

Blazor 6 Fundamentals

Understanding Blazor



Daniel Villamizar

Senior Cloud Solutions Architect - MVP

@danielvillamizara – <https://www.linkedin.com/in/danielvillamizara/>

Module overview



Hello Blazor

Understanding the hosting models

Looking at the finished application

Getting your machine ready

Version Check



This version was created by using:

- ASP.NET Core 6 Blazor
- .NET 6
- Visual Studio 2022

Version Check



This course is 100% applicable to:

- ASP.NET Core 6
- .NET 6
- Visual Studio 2022 (any edition)

**Most code will work fine with older
versions of ASP.NET Core**

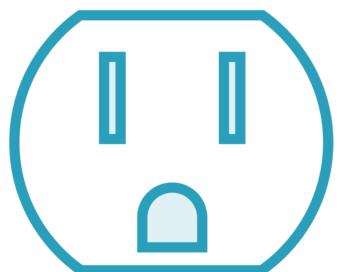
Hello Blazor!

Blazor is a framework
to build interactive web UIs
using C# and HTML.
It's part of ASP.NET Core.

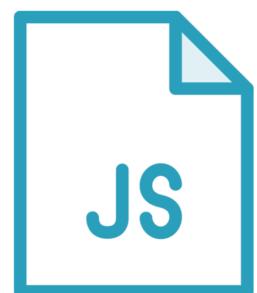
Introducing ASP.NET Core Blazor



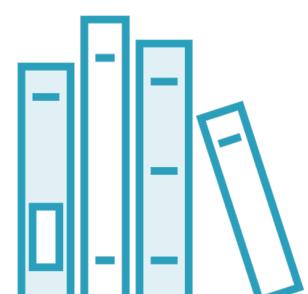
Based on WebAssembly (WASM) or run on server



