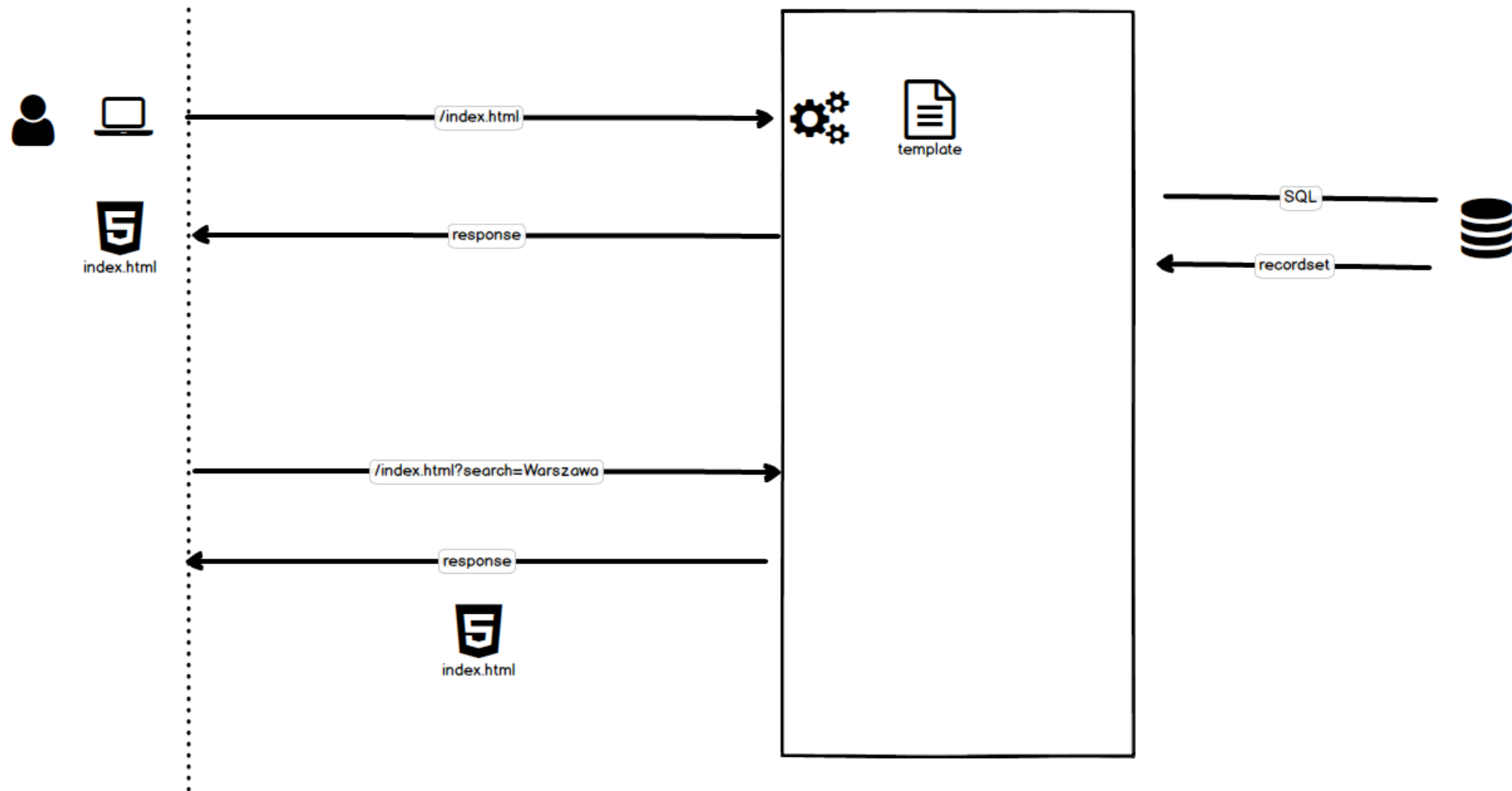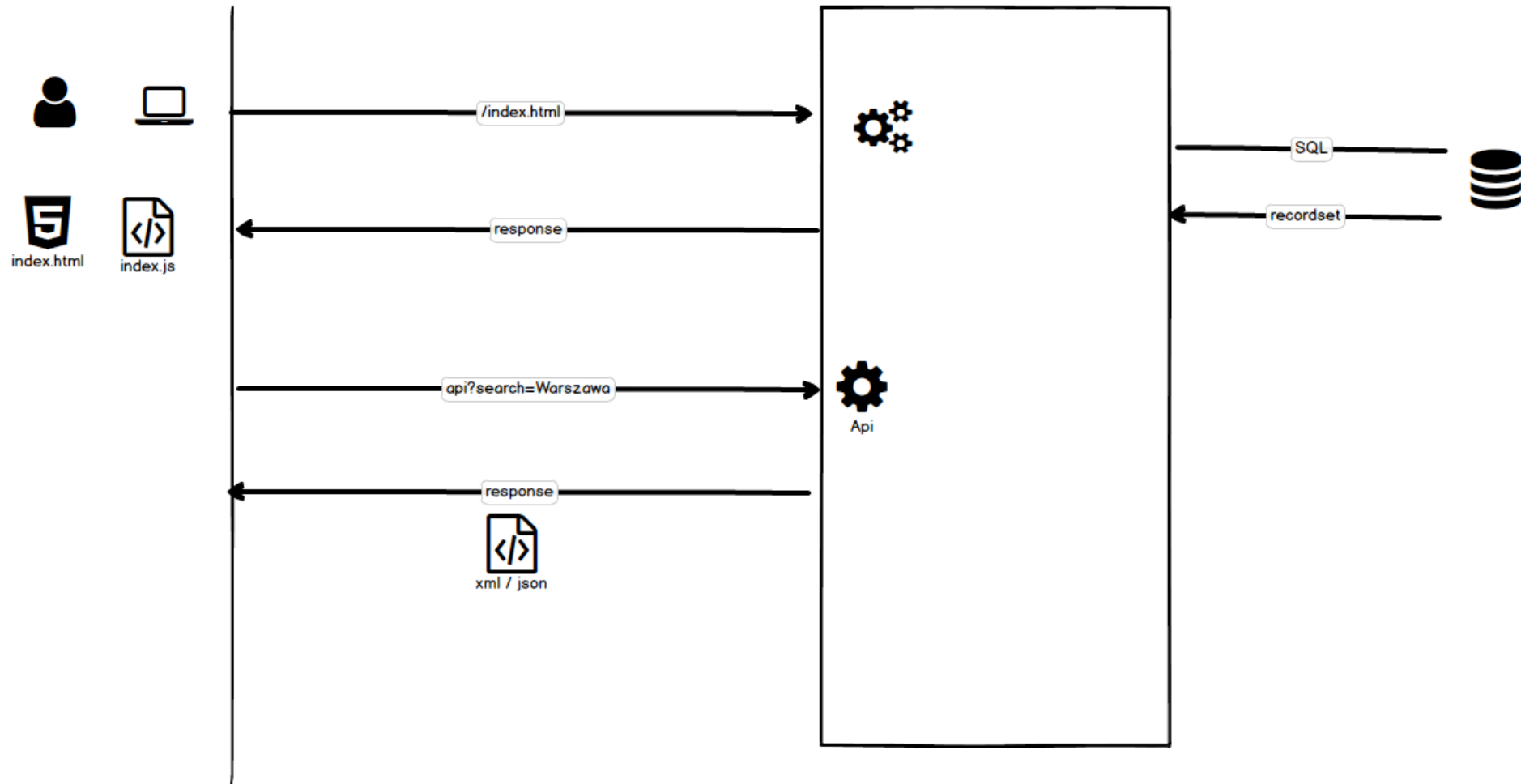# ASP.NET Core Blazor

