C# in the browser:
An intro to Blazor and WebAssembly

Adam Blakey *(he/him)*

.NET Notts, 2026-01-26

Mildly frustrating: I

```
console.log(3  - 1); // -> 2
console.log(3  + 1); // -> 4
console.log('3' - 1); // -> 2

console.log('3' + 1);

// A => 2
// B => 3
// C => 4
// D => 31
```

Mildly frustrating: I

```
console.log(3  - 1); // -> 2
console.log(3  + 1); // -> 4
console.log('3' - 1); // -> 2

console.log('3' + 1);

// A => 2
// B => 3
// C => 4
// D => 31
```

Mildly frustrating: II

```
let x = [1, 2, 3] + [4, 5, 6]

console.log(x);

// A => 1,2,34,5,6
// B => 1,2,3,4,5,6
// C => [5, 7, 9]
// D => [1, 2, 3, 4, 5, 6]
```

Mildly frustrating: II

```
let x = [1, 2, 3] + [4, 5, 6]

console.log(x);

// A => 1,2,34,5,6
// B => 1,2,3,4,5,6
// C => [5, 7, 9]
// D => [1, 2, 3, 4, 5, 6]
```

Mildly frustrating: III

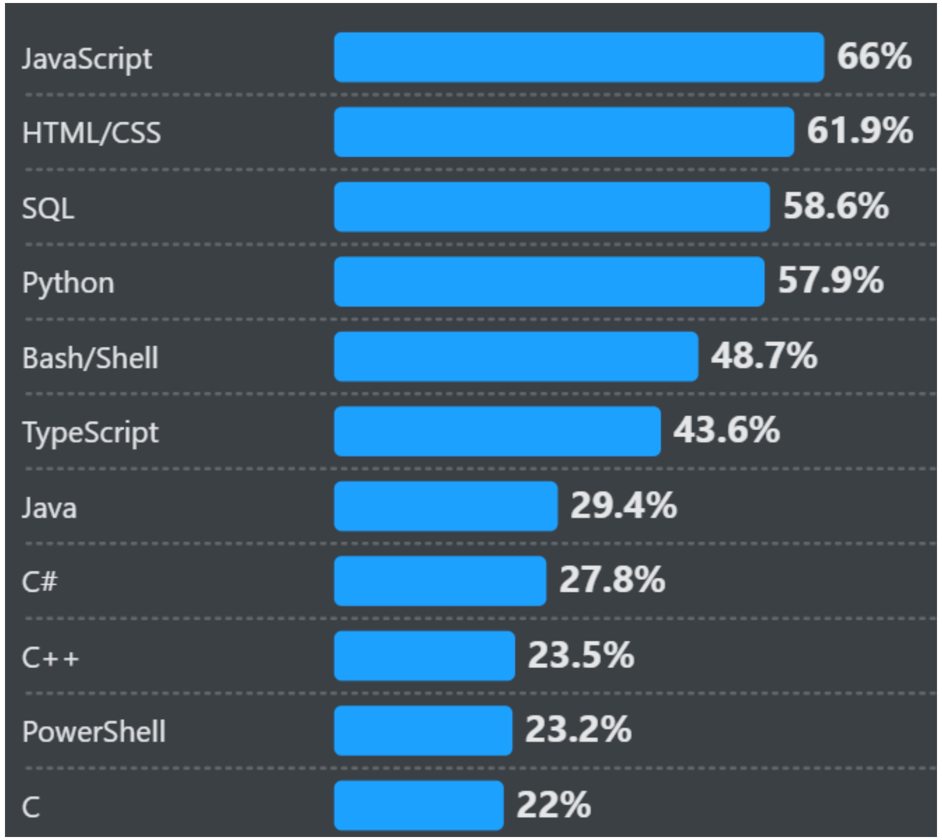
```
console.log("b" + "a" + + "a" + "a");

// A => baNaNa
// B => bNaNaa
// C => bbaa
// D => baba
```

Mildly frustrating: III

```
console.log("b" + "a" + + "a" + "a");

// A => baNaNa
// B => bNaNaa
// C => bbaa
// D => baba
```

Mildly frustrating

- More fun is available: bit.ly/wtfjavascript
- Web is instantly cross-platform
- Some people avoid JavaScript
  - TypeScript
  - **WebAssembly**



[https://survey.stackoverflow.co/2025/technology]

WebAssembly & Blazor

Wasm

- Runs natively in browsers
- Language support
- Complements JS

Blazor

- Framework from Microsoft
- Interactive web apps
- C#- and .NET-based

Hi, I’m Adam

Adam Blakey *(he/him)*

Today’s talk

1. My background
2. Introduction to **Blazor**
3. JavaScript and **WebAssembly**
4. Blazor in action

For the record:

I am not an expert in
Blazor or WebAssembly

Short background

Monkeying around (pre-2016)

- Mother tongue of PHP
- HTML/CSS
- Websites for family & friends

Savoy Systems (2024)

- Ticketing software
- Delphi & C#
- **Blazor!**

MMath & PhD (2016 & 2020)

- Blood flow in the placenta
- Fortran & Python
- Teaching

Capital One (2025)

- Mostly backend
- Java, Kotlin & Swift
- Customer Engagement

2. Blazor

A brief history of interactive web apps

1990s — Web 1.0
- Mostly static
- HTML/PHP/ASP
- JavaScript (1995)

