

Erwin de Gier & Roy Straub  
The ultimate asynchronous stack



[github.com/erwindeg](https://github.com/erwindeg)



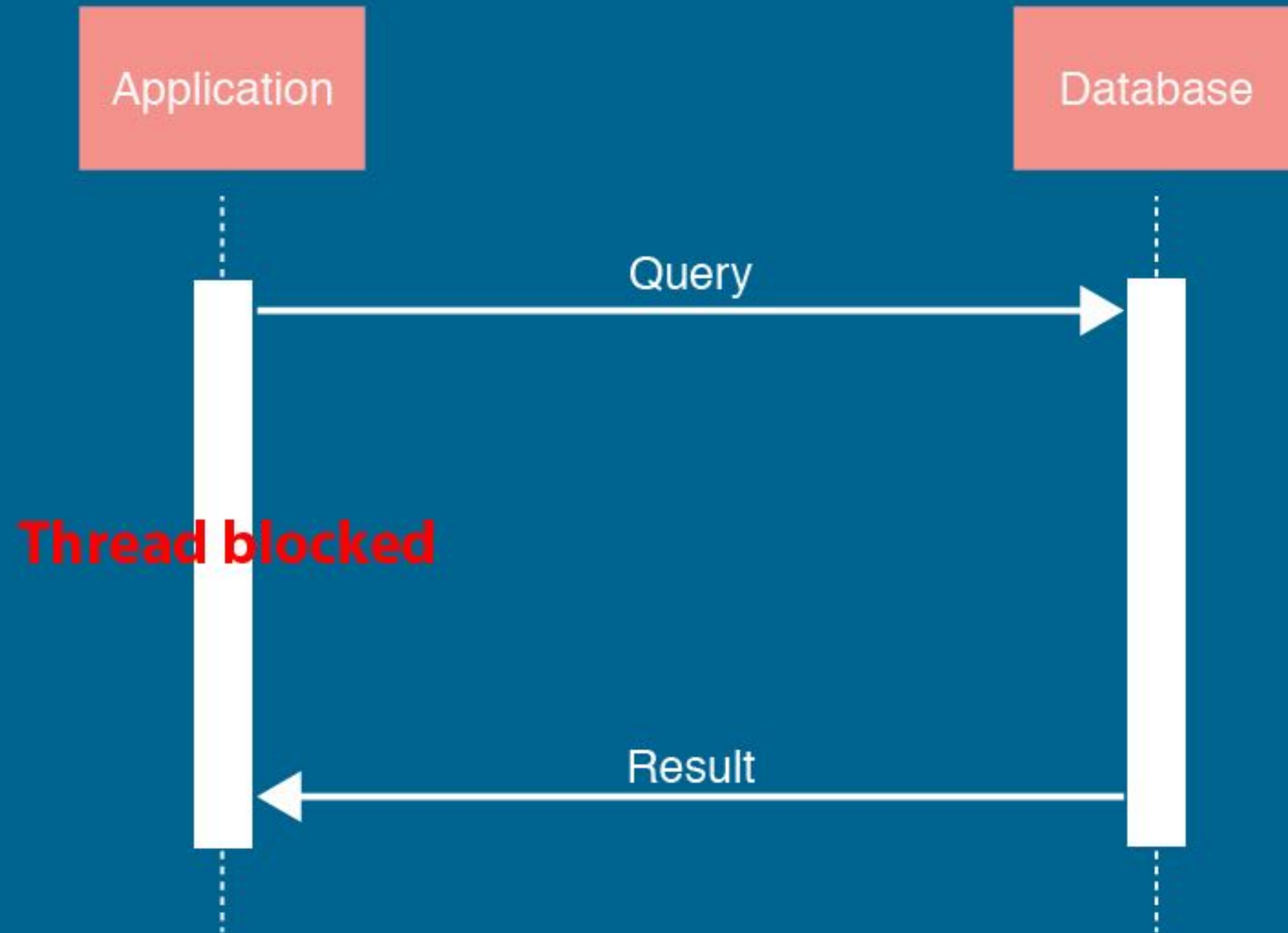
[@erwindeg](https://twitter.com/erwindeg)



