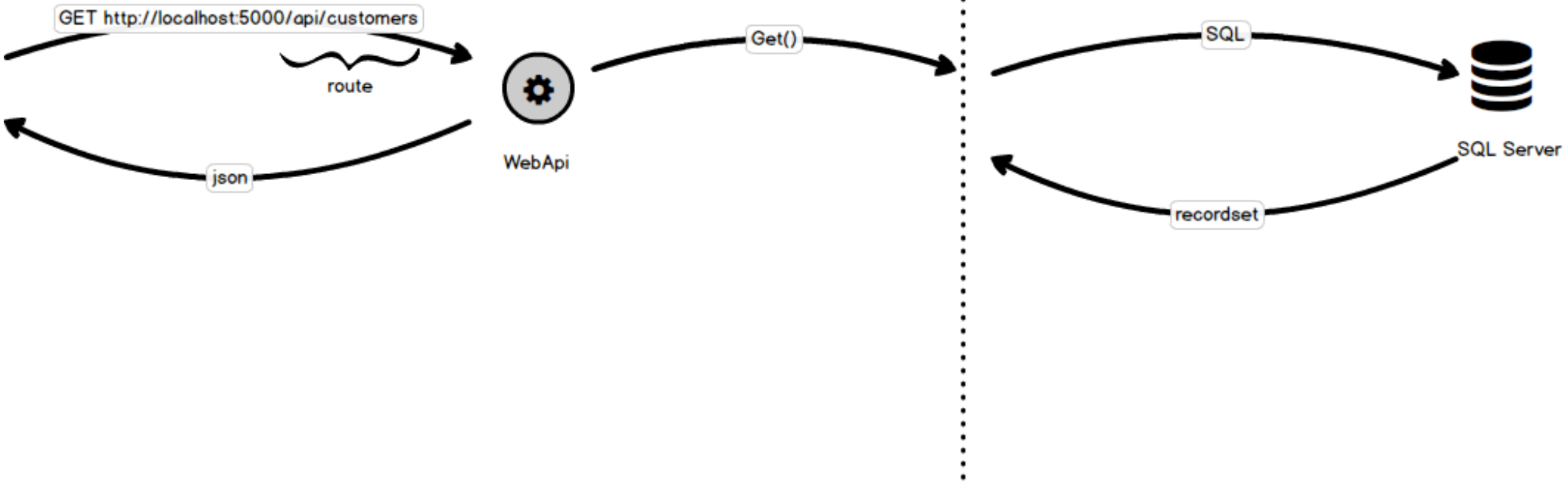


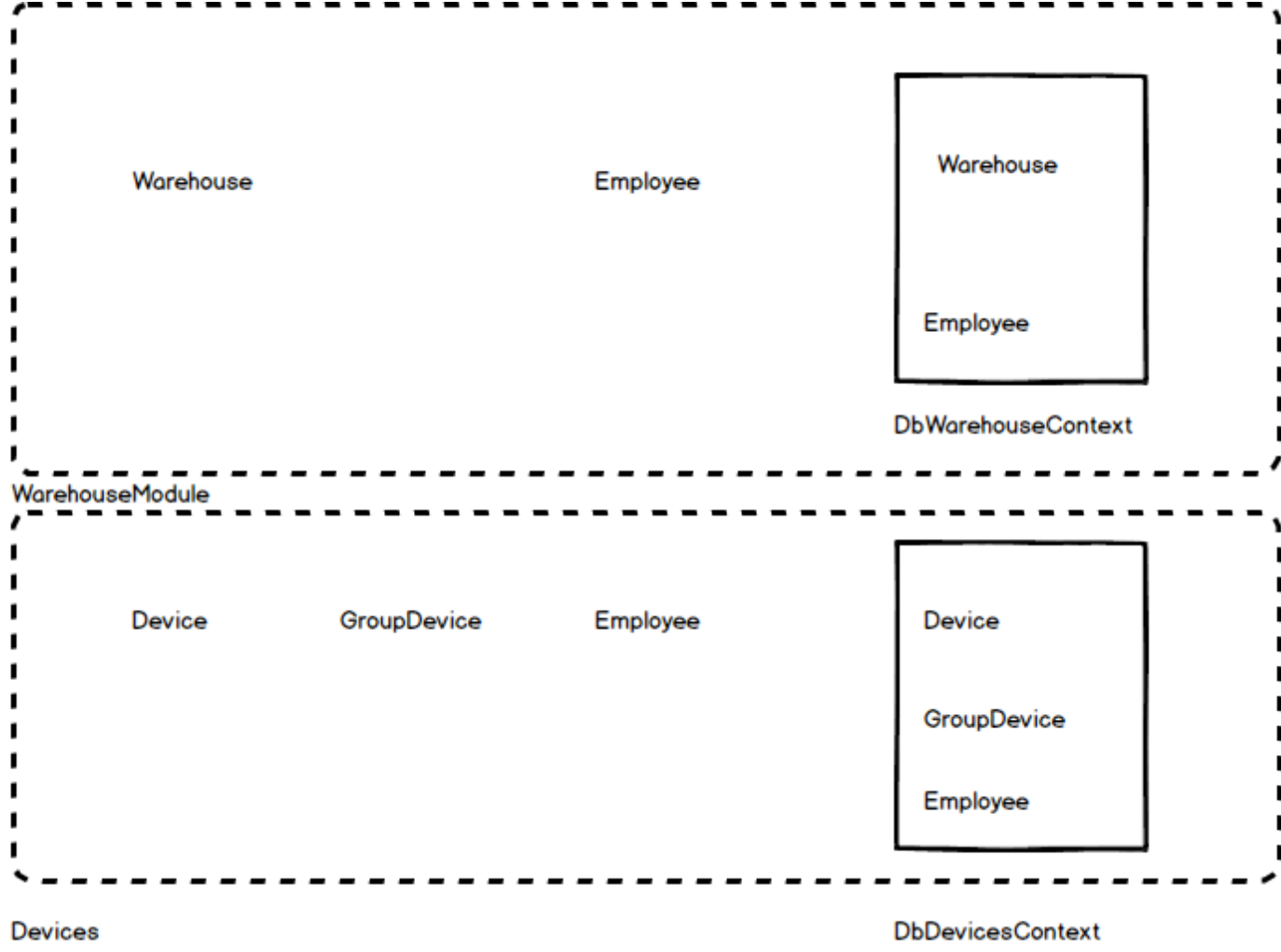
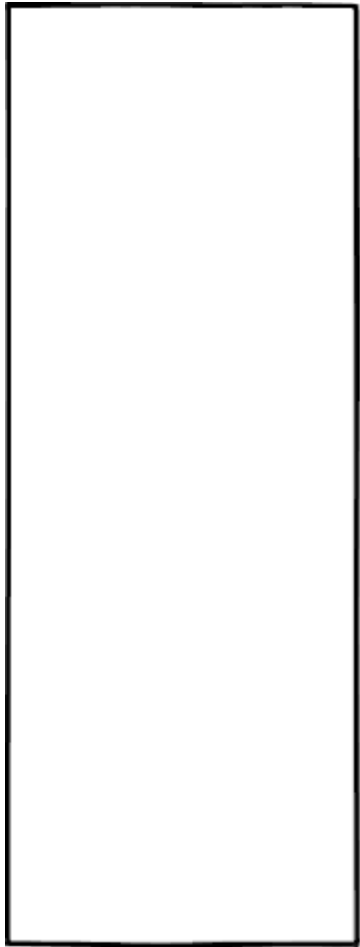
interface ICustomerRepository

class CustomersController



```
builder.Services.Add<ICustomerRepository, FakeCustomerRepository>();
```

Type	Value	Cykl życia
<i>ICustomerRepository</i>	FakeCustomerRepository DbCustomerRepository	Singleton
Faker<Customer>	CustomerFaker	Singleton

