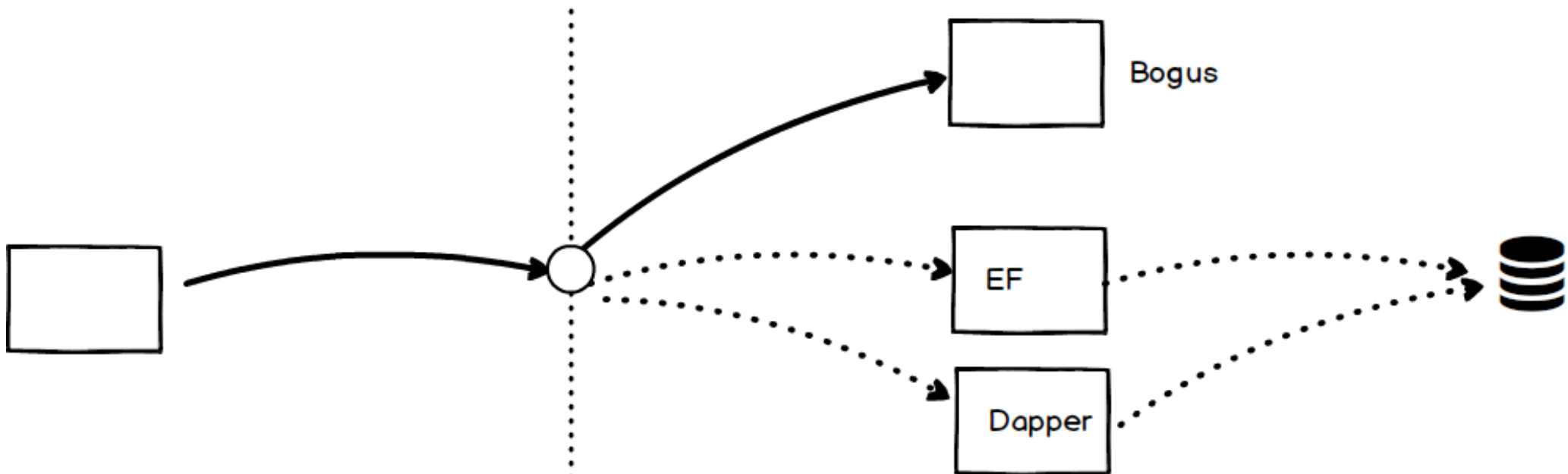


ICustomerRepository



`List<Customer> Get()`

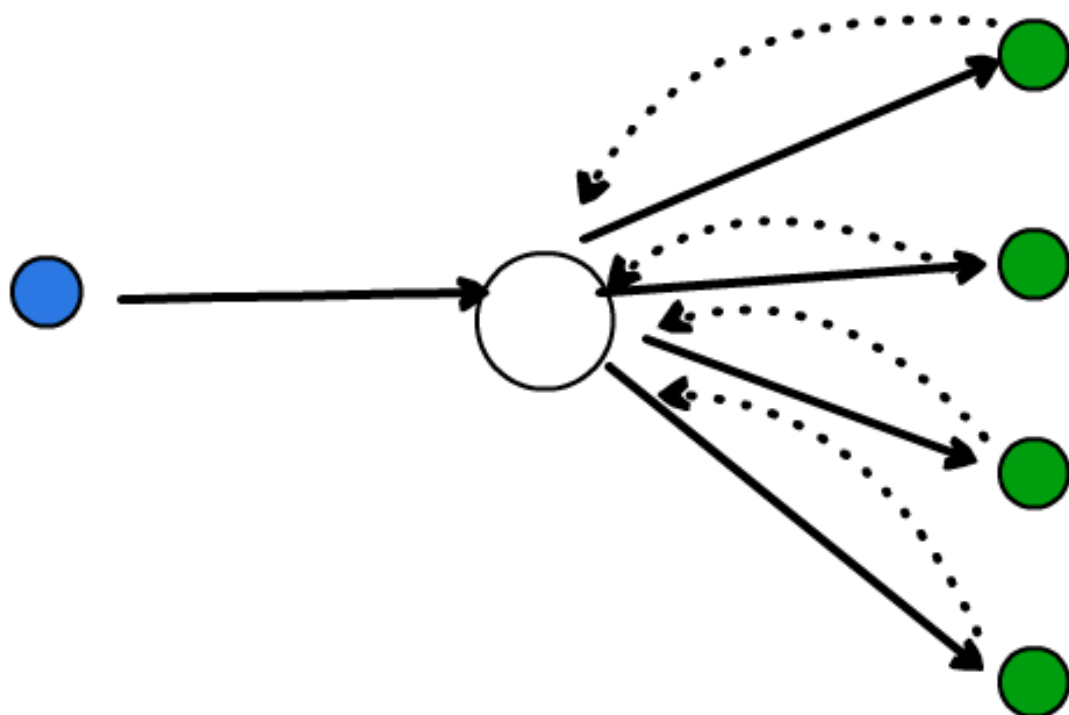
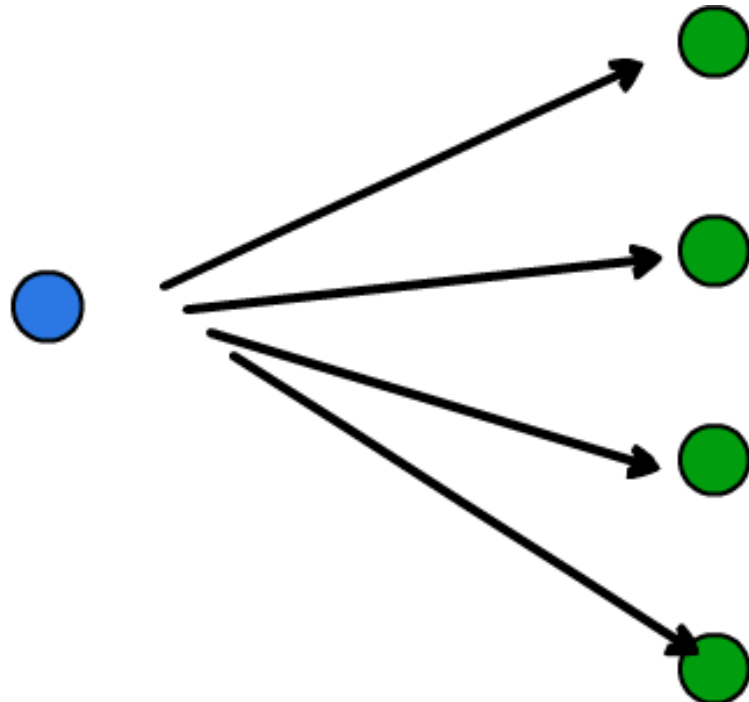
`Customer Get(int id)`

