Authenticating with OpenID Connect

Brian Noyes
CTO, SOLLIANCE INC

@briannoyes www.briannoyes.com

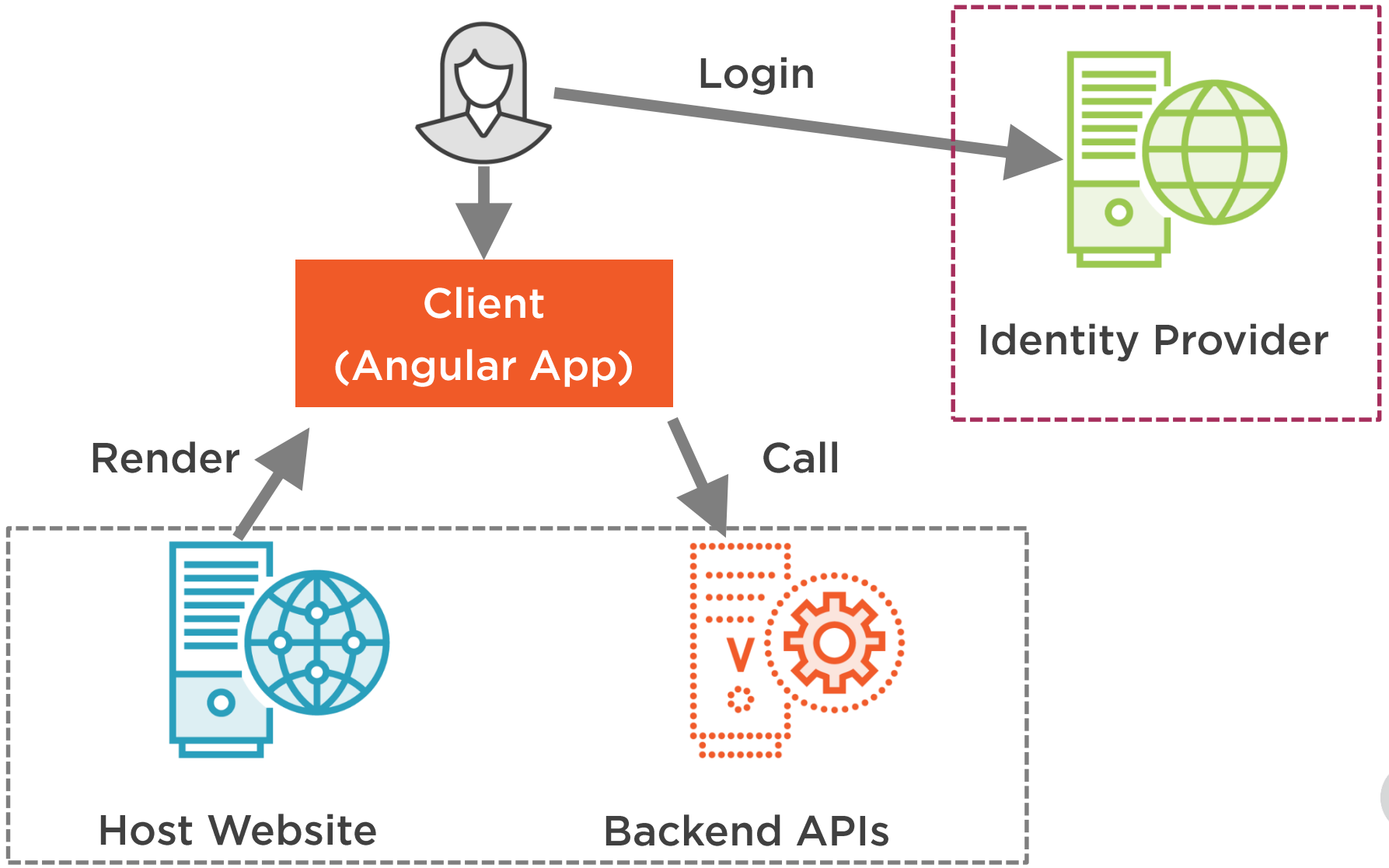
Module Overview

- Why OpenID Connect?
- JSON Web Tokens (JWTs)
- OpenID Connect protocol flows
- A word about oidc-client size
- Getting the demo code running
- Add oidc-client and Angular service for authentication
- Configure client for connecting to STS
- Add UI/logic to start the login flow
- Handle callback to complete login flow
- Initiate and complete logout flow
- Debugging/Troubleshooting techniques

Terminology

OpenID Connect Identity Provider ==

Why OpenID Connect?



Why OpenID Connect?

