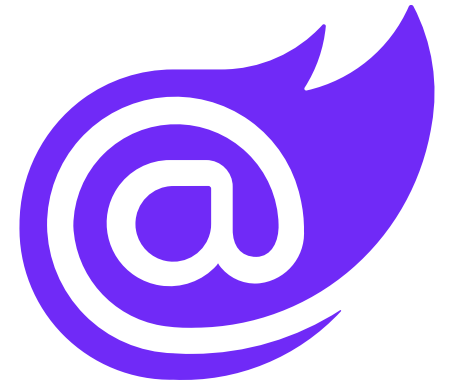


# Blazor

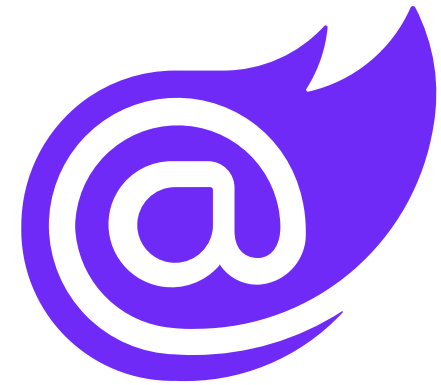
*new way to build client web apps*

Thien Nguyen



# Agenda

1. What is WebAssembly
2. What is Blazor
3. Blazor hosting models
4. PROS and CONS of each Blazor hosting models
5. Demo



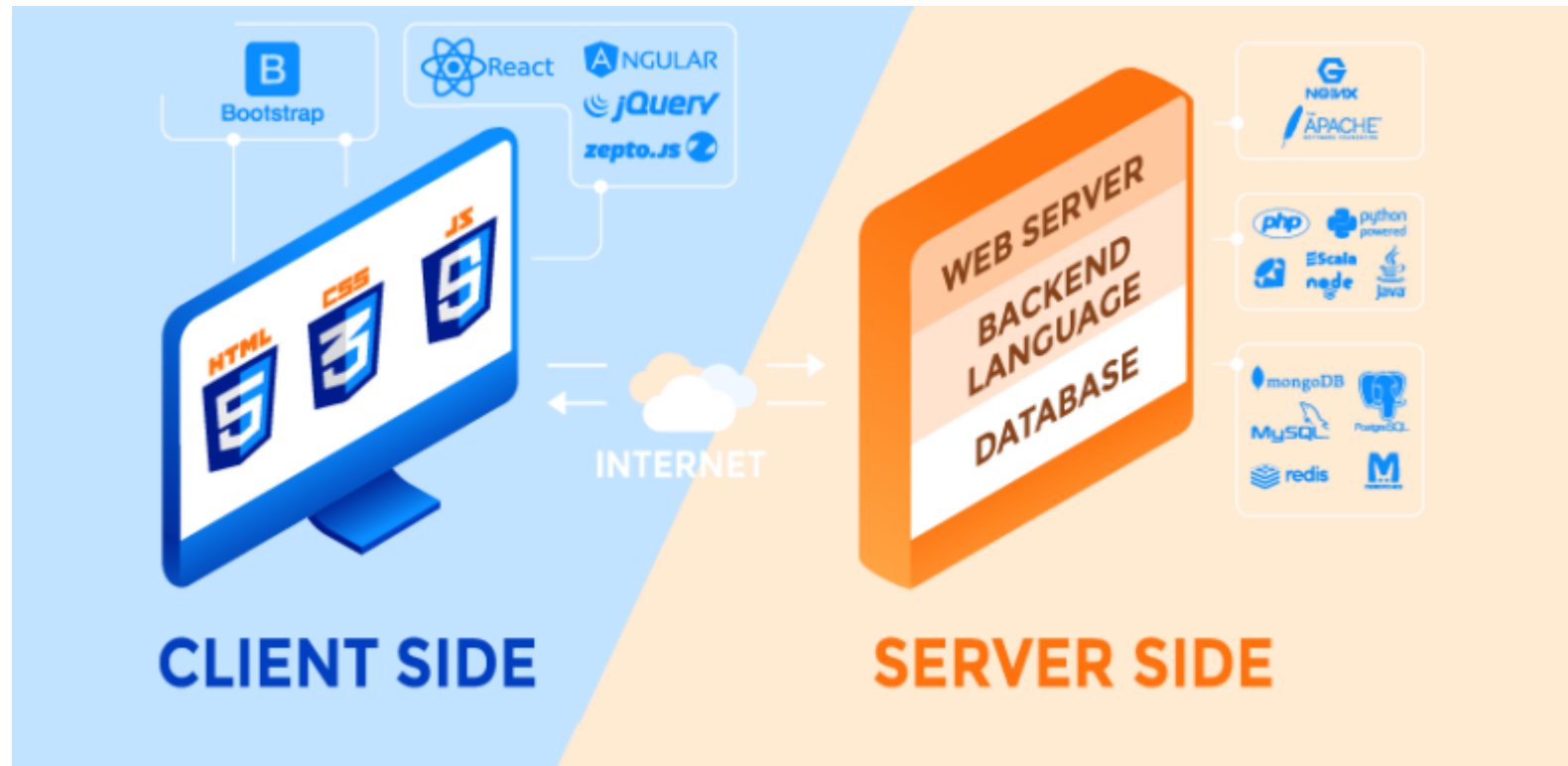
# What is WebAssembly

# 1. What is WebAssembly



# 1. What is WebAssembly

Review about Server-Side and Client-Side Programming Languages



# 1. What is WebAssembly

- ❑ WebAssembly (abbreviated *Wasm*) is a binary instruction format for a stack-based virtual machine
- ❑ Wasm is designed as a portable compilation target for programming languages, enabling deployment on the web for client and server applications.
- ❑ Wasm became the official “**fourth language for web**” on 5th December 2019, i.e. after HTML, CSS & JavaScript
- ❑ Can be written by C/C++, C#, Python,...

<https://webassembly.org/>

# 1. What is WebAssembly

## WASM format



The screenshot shows a web-based editor for WebAssembly. The left pane, titled 'Wat', displays the textual format code. The right pane, titled 'Firefox x86 Assembly', displays the corresponding assembly code. Two blue arrows point from the labels below to the respective panes.

```
Wat
1 (module
2   (table 0 anyfunc)
3   (memory $0 1)
4   (export "memory" (memory $0))
5   (export "_Z12isEvenNumberi" (func
6     $ _Z12isEvenNumberi))
7   (func $ _Z12isEvenNumberi (; 0 ;) (param $0
8     i32) (result i32)
9     (i32.and
10      (i32.xor
11       (get_local $0)
12       (i32.const -1)
13      )
14      (i32.const 1)
15     )
16  )
17 )

Firefox x86 Assembly
wasm-function[0]:
  sub rsp, 8
  mov ecx, edi
  mov eax, ecx
  xor eax, 0xffffffff
  and eax, 1
  nop
  add rsp, 8
  ret
```

