

# Krzysztof Owsiany

## MrDev.pl

---

After.conf



MrDeVlog

---

## SzkolaEventStormingu.pl

## Thenv.pl

**Wiele odstępów CQRSA**

```
public class DirtyClass {  
    private bool _clean;  
    private int _cleanCount = 0;  
  
    public bool CleanAndCheck() {  
        _clean = true;  
        return _clean;  
    }  
  
    public int UpdateAndGetCleanCount() {  
        return ++_cleanCount;  
    }  
}
```



```
public class DirtyClass {  
    private bool _clean;  
    private int _cleanCount = 0;  
  
    public bool CleanAndCheck() {  
        _clean = true;  
        return _clean;  
    }  
  
    public int UpdateAndGetCleanCount() {  
        return ++_cleanCount;  
    }  
}
```







© 1986 Bertrand Meyer

© 1986 Bertrand Meyer

Command 

© 1986 Bertrand Meyer

Command 

Query 



© 1986 Bertrand Meyer

Command 

Query 

Separation



```
public class CleanClass {  
    private bool _clean;  
    private int _cleanCount = 0;  
  
    public void Clean() {  
        _clean = true;  
    }  
  
    public void UpdateCount() {  
        _cleanCount++;  
    }  
  
    public bool IsClean => _clean;  
  
    public int CleanCount => _cleanCount;  
}
```



# Command Query Separation

# Command Query Separation

## Clean Code





# Command Query Separation

Clean Code

Low Coupling



# Command Query Separation

Clean Code

Low Coupling

High Cohesion



# Command Query Separation

Clean Code

Low Coupling

High Cohesion

Tests

# Command Query Separation

Clean Code

Low Coupling

High Cohesion

Tests

Team work

```
public class ComplexCleanClass {  
    private bool _clean;  
    private int _cleanCount = 0;  
  
    public void Clean() { _clean = true; }  
    public void Dirty() { _clean = false; }  
    public void CleanOnlyEverySecond() { if (IsSecond) Clean(); }  
    public void UpdateCleanCount() { _cleanCount++; }  
    public void UpdateCleanCount(int step) { _cleanCount += step; }  
    public void ResetCleanCount() { _cleanCount = 0; }  
  
    public bool IsDirty => _clean == false;  
    public bool IsClean => _clean == true;  
    public bool IsSecond => _cleanCount % 2 == 0;  
}
```



```
public class ComplexCleanClass {  
    private bool _clean;  
    private int _cleanCount = 0;  
  
    public void Clean() { _clean = true; }  
    public void Dirty() { _clean = false; }  
    public void CleanOnlyEverySecond() { if (IsSecond) Clean(); }  
    public void UpdateCleanCount() { _cleanCount++; }  
    public void UpdateCleanCount(int step) { _cleanCount += step; }  
    public void ResetCleanCount() { _cleanCount = 0; }  
  
    public bool IsDirty => _clean == false;  
    public bool IsClean => _clean == true;  
    public bool IsSecond => _cleanCount % 2 == 0;  
}
```





