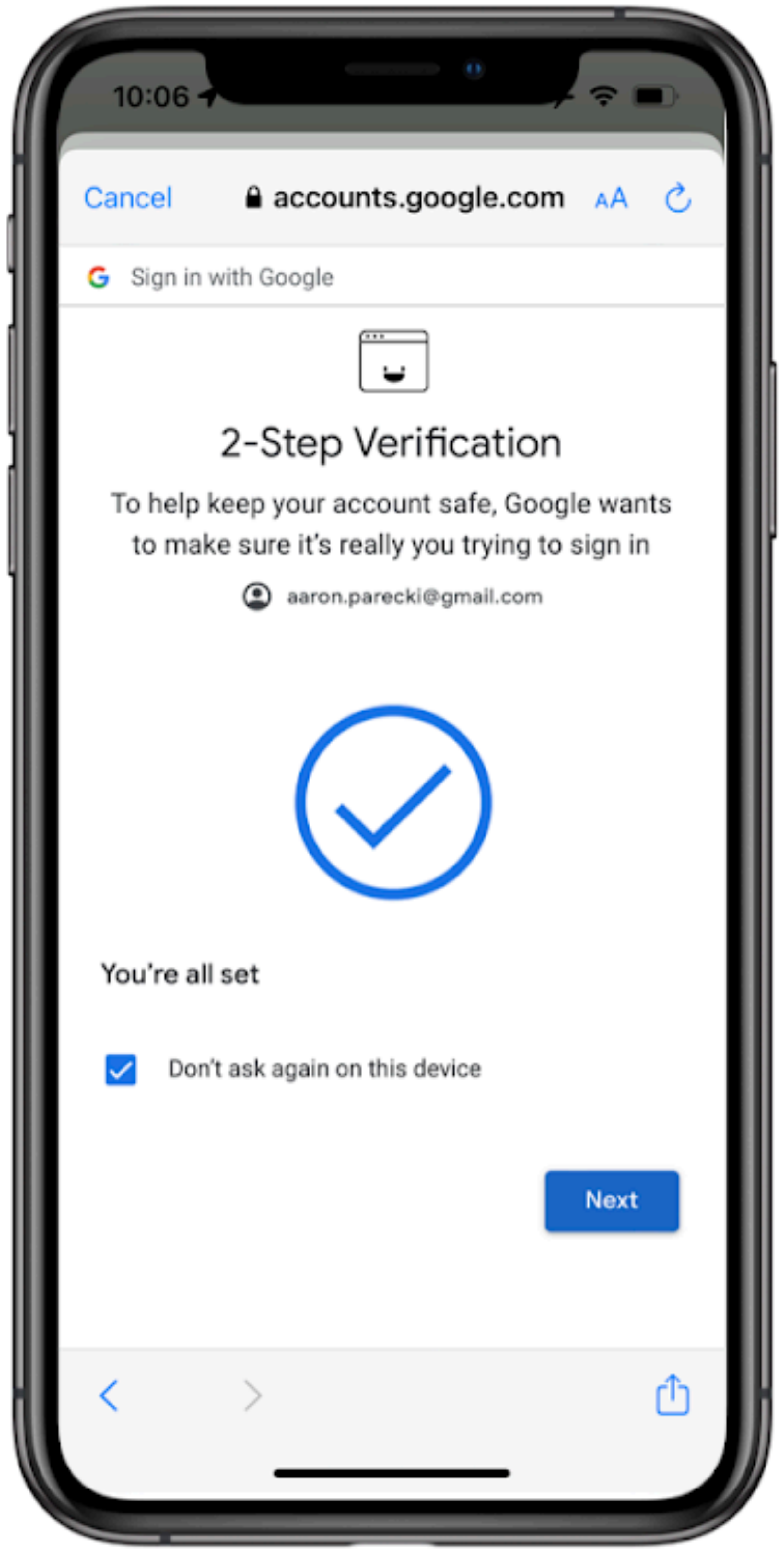
Enter your password

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

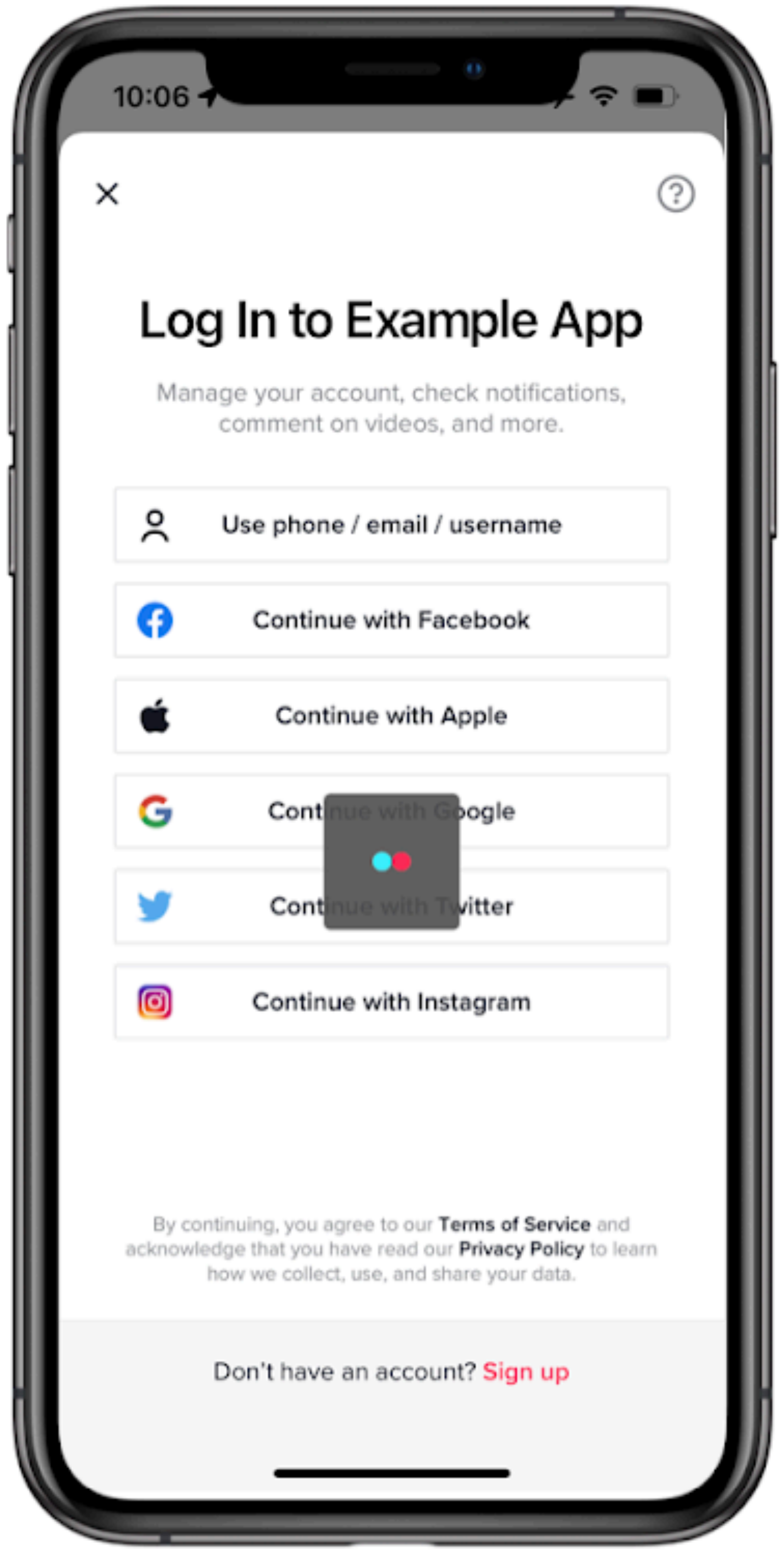
To help keep your account safe, Google wants to make sure it's really you trying to sign in

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

Cont[illegible]ogle

Cont[illegible]itter

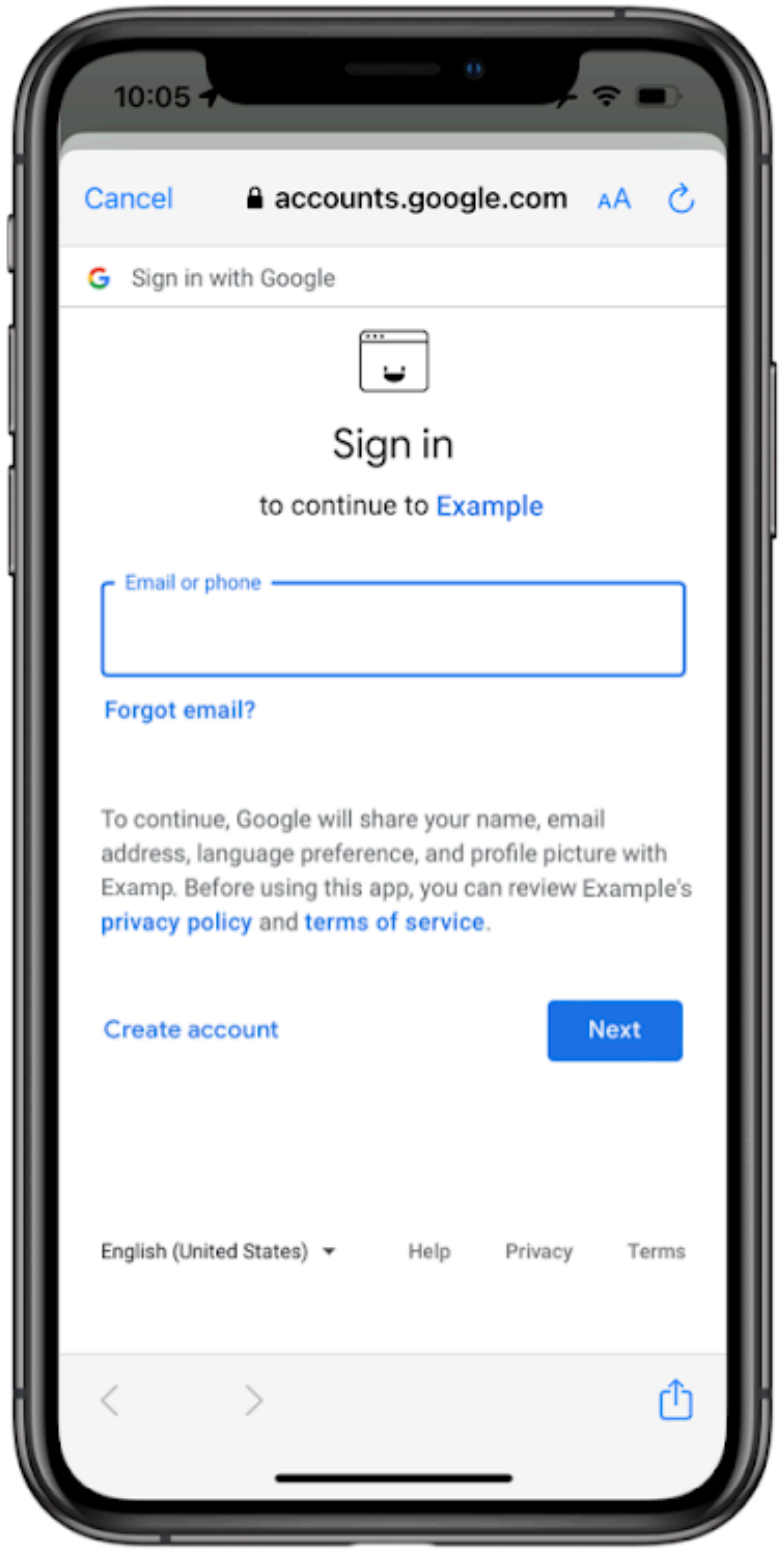
Continue with Instagram

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

System Browser (vs Web View)