textual format  
.wat

binary format  
.wasm

# 1. What is WebAssembly

## **WASM format**

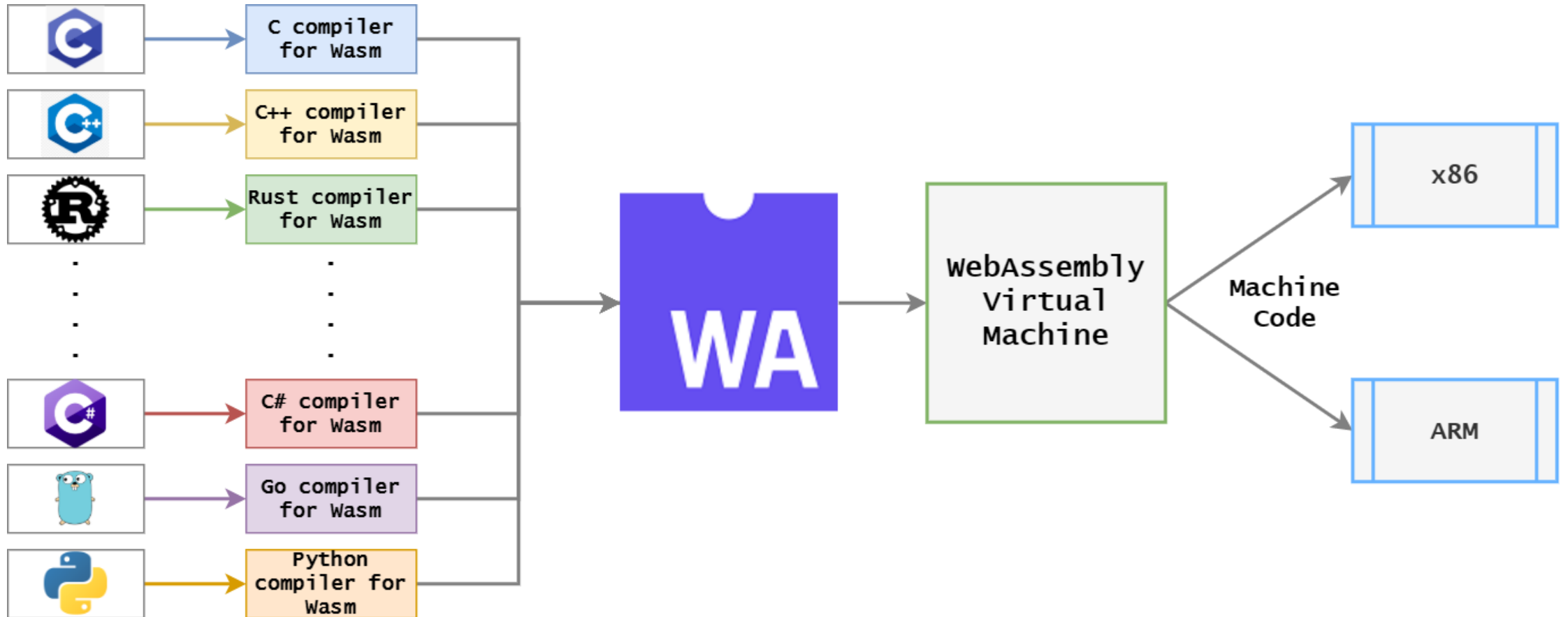
me\* after reading the WASM formats the first time





# 1. What is WebAssembly

## WebAssembly Work



# 1. What is WebAssembly

**Why would I use WebAssembly?**

**How it should be**



# 1. What is WebAssembly

## **Why would I use WebAssembly?**

- ☐ Efficient and fast
- ☐ Safe
- ☐ Compile a WebAssembly module from many languages: C/C++, Rust, C#, F#, Go, Kotlin, Swift,...
- ☐ Can be executed on web, stand-alone

# 1. What is WebAssembly

## Besides, WebAssembly not-so-great

- ❑ Many features under development or not ready yet
- ❑ IE and older browsers don't support wasm
- ❑ By design, code needed to be compiled before distribution. Therefore, “***fix JS directly on CDN***” kind of approaching doesn't supported

# What is Blazor

## 2. What is Blazor

- ❑ Release in 2018
- ❑ A client-side library that use .NET on WebAssembly
- ❑ Support SPAs written in C# with Razor template