No plugin, based on web standards

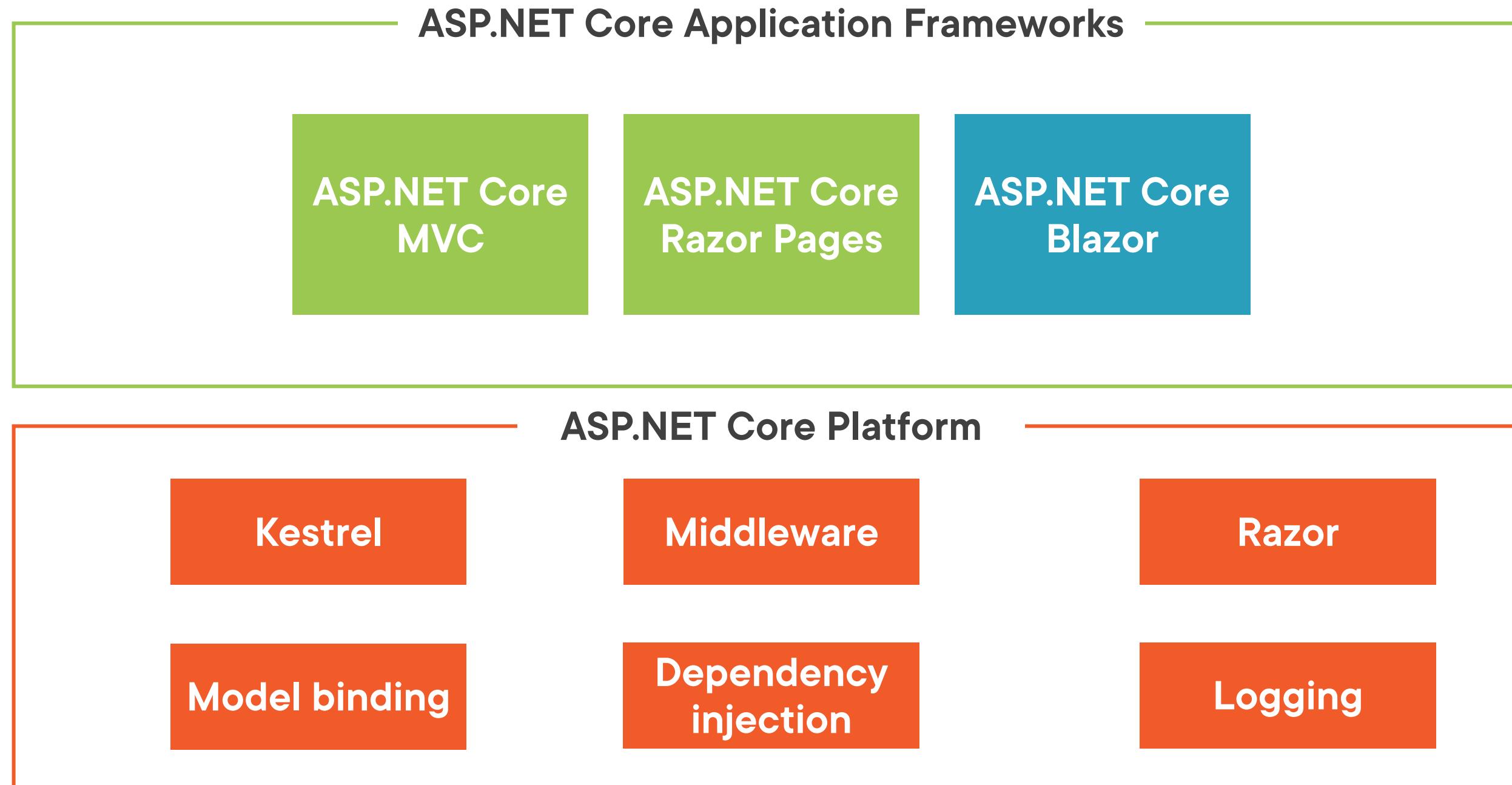


Integrate with JavaScript

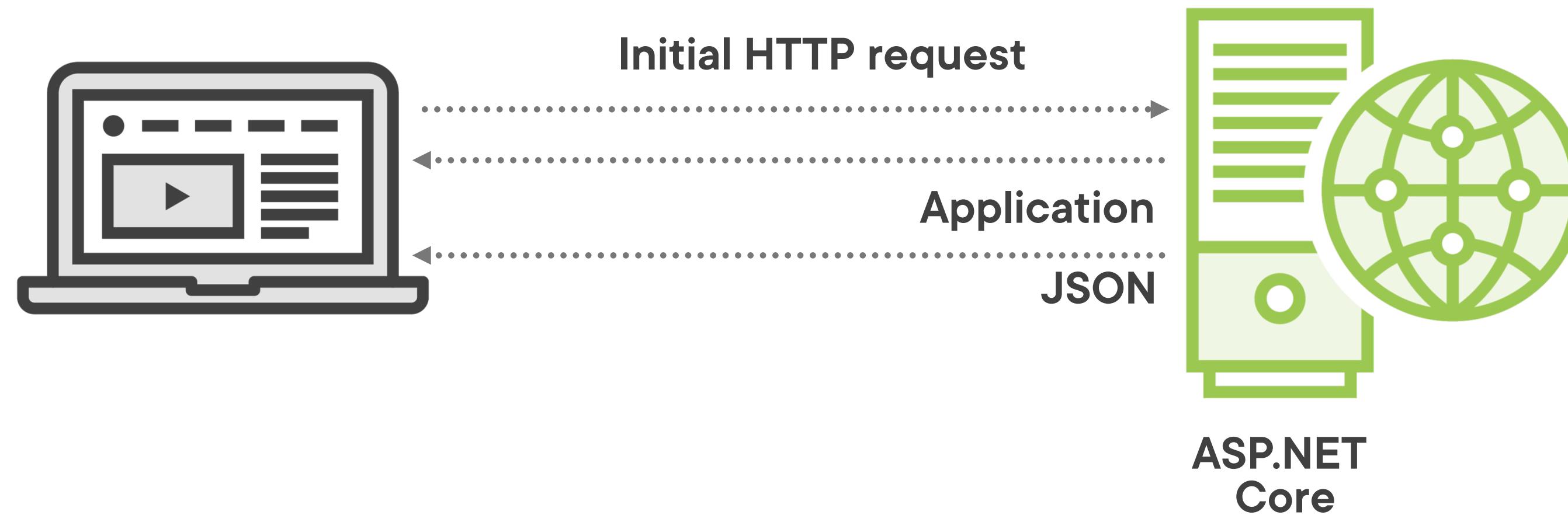