- Platform-specific API to launch a browser
- The browser is not able to be observed or modified by the application
- Safe to enter passwords, phishing-resistant MFA, etc
- Domain name is visible in the popup browser



Sign in

to continue to Example

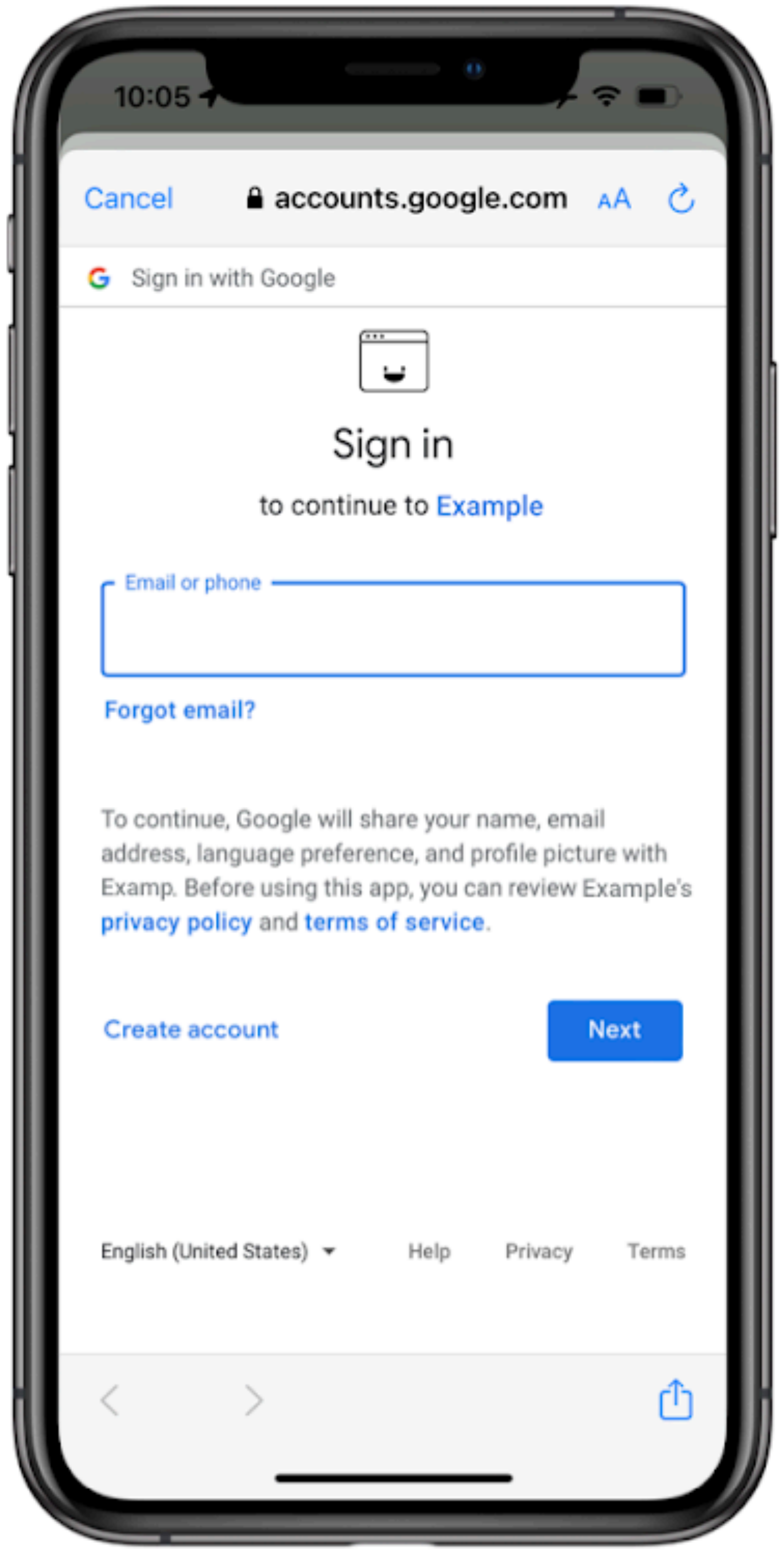
To continue, Google will share your name, email address, language preference, and profile picture with Examp. Before using this app, you can review Example's privacy policy and terms of service.

