

Lights, Camera, Blazor:

Modernising cinema software
with C# and Blazor

Adam Blakey (*he/him*)

Notts Techfast, 2025-07-16

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
```

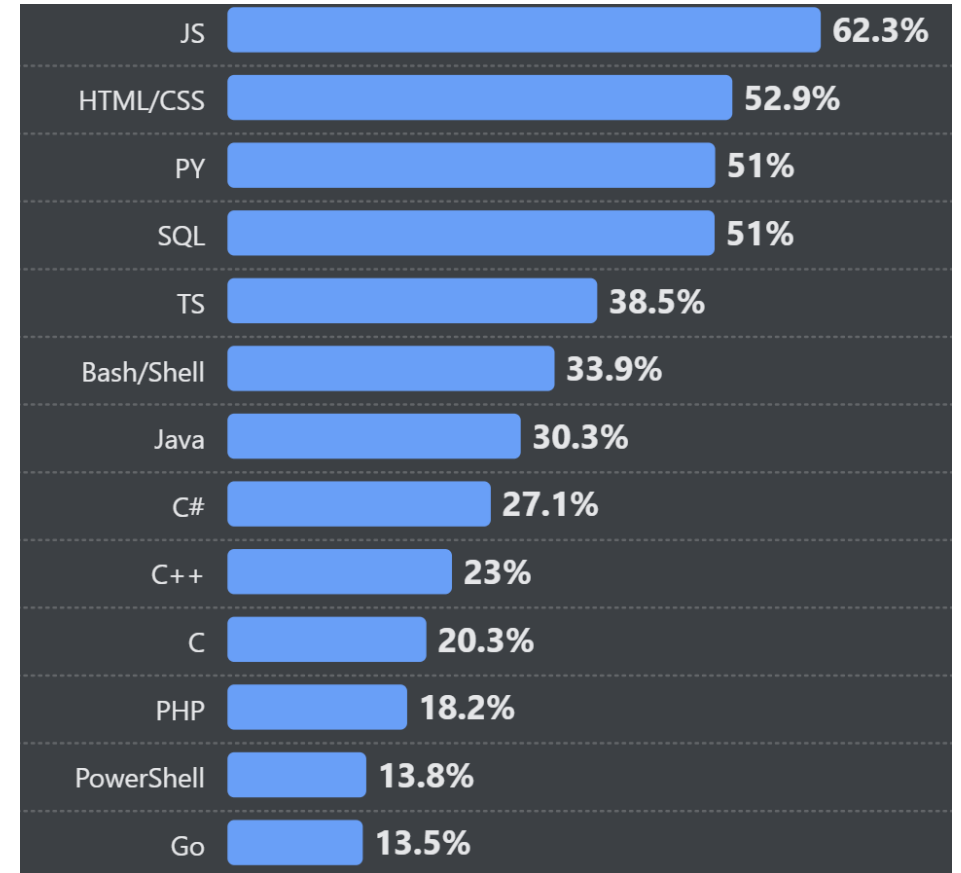
```
// B => bNaNaa
```

```
// 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/2024/technology>]

Introducing WebAssembly

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



Mozilla 



fastly[®]

intel[®]



Google





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
 - a) (Tiny) snippet of Savoy Systems' work



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



Mathematics PhD (2020)

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



Mathematics MMath (2016)