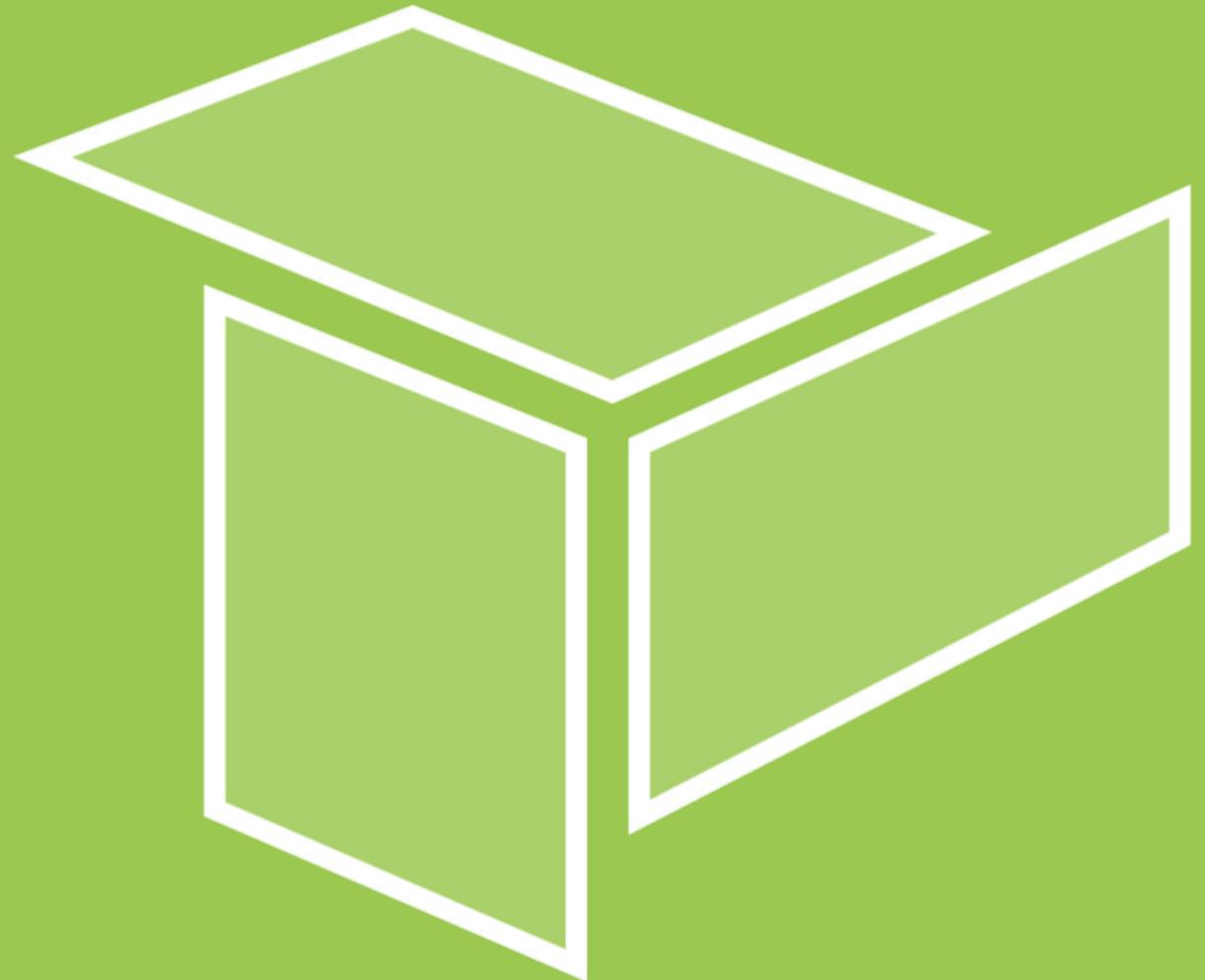
Benefits of Visual Studio and .NET including performance and libraries