<https://docs.microsoft.com/en-us/aspnet/core/blazor/?view=aspnetcore-6.0>

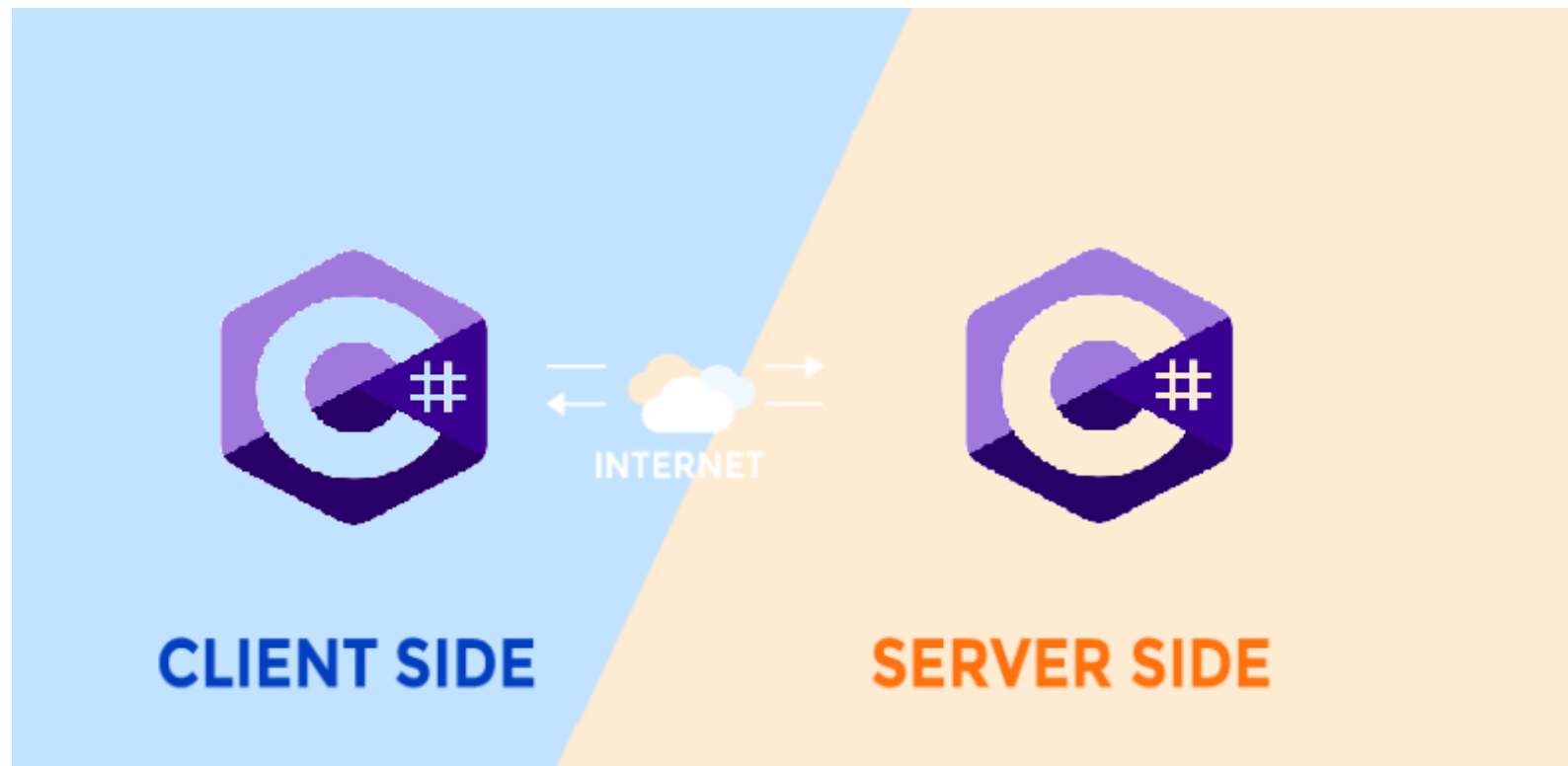
## 2. What is Blazor

- ❑ Not like Silverlight
- ❑ Can run without plugin
- ❑ Share libraries between client and server



## 2. What is Blazor

Server-Side and Client-Side Programming Languages

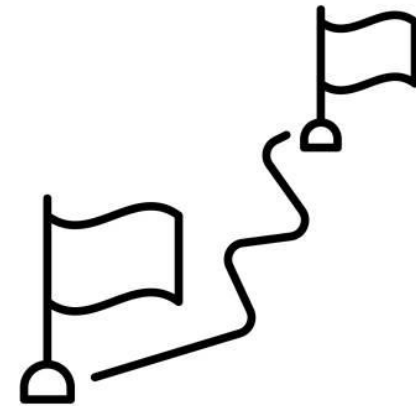




## 2. What is Blazor

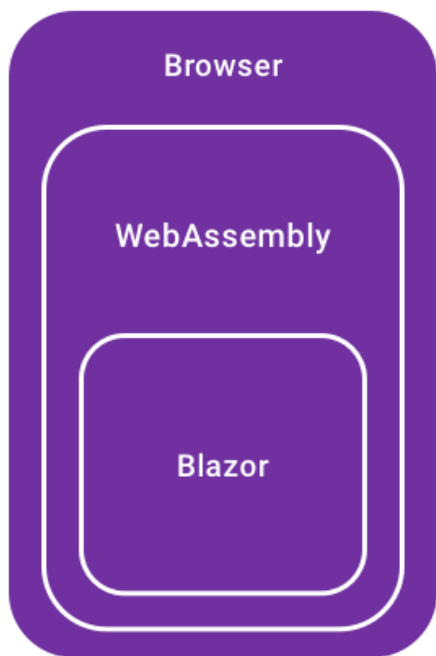
### Milestones

- 2017 : Web Assembly
- 2017 : Blazor 1<sup>st</sup> announced
- 2019 : Blazor 1<sup>st</sup> preview
- 2019 : Blazor Server
- 2019 : Blazor Web Assembly
- 2021 : LTS for Blazor
- 2022 : .NET 7