Modele renderowania

Server Side Rendering (SSR)
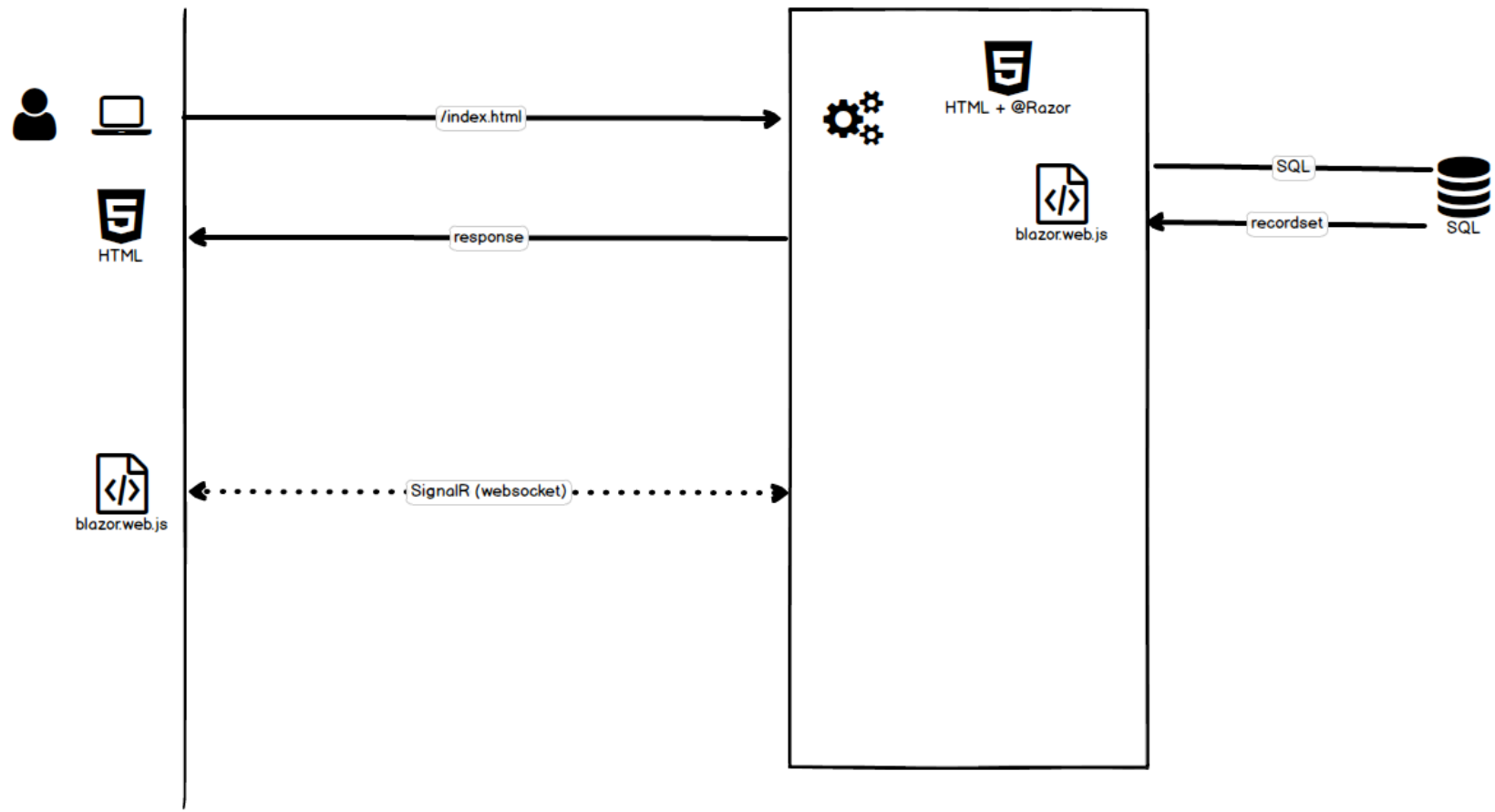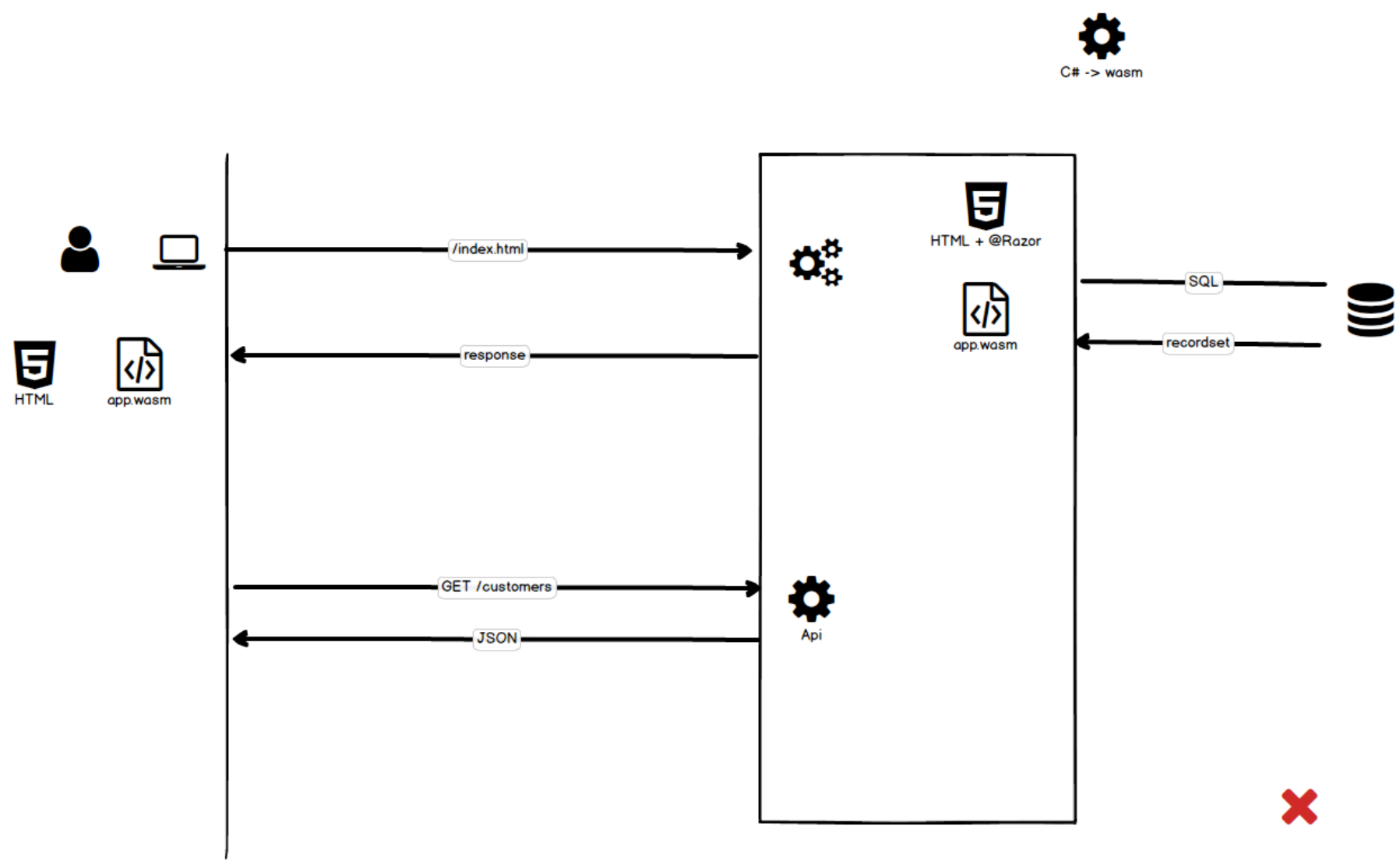
Client Side Rendering (CSR)