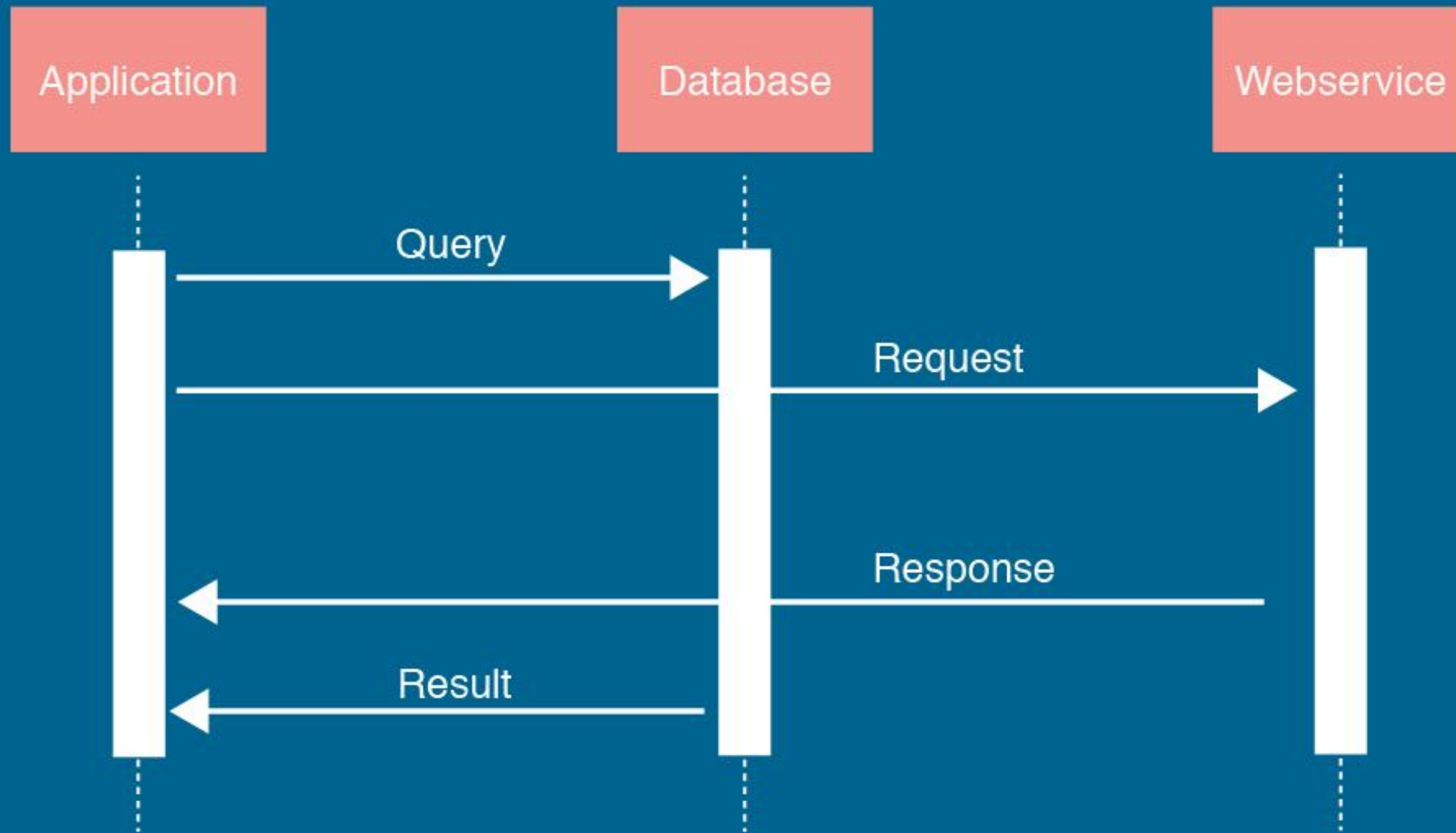
[edegier.nl](https://edegier.nl)



