# Securing Web APIs and Web Apps with IdentityServer

Brian Noyes

CTO & Co-founder, Solliance

brian.noyes@gmail.com, @briannoyes


Demos/Slides - http://noyes.me/devint2017-ids

# About Brian Noyes

CTO and Co-founder, Solliance
www.solliance.net


Microsoft Regional Director

Microsoft
Regional Director

Microsoft MVP


Pluralsight author
www.pluralsight.com


Web API Insider, Windows Azure Insider,
Window Store App Insider, C#/VB Insider

brian.noyes@solliance.net

@briannoyes

http://briannoyes.net

DEV intersection

# Agenda

- Authentication Overview
- IdentityServer Overview
- Setting Up IdentityServer
- Securing an API
- Securing a Web App
- Using ASP.NET Identity

# What does it mean to “secure”?

- More than just “logging in”
- Authentication
- Authorization
- Transport protection
- Cross Origin Resource Sharing (CORS)
- Cross Site Request Forgery (CSRF/XSRF)
- Cross Site Scripting (XSS)
- User and access control management

# Authentication Options

- Windows authentication
- Basic authentication
- Cookie-based authentication with host site
- Token-based authentication (STS)

# Protocols

- **OAuth2**
  - Just about authorization
  - Issued access token after user is authenticated “somehow”
- **OpenID Connect**
  - Builds on OAuth2
  - Just about authentication
  - Issued id token after presenting valid credentials

