

method      resource      protocol

headers {  
GET api/products HTTP/1.1  
Host: localhost  
Accept: application/xml  
body {

request

status code

headers {  
200 OK  
Content-Type: application/json  
Server: kestrel  
body {  
{  
  json  
}

response

```
<xml>
<envelope>
  <GetProductById>
    <Id>1</Id>
  </GetProductById>
</envelope>
</xml>
```

request

method	opis	status code	
GET	pobierz zasób	200 OK	404 NotFound
POST	utwórz zasób	201 Created	
PUT	podmień zasób		
PATCH	zmodyfikuj zasób		
DELETE	usuń zasób		
HEAD	pobierz nagłówek		
OPTIONS	pobierz dopuszczalne operacje na zasobie		



appsettings.json



appsettings.development.json

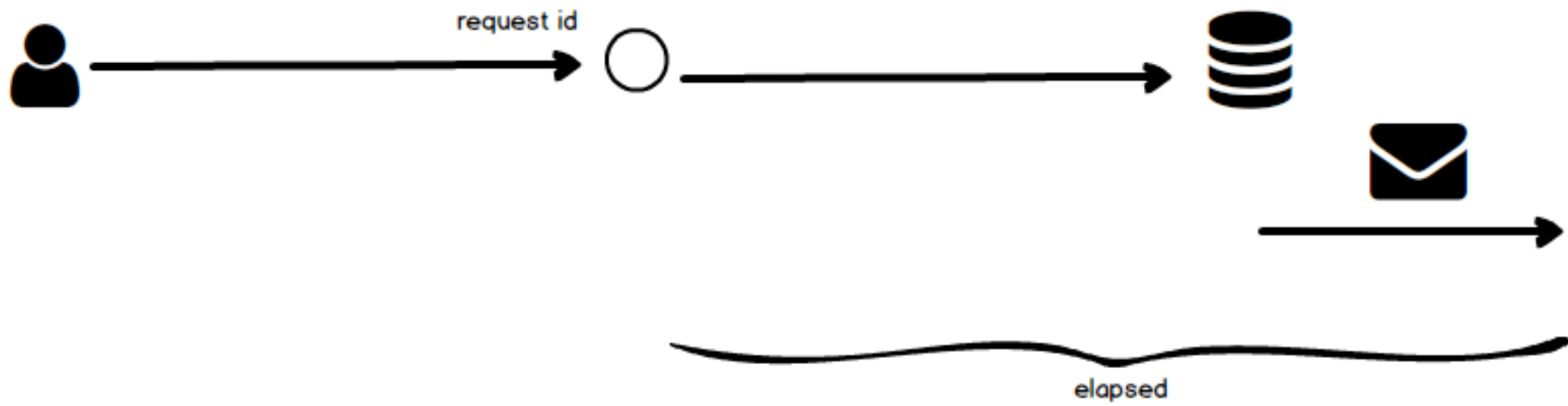


appsettings.xml



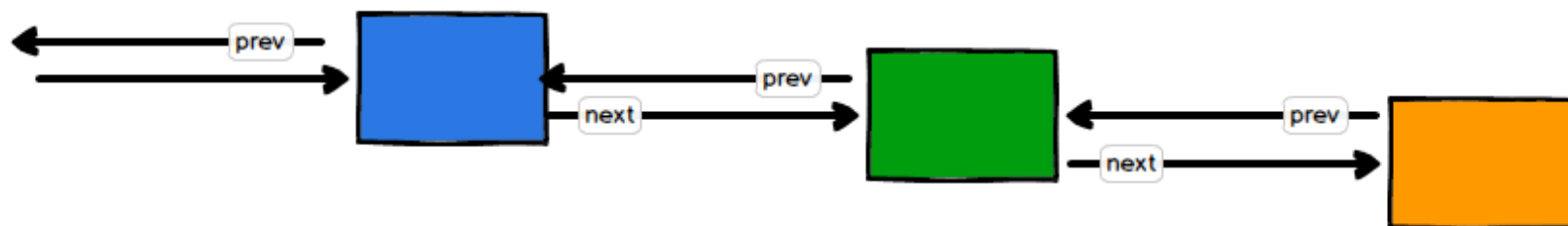
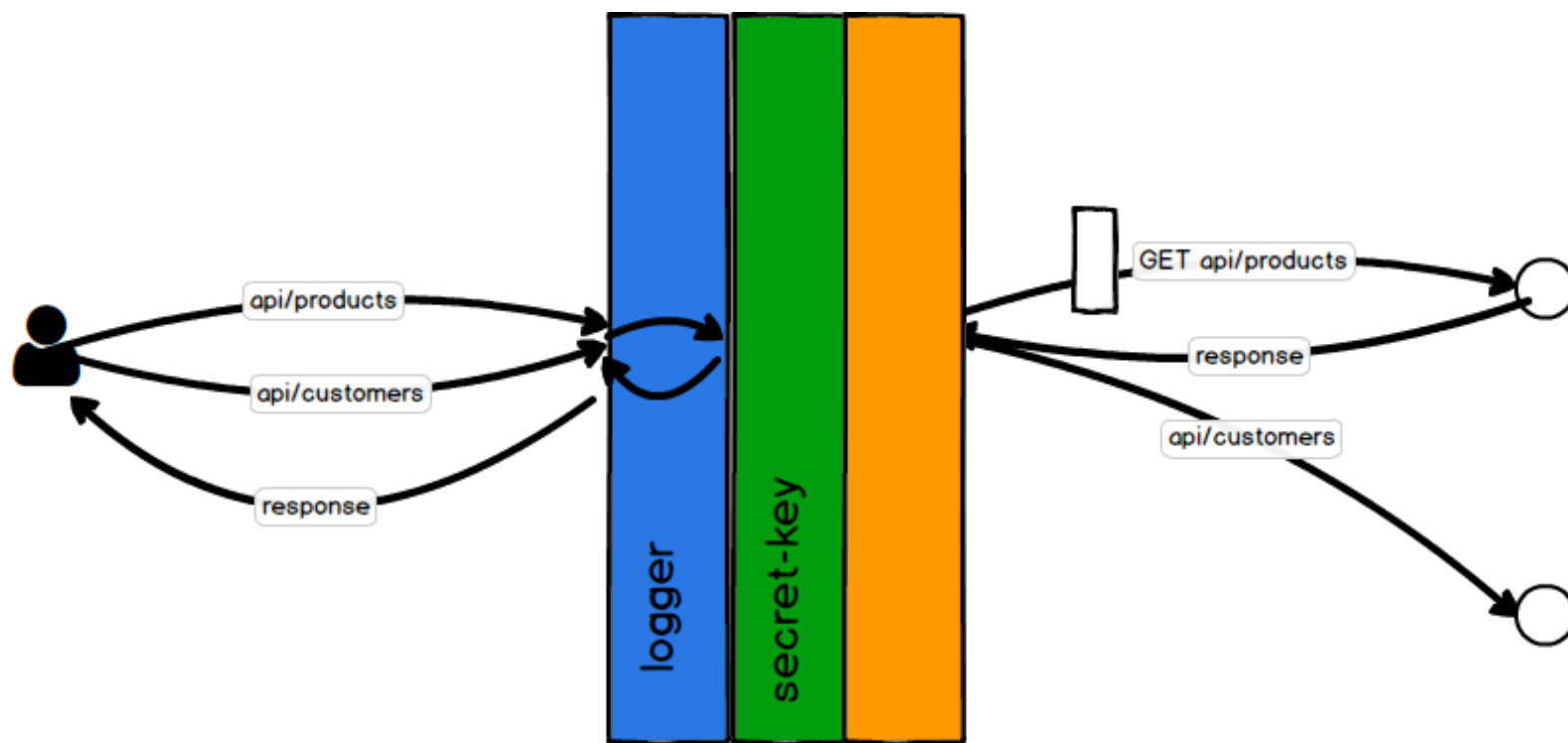
builder.Configuration





Type	Implementation	Life cycle
ICustomerRepository IEnumerable<Customer>	InMemoryCustomerRepository ( ) => ...	singleton  scoped  transient





Chain of Responsibility (Łancuch odpowiedzialności)

**Uwierzytelnianie**

Authentication



kim jesteś?