Blazor Hosting Models

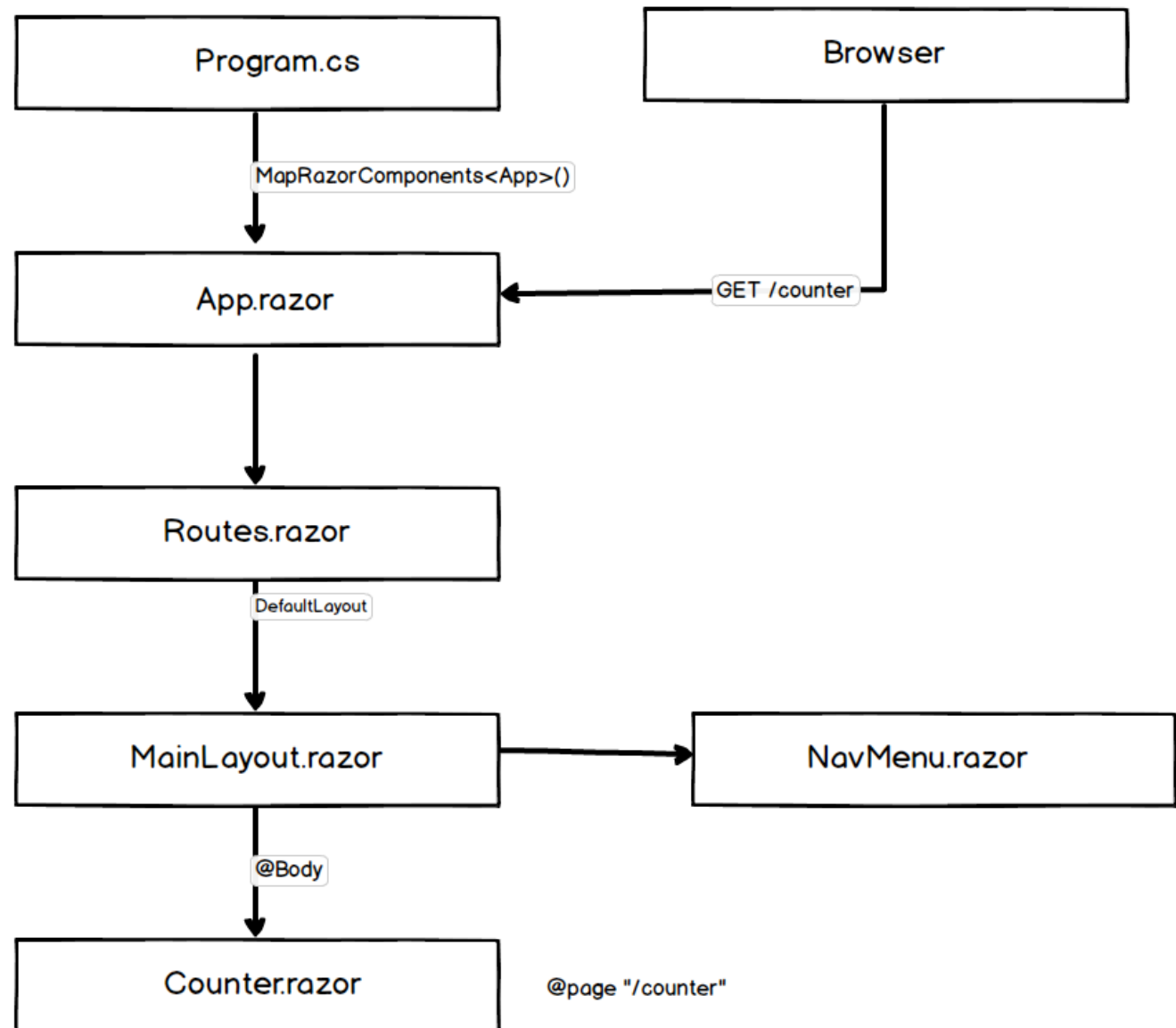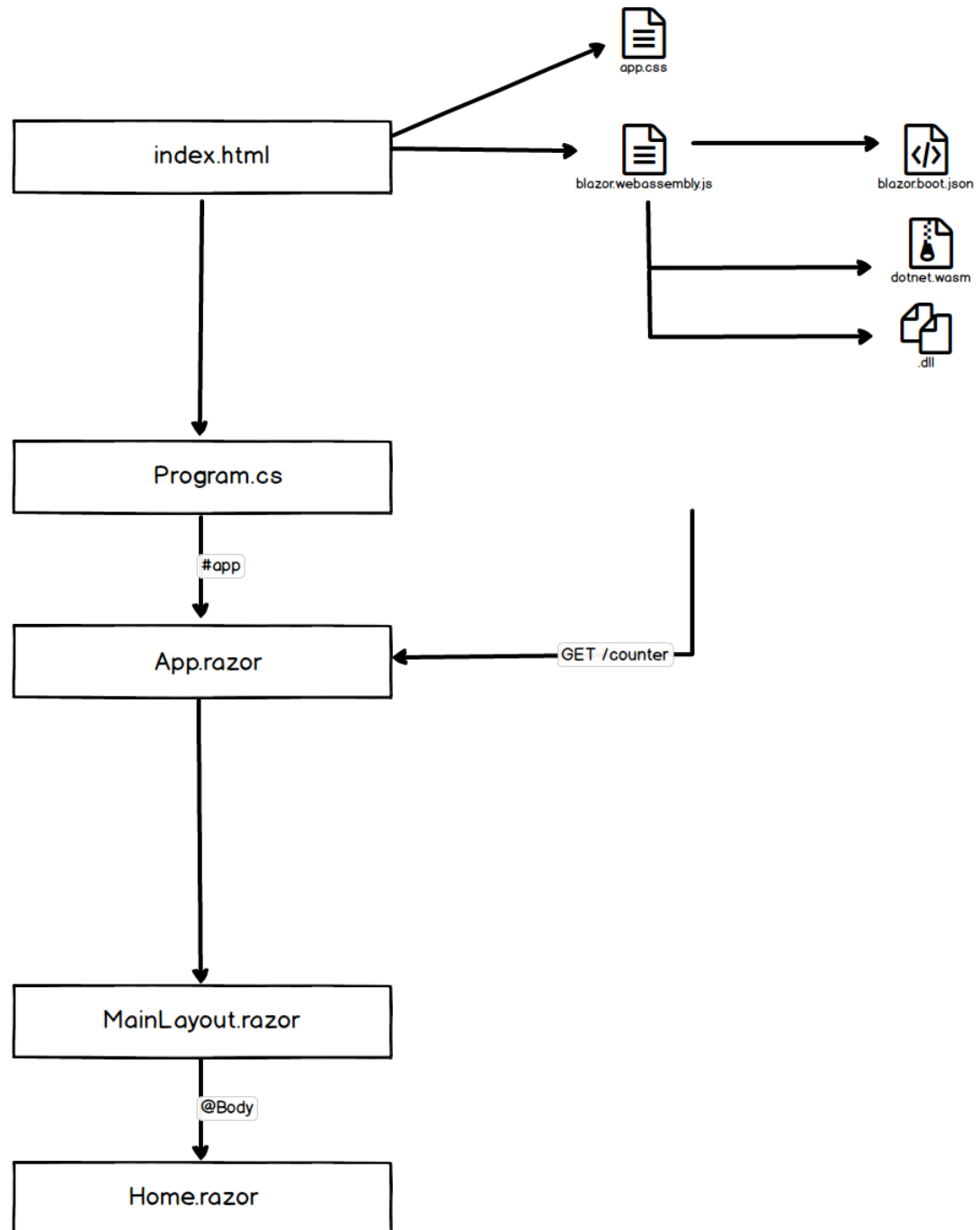# Blazor Server (Blazor Server-Side)

