

Authenticating with OpenID Connect



Brian Noyes

CTO, SOLLIANCE INC

@briannoyes www.briannoyes.com



Module Overview



Why OpenID Connect?

OpenID Connect Protocol Tokens and Flows

Adding OpenID Connect Authentication to Your App



Terminology

OpenID Connect Identity Provider ==



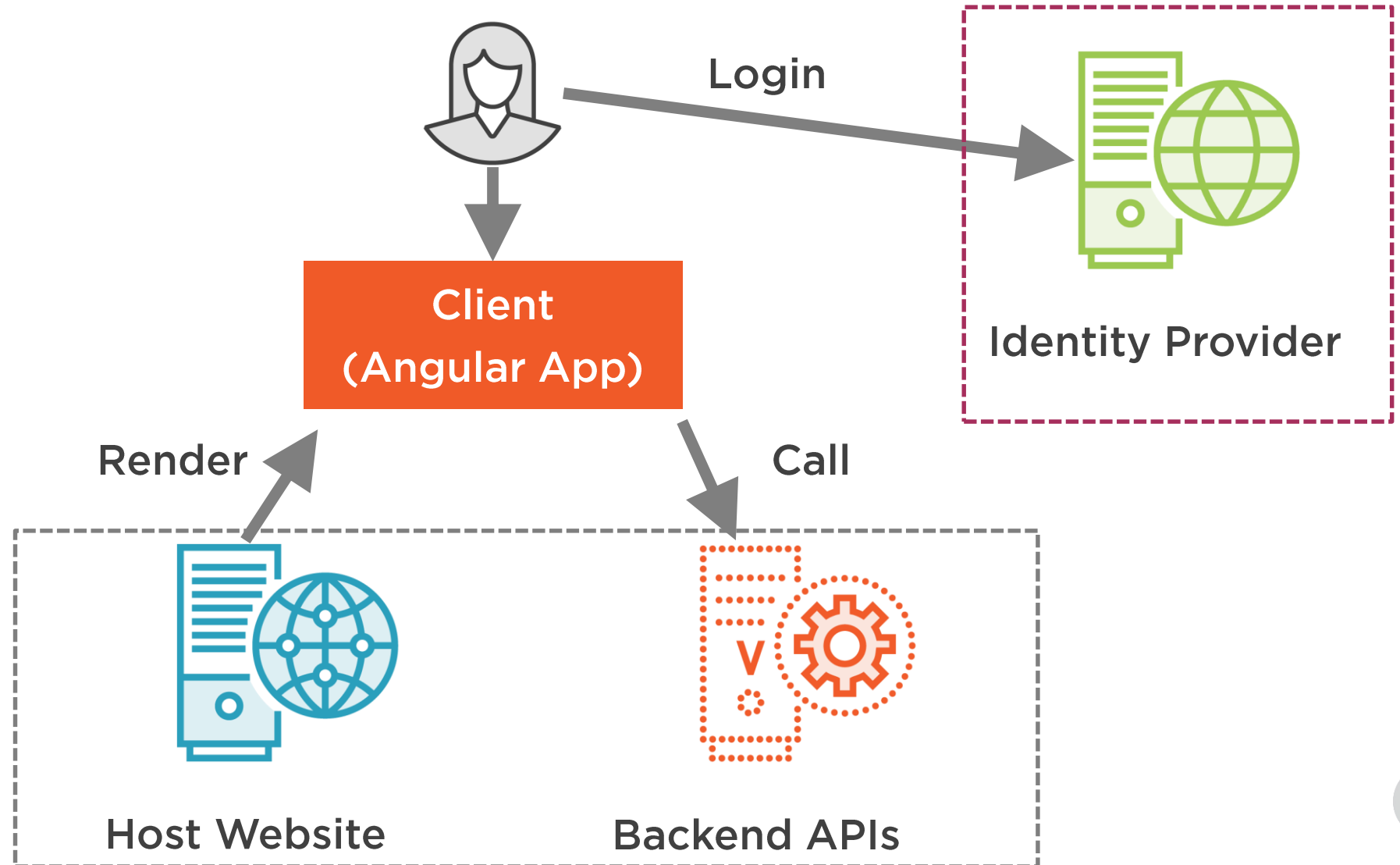
IdP



STS



Why OpenID Connect?



