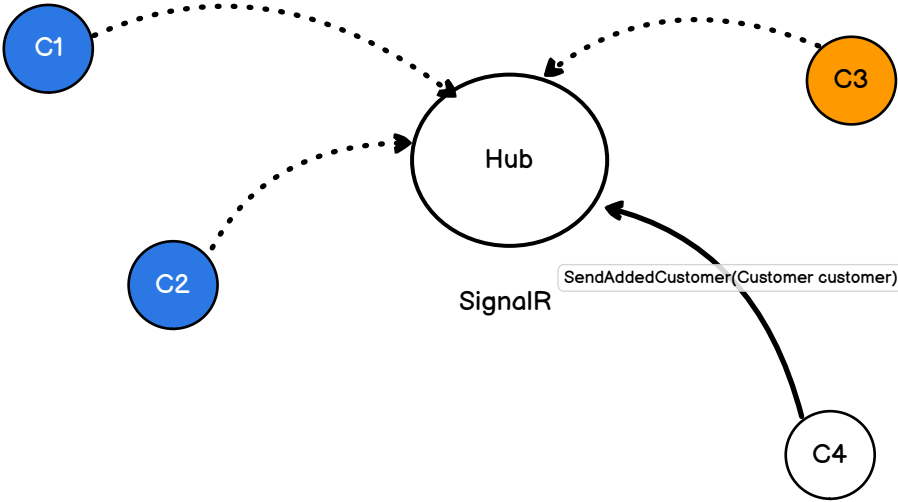


Documents

Mediator

ReceiverConsoleClient



SignalR

SenderConsoleClient

gRPC



Protocol Buffers

tracking.proto

Client

Server

POST GraphQL

GET api/users

GetUsers(string city)

