ASP.NET Core Blazor

Modele renderowania

Server Side Rendering (SSR)

/index.html
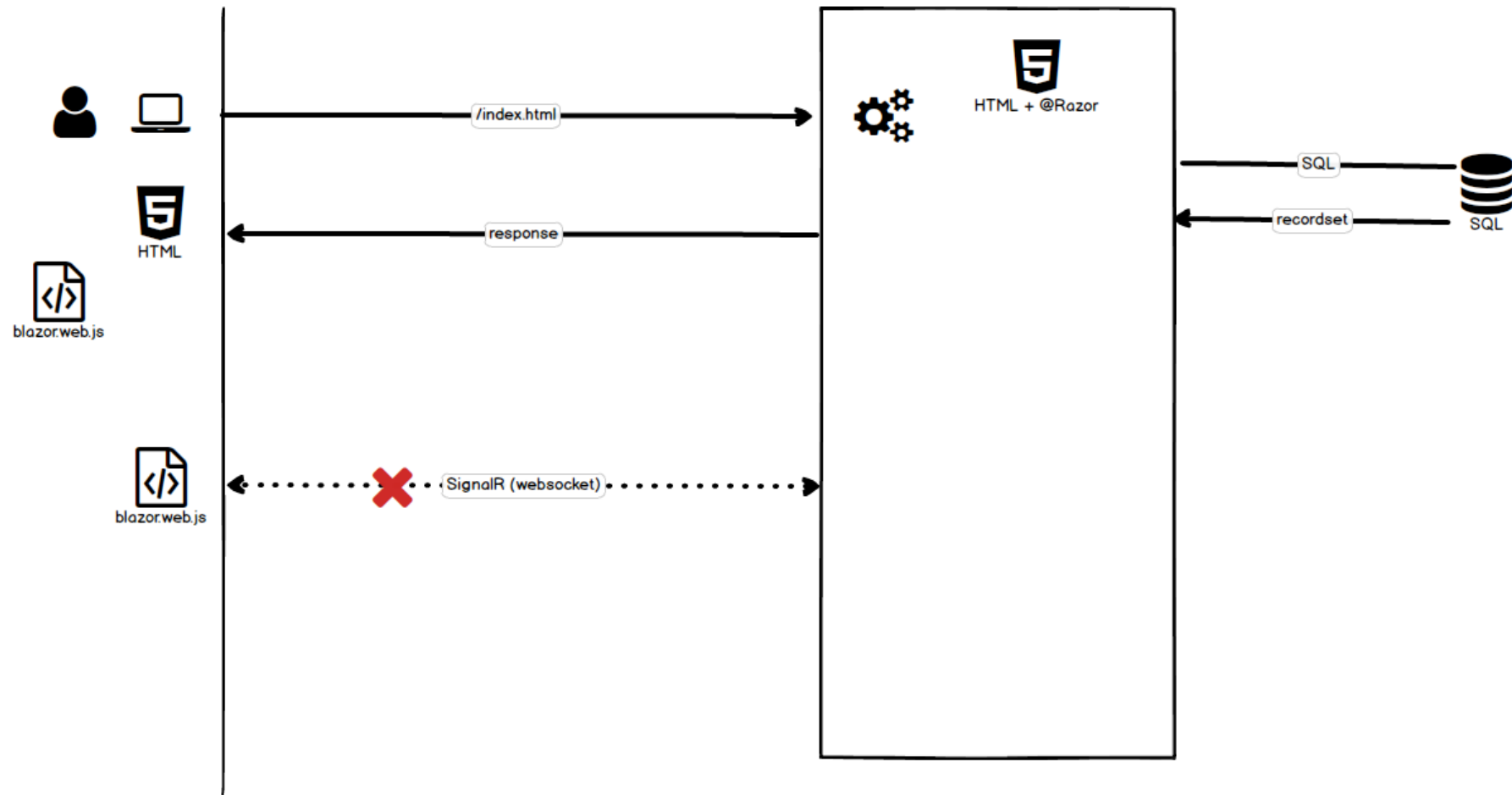
template

SQL

response

recordset

index.html

/index.html?search=Warszawa

response

index.html
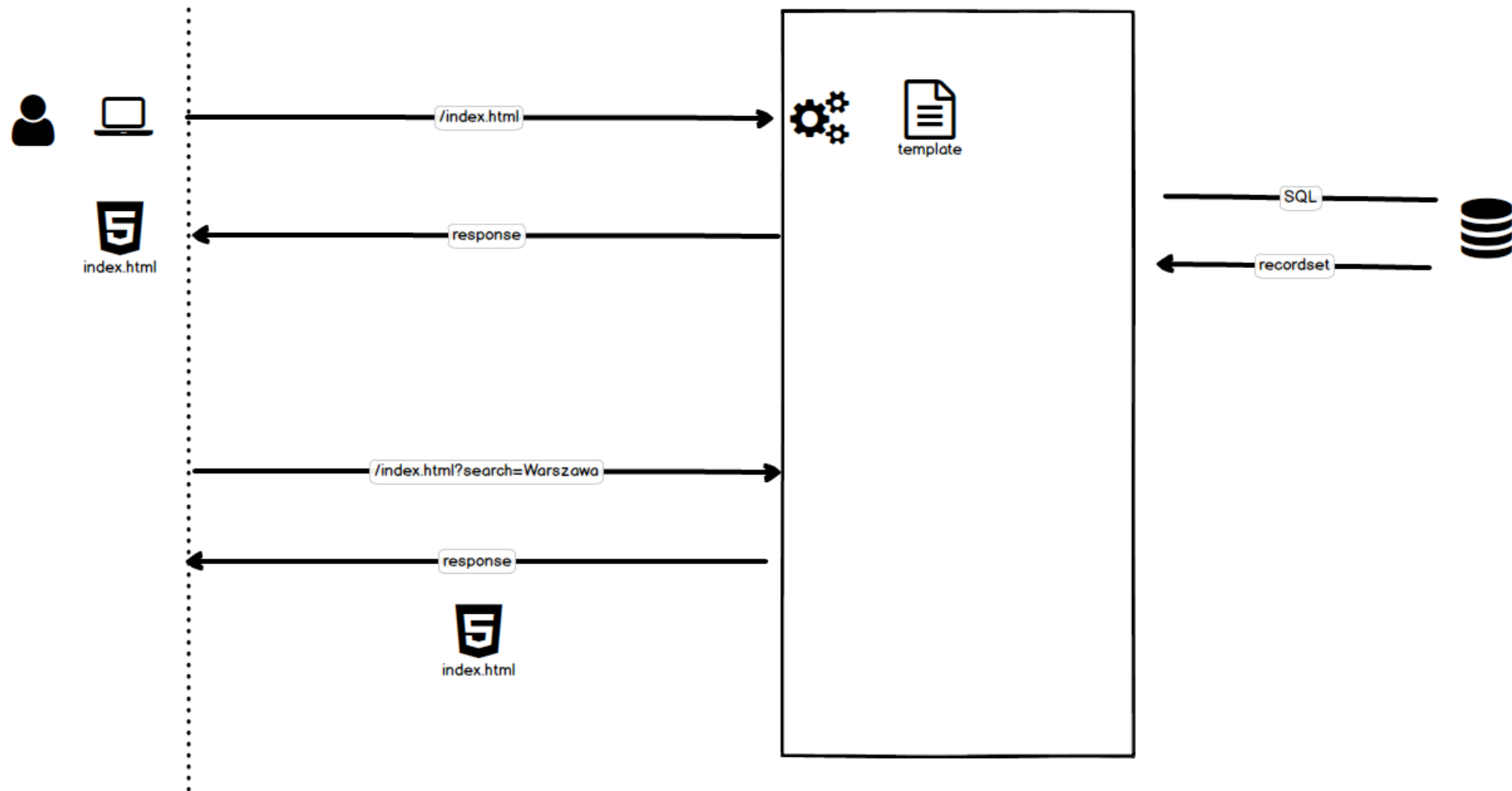
Client Side Rendering (CSR)

