Securing Angular Apps with OpenID Connect and OAuth2

ANGULAR APP SECURITY BIG PICTURE

Brian Noyes
CTO, SOLLIANCE INC

@briannoyes www.briannoyes.com

Course Overview

Angular Security Big Picture

Authenticating with OpenID Connect

Authorizing Calls to Your Backend with OAuth2

Enhancing the User Experience with Client Security Context

Prerequisites

Angular Fundamentals

https://app.pluralsight.com/paths/skills/angular

ASP.NET Core Fundamentals

https://app.pluralsight.com/library/courses/aspdotnet-core-fundamentals/

Module Overview

Security Considerations for Angular Apps

Authentication and Authorization with OpenID Connect and OAuth2

Identity Provider Options

Client Library Options

Security Design Considerations

Authentication

Authorization

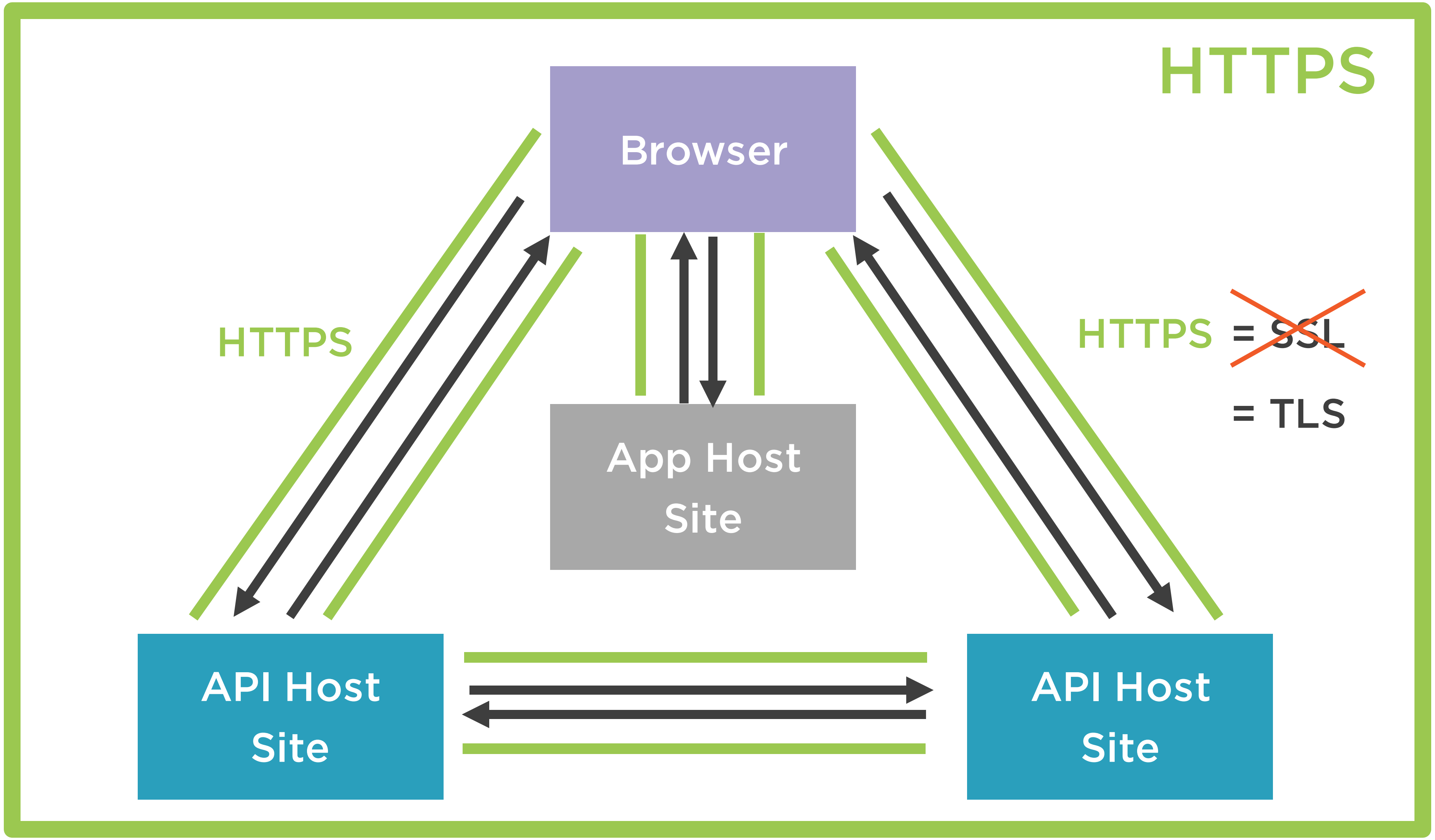
Transport Protection

Cross Origin Resource Sharing (CORS)

