Confidential

GlobalLogic®

Blazor - SPD

GlobalLogic®

About me

Jan Kaczor

jan.kaczor@globallogic.com

Blazor

Blazor is a web UI framework for building single page app using .net and c# instead of JS. Blazor allows to build entire app using.net.

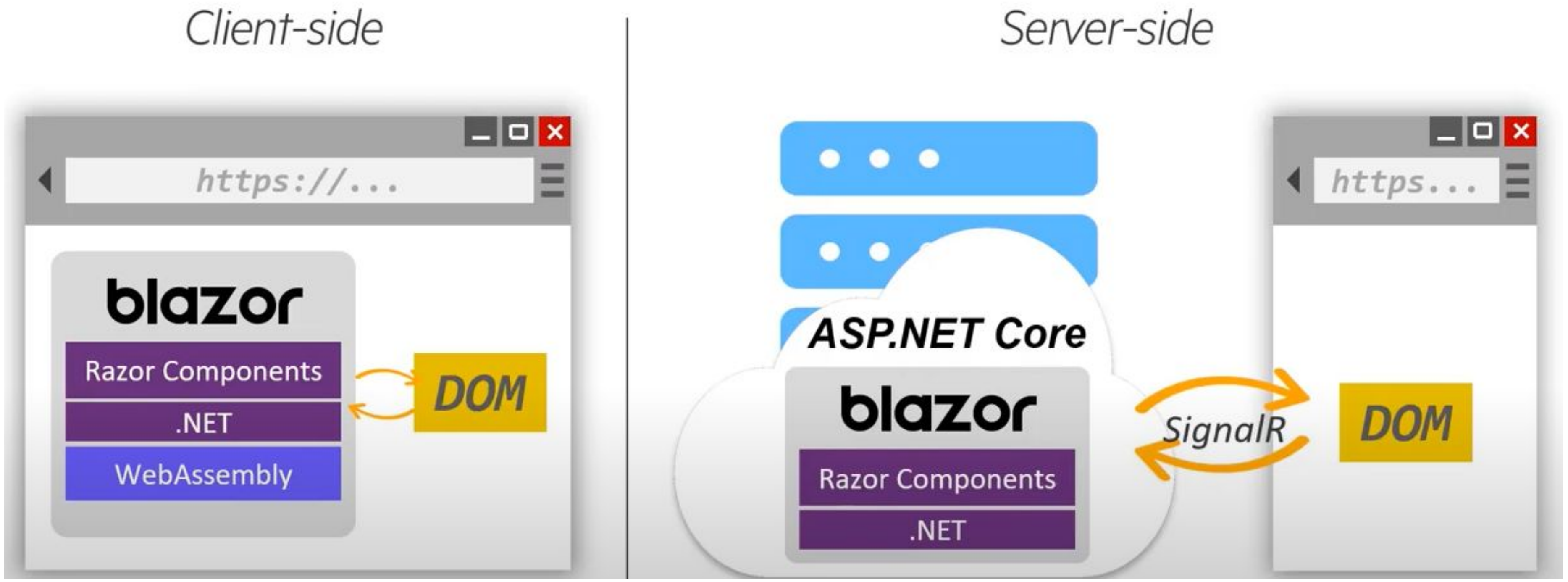
Both client and server code is written in C#, allowing you to share code and libraries.

Blazor uses open web standards without plugins or code transpilation

Huge ecosystem of .net packages, because it is compatible of .net standard

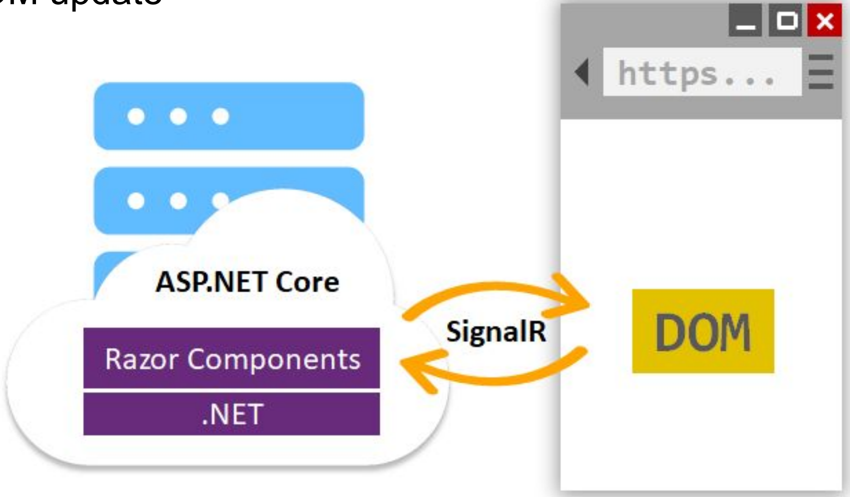
Using JS in web development
C# in the browser

Hosting models



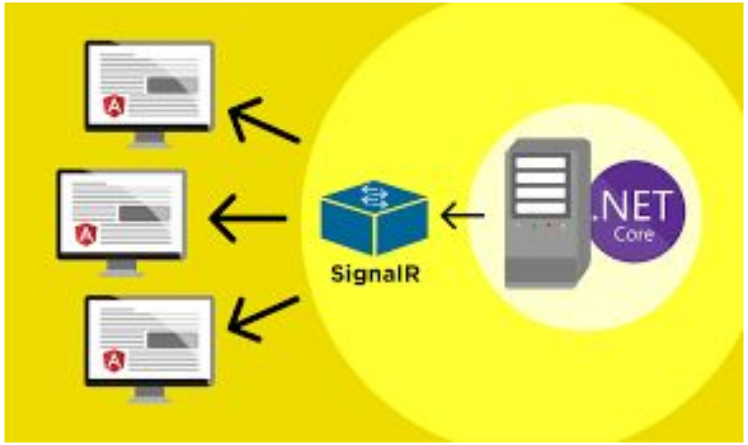
Blazor server-side

- Real time connection between server and browser
- The C# code runs on the server.
- Javascript hooks are used to access the DOM.
- Binary messages are used to pass information between the browser and the server using SignalR.
- If something is changed the server sends back DOM update messages.