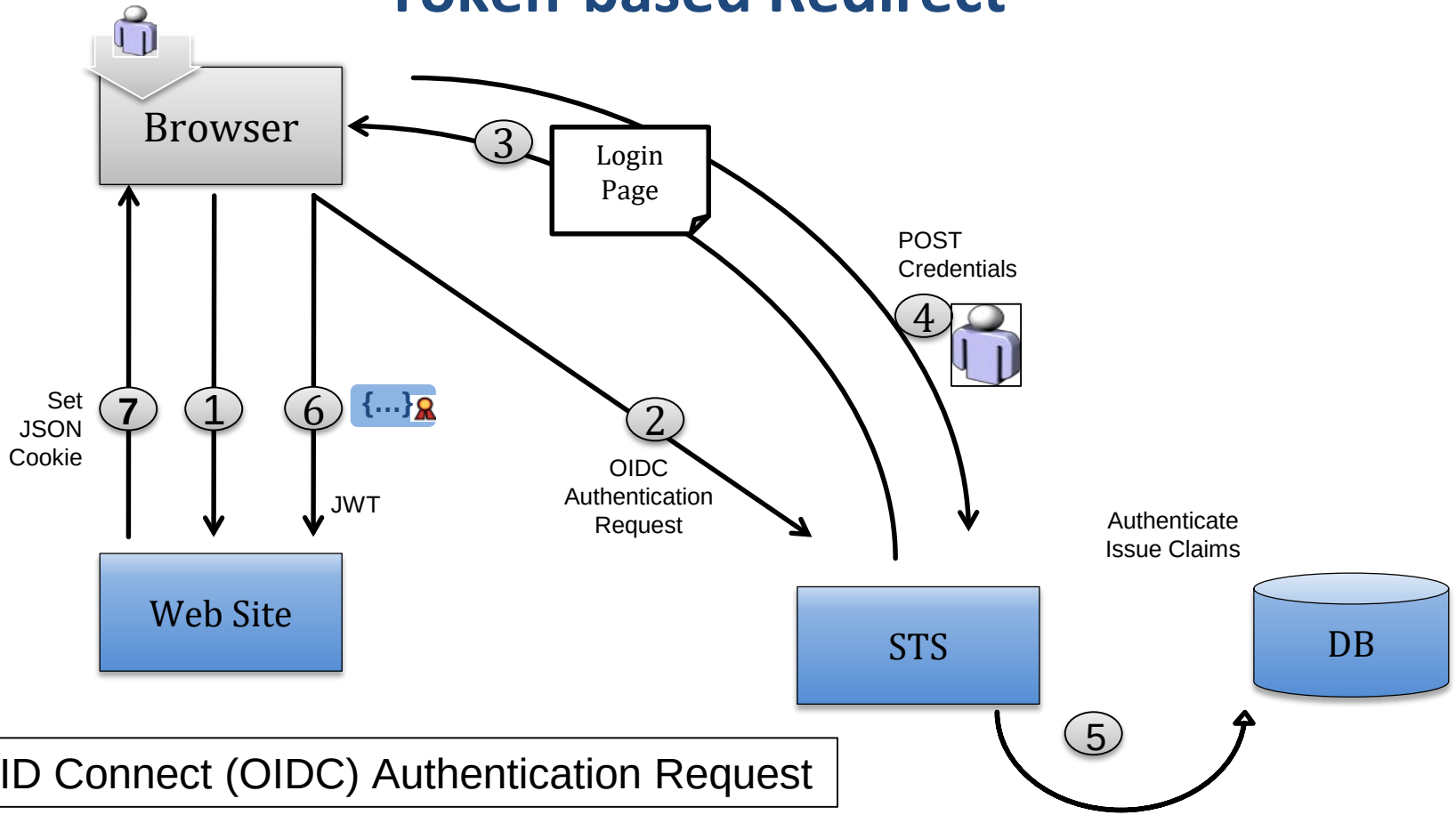
# Terminology

- **Client – application requesting access to a Resource**
- **Resource / Relying Party – a secured API/app that Client wants to call**
- **Resource Owner – end user using the Client**
- **Scope – a named resource that authorization is needed for**
- **Identity Provider (IdP) / STS / SSO server / Authentication Server / Authorization Server**
  - App that manages identities, authenticates users, returns ID and Access tokens for use by Client
  - IdentityServer, Azure AD, ADFS, Domain Controller, Auth0 server
- **JWT – "jawt" – token format used for OpenID Connect and OAuth2**

# IdentityServer Overview

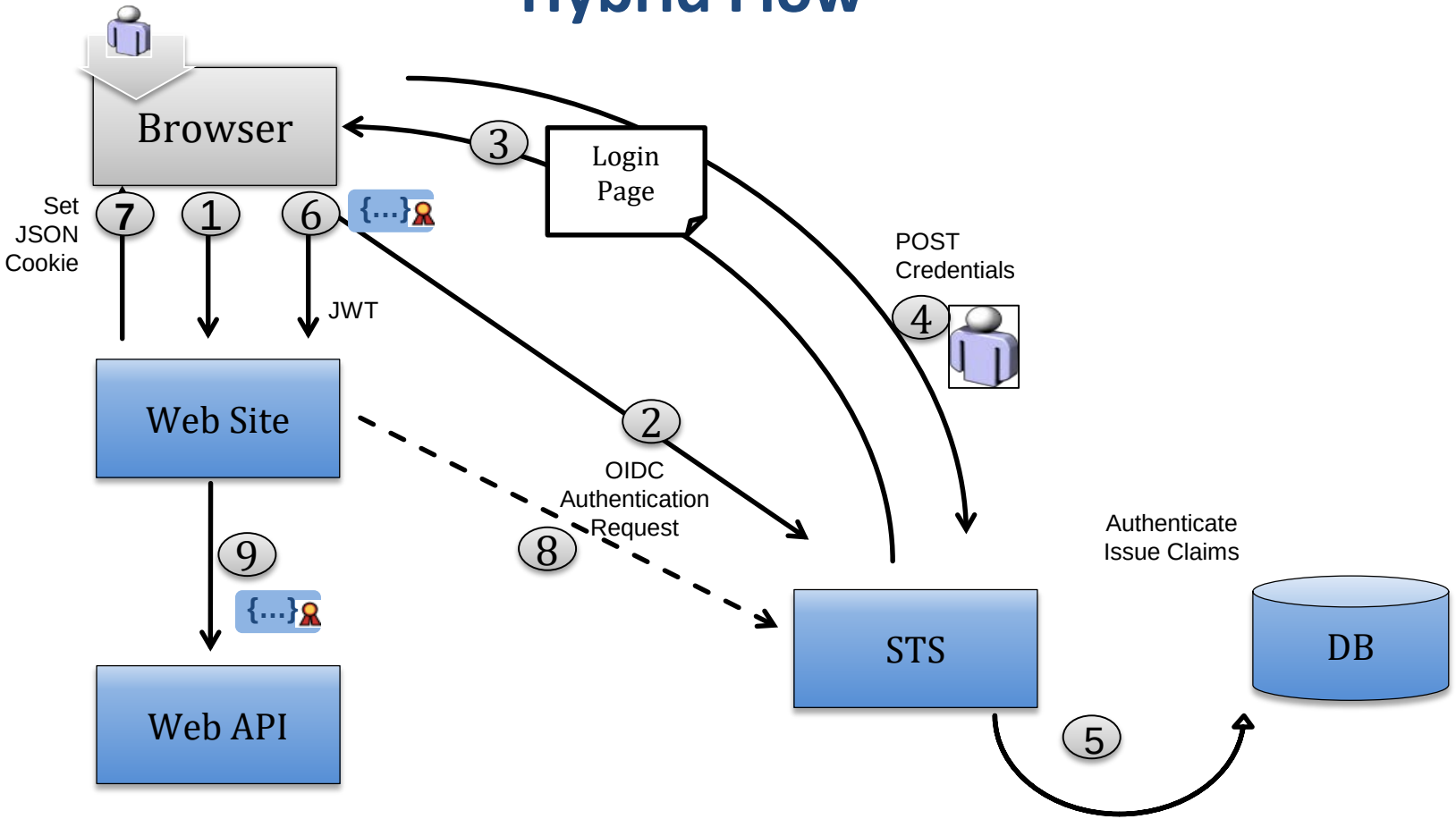
- **STS / SSO server / Authentication server**
- **Two current versions**
  - V3 – ASP.NET 4.x and up
  - V4 – ASP.NET Core
- **Open source framework**
- **Extensibility points for defining Clients, Scopes, Users, custom UI**
- **Very widely used, well documented, recommended by Microsoft**