System Browser

✅ Good for security

✅ Good for third-party apps

❌ Bad UX for first-party apps



10:05

Cancel accounts.google.com AA

Sign in with Google

Sign in

to continue to Example

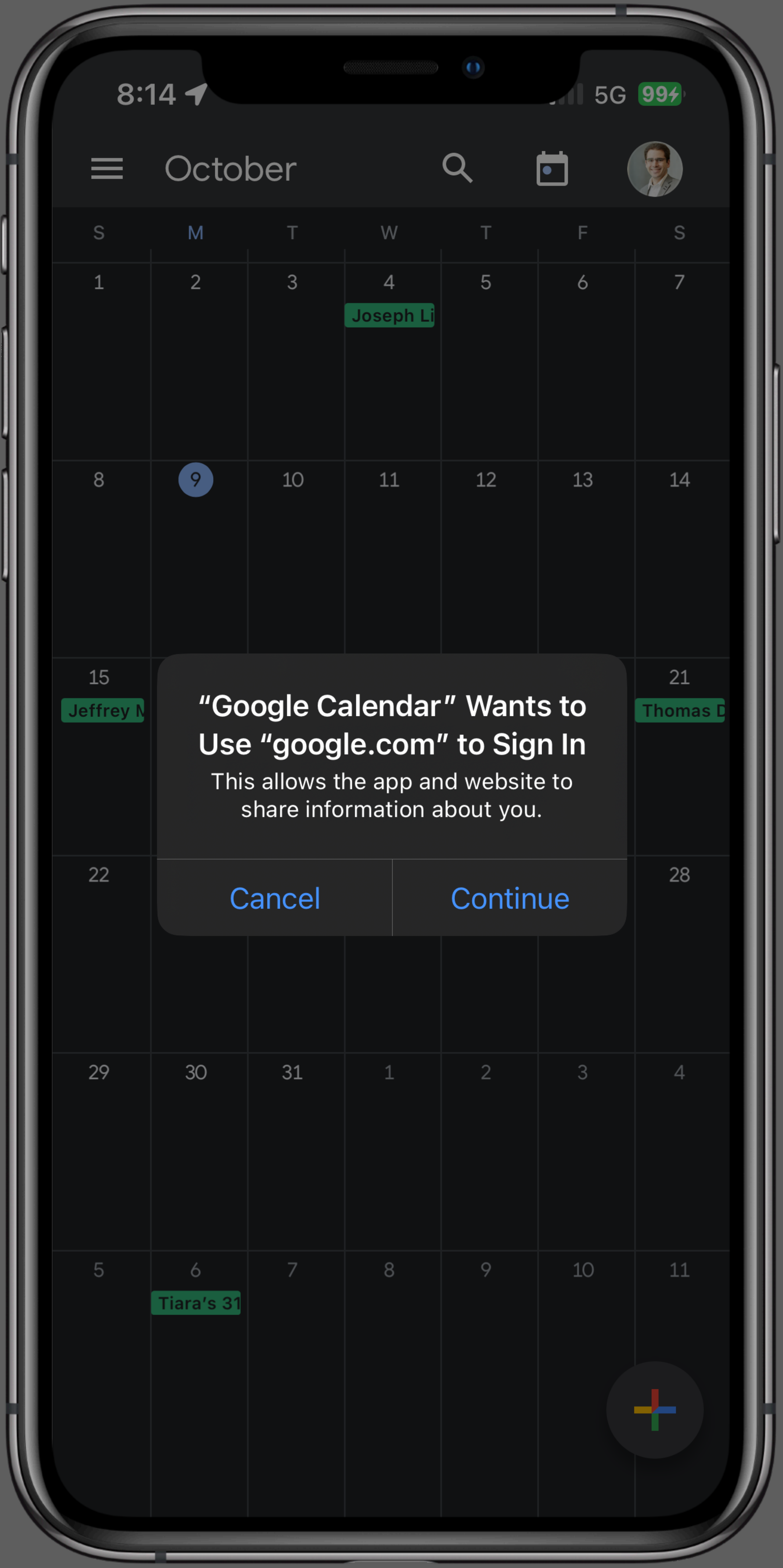
Email or phone

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



8:14 5G 99

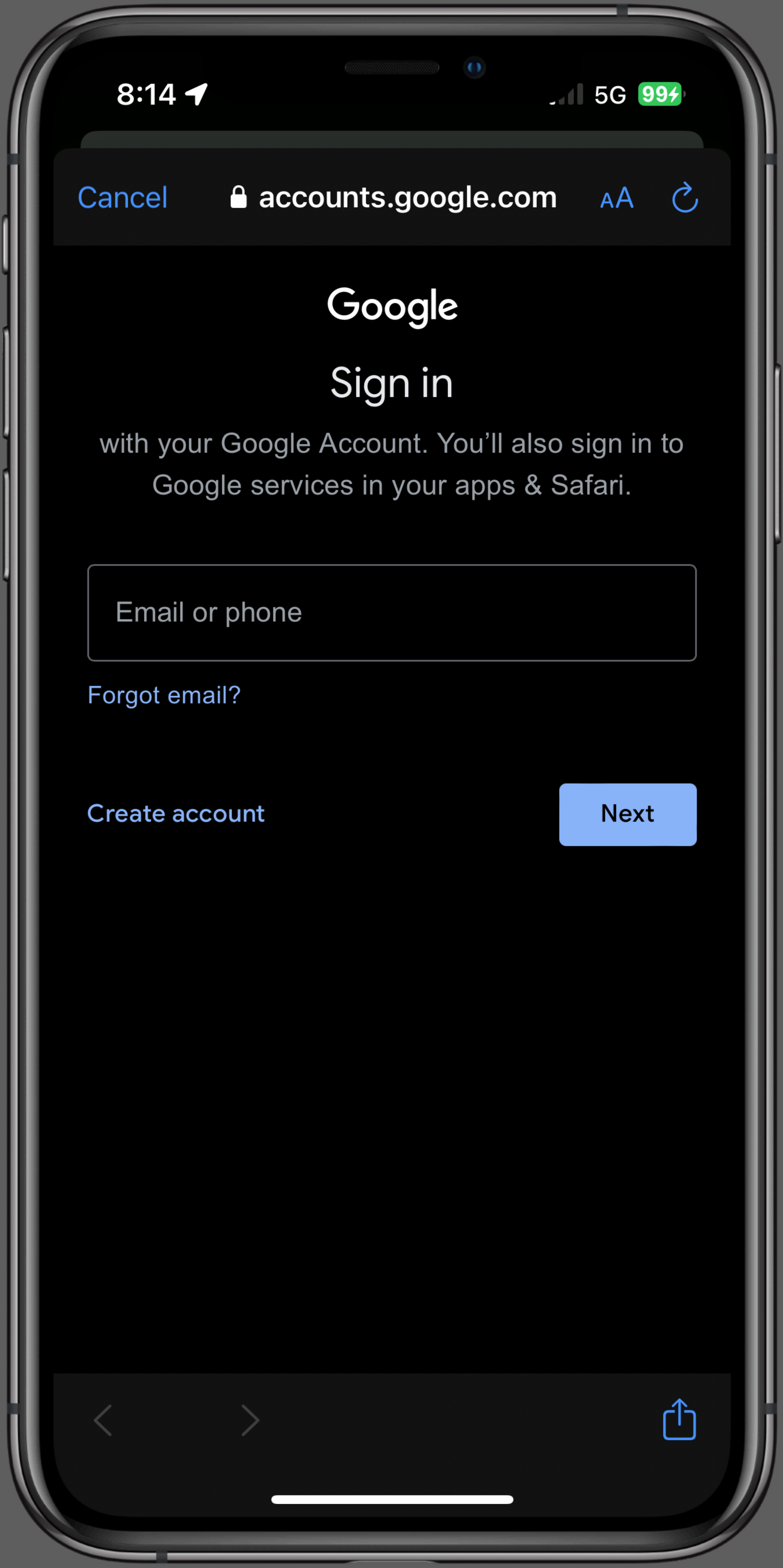
October

S M T W T F S

“Google Calendar” Wants to Use “google.com” to Sign In

This allows the app and website to share information about you.

Cancel Continue



8:14 5G 99

Cancel accounts.google.com

Google

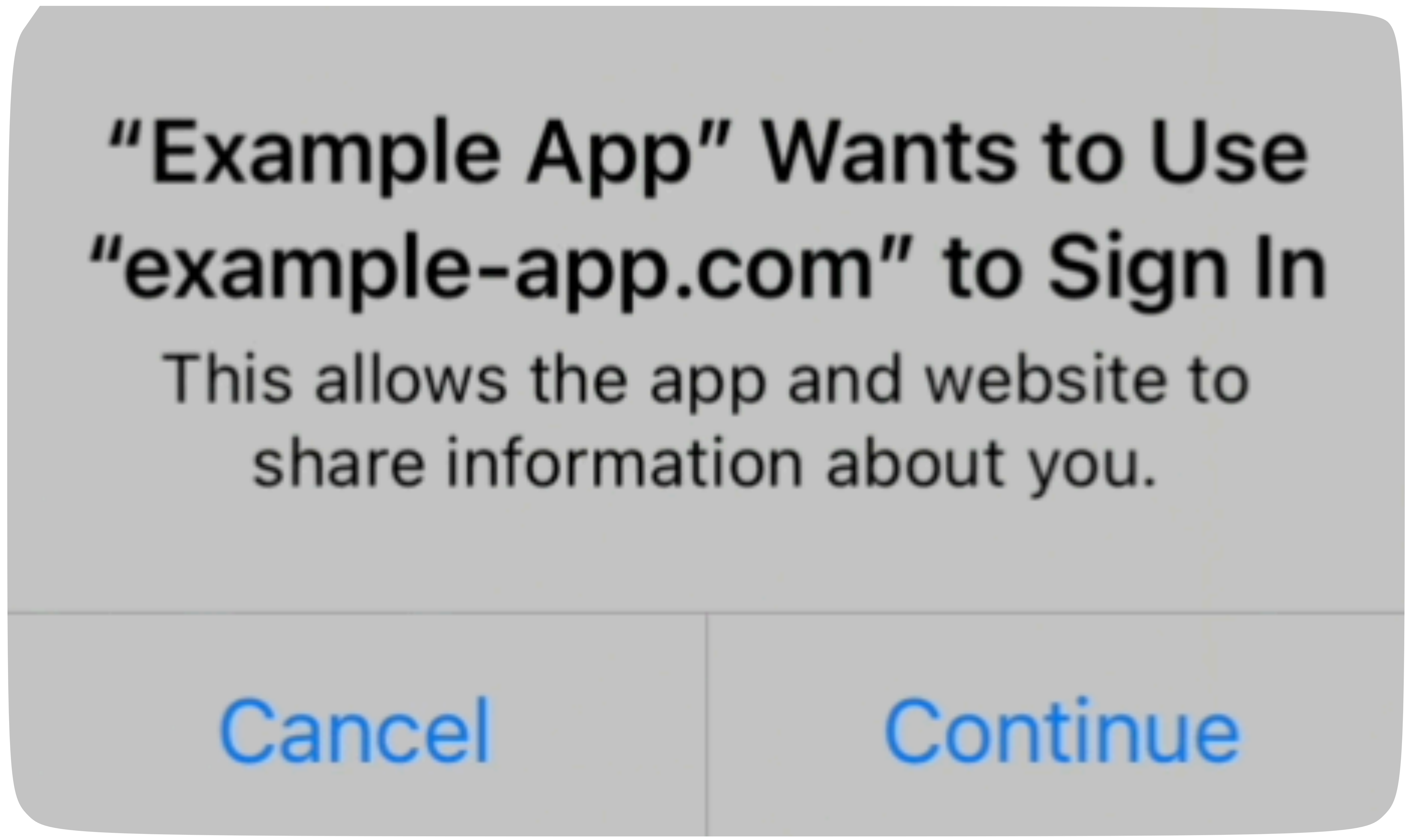
Sign in

with your Google Account. You’ll also sign in to Google services in your apps & Safari.

Email or phone

Forgot email?

Create account Next


“Example App” Wants to Use “example-app.com” to Sign In

This allows the app and website to share information about you.

MUST be treated as a public client

`native apps are classified as public clients, as defined by Section 2.1 of OAuth 2.0 [RFC6749]; they MUST be registered with the authorization server as such`

Public Clients

The application can't be deployed with a secret

JavaScript/Single-Page apps: "view source"
Native apps: decompile and extract strings

High score leaderboards

Player 1	9000
Player 4	7800
Player 2	4495
Player 8	2100
Player 5	700

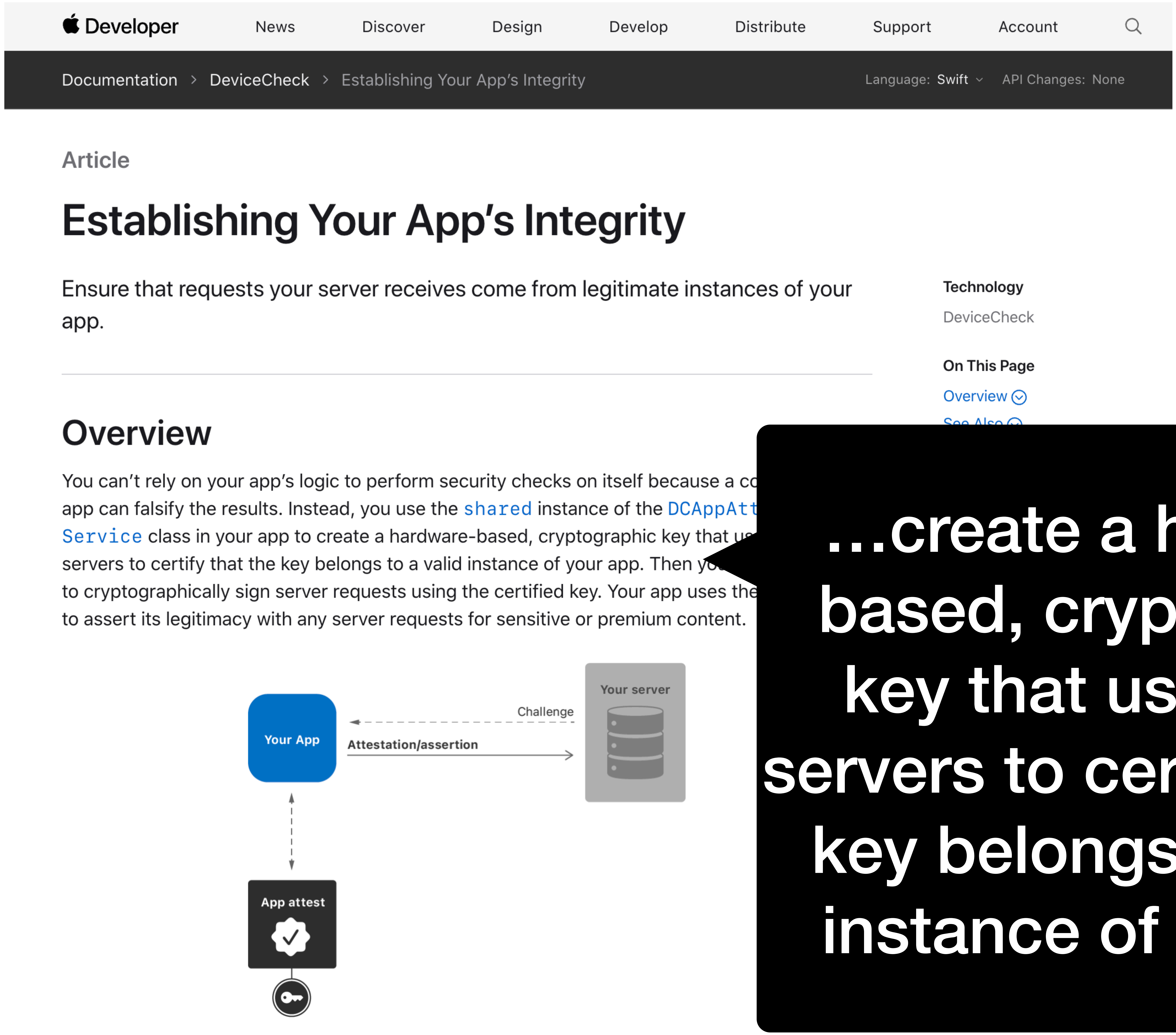
Mobile game reports new high score

```
POST https://api.game-server.example/score
  display_name=Hacker&
  score=99999999
```

Mobile game reports new high score with an access token

```
POST https://api.game-server.example/score
Authorization: Bearer XXXXXXXXXXXX

  score=99999999
```

“Is this request to the server being made by a legitimate instance of my application?”