Decoupling

Single Sign-on

Centralized Security Management

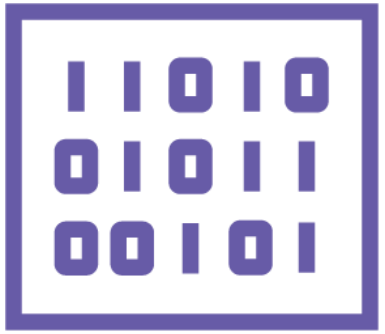
OpenID Connect JWT Token Contents

User

Client App

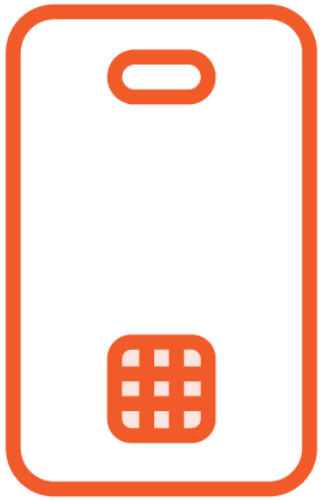
IdP

Resource



Protocol

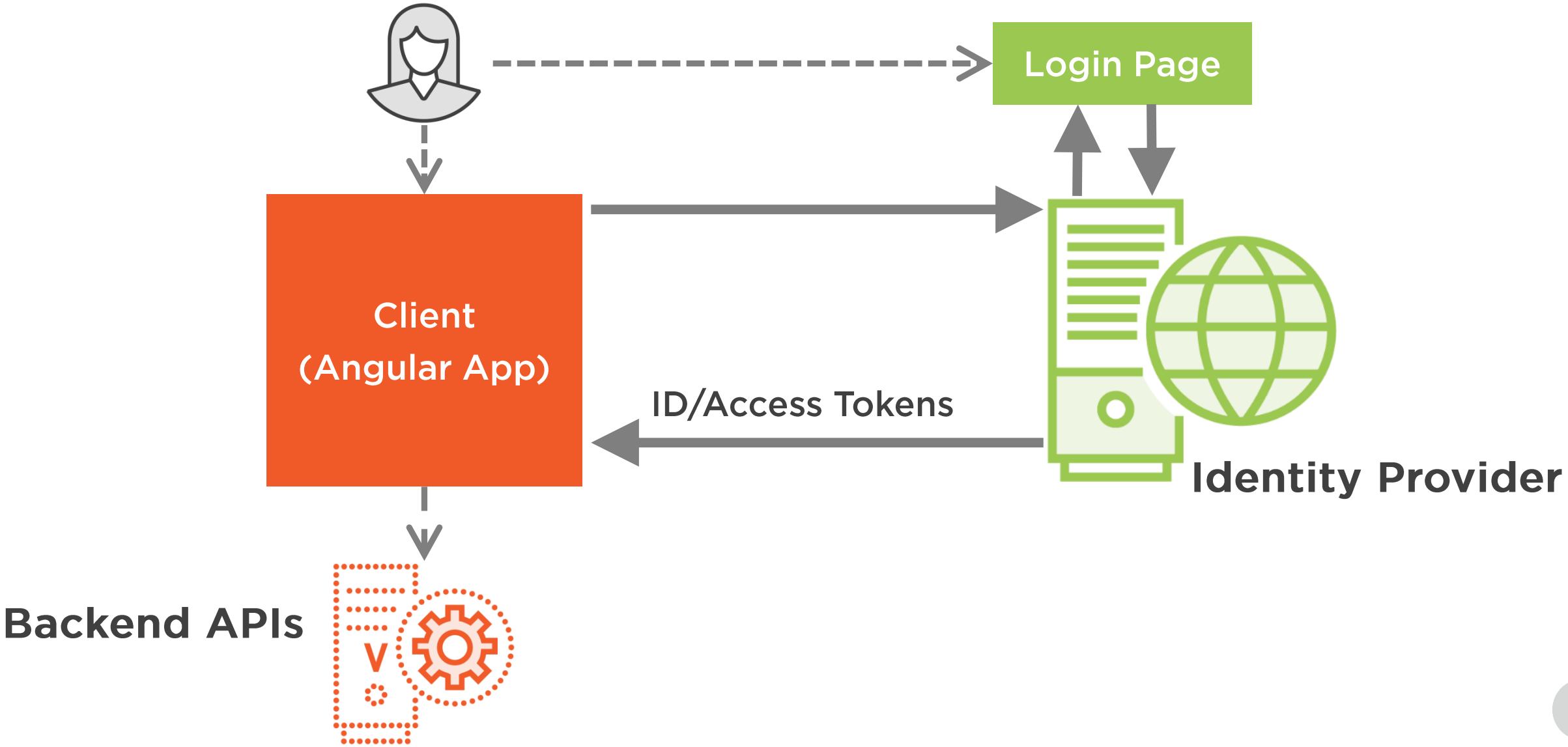
JWT Token Types



ID Token

Access Token

OpenID Connect Flows