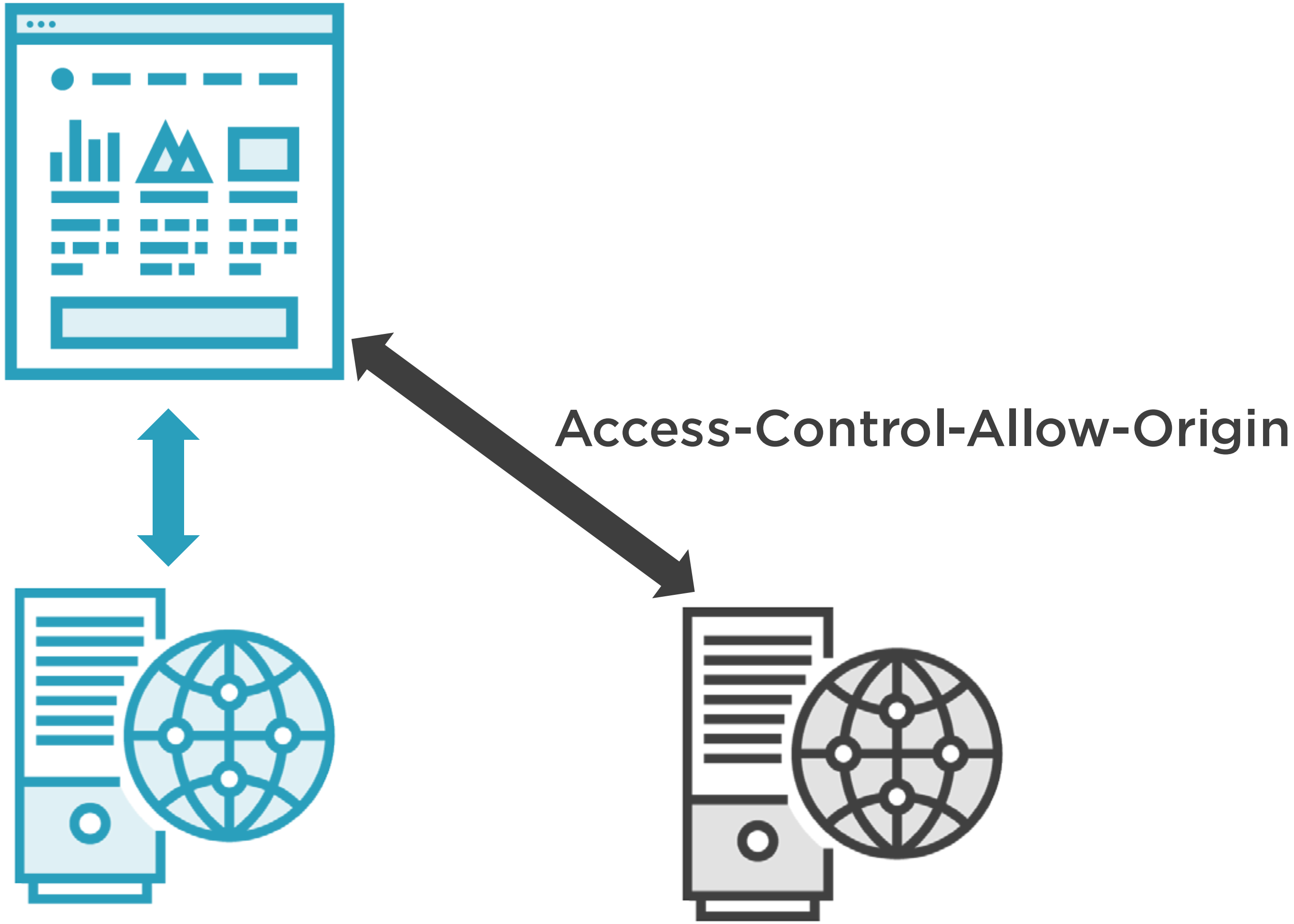
Cross Site Request Forgery (CSRF)