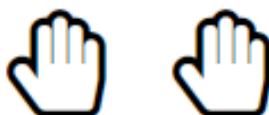


GET api/customers/{id}/orders



CustomersController

OrdersController





View

Controller

Model



MVC

GET api/customers HTTP/1.1

Host: localhost:5001

Accept: application/xml

—

200 OK

Content-Type: application/json

Server: kestrel

POST api/customers HTTP/1.1

Host: localhost:5001

Content-Type: application/json

Accept: application/xml

{ json }

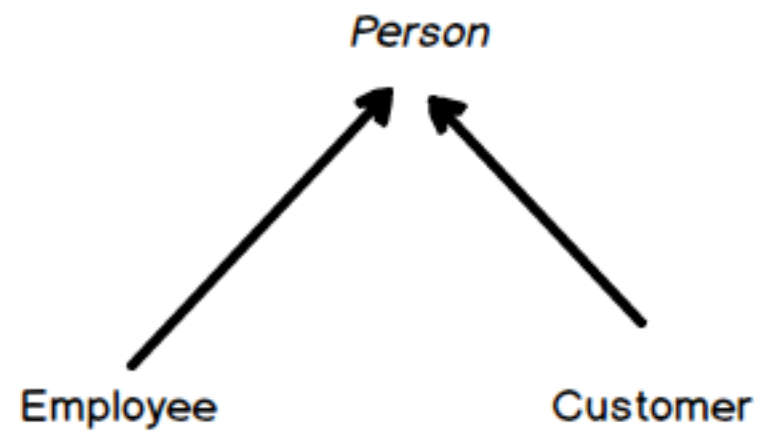
—

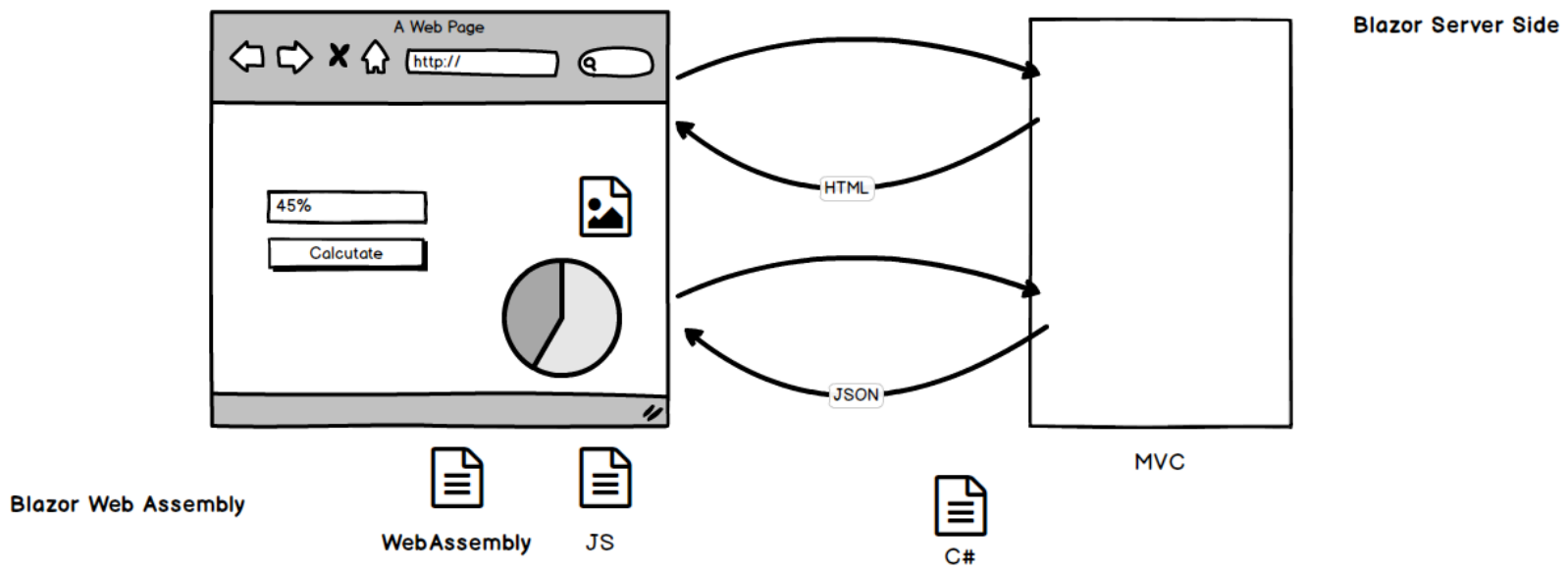
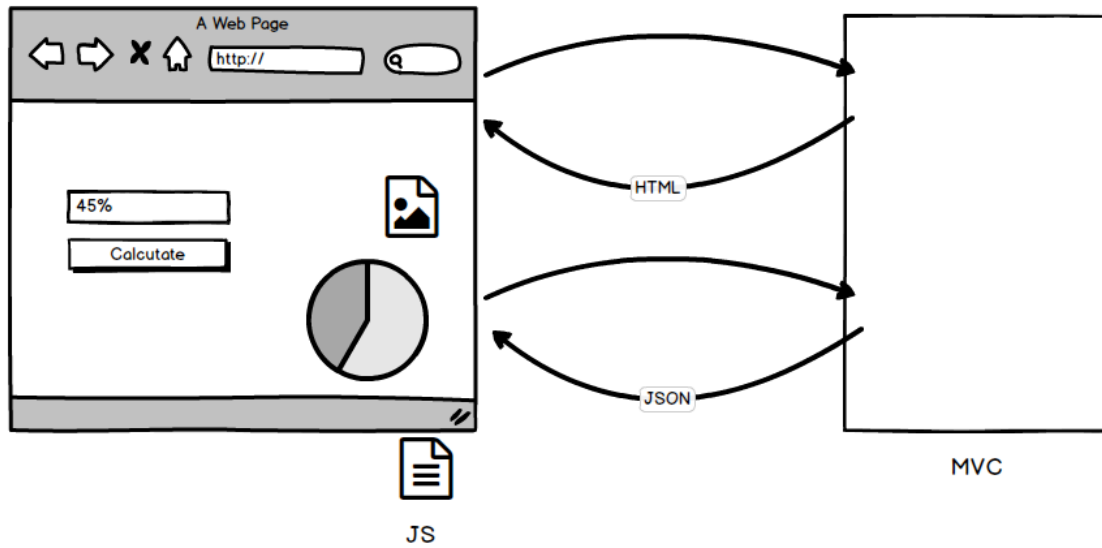
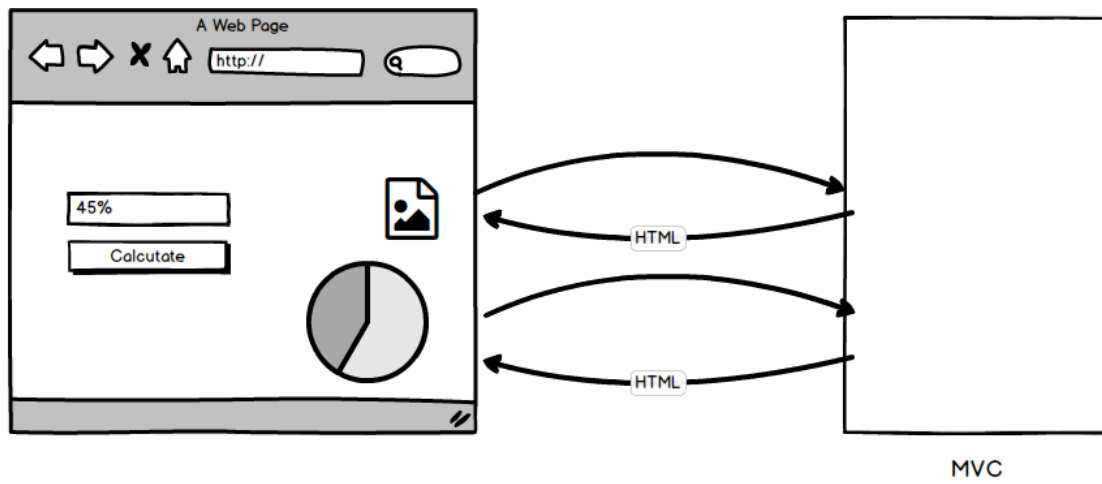
201 Created