/index.html

response

index.html index.js

api?search=Warszawa

Api

response

xml / json

SQL

recordset
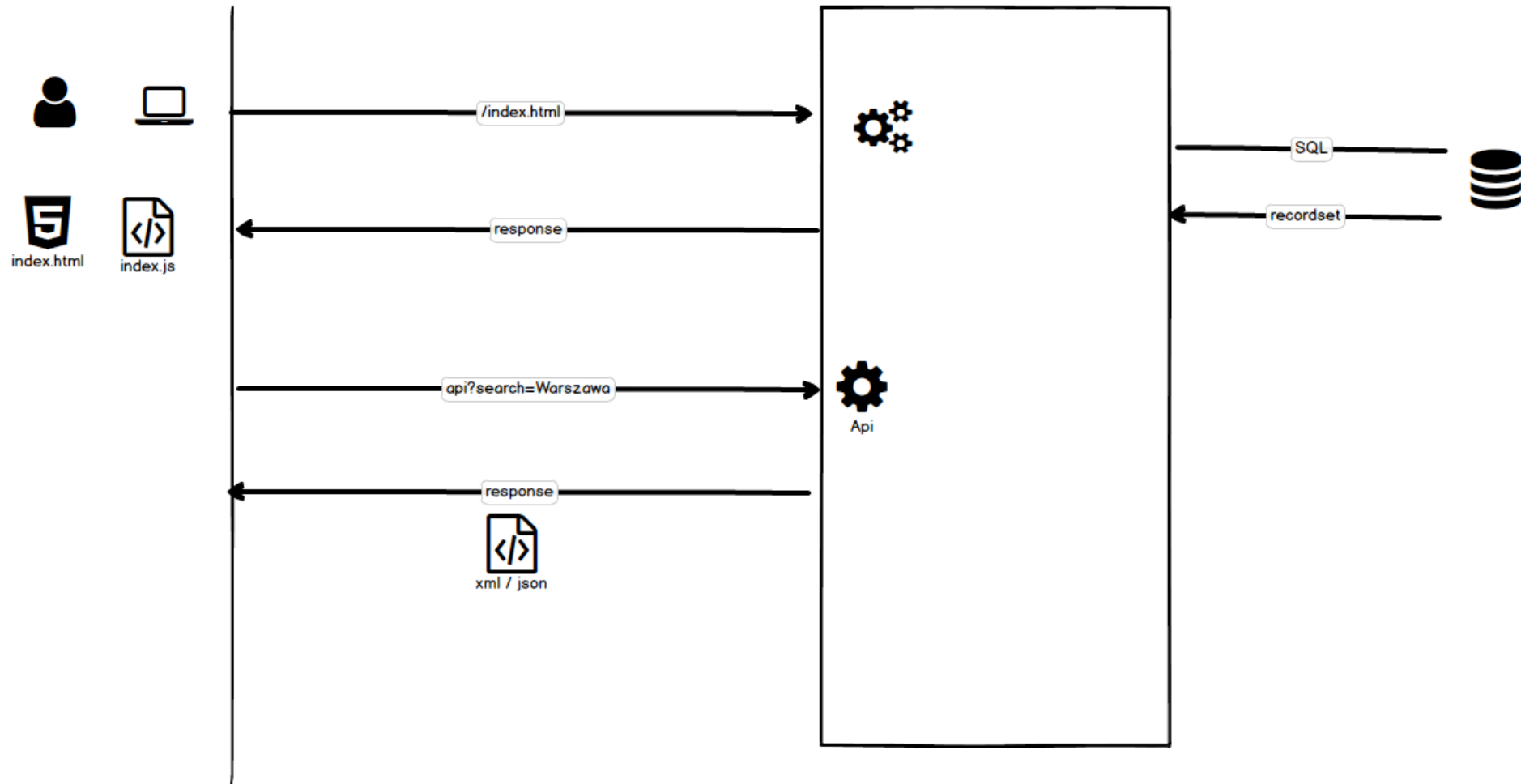
Blazor Hosting Models

Blazor Server (Blazor Server-Side)

/index.html

response

HTML + @Razor

SQL

recordset

SQL

HTML

blazor.web.js

blazor.web.js

SignalR (websocket)

Blazor Web Assembly (Blazor Client-Side)