# Greg Young & Udi Dahan



Greg Young & Udi Dahan

Command 

Greg Young & Udi Dahan

Command 

Query 

**Greg Young & Udi Dahan**

**Command** 

**Query** 

**Responsibility**

**Greg Young & Udi Dahan**

**Command** 

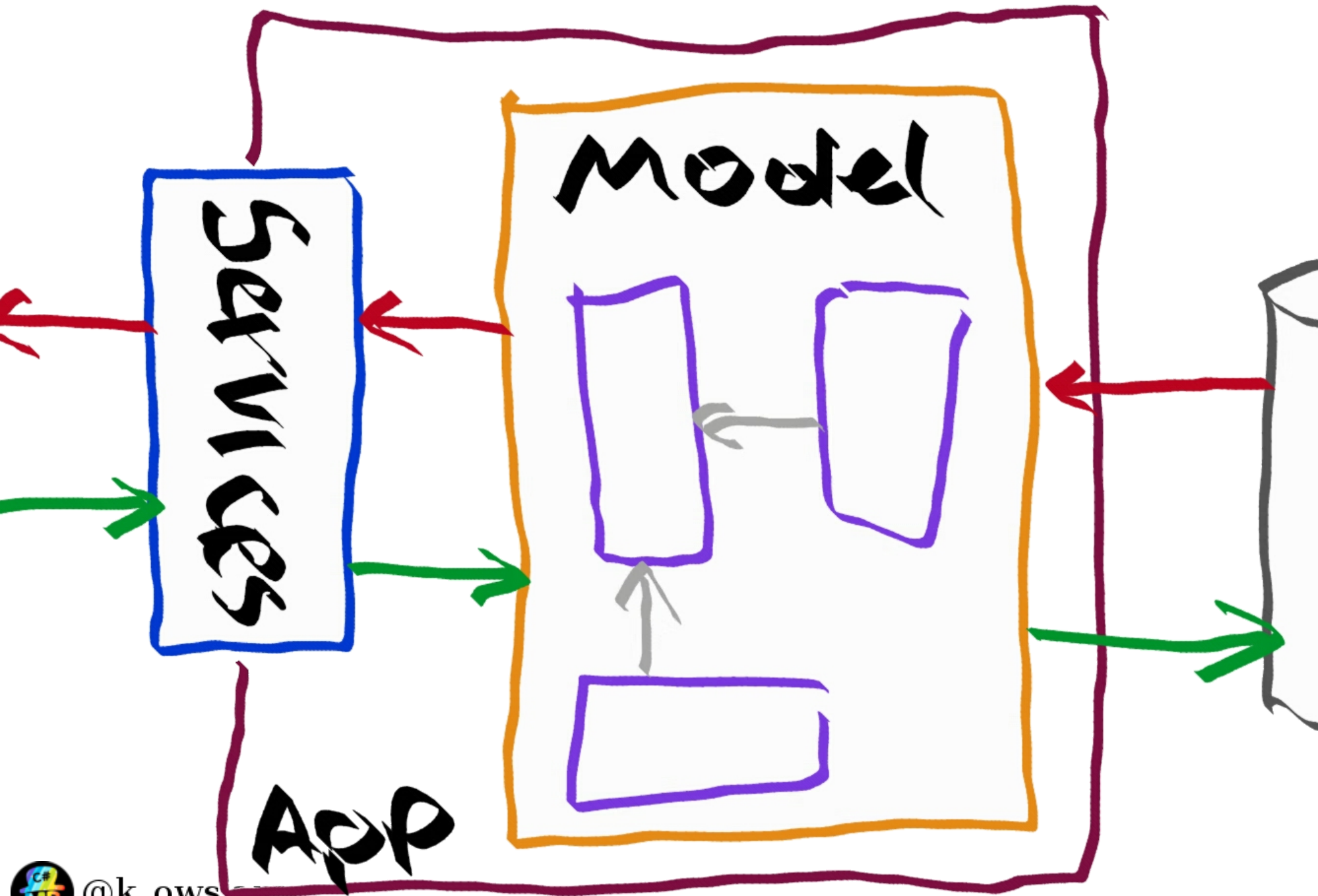
**Query** 

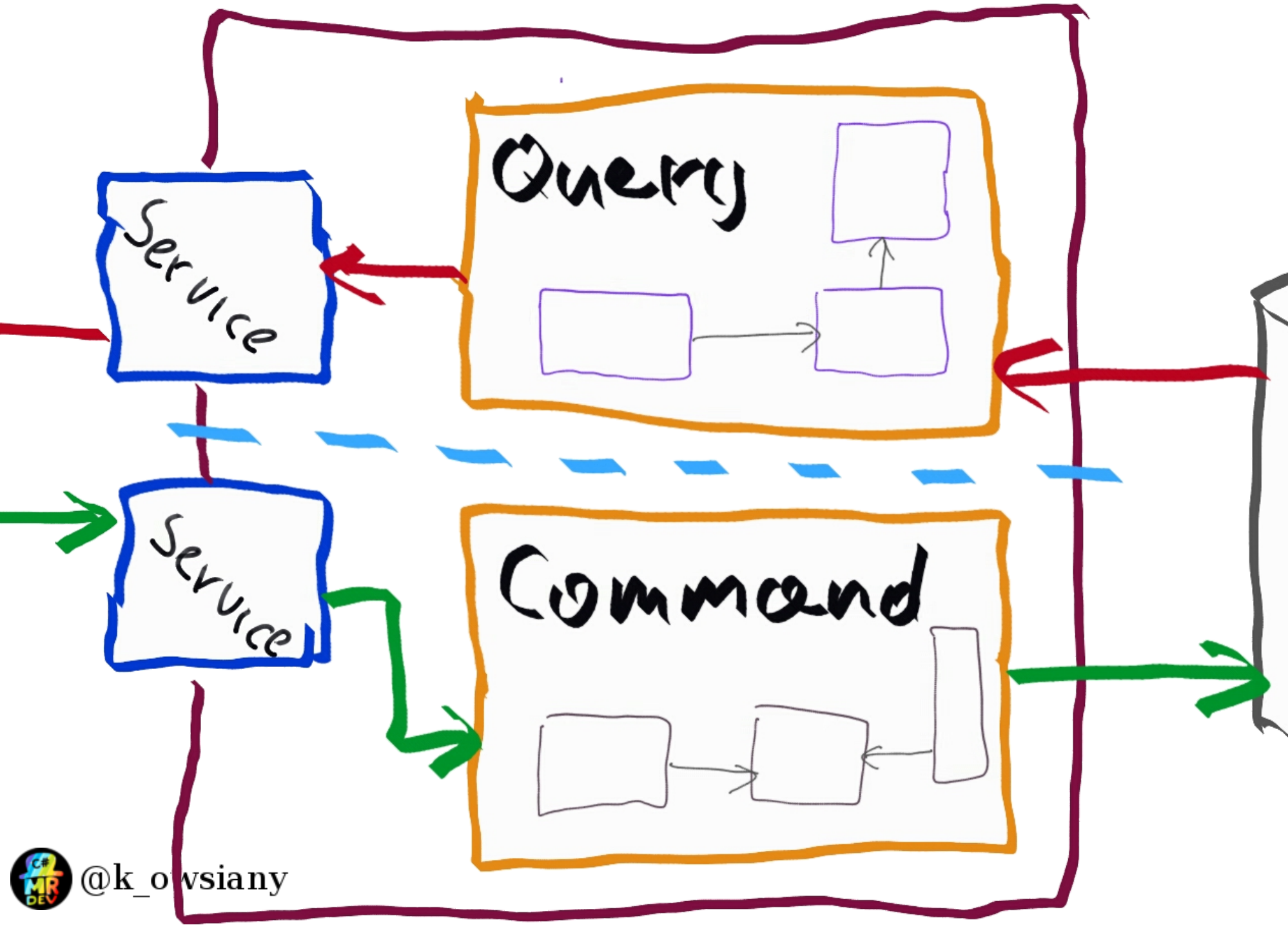