Content-Type: application/json

Server: kestrel

Location:





Controller

Fake

interface

Models

GET api/products

GET api/products/{id}

POST api/products

PUT api/products/{id}

DELETE api/products/{id}

GET api/products?pricefrom=10&priceto=20

Product

Id

Name

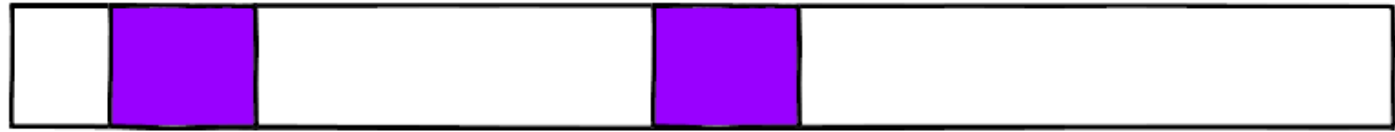
Color

UnitPrice

Size

{ S, M, L, XL }

appsettings.ini



appsettings.xml

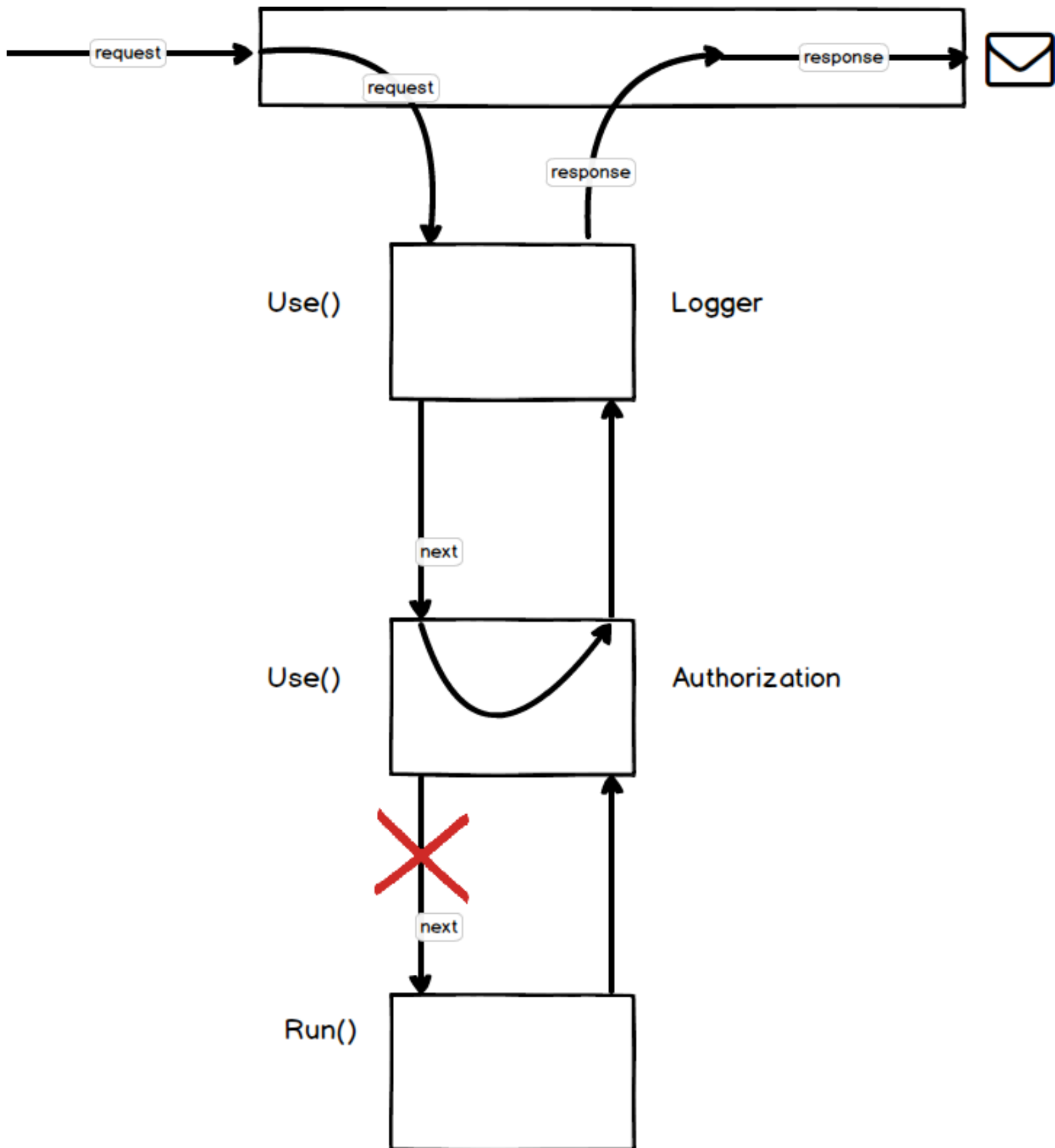


appsettings.json



configuration

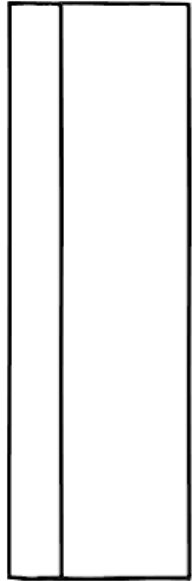
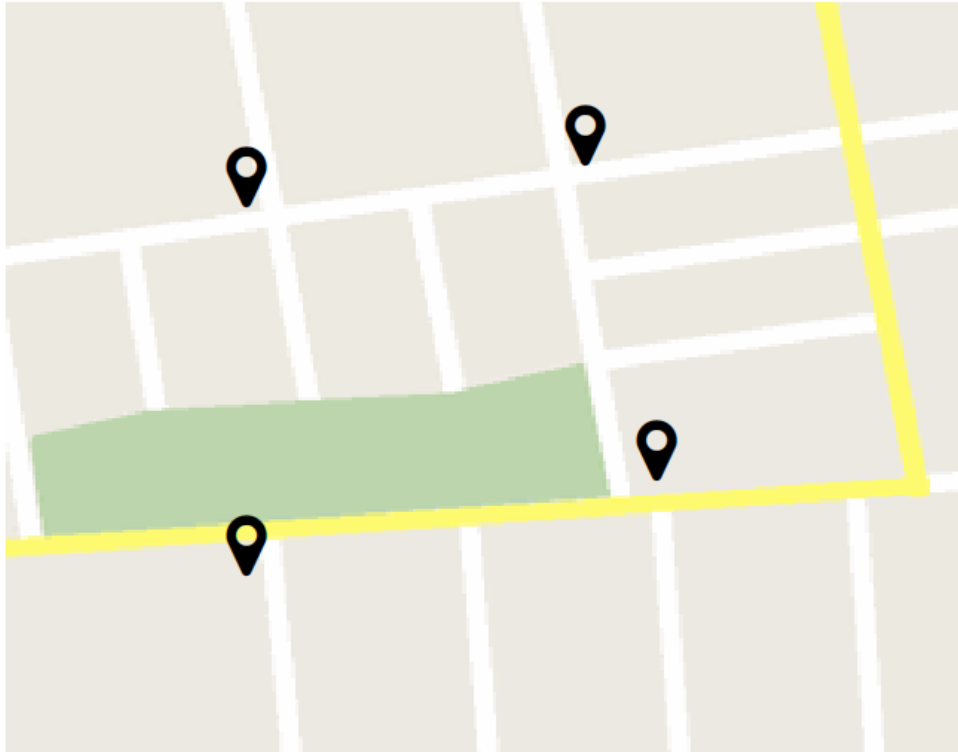