C# -> wasm

/index.html

HTML + @Razor

SQL

response

app.wasm

recordset

HTML   app.wasm

GET /customers

JSON

Api
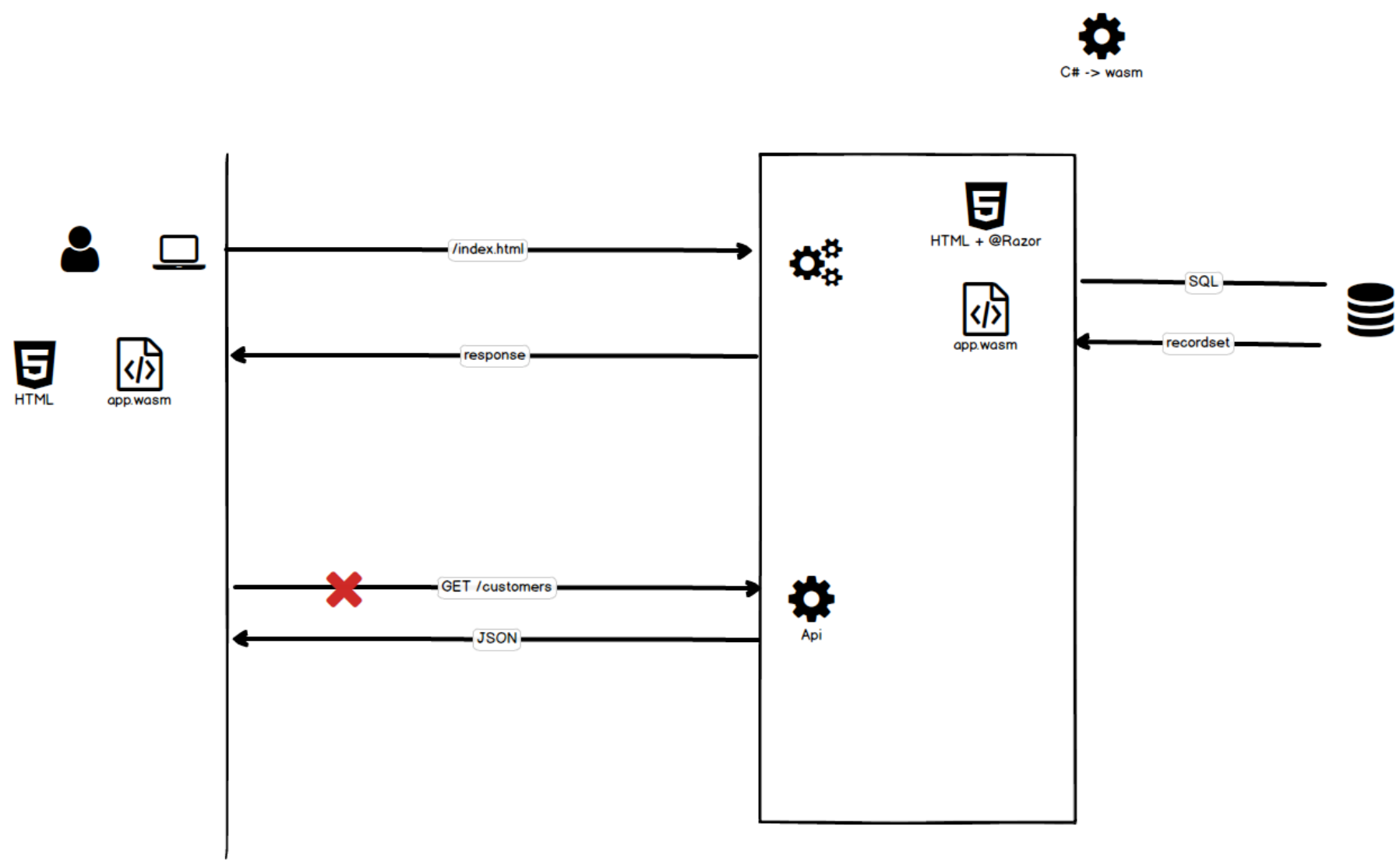
Project Structure

Project Structure - Blazor Server

```
┌─────────────────────────┐              ┌─────────────────────────┐
│       Program.cs        │              │        Browser          │
└─────────────────────────┘              └─────────────────────────┘
             │                                        │
     MapRazorComponents<App>()                        │
             │                                        │
             ▼                                        │
┌─────────────────────────┐              GET /counter │
│        App.razor        │◄──────────────────────────┘
└─────────────────────────┘
             │
             ▼
┌─────────────────────────┐
│       Routes.razor      │