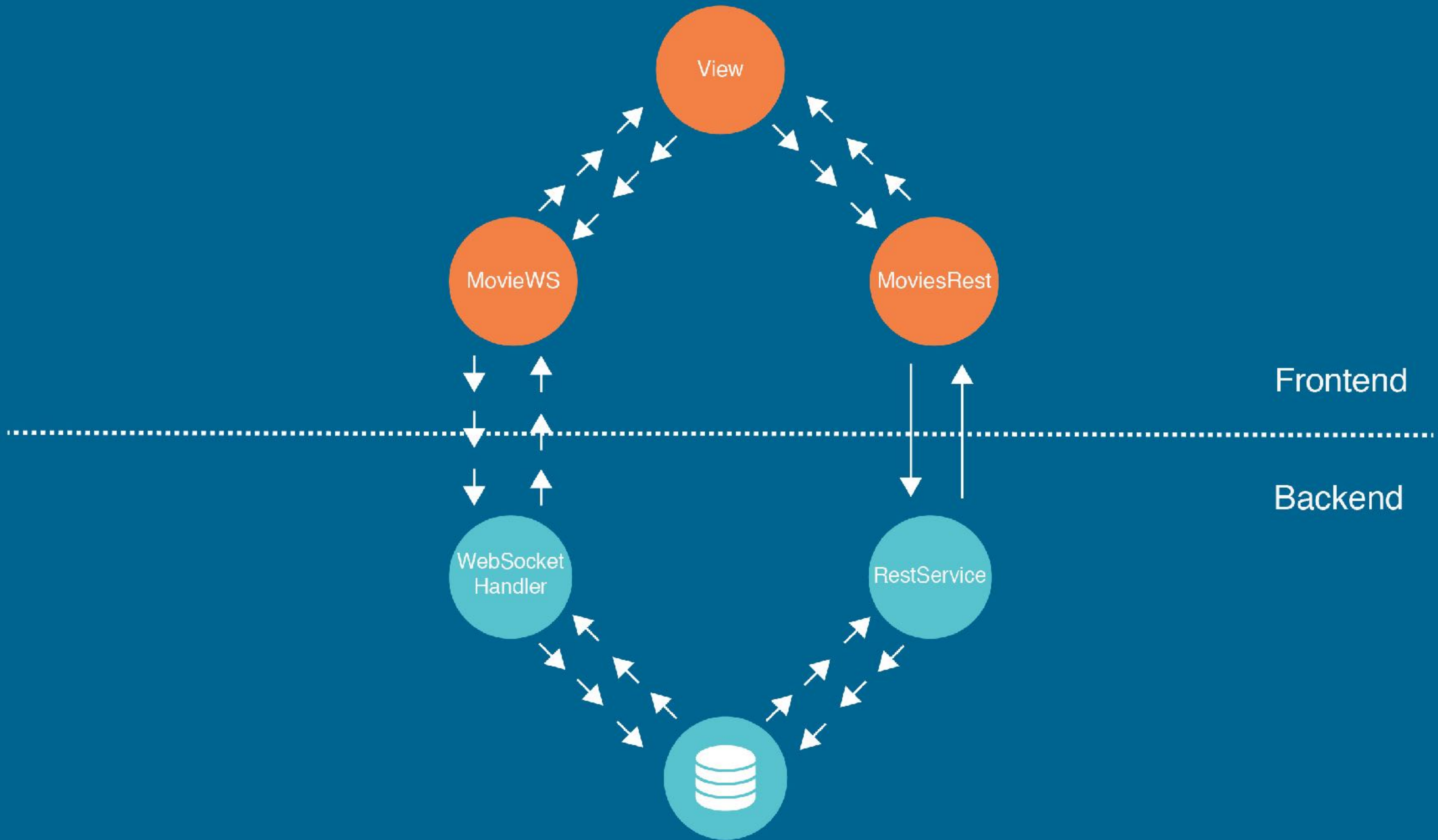
# Synchronous



# Asynchronous



**Demo**



VERT.X

 ANGULAR

 mongoDB



*//PersonRepository sync*

```
public List<Person> findByName(String name);
```

```
public BigDecimal getIncome(String name);
```

*//PersonRepository async*

```
public void findByName(String name, Callback<List<Person>> persons);
```

```
public void getIncome(String name, Callback<BigDecimal> income);
```