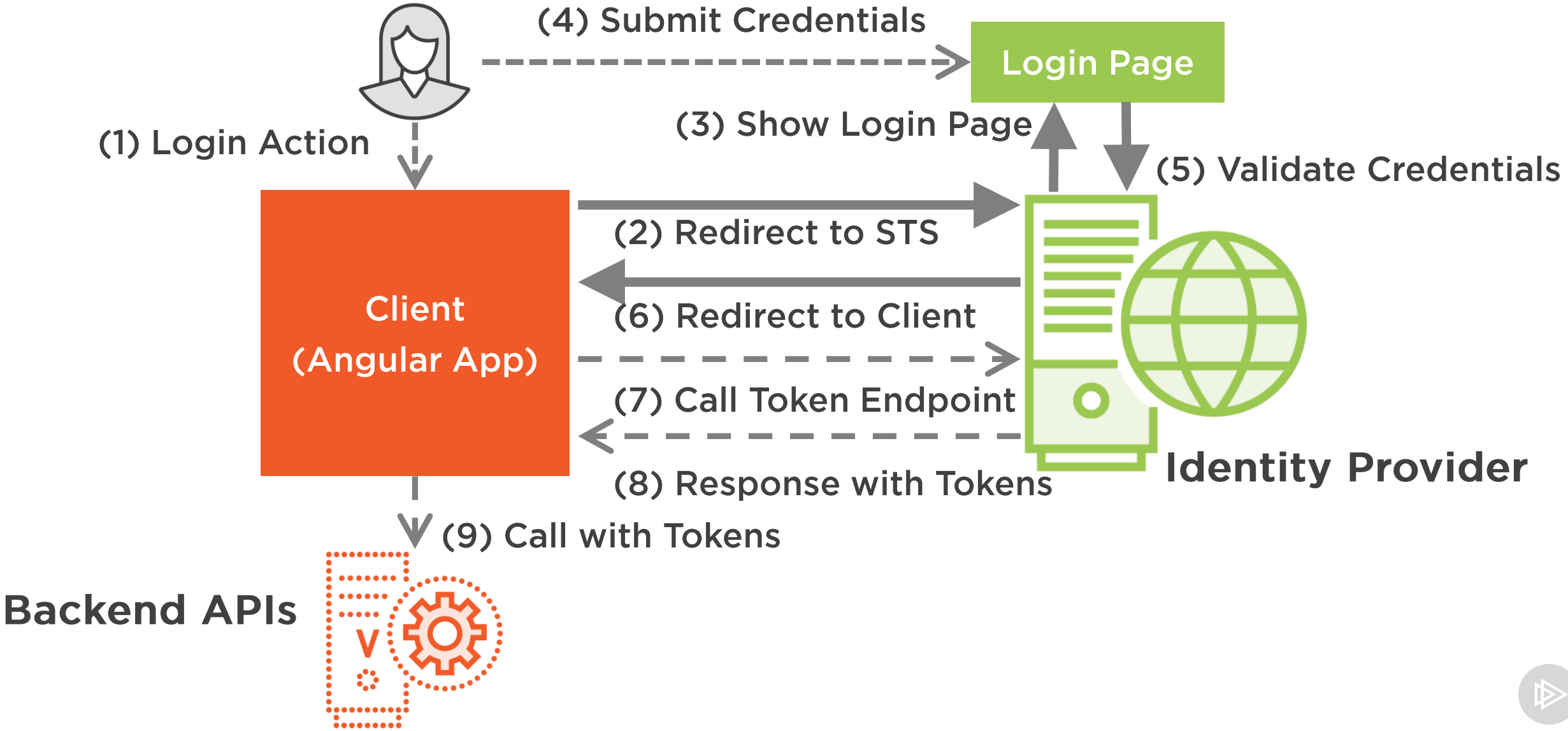
OpenID Connect Flows

Authorization Code

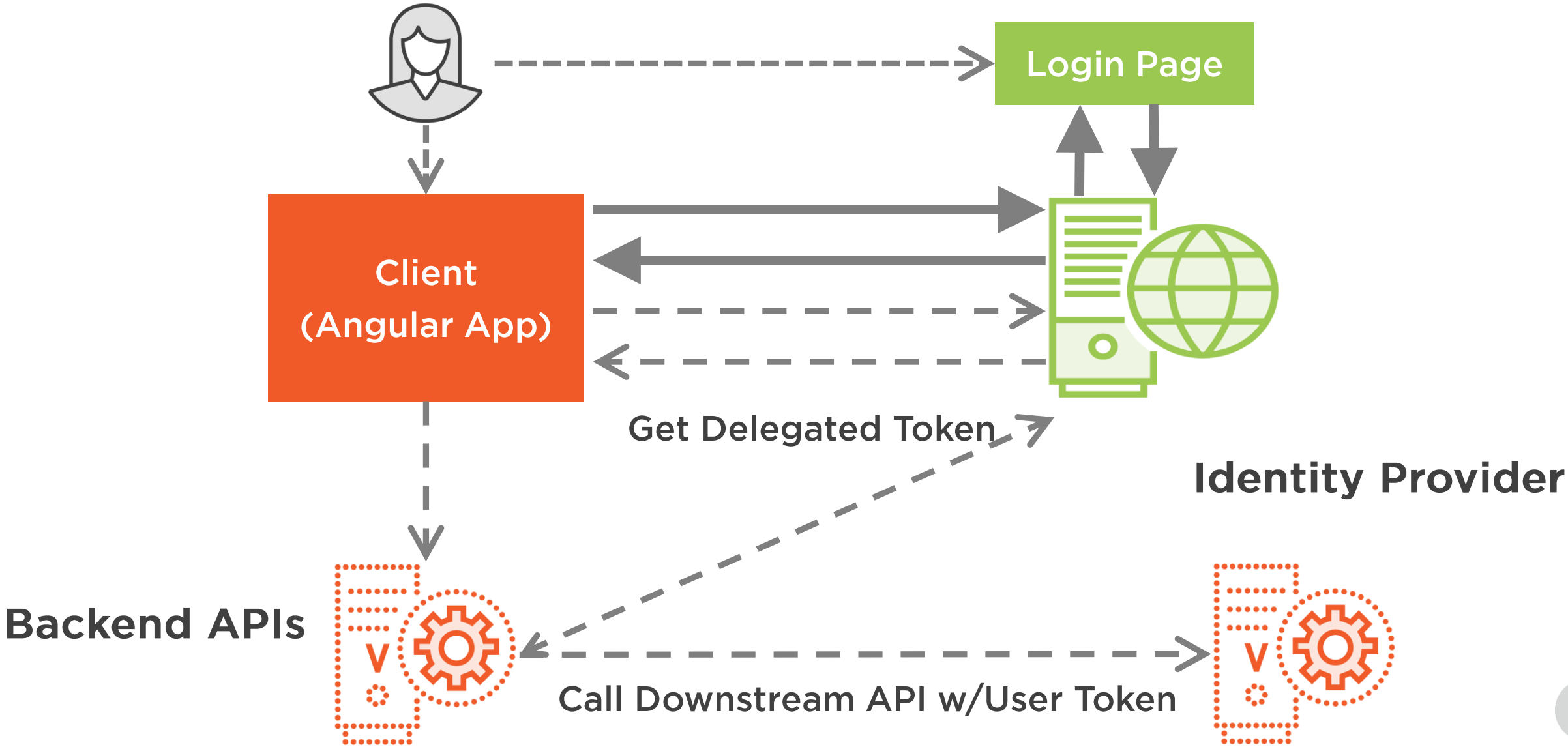
Hybrid

Implicit

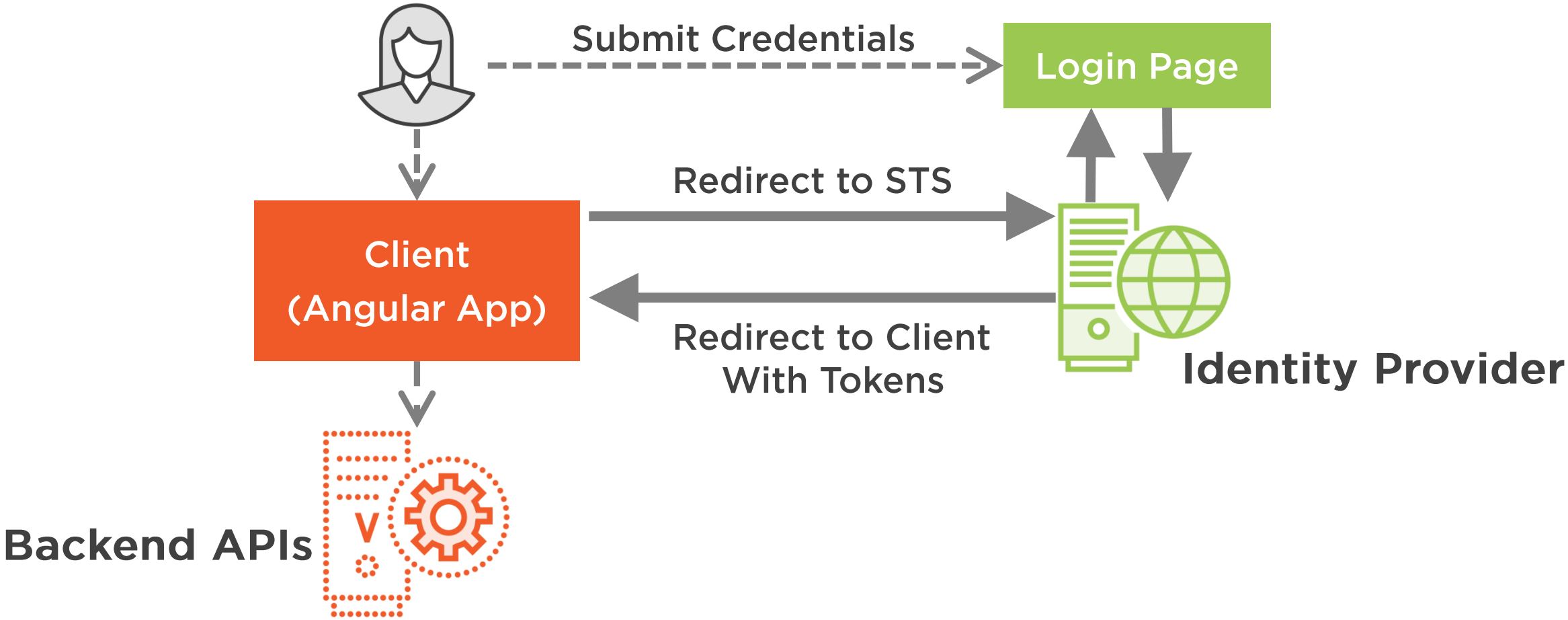
Authorization Code Flow



Hybrid Flow


Login Page
Client
(Angular App)
Get Delegated Token
Identity Provider
Backend APIs