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 - 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


```
app.css

index.html ──→ blazor.webassembly.js ──→ blazor.boot.json
                                      ──→ dotnet.wasm
                                      ──→ .dll
    │
    ▼
Program.cs
    │ #app
    ▼
App.razor ←── GET /counter
    │
    ▼
MainLayout.razor
    │ @Body
    ▼
Home.razor
```

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; }

Event Handler

Page

ParentComponent

ChildComponent A
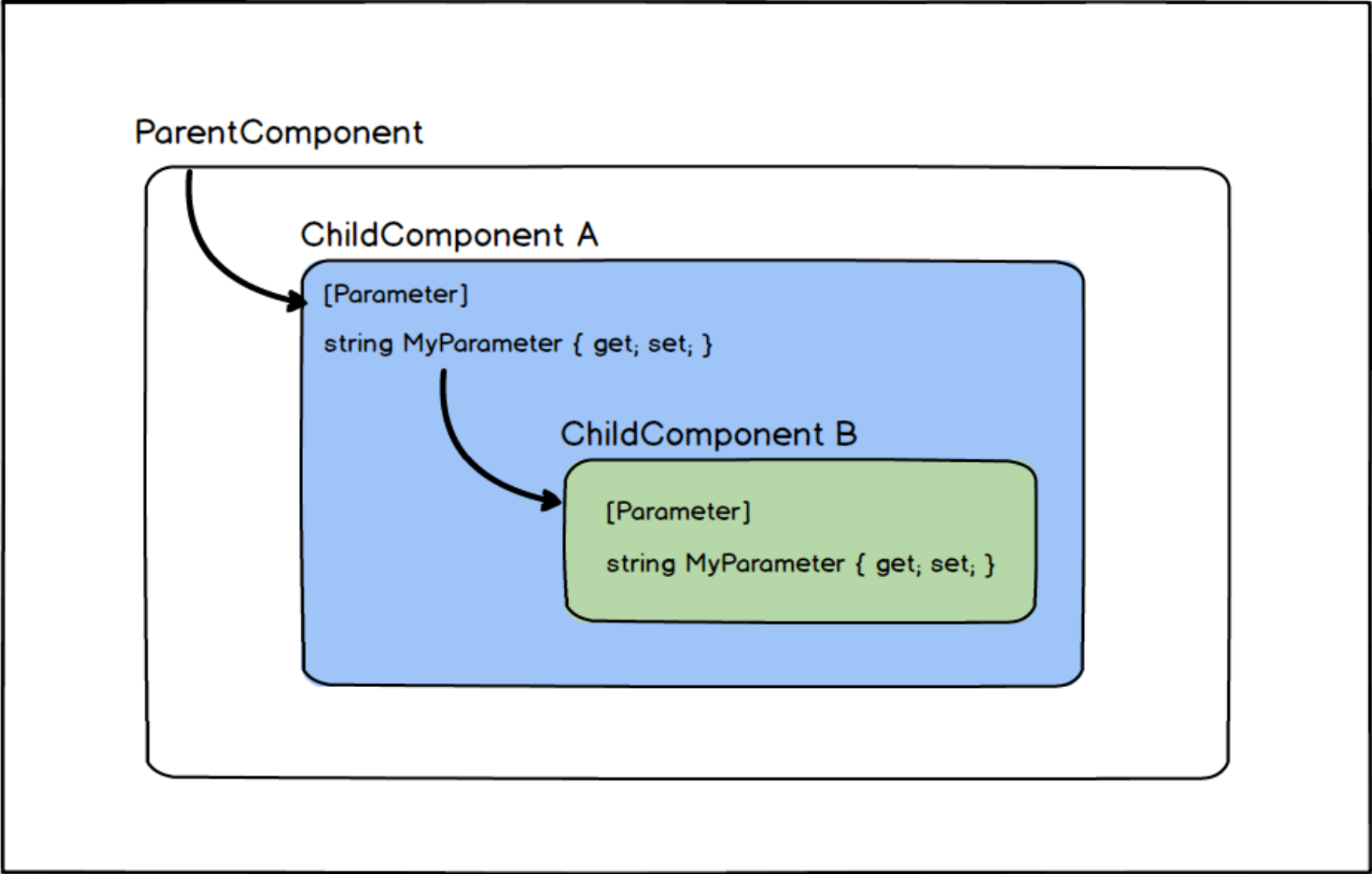
[Parameter]

