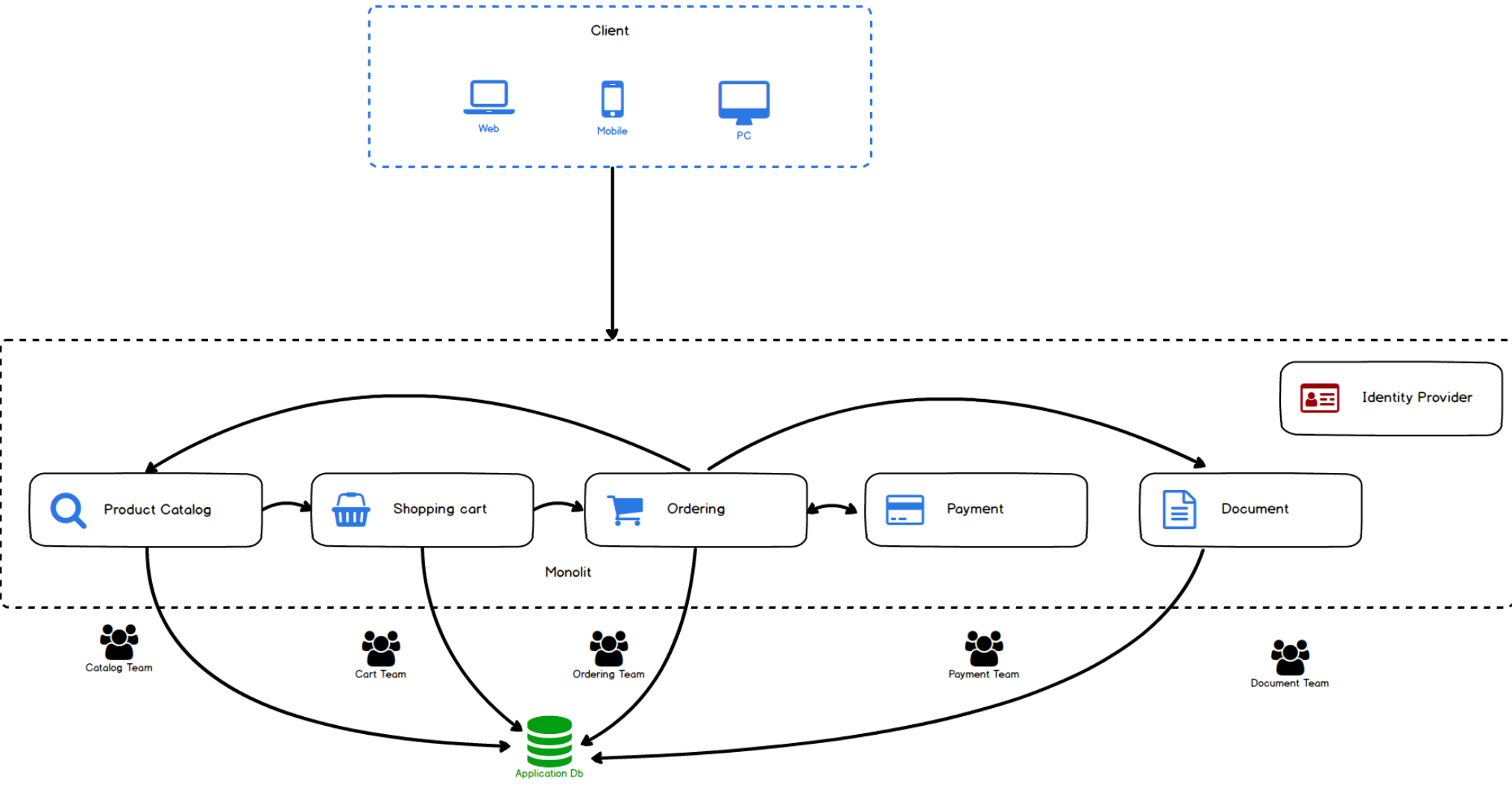


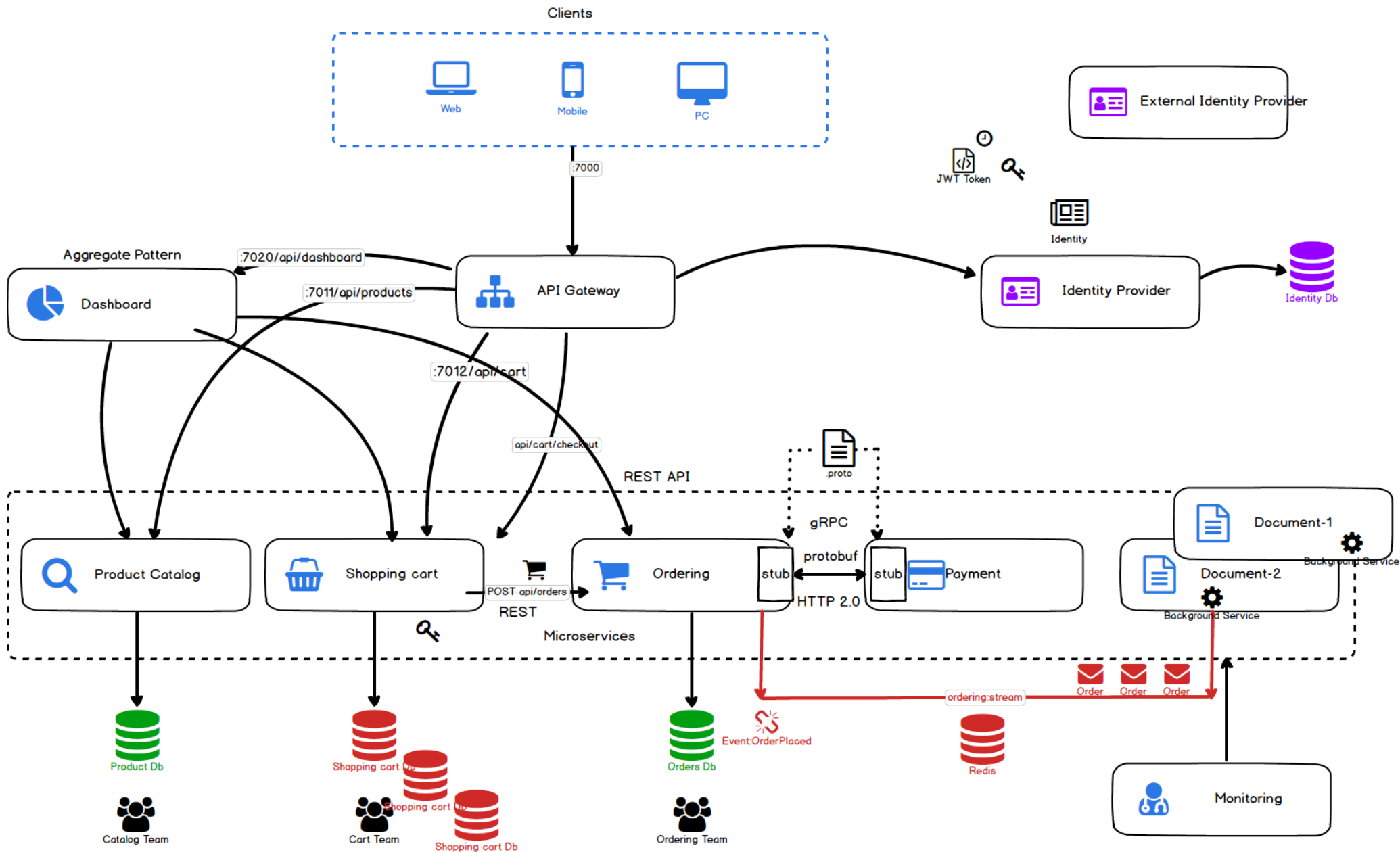


Fundamentals

# Architektura monolitu



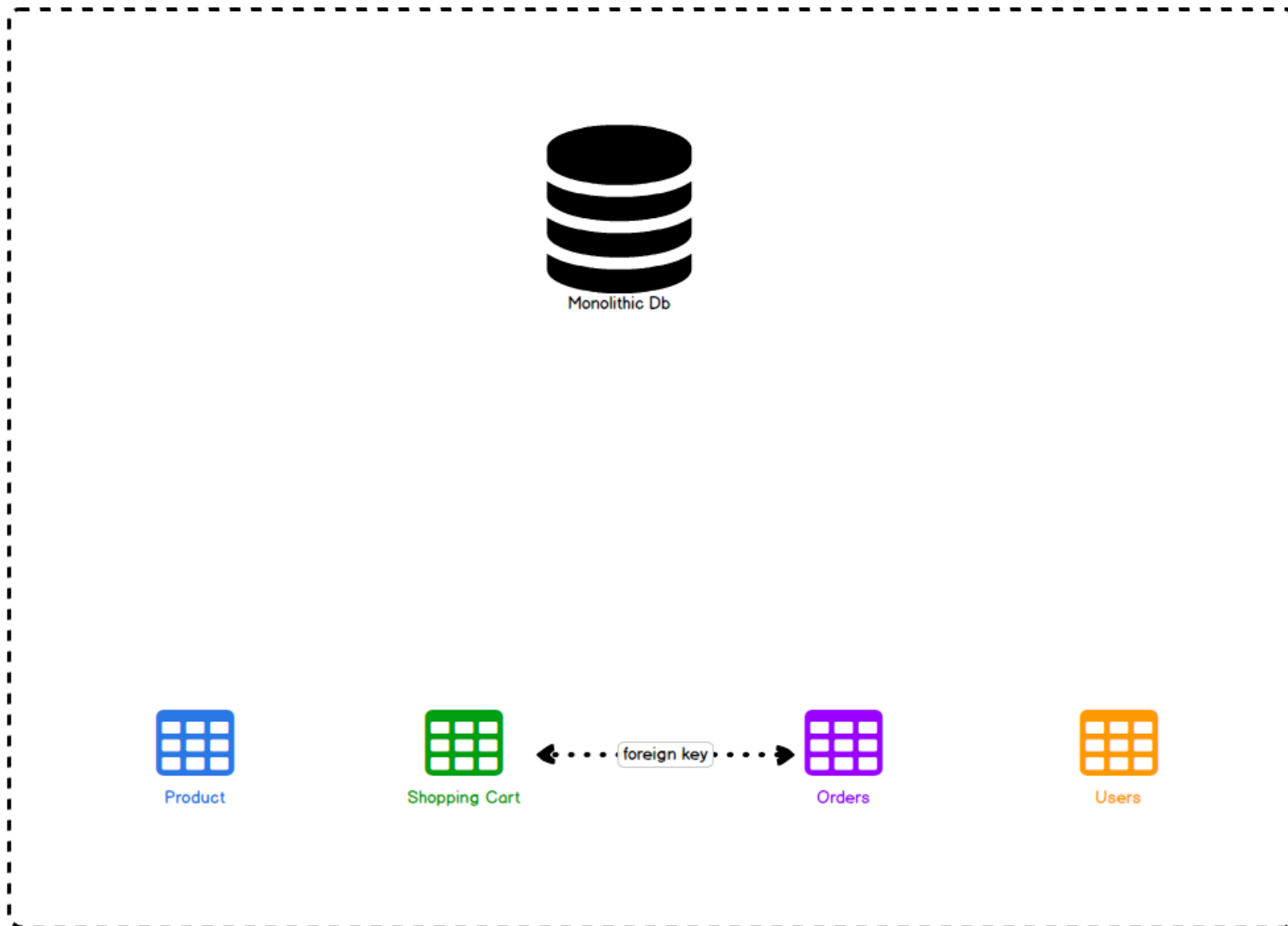
# Architektura mikro-usług