2000s — Interactivity
- e.g., AJAX, jQuery, Silverlight, Flash

2010-ish — Single-page apps
- e.g., AngularJS, Backbone.js

2013–2016 — Modern frameworks
- e.g., React, Vue.js

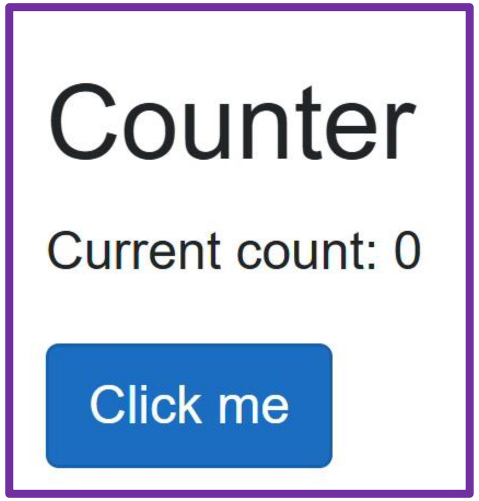
2017 — WebAssembly released
- Chrome, Edge, Firefox, and WebKit
- Browser execution

2018 — Blazor experiment
- Unofficial project
- Blazor = "Browser" + "Razor"

2020 — Blazor released
- Officially supported
- WebAssembly

What is Blazor?

- Web framework
- SPA
- Uses Razor



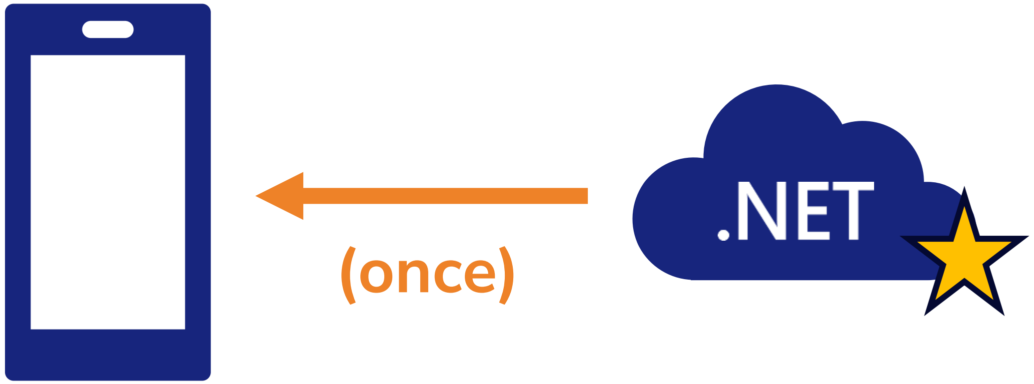

```
@page "/counter"
@rendermode InteractiveServer

<PageTitle>Counter</PageTitle>

<h1>Counter</h1>

<p role="status">Current count: @currentCount</p>

<button class="btn btn-primary" @onclick="IncrementCount">Click me</button>

@code {
    private int currentCount = 0;

    private void IncrementCount()
    {
        currentCount++;
    }
}
```

Hosting models

- Blazor Web app (Server)
  - Static
  - Interactive server
  - Interactive WebAssembly
  - Interactive auto
- WebAssembly (Client)
- Hybrid (e.g., .NET MAUI)

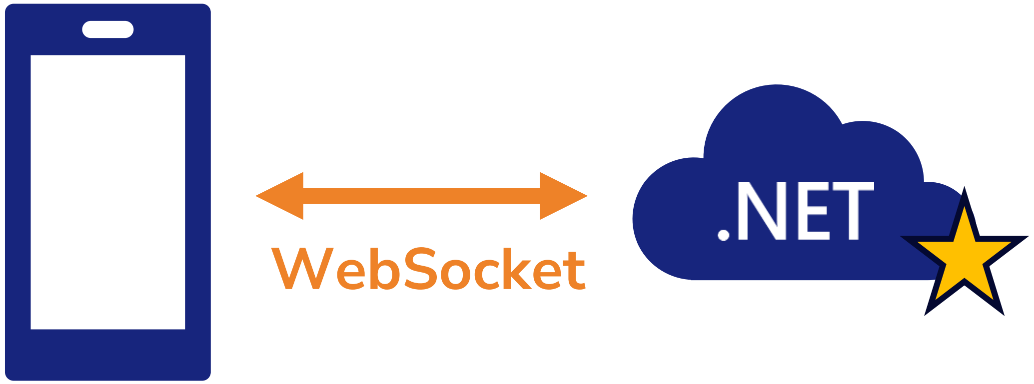
Hosting models

- Blazor Web app (Server)
  - **Static**
  - Interactive server
  - Interactive WebAssembly
  - Interactive auto
- WebAssembly (Client)
- Hybrid (e.g., .NET MAUI)



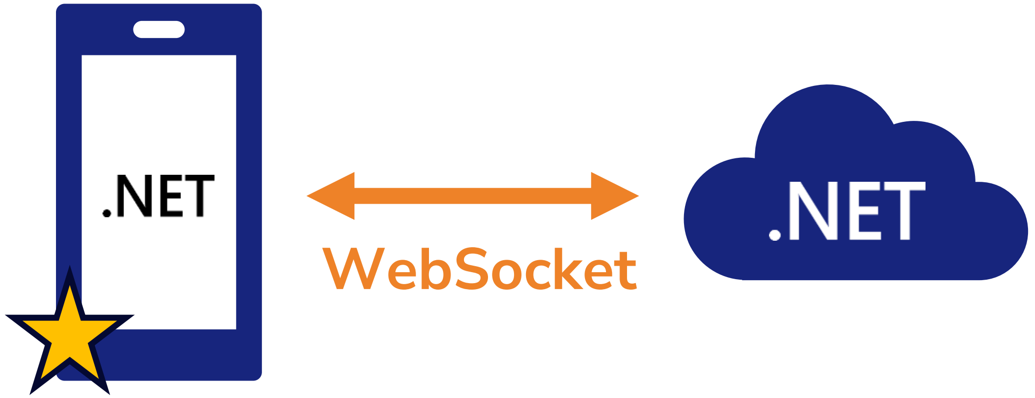
Hosting models

- Blazor Web app (Server)
  - Static
  - **Interactive server**
  - Interactive WebAssembly
  - Interactive auto
- WebAssembly (Client)
- Hybrid (e.g., .NET MAUI)