Call Downstream API w/User Token

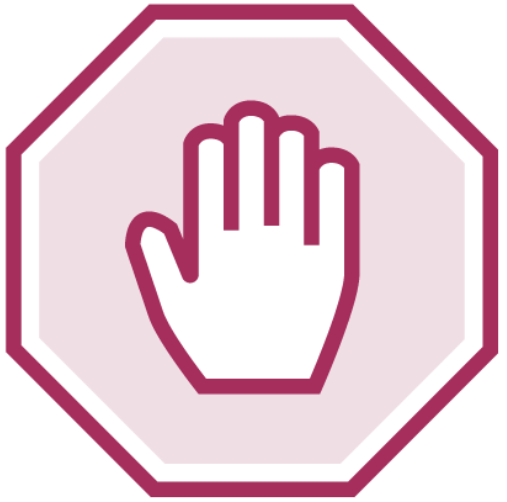
Implicit Flow



Implicit Flow

< 2019

Recommended for SPAs



Authorization Code

Hybrid

New OpenID Connect/OAuth 2 Guidance

Source: Internet Engineering Task Force (IETF)

OAuth 2.0 Security Best Current Practice

(https://noyes.me/oauth2-sbcp)

OAuth 2.0 For Browser-Based Apps

(https://noyes.me/obba)

Driven by widespread browser support for:

- CORS
- Same-site Cookies

Proof Key for Code Exchange (PKCE)

aka “Pixie”

Makes Authorization Code flow safe for “public clients”

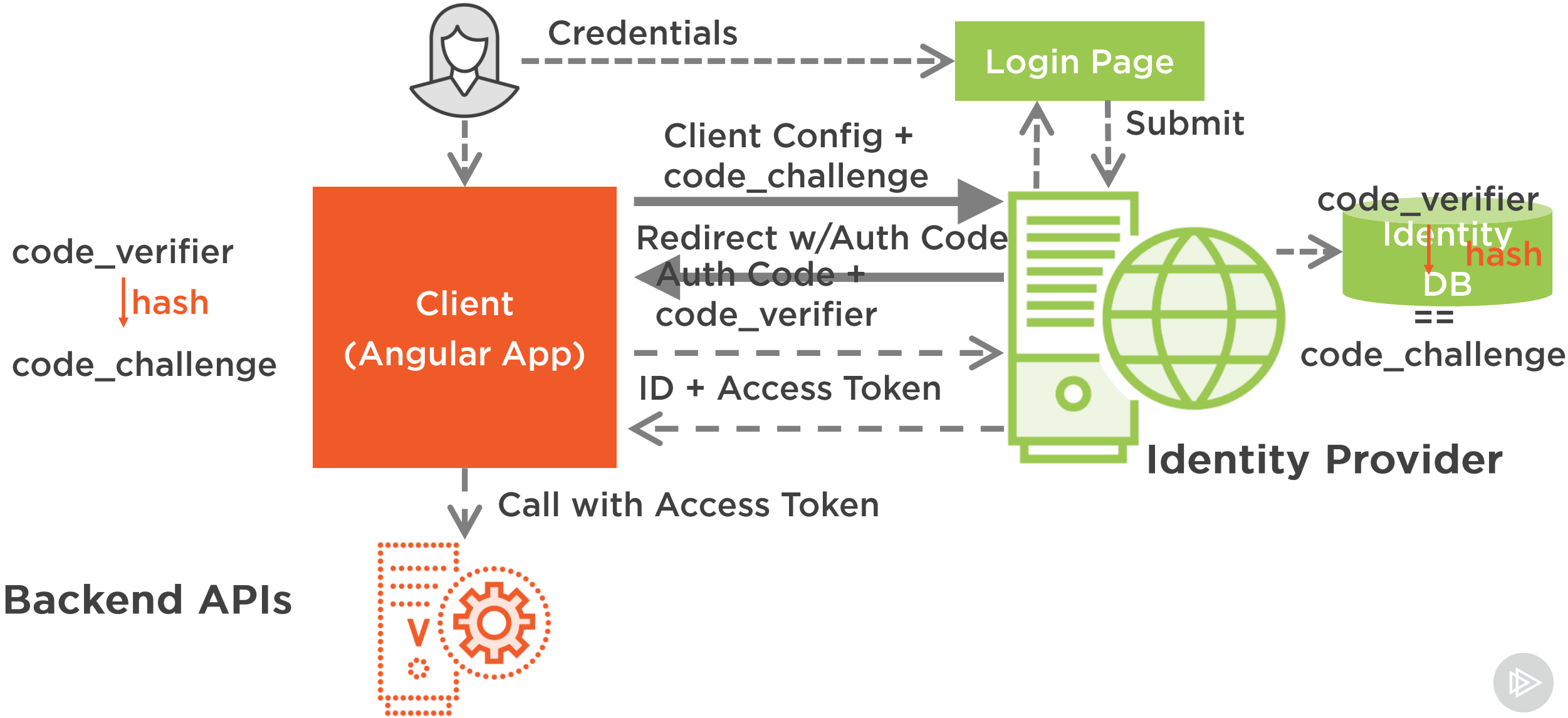
Generate and hash additional code/key, private to client, and STS