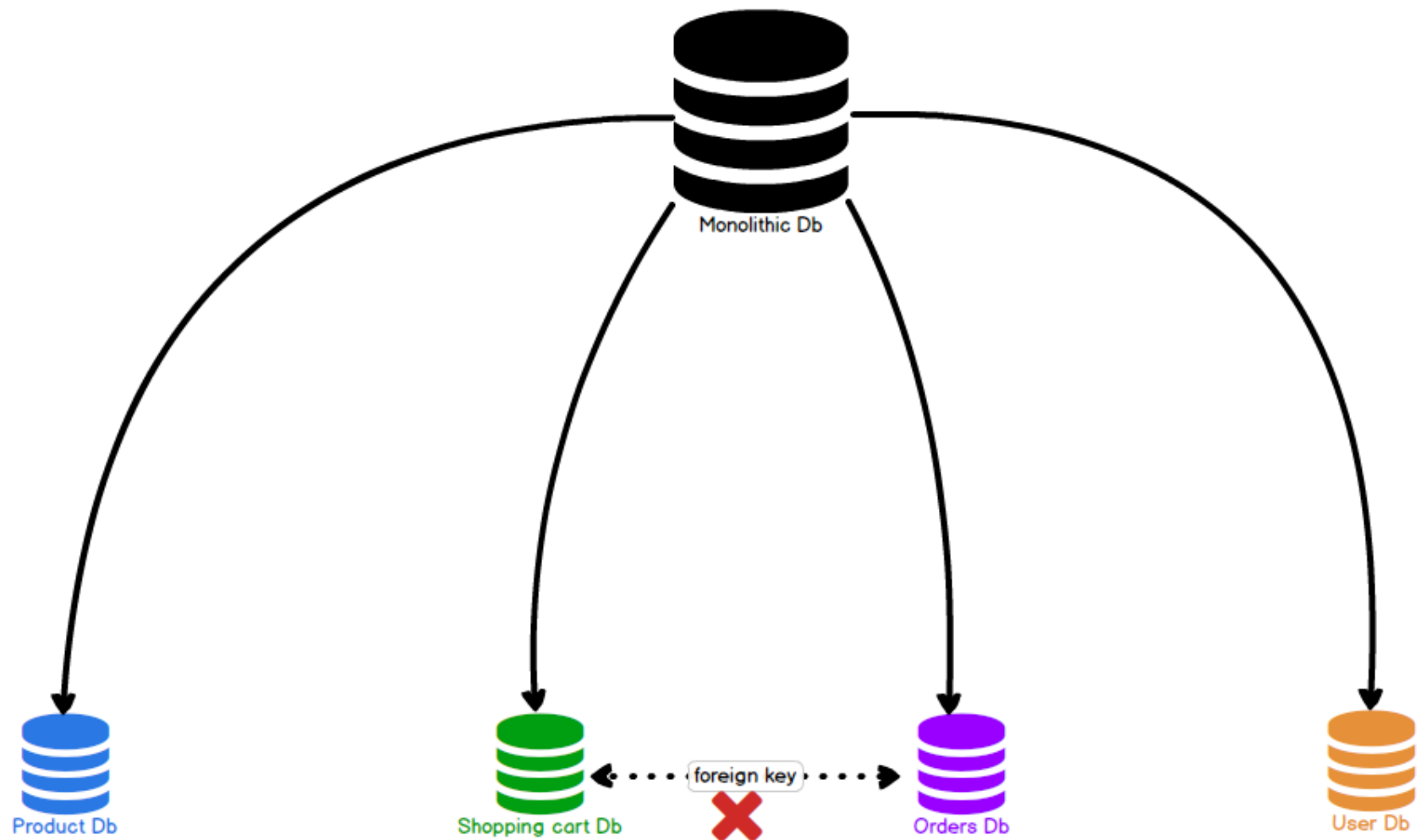


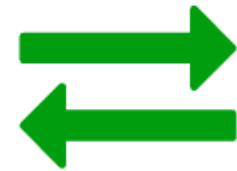
Database

# Baza danych - monolit



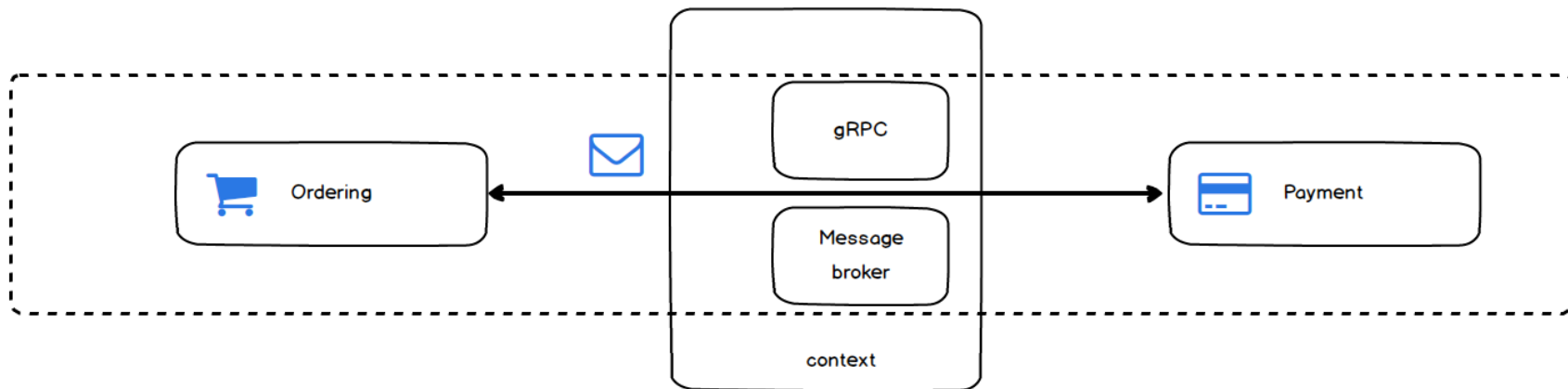
# Baza danych - mikrousługi





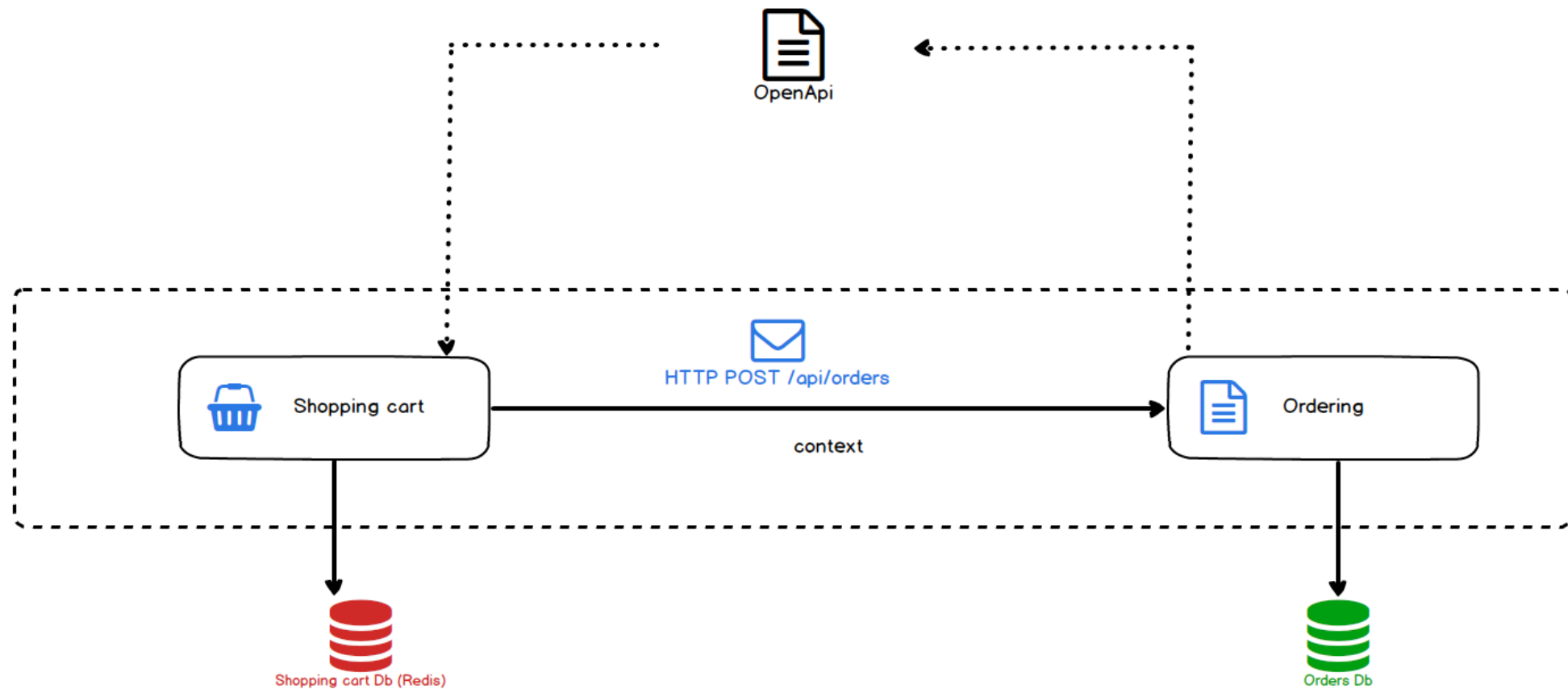