url <http://localhost:5000/api/customers?format=kml>

Accept: kml

Window Name



FileStream

MemoryStream

NetworkStream

Stream

Writer

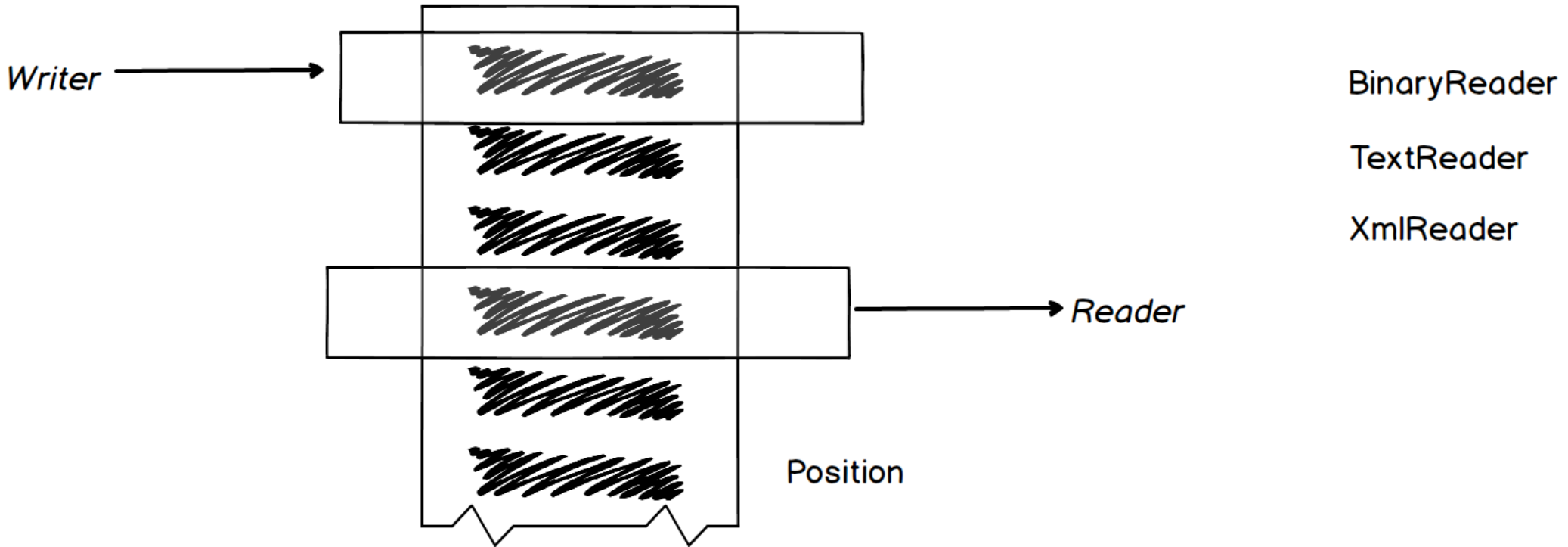
BinaryReader

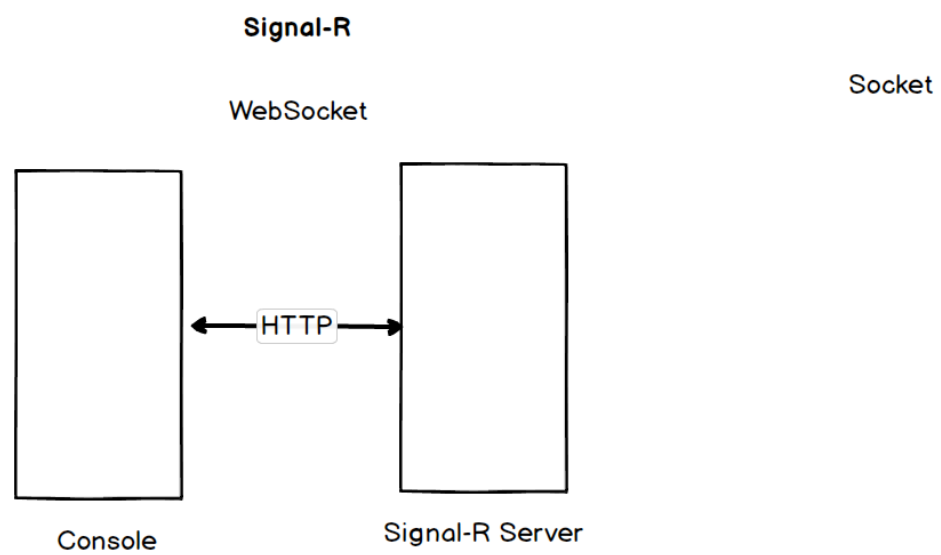
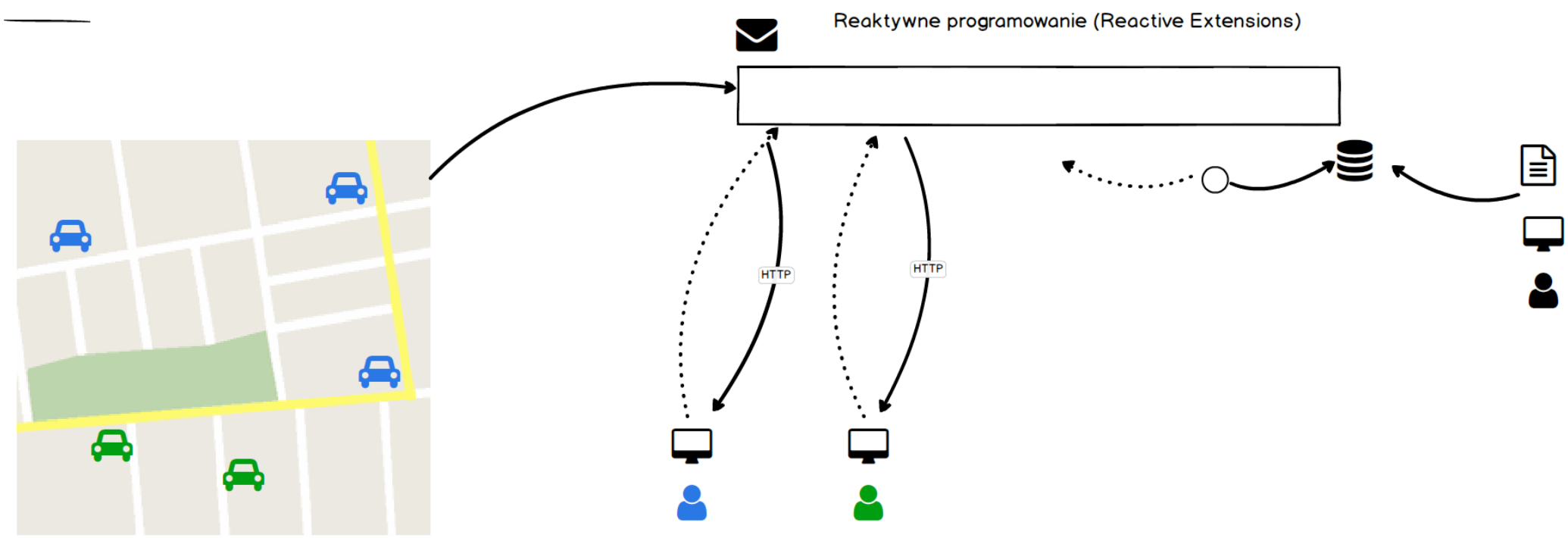
TextReader