ASP.NET Core Application Frameworks



Introducing ASP.NET Core Blazor





Components in Blazor

“Everything” is a component in Blazor

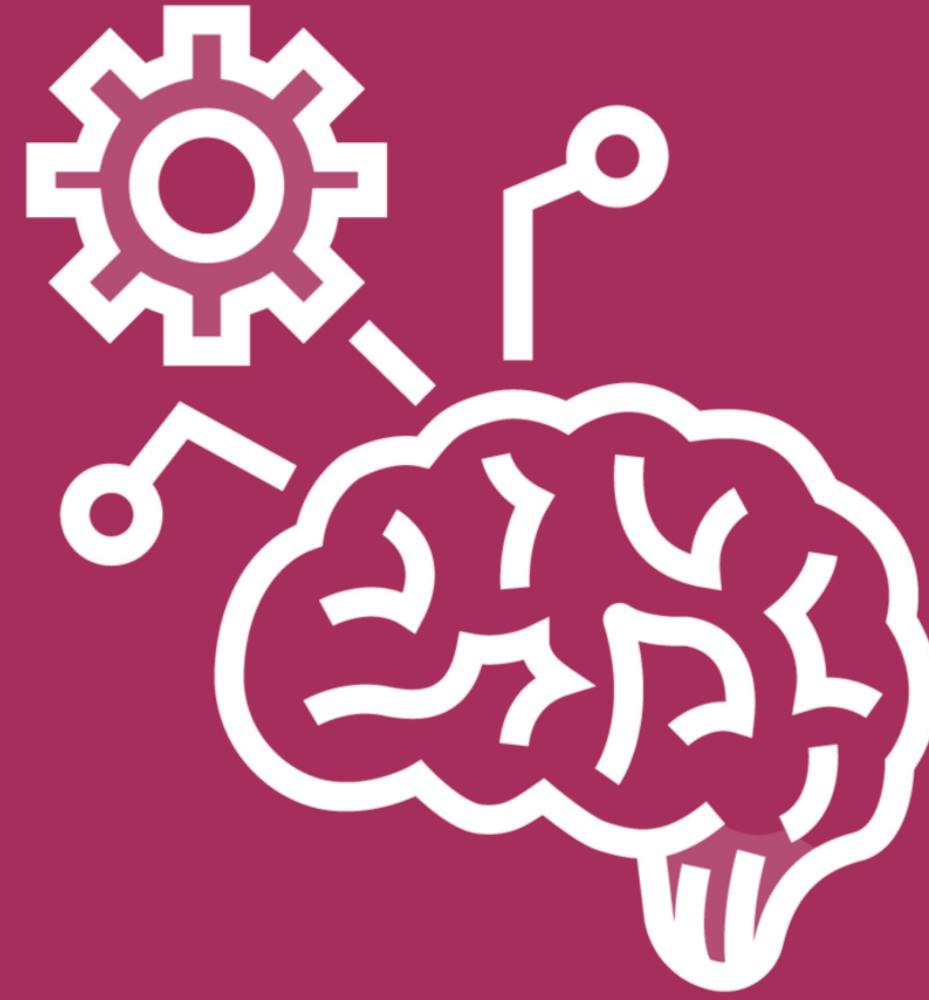
UI and logic

Created using HTML and C#

Looking at a First Component

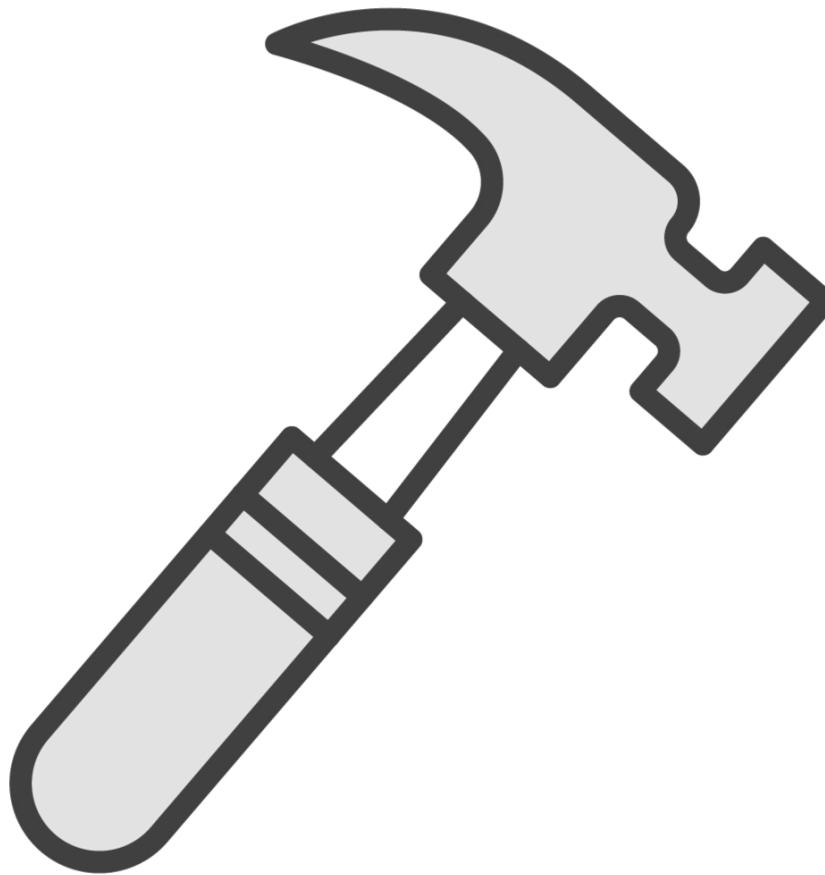
```
@page "/counter"

<h1>Counter</h1>
<p>Current count: @currentCount</p>
<button class="btn btn-primary" @onclick="IncrementCount">Click me</button>
@code {
    int currentCount = 0;
    void IncrementCount()
    {
        currentCount++;
    }
}
```



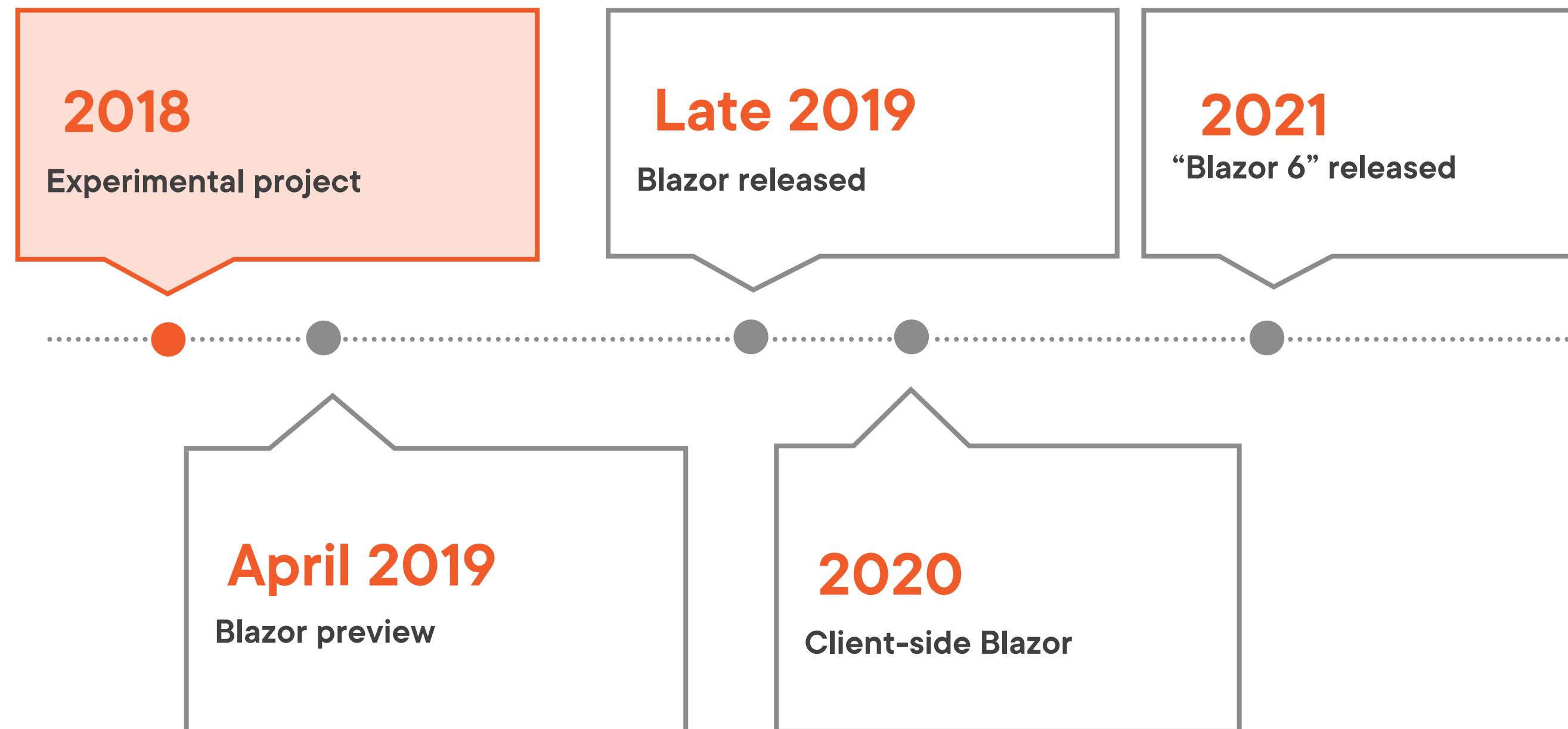
Leverage your C# skills
to build interactive web
applications.

What Can We Build with Blazor?