Communication

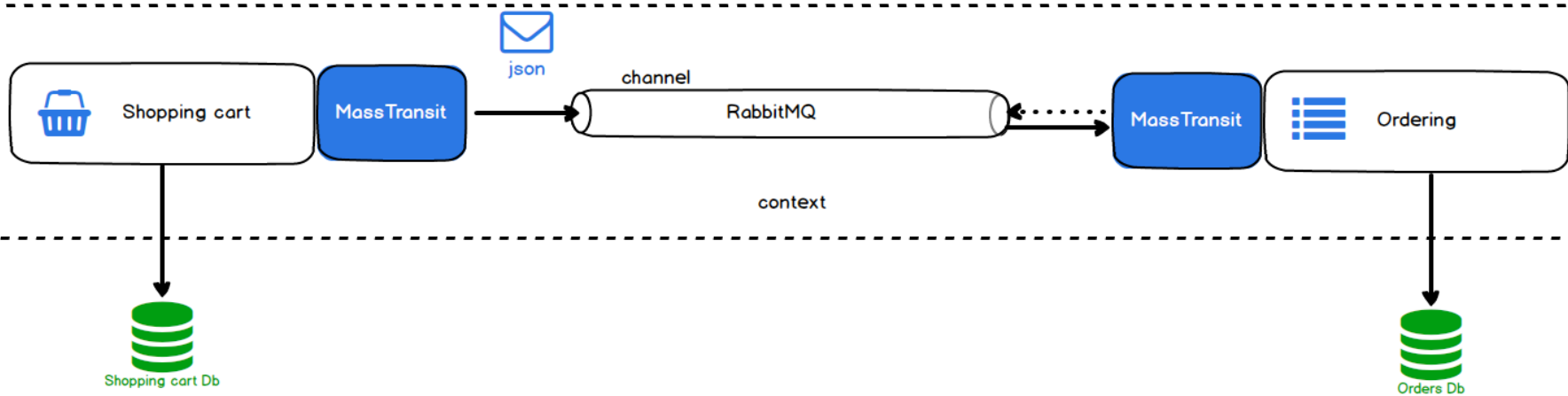
## Komunikacja pomiędzy mikro-usługami



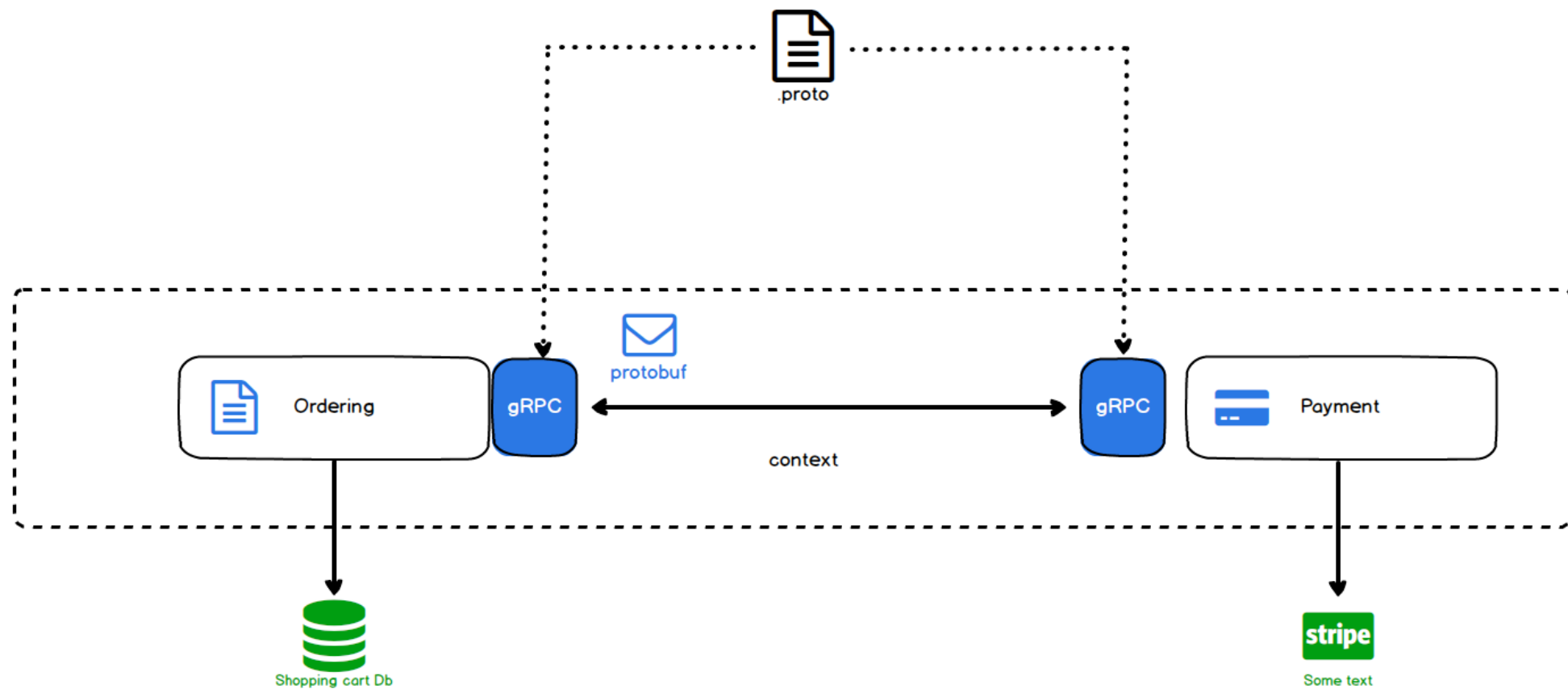


# Komunikacja pomiędzy mikro-usługami za