`void Add(Customer customer)`

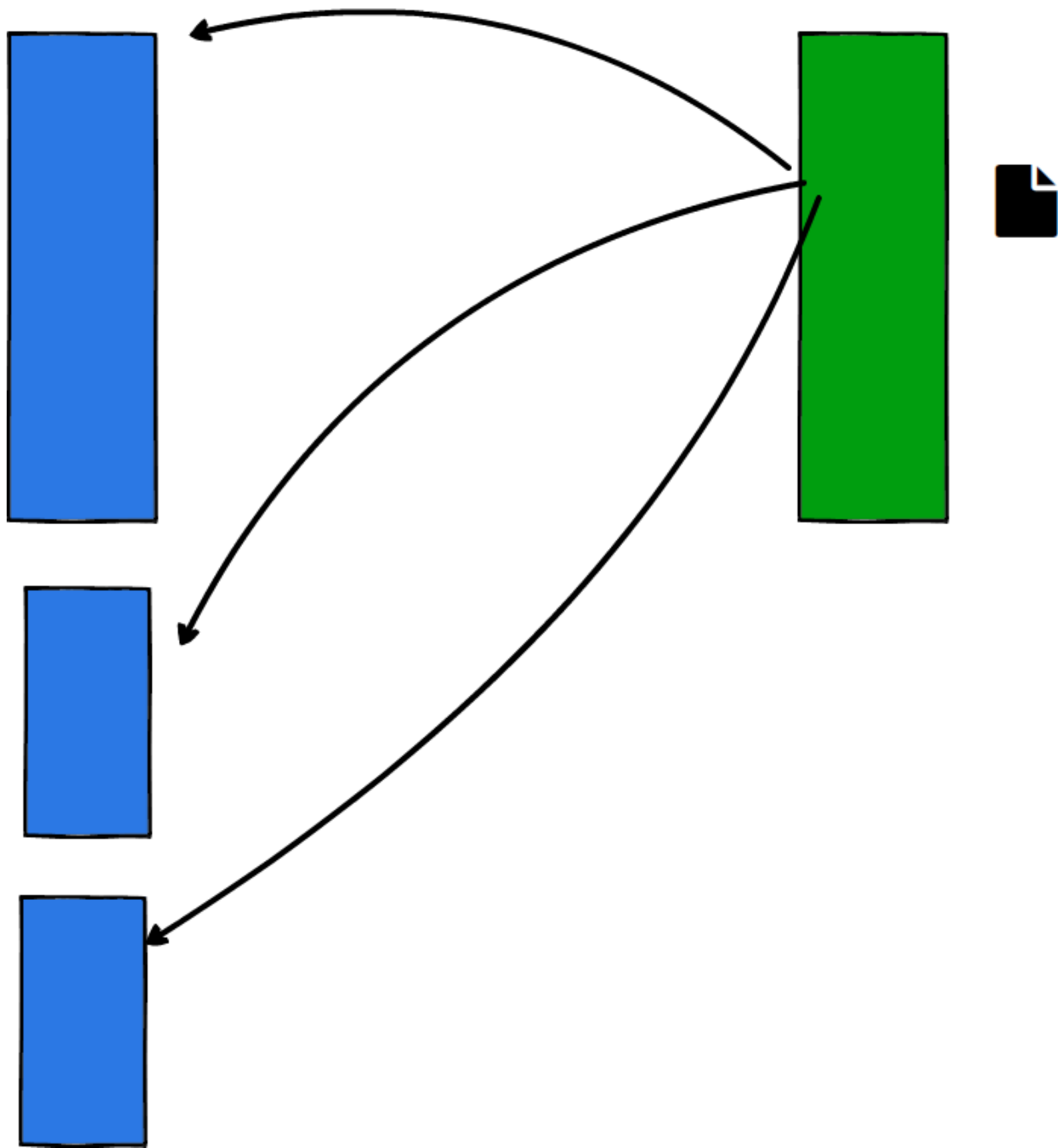
`void Update(Customer customer)`

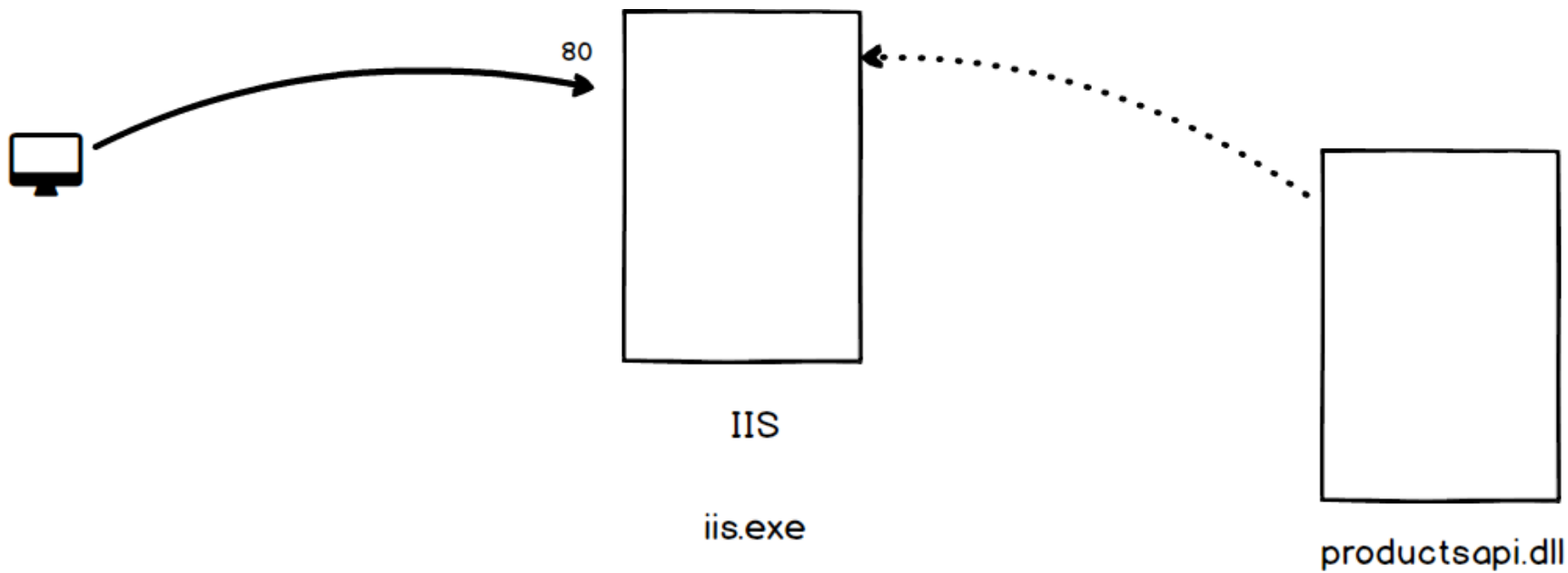