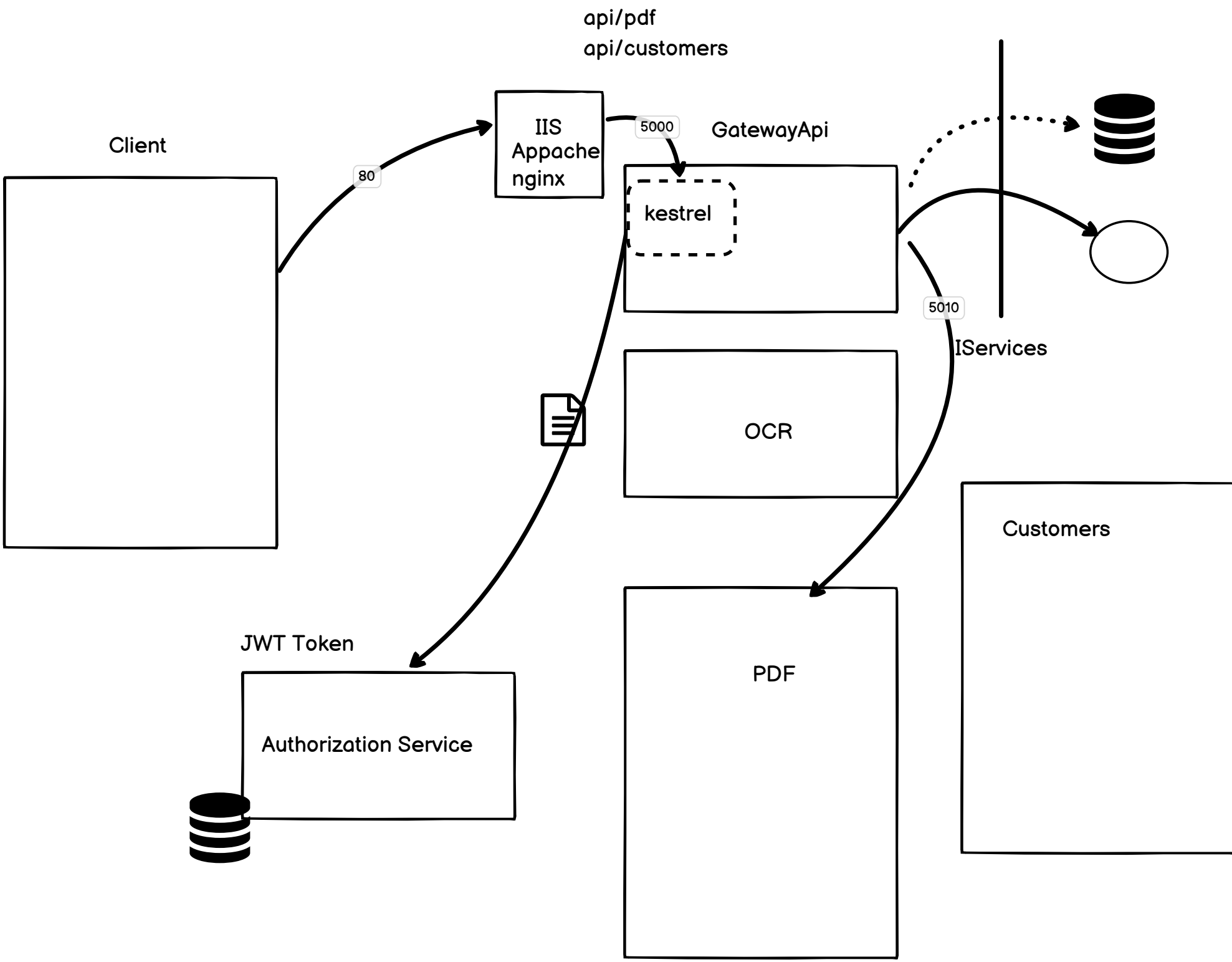
stub

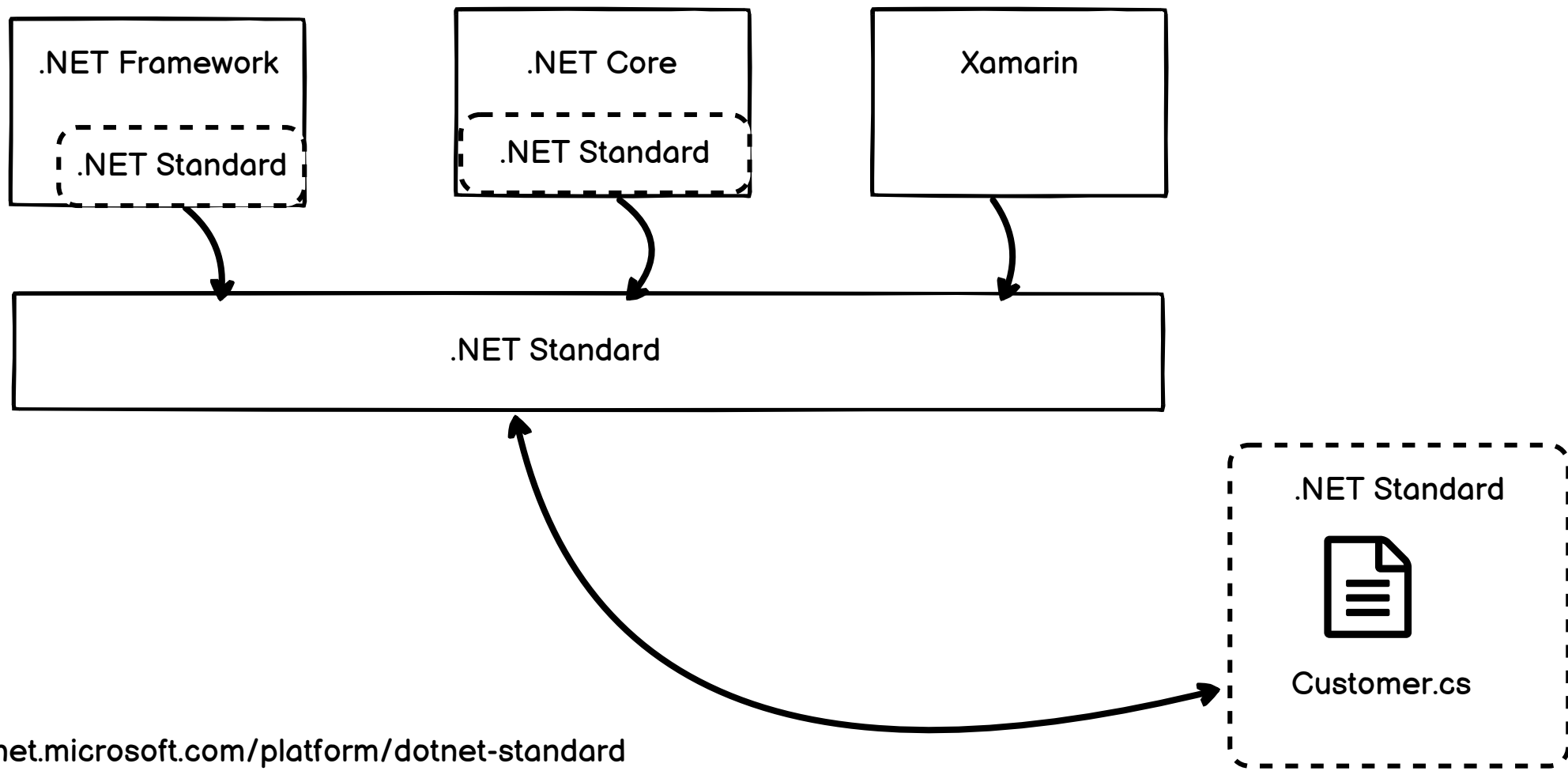
stub

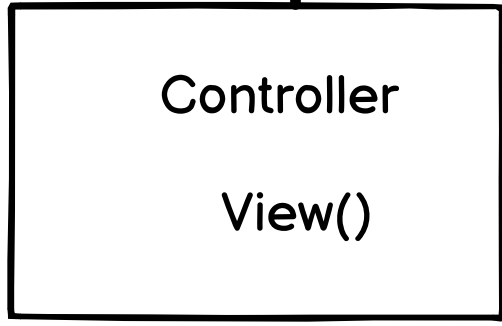
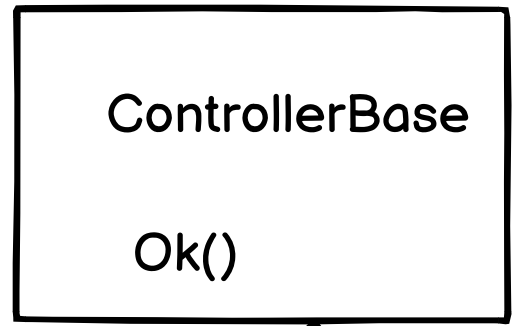
http/2