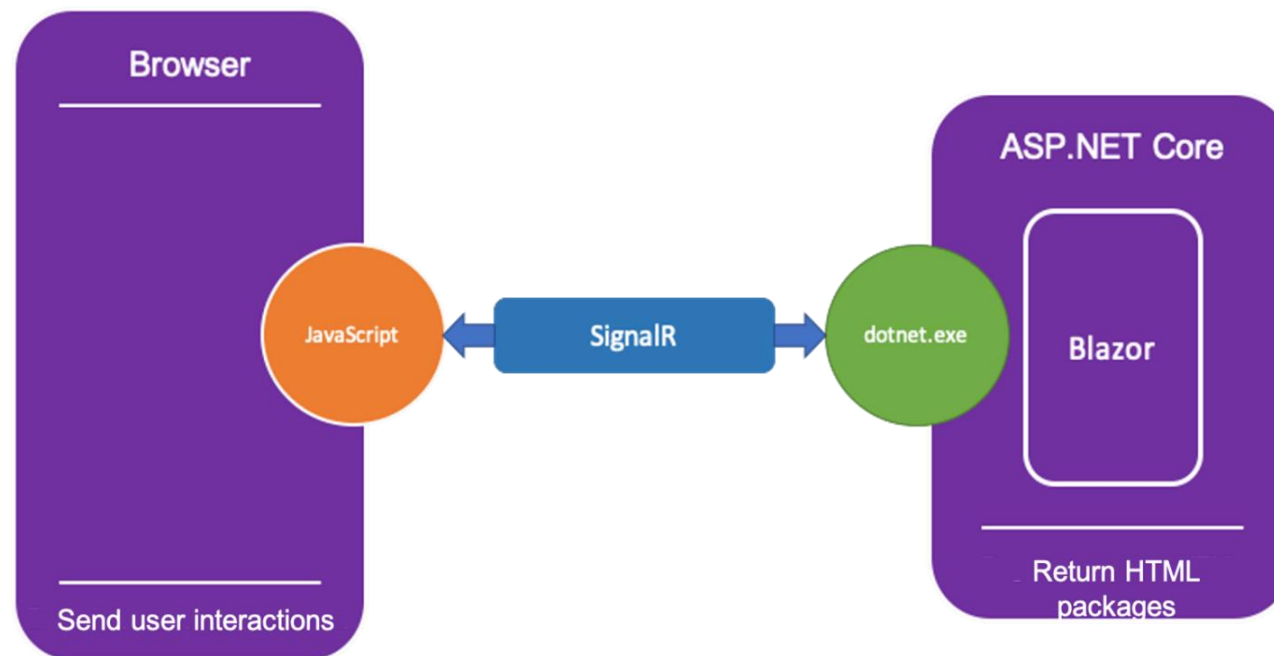
# Blazor hosting models

### 3. Blazor hosting models



#### Blazor WebAssembly

- ❑ Client-side rendering
- ❑ Incremental DOM



#### Blazor Server

- ❑ Server-side logic
- ❑ DOM updated by SignalR

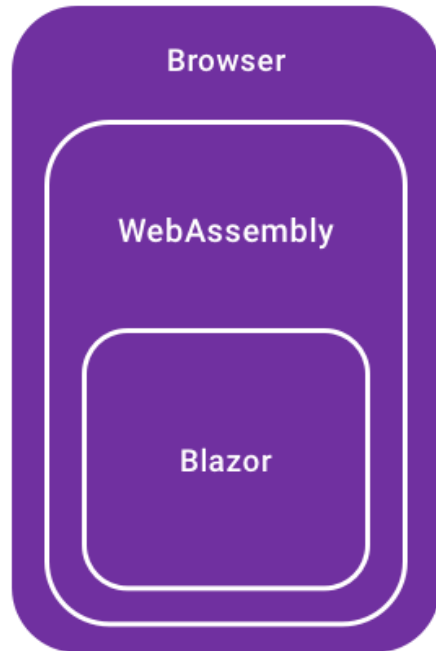


PROS and CONS

Blazor WebAssembly and Blazor Server

## 4. PROS and CONS Blazor WebAssembly and Blazor Server

### Blazor WebAssembly



#### PROS

- ❑ .NET code for browser
- ❑ Efficient and fast
- ❑ NO server-side dependencies
- ❑ Progressive Web App (PWA)

#### CONS

- ❑ Need download resources before 1<sup>st</sup> running (~700 KB)
- ❑ Old browser cannot be used
- ❑ Need times to developing tools for developers

# 4. PROS and CONS Blazor WebAssembly and Blazor Server

## Blazor WebAssembly

BlazorWebAssembly

[Home](#)

[Counter](#)

[Fetch data](#)

About

Hello, world!