`void Remove(int id)`

`List<Customer> Get(string city)`



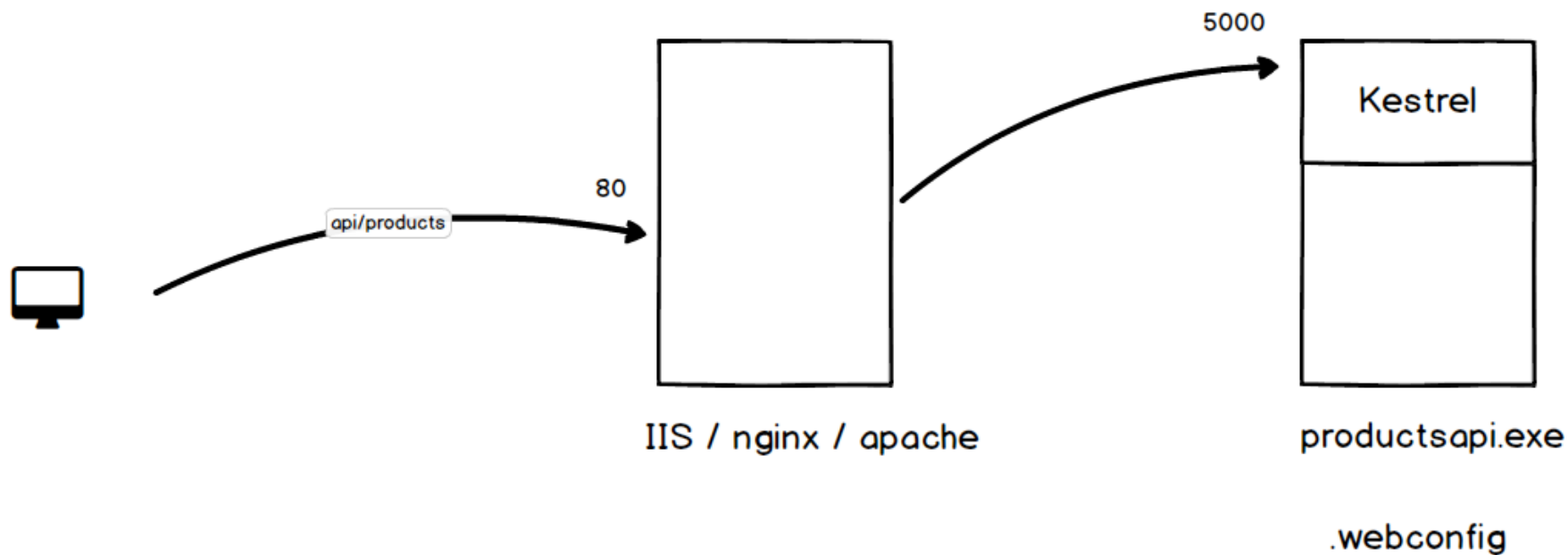
POST api/webhooks

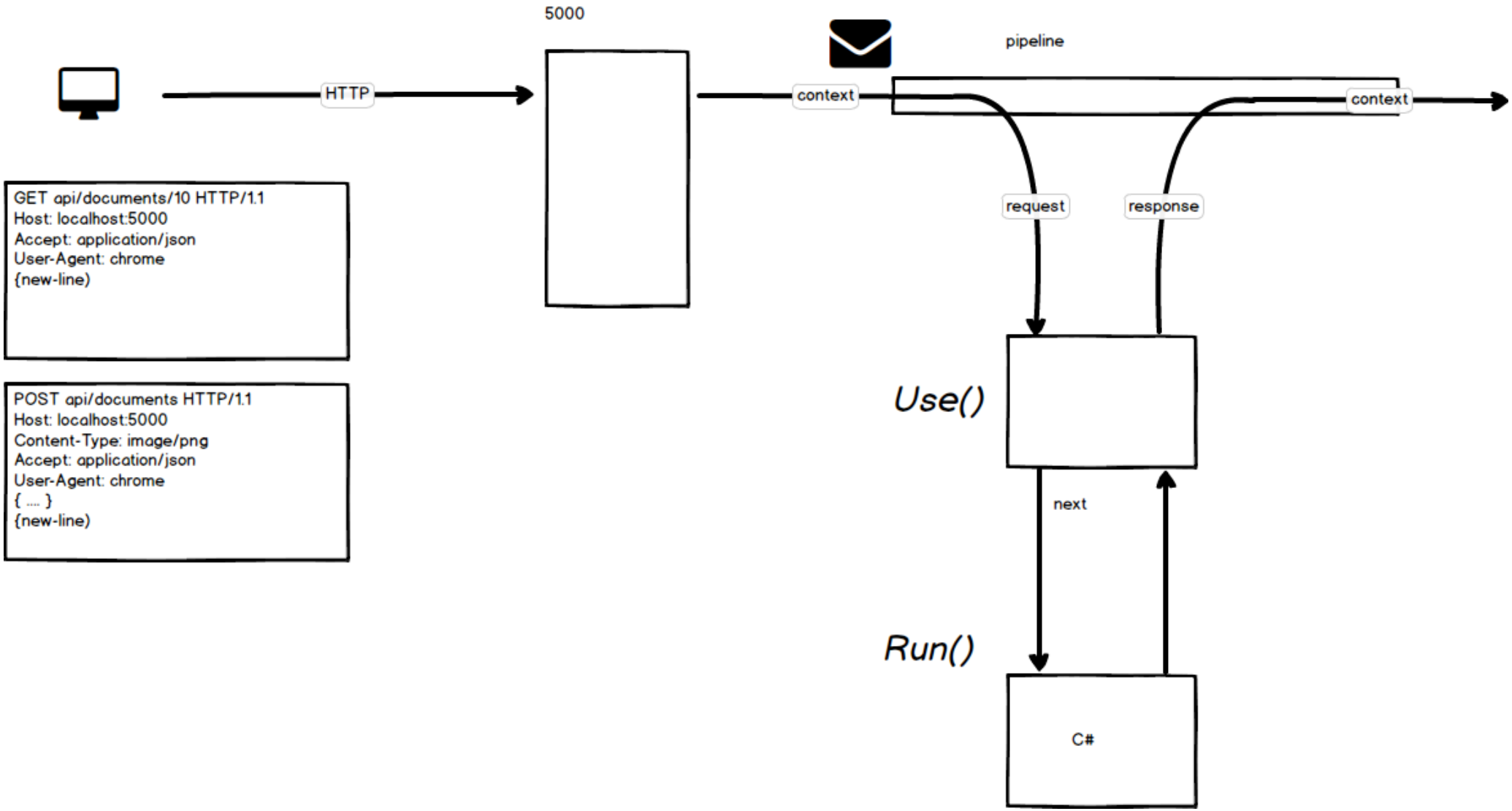