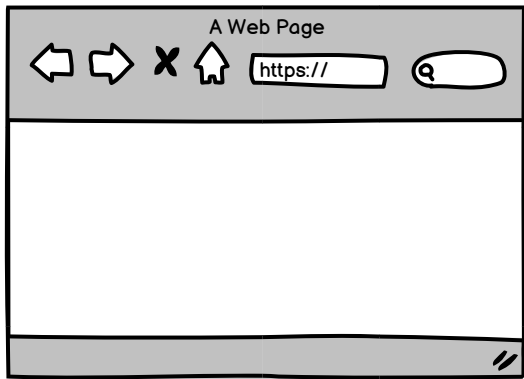
html





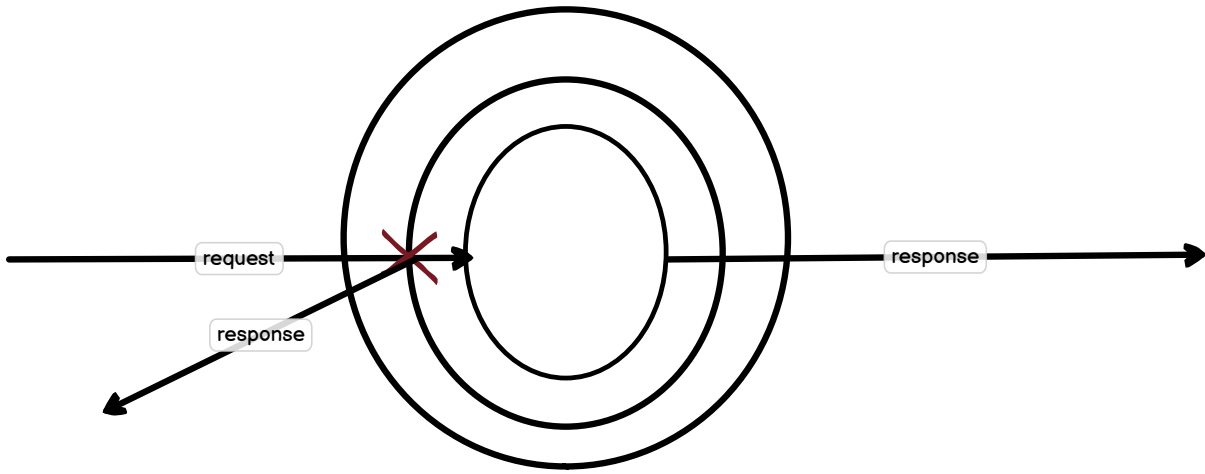
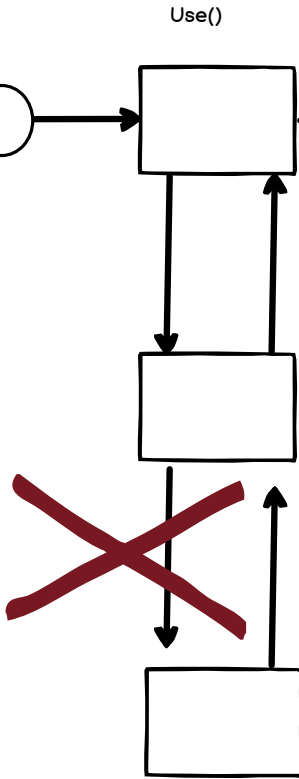






GET api/customers HTTP/1.1
host: localhost
accept: application/json
...

POST api/customers HTTP/1.1
host: localhost
content-type: application/json
accept: application/json
{json}



OWIN = Open Web Server Interface for .NET

```
GET api/customers HTTP/1.1  
host: localhost  
accept: application/json
```

...

Key	Value
owin.RequestMethod	GET
owin.RequestPath	api/customers
owin.RequestAccept	application/json
owin.ResponseBody	{json}