Applications
Websites
Games
Mobile applications

The History of Blazor



Understanding the Hosting Models

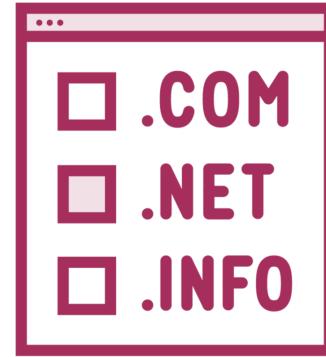
The Blazor Hosting Models

WebAssembly

Server

Hybrid

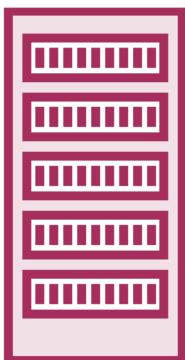
Blazor WebAssembly (WASM)



Client-side, in-browser

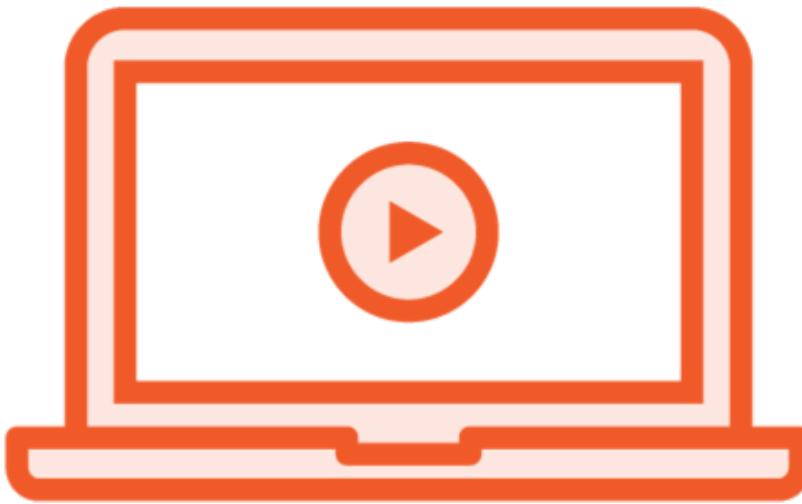


Compiled application is downloaded, along with .NET runtime

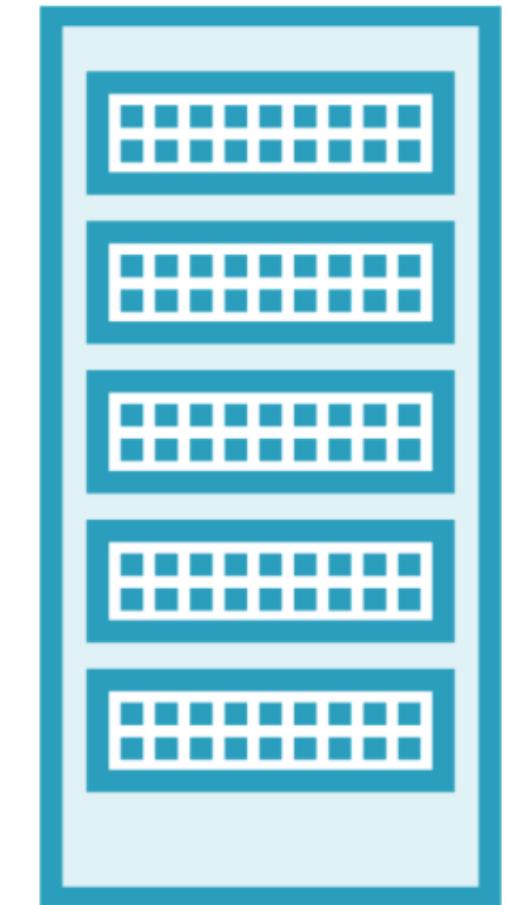


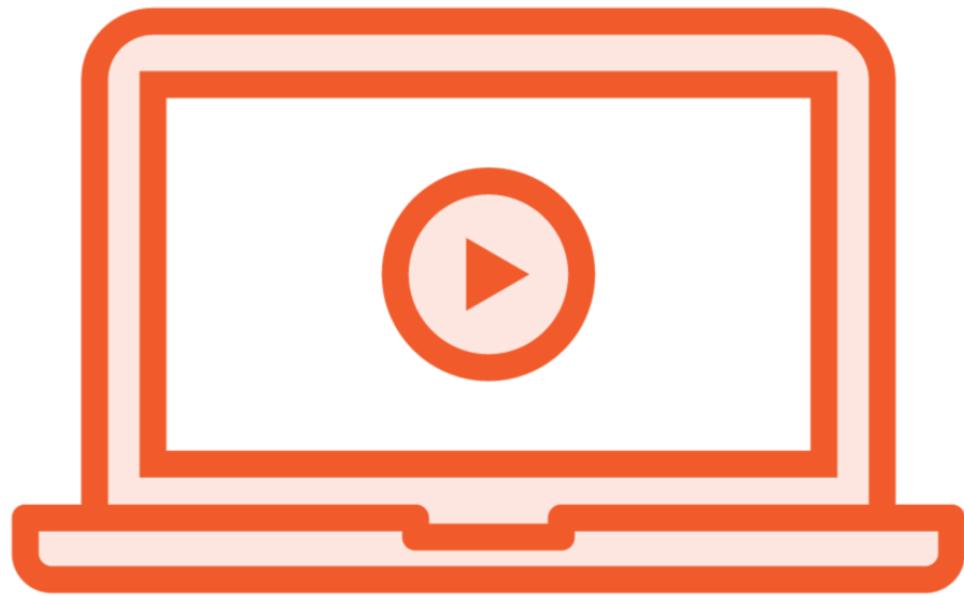
Deployed as static files

How Blazor WebAssembly Works



Blazor
app





Runs on all modern browsers

No .NET required on server

SPA user experience

But...