pomocą HTTP



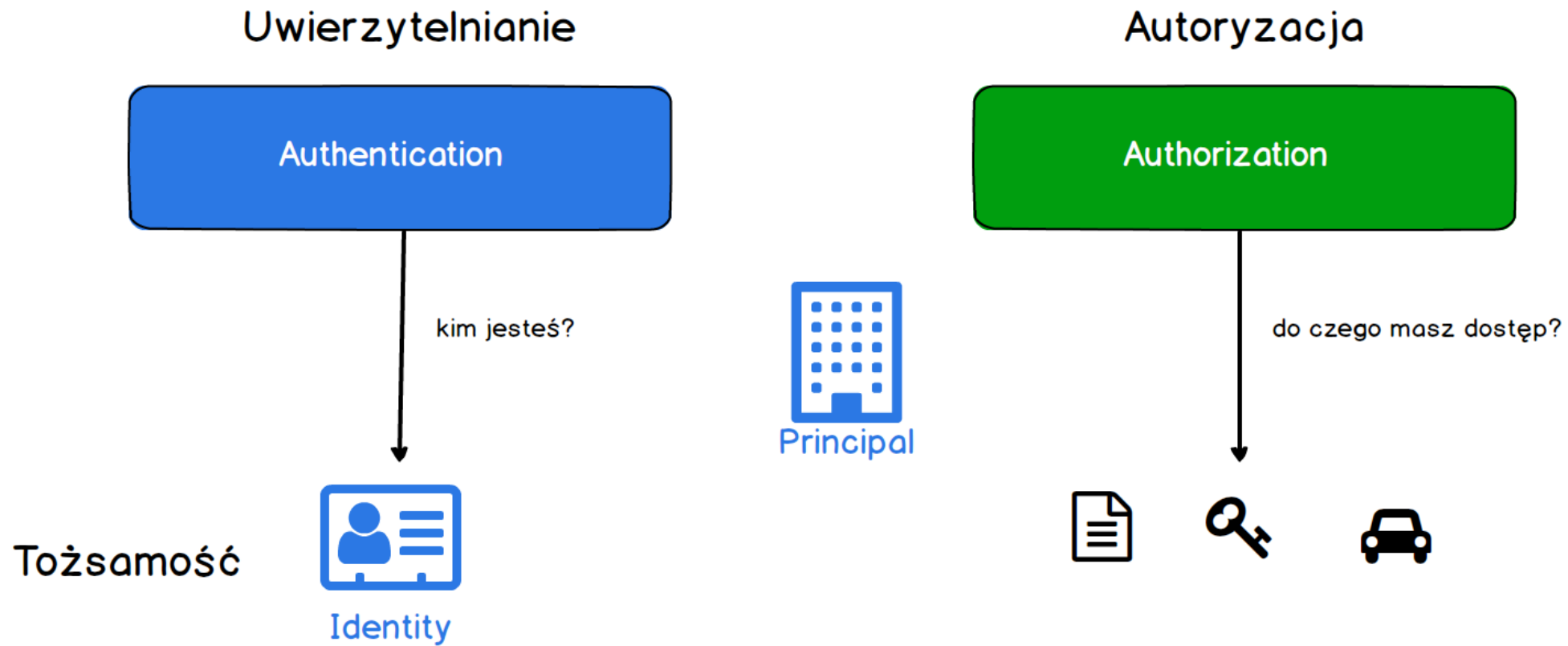


# Komunikacja pomiędzy mikro-usługami za pomocą gRPC





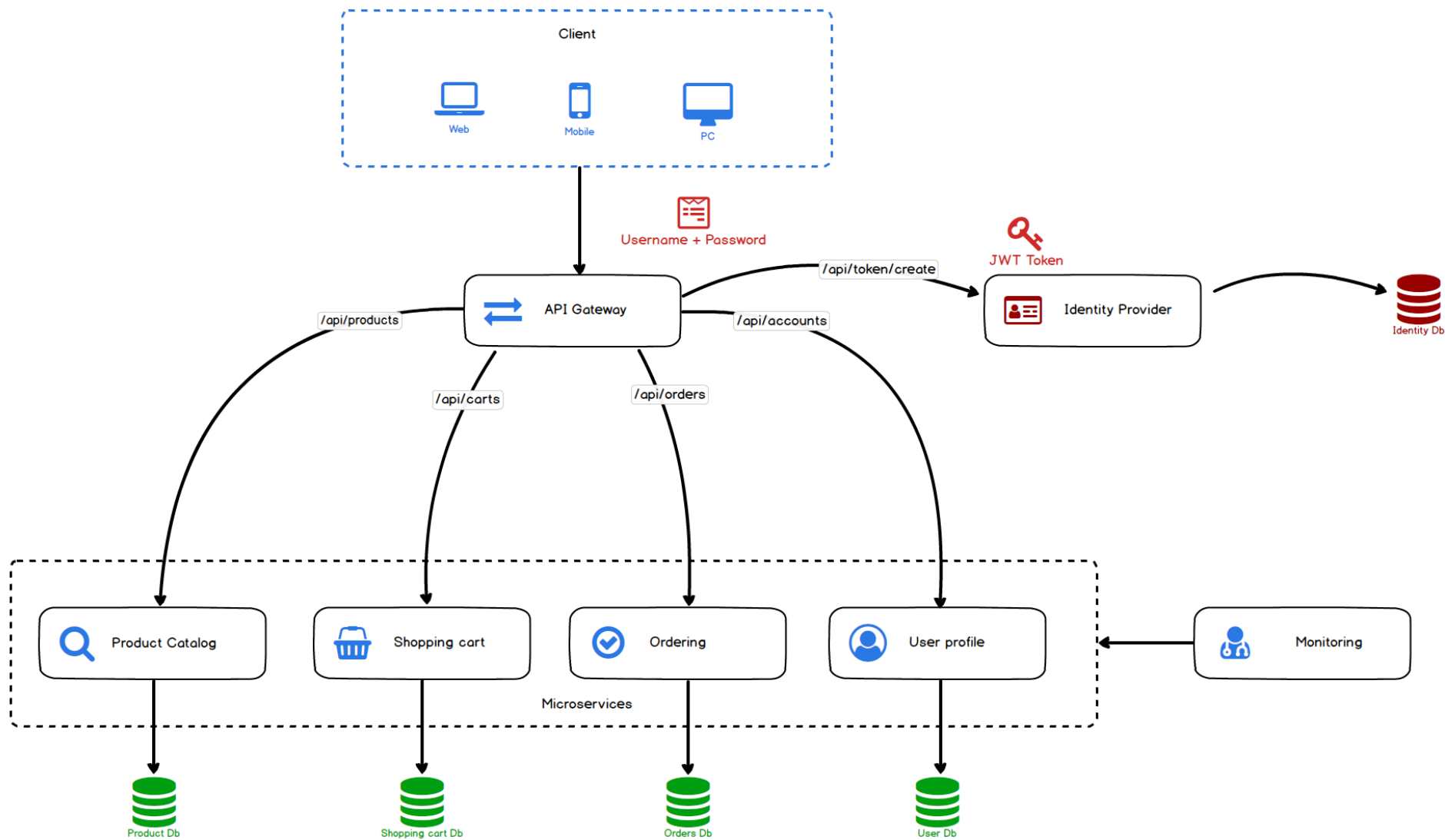
## Authentication / Authorization



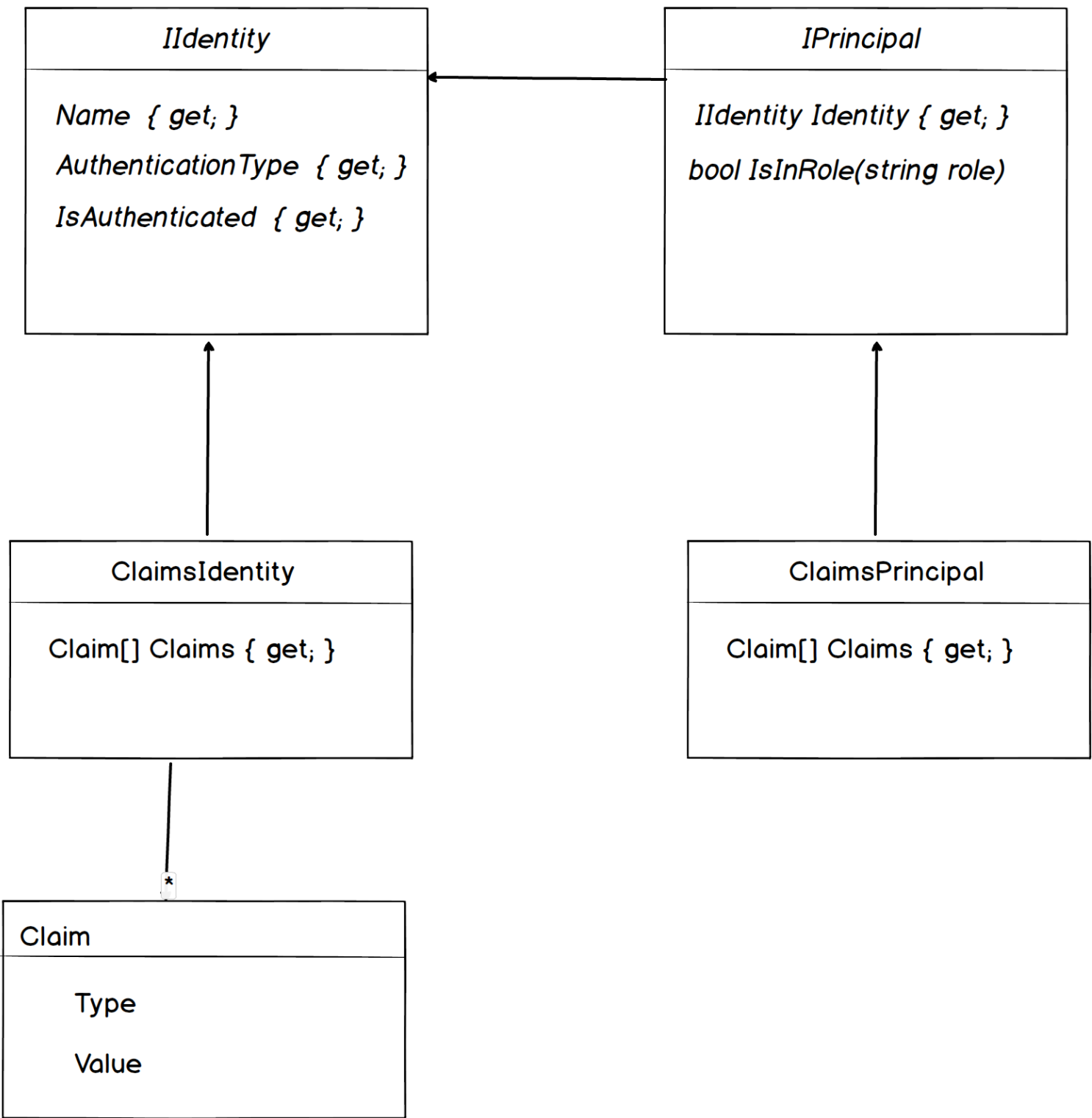