Cross Site Scripting (XSS)

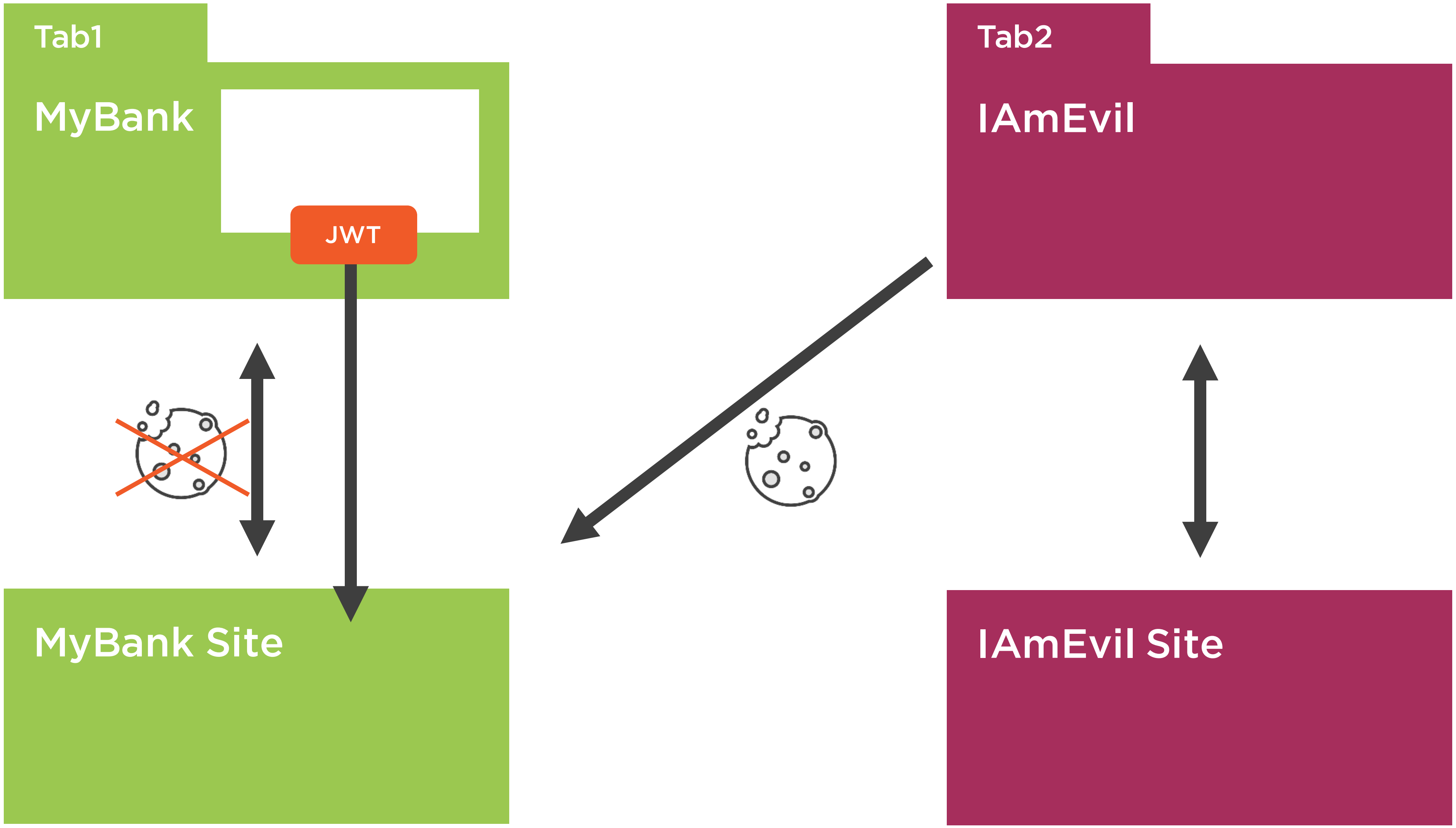
Transport Protection