**Responsibility**

**Segregation**



@k\_owsiany









# Command





# Command Handler



@k\_owsiany



**Command  
Handler**



**Query**



@k\_owsiany

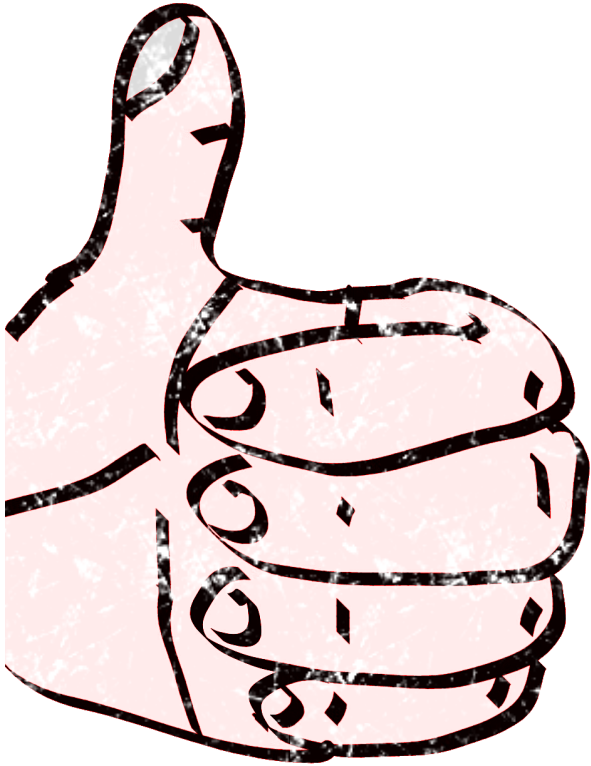


**Command  
Handler**

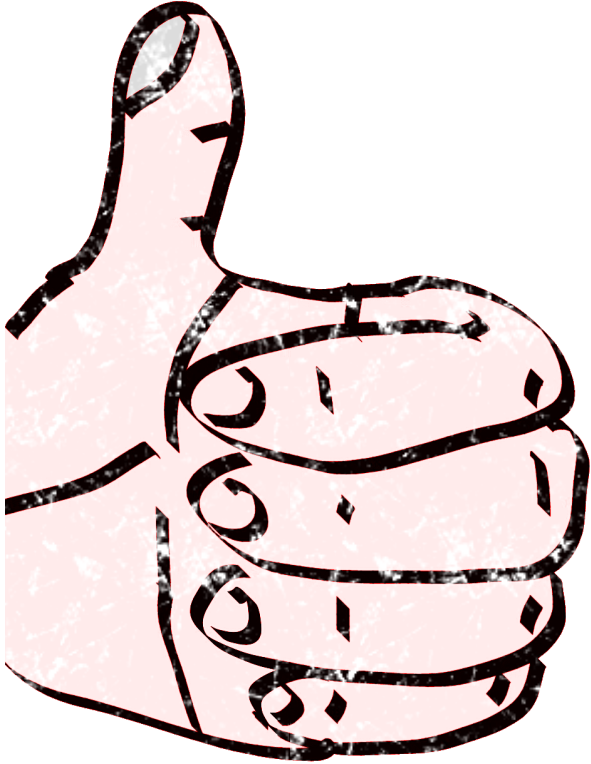


**Query  
Handler**