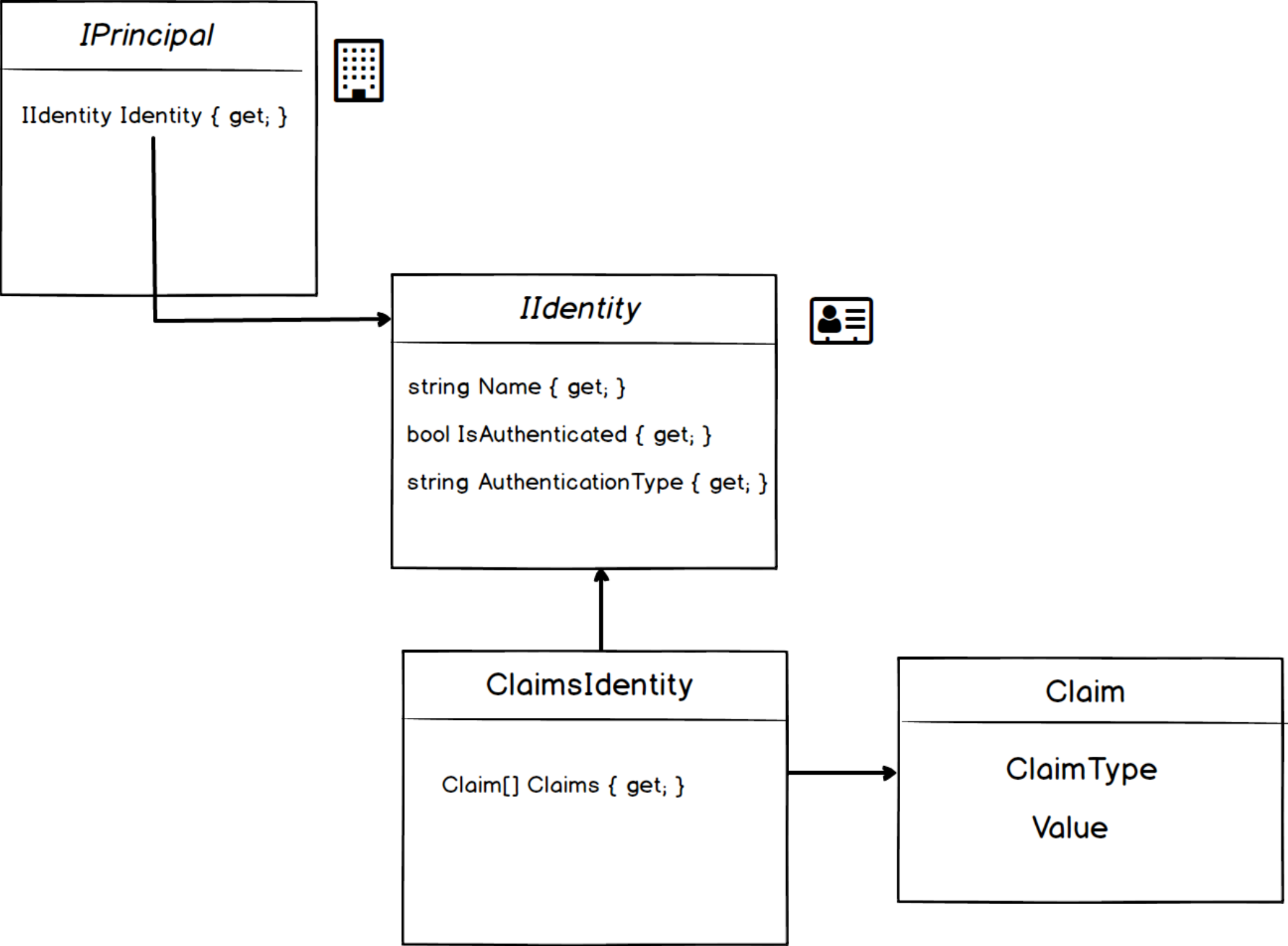
**Autoryzacja**