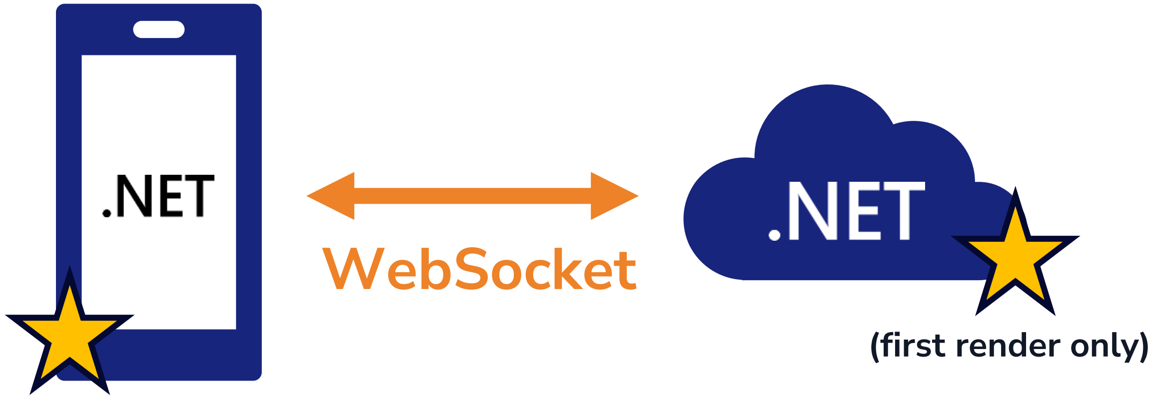
Hosting models

- Blazor Web app (Server)
  - Static
  - Interactive server
  - **Interactive WebAssembly**
  - Interactive auto
- WebAssembly (Client)
- Hybrid (e.g., .NET MAUI)



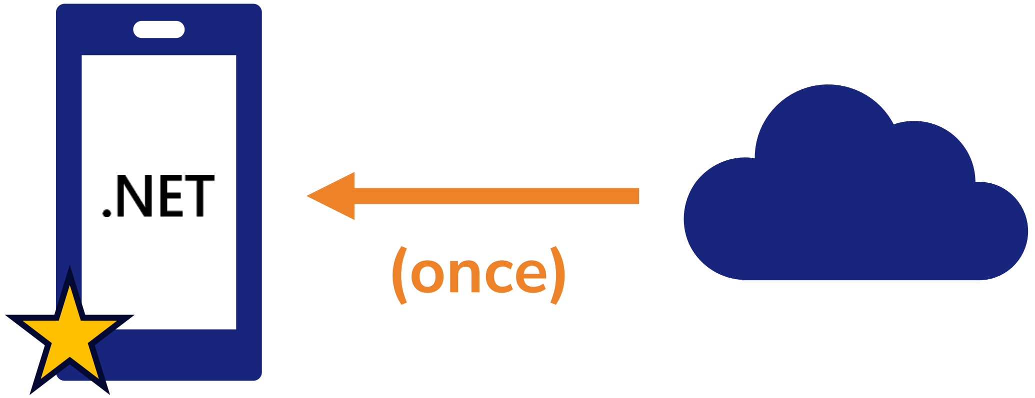
Hosting models

- Blazor Web app (Server)
  - Static
  - Interactive server
  - Interactive WebAssembly
  - **Interactive auto**
- WebAssembly (Client)
- Hybrid (e.g., .NET MAUI)