└─────────────────────────┘
             │
       DefaultLayout
             │
             ▼
┌─────────────────────────┐          ┌─────────────────────────┐
│     MainLayout.razor    │─────────▶│      NavMenu.razor       │
└─────────────────────────┘          └─────────────────────────┘
             │
          @Body
             │
             ▼
┌─────────────────────────┐
│      Counter.razor      │     @page "/counter"
└─────────────────────────┘
```
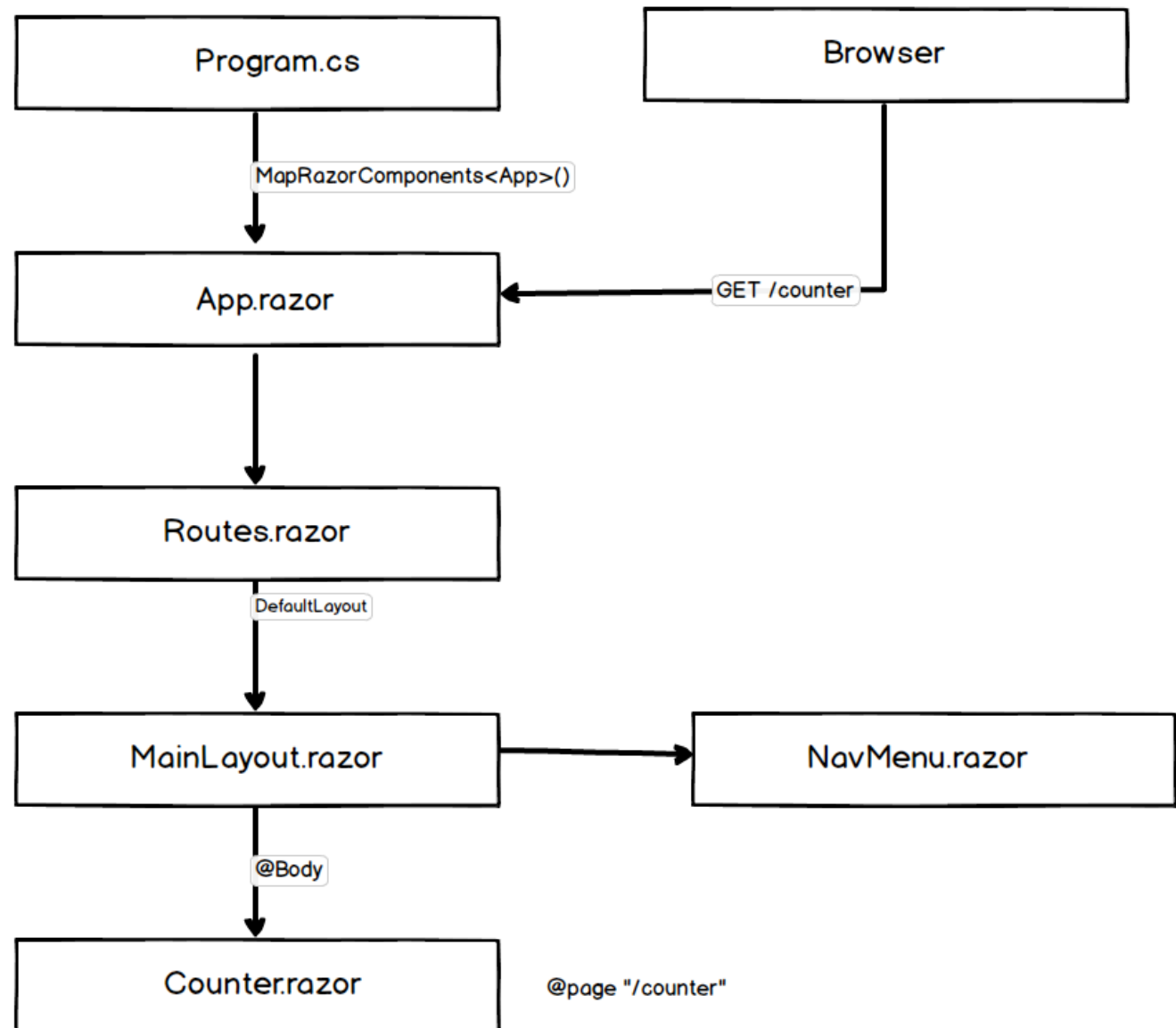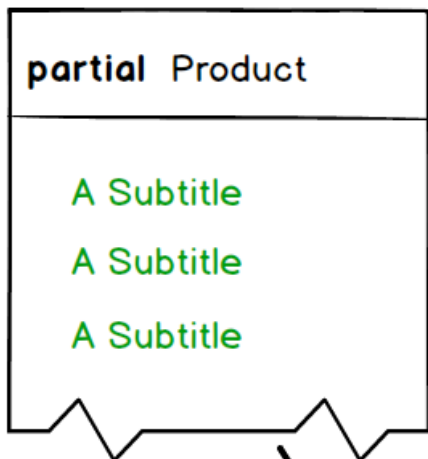
Blazor Web Assembly

Pages

product.cs

product.g.cs

partial Product

A Subtitle

A Subtitle

A Subtitle

partial Product

A Subtitle

A Subtitle

partial Product : Base

A Subtitle

A Subtitle

A Subtitle

A Subtitle

A Subtitle
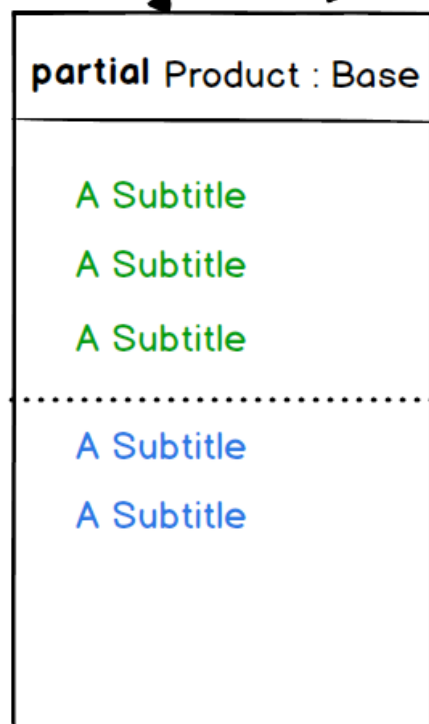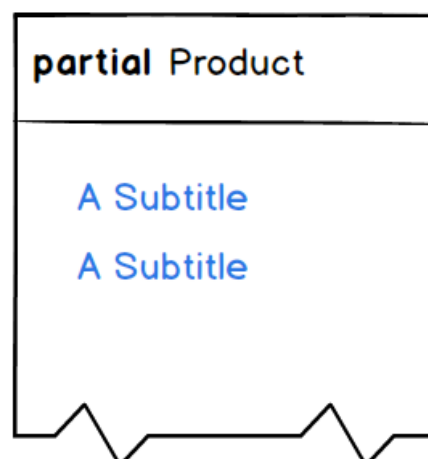
Parameters