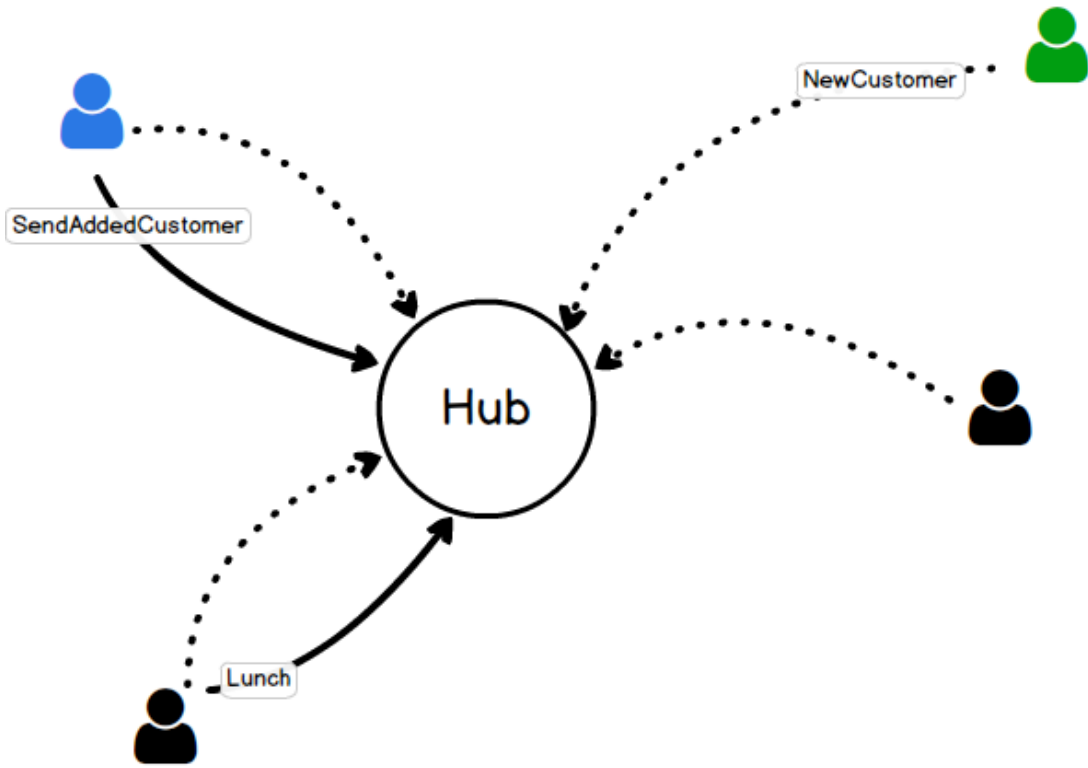
XmlReader

Reader

Position







IEnumerator<T>

Current

bool MoveNext();



IEnumerable<T>

[]

Array

Collection

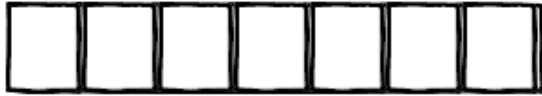
List

for

foreach

Imperatywne programowanie

IEnumerable<T>

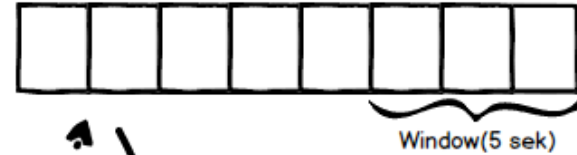


Linq



Reaktywne programowanie (Reactive Extensions)

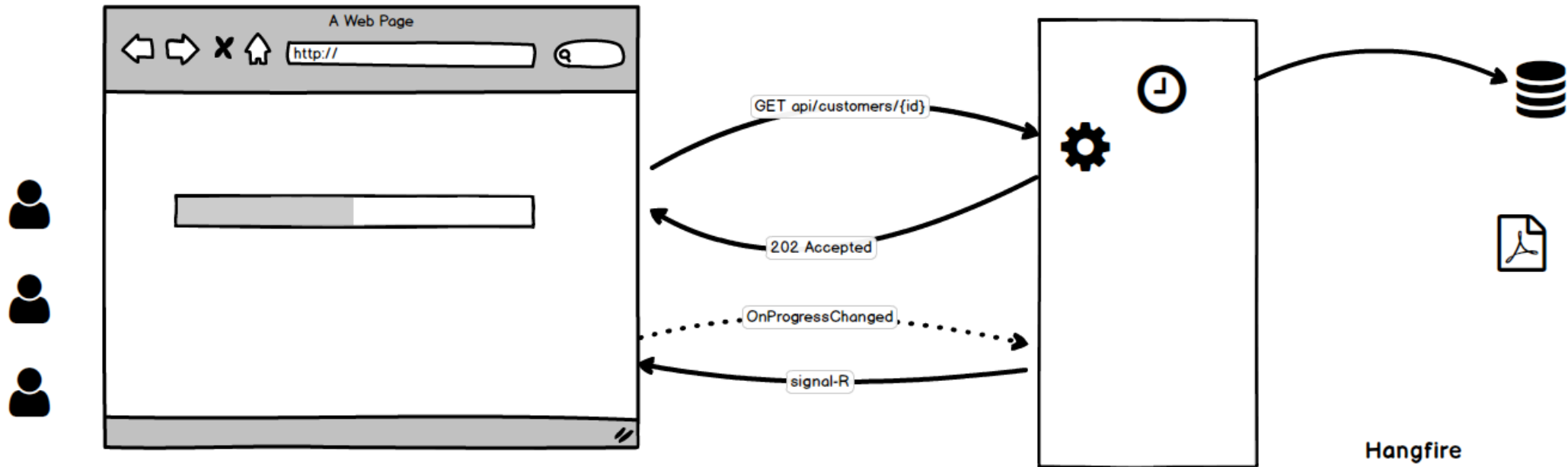
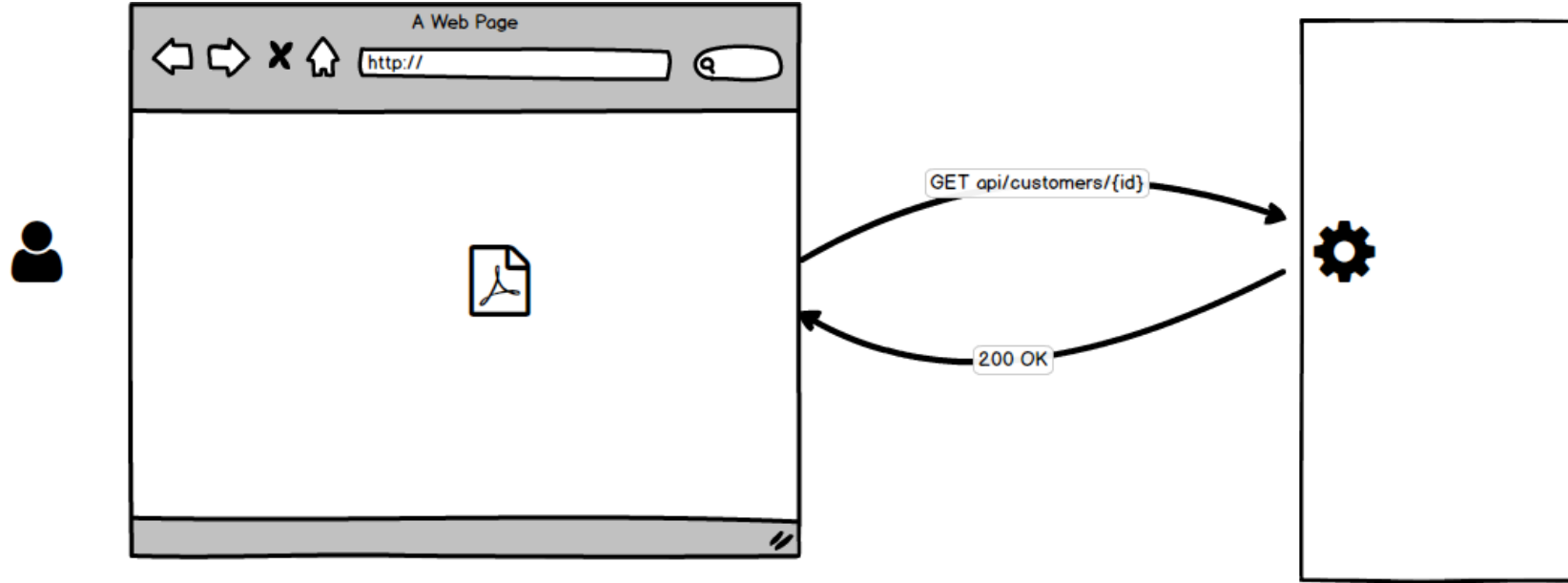
IObservable<T>