---

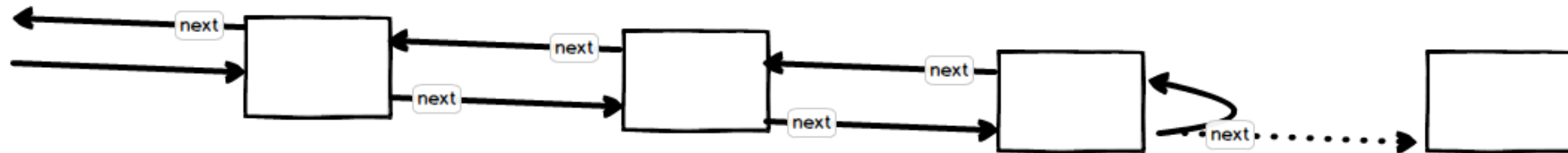
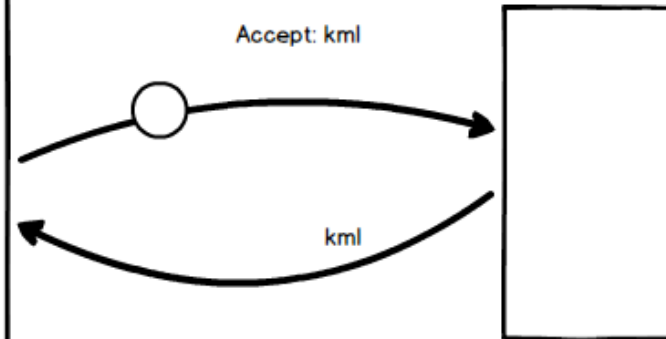
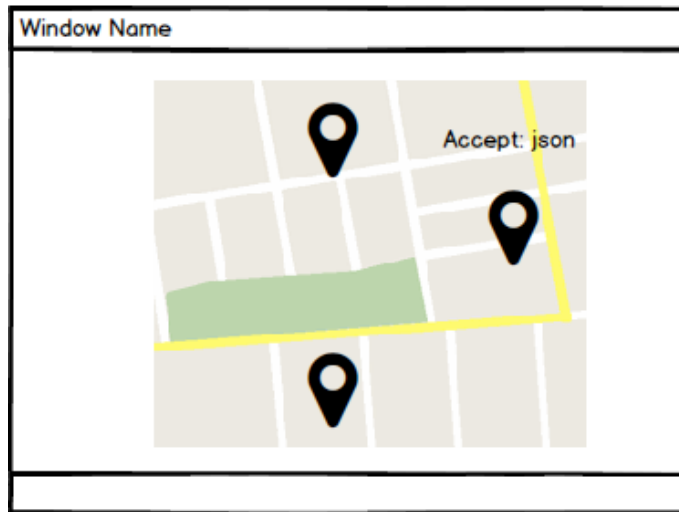
## Self-hosting