Parameter

Page

ParentComponent

ChildComponent A

[Parameter]

string MyParameter { get; set; }

ChildComponent B

[Parameter]

string MyParameter { get; set; }

Cascading Parameter

Page

CascadingValue

ParentComponent

ChildComponent A

[CascadingParameter]

string MyParameter { get; set; }

ChildComponent B

[CascadingParameter]

string MyParameter { get; set; }
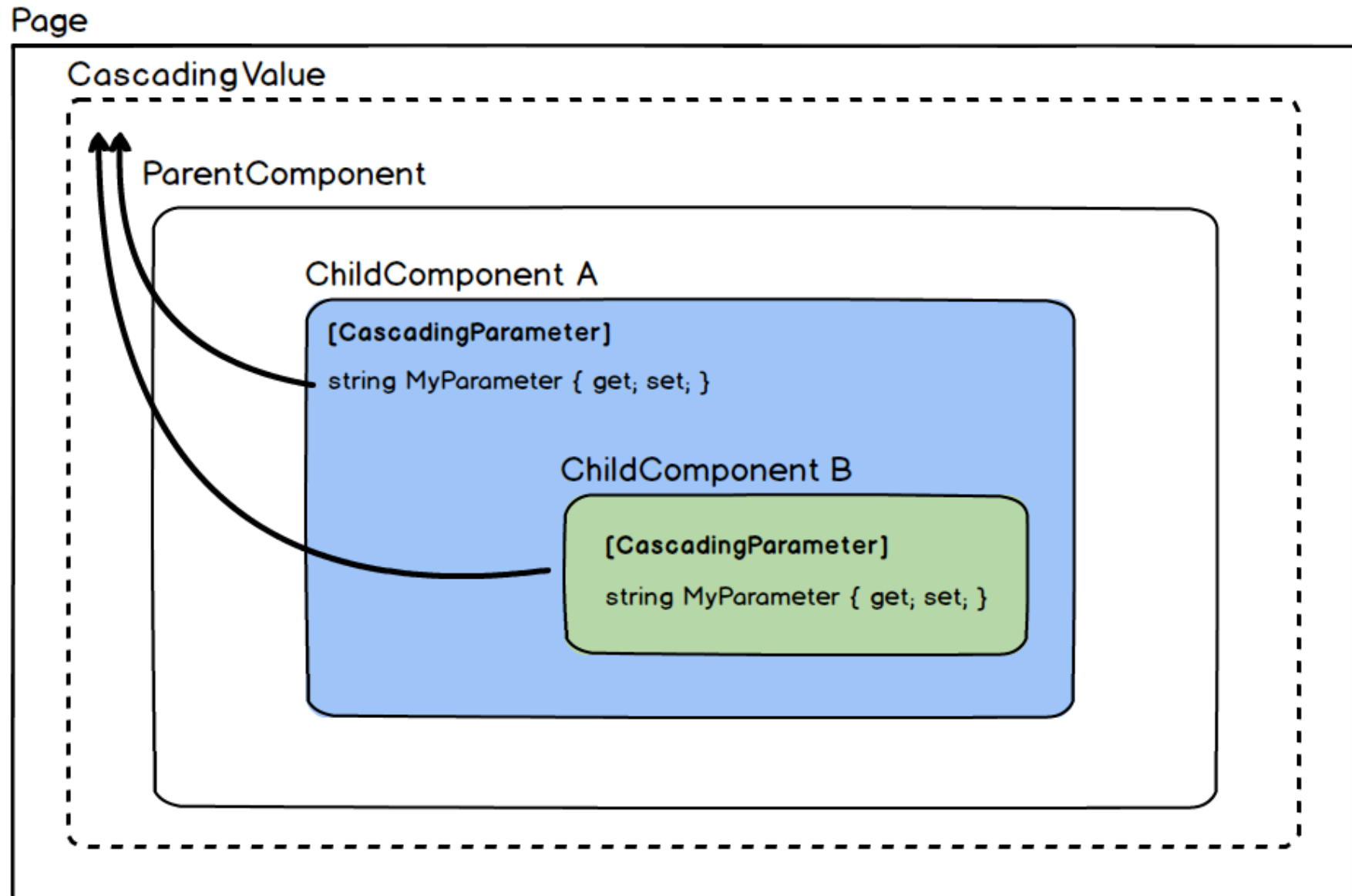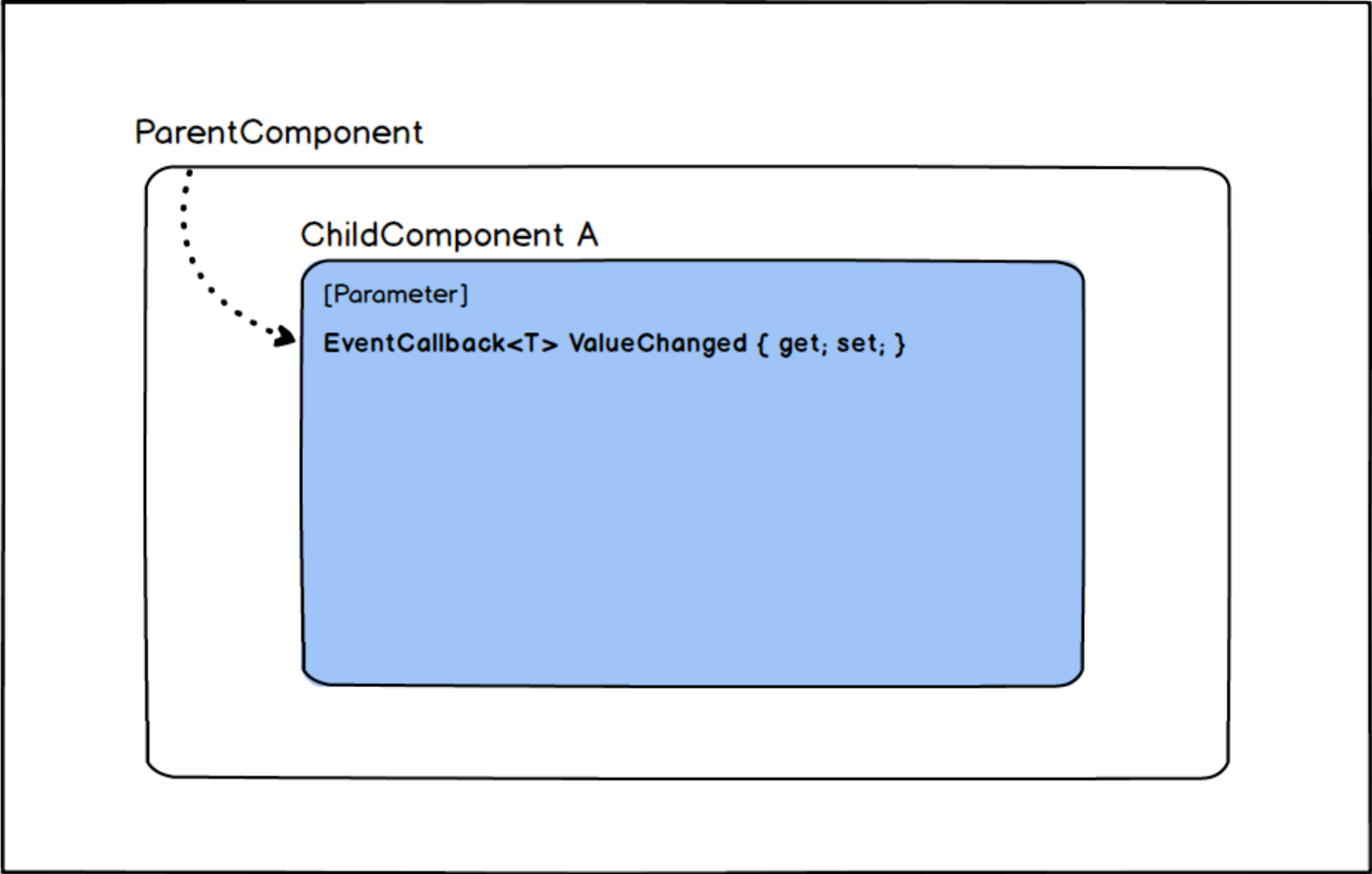
EventCallback