Improvement on Implicit Flow hash fragment tokens vulnerability

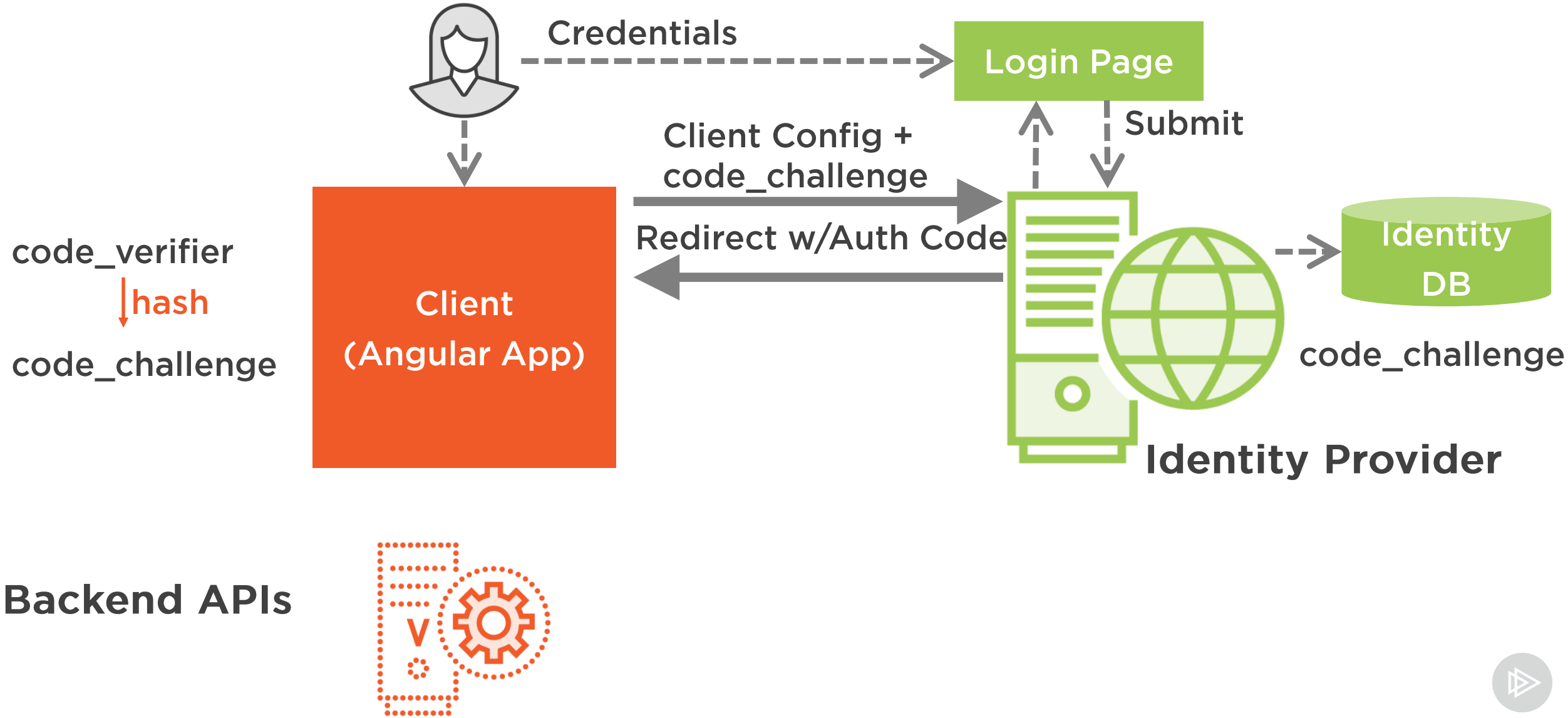
Authorization Code + PKCE deprecates Implicit Flow

Update Existing Implicit Flow Apps?