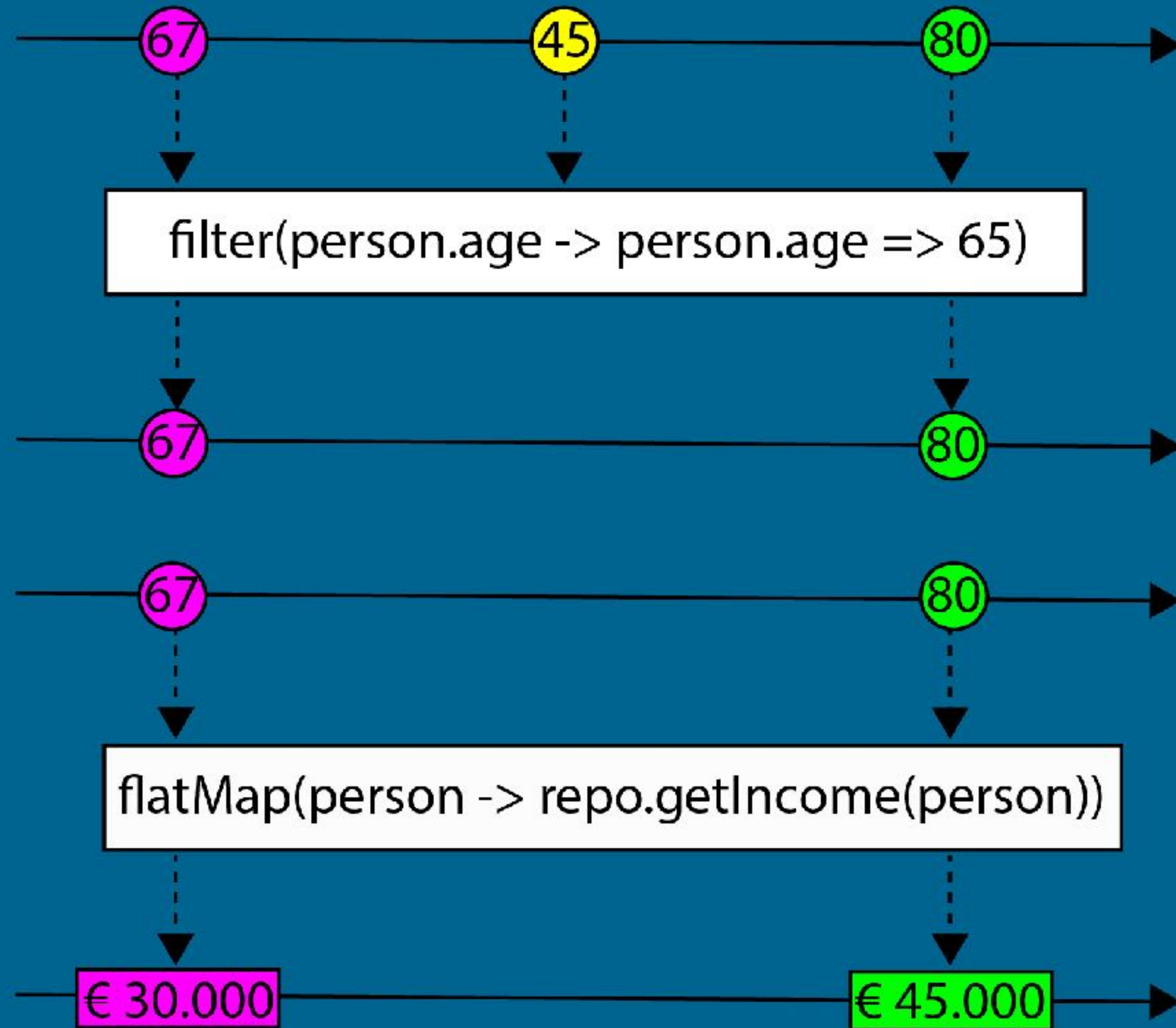
```
//Client call
repository.findByName("Erwin",
    persons -> {
        persons.stream().filter(person -> person.getAge() >= 65)
            .forEach(person -> {
                repository.getIncome(person, income ->
                    totalIncome = totalIncome.add(income));
            });
    }
);
```



```
//PersonRepository
public Observable<Person> findByName(String name);
public Observable<BigDecimal> getIncome(Person person);

//Client call
repository.findByName("Erwin")
    .filter(person -> person.getAge() >= 65)
    .flatMap(person -> repository.getIncome(person))
    .subscribe(income -> totalIncome = totalIncome.add(income));
```

# RxJava





# VERT.X

*“Vert.x is a toolkit for creating reactive applications on the JVM, supporting multiple development languages simultaneously.”*