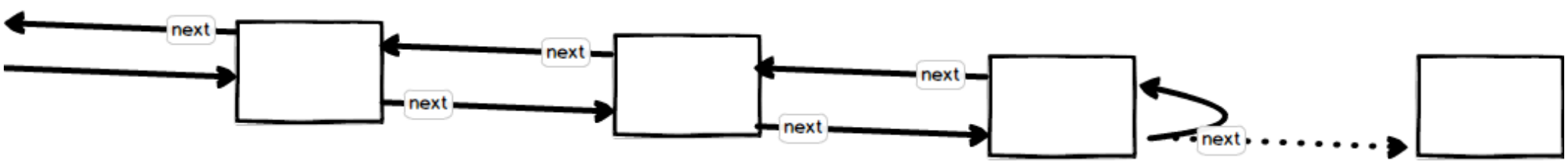


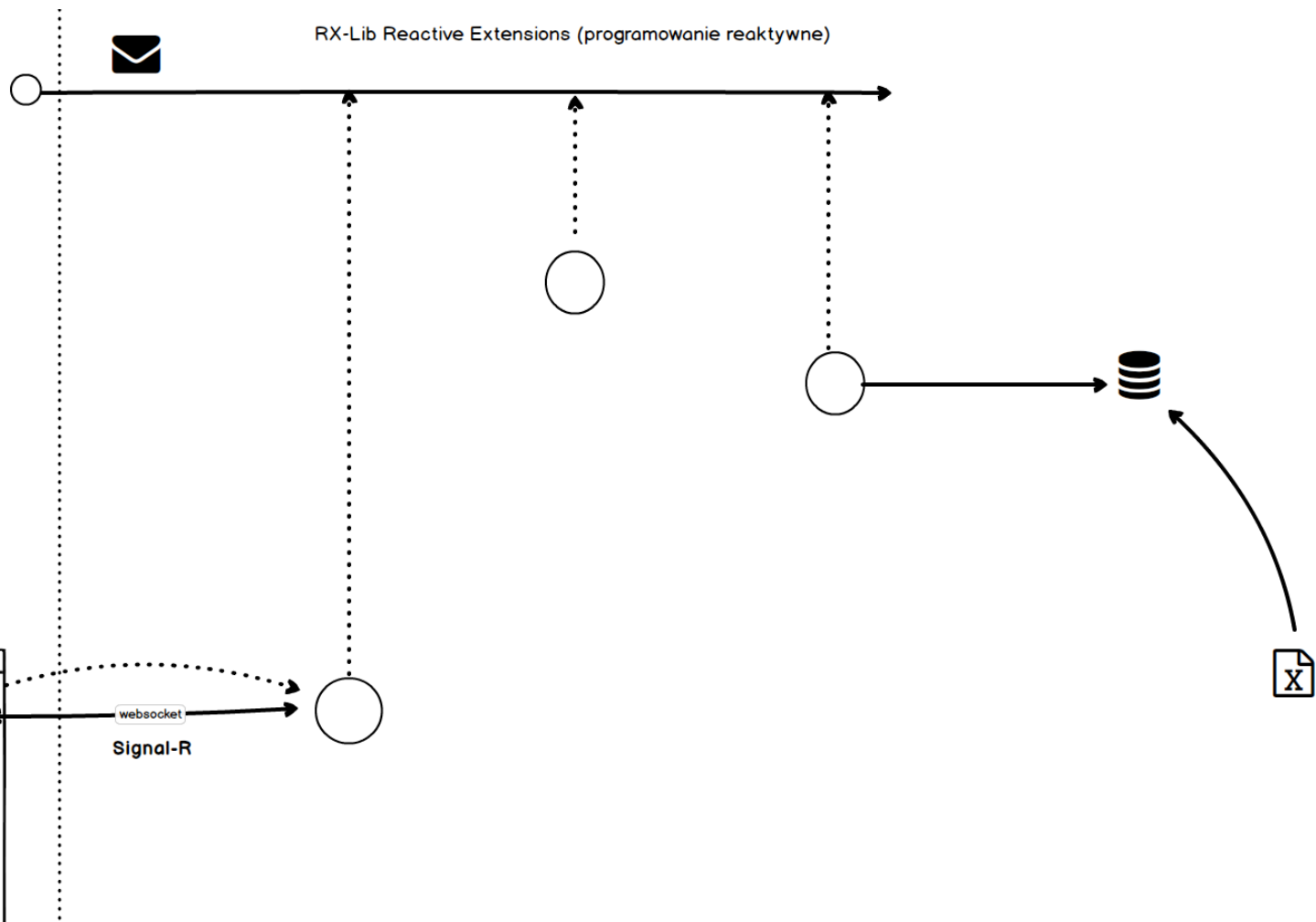
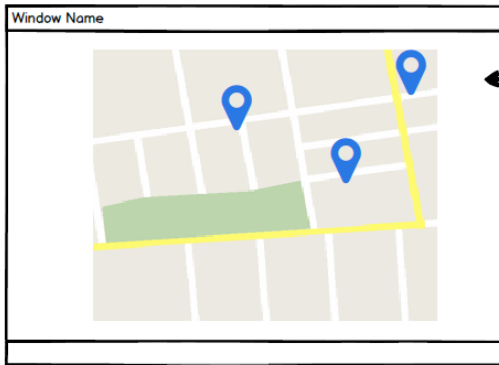
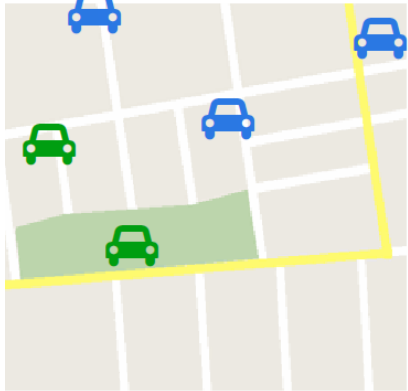