Page

ParentComponent

ChildComponent A
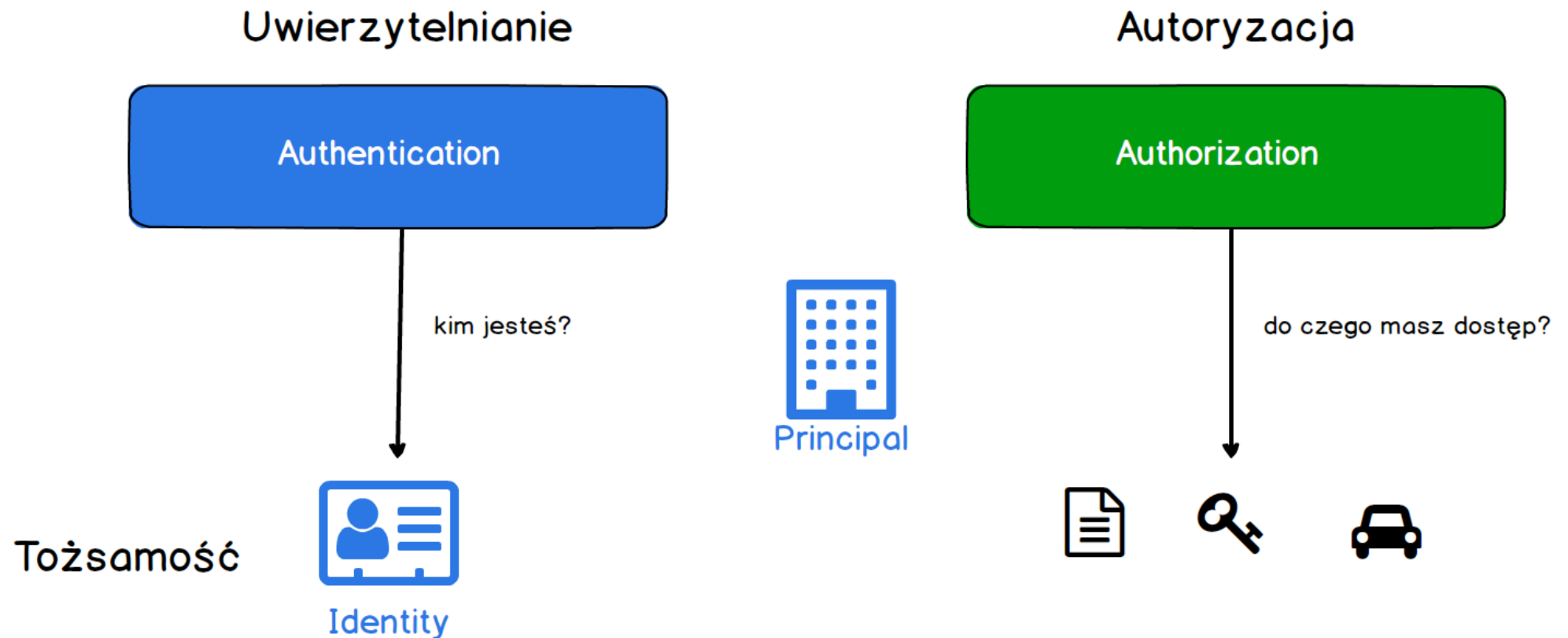
[Parameter]

EventCallback<T> ValueChanged { get; set; }

Authentication Authorization

Authentication / Authorization

Uwierzytelnianie

Authentication

kim jesteś?

Tożsamość

Identity

Principal

Autoryzacja

Authorization

do czego masz dostęp?

.UseAuthentication()

.UseAuthorization()

IIdentity vs IPrincipal

Identity

Principal

IIdentity
AuthenticationType { get; }
IsAuthenticated { get; }
Name { get; }

IPrincipal
Identity { get; }
bool IsInRole(role)