EventHandler ValueChanged { get; set; }
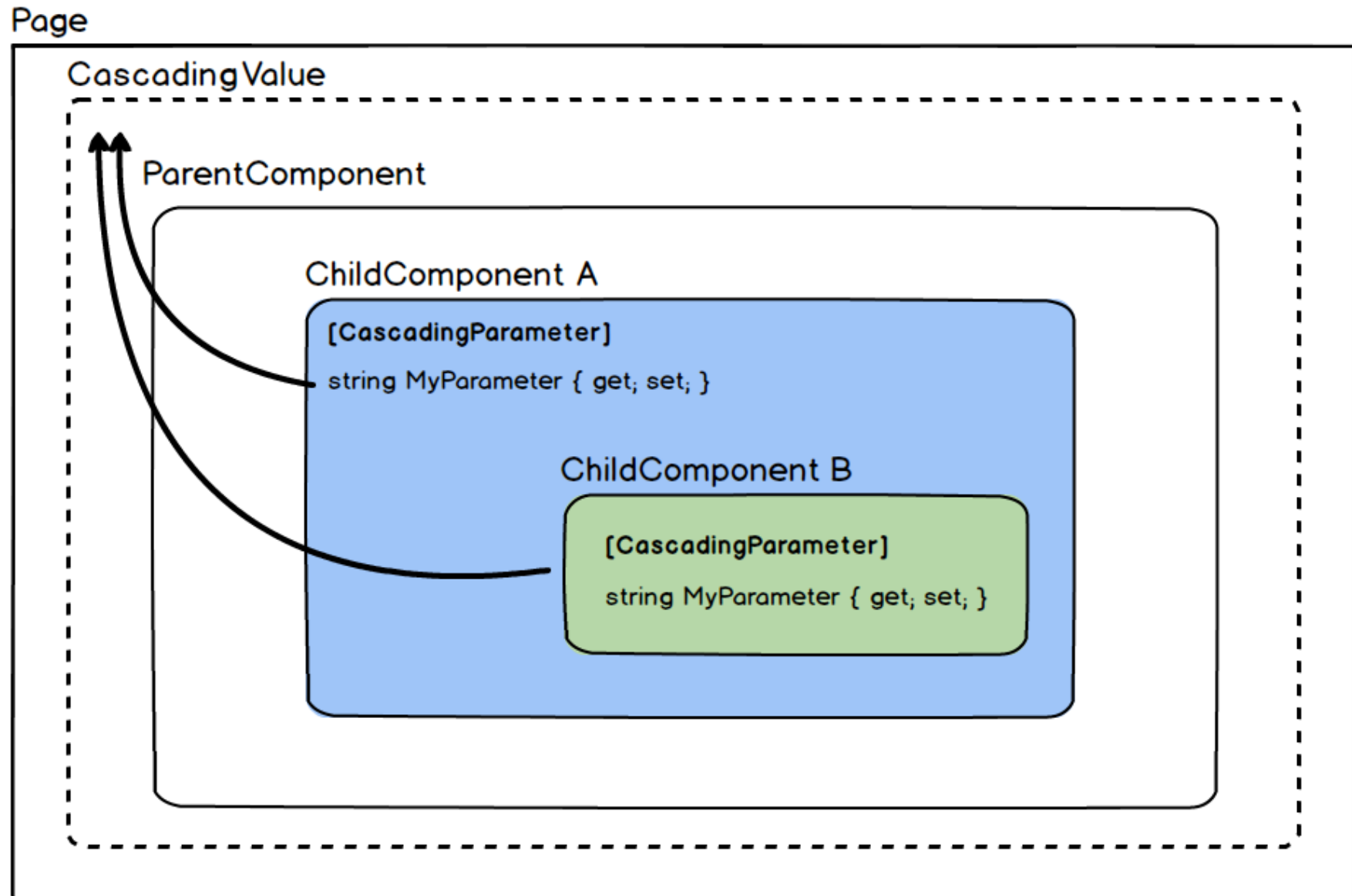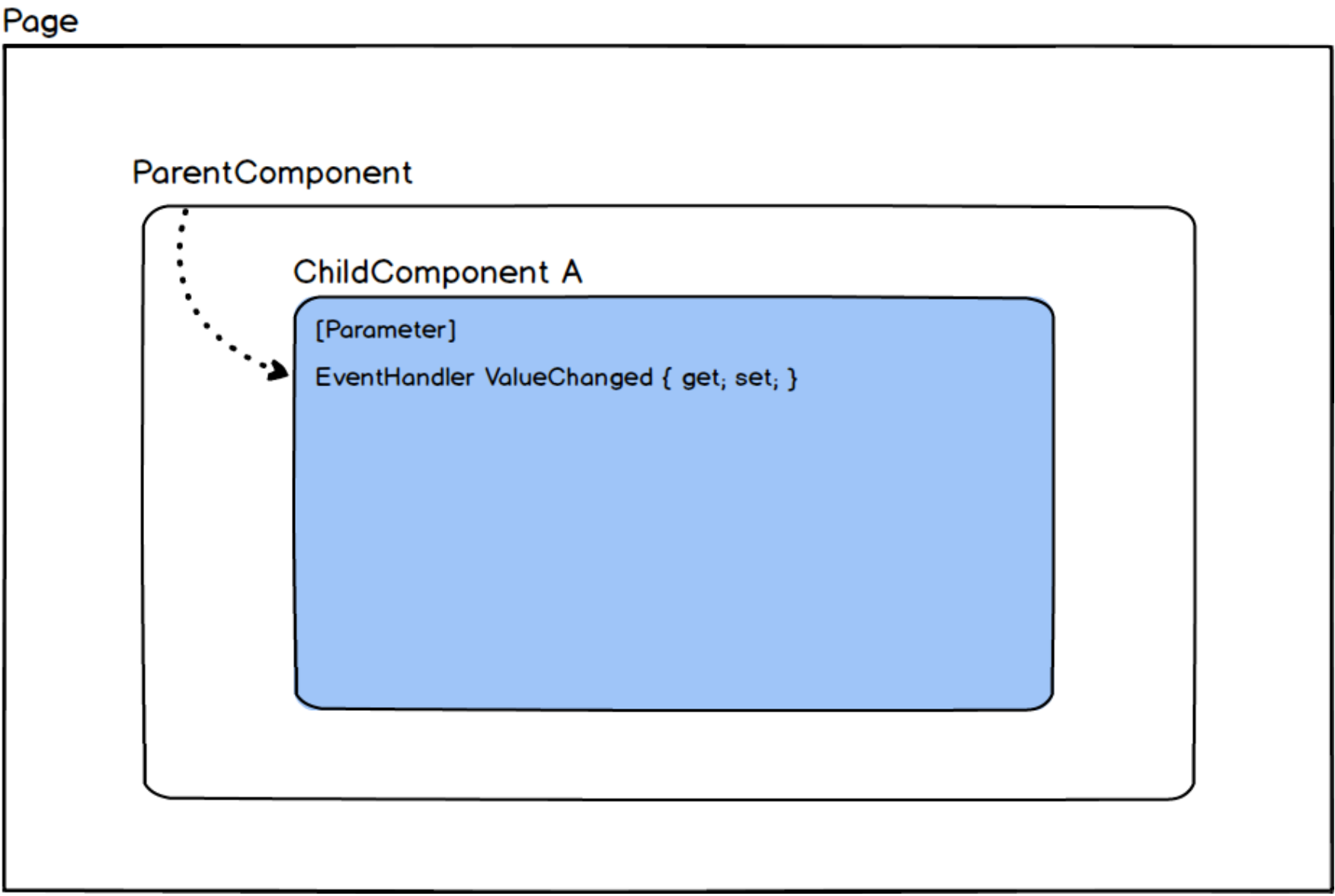
Authentication Authorization

Authentication vs Authorization

Uwierzytelnianie

Authentication

Principal

kim jesteś?