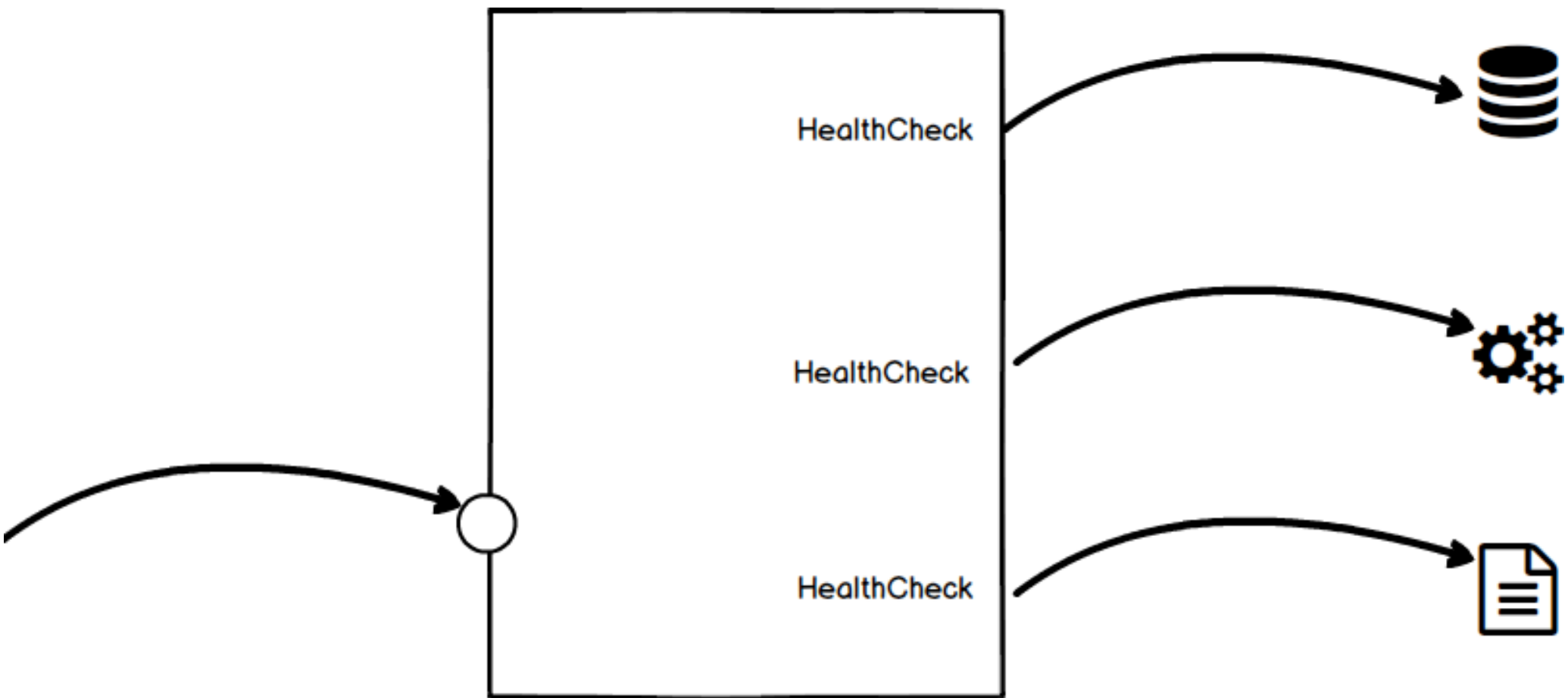


Linq

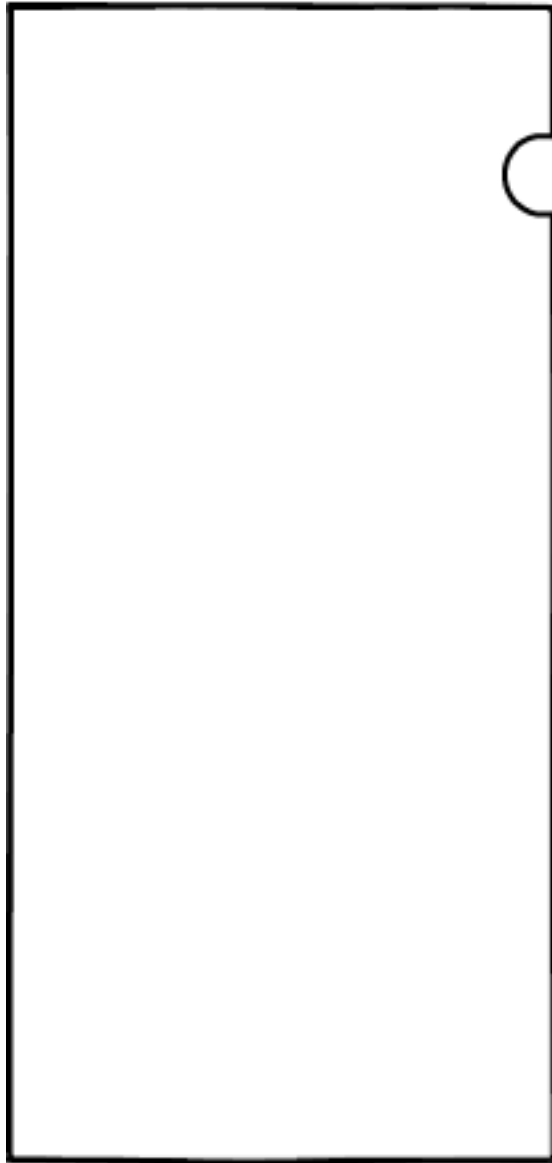


events + Linq = Reactive Extensions

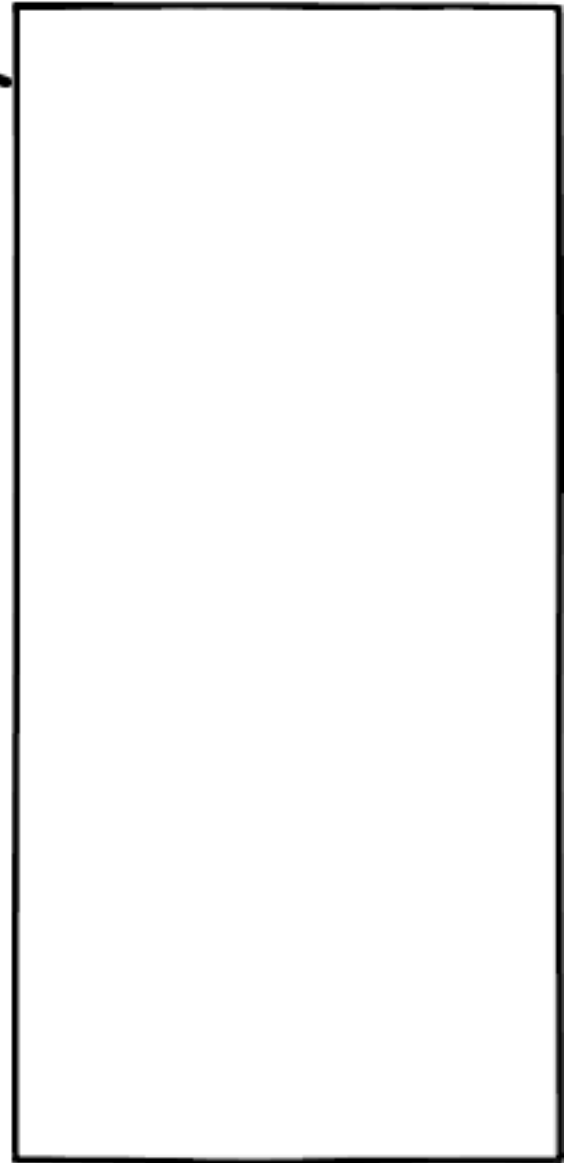




POST api/webhook



A



B

REST API

GET api/order/{id}?action=sent

GET api/customers

GET api/customers/{id}

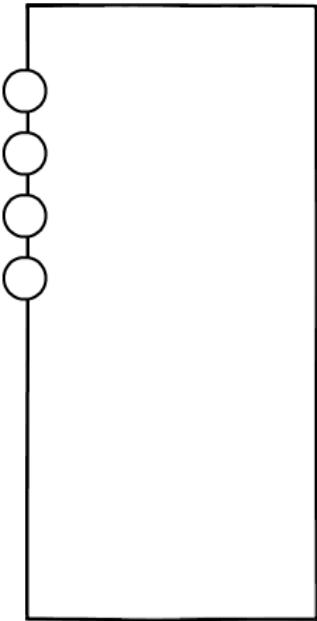
POST api/customers/{id}

GET api/customers/{id}?fields=firstname,lastname

OData

OData - SQL w Url'u ;-)