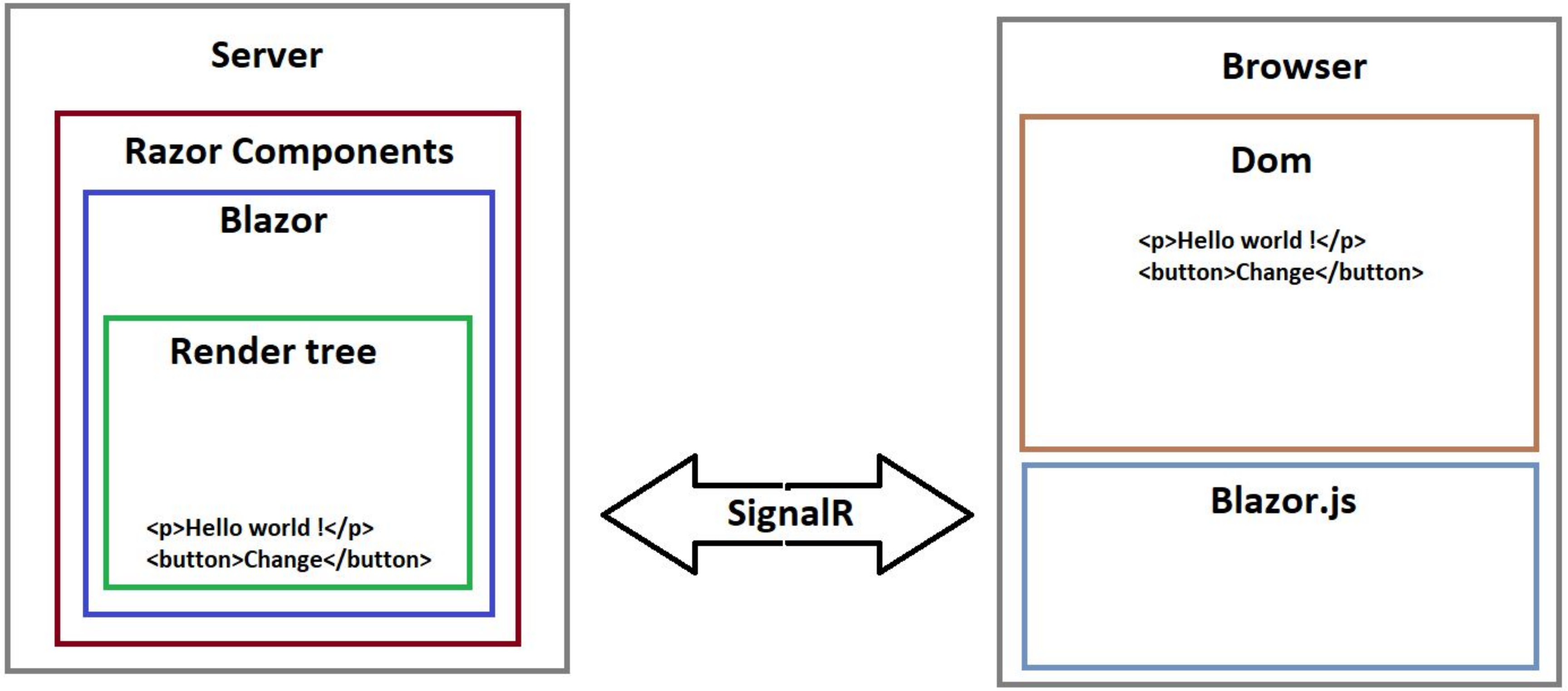
SiganlR

SignalR is open-source library that allows us to create real time web functionality. Real time functionality enables server-side code to send content to clients instantly



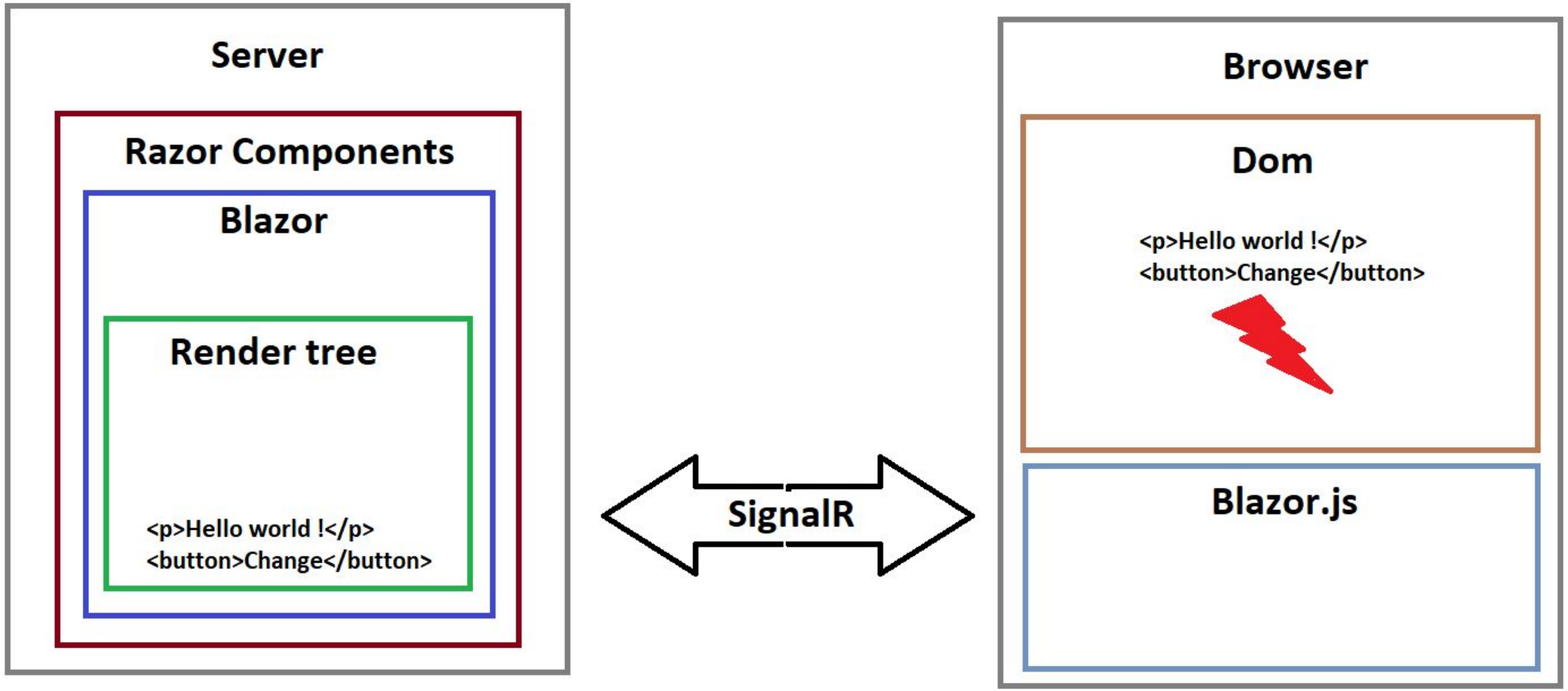
Server-side pros and cons

Pros	**Cons**
Is fully compatible with any .NET libraries and .NET tooling	Does not have performance benefits of the client-side version
Uses exactly the same syntax as the client-side Blazor	.NET server is required
Small size of client-side components	Reduced scalability (SignalR limit)
Works with thin clients. (all browsers)	