- Not supported on “really old browsers”
- Initial download may take longer



Running a Blazor App as a PWA

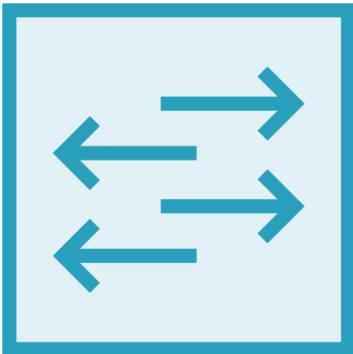
Take advantage of specific APIs of the
browser

Occasionally-connected apps

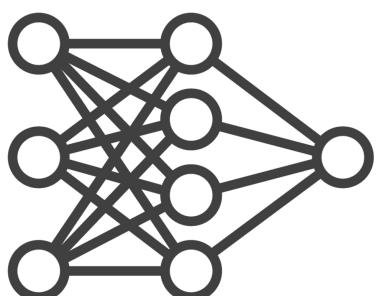
Blazor Server



Executes on server in ASP.NET Core app

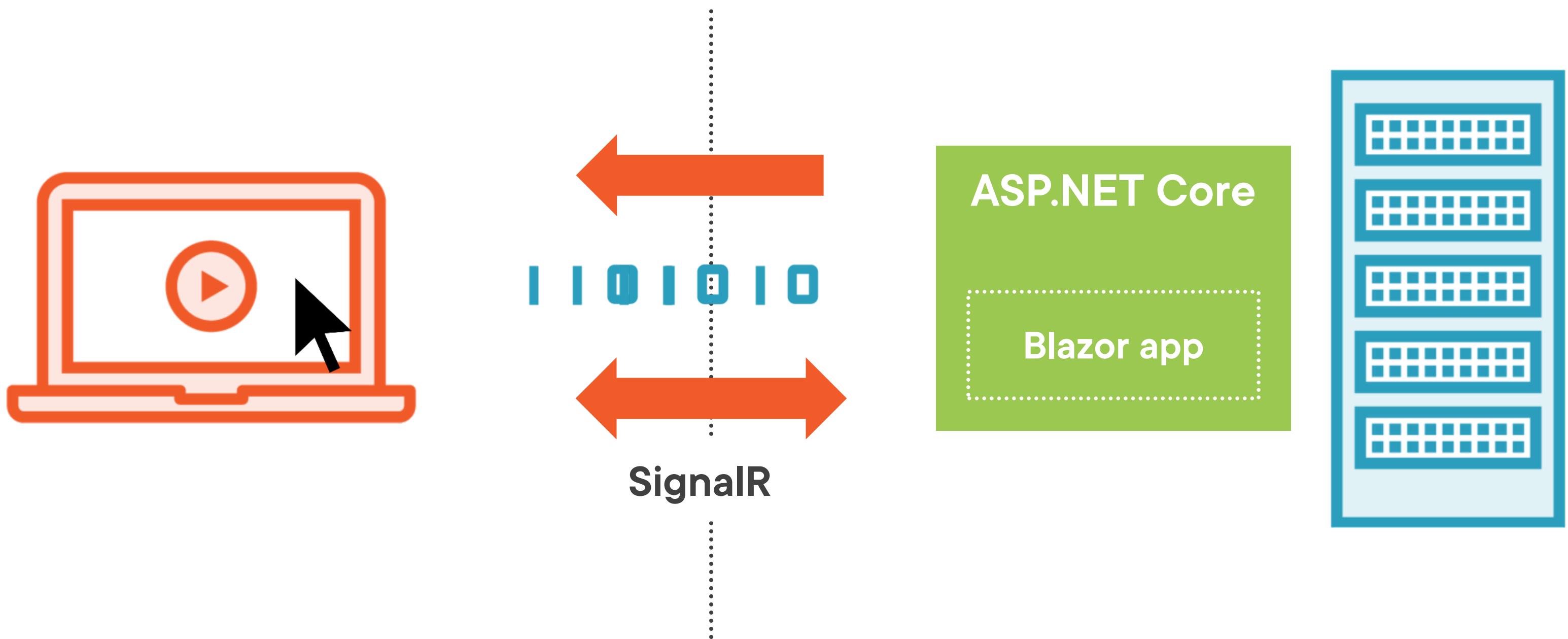


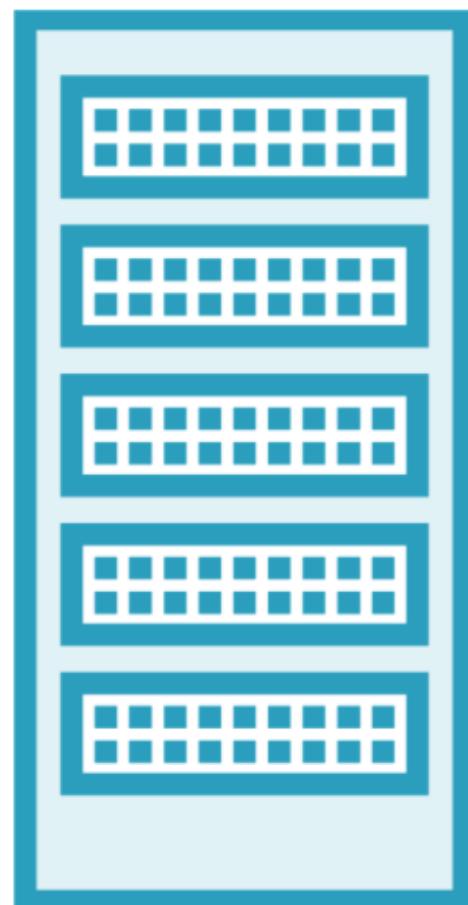
Backed by SignalR connection



Application instance is created per “circuit”