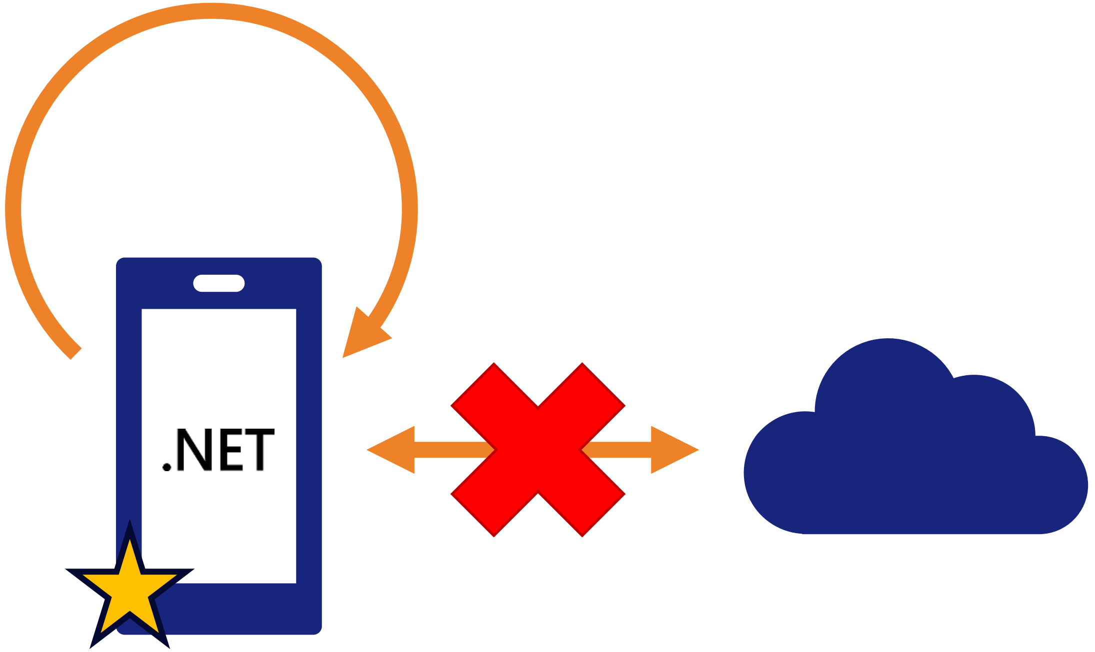
Hosting models

- Blazor Web app (Server)
  - Static
  - Interactive server
  - Interactive WebAssembly
  - Interactive auto
- **WebAssembly (Client)**
- Hybrid (e.g., .NET MAUI)



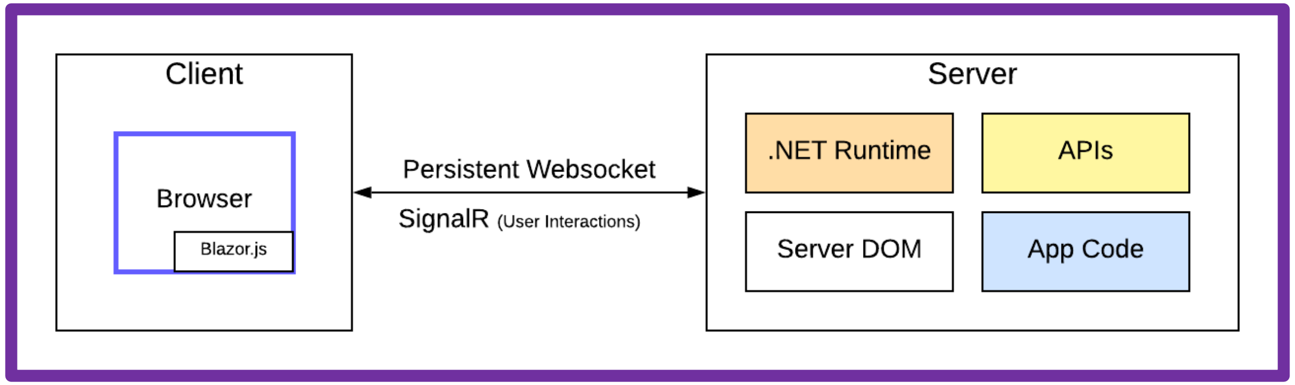
Hosting models

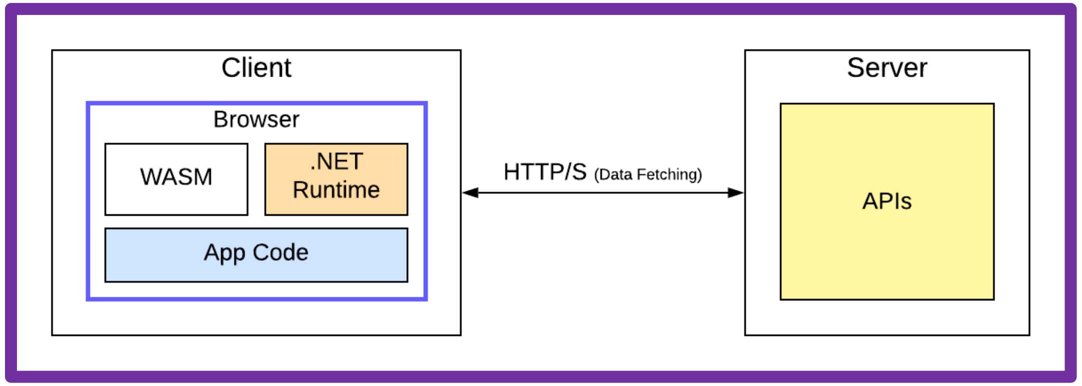
- Blazor Web app (Server)
  - Static
  - Interactive server
  - Interactive WebAssembly
  - Interactive auto
- WebAssembly (Client)
- **Hybrid (e.g., .NET MAUI)**



Some details

- Components are .NET classes
- Binding parameters
- JavaScript interop
- SignalR