Welcome to your new app.

How is Blazor working for you? Please take our [brief survey](#) and tell us what you think.

Web SQL

Cookies

Trust Tokens

Interest Groups

Cache

Cache Storage

blazor-resources/- - https://localhost:5001

Back/forward cache

Background Services

Background Fetch

Background Sync

Notifications

Payment Handler

Periodic Background Sync

Push Messaging

Reporting API

Frames

top

Elements

Console

Sources

Network

Performance

Memory

Application

Security

Lighthouse

Recorder

Performance insights

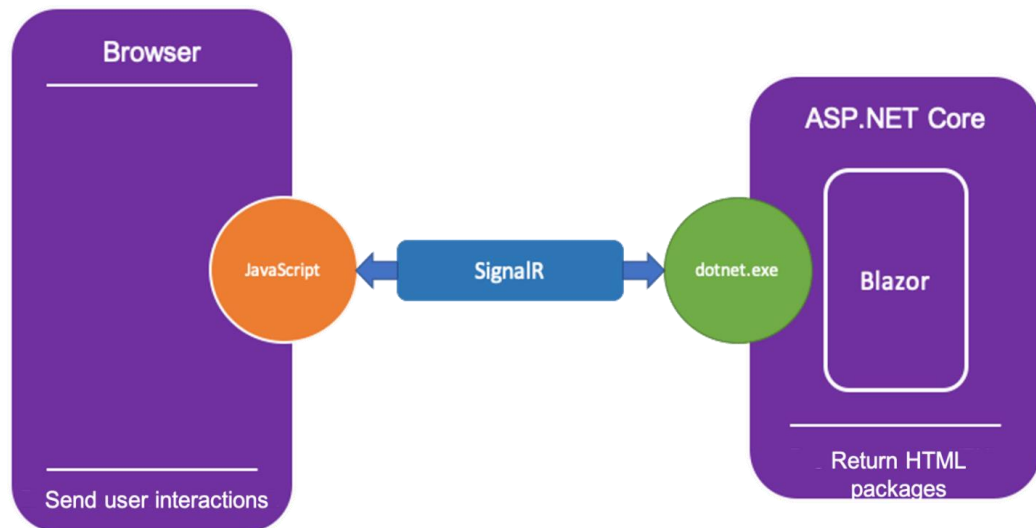
Filter by Path

#	Name	Response-Ty..	Content-Type	Content-Length	Time Cached	Vary Header
0	framework/BlazorWebAssembly.dll.sha256-T8q/TL22jtNizT/4jTZOynhxORmsZFDKrx+OR4WWZl=	default	application/octet-str...	21,504	6/20/2022, 4:34:35 PM	
1	framework/BlazorWebAssembly.pdb.sha256-JDoJe+K5d1EK1F9ZV9GvatH/Of7BiBhIvR6Mpl7S758=	default	application/octet-str...	15,184	6/20/2022, 4:34:35 PM	
2	framework/Microsoft.AspNetCore.Authorization.dll.sha256-ktawu1p6pD2yy7T/20KVYU7oihmpYm3jrP...	default	application/octet-str...	21,008	6/20/2022, 4:34:34 PM	
3	framework/Microsoft.AspNetCore.Components.Forms.dll.sha256-tr1KtXqlVmQYhG+jn6wsF6Jx7IIPbIEL...	default	application/octet-str...	16,027	6/20/2022, 4:34:34 PM	
4	framework/Microsoft.AspNetCore.Components.Web.dll.sha256-ep2UIBFG2zNXHGp7GY9eWSZYOfsf/R...	default	application/octet-str...	41,119	6/20/2022, 4:34:34 PM	
5	framework/Microsoft.AspNetCore.Components.WebAssembly.dll.sha256-xpgWbaazykL3eLo190s8Jv3...	default	application/octet-str...	32,607	6/20/2022, 4:34:34 PM	
6	framework/Microsoft.AspNetCore.Components.dll.sha256-alrtyx++CQU1VO/6akeZFBPsYgrGGm+Z5ax...	default	application/octet-str...	75,174	6/20/2022, 4:34:34 PM	
7	framework/Microsoft.AspNetCore.Metadata.dll.sha256-dz1QY4sEbXr3Ei0PsRO3VB4ZTBTRL6t3xgcgXuK...	default	application/octet-str...	8,467	6/20/2022, 4:34:34 PM	
8	framework/Microsoft.CSharp.dll.sha256-PpkoBrb5GDDzBCVjEGJ15VkONpHLxk8ZZ3uc0eDYPaY=	default	application/octet-str...	136,891	6/20/2022, 4:34:35 PM	
9	framework/Microsoft.Extensions.Configuration.Abstractions.dll.sha256-NYecKM0ZpUvzj4spvgi6xUu80r...	default	application/octet-str...	11,146	6/20/2022, 4:34:34 PM	