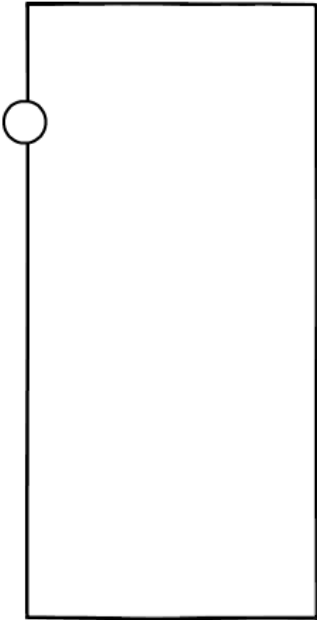
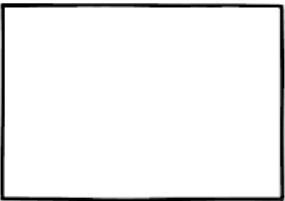
GET api/customers/{id}?fields=firstname,lastname&order=lastname



WebApi

GraphQL

POST /graphql



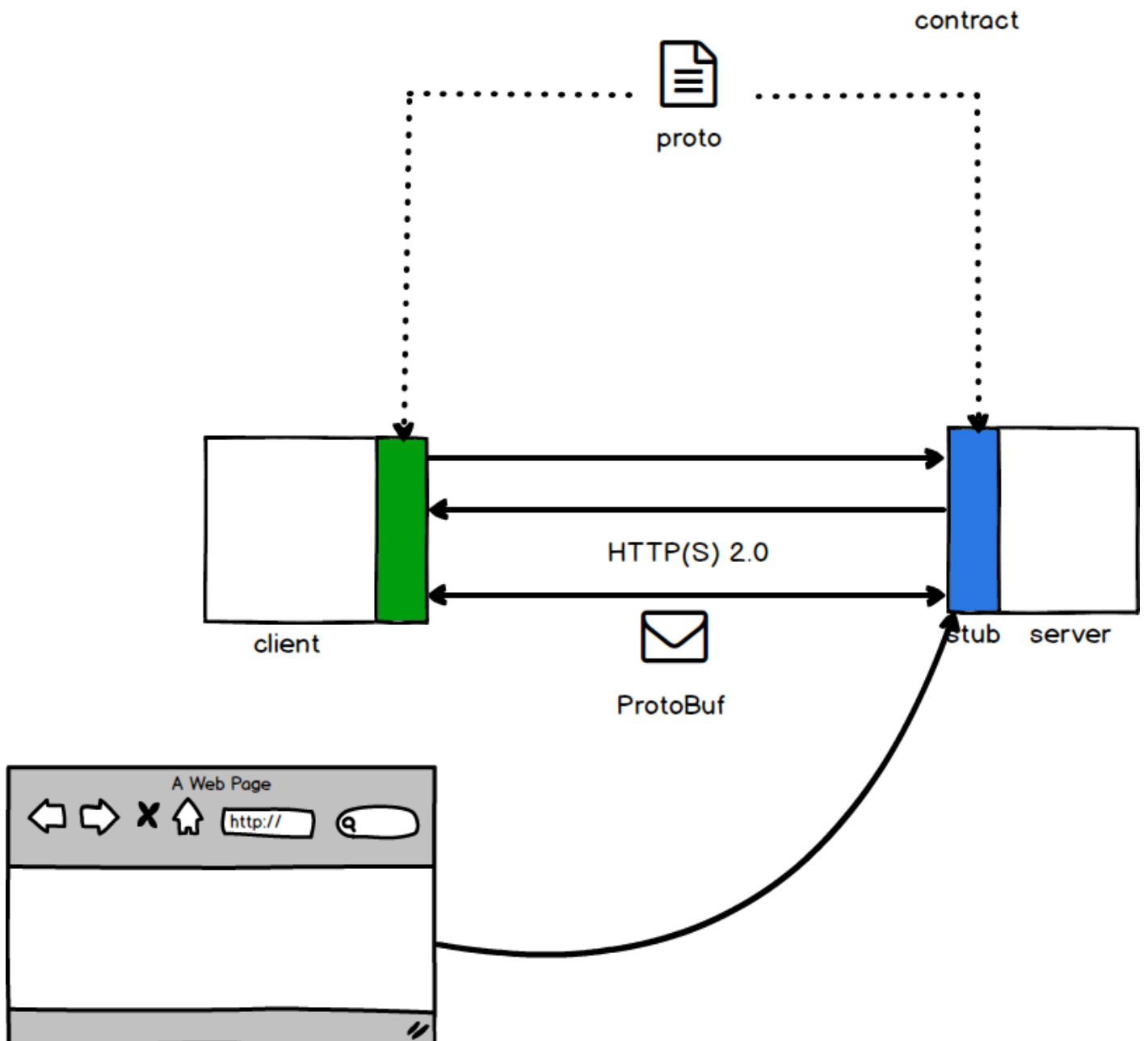
Api

XML-RPC

JSON-RPC

gRPC

ProtoBuf





Uwierzytelnianie (Authentication)

kim jest użytkownik?

Autoryzacja (Authorization)

do czego użytkownik ma dostęp?

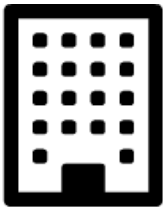
Authenticated

Role

Claim

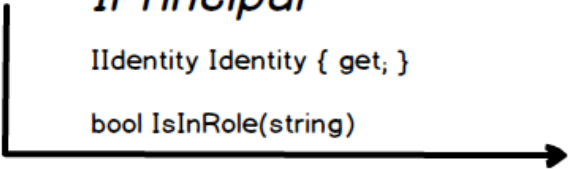
Policy

Resource



IPrincipal

IIdentity Identity { get; }
bool IsInRole(string)