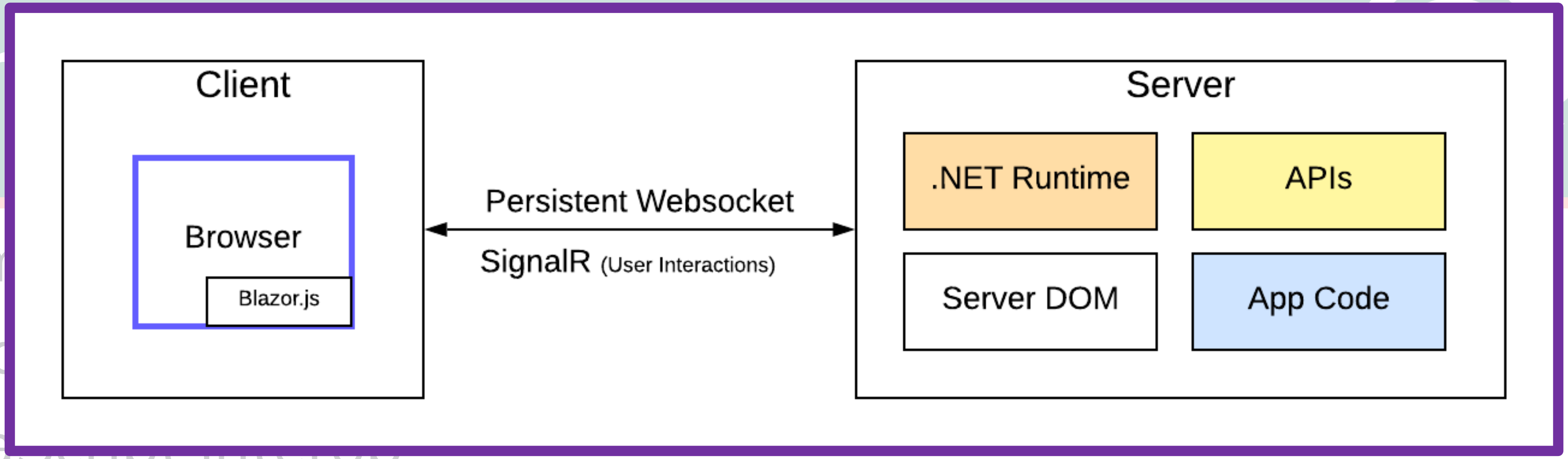


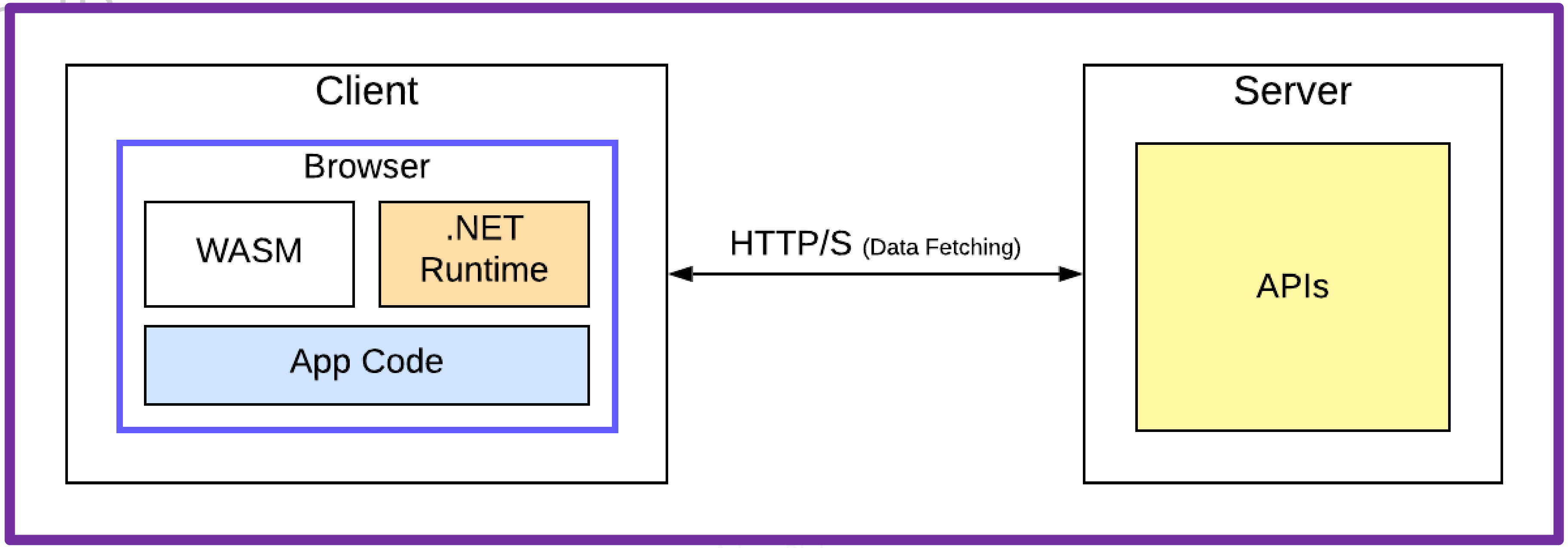


[images from https://www.telerik.com/]

Sor

- Com
- Binc
- JavaScript interop
- Sign





Keen?

- https://dotnetcore.show/
- https://blazor-university.com/
- https://learn.microsoft.com
- https://dotnet.microsoft.com/en-us/learn/aspnet/blazor-tutorial/intro
- https://blog.stevensanderson.com/2018/02/06/blazor-intro/
- Google, obvs

3. WebAssembly

Introducing WebAssembly

- Runs in browsers
- Low-level
- Not a replacement to JavaScript
- JavaScript, Web, and WASI APIs

WebAssembly

- Client-side computation
- JavaScript for non-intensive tasks
- **Made with Wasm:**
  - **Figma**
  - **Ebay** (barcode scanning)
  - **WordPress Playground**
  - **Photoshop** (beta)
  - **BBC iPlayer** (soon?)