Article

Establishing Your App's Integrity

Ensure that requests your server receives come from legitimate instances of your app.

Technology

DeviceCheck

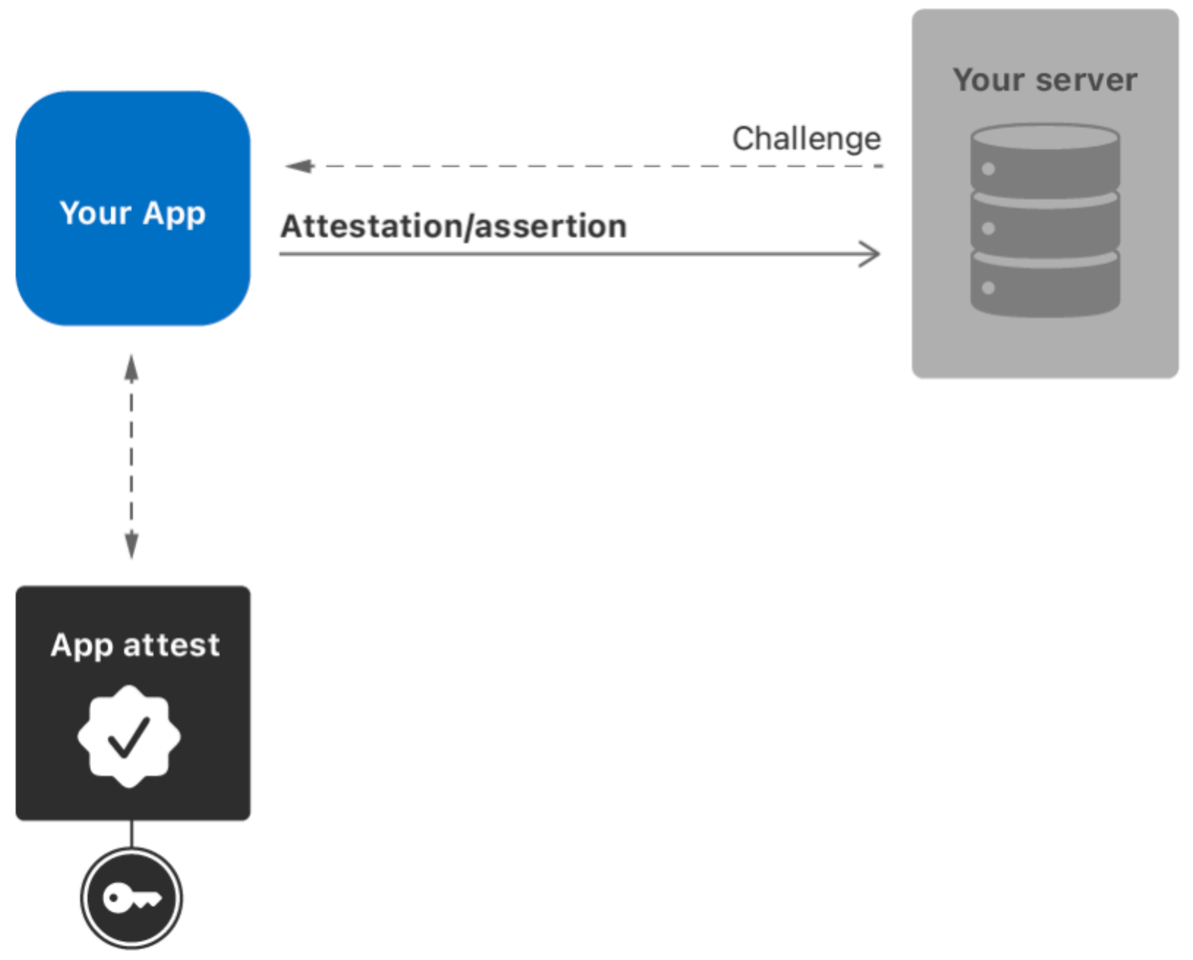
On This Page

Overview

See Also

Overview

You can't rely on your app's logic to perform security checks on itself because a co
app can falsify the results. Instead, you use the `shared` instance of the `DCAppAtt`
`Service` class in your app to create a hardware-based, cryptographic key that u
servers to certify that the key belongs to a valid instance of your app. Then yo
to cryptographically sign server requests using the certified key. Your app uses the
to assert its legitimacy with any server requests for sensitive or premium content.



...create a hardware-based, cryptographic key that uses Apple servers to certify that the key belongs to a valid instance of your app.

https://developer.apple.com/documentation/devicecheck/establishing-your-app-s-integrity?language=objc

developers Platform Android Studio Google Play Jetpack Kotlin Docs Games Search Language Sign in

Play Integrity API

The Play Integrity API helps protect your apps and games from potentially risky and fraudulent interactions, allowing you to respond with appropriate actions to reduce attacks and abuse such as fraud, cheating, and unauthorized access.

View documentation

Genuine app binary

Determine whether you're interacting with an unmodified binary that's recognized by Google Play.

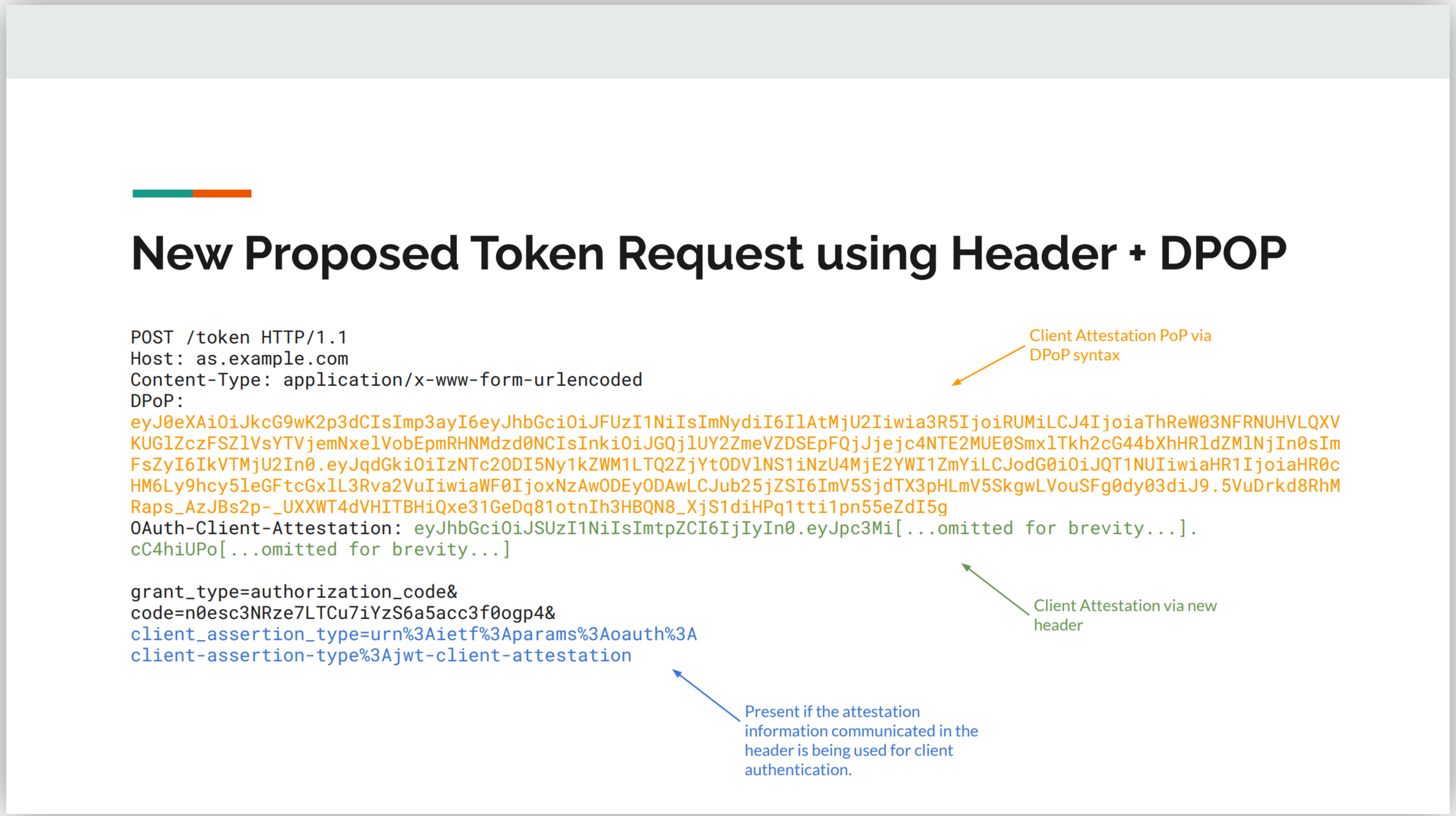
Genuine Play install

Determines whether the current user account has acquired the app or game legitimately, such as by installing or paying for it from Google Play.

Genuine Android device

Determine whether your app is running on a known, unmodified Android device powered by Google Play services.

https://developer.android.com/google/play/integrity



New Proposed Token Request using Header + DPOP

```
POST /token HTTP/1.1
Host: as.example.com
Content-Type: application/x-www-form-urlencoded
DPoP:
eyJ0eXAiOiJkcG9wK2p3dCIsImp3ayI6eyJhbGciOiJFUzI1NiIsImNydiI6IlAtMjU2Iiwia3R5IjoiRUMiLCJ4IjoiaThReW03NFRNUHVLQXV
KUGlZczFSZlVsYTVjemNxelVobEpmRHNMdzd0NCIsInkiOiJGQjlUY2ZmeVZDSEpFQjJjejc4NTE2MUE0SmxlTkh2cG44bXhHRldZMlNjIn0sIm
FsZyI6IkVTMjU2In0.eyJqdGkiOiIzNTc2ODI5Ny1kZWM1LTQ2ZjYtODVlNS1iNzU4MjE2YWI1ZmYiLCJodG0iOiJQT1NUIiwiaHR1IjoiaHR0c
HM6Ly9hcy5leGFtcGxlL3Rva2VuIiwiaWF0IjoxNzAwODEyODAwLCJub25jZSI6ImV5SjdTX3pHLmV5SkgwLVouSFg0dy03diJ9.5VuDrkd8RhM
Raps_AzJBs2p-_UXXWT4dVHITBHiQxe31GeDq81otnIh3HBQN8_XjS1diHPq1tti1pn55eZdI5g
OAuth-Client-Attestation: eyJhbGciOiJSUzI1NiIsImtpZCI6IjIyIn0.eyJpc3Mi[...omitted for brevity...].
cC4hiUPo[...omitted for brevity...]

grant_type=authorization_code&
code=n0esc3NRze7LTCu7iYzS6a5acc3f0ogp4&
client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3A
client-assertion-type%3Ajwt-client-attestation
```