Func<IDictionary<string, object>, Task>




Serilog - strukturalne logi




Seq - wizualizacja logów



TopShelf - biblioteka do tworzenia usług Windows Service



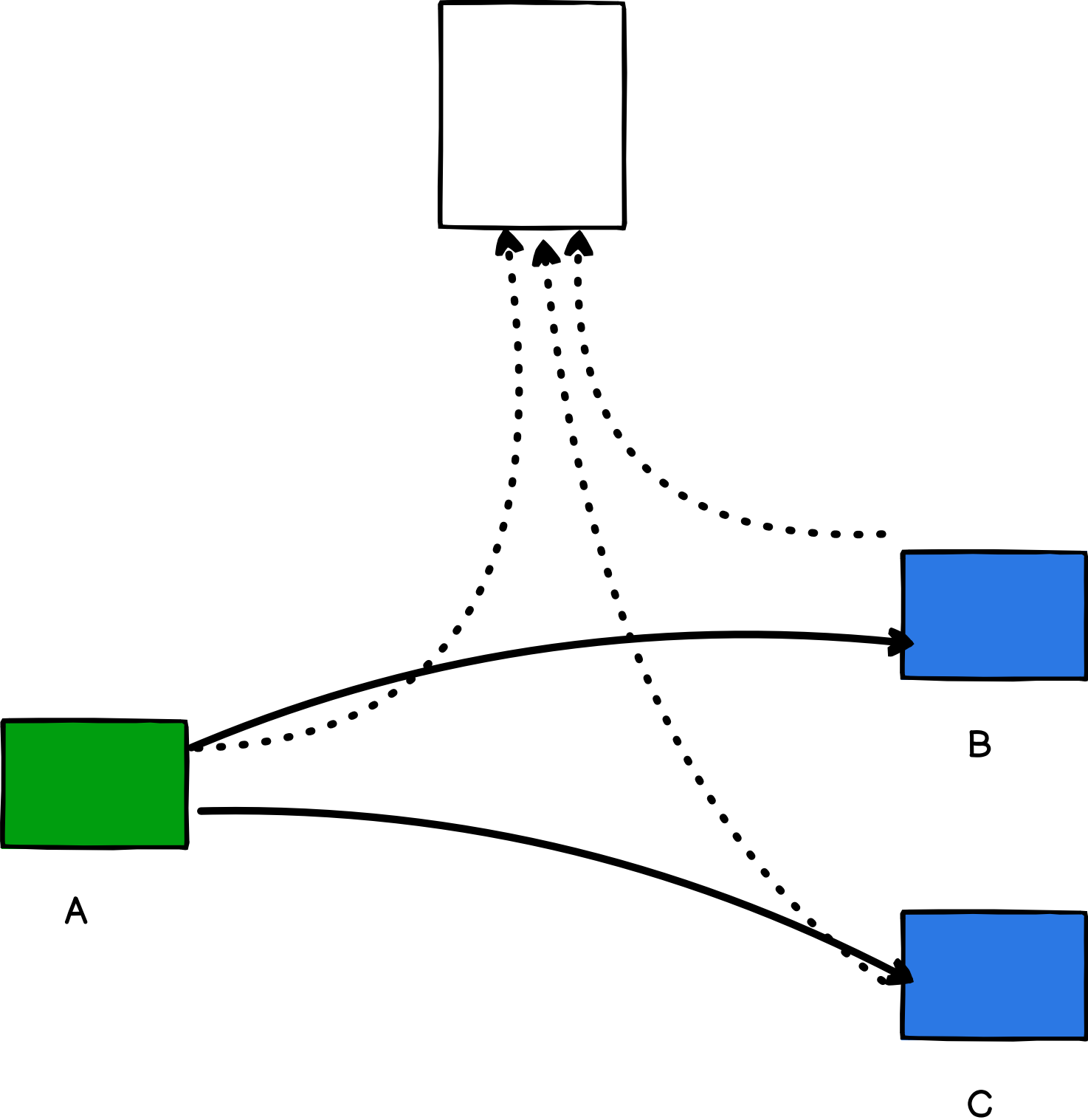
HangFire - serwer do przetwarzania długo-trwających operacji

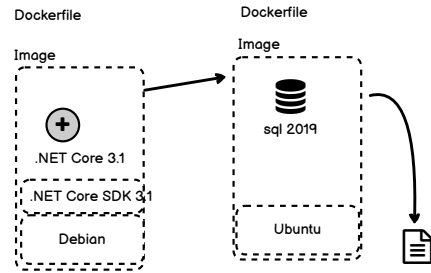
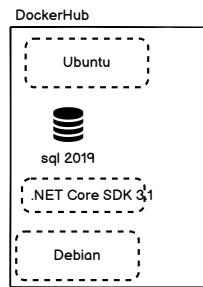