- Not-so-good for:
  - Blogs
  - CRUD apps
  - SEO-critical apps
  - Heavy DOM manipulation

[more at https://web.dev and https://madewithwebassembly.com]

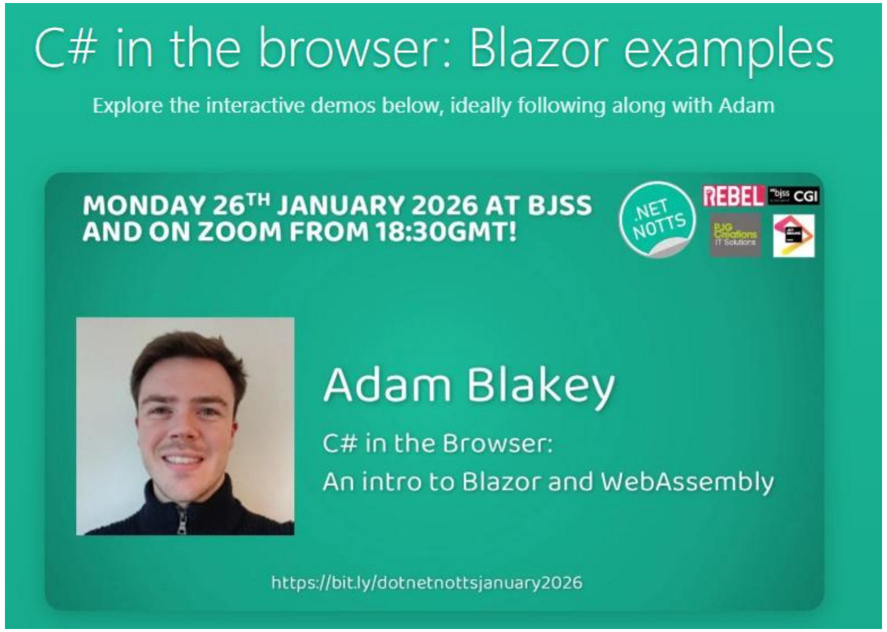
Use WebAssembly

- Ada
- AssemblyScript
- C/C++
- C#
- Cobol
- D
- Dart
- F#
- Go
- Grain
- Haskell
- Java
- Kotlin
- Lua
- Moonbit
- Pascal
- PHP
- Python
- R
- RemObjects Elements
- Ruby
- Rust
- Scala.js
- Scheme
- Swift
- Zig

[https://webassembly.org/getting-started/developers-guide/]