Identity

Autoryzacja

Authorization

do jakich zasobów masz dostęp?
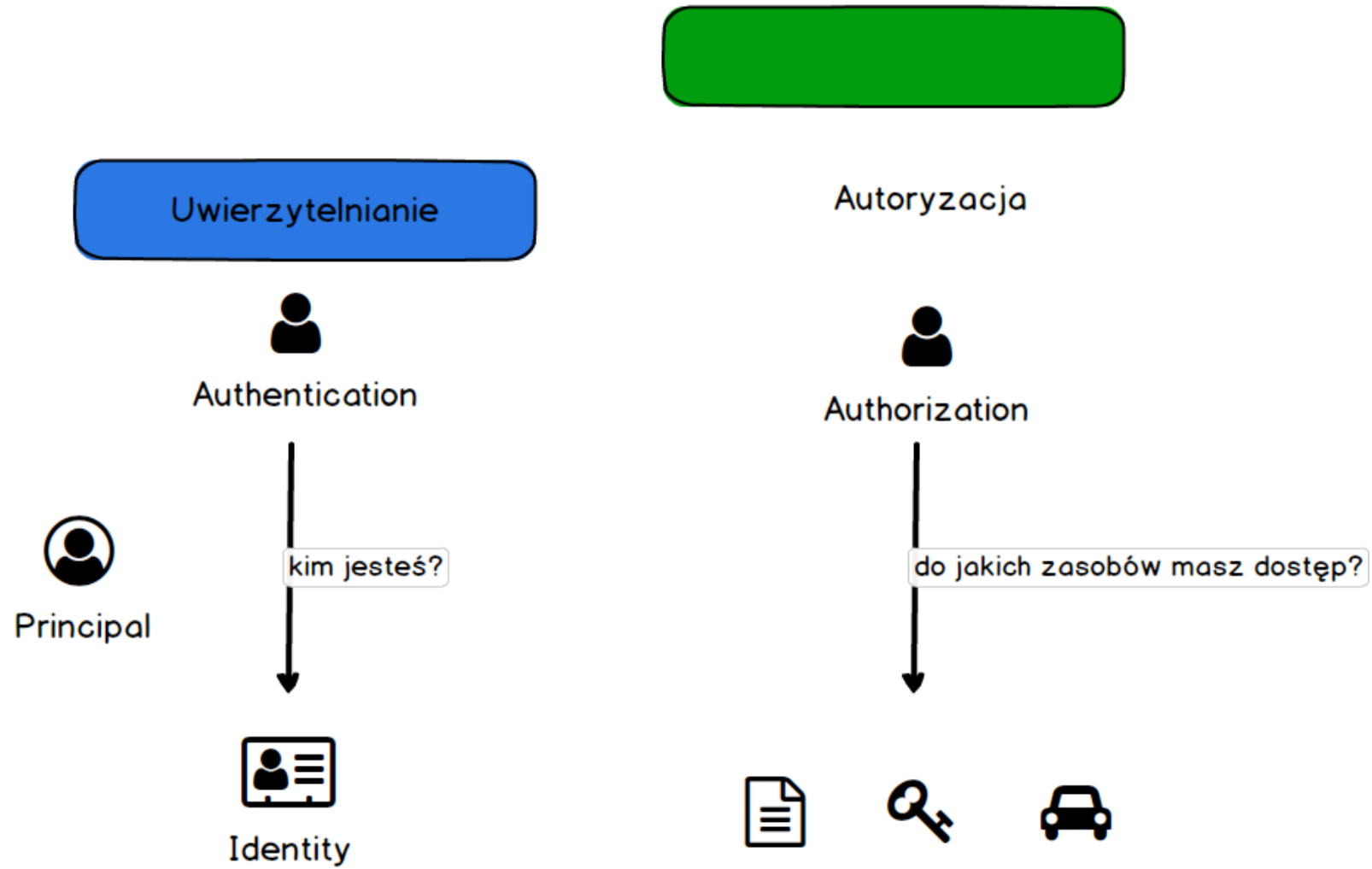
IIdentity vs IPrincipal

```
┌─────────────────────────────────┐        ┌─────────────────────────────────┐
│            IIdentity            │        │            IPrincipal           │
├─────────────────────────────────┤        ├─────────────────────────────────┤
│ AuthenticationType { get; }  ◄┄┄┄┄┄┄┄┄┄┄┄ Identity { get; }                │
│                                 │        │                                 │
│ IsAuthenticated  { get; }       │        │ bool IsInRole(role)             │
│                                 │        │                                 │
│ Name { get; }                   │        │                                 │
└─────────────────────────────────┘        └─────────────────────────────────┘
              ▲                                          ▲
              │                                          │
┌─────────────────────────────────┐        ┌─────────────────────────────────┐
│          ClaimsIdentity         │        │         ClaimsPrincipal         │
├─────────────────────────────────┤        ├─────────────────────────────────┤
│ Claim[] Claims { get; }         │        │ Claim[] Claims { get; }         │
│                                 │        │                                 │
└─────────────────────────────────┘        └─────────────────────────────────┘
```
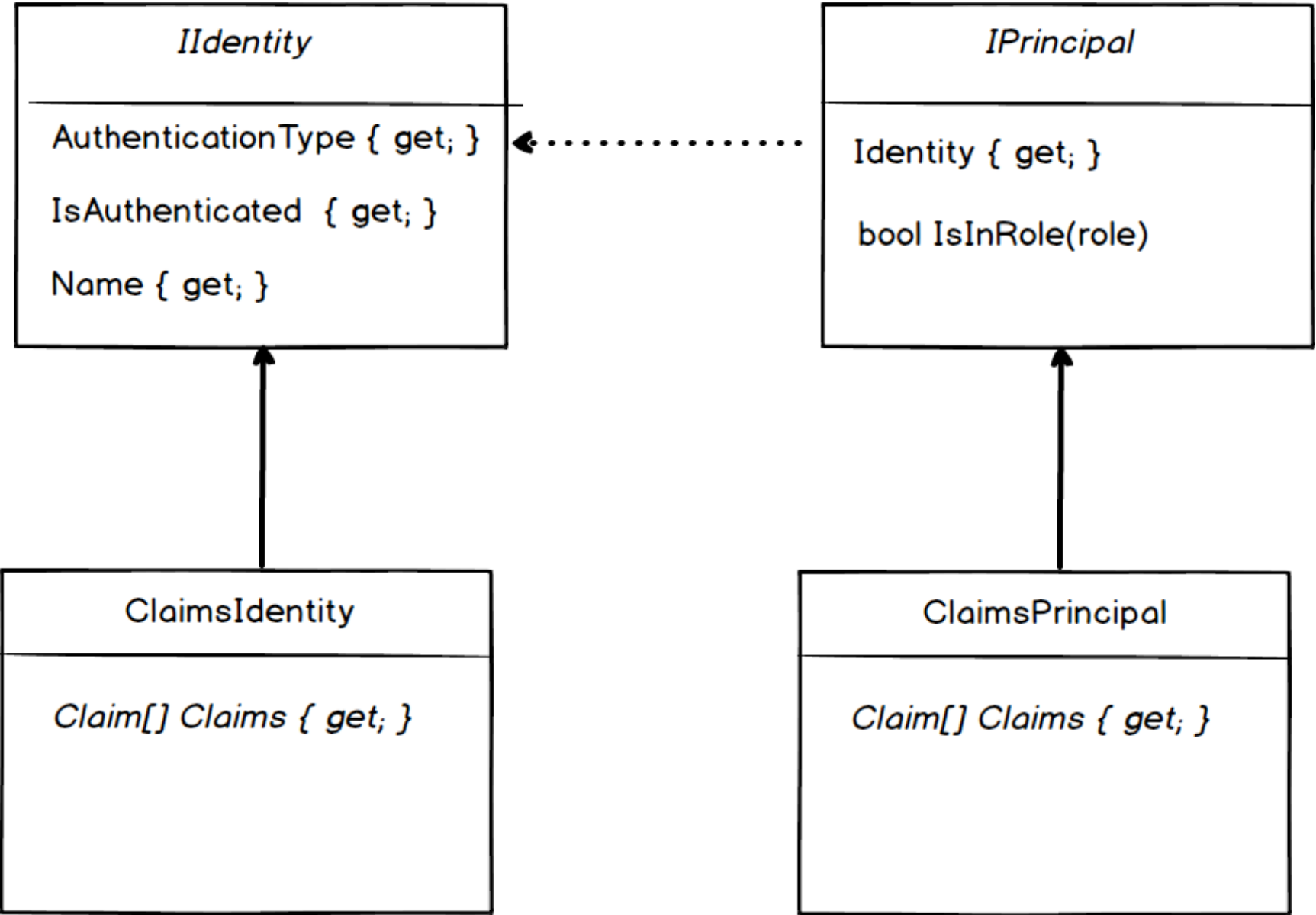
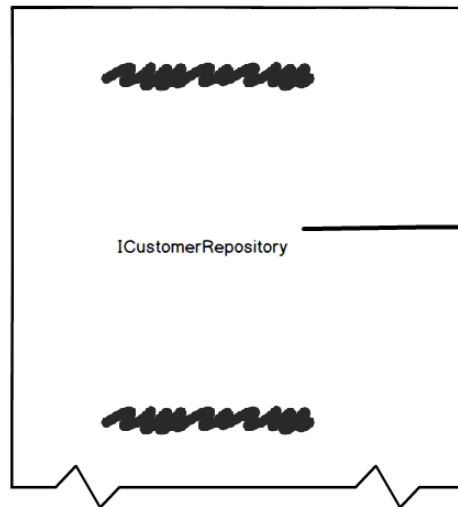*ICustomerRepository*
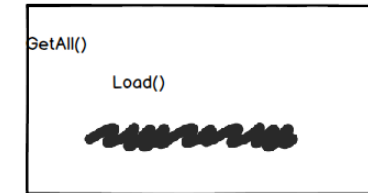
GetAll()