Ocelon - gateway do mikro-usług

<https://github.com/ThreeMammals/Ocelot>

WebRTC

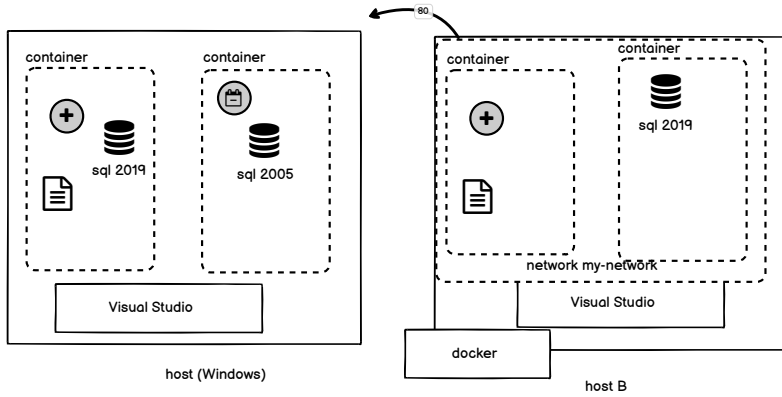




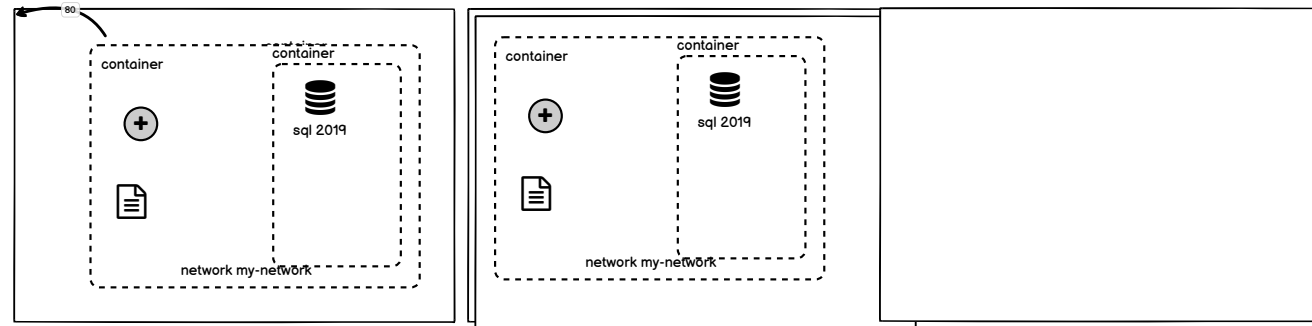
docker-compose up

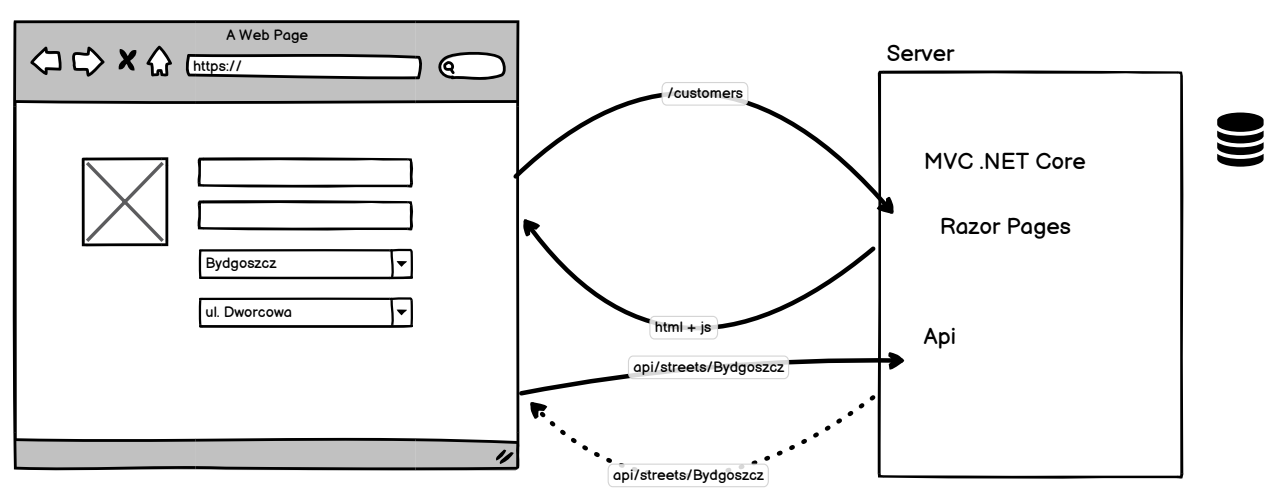
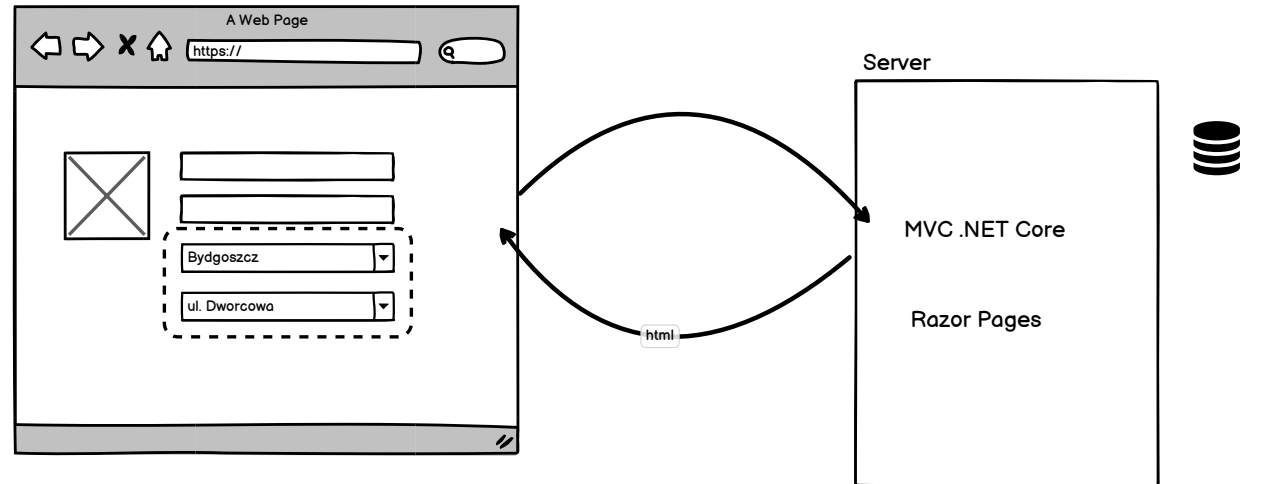
docker-compose down