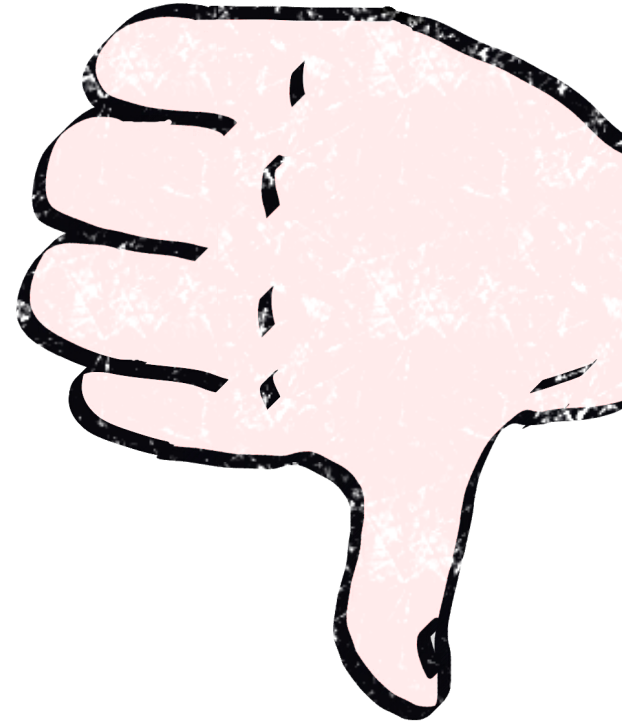
# CORS



**CQRS**



**CQRS**



# EVENT STORMING

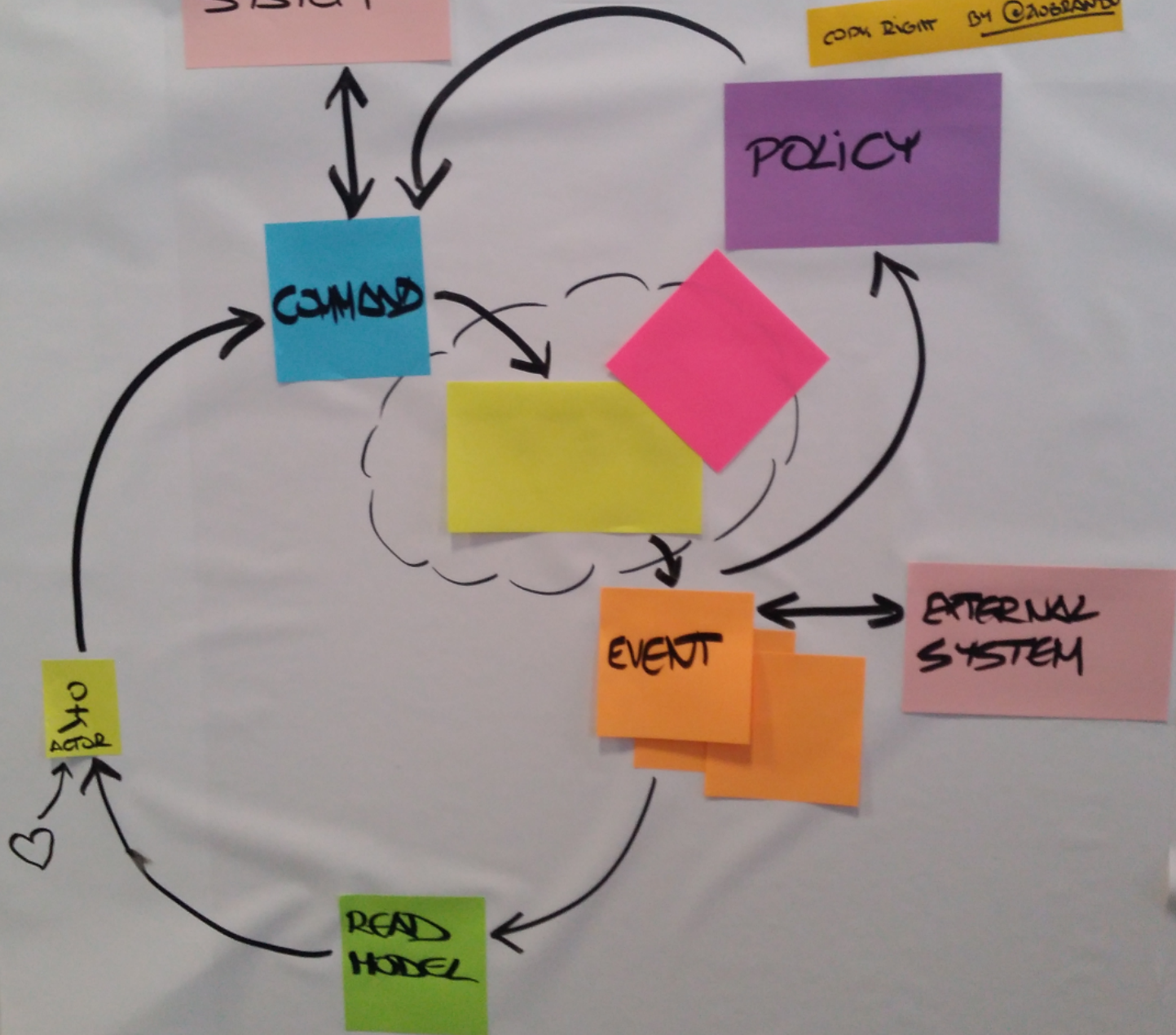
by Ziobrand



@k\_owsiany

PICTURE THAT EXPLAINS  
(ALMOST) EVERYTHING

COPY RIGHT BY @210BRAND





PICTURE THAT EXPLAINS  
(ALMOST) EVERYTHING

COPY RIGHT BY @20BRAND0

EXTERNAL  
SYSTEM

