Converting to Server-side Blazor



Gill Cleeren
CTO XPIRIT BELGIUM

@gillcleeren www.snowball.be



Overview



Introducing server-side Blazor

Converting the app to server-side Blazor



Introducing Server-side Blazor



Blazor Hosting Models

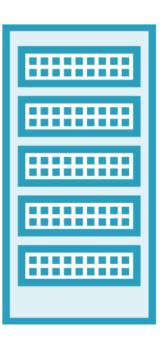
Server-side Client-side



Client-side Blazor Architecture

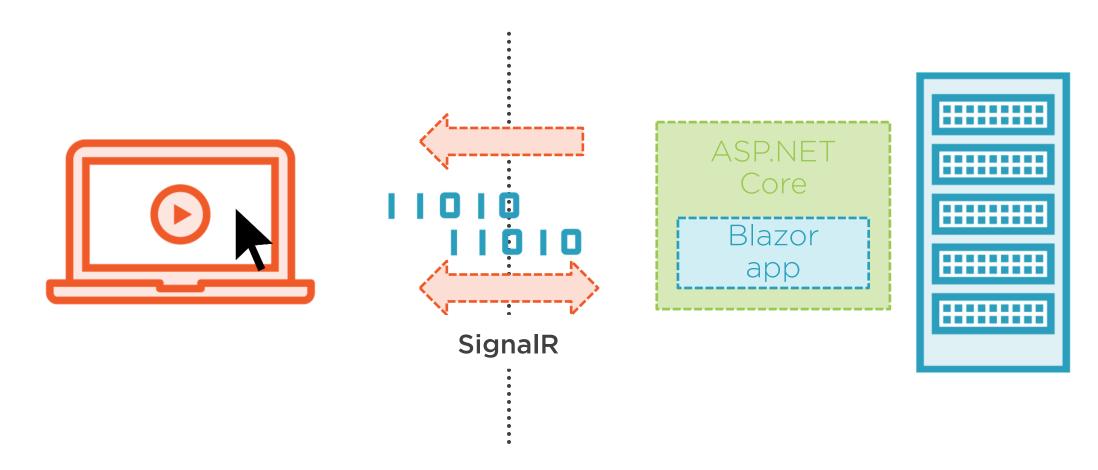




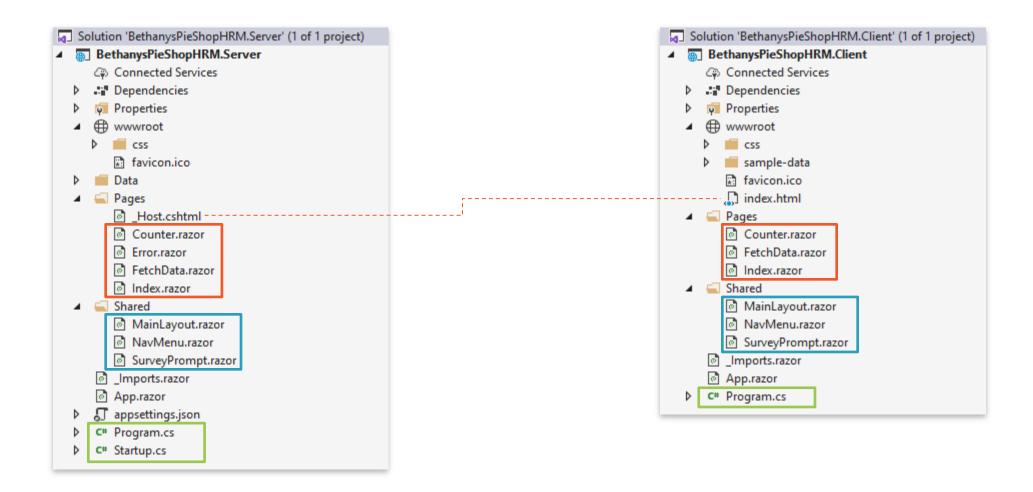




Server-side Blazor Architecture



Comparing Server-side and Client-side Blazor



Invoking Server-side Blazor

Different js File

```
_Host.cshtml
```

```
<html lang="en">
<body>
    <app>
        <component type="typeof(App)" render-mode="ServerPrerendered" />
    </app>
    <script src="_framework/blazor.server.js"></script>
</body>
</html>
```

Choosing for Server-side Blazor



No initial download, all code executes on server



All .NET Core server features become available



Code isn't shared with clients



No offline support



Demo



Exploring File → New Project for server-side Blazor

Looking at the running app with the browser tools



Converting the App to Server-side Blazor



Server-side to Client-side

Client-side Blazor app



Server-side Blazor app





Sharing code

- Ensure your app can be easily switched
- Extract what is different
- Share what is common

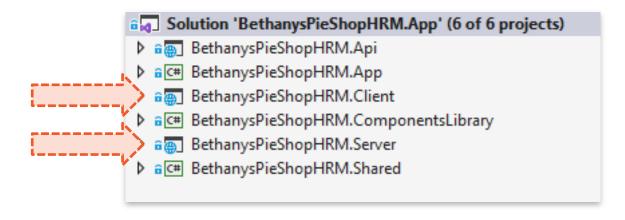


Things to Consider

EF Core REST APIs HttpClient usage



Proposed Approach





Server-side Project

```
■ BethanysPieShopHRM.Server

Connected Services

Dependencies

Properties

Comparises

Co
```



Client-side Project

```
■ BethanysPieShopHRM.Client

C Connected Services

Dependencies

Properties

www.root

c C program.cs
```



▲ a C# BethanysPieShopHRM.App Dependencies ▶ a v Properties ▶ a Components ▲ a □ Pages Da C# EmployeeDetail.cs ▶ a C# EmployeeEdit.cs ▶ a Services ▶ a Shared

App Project

- index.html is removed
- Project is now a Razor Class Library
- All code can be fully reused



Demo



Adding a server-side project

Adding a new client-side project

Converting our original project



Summary

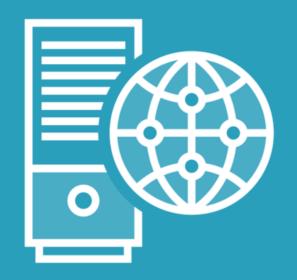


Client-side Blazor projects can easily be converted to server-side by following some patterns

All code is by default possible to share

- It's essentially the same product





Up next:

Deploying the application