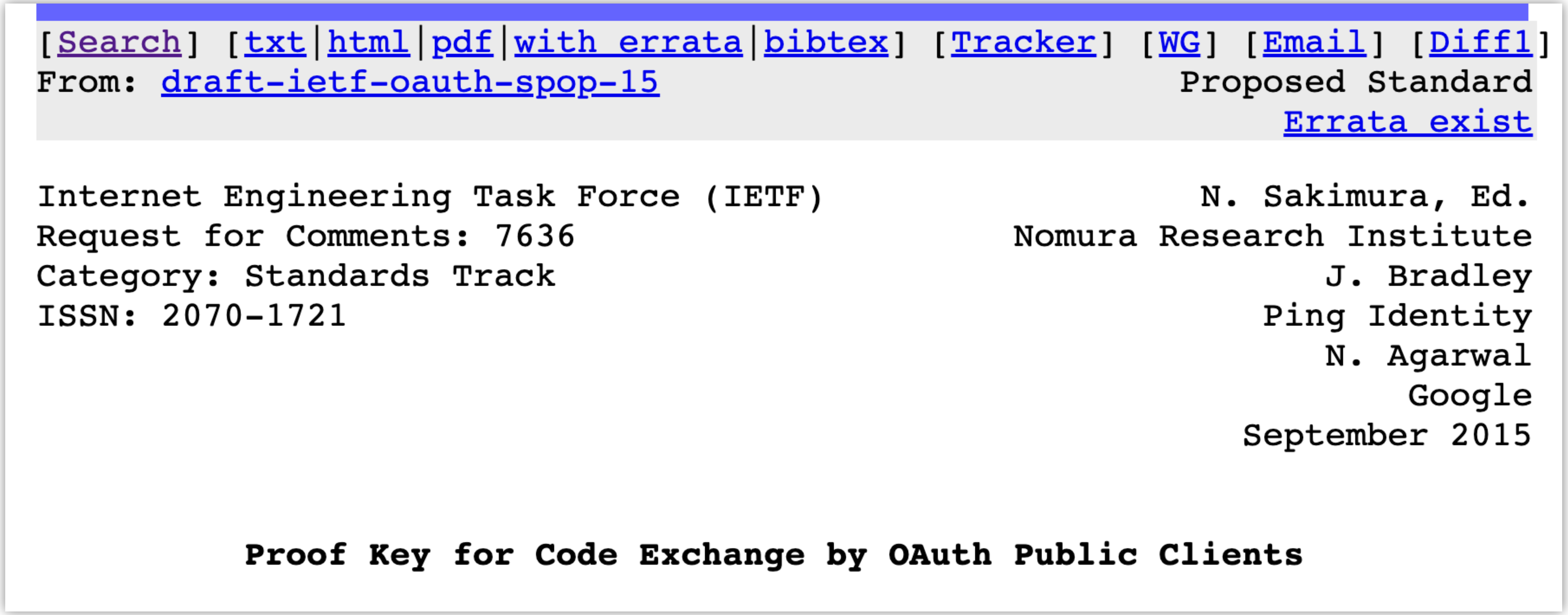
Client Attestation PoP via DPoP syntax

Client Attestation via new header

Present if the attestation information communicated in the header is being used for client authentication.

via OAuth 2.0 Attestation-Based Client Authentication presented at IETF 119

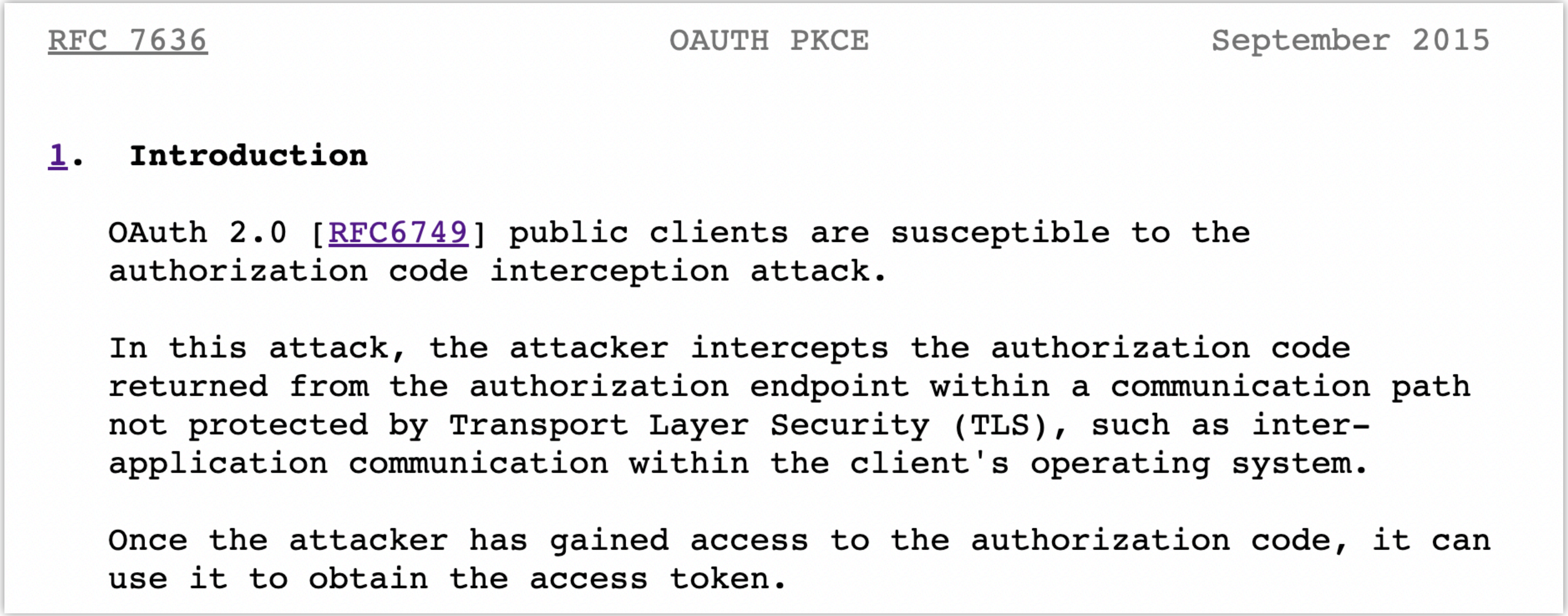
PKCE bOPC?



[Search] [txt|html|pdf|with errata|bibtex] [Tracker] [WG] [Email] [Diff1]
From: draft-ietf-oauth-spop-15 Proposed Standard
Errata exist

```
Internet Engineering Task Force (IETF)                  N. Sakimura, Ed.
Request for Comments: 7636                 Nomura Research Institute
Category: Standards Track                                 J. Bradley
ISSN: 2070-1721                                        Ping Identity
                                                          N. Agarwal
                                                              Google
                                                      September 2015
```

Proof Key for Code Exchange by OAuth Public Clients


RFC 7636 OAUTH PKCE September 2015

1. Introduction

OAuth 2.0 [RFC6749] public clients are susceptible to the authorization code interception attack.

In this attack, the attacker intercepts the authorization code returned from the authorization endpoint within a communication path not protected by Transport Layer Security (TLS), such as inter-application communication within the client's operating system.

Once the attacker has gained access to the authorization code, it can use it to obtain the access token.

```
POST /token

client_id=XXXXX
&authorization_code=XXXXX
```

```
POST /token

client_id=XXXXX
&client_secret=XXXXX
&authorization_code=XXXXX
```

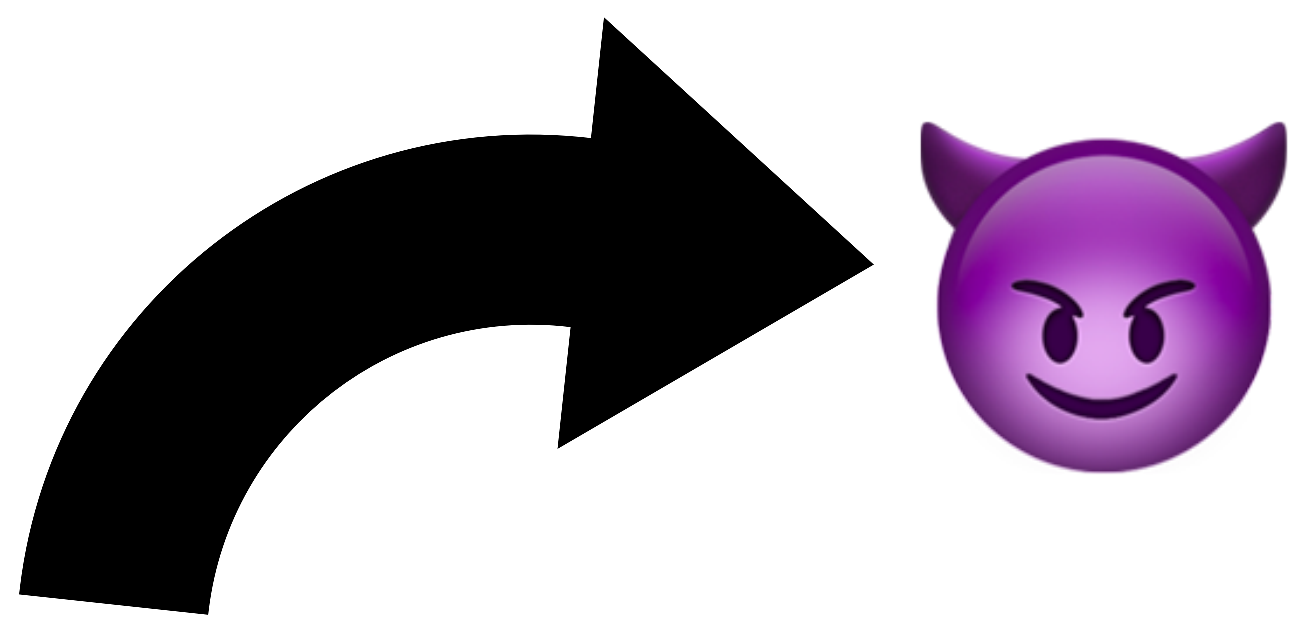
```
POST /token

client_id=XXXXX
&code_verifier=XXXX
&authorization_code=XXXXX
```