kubernetes

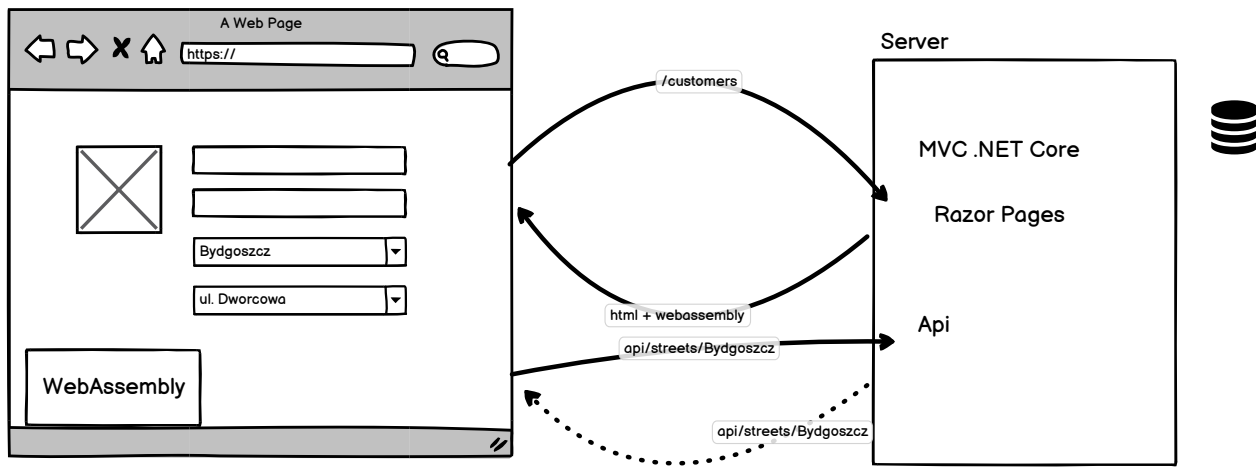


docker-compose.yml



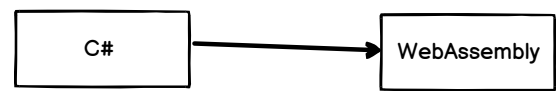


Angular / React



.razor

```
<html>
@foreach()
{
}
</html>
```



Blazor Client-Side Blazor Server-Side