10	framework/Microsoft.Extensions.Configuration.Binder.dll.sha256-JFLykh3r5WjvztBuVlaMM9wfVswD...	default	application/octet-str...	14,343	6/20/2022, 4:34:34 PM	
11	framework/Microsoft.Extensions.Configuration.FileExtensions.dll.sha256-5tCmu87qBh0Xgsp/dG/x9WH...	default	application/octet-str...	13,207	6/20/2022, 4:34:35 PM	
12	framework/Microsoft.Extensions.Configuration.Json.dll.sha256-OM0kYGXmRhGJ0UP2Vf86pt4Ss7QB2/f...	default	application/octet-str...	12,915	6/20/2022, 4:34:35 PM	
13	framework/Microsoft.Extensions.Configuration.dll.sha256-x9zYR1t1is/fStNpBiQnXyPdMpLcQcra3w4R...	default	application/octet-str...	15,801	6/20/2022, 4:34:34 PM	
14	framework/Microsoft.Extensions.DependencyInjection.Abstractions.dll.sha256-D9y5RXbA/sEzkw6cnGb...	default	application/octet-str...	20,083	6/20/2022, 4:34:35 PM	
15	framework/Microsoft.Extensions.DependencyInjection.dll.sha256-W/Xy5S/Zin/Xx5oGhOYdTKJLd1mRSi...	default	application/octet-str...	34,724	6/20/2022, 4:34:35 PM	
16	framework/Microsoft.Extensions.FileProviders.Abstractions.dll.sha256-aUKLFqSgB/hwZc41dLVKoRSKRg...	default	application/octet-str...	9,305	6/20/2022, 4:34:35 PM	

Total entries: 193

## 4. PROS and CONS Blazor WebAssembly and Blazor Server

### Blazor Server



### PROS

- ❑ NO need to download before running
- ❑ Server-side rendering
- ❑ Load fast in 1<sup>st</sup> time
- ❑ Can be used in older browser
- ❑ API private

### CONS

- ❑ SignalR -> latency on each event
- ❑ Always keeps connection
- ❑ NO offline mode
- ❑ High memory usage

## 4. PROS and CONS Blazor WebAssembly and Blazor Server

### Blazor Server

```
19 public static IHostBuilder CreateHostBuilder(string[] args) =>
20     Host.CreateDefaultBuilder(args)
21     .ConfigureLogging(logging =>
22     {
23         logging.ClearProviders();
24         logging.AddConsole();
25         logging.AddFilter(
26             "Microsoft.AspNetCore.SignalR", LogLevel.Trace);
27         logging.AddFilter(
28             "Microsoft.AspNetCore.Http.Connections",
29             LogLevel.Trace);
30     })
31     .ConfigureWebHostDefaults(webBuilder =>
32     {
33         webBuilder.UseStartup<Startup>();
34     });
35 }
36 }
37
```