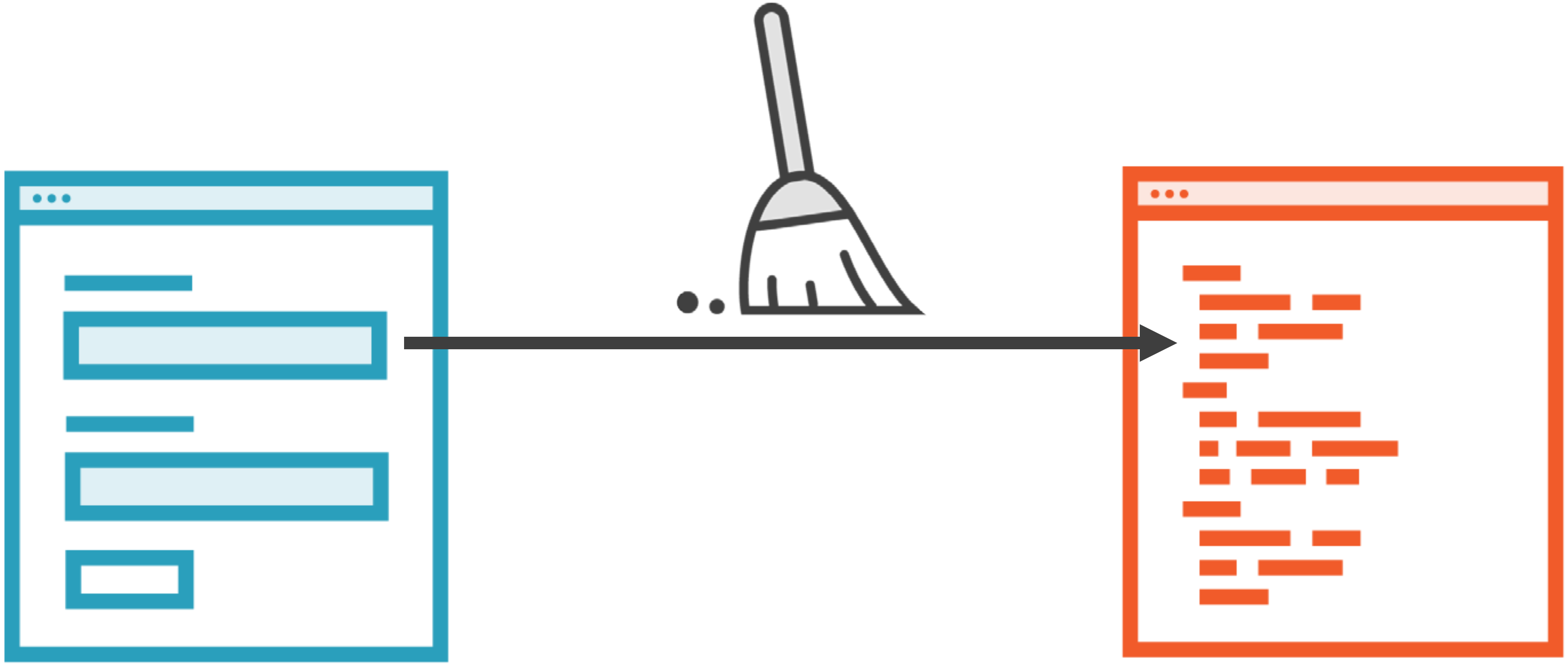
Cross Origin Resource Sharing



Cross Site Request Forgery (CSRF)