Authorization Code Flow with PKCE



Authorization Code Flow with PKCE



Authorization Code Flow with PKCE

Login Page

Client
(Angular App)

Auth Code +
code_verifier

ID + Access Token

code_verifier
hash
==
code_challenge

Identity Provider

Call with Access Token

Backend APIs

A Word About oidc-client

Is oidc-client “heavy”?

- ~370KB uncompressed

Compared to your app binaries, no

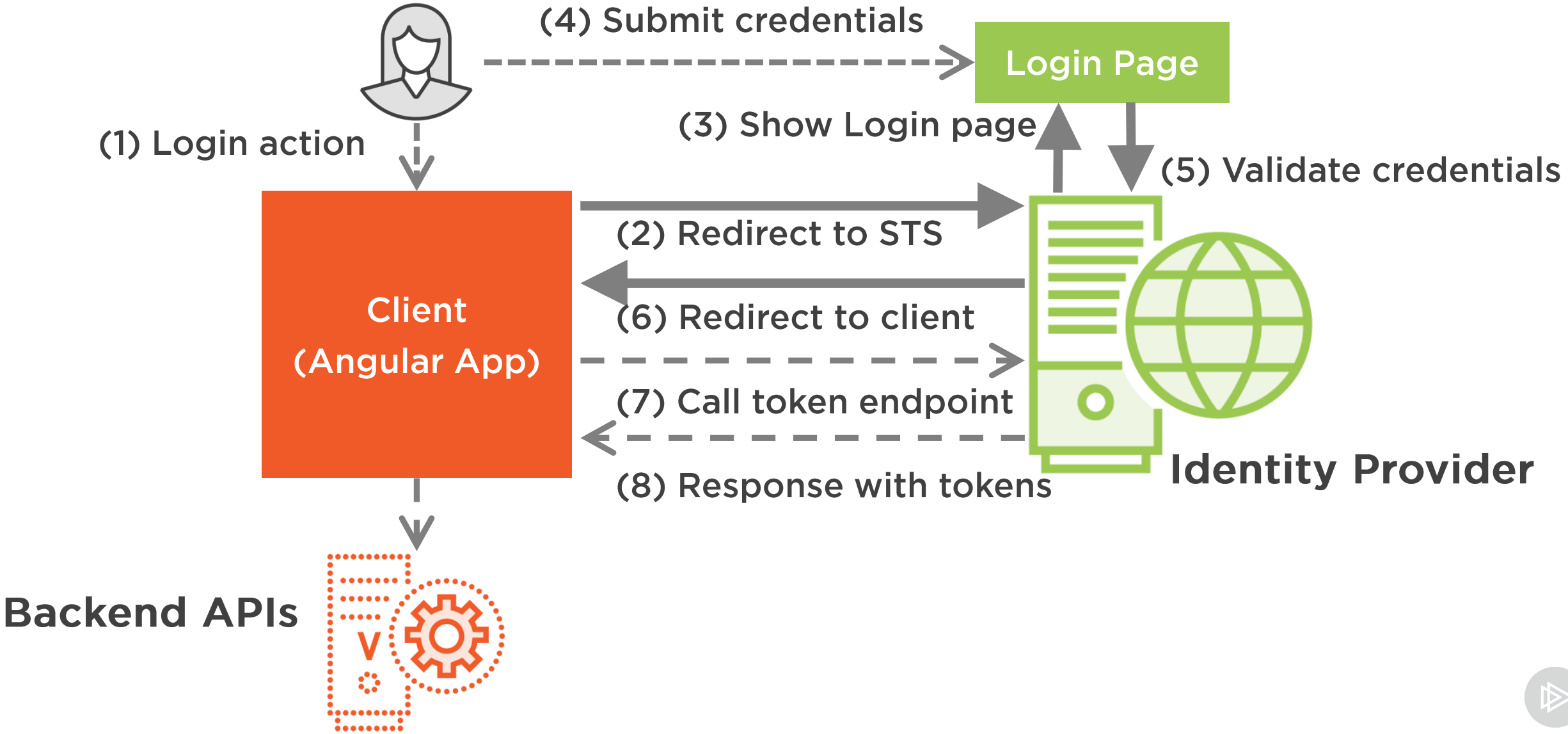
- Hello World: ~210KB
- Real App: 1 - 20MB