<http://localhost:5000/api/customers?format=kml>

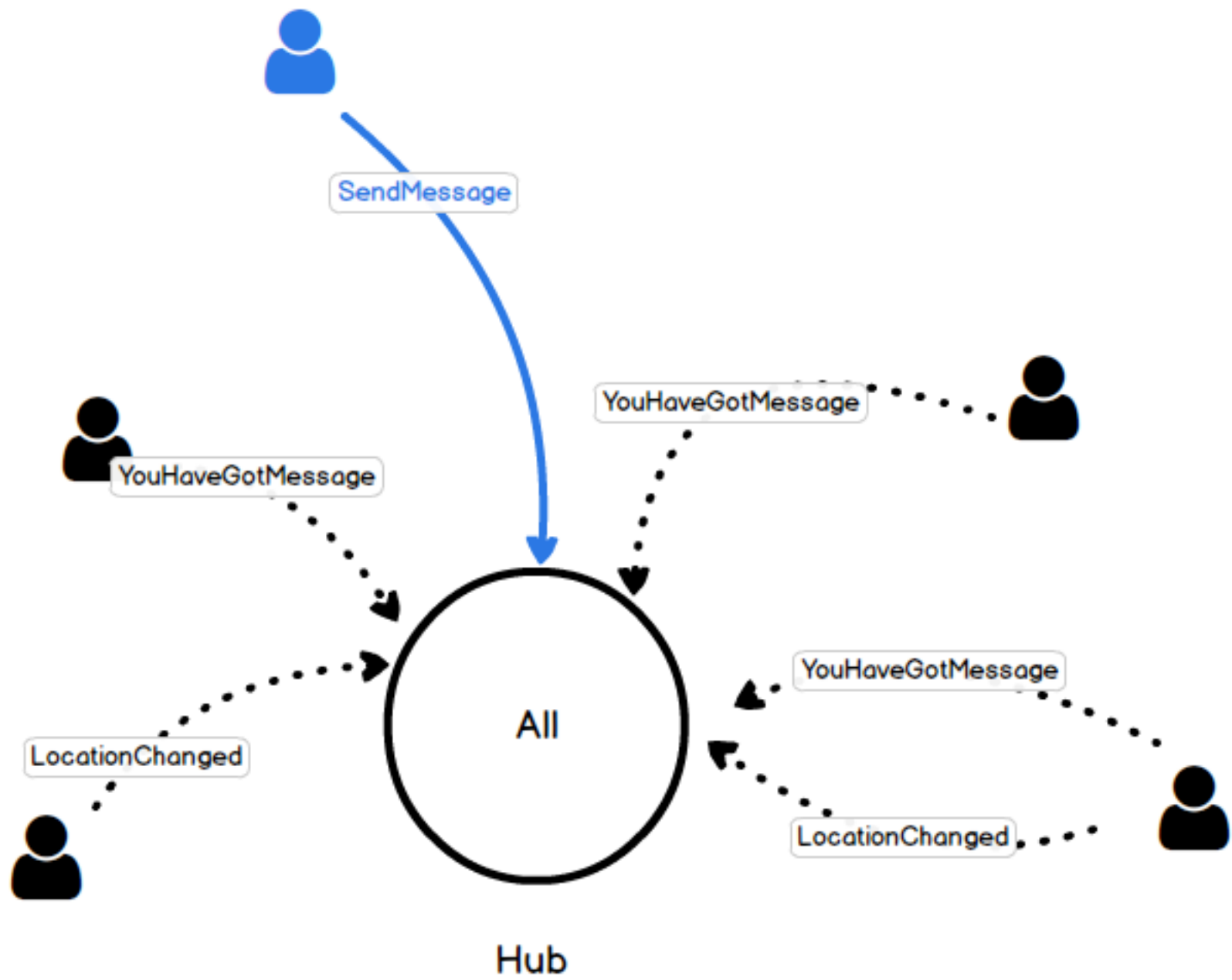
<http://localhost:5000/api/customers.kml>