@Page "/customers"
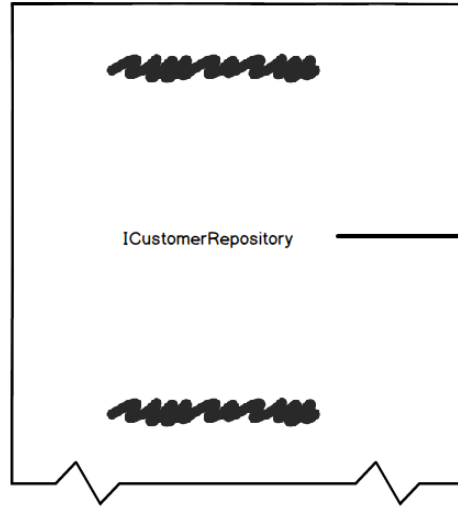
ICustomerRepository

DbCustomerRepository

GetAll()

SQL

SQL

FileCustomerRepository

GetAll()

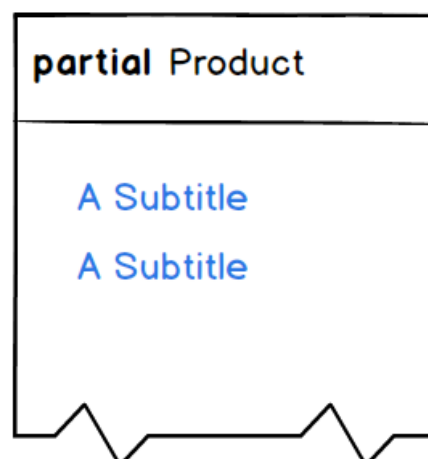Load()

System.IO

customers.csv

@Page "/customer"