ClaimsIdentity
Claim[] Claims { get; }

ClaimsPrincipal
Claim[] Claims { get; }

*

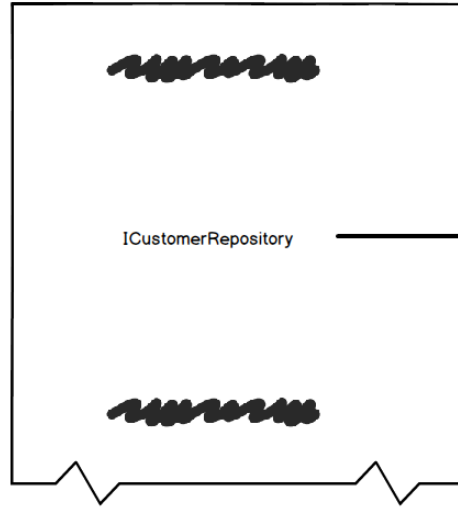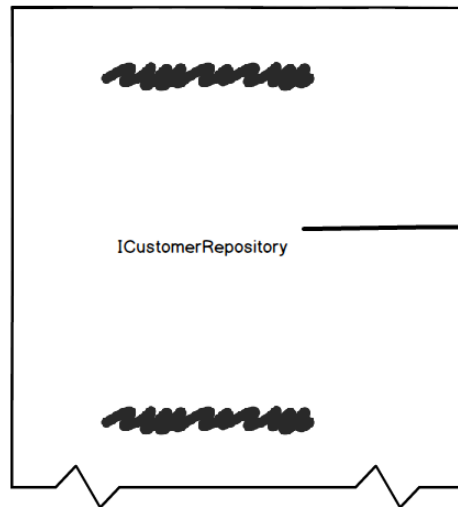| Claim |
|---|
| *string Type { get; }* |
| *object Value { get; }* |
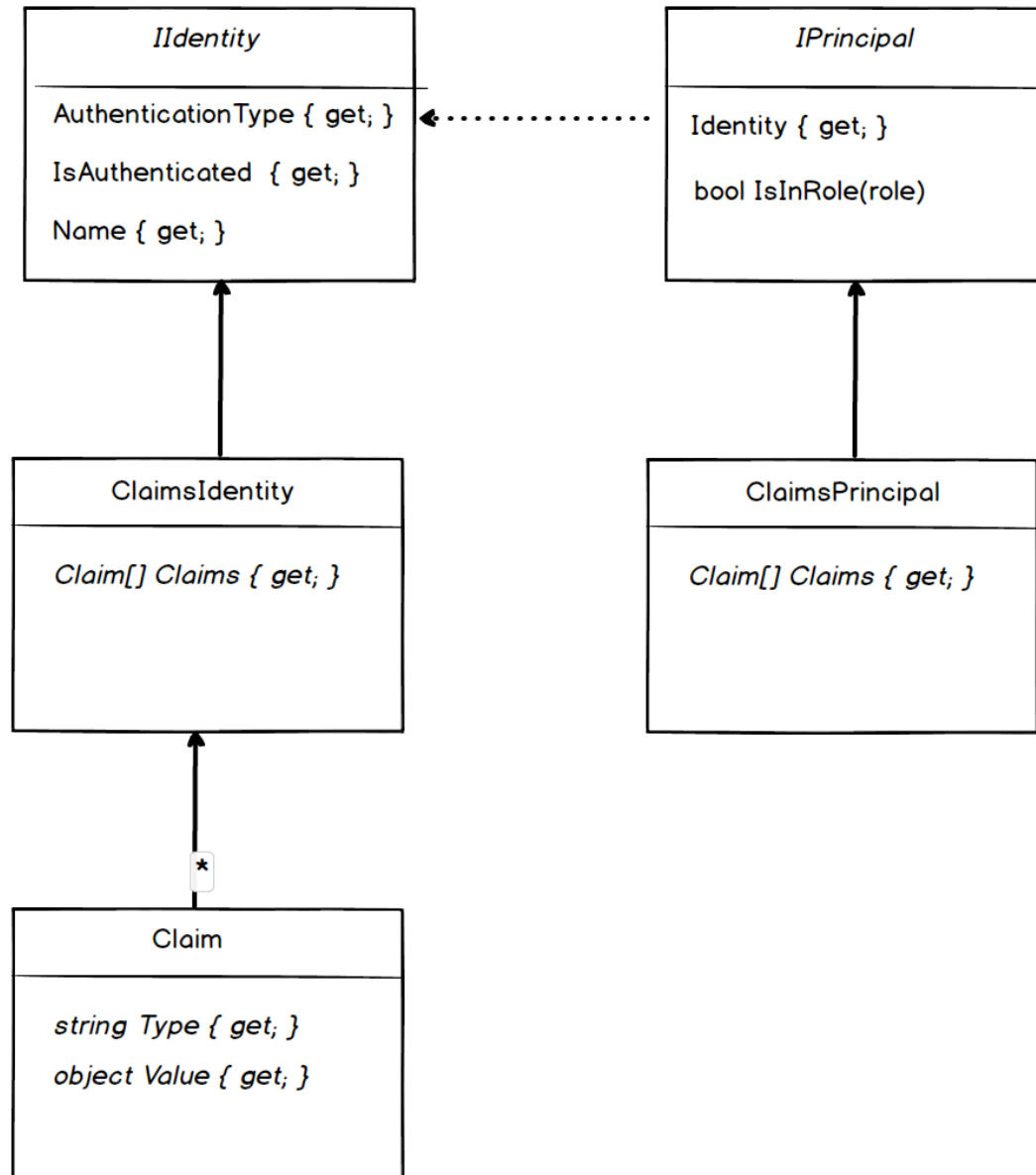
Dependency Injection

ICustomerRepository

GetAll()

@Page "/customers"

ICustomerRepository

DbCustomerRepository

GetAll()

SQL

SQL

FileCustomerRepository

GetAll()

Load()

System.IO

customers.csv

@Page "/customer"

ICustomerRepository
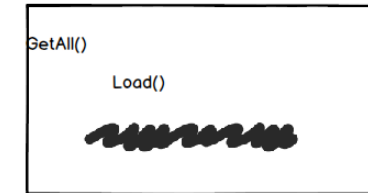
Real Time Communication