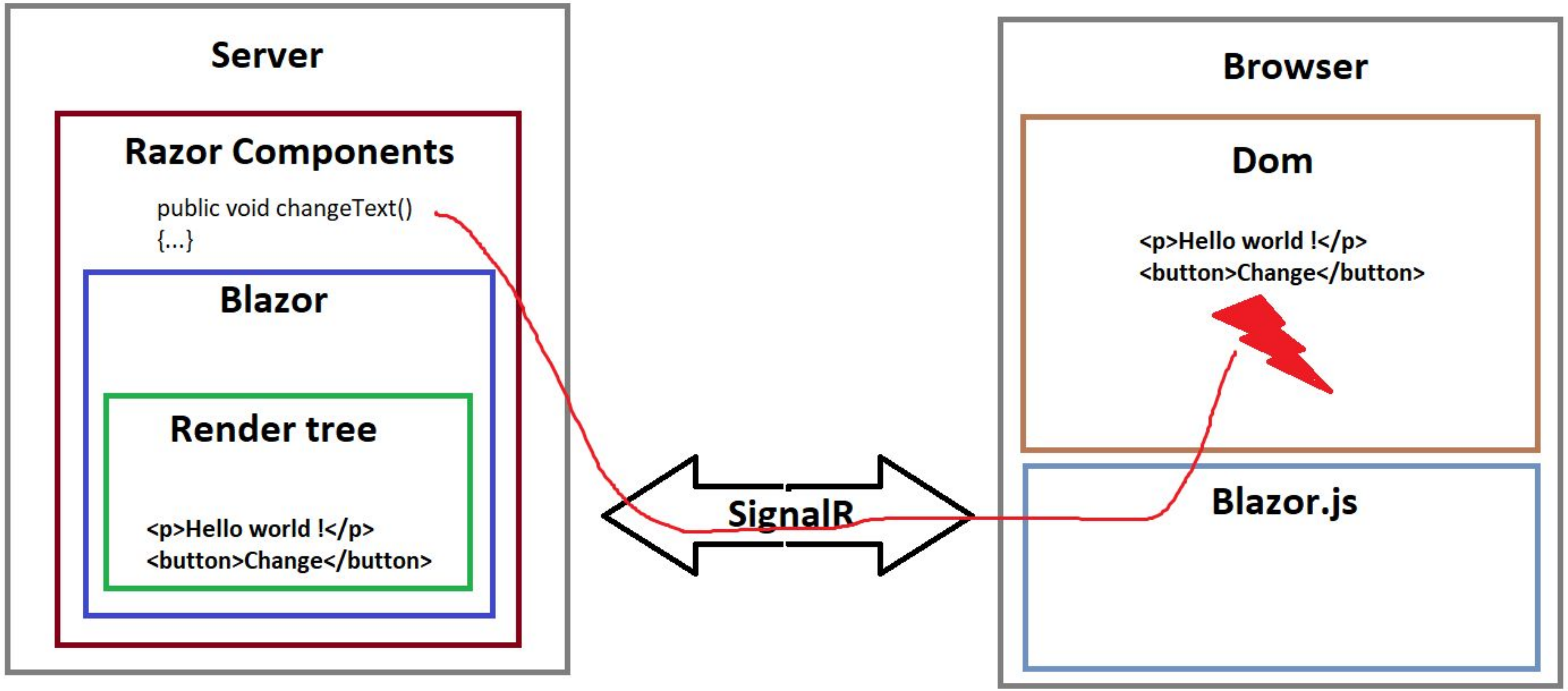
```
<p>Hello world !</p>
<button>Change</button>
```

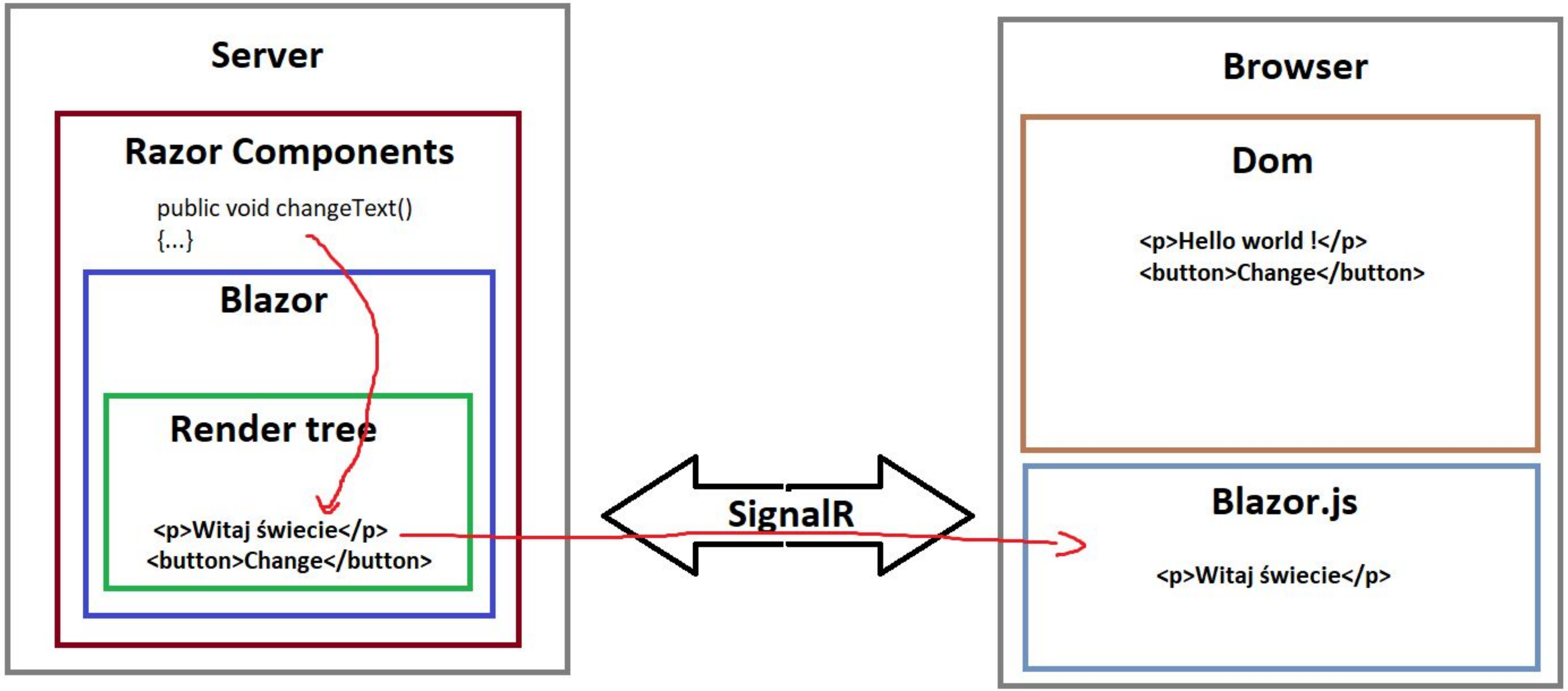
```
<p>Hello world !</p>
<button>Change</button>
```

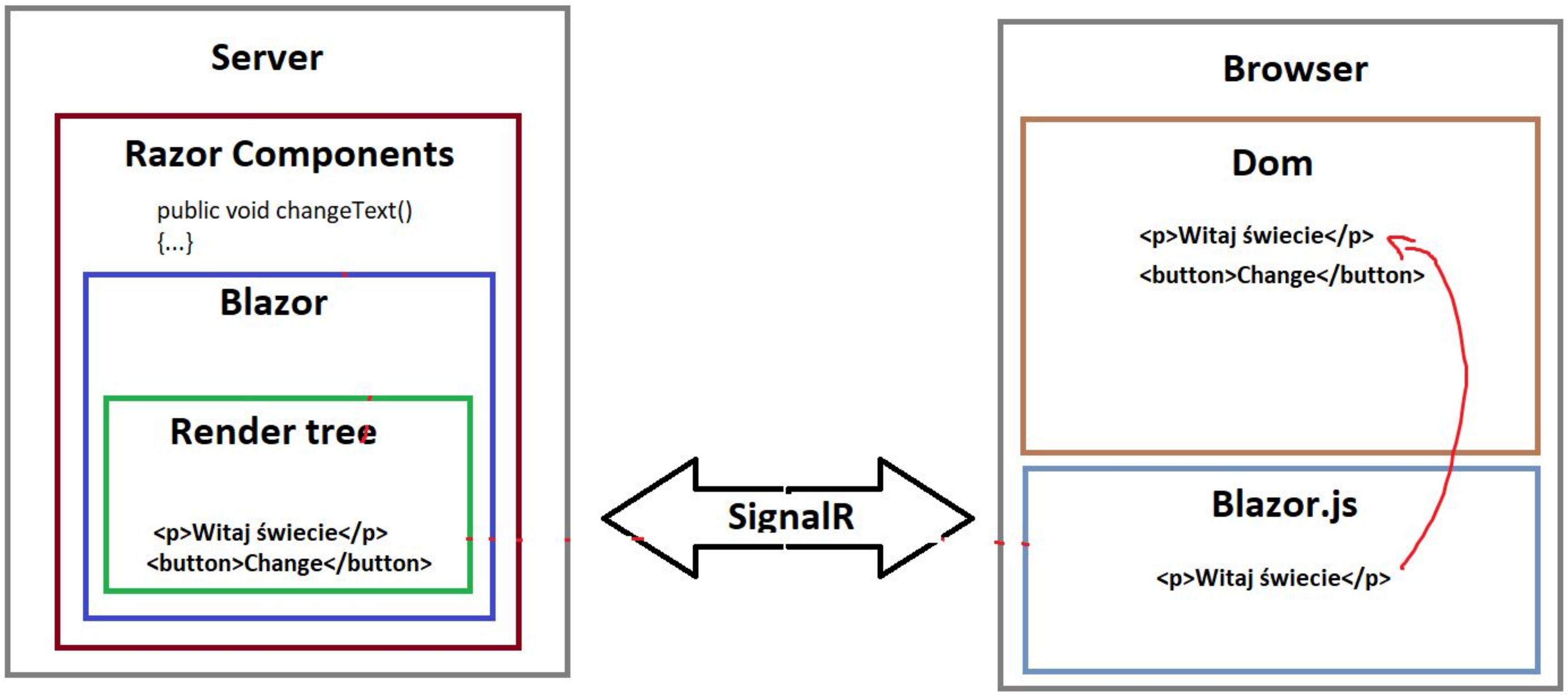


```
<p>Hello world !</p>
<button>Change</button>
```