4. Demos

Demos

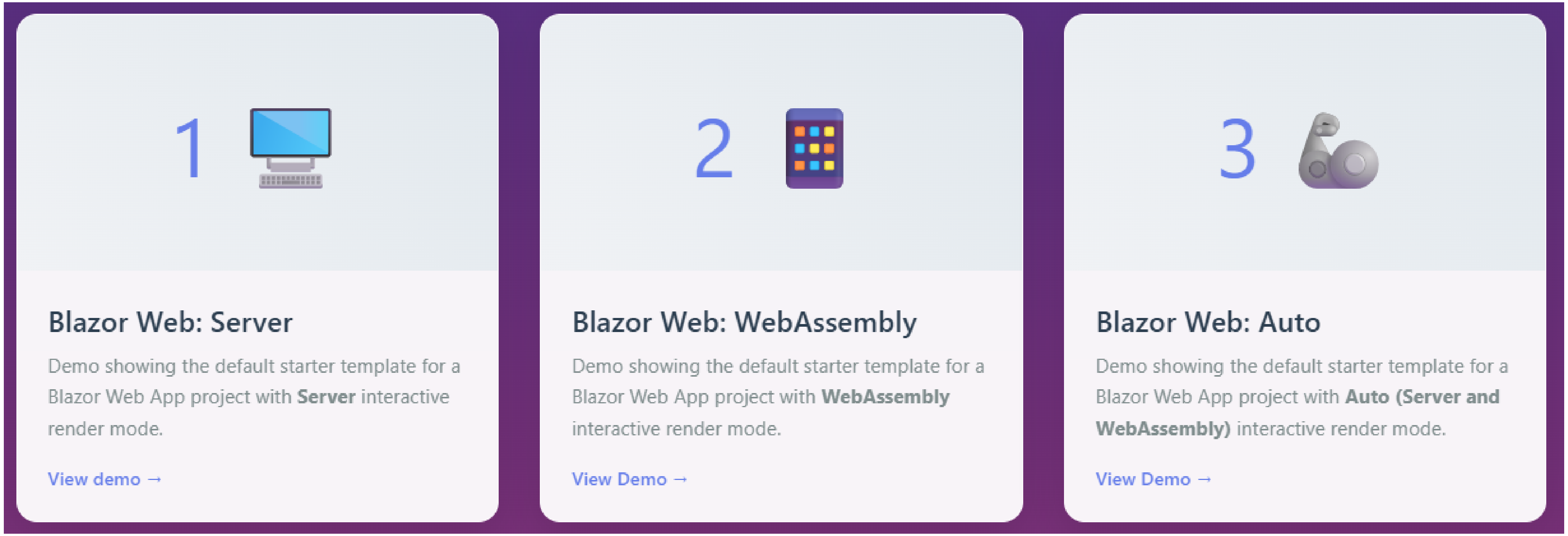


https://blazor.blakey.family

Demos 1—3

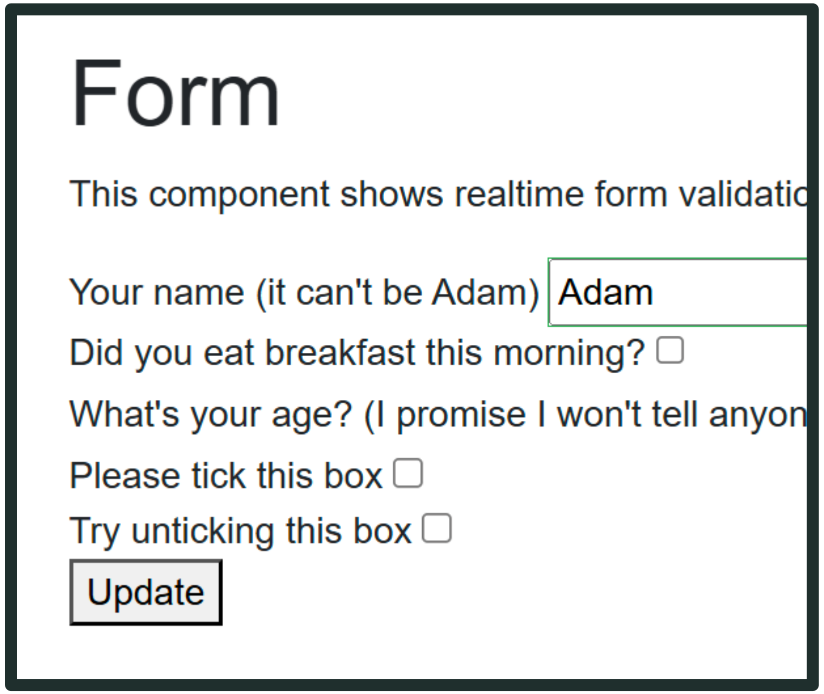
https://blazor.blakey.family

- Default app x3
- Server offline
- File sizes



Demo 4

https://blazor.blakey.family

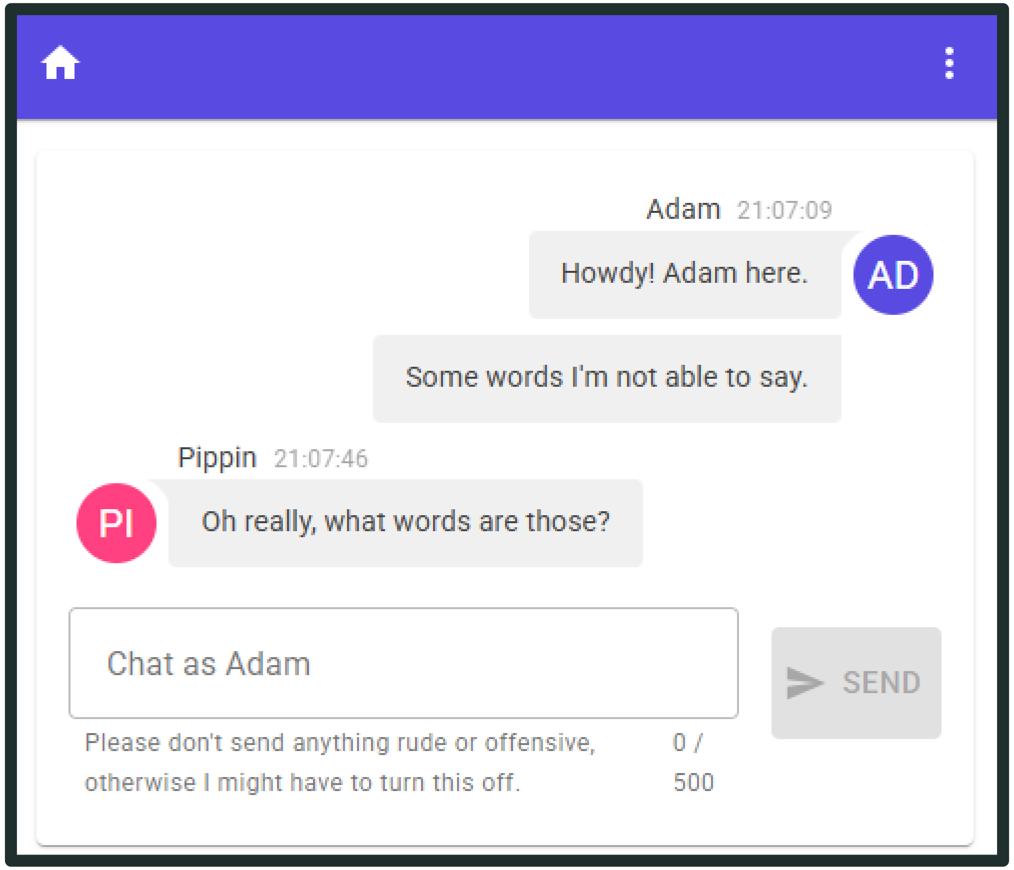


- Silly form validation
- Entirely client-side in WASM
- Bound checkboxes

```
<InputCheckbox @bind-Value="formModel.TickThisBox" />
```

Demo 5

https://blazor.blakey.family



- Features
  - Real-time chat
  - Event handlers in Singleton C# web server service
  - Client-side browser storage
- Packages
  - MudBlazor
  - Blazored.LocalStorage
  - Profanity.Detector
  - morelinq