Signal-R

/index.html

template

response

dashboard.html

SQL

recordset

Client ········· websocket ·········► Hub

json

BackgroundService

request → Sensor

request → External Api

Blazor ♥ JS

Blazor → script.js
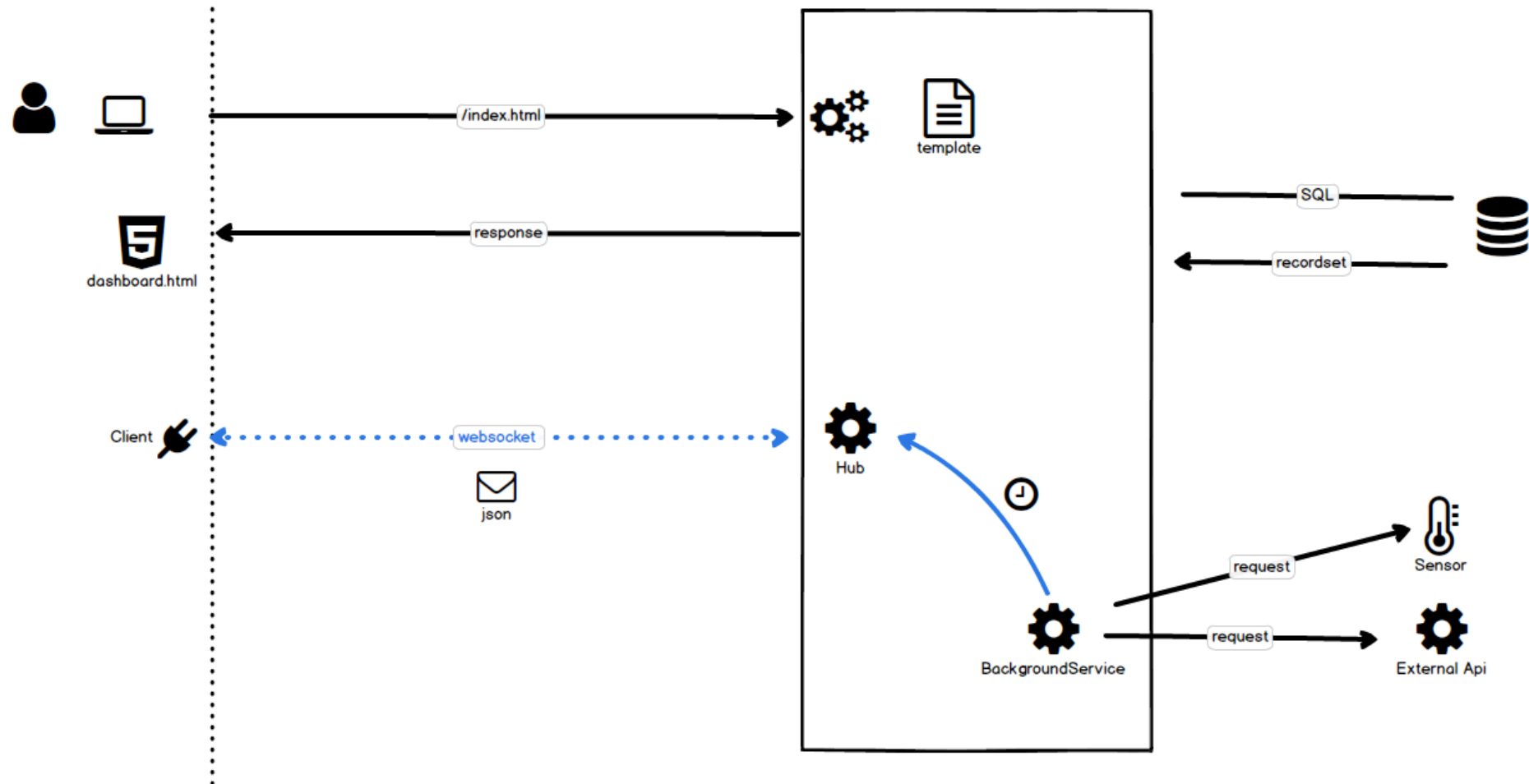
Blazor ← script.js
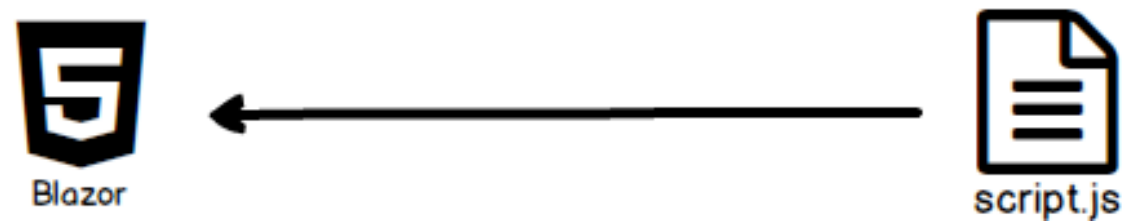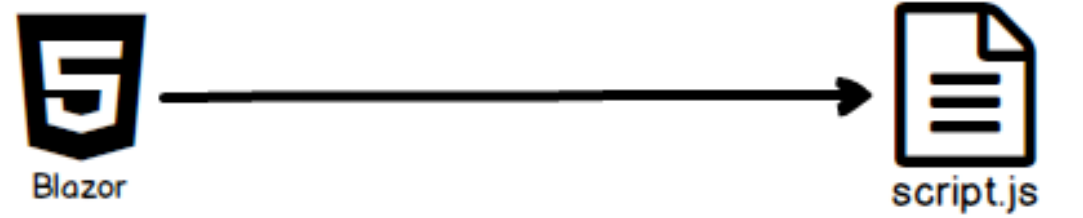
Gateway API

CORS (Cross-Origin Resource Sharing)

localhost:5000/index.html

HTML + @Razor

C# -> wasm

SQL

response

app.wasm

recordset

HTML

app.wasm

CORS — GET localhost:5001/api/customers — Api

CORS — GET localhost:5002/api/products — Api

Gateway API

localhost:5000/index.html → forward → localhost:5000/index.html

appsettings

response

HTML

app.wasm

Gateway.Api

GET localhost:5000/api/customers → forward → GET localhost:5001/api/customers

response

GET localhost:5000/api/products → forward → GET localhost:5002/api/products

response

YARP / Ocelot

HTML + @Razor

C# -> wasm

SQL

app.wasm

recordset

Api

Api

Api

Zadania

Zadanie #1 Lista produktów

A Web Page

http://localhost:5000/products

Product List

Name	Color	Price	
Produkt 1	red	100	View
Produkt 2	blue	90	View
Produkt 3	green	100	View
Produkt 4	red	50	View
Produkt 5	red	20	View

http://localhost:5000/products/1

Product1

Color:

200 zl

Zadanie #2 Dashboard

http://localhost/dashboard

Customers

1000

Products

10

Avg Price

99,99 zł

Status

MAY 2025

S	M	T	W	T	F	S
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31