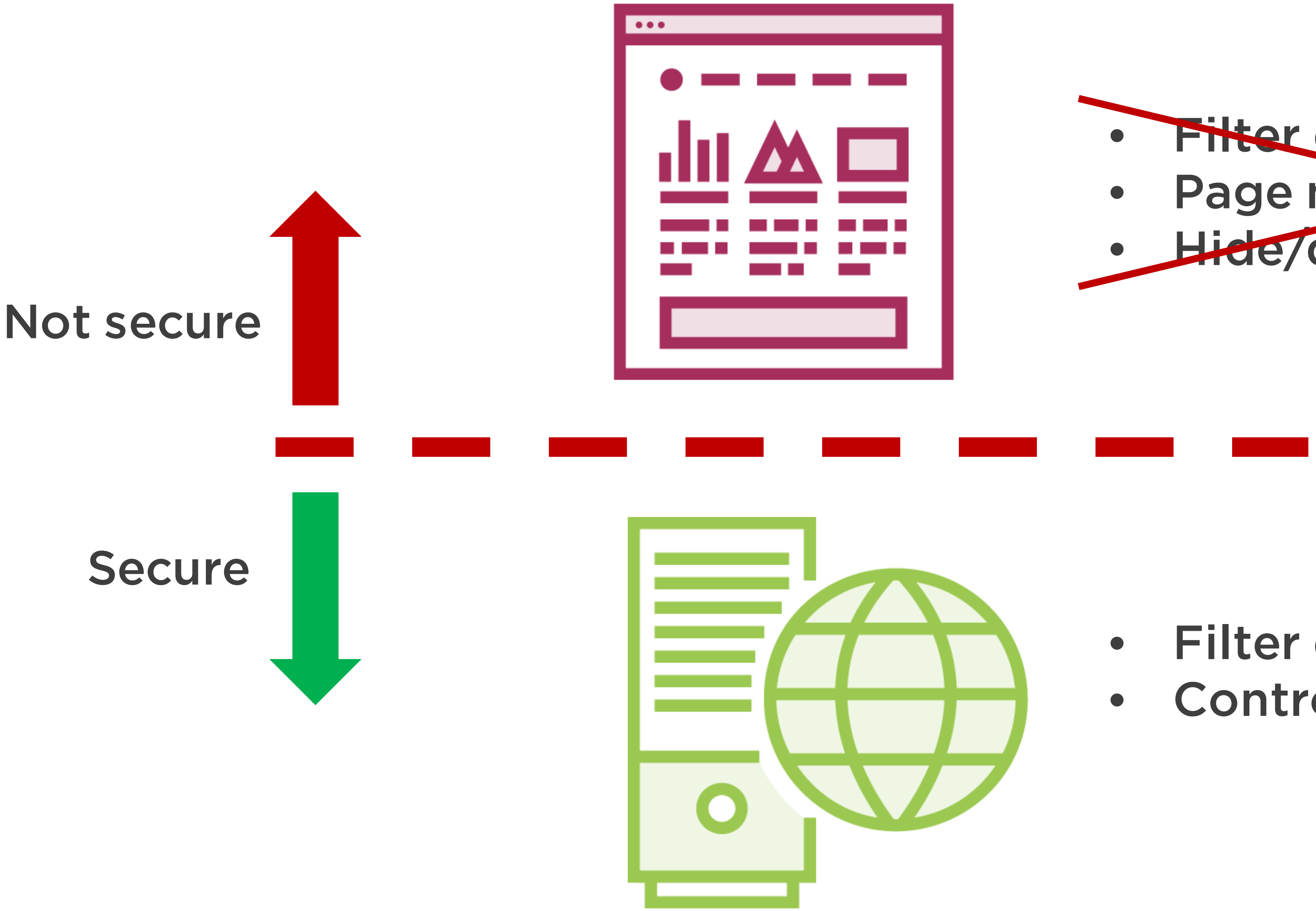
Cross Site Scripting (XSS)



For more coverage of these security considerations:
AngularJS Security Fundamentals – by Troy Hunt