## CustomersController

IOrderRepository      Get()  
ICustomerRepository    Get(int id)



GET api/customers



GET api/customers/10

GET api/customers?fields=firstname,lastname

GET api/customers/10/invoices



## OrdersController

IOrderRepository



GET api/orders

Get(int customerId)

GET api/orders/10

GET api/orders/10/details

*[HttpGet("/api/customers/{customerId}/orders")]*

GET api/customers/10/orders



## ReportsController

POST api/report-salaries?from-date...&to....

## REST API

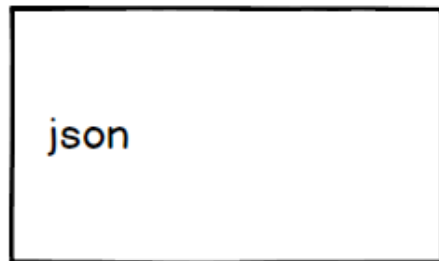
GET `api/customers?fields=firstname,lastname`

**OData**     SQL w URL ;-)

*GET `api/customers?fields=firstname,lastname&order=age$filter=IsRemoved eq 1`*

## GraphQL

POST `api/graphql`



Window Name

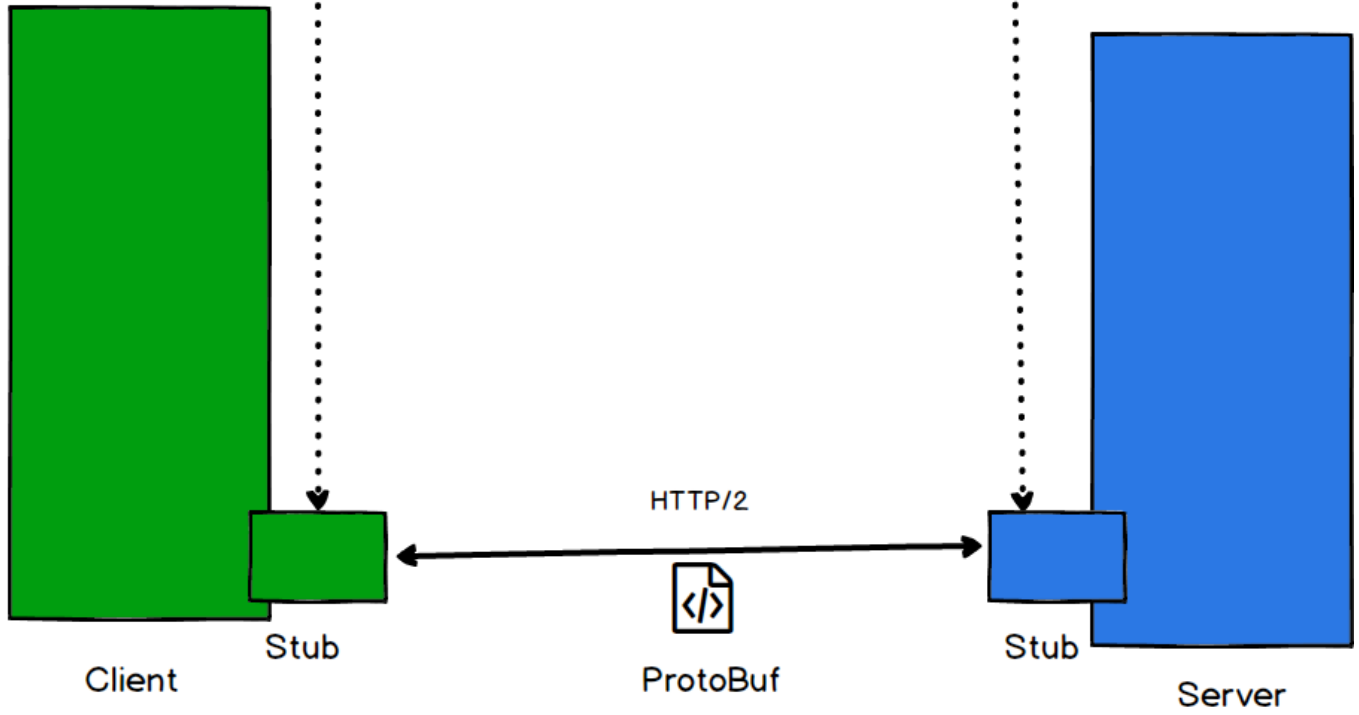
Button



gRPC

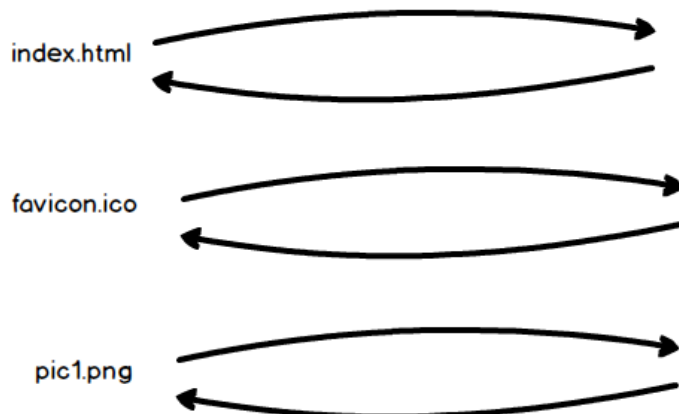


.proto

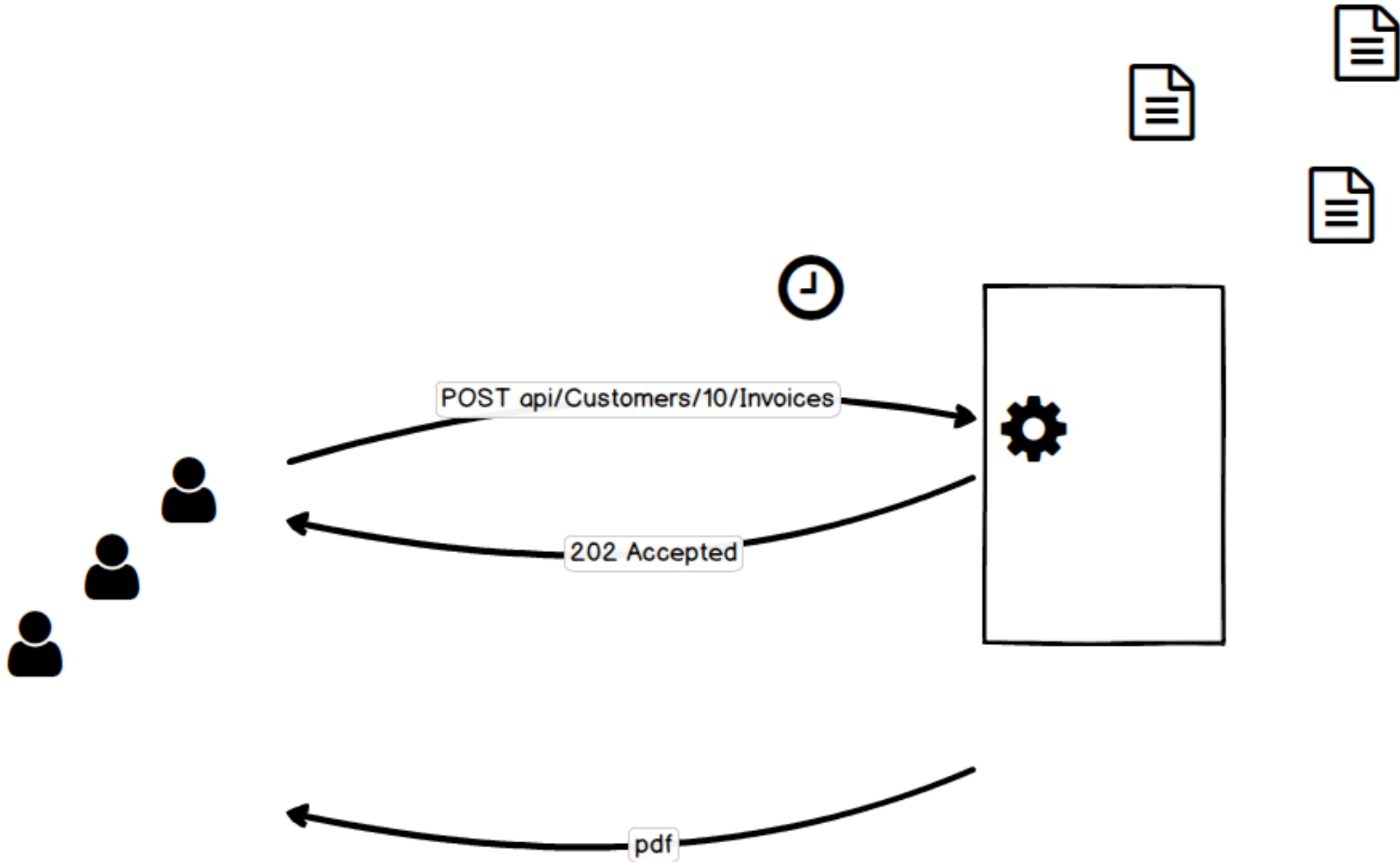


HTTP/2

HTTP/1.1



# Hangfire







### ***IPrincipal***

IIdentity Identity {get; }  
bool IsInRole(string role)



### ***IIdentity***

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

Type	Value
kat	B
kat	C
termin	bezterminowe
phone	555-111-222
email	marcin.sulecki@sulmar.pl
role	trainer
role	developer

Claim