- Focus on scientific computation
- MATLAB & C++

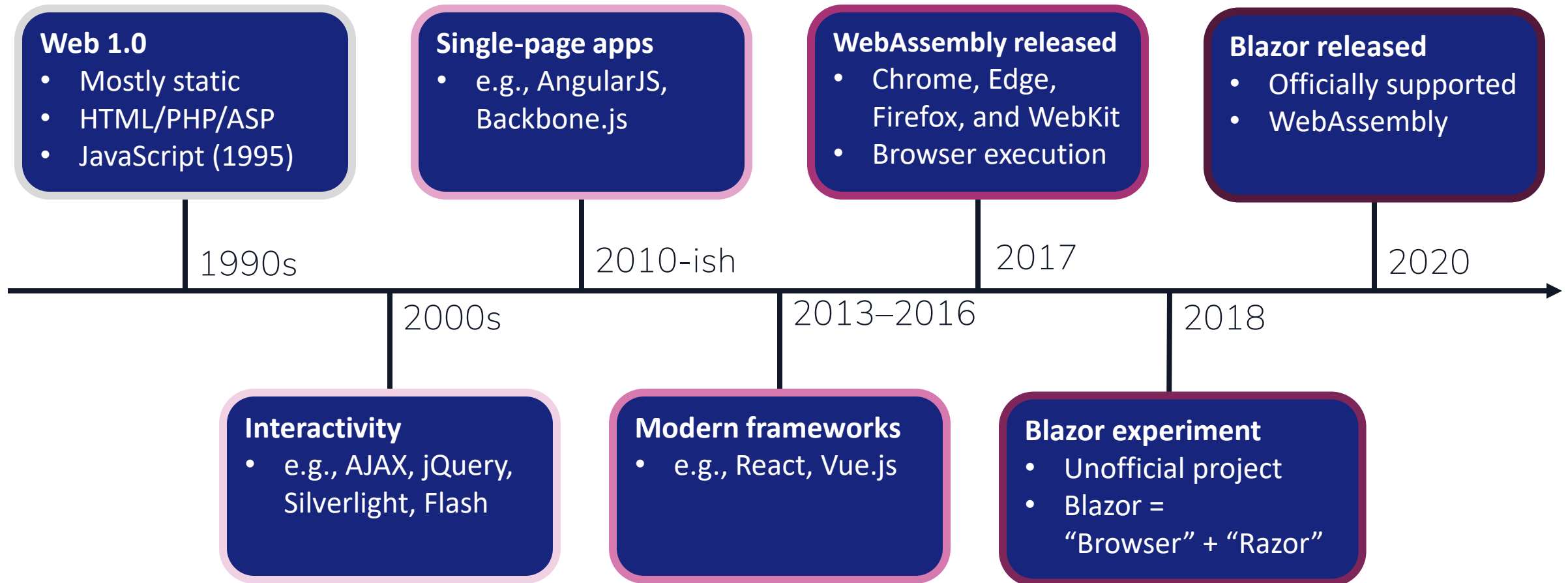


Savoy Systems (2024)

- Ticketing software
- Delphi & C#
- Blazor!

2. Blazor

A brief history of interactive web apps



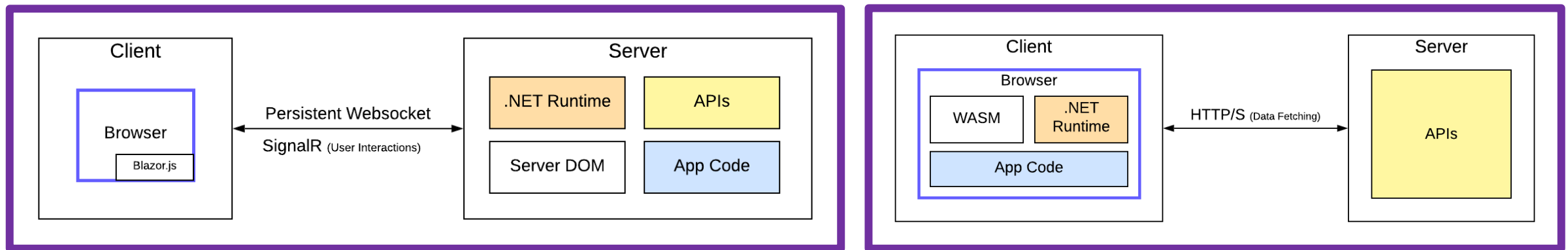
What is Blazor?

- Web framework
- SPA
- Uses Razor
- Hosting models:
 - Blazor Web app (Server)
 - Static
 - Interactive server
 - Interactive WebAssembly
 - Interactive auto
 - WebAssembly (Client)
 - Hybrid (e.g., .NET MAUI)