Spoiler:
You can’t truly secure anything in your Angular app code

App Security



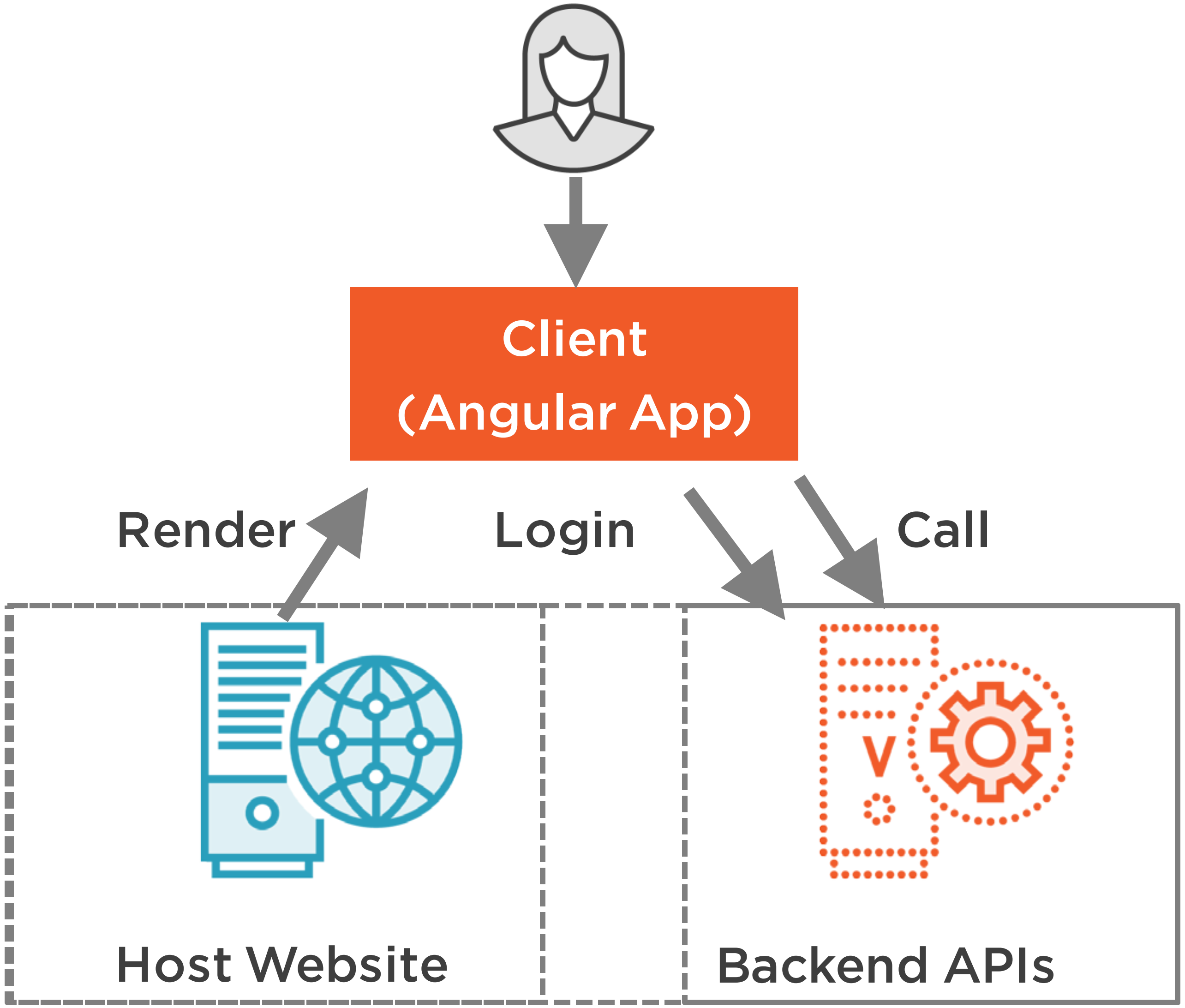
Not secure

Secure

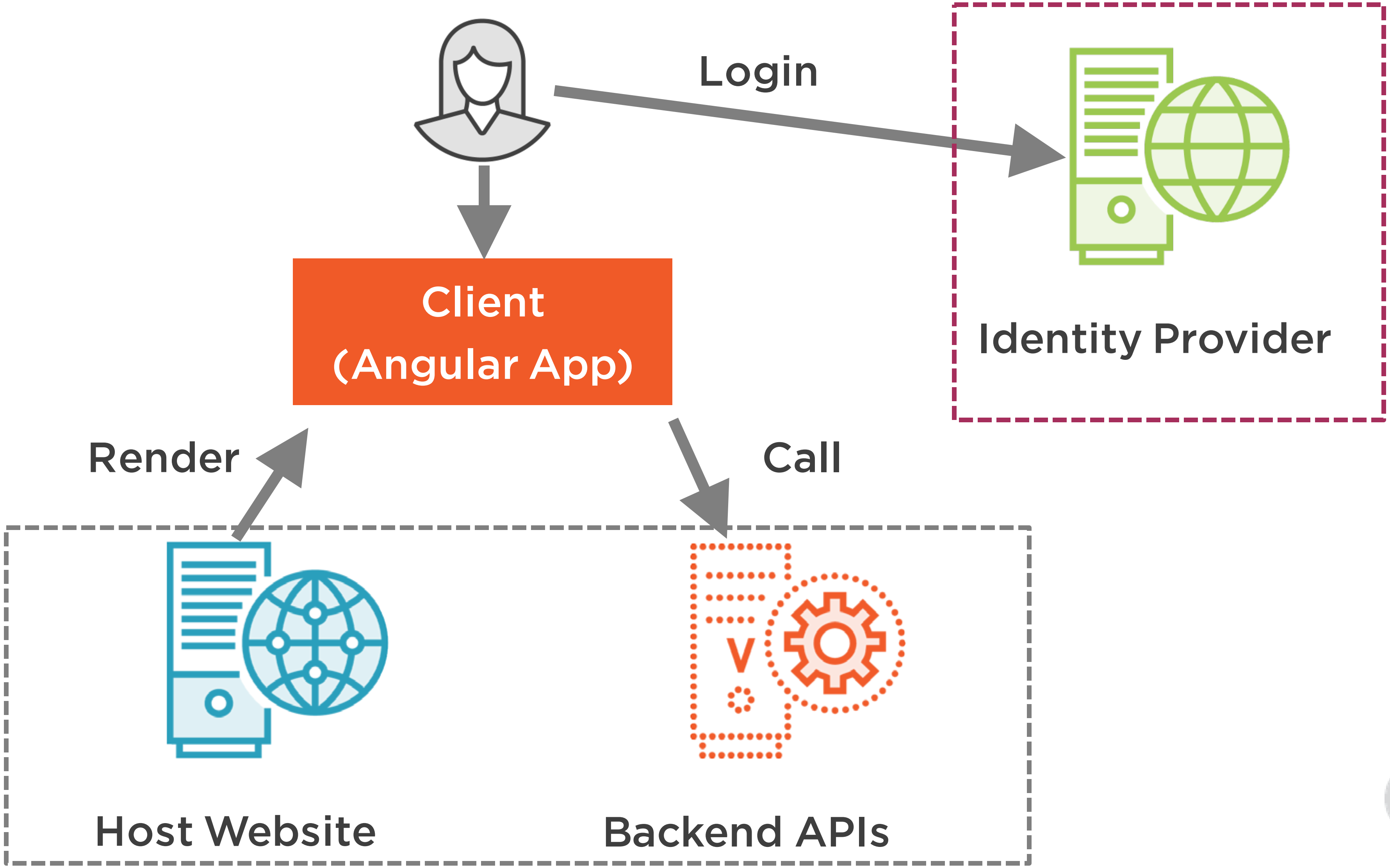
- ~~Filter data~~
- ~~Page navigation restrictions~~
- ~~Hide/disable actions~~

- Filter data
- Control access to APIs

Traditional Authentication Architecture



Angular + OpenID Connect + OAuth2 Architecture



Authentication

Determine who the user/client is and issue temporary ID

Request credentials

Collect credentials and validate

Issue temporary credential (token) for specific App / API (scope)

Authentication

Authorization

Deciding what to allow the user/client to do/see

Check and validate roles

Look up and validate permissions

Block / grant access to actions

Terminology

Identity Provider
Authentication Server
Authorization Server
SSO Server
STS

User Agent

Client

Resource

Scope

JWT

OAuth

OAuth 1.0

- Began in 2006
- Focused on Twitter API access
- Approved standard 2010

OAuth 2.0

- Focused on web, mobile, desktop apps and APIs
- Approved standard 2012

Lacked any specification of how authentication happens

OpenID Connect

Derivative from OAuth 2

Same token format – JWT

Approved standard 2014

Standardizes flows for collecting credentials from user/client and issuing tokens

Identity Providers

Google

Facebook

Twitter

Azure Active Directory (AAD)

Azure Active Directory v1

No OpenID Connect

Microsoft organizational accounts only

Azure Active Directory v2

OpenID Connect

Microsoft organizational & personal accounts

AAD Business to Consumer (B2C)

OpenID Connect

All Microsoft accounts & custom accounts

Identity-as-a-Service Providers

Auth0

okta

Ping Identity®

IdentityServer4

identity SERVER

Open source identity provider framework

Requires some coding and configuration

Have to host yourself

Most flexible option for Single Sign-On (SSO) federation scenarios

Certified protocol compliant

- https://openid.net/certification/

Client Libraries

angular-jwt

ADAL

MSAL

oidc-client