Only downloaded once per published version of your app

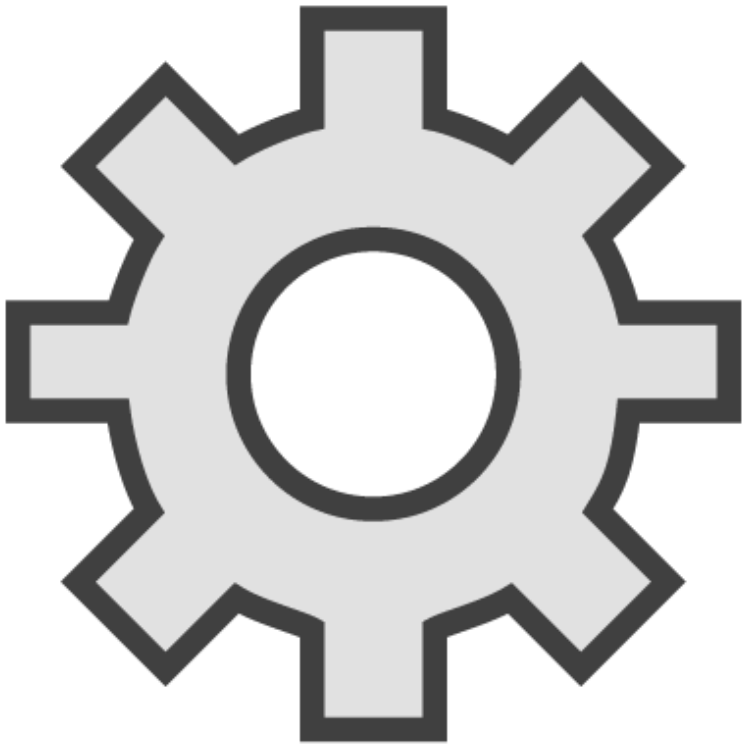
Bottom line:

- Security is hard
- You better get it right

Authorization Code Flow



Debugging Client Configuration Errors



Most common: client configuration

Client OIDC settings must match IdP's Client configuration exactly

Two common errors:

- Client ID
- Redirect URLs

A Word About User Registration

User registration not part of OpenID Connect

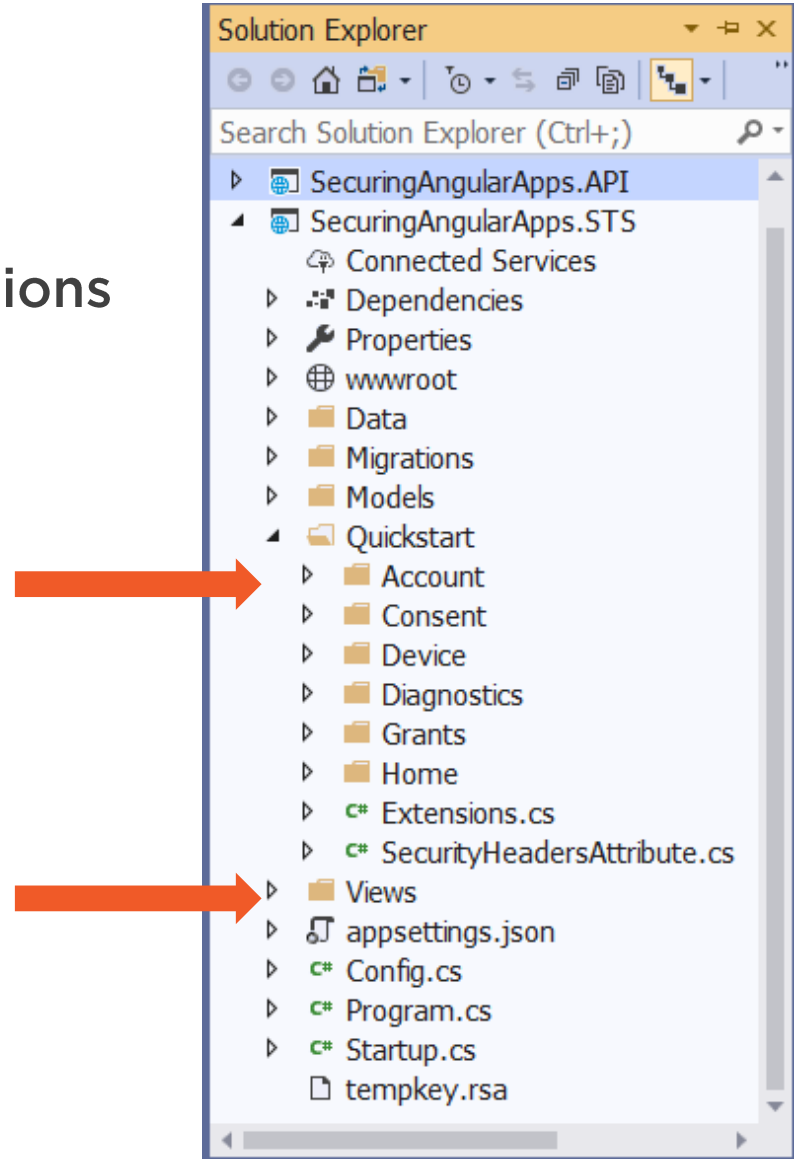
Business environments: No self-registration

Admin invites and provisions users

A Word About User Registration

IdentityServer4:

- **Add Pages and Controller Actions**



Summary

OpenID Connect protocol

- Why use it
- What are JWT tokens
- Protocol flows

Getting the sample code running

Adding OpenID Connect authentication

- Add oidc-client and authentication service
- Add login button and initiate login
- Handle callback and complete login flow
- Add code to modify UI based on authentication status
- Initiate and complete logout process

Debugging techniques