How Blazor Server Works





Small download

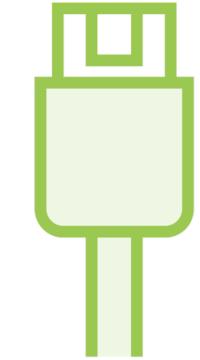
Works with all server-side APIs

Blazor apps even in non-supported browsers

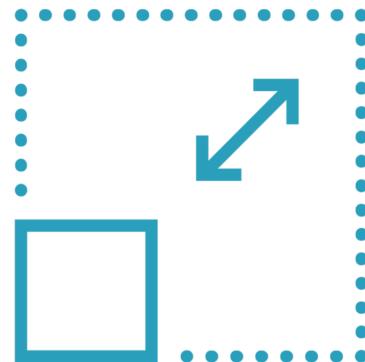
Disadvantages of Blazor Server



No offline support

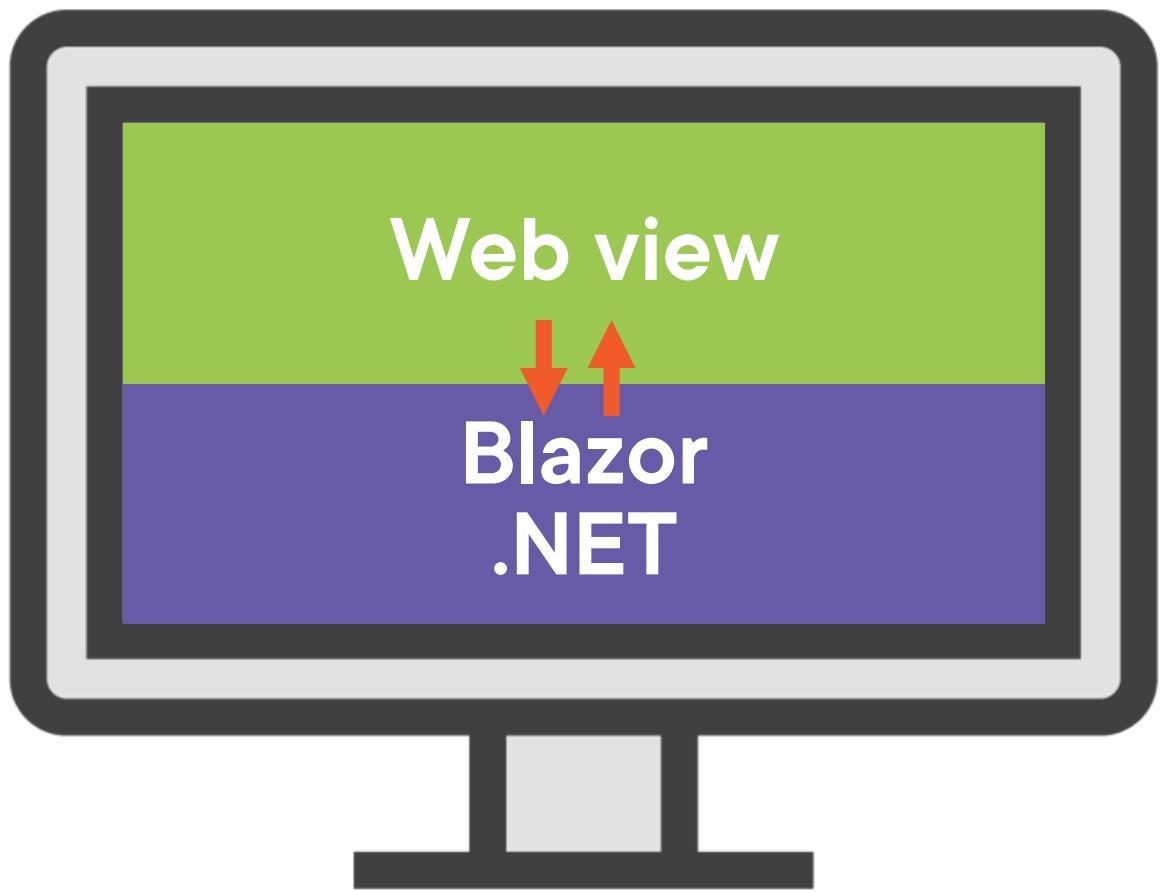


Network delay

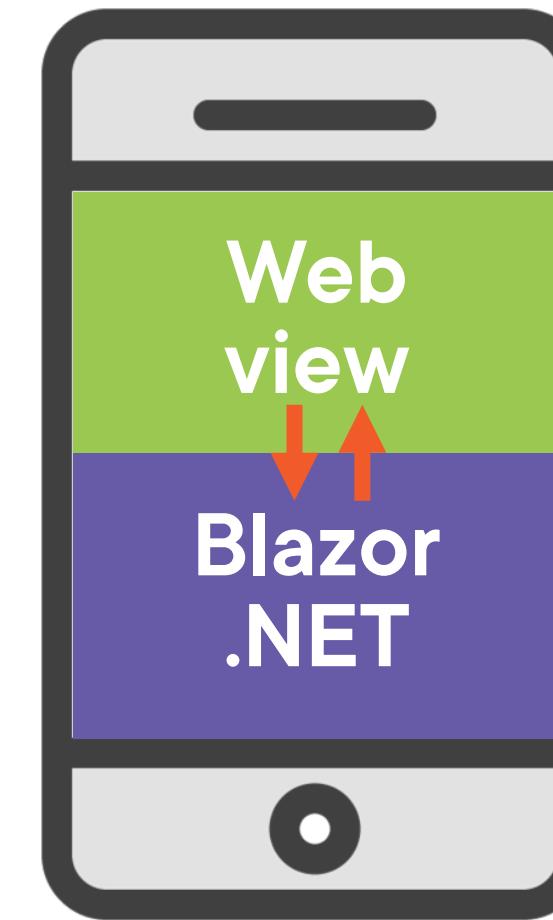


Scalability

Blazor Hybrid



Native desktop app

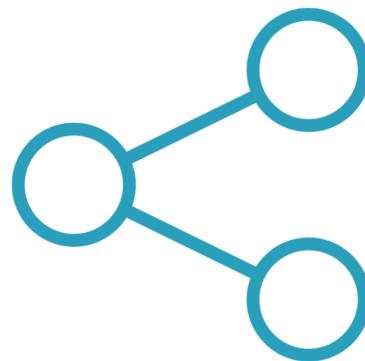


Native mobile app
with Maui

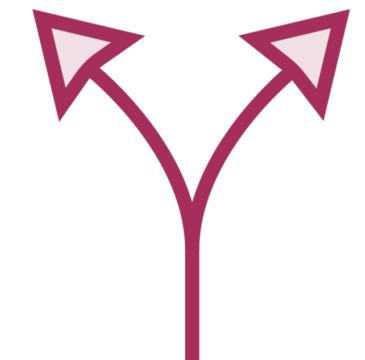
Blazor Hybrid



Access to native device features



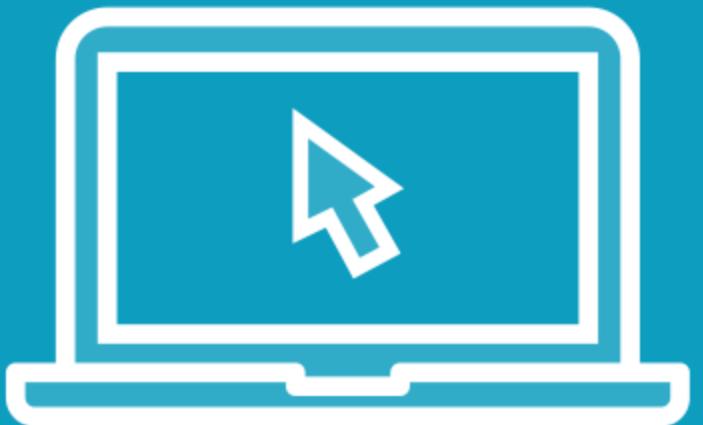
Share code between web, mobile and even desktop



Different apps for different platforms though

We will use Blazor WebAssembly in this course but “convert” the app to Blazor Server later.