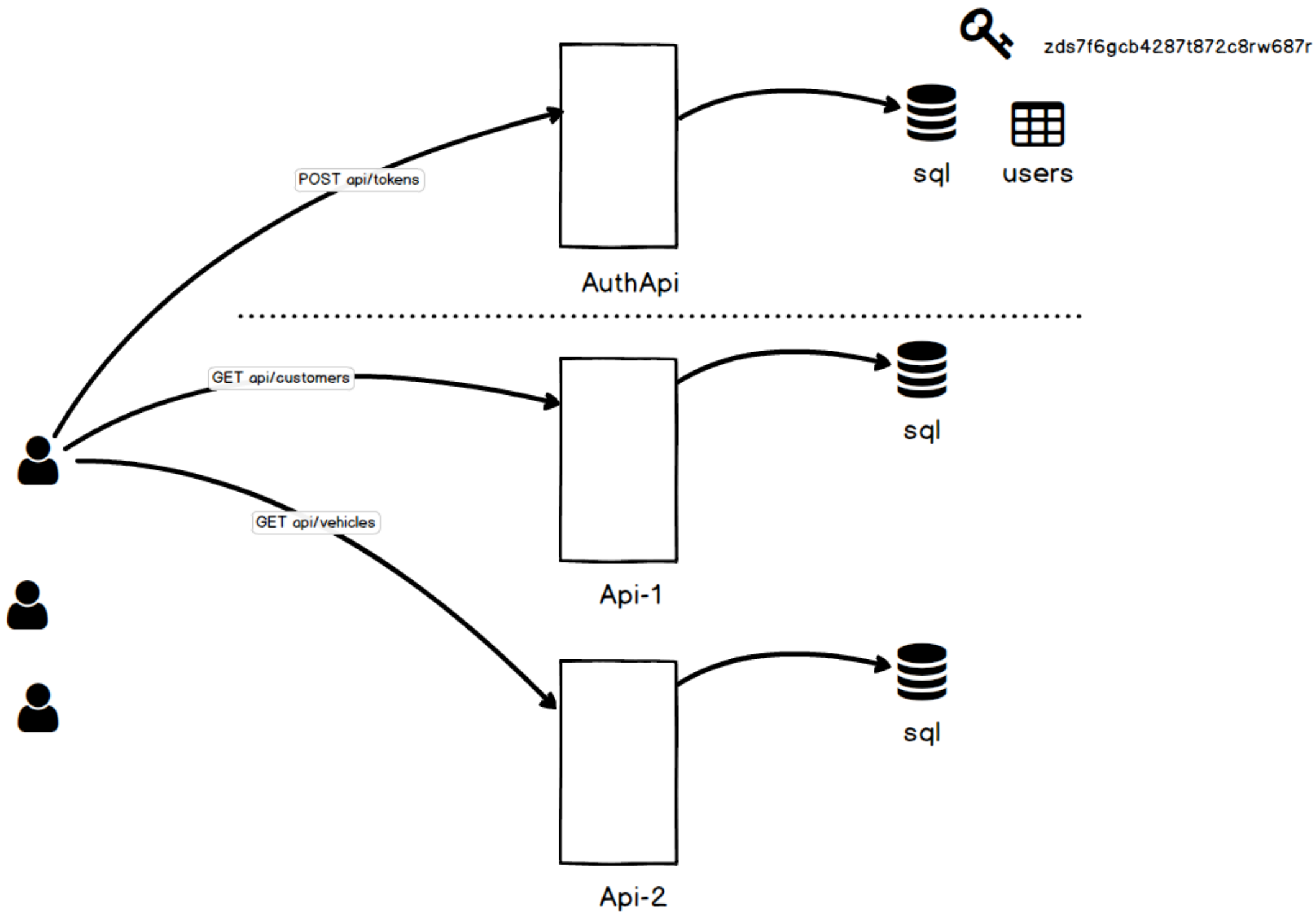
Authorization



do czego masz dostęp?







