

Krzysztof Owsiany

MrDev.pl

After.conf



MrDevlog

SzkolaEventStormingu.pl

Thenv.pl



@k_owsiany

Wiele odsłon CQRSa



@k_owsiany

```
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;
    }
}
```





@k_owsiany

© 1986 Bertrand Meyer

© 1986 Bertrand Meyer

Command



@k_owsiany

© 1986 Bertrand Meyer

Command 

Query 



@k_owsiany

© 1986 Bertrand Meyer

Command 

Query 

Separation



@k_owsiany

```
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



@k_owsiany

Command Query Separation

Clean Code

Command Query Separation

Clean Code

Low Coupling



@k_owsiany

Command Query Separation

Clean Code

Low Coupling

High Cohesion



@k_owsiany

Command Query Separation

Clean Code

Low Coupling

High Cohesion

Tests



@k_owsiany

Command Query Separation

Clean Code

Low Coupling

High Cohesion

Tests

Team work



@k_owsiany

```
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;  
}
```





@k_owsiany

Greg Young & Udi Dahan

Greg Young & Udi Dahan

Command



@k_owsiany

Greg Young & Udi Dahan

Command 

Query 



@k_owsiany

Greg Young & Udi Dahan

Command 

Query 

Responsibility



@k_owsiany

Greg Young & Udi Dahan

Command 

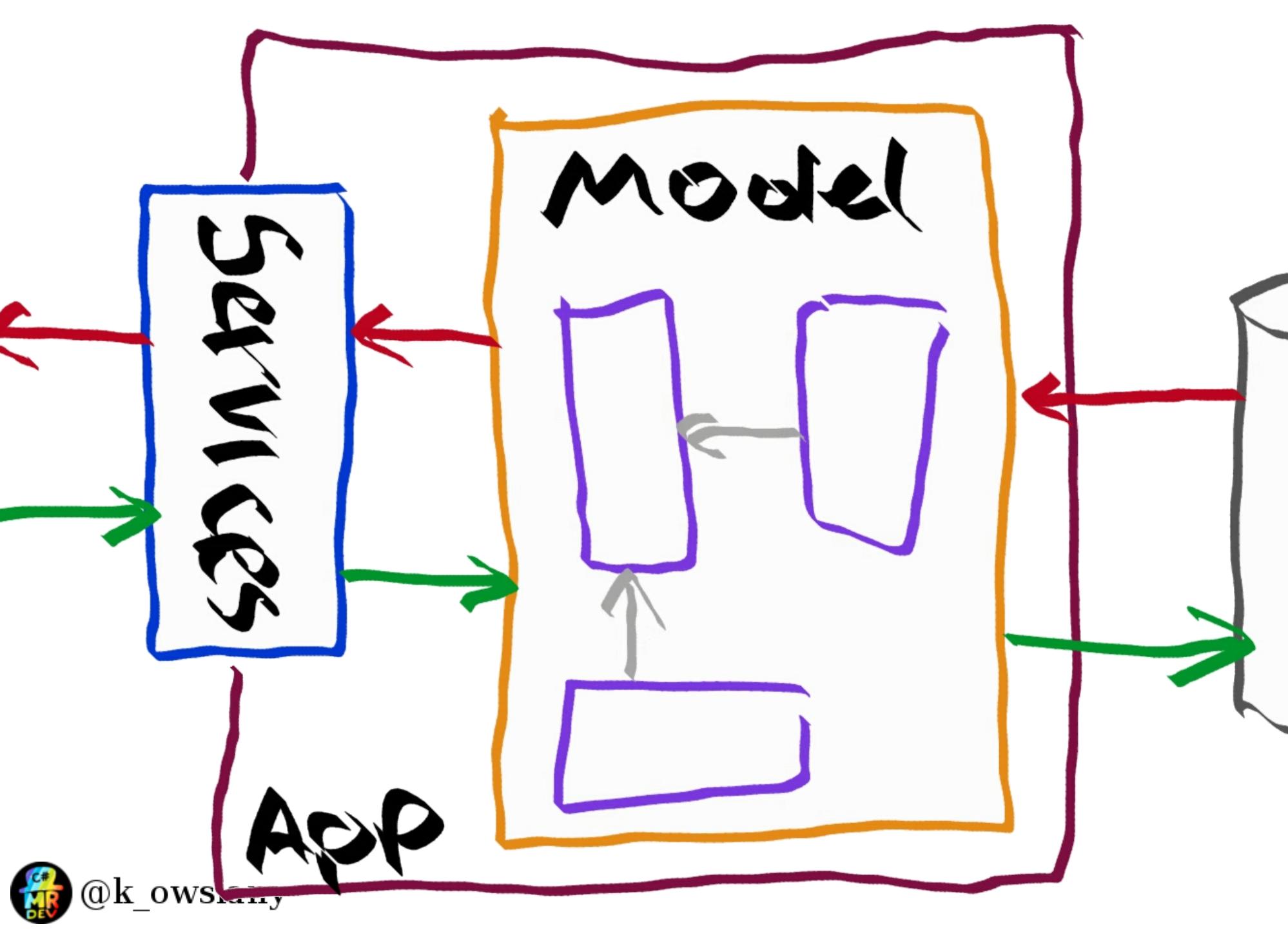
Query 

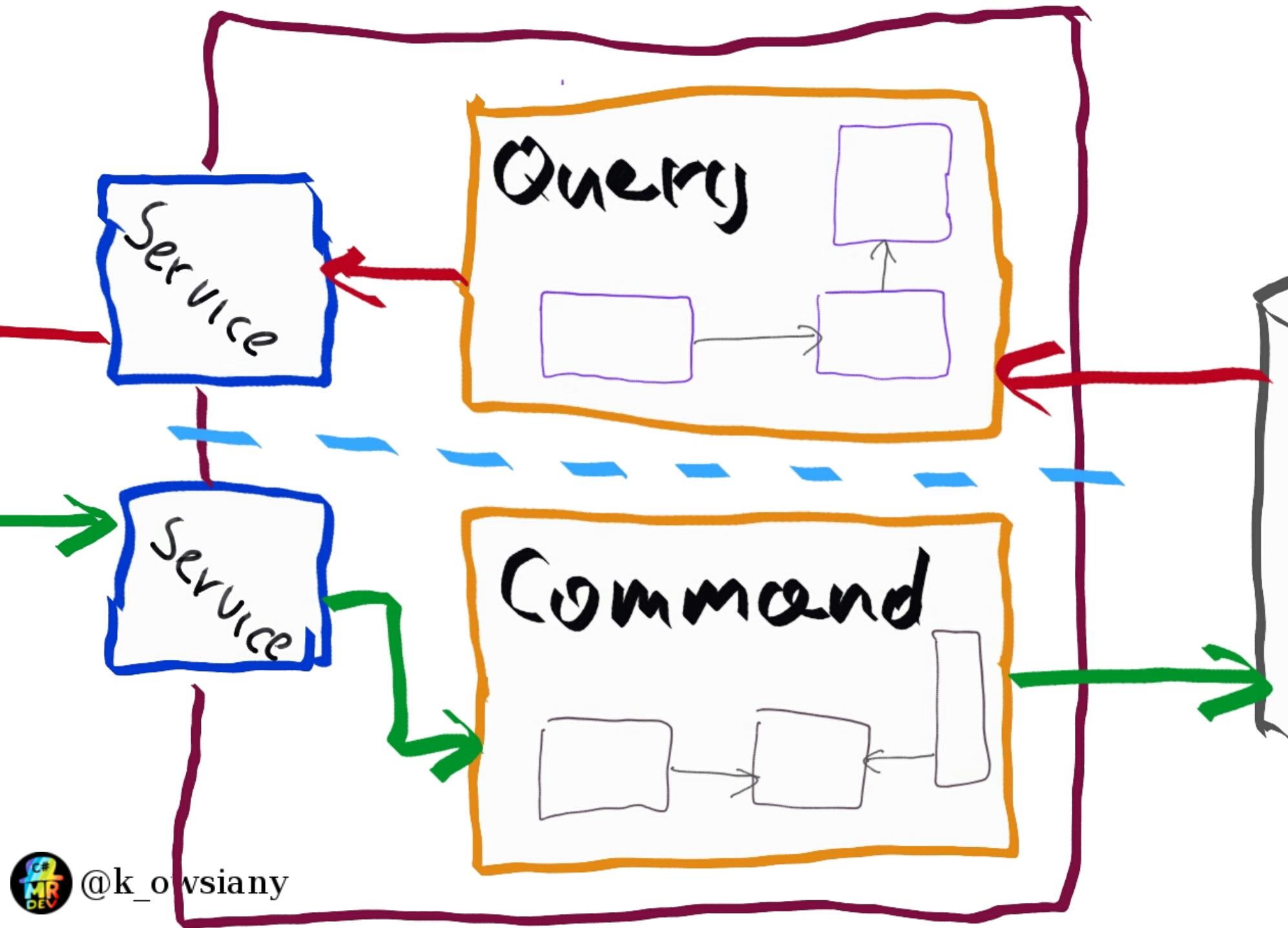
Responsibility

Segregation



@k_owsiany









Command



Command Handler



Command
Handler



Query

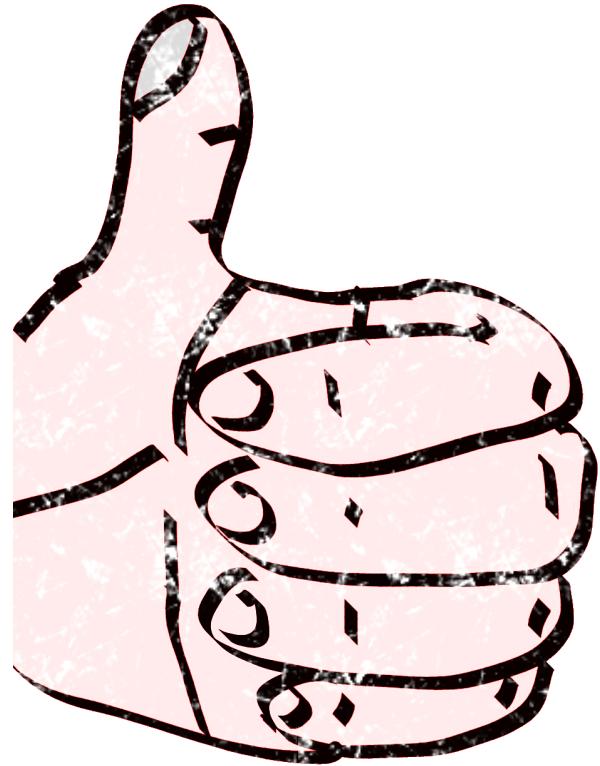


Command
Handler



Query
Handler

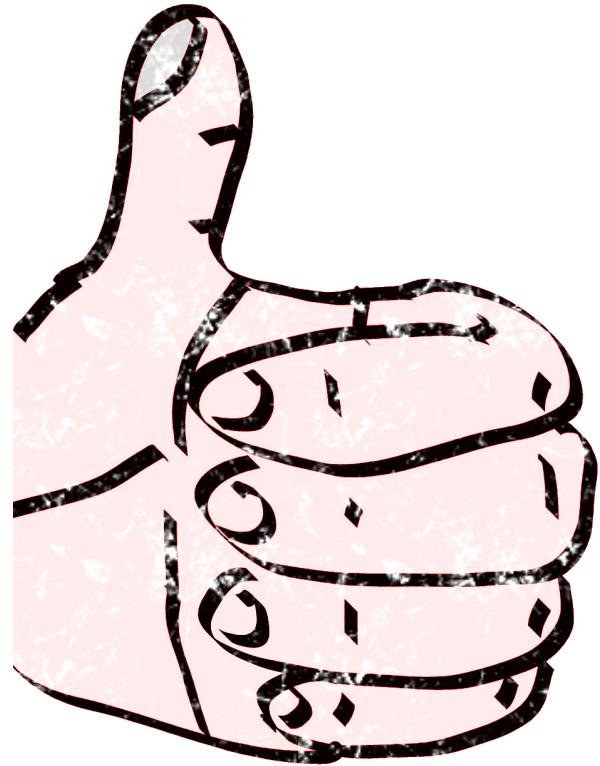
CQRS