POLICY

COMMAND

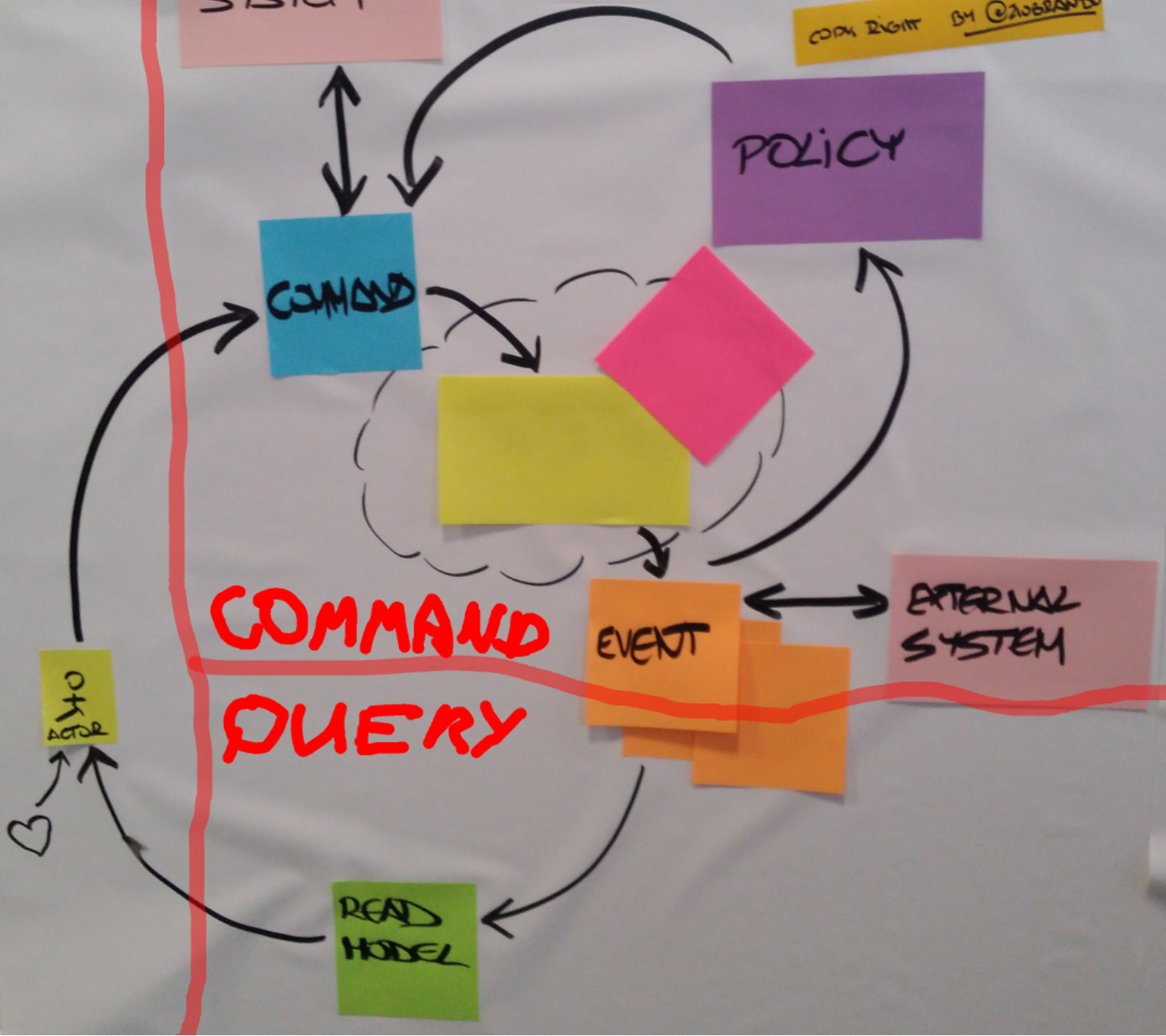
COMMAND  
QUERY

EVENT

EXTERNAL  
SYSTEM

READ  
MODEL

OUT  
SIDE



DDD

## Strategiczne DDD

API

DB



## Strategiczne DDD

API

CQRS

DB



@k\_owsiany

# Strategiczne DDD

API

CQRS

Queries

DB



# Strategiczne DDD

API

CQRS

DB

Queries