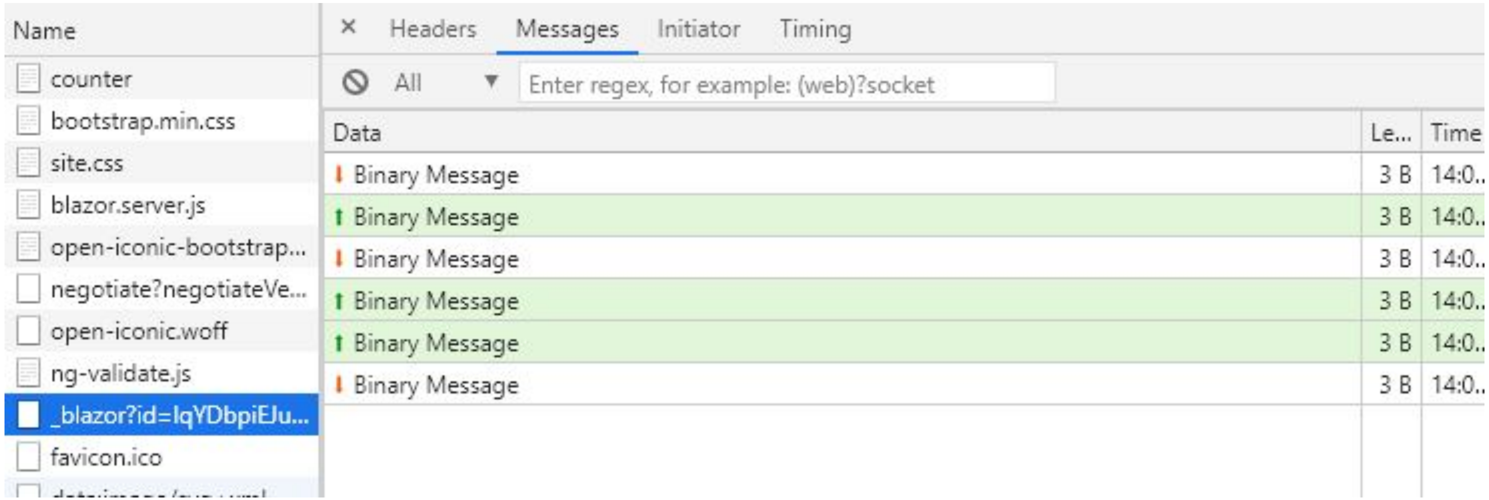
```
<p>Hello world !</p>
<button>Change</button>
```


Server
Razor Components
public void changeText()
{...}
Blazor
Render tree
<p>Hello world !</p>
<button>Change</button>
SignalR
Browser
Dom
<p>Hello world !</p>
<button>Change</button>
Blazor.js


Server
Razor Components
public void changeText()
{...}
Blazor
Render tree
<p>Witaj świecie</p>
<button>Change</button>
SignalR
Browser
Dom
<p>Hello world !</p>
<button>Change</button>
Blazor.js
<p>Witaj świecie</p>


Server
Razor Components
public void changeText()
{...}
Blazor
Render tree
<p>Witaj świecie</p>
<button>Change</button>
SignalR
Browser
Dom
<p>Witaj świecie</p>
<button>Change</button>
Blazor.js
<p>Witaj świecie</p>

In action - demo



Name
- counter
- bootstrap.min.css
- site.css
- blazor.server.js
- open-iconic-bootstrap...
- negotiate?negotiateVe...
- open-iconic.woff
- ng-validate.js
- _blazor?id=IqYDbpiEJu...
- favicon.ico

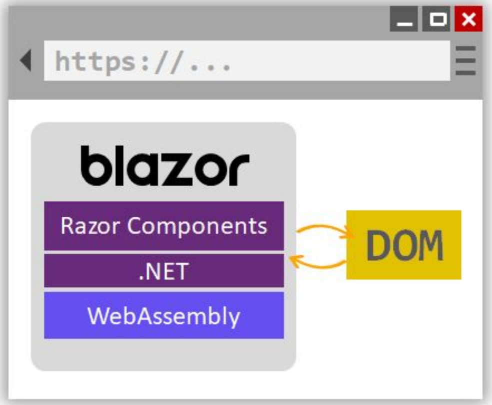
× Headers | Messages | Initiator | Timing

All ▼ Enter regex, for example: (web)?socket

Data	Le...	Time
⬇ Binary Message	3 B	14:0...
⬆ Binary Message	3 B	14:0...
⬇ Binary Message	3 B	14:0...
⬆ Binary Message	3 B	14:0...
⬆ Binary Message	3 B	14:0...
⬇ Binary Message	3 B	14:0...