Some details

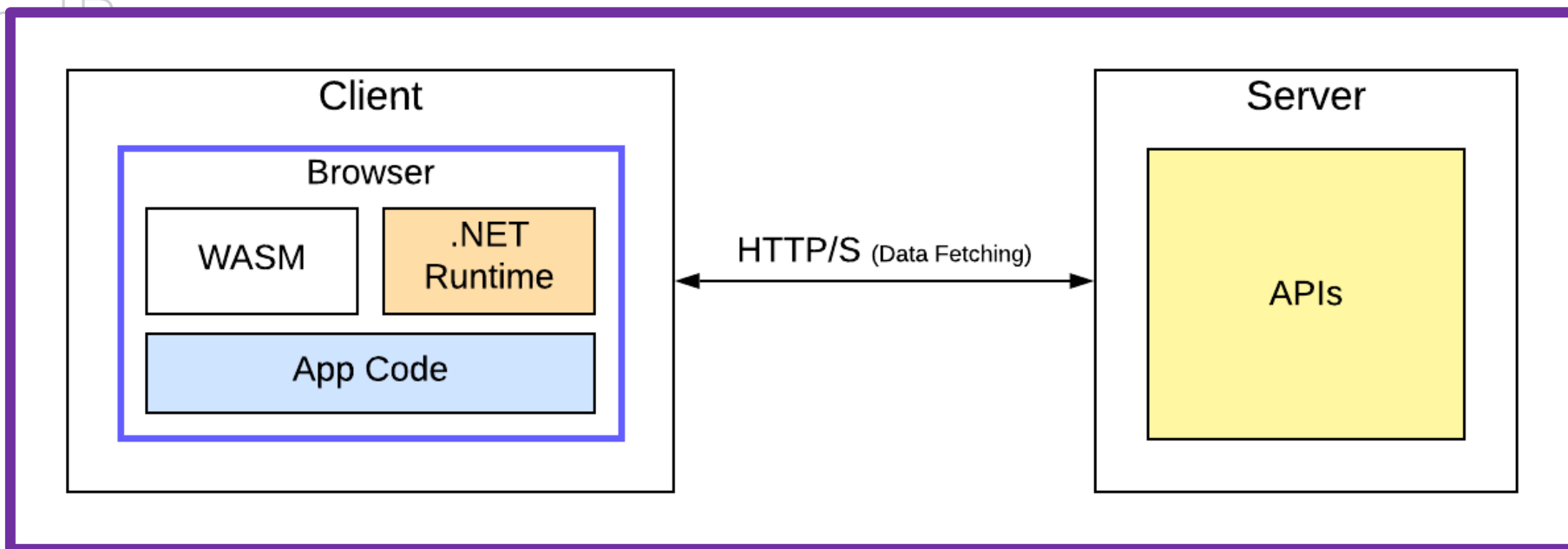
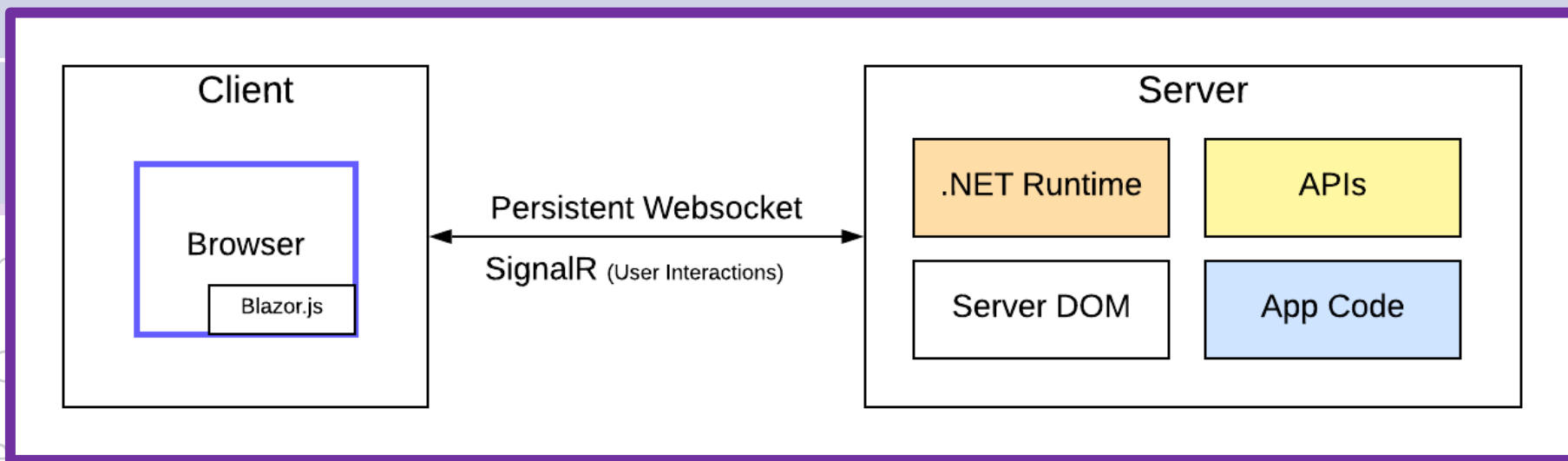
- Components are .NET classes
- Binding parameters
- JavaScript interop
- SignalR



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

Some

- Com
- Bind
- JavaScript interop
- SignalR



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

WebAssembly

- Client-side computation
- More locked-down than JavaScript
- JavaScript for non-intensive tasks
- Figma, Ebay (barcode scanning), WordPress Playground, Photoshop (beta), BBC iPlayer (soon?)