ICustomerRepository

product.cs

partial Product : Base

A Subtitle

A Subtitle

A Subtitle

product.g.cs

partial Product

A Subtitle

A Subtitle

partial Product : Base

A Subtitle

A Subtitle

A Subtitle

A Subtitle

A Subtitle
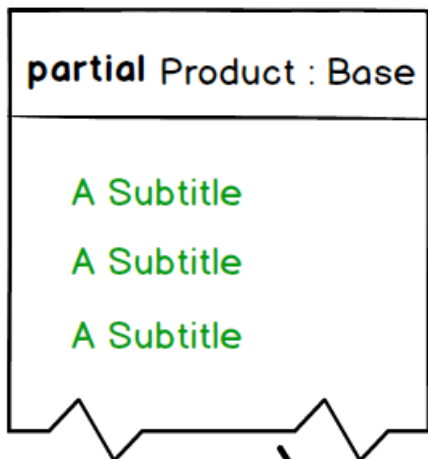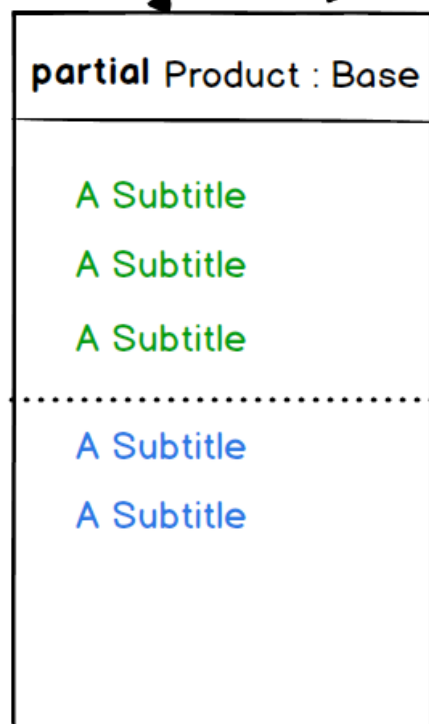
Signal-R
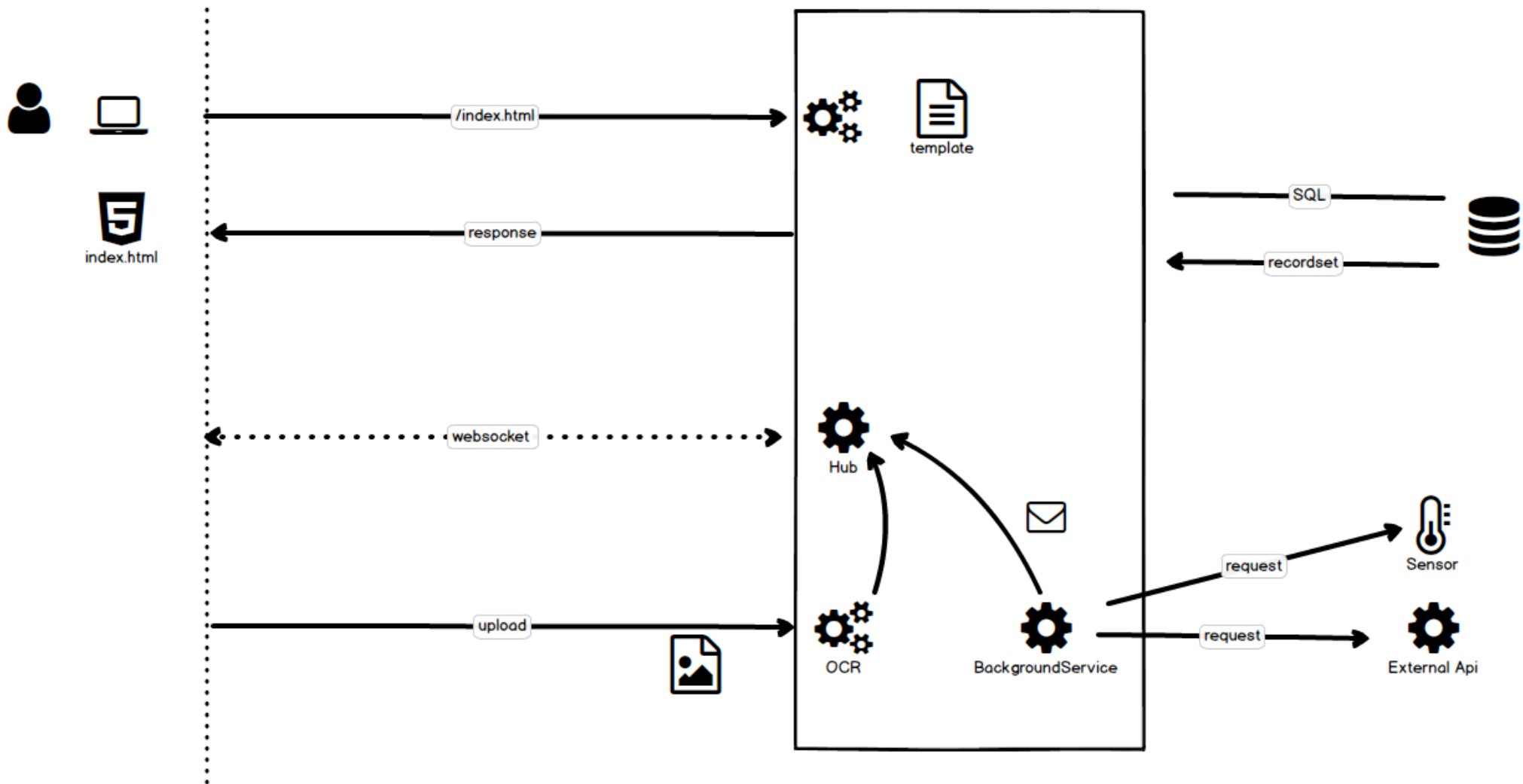
Blazor → script.js

Blazor ← script.js

Blazor Web Assembly (Blazor Client-Side)

localhost:5000/index.html

response

HTML

app.wasm

GET localhost:5000/api/customers

JSON

Gateway.Api

YARP

Ocelot

localhost:5000/index.html

GET localhost:5001/api/customers

Api

Api

C# -> wasm

HTML + @Razor

app.wasm

SQL

recordset
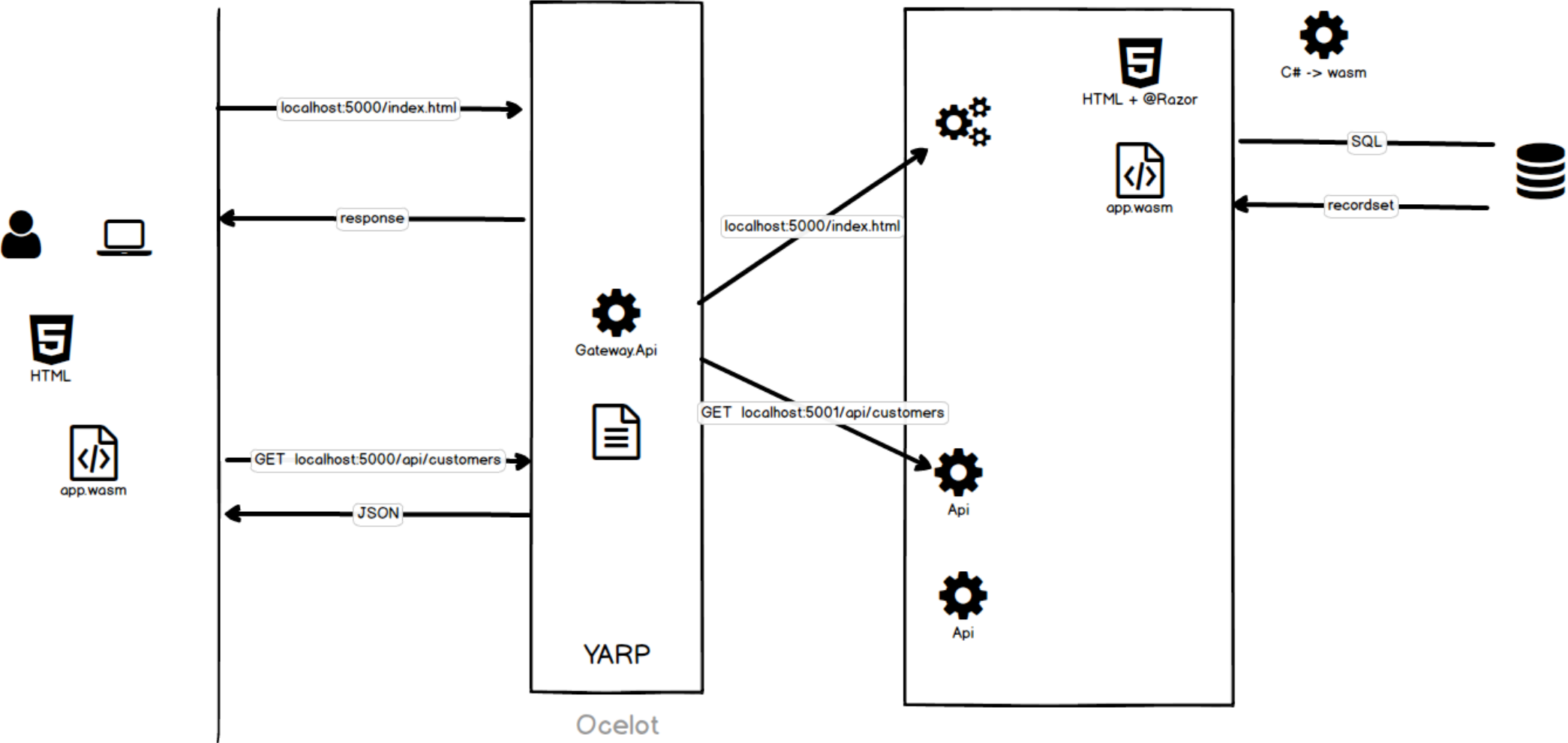
Zadania

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