# Token-based Redirect

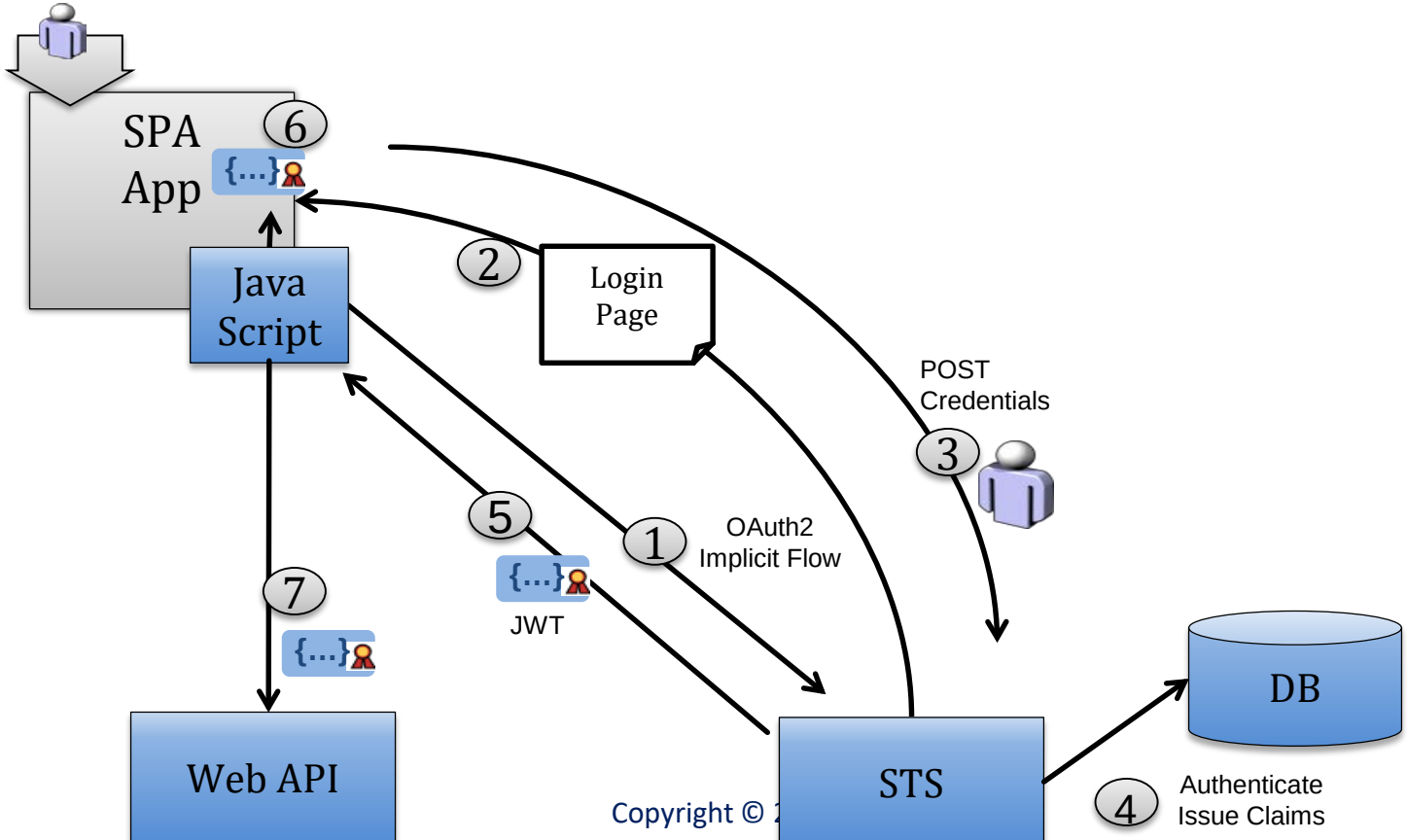



OpenID Connect (OIDC) Authentication Request

# Hybrid Flow



# OAuth2 Implicit Flow



# ASP.NET (Core) Identity

- Handles creation and management of identities
- Handles password hashing / creation, crypto protocols, etc.
- Layers in nicely “below” IdentityServer

# Dealing with CORS

- **Cross Origin Resource Sharing**
- **Web APIs on a different host than pages rendered from**
- **Built in to all modern browsers**
- **Simple CORS**
  - GET/POST, form encoded, no additional header
  - Sends Origin header in request, expects Access-Control-Allow-Origin in response
- **Most CORS**
  - Sends “preflight” OPTIONS request specifying what is being requested (Verb, headers, cookies,etc)
- **Destination server decides who gets in**
- **Have to populate appropriate headers in your $http service calls**
- **Automatic with Angular $http service with right configuration**

# Summary

- **IdentityServer is a great way to secure your web applications and APIs**
- **Docs:**
  - https://identityserver4.readthedocs.io/en/release
- **Code:**
  - https://github.com/IdentityServer



brian.noyes@solliance.net


@briannoyes


http://briannoyes.net


Demos/Slides - http://noyes.me/devint2017-ids