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

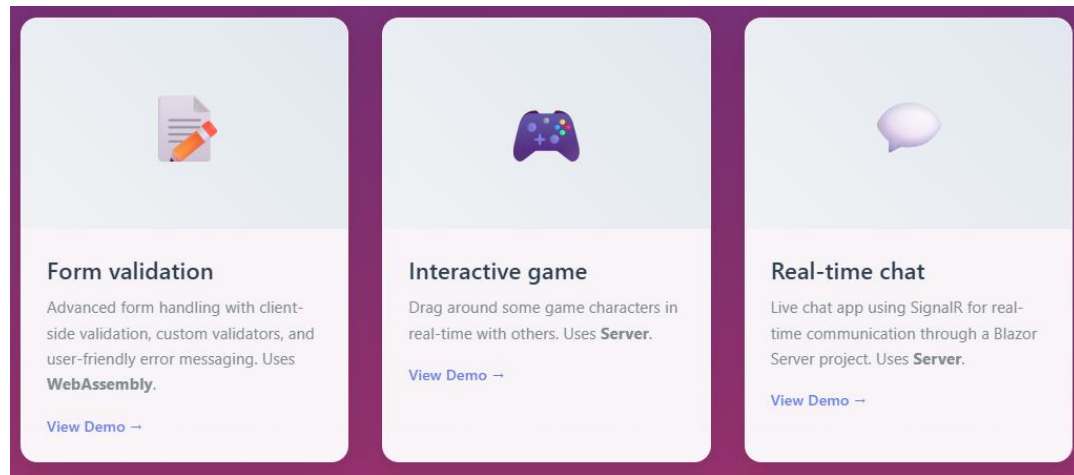
Use WebAssembly

- 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

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

4. Demos

Demos



<https://techfast.blakey.family>



Demos 1—3

<https://techfast.blakey.family>

- Default app x3
- Server offline
- File sizes

1 

Blazor Web: Server
Demo showing the default starter template for a Blazor Web App project with **Server** interactive render mode.
[View demo →](#)

2 

Blazor Web: WebAssembly
Demo showing the default starter template for a Blazor Web App project with **WebAssembly** interactive render mode.
[View Demo →](#)

3 

Blazor Web: Auto
Demo showing the default starter template for a Blazor Web App project with **Auto (Server and WebAssembly)** interactive render mode.
[View Demo →](#)

Demo 4

<https://techfast.blakey.family>

Form

This component shows realtime form validation

Your name (it can't be Adam)

Did you eat breakfast this morning? ☐

What's your age? (I promise I won't tell anyone)