PKCE was recommended for mobile apps, which can’t use a secret

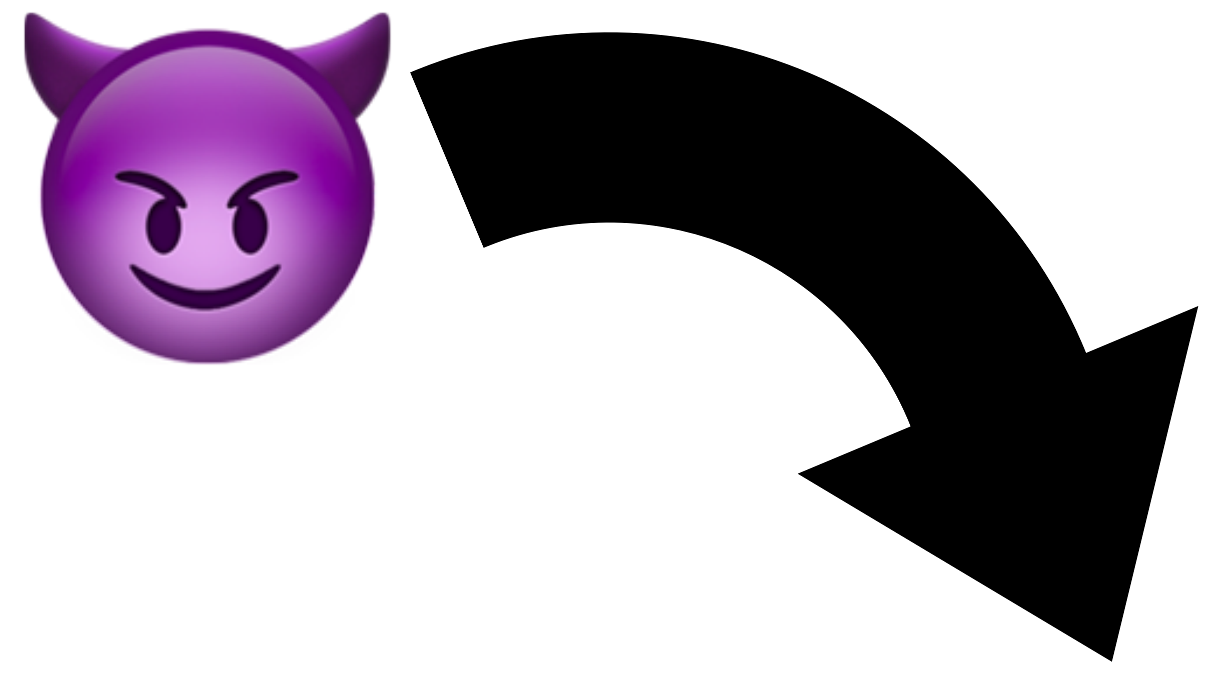
Is PKCE is a replacement for a client secret?

NO

Interception



/redirect?code=XXXXX



Injection

/redirect?code=XXXXX

Redirect URLs

To fully support this best practice, authorization servers MUST offer at least the three redirect URI options described in the following subsections to native apps.

Redirect URLs in Mobile Apps

Custom URL Scheme

```
example-app://redirect?
  code=AUTHORIZATION_CODE_HERE&
  state=1234zyx
```

App-Claimed URL Pattern

```
https://example-app.com?
  code=AUTHORIZATION_CODE_HERE&
  state=1234zyx
```

Redirect URLs in Mobile Apps

Custom URL Scheme

No registry
No validation
Any app can claim any URL scheme
Sometimes undefined behavior if multiple apps use the same URL scheme

App-Claimed URL Pattern

aka "Universal Links" on iOS
Requires proving ownership of the domain name by the app publisher
Verified on app install and sometimes periodically afterwards

Redirect URLs in Mobile Apps

But...

none of this really matters

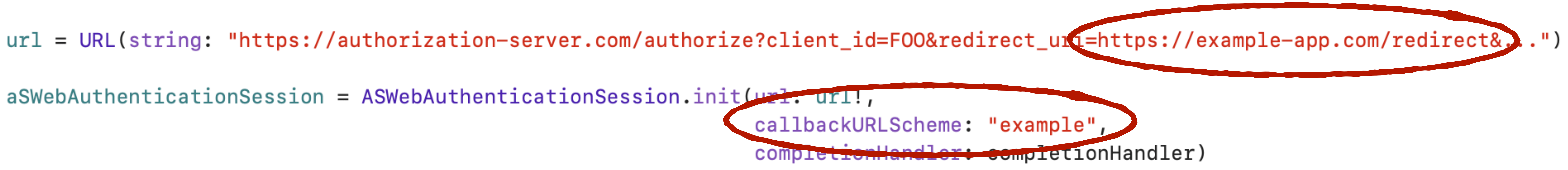
Redirect URLs

+

Use the System Browser

+

follow best practices



```
url = URL(string: "https://authorization-server.com/authorize?client_id=FOO&redirect_uri=https://example-app.com/redirect&...")

aSWebAuthenticationSession = ASWebAuthenticationSession.init(url: url!,
                                                             callbackURLScheme: "example",
                                                             completionHandler: completionHandler)
```

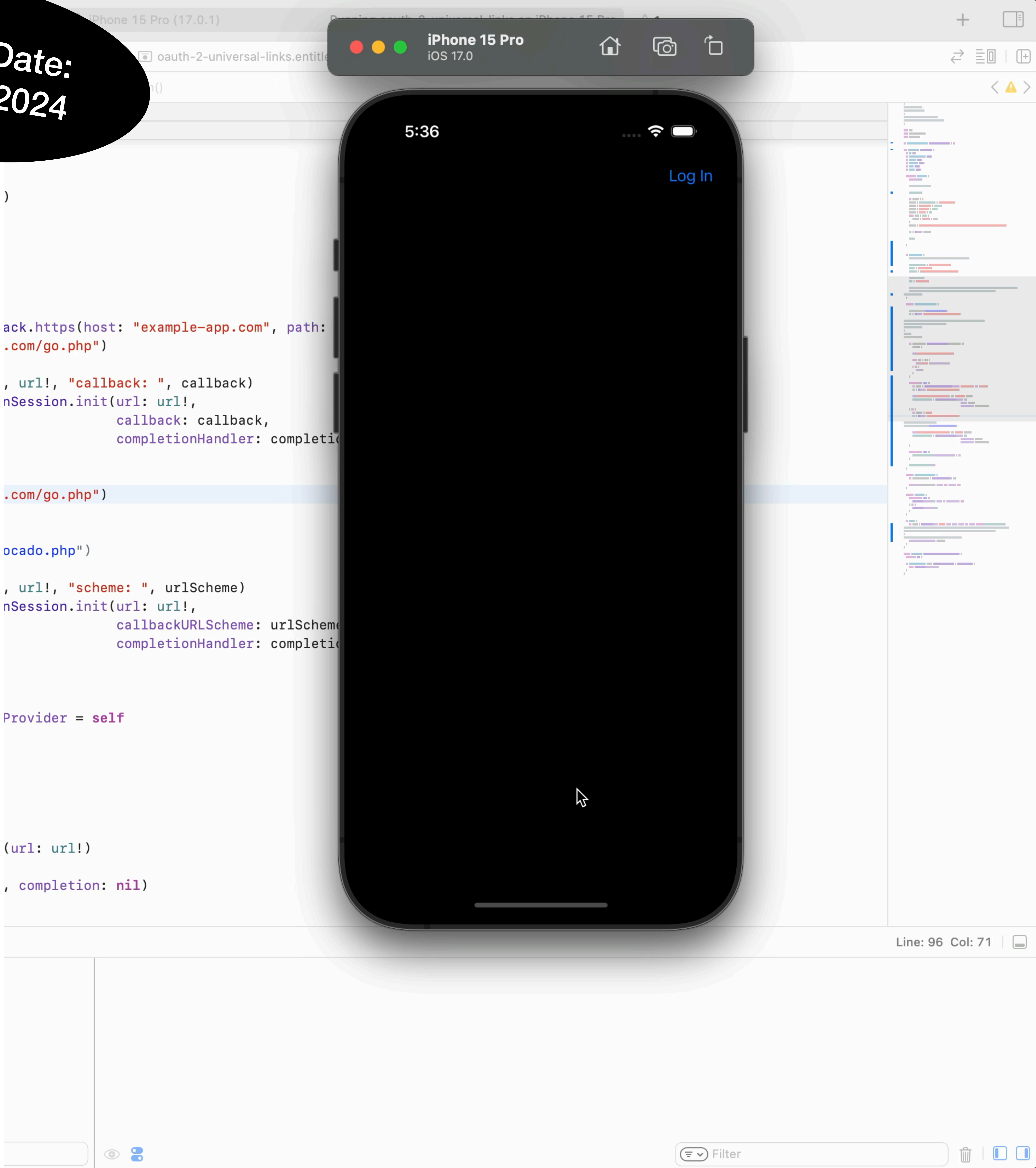
- Include https redirect URI in authorization request
- Custom URL scheme is still required to launch ASWebAuthenticationSession

Before iOS 17.4

No User Interaction

Release Date:
March 5, 2024

- Include https redirect URI in authorization request
- Custom URL scheme is still required to launch ASWebAuthenticationSession
- (User already is logged in)
- Universal Link is not triggered
- Browser ends up at redirect URL loaded in the browser
- Native app has no way to recover

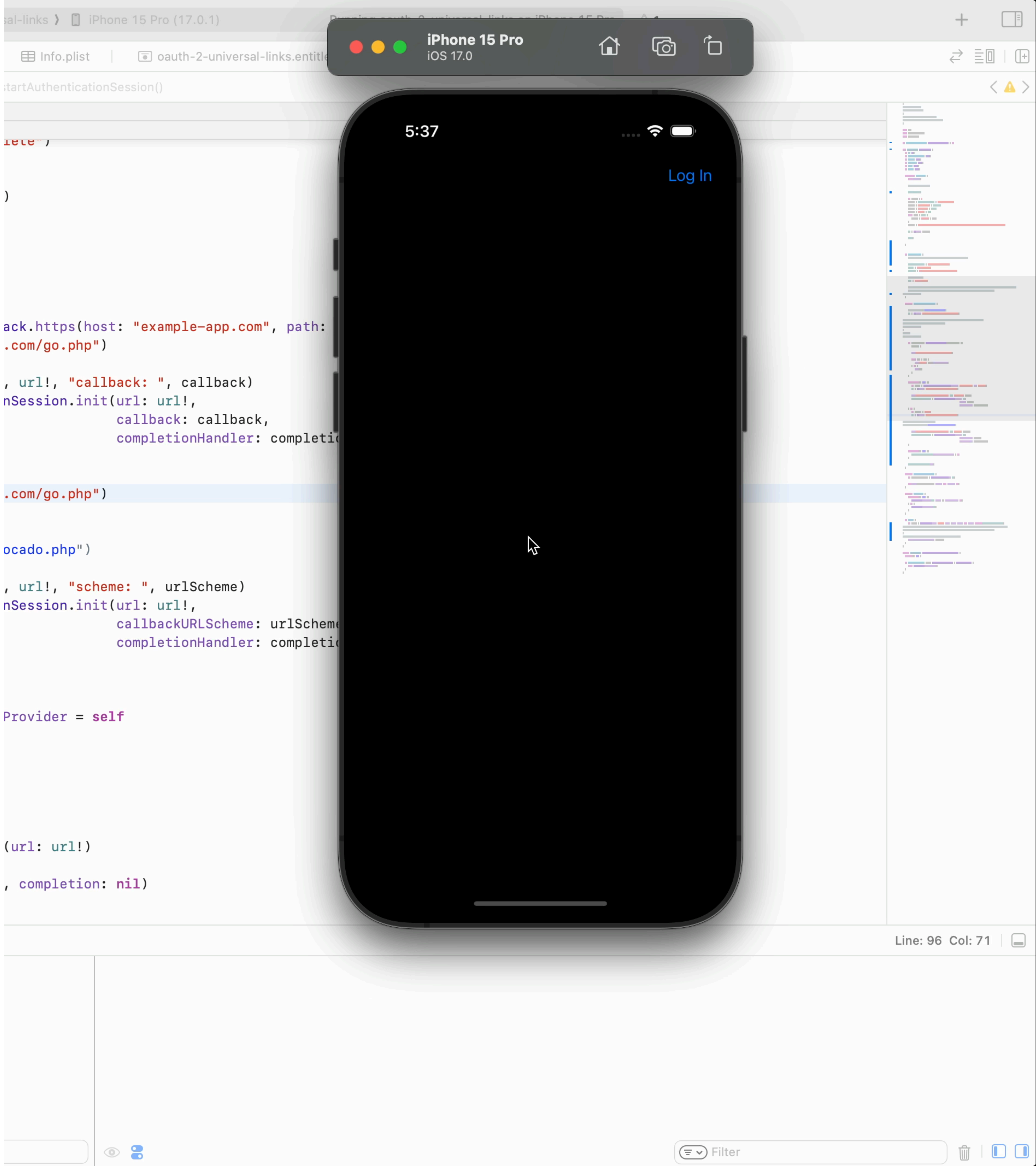


```
)

ack.https(host: "example-app.com", path:
.com/go.php")

, url!, "callback: ", callback)
nSession.init(url: url!,
              callback: callback,
              completionHandler: completi

.com/go.php")

ocado.php")

, url!, "scheme: ", urlScheme)
nSession.init(url: url!,
              callbackURLScheme: urlScheme
              completionHandler: completi

Provider = self

(url: url!)

, completion: nil)
```