Commands



@k\_owsiany

# Strategiczne DDD

API

CQRS

DB

Queries

Commands

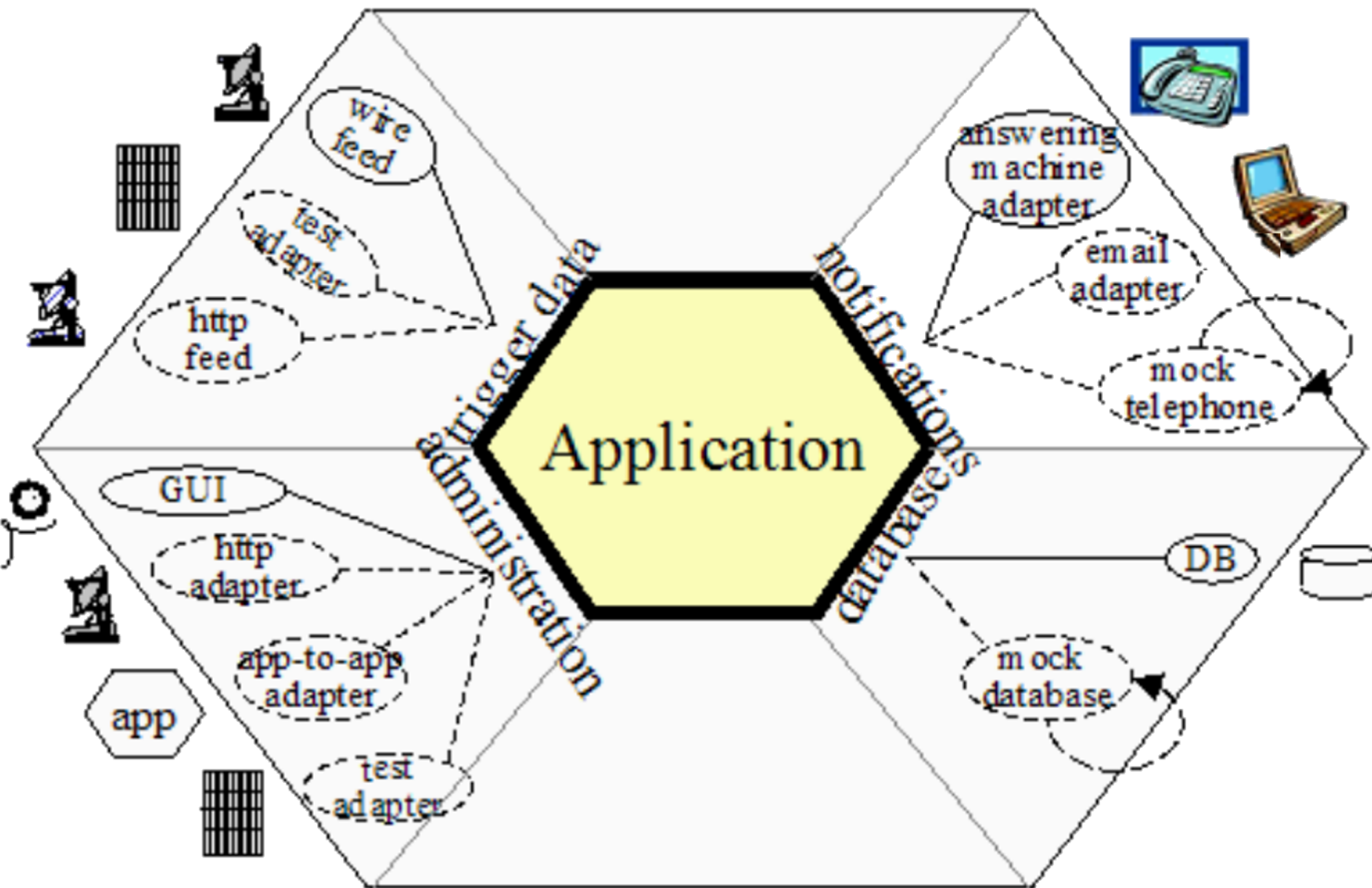
Taktyczne DDD



# Ports & Adapters

Dr. Alistair Cockburn

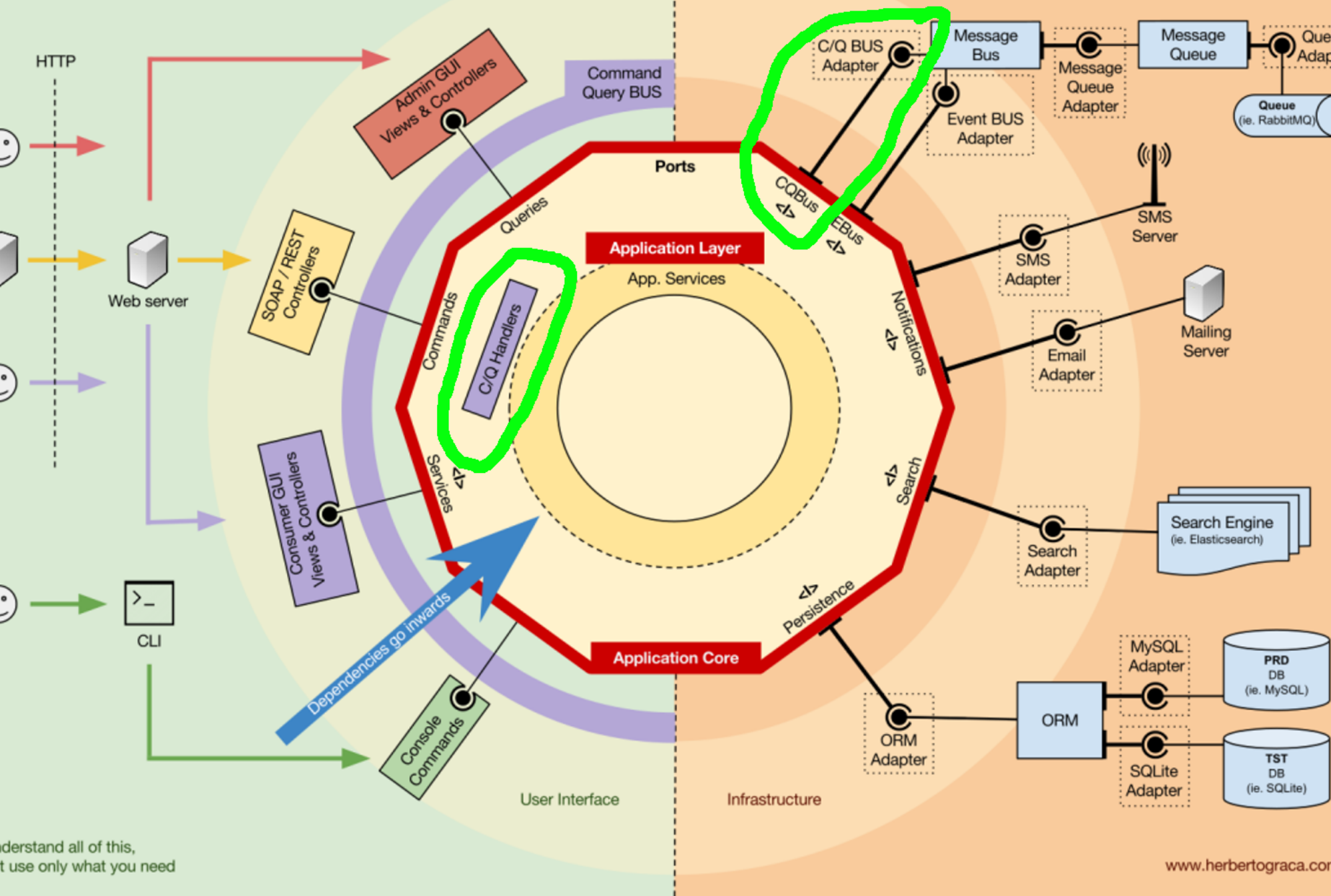




**Herberto Graça**

**herbertograca.com**

## Primary/Driving Adapters





**Command  
Handler**



**Query  
Handler**



**Command  
Handler  
Bus**



**Query  
Handler**



@k\_owsiany



**Command  
Handler  
Bus**



**Query  
Handler  
Bus**



@k\_owsiany

# Swój własny CORS?

es.io

Search ▼

cqrs



or the libraries you already use, from the project maintainers. [Request a demo.](#)