`.UseAuthentication()`

`.UseAuthorization()`

# Architektura mikro-usług



Tożsamość

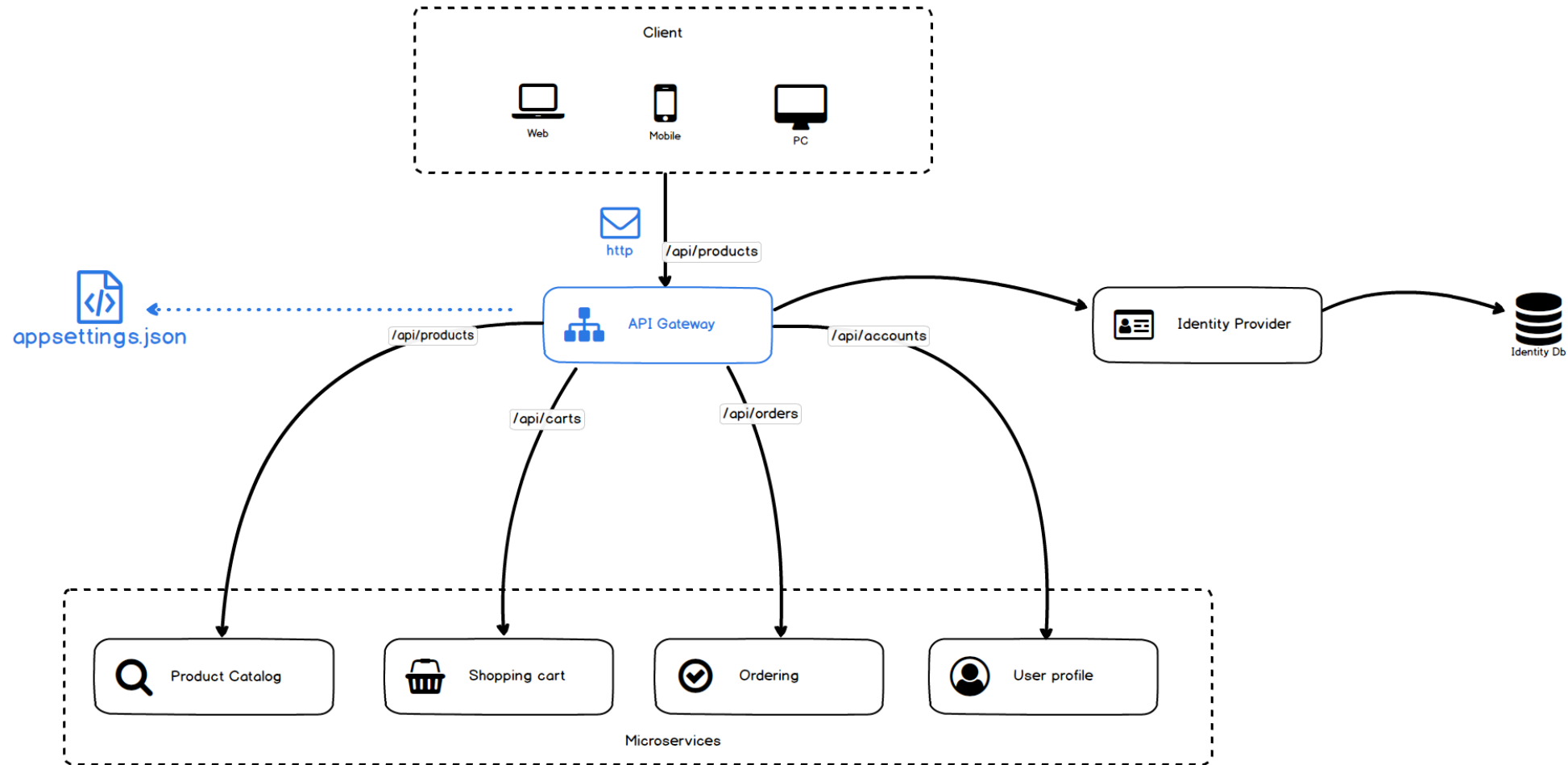




Gateway API



## ReverseProxy (YARP)