Before iOS 17.4

With User Interaction

- Include https redirect URI in authorization request
- Custom URL scheme is still required to launch ASWebAuthenticationSession
- (User already is logged in)
- Universal Link is triggered
- iOS runs the Universal Link callback
- Native app has to dismiss the active ASWebAuthenticationSession to resume



```
tartAuthenticationSession()

lete")

)

ack.https(host: "example-app.com", path:
.com/go.php")

, url!, "callback: ", callback)
nSession.init(url: url!,
              callback: callback,
              completionHandler: completi

.com/go.php")

ocado.php")

, url!, "scheme: ", urlScheme)
nSession.init(url: url!,
              callbackURLScheme: urlScheme
              completionHandler: completi

Provider = self

(url: url!)

, completion: nil)
```

ASWebAuthenticationSession in iOS 17.4

- Adds `ASWebAuthenticationSession.Callback`
- Takes an https URL that is validated the same way as Universal Links

`ASWebAuthenticationSession` in iOS 17.4

```
let callback = ASWebAuthenticationSession.Callback.https(host: "example-app.com",
                                                          path: "/redirect")

url = URL(string: "https://authorization-server.com/authorize")



print("Starting ASWebAuthenticationSession to ", url!, "callback: ", callback)

aSWebAuthenticationSession = ASWebAuthenticationSession.init(url: url!,
                                                        callback: callback,
                                               completionHandler: completionHandler)
```

`ASWebAuthenticationSession` in iOS 17.4

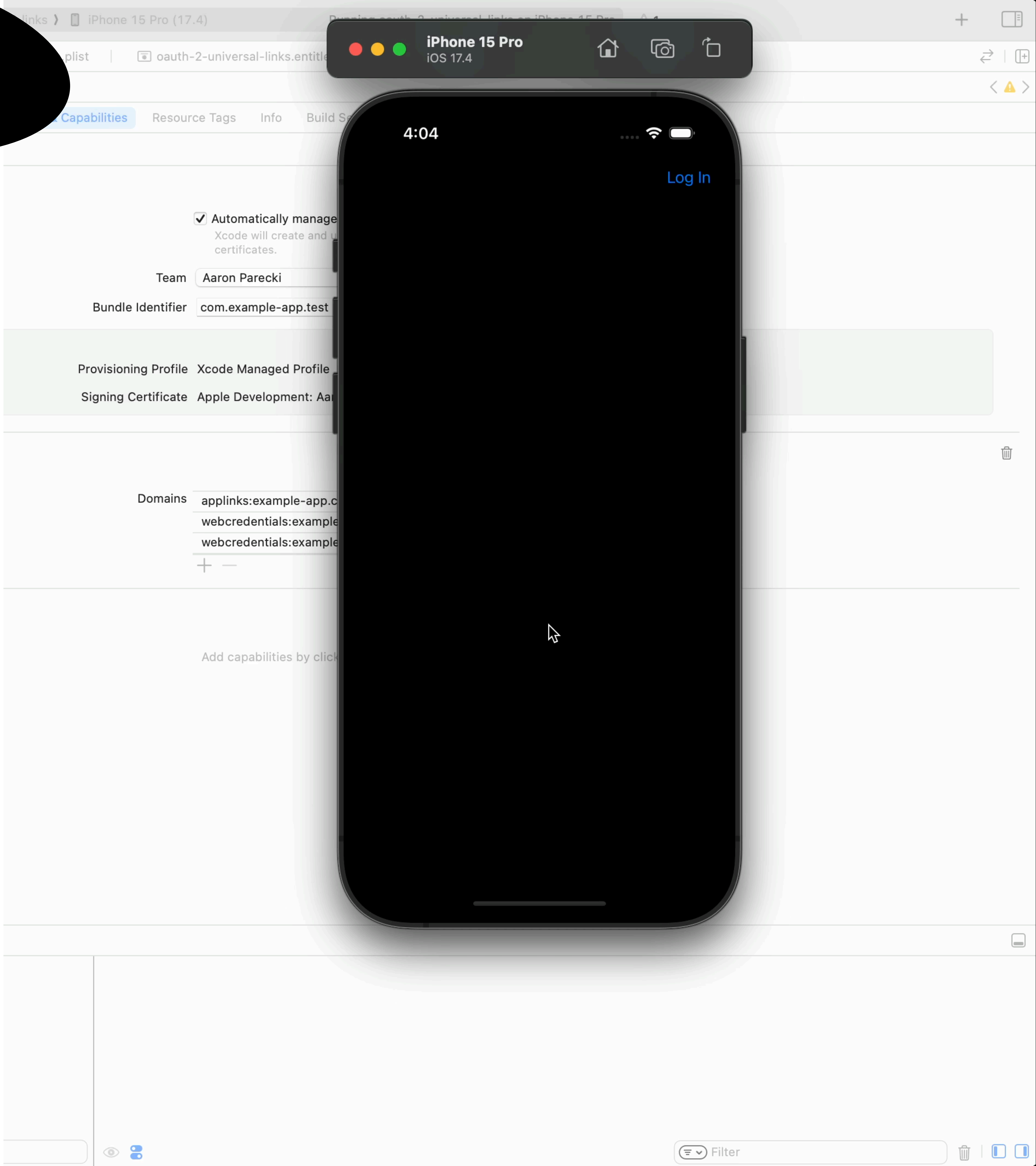
Attempting to use another app’s Universal Link as redirect URL

```
ERROR:  The operation couldn’t be completed. Application with
identifier com.example-app.test is not associated with domain
avocado.lol. Using HTTPS callbacks requires Associated Domains
using the `webcredentials` service type for avocado.lol.
```

After iOS 17.4

With User Interaction

- Universal Link binding is enforced
- iOS runs the ASWebAuthenticationSession as expected



☑ Automatically manage
Xcode will create and u
certificates.

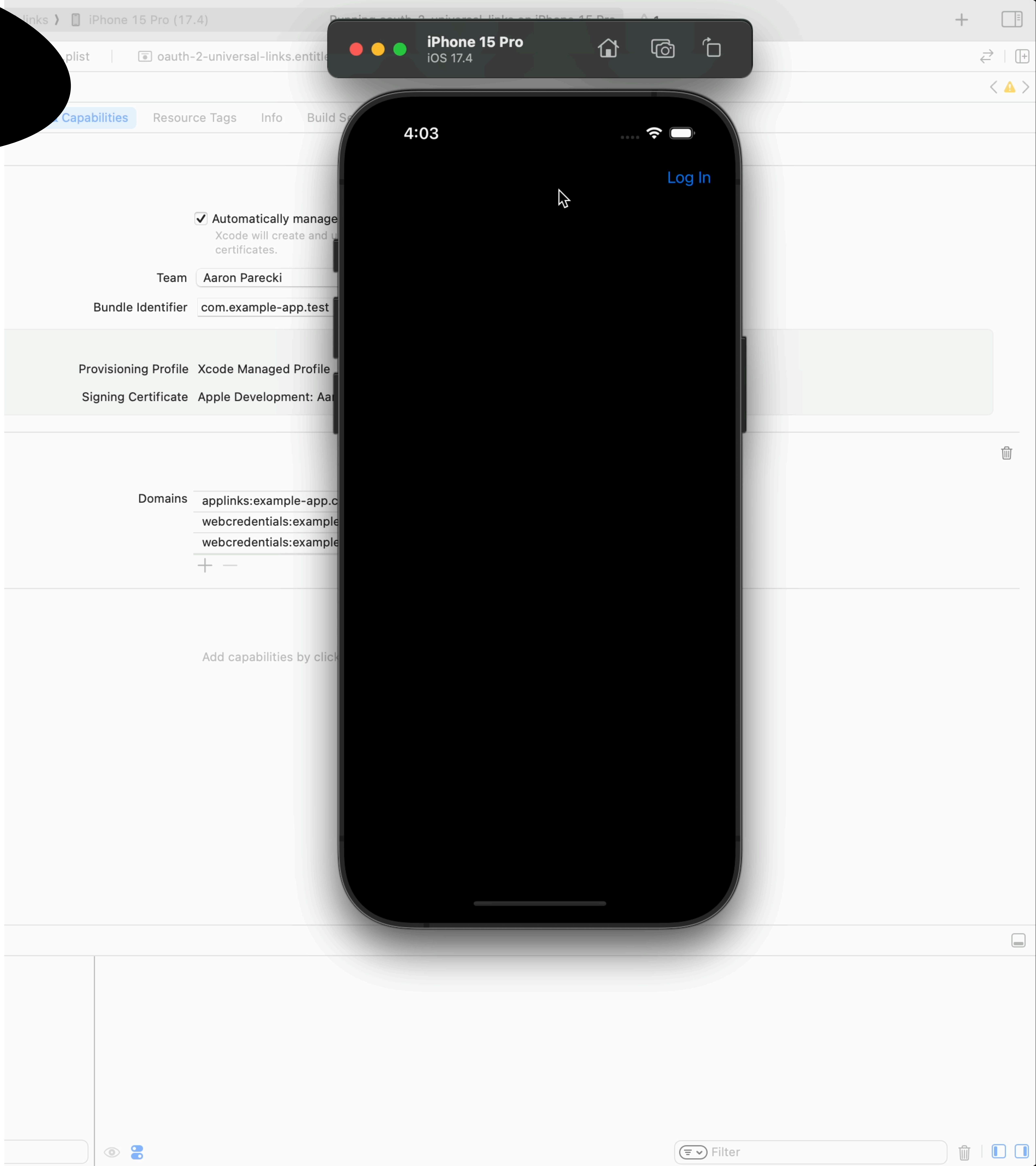
Team	Aaron Parecki
Bundle Identifier	com.example-app.test
Provisioning Profile	Xcode Managed Profile
Signing Certificate	Apple Development: Aa
Domains	applinks:example-app.c
	webcredentials:example
	webcredentials:example

Release Date: March 5, 2024

After iOS 17.4

No User Interaction

- No change from previous example with user interaction



Automatically manage
Xcode will create and u certificates.