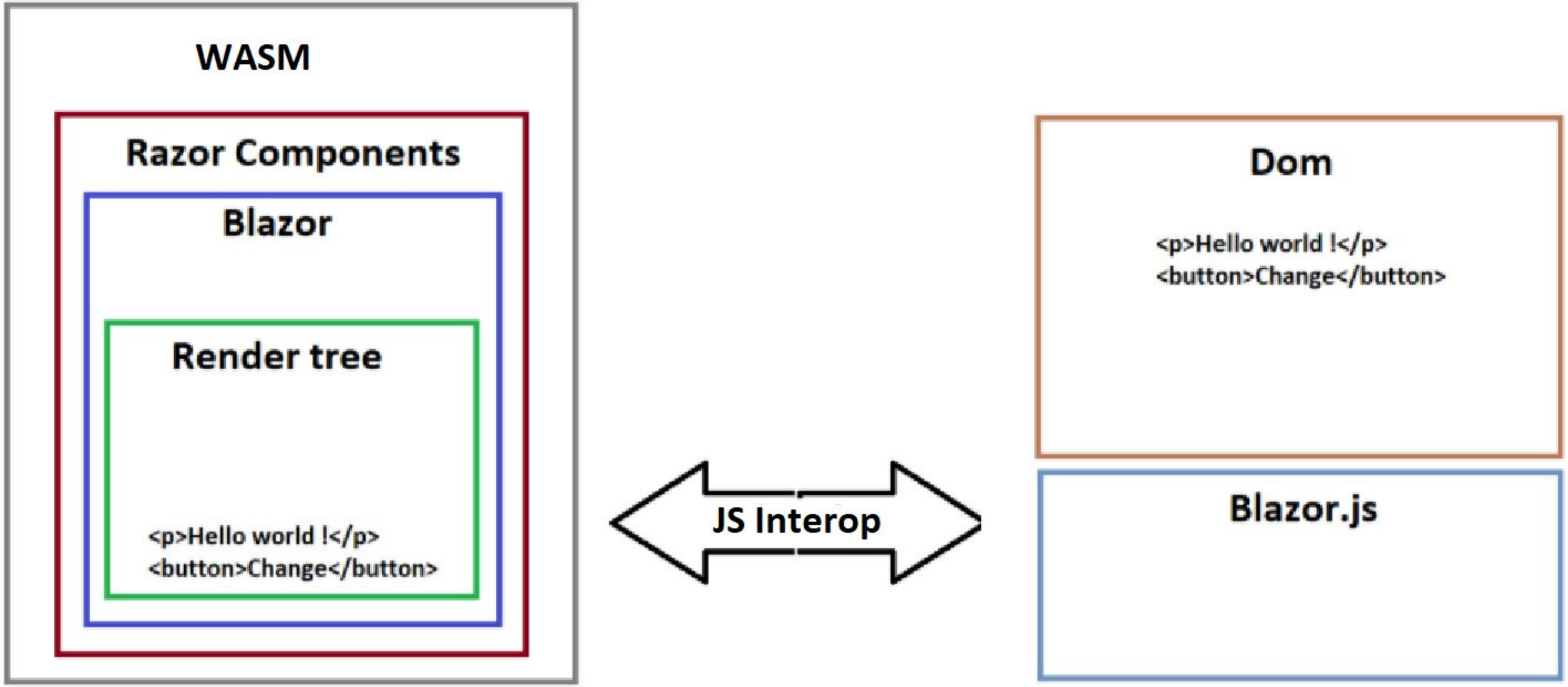
Blazor web assembly (client-side)

- WASM runs in the browser on the client.
- The first request to the WASM application downloads the CLR, Assemblies, JavaScript, CSS (React and Angular work similar).
- It runs in the secure WASM sandbox.
- The Blazor Javascript handler accesses the DOM (Document Object Model).
- c# code files and Razor pages are compiled into .Net Assemblies



Client-side pros and cons

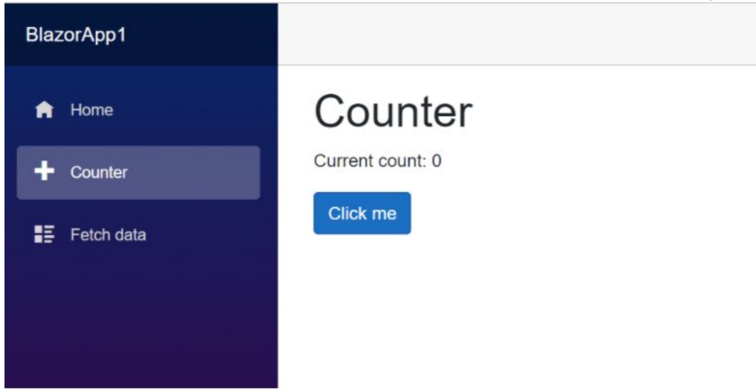
Pros	**Cons**
Running .NET code directly in browser	Requires the whole runtime to be shipped
Faster than JavaScript, thanks to WebAssembly	Doesn’t work with thin clients
The same validation code can be applied on the client and on the server	Limited debugging capability and .NET tooling
Works offline	



```
<p>Hello world !</p>
<button>Change</button>
```

```
<p>Hello world !</p>
<button>Change</button>
```