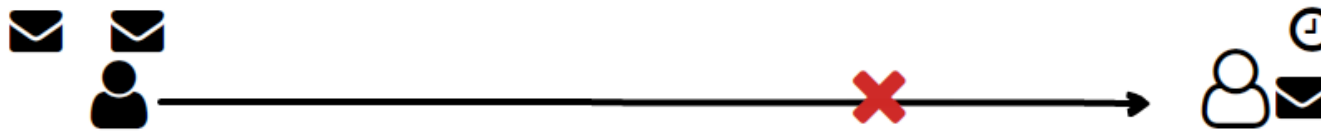




Docker



Notes



*Iterator Design Pattern*

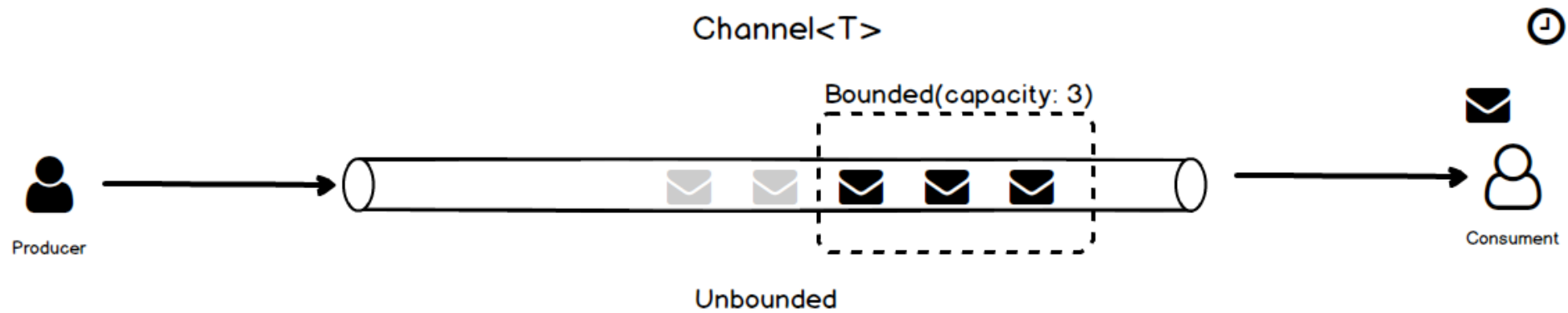
`ConcurrentQueue<T>`

*Observer Design Pattern*

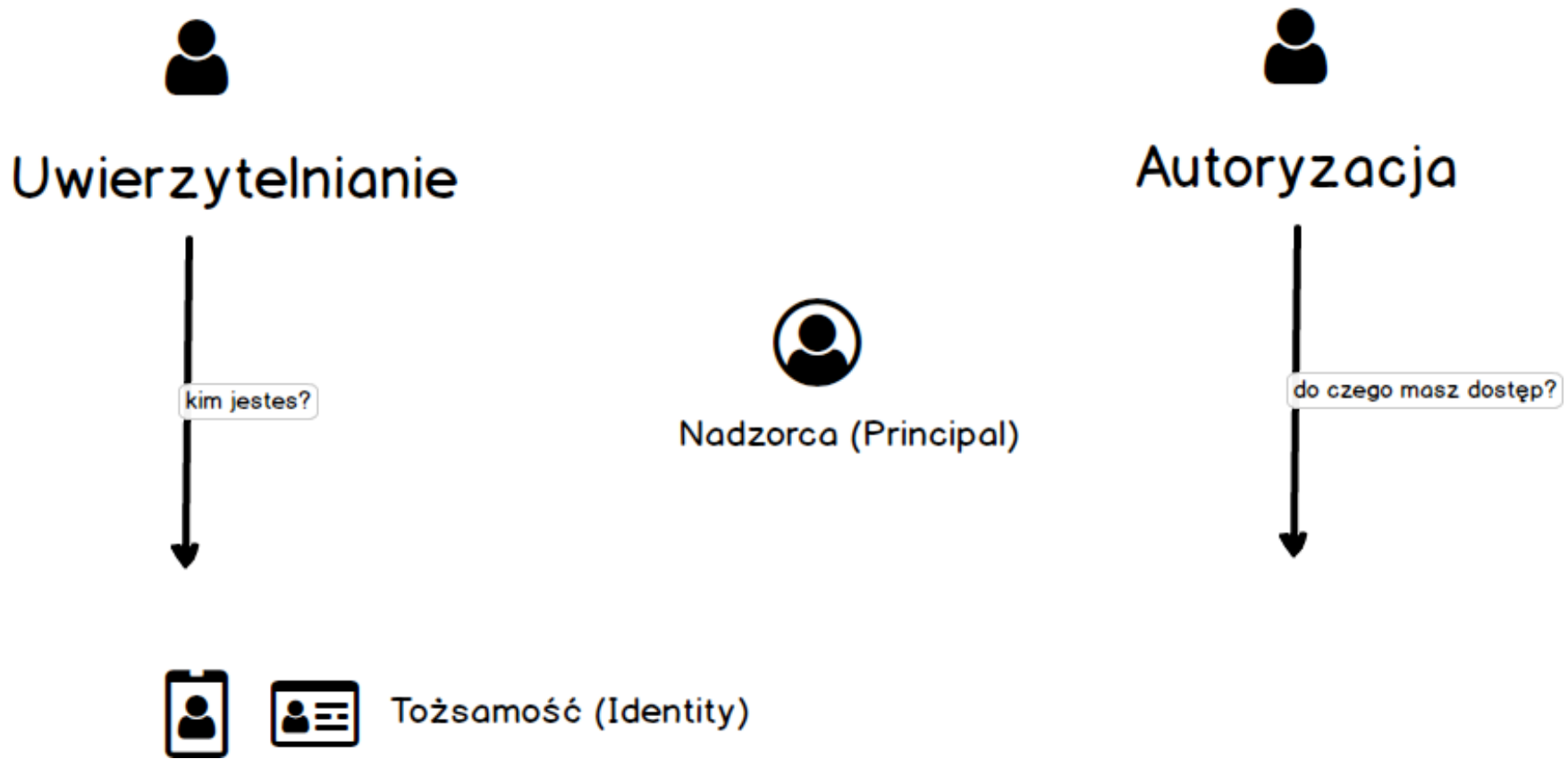