Demo



Exploring the different hosting models

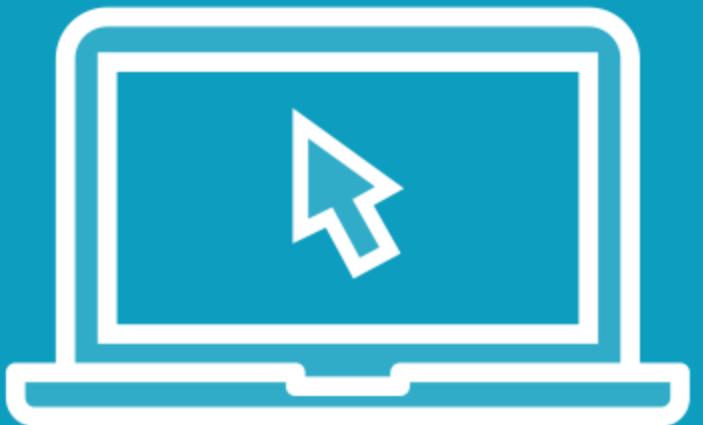
Looking at the Finished Application



The Scenario: Bethany's Pie Shop HRM

- List of employees
- Quick view and full details
- Add and edit employee

Demo



Exploring Bethany's Pie Shop HRM

Getting Your Machine Ready

Installation Requirements

Visual Studio

Windows: Visual Studio 2022

Mac: Visual Studio 2022

Not available for Linux

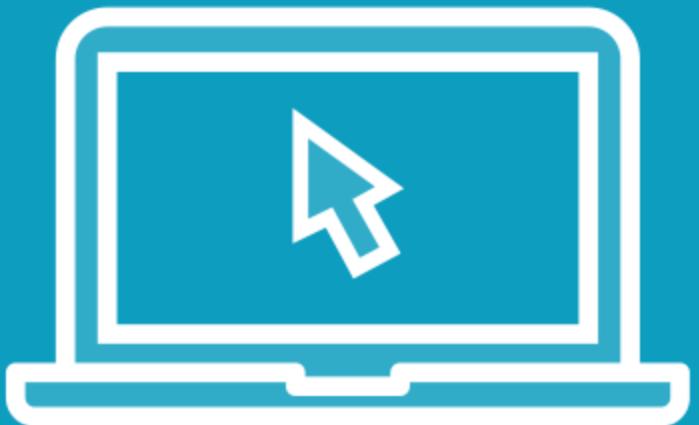
Visual Studio Code

.NET CLI

Code with C# plugin

All platforms supported

Demo



Getting your machine ready

Summary



Using Blazor, we can build rich client-side applications using C# and HTML

Part of the ASP.NET Core family



Up next:
Creating a new Blazor application