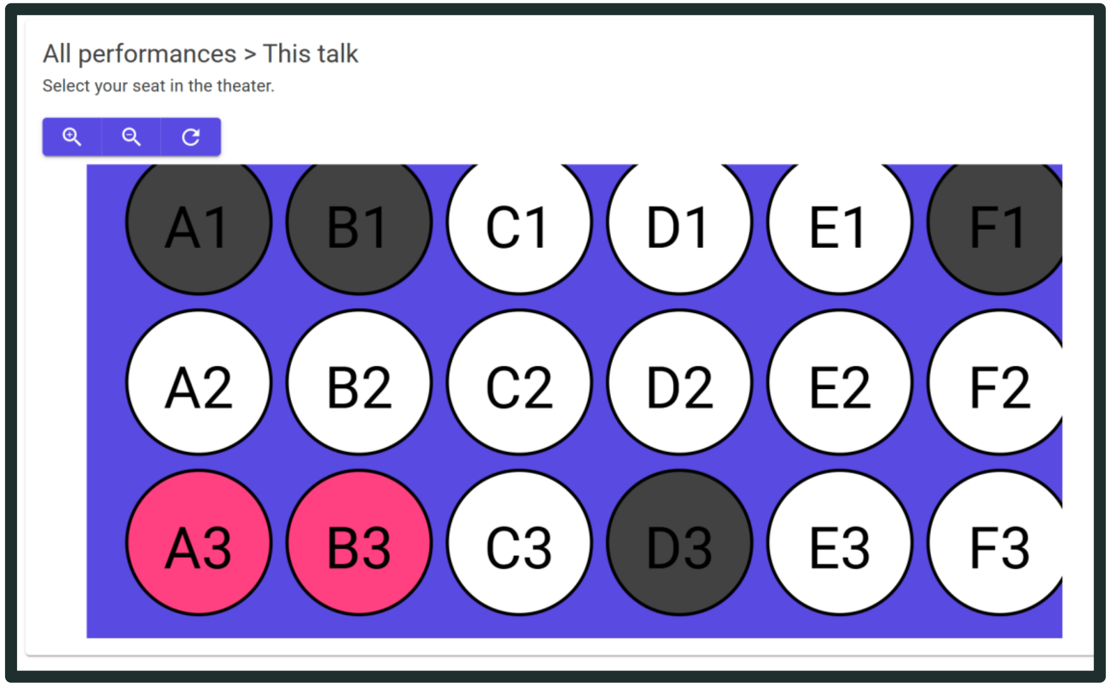
Demo 6

https://blazor.blakey.family

- Features
  - SignalR hubs
  - MouseUp
  - Simple CSS transformations
  - **No** persistence

Demo 7

https://blazor.blakey.family



- Features
  - Select seat in real-time
  - Clicking, dragging, and zooming
  - **No** persistence

Demo summary

https://blazor.blakey.family

- Hosting models
- Data binding
- Entirely Wasm apps
- Rich package availability
- Real-time via event handlers & SignalR
- *Super* easy to code-up

Talk summary

- Blazor is pretty cool
  - Interactive C# web apps
- WebAssembly is pretty cool
  - Client-side byte-code

Adam Blakey

https://adam.blakey.family

adam.blakey.family

/adamblakey

Form code

```
<EditForm EditContext="editContext" OnValidSubmit="Submit">
    <div>
        <label>
            Your name (it can't be Adam)
            <InputText @bind-Value="formModel.YourName" />
            <ValidationMessage For="() => formModel.YourName" />
        </label>
    </div>
    <div>
        <label>
            Did you eat breakfast this morning?
            <InputCheckbox @bind-Value="formModel.EatBreakfast" />
            <ValidationMessage For="() => formModel.EatBreakfast" />
        </label>
    </div>
    <div>
        <label>
            What's your age? (I promise I won't tell anyone)
            <InputNumber @bind-Value="formModel.Age" />
            <ValidationMessage For="() => formModel.Age" />
        </label>
    </div>
```

Game code

```
@if (UserIcon == null)
{
    <h3>Select Your Icon</h3>
    <div class="icon-selection">
        @for (int i = 0; i < AvailableIcons.Count; i++)
        {
            int iLocal = i;
            <Blazicon Svg="AvailableIcons[i]" style="height: 3em; width: 3em;" @onclick="() => SelectIcon(iLocal)" class="@($"icon-choice {(IconAlreadySelected(iLocal) ?
"disabled" : "")}")" />
        }
    </div>
}
else
{
    <div class="map-container" @onmouseup="StopDragging" @onmousemove="Drag">
        @foreach (var kvp in Characters)
        {
            <Blazicon Svg="AvailableIcons[kvp.Value.IconId]" style="@($"position: absolute; left: {Math.Clamp(kvp.Value.X, MIN_POSITION, MAX_POSITION)}px; top: {Math.Clamp
(kvp.Value.Y, MIN_POSITION, MAX_POSITION)}px;")" class="@($"character {(kvp.Key == UserId ? "draggable user-character" : "")}")"
                      @onmousedown="() => StartDragging(kvp.Key)" />
        }
    </div>
}
```

```
            // Existing characters.
            Characters = CharacterService.Characters;

            hubConnection = new HubConnectionBuilder()
                .WithUrl(NavigationManager.ToAbsoluteUri("/characterhub"))
                .WithAutomaticReconnect()
                .Build();

            hubConnection.On<string, Game.Models.CharacterState>("ReceivePosition", (userId, state) =>
            {
                Characters[userId] = state;
                InvokeAsync(StateHasChanged);
            });

            hubConnection.On<string>("ReceiveCharacterRemoval", (userId) =>
            {
                Characters.Remove(userId);
                InvokeAsync(StateHasChanged);
            });

            await hubConnection.StartAsync();
```

```
private void StartDragging(string userId)
{
    if (userId != UserId) return;
    IsDragging = true;
    DraggingUserId = userId;
}

private void Drag(MouseEventArgs e)
{
    if (IsDragging && DraggingUserId == UserId && Characters.TryGetValue(UserId, out var state))
    {
        state.X = (int)e.ClientX - (CHARACTER_WIDTH / 2) - 2; // 2px for the container border.
        state.Y = (int)e.ClientY - (CHARACTER_WIDTH / 2) - 2; // 2px for the container border.
        Characters[UserId] = state;
    }
}

private async void StopDragging()
{
    await SendPosition();

    IsDragging = false;
    DraggingUserId = null;
}

private async Task SendPosition()
{
    if (hubConnection is not null)
    {
        await hubConnection.SendAsync("UpdatePosition", UserId, Characters[UserId]);
    }
}
```