`BlockingCollection<T>`

*Observer Design Pattern*

`Channel<T>`



# Authentication and Authorization





*Identity*

string Name { get; }

bool IsAuthenticated { get; }

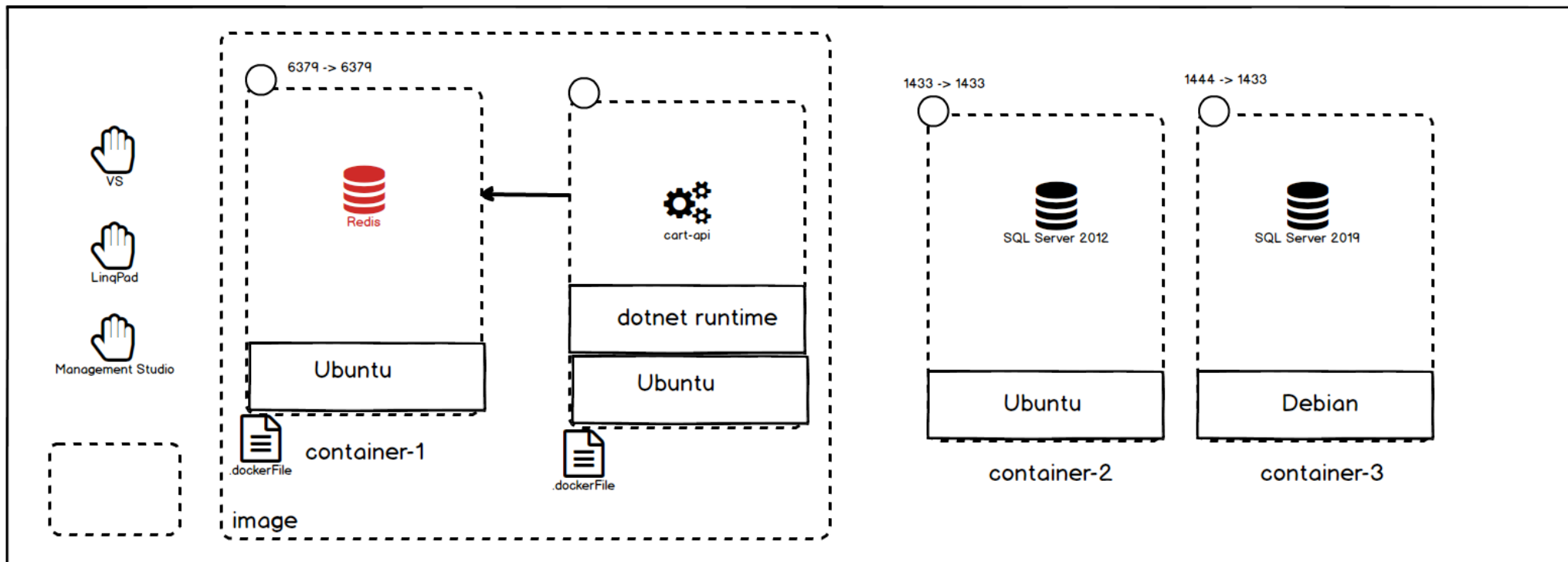
bool AuthenticationType { get; }




**ClaimIdentity**

Claim[] Claims { get; }

*Principal*



Windows / Linux / MacOS


 > *docker compose up*  
 docker-compose.yml > *docker compose down*