Team Aaron Parecki

Bundle Identifier com.example-app.test

Provisioning Profile Xcode Managed Profile

Signing Certificate Apple Development: Aa

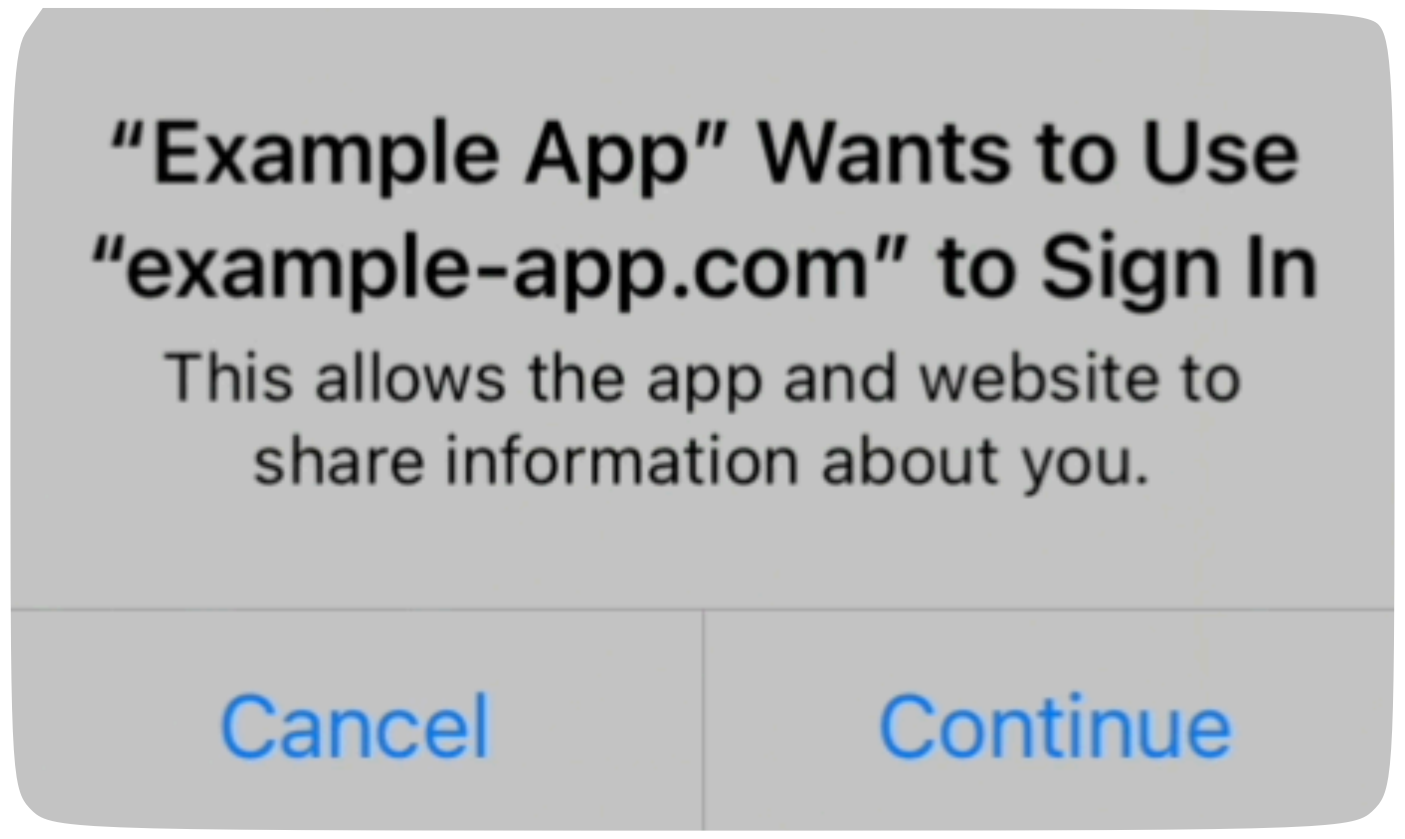
Domains
applinks:example-app.c
webcredentials:example
webcredentials:example

`ASWebAuthenticationSession`

- iOS < 17.4 only allows passing custom URL scheme to `ASWebAuthenticationSession`
- Any app can put in any scheme, it doesn’t actually launch the app, it just waits for that scheme to be returned in an HTTP Location header then dismisses the view and runs the callback
- In order to use a Universal Link as the redirect URI in < 17.4, you have to hack your way around the API

The Hack

- Find your target application's Client ID (easy)
- Find your target application's custom URL scheme (easy)
- Launch the system browser with a legitimate looking URL under the attacker's control, passing in the target application's custom URL scheme



“Example App” Wants to Use “example-app.com” to Sign In

This allows the app and website to share information about you.

The Hack

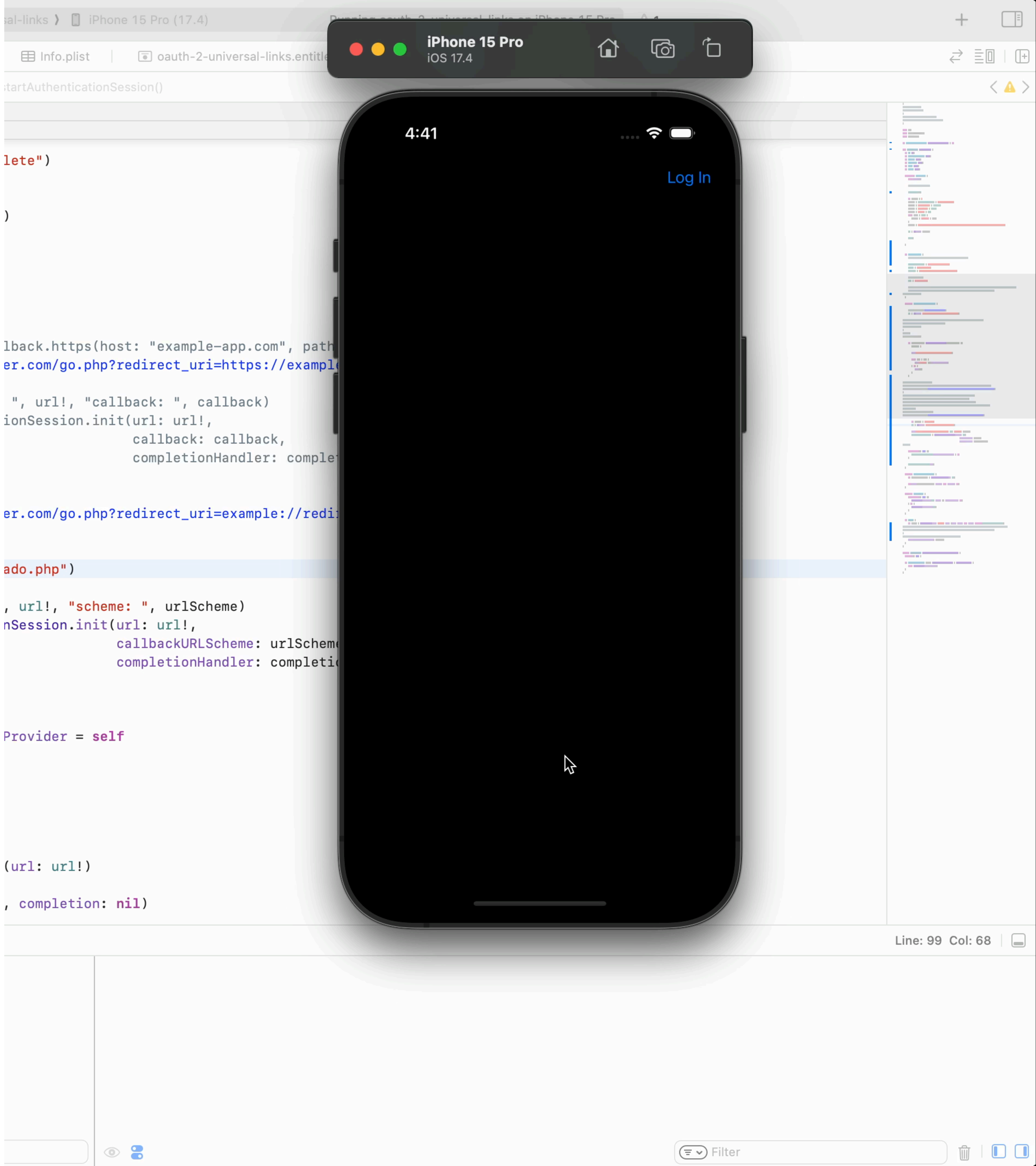
- Redirect from your server to the target application's AS
  - `example-app.com -> authorization-server.com`
- The AS will redirect to the custom URL scheme, which will trigger the `ASWebAuthenticationSession` callback
  - `authorization-server.com -> example-app://redirect`
  - If the user already has a session, they might not even see anything!

The Hack

“Example App” starts ASWebAuthenticationSession using “lol.avocado://” custom URL scheme that belongs to another app.

User already has a session, no interaction needed, authorization code is delivered to the callback.

PKCE and DPoP didn't help, because the attacker uses their own secrets to initiate the flow.


al-links › iPhone 15 Pro (17.4)
Info.plist
oauth-2-universal-links.entitle
startAuthenticationSession()
iPhone 15 Pro
iOS 17.4
4:41
Log In
Line: 99 Col: 68
Filter

```
lete")

)

lback.https(host: "example-app.com", path
er.com/go.php?redirect_uri=https://exampl

 ", url!, "callback: ", callback)
ionSession.init(url: url!,
                callback: callback,
                completionHandler: comple

er.com/go.php?redirect_uri=example://redi

ado.php")

, url!, "scheme: ", urlScheme)
nSession.init(url: url!,
              callbackURLScheme: urlScheme
              completionHandler: completi

Provider = self

(url: url!)

, completion: nil)
```

Mitigations

- Use https redirect URIs, and work around the iOS <17.4 limitation
- Don't support custom URL scheme in your app or AS at all,
  even for old iOS versions
- Always require user interaction at the AS web page,
  even with an existing session