(JWT) Json Web Tokens

GET api/products HTTP/1.1  
Host: localhost  
Accept: application/xml  
Authorization: {scheme} {parameters}

request

GET api/products HTTP/1.1  
Host: localhost  
Accept: application/xml  
Authorization: Basic base64({login:password})

Basic

GET api/products HTTP/1.1  
Host: localhost  
Accept: application/xml  
Authorization: Bearer {token}

Bearer

RPC (Remote Procedure Call)				Resource	
DCOM	XML-RPC	SOAP	WCF	REST API	GraphQL
gRPC					

gRPC

HTTP 2.0

binary    streaming    multi-platform

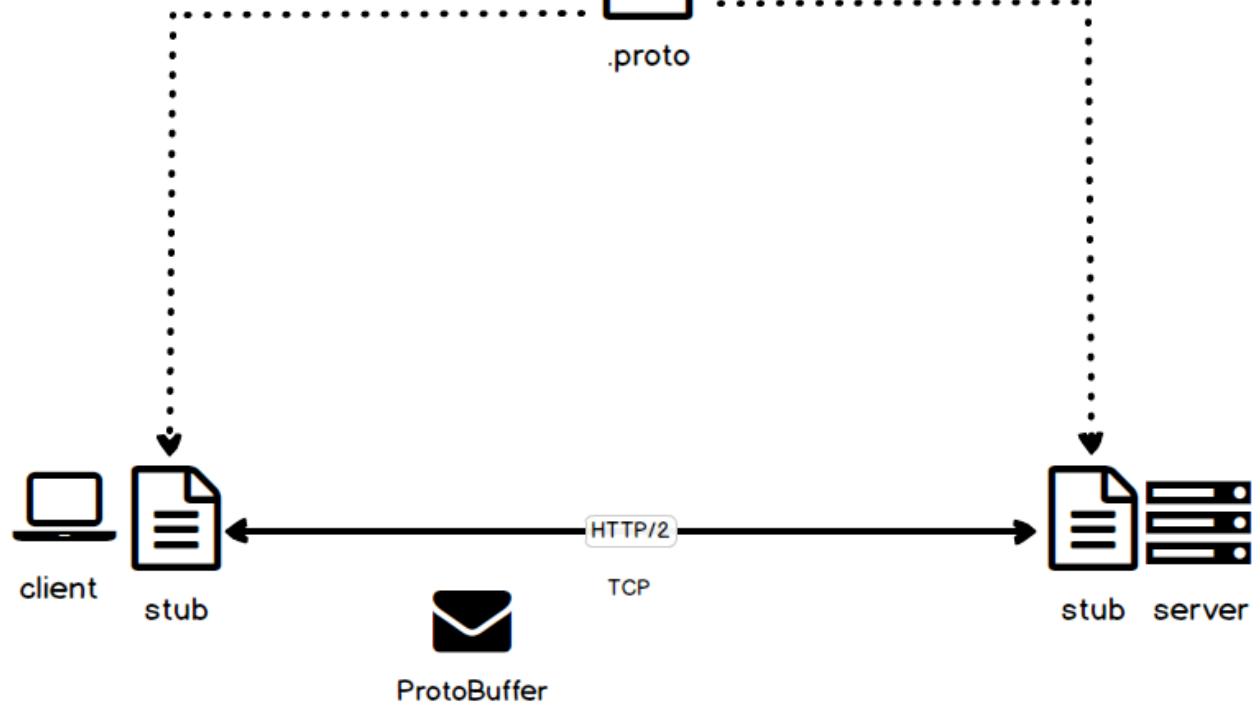
# gRPC

IDL (Interface Definition Language)



.proto

protoc.exe



Unary

Server Streaming

Client Streaming

Duplex Streaming