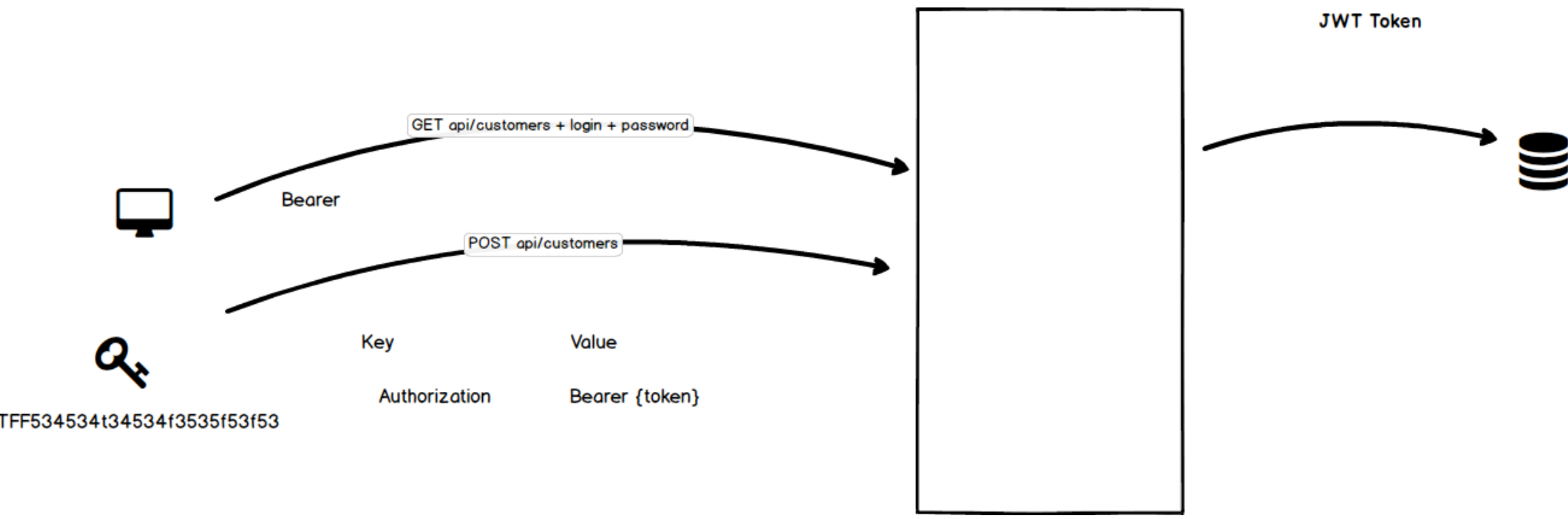
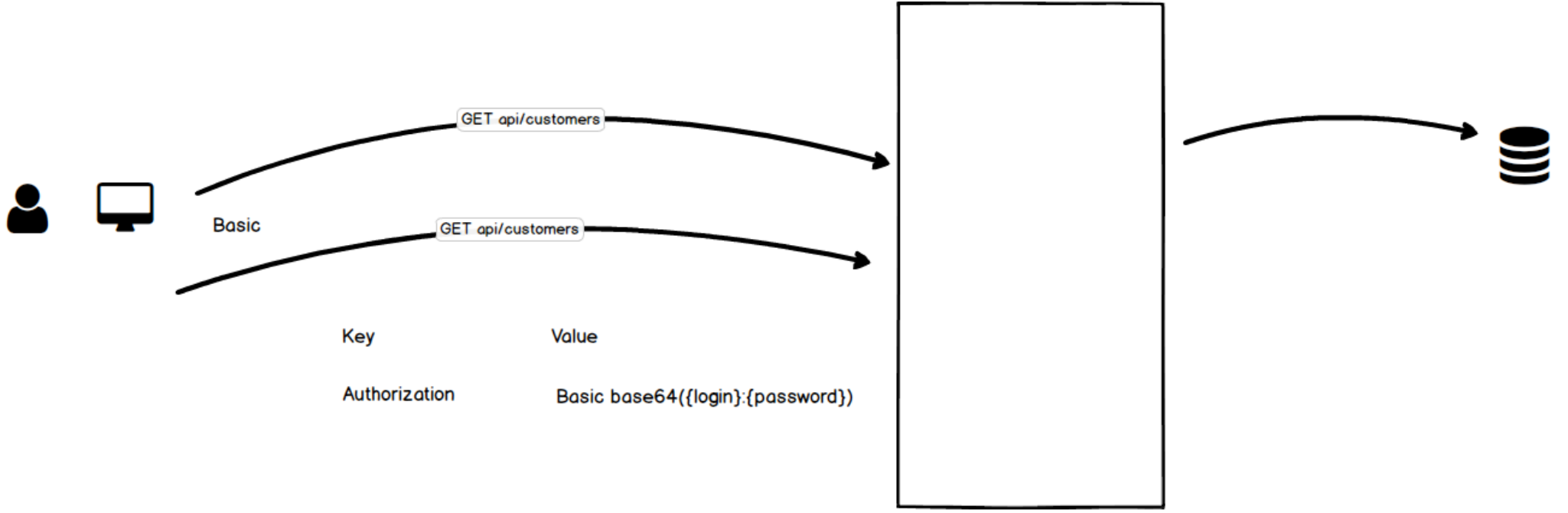
IIdentity

string AuthenticationType { get; }
string Name { get; }
bool IsAuthenticaton { get; }

Type	Value (string)
kat	B
kat	A
termin	bezterminowe
phone	555-001-001
email	marcin.sulecki@sulmar.pl

Claim

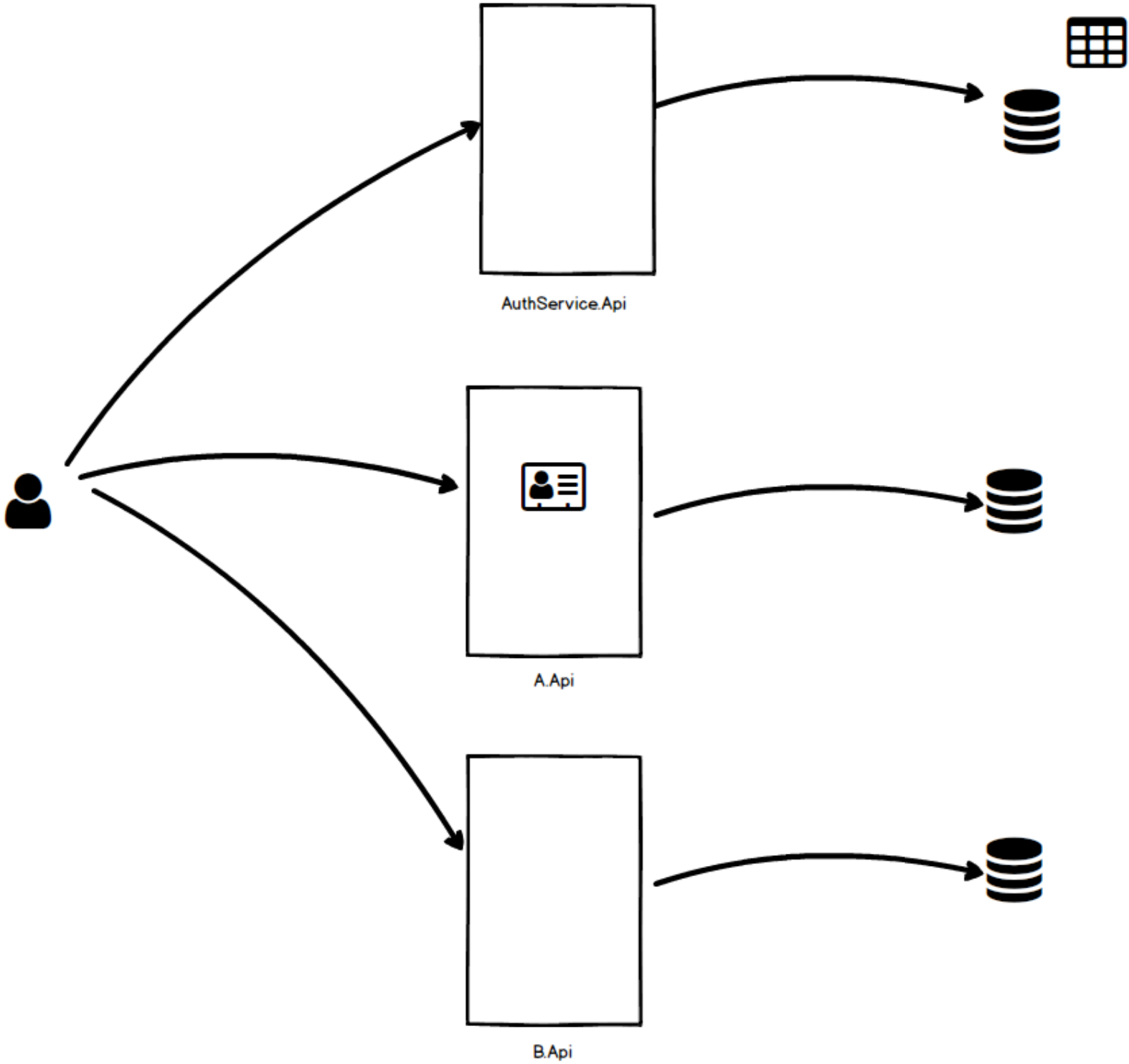
ClaimsIdentity
Claims



[AllowAnonymous]

POST api/tokens/create

login + password



eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c

header

payload

signature

UserManager

RoleManager

IUserStore

IRoleStore

IdentityUser

IdentityRole