Why OpenID Connect?

Decoupling

Single Sign-On

**Centralized
Security
Management**



OpenID Connect JWT Token Contents



User



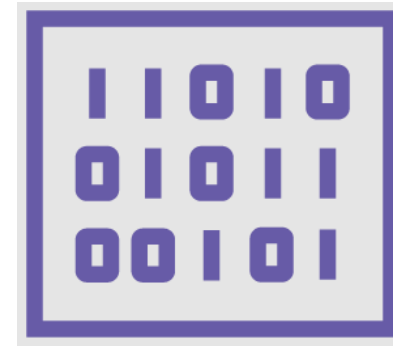
Client App



IdP

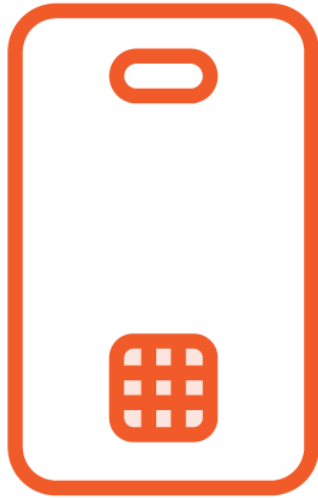


Resource



Protocol

JWT Token Types



ID Token



Access Token

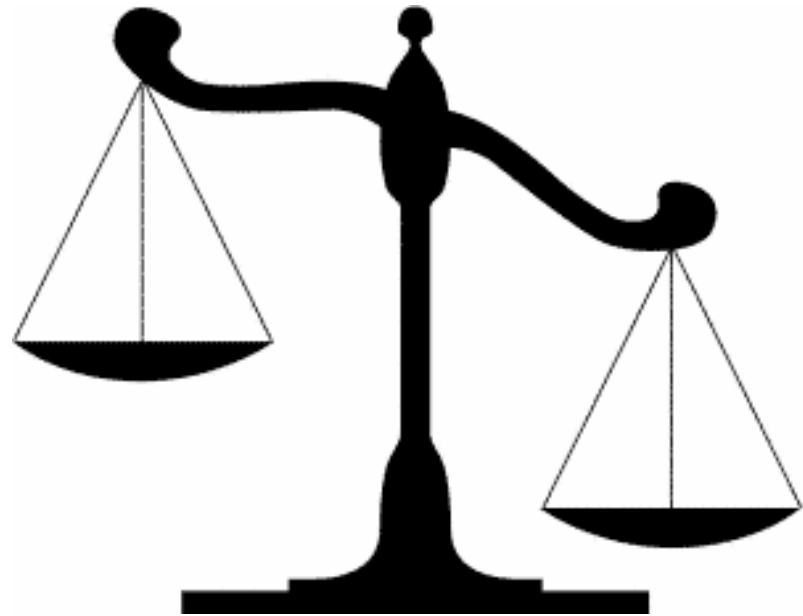
OpenID Connect Flows

**Authorization
Code**

Hybrid

Implicit





Using the oidc-client Library

- **Is oidc-client “heavy”?**
 - ~100KB
- **Compared to your app bundles, no**
 - Hello World: ~150KB
 - Real App: 1 - 20MB
- **Only downloaded once per published version of your app**
- **Bottom line:**
 - Security is hard
 - You better get it right

Demo



Add oidc-client and auth service

Add login redirect

Add post-login callback page

Work with User object and logged-in status

Inspect the JWTs



Demo



Choosing how to integrate with a provider

Setting up tenants, apps, and users in Auth0

Change client configuration for Auth0

Address configuration differences for Auth0



Provider Specific Considerations

OpenID Connect
is a standard, but
there are
configuration
differences

Use provider-
specific library, or
spec compliant
one like oidc-
client

Will stick to oidc-
client to show
commonalities
and differences



Summary



Deeper dive into OpenID Connect protocol

Implemented OpenID Connect login & logout with IdentityServer4

Switched to doing same with Auth0