Please tick this box ☐

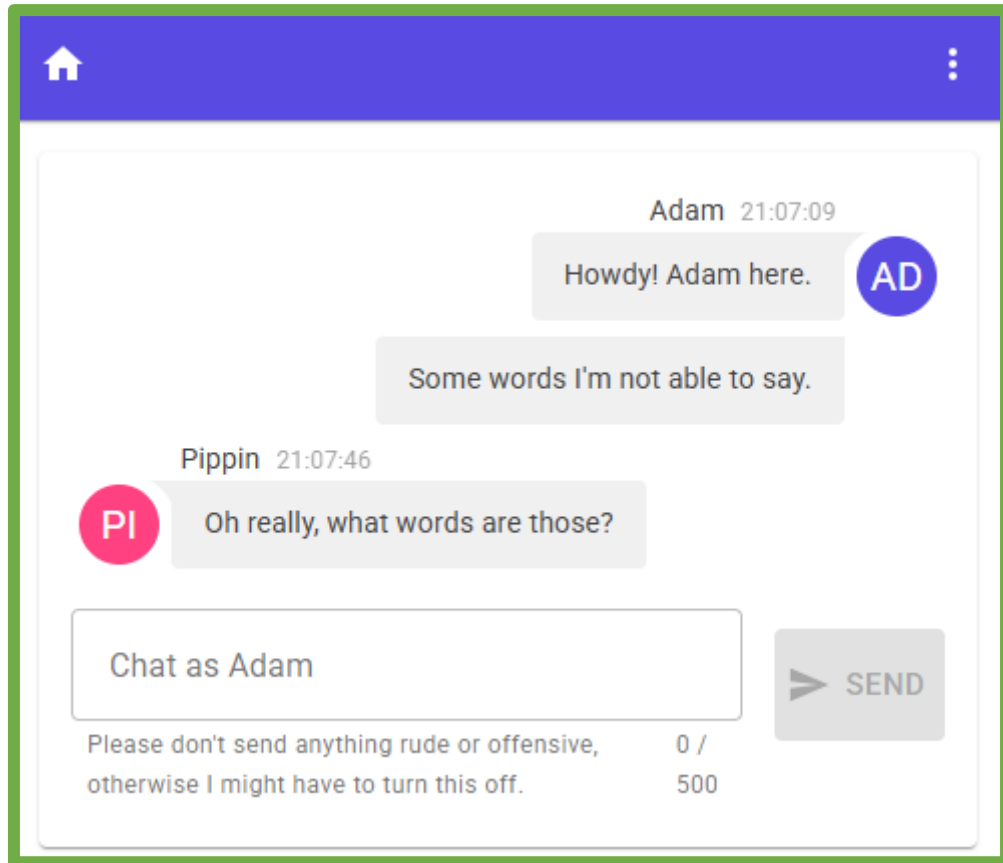
Try unticking this box ☐

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

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

Demo 5

<https://techfast.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

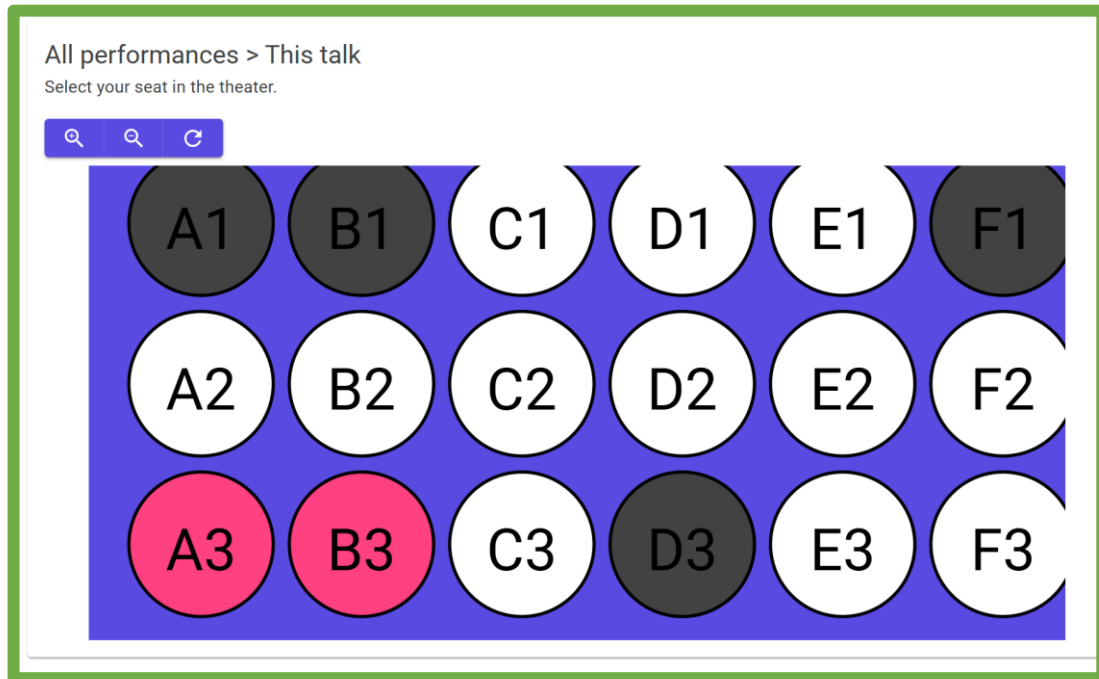
<https://techfast.blakey.family>



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

Demo 7

<https://techfast.blakey.family>



- Features

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

Demo summary

<https://techfast.blakey.family>

- Hosting models
- Data binding
- Entirely WASM apps
- Rich package availability
- Real-time via event handlers & SignalR

In 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>



</adamlakey>

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>
</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]);
    }
}

```