Blazor counter in details



BlazorApp1

Home

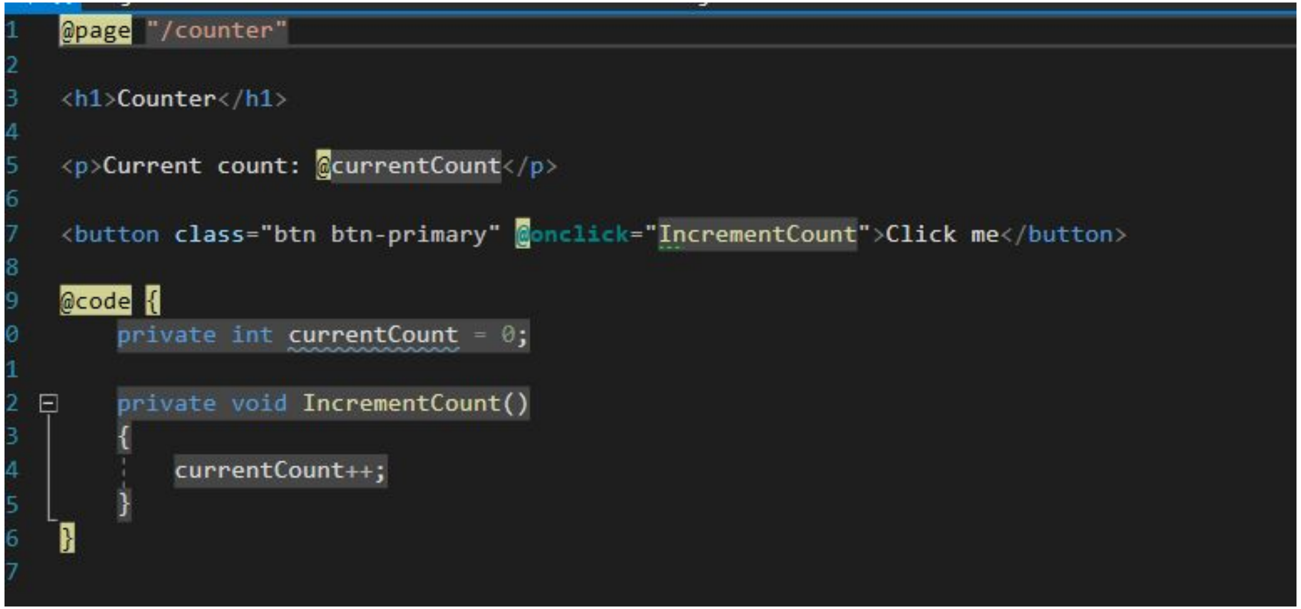
Counter

Fetch data

Counter

Current count: 0

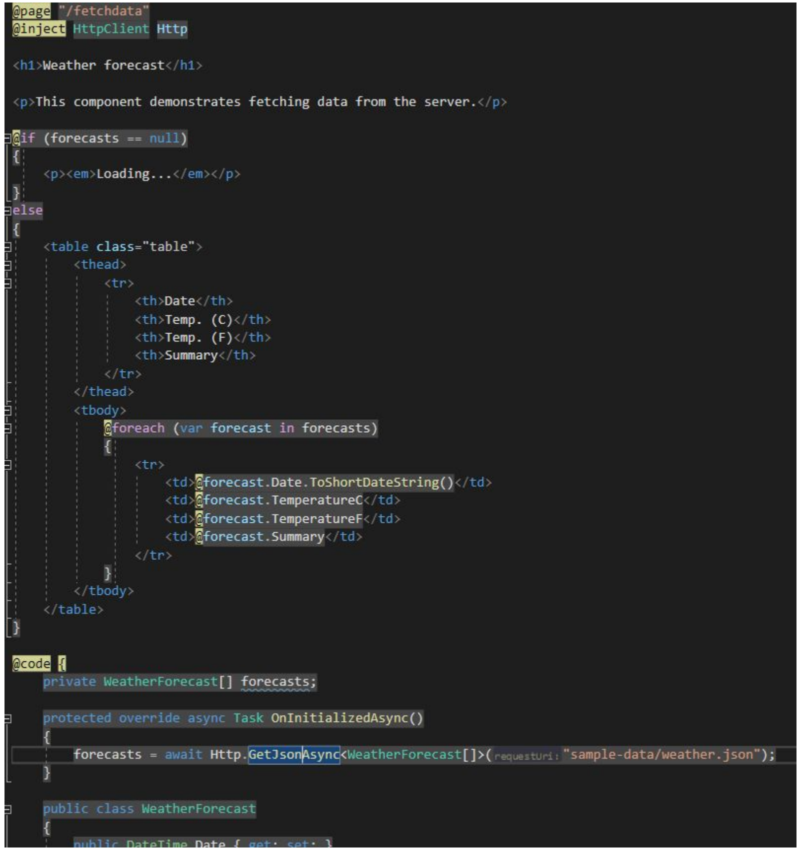
Click me



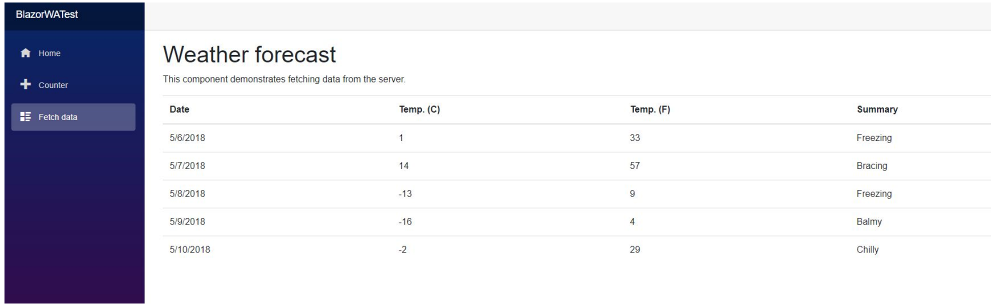
```
@page "/counter"

<h1>Counter</h1>

<p>Current count: @currentCount</p>

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

@code {
    private int currentCount = 0;

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

GlobalLogic®

Fetch data in details



```
@page "/fetchdata"
@inject HttpClient Http

<h1>Weather forecast</h1>

<p>This component demonstrates fetching data from the server.</p>

@if (forecasts == null)
{
    <p><em>Loading...</em></p>
}
else
{
    <table class="table">
        <thead>
            <tr>
                <th>Date</th>
                <th>Temp. (C)</th>
                <th>Temp. (F)</th>
                <th>Summary</th>
            </tr>
        </thead>
        <tbody>
            @foreach (var forecast in forecasts)
            {
                <tr>
                    <td>@forecast.Date.ToShortDateString()</td>
                    <td>@forecast.TemperatureC</td>
                    <td>@forecast.TemperatureF</td>
                    <td>@forecast.Summary</td>
                </tr>
            }
        </tbody>
    </table>
}

@code {
    private WeatherForecast[] forecasts;

    protected override async Task OnInitializedAsync()
    {
        forecasts = await Http.GetJsonAsync<WeatherForecast[]>(requestUri: "sample-data/weather.json");
    }

    public class WeatherForecast
    {
        public DateTime Date { get; set; }
```



Weather forecast

This component demonstrates fetching data from the server.

Date	Temp. (C)	Temp. (F)	Summary
5/6/2018	1	33	Freezing
5/7/2018	14	57	Bracing
5/8/2018	-13	9	Freezing
5/9/2018	-16	4	Balmy
5/10/2018	-2	29	Chilly

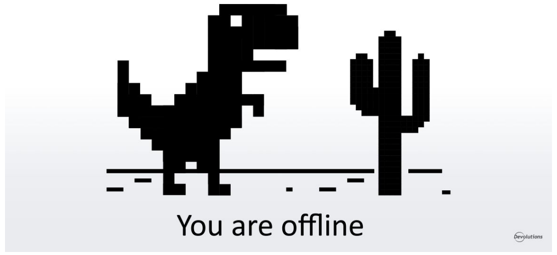
Ready to use components

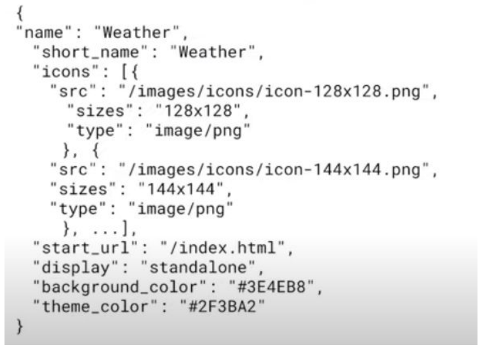
- Telerik
- DevExpress
- Syncfusion
- Radzen
- Infragistics
- GrapeCity
- MatBlazor
- …..

https://www.matblazor.com/
https://www.telerik.com/blazor-ui

PWA- Progressive web app

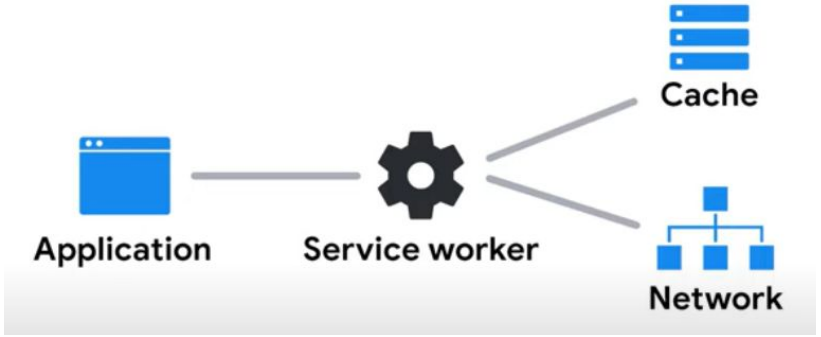
Web apps that behave like native apps using modern web standards.





```
{
"name": "Weather",
  "short_name": "Weather",
  "icons": [{
    "src": "/images/icons/icon-128x128.png",
      "sizes": "128x128",
      "type": "image/png"
     }, {
    "src": "/images/icons/icon-144x144.png",
    "sizes": "144x144",
    "type": "image/png"
     }, ...],
  "start_url": "/index.html",
  "display": "standalone",
  "background_color": "#3E4EB8",
  "theme_color": "#2F3BA2"
}
```

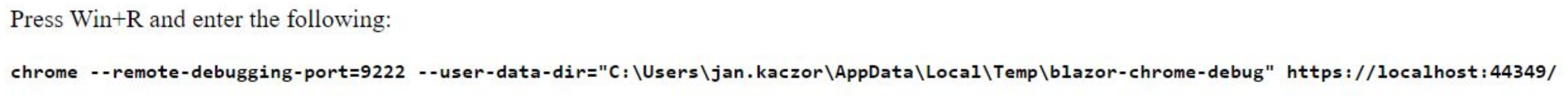
Look



Layer

Blazor client-side debugging for chrome

1. Build app in debug mode
2. Press Shift + Alt + D
3. Copy and execute command given from chrome. New chrome instance should execute



Press Win+R and enter the following:

```
chrome --remote-debugging-port=9222 --user-data-dir="C:\Users\jan.kaczor\AppData\Local\Temp\blazor-chrome-debug" https://localhost:44349/
```

4. Press once again Shift + Alt + D, new tab should be opened.
5. Select a page and put breakpoint in code.
6. From now u can debug page.

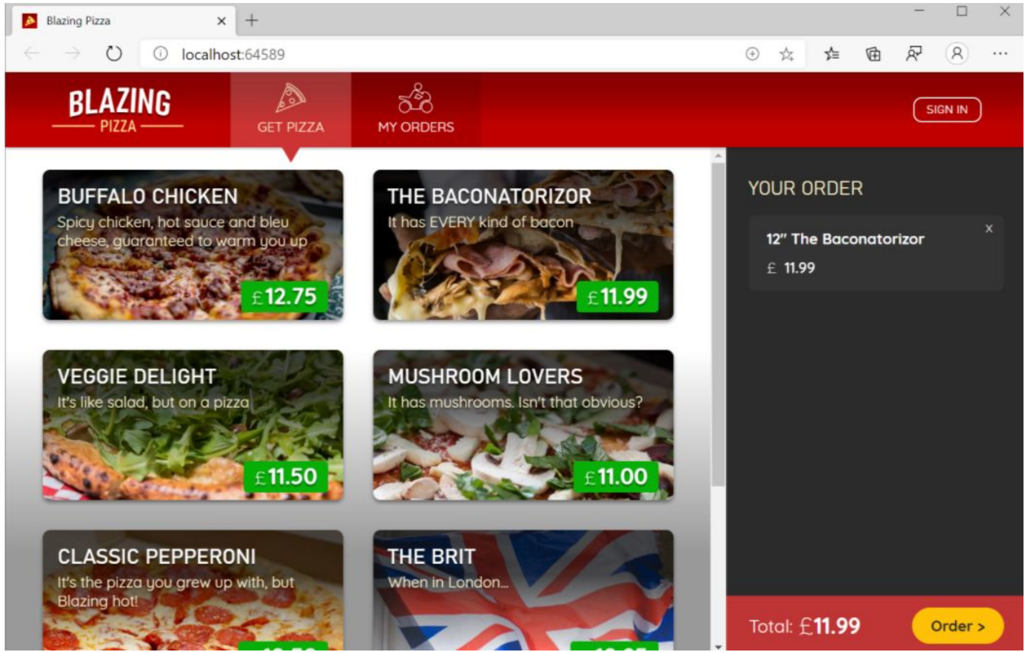
New for Blazor in .NET 5

- CSS Isolation
- Auto-refresh
- Packages auto-add their own CSS/JS
- Better debugging
- InputFile support
- Virtualization
- Increased performance
- …. and more

GlobalLogic®

Car checker and pizza demo.

https://github.com/dotnet-presentations/blazor-workshop
https://github.com/SteveSandersonMS/CarChecker



BUFFALO CHICKEN
Spicy chicken, hot sauce and bleu cheese, guaranteed to warm you up
£12.75

THE BACONATORIZOR
It has EVERY kind of bacon
£11.99

VEGGIE DELIGHT
It's like salad, but on a pizza
£11.50

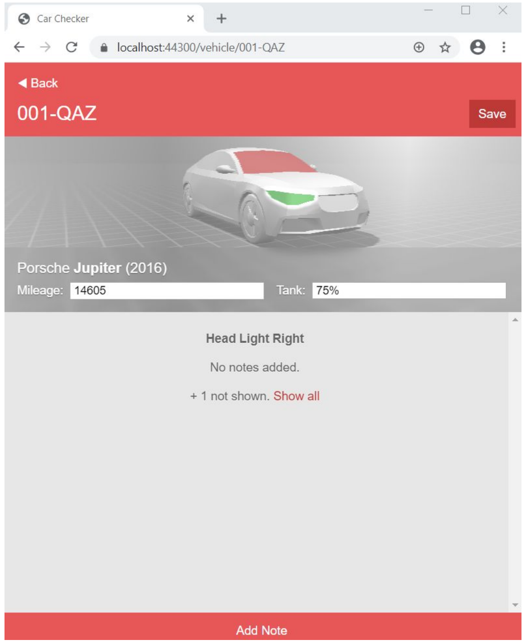
MUSHROOM LOVERS