7K packages

Sort: **Relevance** ▼

License

vascript

se 3.2.1-pre - Updated Feb 18, 2018 - 231 stars

ght enterprise framework to write CQRS, event-sourced and  
vice applications in...

- ☐ MIT 750 projects
- ☐ Apache-2.0 20
- ☐ BSD-3-Clause
- ☐ ISC 37 projects
- ☐ LGPL-2.1 36 p
- ☐ Other 34 projects
- ☐ MPL-2.0 28 pro
- ☐ GPL 2.0 22



op faster and manage open source risks with the Tidelift Subscription. [Learn more](#)

ackages

Sort: **Relevance** ▼

License

cript

.2.1-pre - Updated Feb 18, 2018 - 231 stars

enterprise Function as a Service (FaaS) framework to write  
d serverles...

- ☐ MIT 888 projects
- ☐ Apache-2.0 25
- ☐ BSD-3-Clause
- ☐ ISC 48 projects
- ☐ GPL-3.0 45 pro
- ☐ LGPL-2.1 36 p
- ☐ Other 36 projec

# JavaScript

wolkenkit

nestjs/cqrs

PHP

prooph

lifecycle-php

honeybee

**Ruby**

**sequent**

**rcqs**

Go

rabbit

eventhorizon

rethinkdb

# Python

## recall

### cq

**JAVA**

**axon**

**jaster**

# Pascal

?

?



**CSharp**

**EventFlow**

**MediatR**

**SimpleCQRS**



@k\_owsiany

```
public interface IQuery<TResult>
{
}
```

```
public interface IQueryHandler<in TQuery, TResult>
    where TQuery : IQuery<TResult>
{
    TResult Execute(TQuery query);
}
```

```
public interface IQueryBus
{
    TResult Process<TQuery, TResult>(TQuery query)
        where TQuery: IQuery<TResult>;
}
```



```
public class GetDataQuery : IQuery<string>
{
    public GetDataQuery(string text)
    {
        Text = text;
    }
    public string Text { get; private set; }
}
```

```
public class GetDataQueryHandler : IQueryHandler<GetDataQuery, string>
{
    ...
    public string Execute(GetDataQuery query)
    {
        return _dataRepository.Get(query.Text);
    }
}
```



```
public class QueryBus : IQueryBus
{
    ...
    public TResult Process<TQuery, TResult>(TQuery query)
        where TQuery : IQuery<TResult>
    {
        var handler = _container.Resolve<IQueryHandler<TQuery, TResult>>();

        if (handler == null)
        {
            throw new Exception($"Not found handler for Query:
                                '{query.GetType().FullName}'");
        }

        return handler.Execute(query);
    }
}
```



```
public interface ICommand
{
}
```

```
public interface ICommandHandler<in TCommand>
    where TCommand : ICommand
{
    void Handle(TCommand command);
}
```

```
public interface ICommandBus
{
    SendCommand<TCommand>(TCommand command)
        where TCommand: ICommand;
}
```



```
public class DeleteDataCommand : ICommand
{
    public string Key { get; private set; }

    public DeleteDataCommand(string key)
    {
        Key = key;
    }
}
```

```
public class DeleteDataCommandHandler : ICommandHandler<DeleteDataCommand>
{
    ...
    public void Handle(DeleteDataCommand command)
    {
        _dataRepository.Delete(command.Key);
    }
}
```



```
public class CommandBus : ICommandBus
{
    ...
    public SendCommand<TCommand>(TCommand command)
        where TCommand : ICommand
    {
        var handler =
            _container.ResolveOptional<ICommandHandler<TCommand>>

        if (handler == null)
        {
            throw new Exception($"Not found handler for Command:
                                '{command.GetType().FullName}'");
        }

        handler.Handle(command);
    }
}
```



```
_commandBus.SendCommand(new DeleteDataCommand("klucz"));

_queryBus.Process(new GetDataQuery("filterek"));
```



# Swój własny CORS?

# Krzysztof Owsiany

## MrDev.pl

---

After.conf



MrDeVlog

---

## SzkolaEventStormingu.pl

## Thenv.pl



@k\_owsiany