# 4. PROS and CONS Blazor WebAssembly and Blazor Server

## Blazor Server

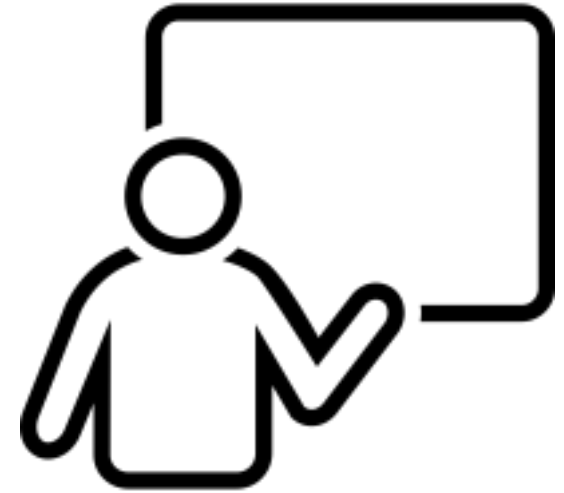
The screenshot shows a Blazor Server application running in a browser. The application has a counter and a 'Fetch data' button. The console window is open, showing network logs and a debugger. The debugger shows the following log:

```
InvocationId 0: Sending result of type 'System.String'.
trce: Microsoft.AspNetCore.Http.Connections.Internal.Transports.WebSocketsTransp
ort[11]
Sending payload: 229 bytes.
trce: Microsoft.AspNetCore.Http.Connections.Internal.Transports.WebSocketsTransp
ort[9]
Message received. Type: Binary, size: 26, EndOfMessage: True.
dbug: Microsoft.AspNetCore.SignalR.Internal.DefaultHubDispatcher[1]
Received hub invocation: InvocationMessage { InvocationId: "", Target: "On
RenderCompleted", Arguments: [ 2, ], StreamIds: [ ] }.
trce: Microsoft.AspNetCore.SignalR.Internal.DefaultHubDispatcher[7]
InvocationId (null): Sending result of type 'System.Void'.
trce: Microsoft.AspNetCore.Http.Connections.Internal.Transports.WebSocketsTransp
ort[11]
Sending payload: 93 bytes.
trce: Microsoft.AspNetCore.Http.Connections.Internal.Transports.WebSocketsTransp
ort[9]
Message received. Type: Binary, size: 44, EndOfMessage: True.
dbug: Microsoft.AspNetCore.SignalR.Internal.DefaultHubDispatcher[1]
Received hub invocation: InvocationMessage { InvocationId: "", Target: "En
dInvokeJSFromDotNet", Arguments: [ 2, True, [2,true,null] ], StreamIds: [ ] }.
trce: Microsoft.AspNetCore.SignalR.Internal.DefaultHubDispatcher[7]
InvocationId (null): Sending result of type 'System.Void'.
```

The network logs show the following requests:

Request	Method	URL	Size	Time	Waterfall
localhost	GET	localhost	0 B	Pending	
data	GET	data	3.6 kB	43 ms	
bootstrap.min.css	GET	bootstrap.min.css	89 B	20 ms	
css	GET	css	967 B	25 ms	
orServer.styles.css	GET	orServer.styles.css	65 B	15 ms	
or.server.js	GET	or.server.js	88 B	27 ms	
netcore-browser-refresh.js	GET	netcore-browser-refresh.js	12.1 kB	27 ms	
n-iconic-bootstrap.min.css	GET	n-iconic-bootstrap.min.css	35 B	7 ms	
otiate?negotiateVersion=1	GET	otiate?negotiateVersion=1	370 B	14 ms	
n-iconic.woff	GET	n-iconic.woff	57 B	12 ms	
on.ico	GET	on.ico	5.5 kB	12 ms	
zor?id=POd3v-azeG_Gf2EuaJmAuQ	GET	zor?id=POd3v-azeG_Gf2EuaJmAuQ	0 B	Pending	

Demo



## References

- ❑ Server-Side vs Client-Side Programming Languages

<https://www.techwebpace.com/server-side-vs-client-side-programming-languages/>

- ❑ WebAssembly

<https://arghya.xyz/articles/webassembly-wasm-wasi/>

[https://developer.mozilla.org/en-US/docs/WebAssembly/Understanding the text format](https://developer.mozilla.org/en-US/docs/WebAssembly/Understanding_the_text_format)

- ❑ Blazor

<https://docs.microsoft.com/en-us/aspnet/core/blazor/hosting-models?view=aspnetcore-5.0>

## Some relevant questions

- ❑ ASP.NET Core hosted vs Blazor Serverside

<https://stackoverflow.com/questions/58093386/whats-the-difference-between-asp-net-core-hosted-and-server-side-blazor-really>

- ❑ JS vs WASM

<https://medium0.com/@OPTASY.com/webassembly-vs-javascript-is-wasm-faster-than-js-when-does-javascript-perform-better-db86d2ecf2cc>

Thank you for watching

---