It has mushrooms. Isn't that obvious?
£11.00

CLASSIC PEPPERONI
It's the pizza you grew up with, but Blazing hot!

THE BRIT
When in London...

12" The Baconatorizor
£ 11.99

Total: £11.99

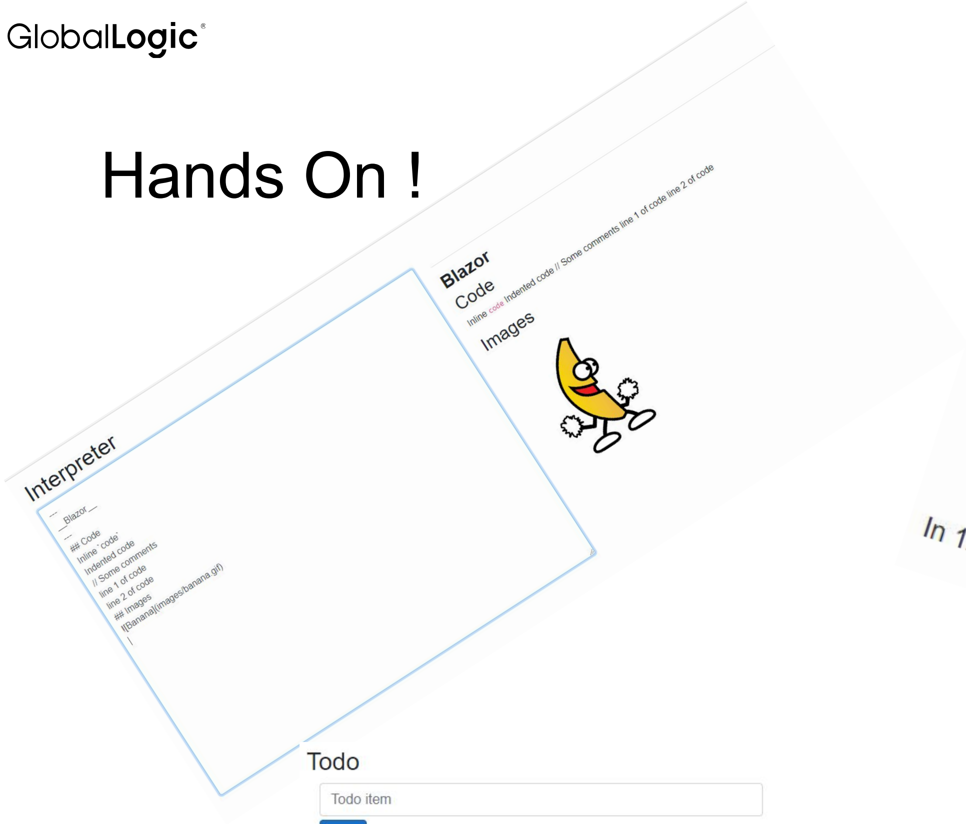


001-QAZ

Porsche **Jupiter** (2016)
Mileage: 14605 Tank: 75%

Head Light Right
No notes added.
+ 1 not shown. Show all

Hands On !



Interpreter

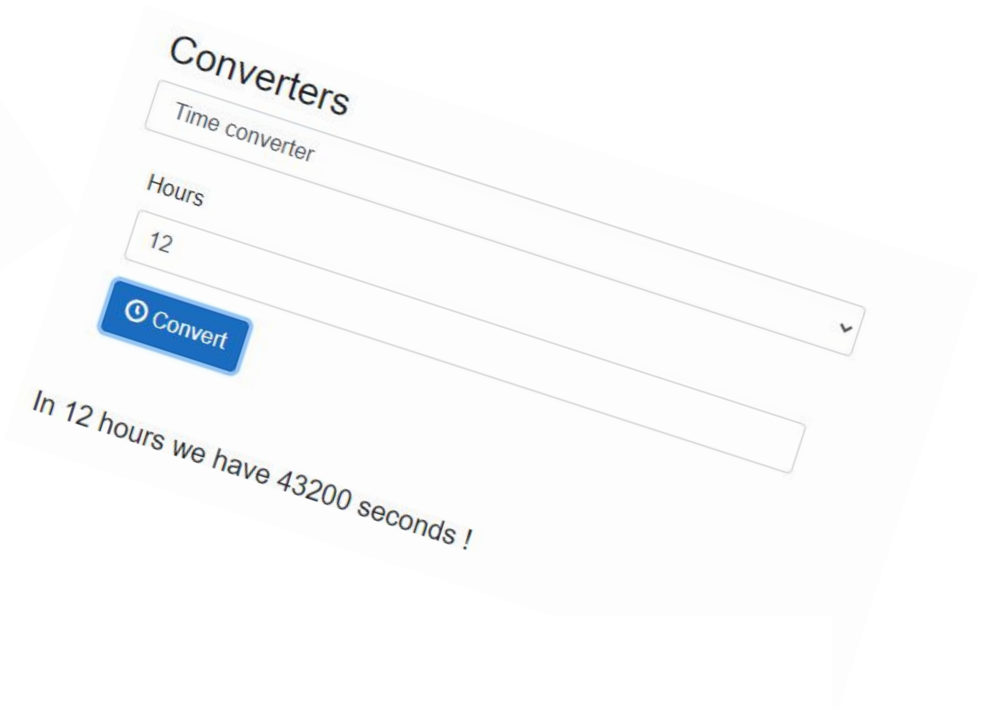
```
--
__Blazor__
--
## Code
Inline `code`
Indented code
// Some comments
line 1 of code
line 2 of code
## Images
![Banana](images/banana.gif)
```

Blazor

Code

Inline `code` Indented code // Some comments line 1 of code line 2 of code

Images



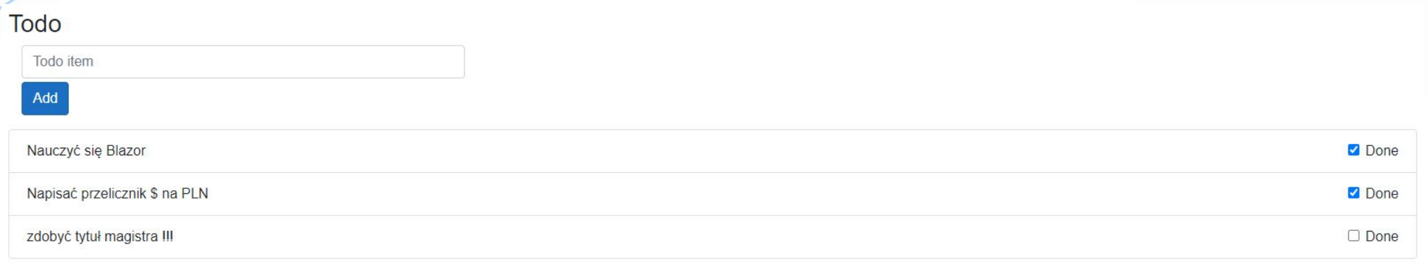
Converters

Time converter

Hours

12

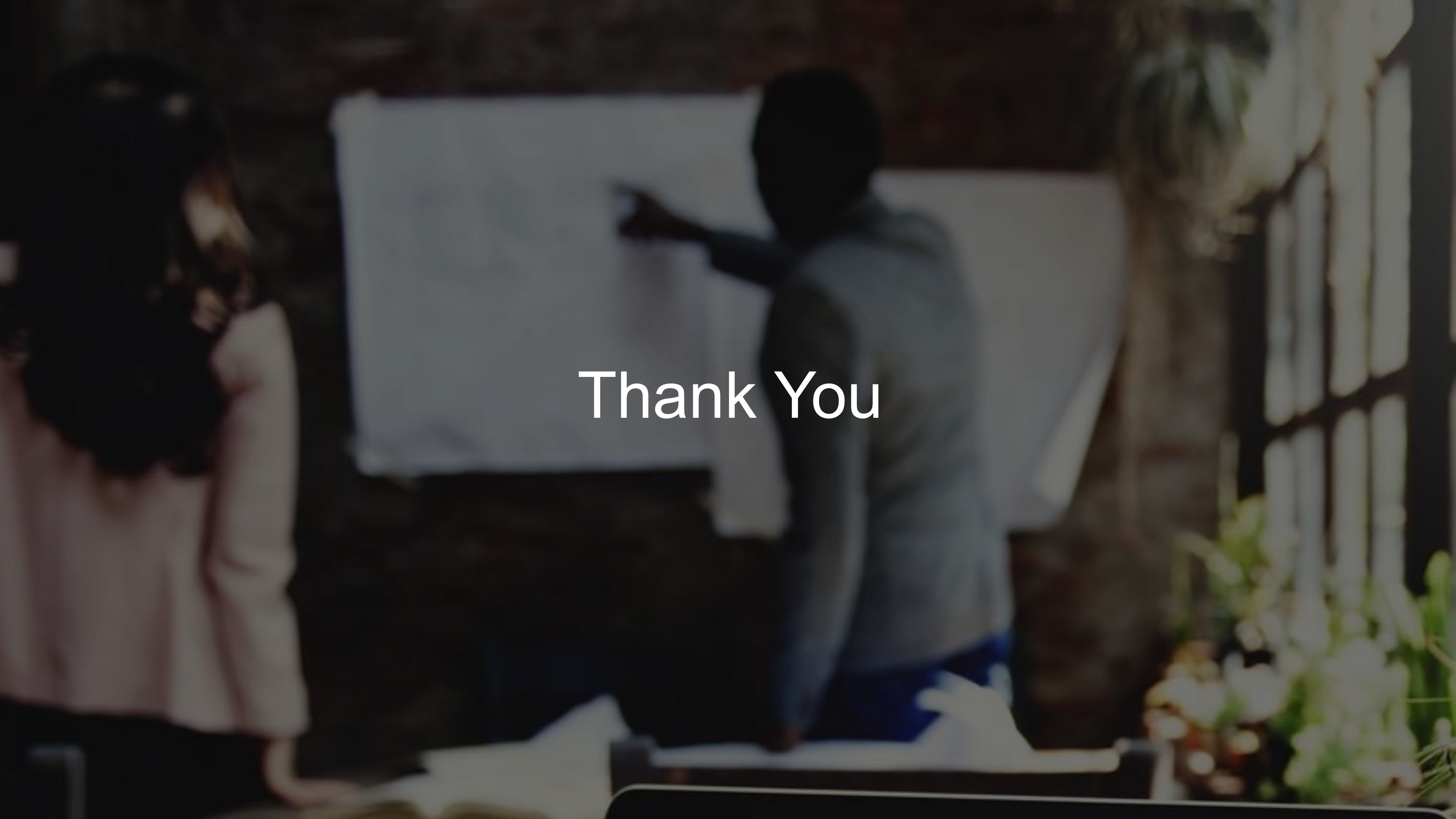
In 12 hours we have 43200 seconds !



Todo

Todo item

Nauczyć się Blazor	☑ Done
Napisać przelicznik $ na PLN	☑ Done
zdobyć tytuł magistra !!!	☐ Done



Thank You