# VERT.X

✔ **Runnable Jar**

✔ **Reactive**

✔ **Polyglot**

✔ **Distributed**





HTTP2

Websockets

Auth

REST

Integration

Metrics

Event bus

Redis

Mongo

JDBC

Zookeeper

Hazelcast

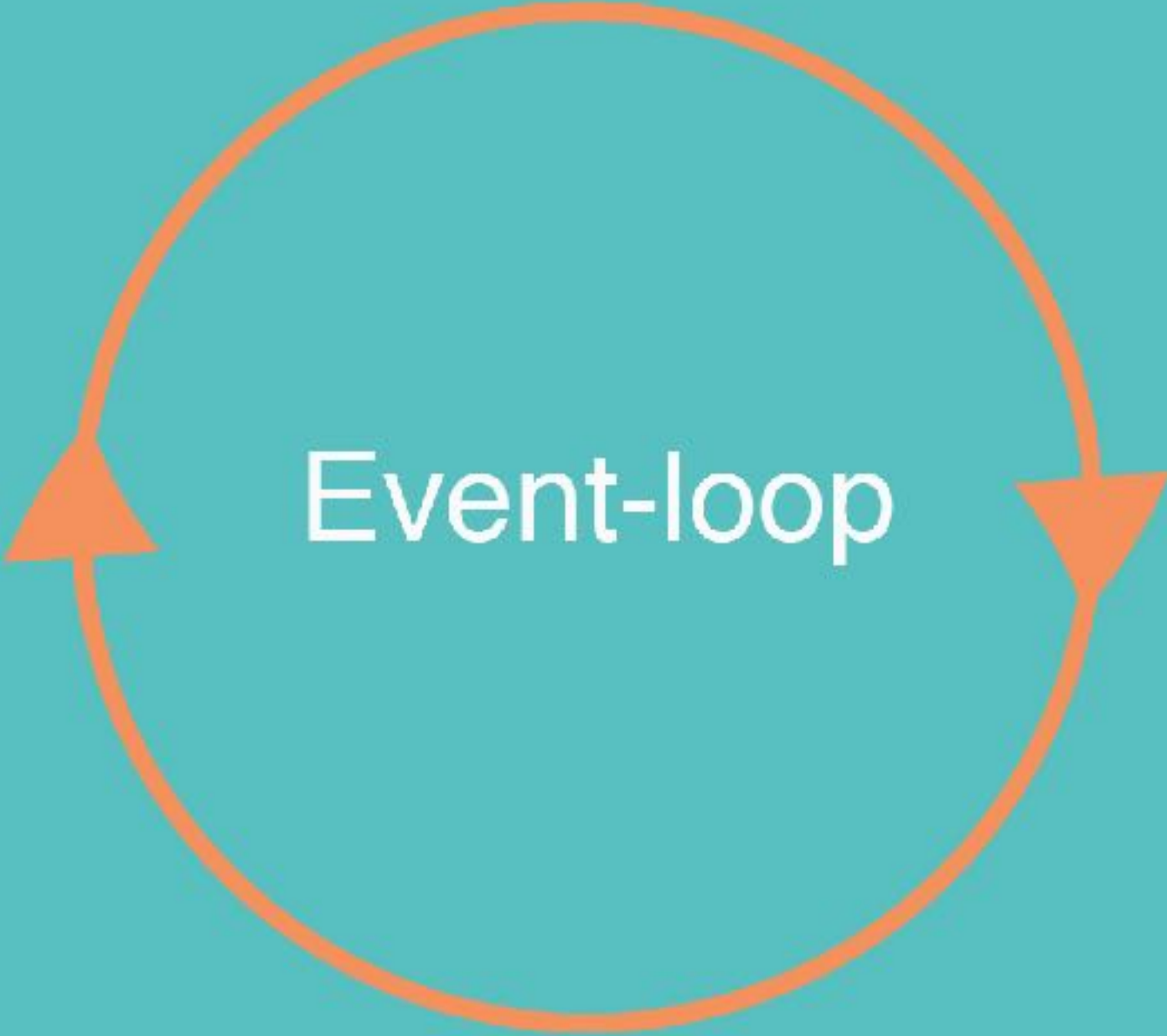
Kubernetes

Non blocking single-threaded

Event

Event

Event



Verticle



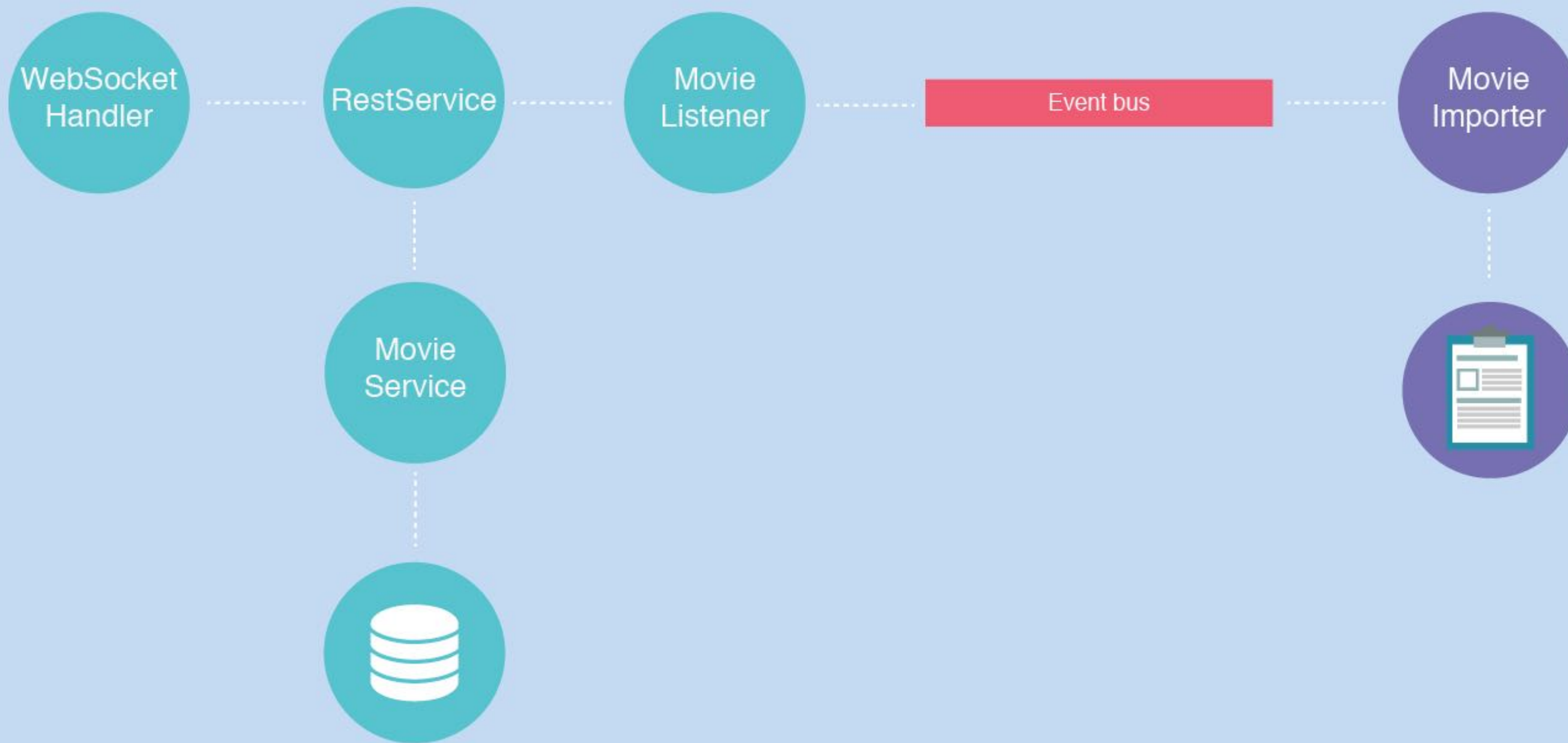
Worker Verticle

Worker Verticle

Blocking multi-threaded

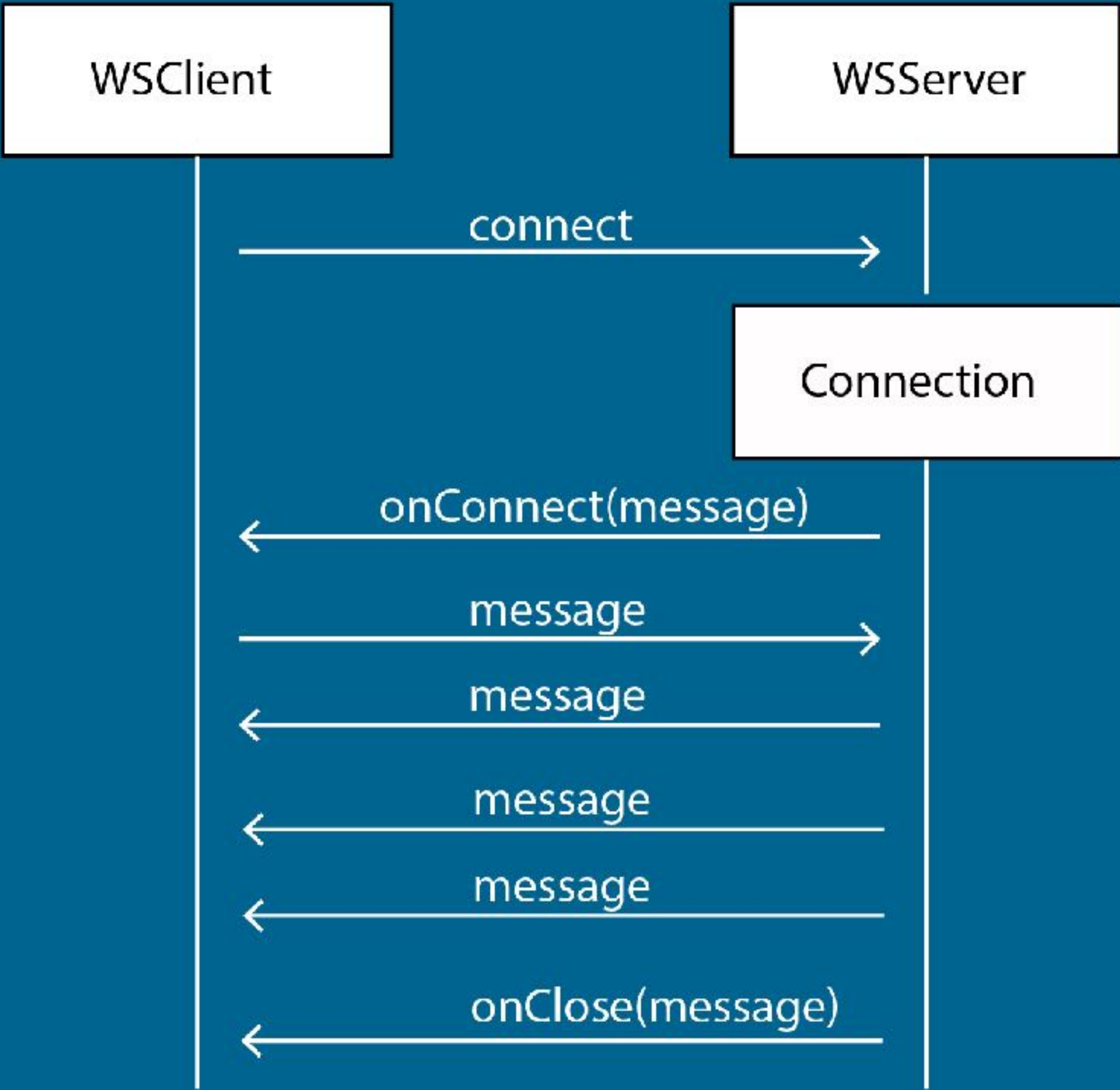








```
1 package io.vertx.starter.helloworld;
2
3 import io.vertx.core.AbstractVerticle;
4 import io.vertx.core.Vertx;
5
6
7 public class HelloWorldVerticle extends AbstractVerticle{
8
9     @Override
10    public void start() throws Exception {
11        vertx.eventBus().consumer("hello-channel", message -> System.out.println(message.body()));
12
13        vertx.eventBus().send("hello-channel", "Hello world!");
14    }
15
16    public static void main(String[] args) {
17        Vertx.vertx().deployVerticle(new HelloWorldVerticle());
18    }
19 }
20 |
```





```
public class MyWebSocketHandler implements Handler<ServerWebSocket> {

    @Override
    public void handle(ServerWebSocket ws) {
        ws.writeTextMessage(new JsonObject().put("status", "connected").encode());
        ws.textMessageHandler(message -> {
            System.out.println(message);
            ws.writeTextMessage(new JsonObject().put("message", "hello world!").encode());
        });
    }
}
```