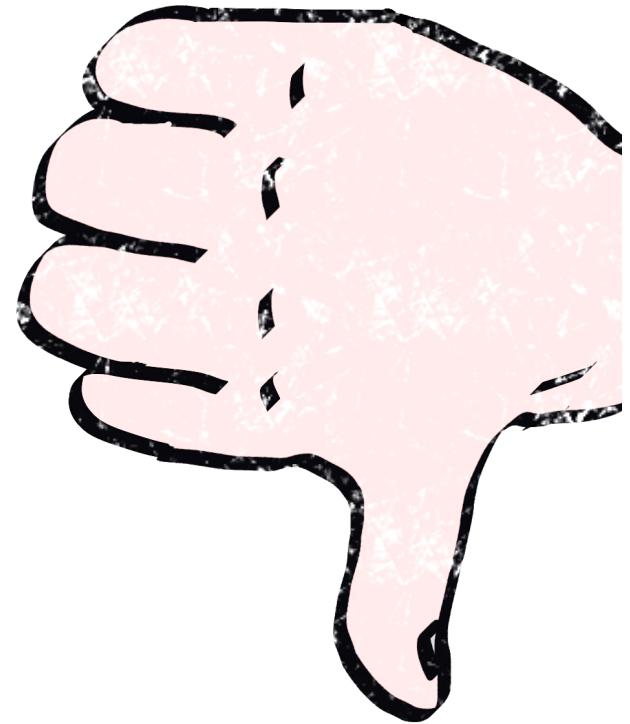
CQRS



@k_owsiany



CQRS



EVENT STORMIN

by ziobrand



@k_owsiany

PICTURE THAT EXPLAINS (ALMOST) EVERYTHING

COPYRIGHT BY @UBBRANDO



PICTURE THAT EXPLAINS (ALMOST) EVERYTHING

COPYRIGHT BY @UBBRANDO

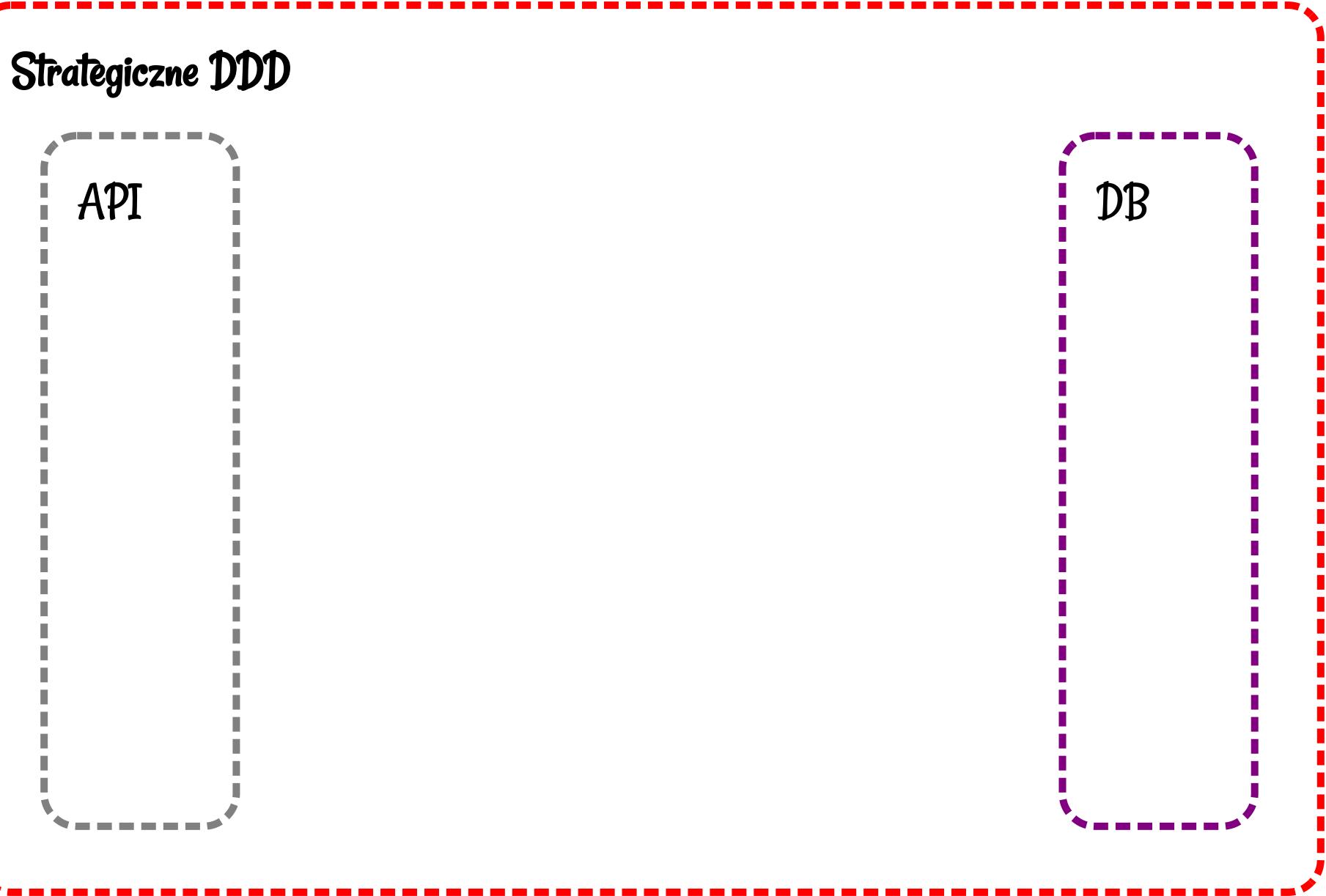


DDD

Strategiczne DDD

API

DB



Strategiczne DDD

API

CQRS

DB



Strategiczne DDD

API

CQRS

Queries

DB



Strategiczne DDD

API

CQRS

DB

Queries

Commands



Strategiczne DDD

API

CQRS

DB

Queries

Commands

Taktyczne DDD

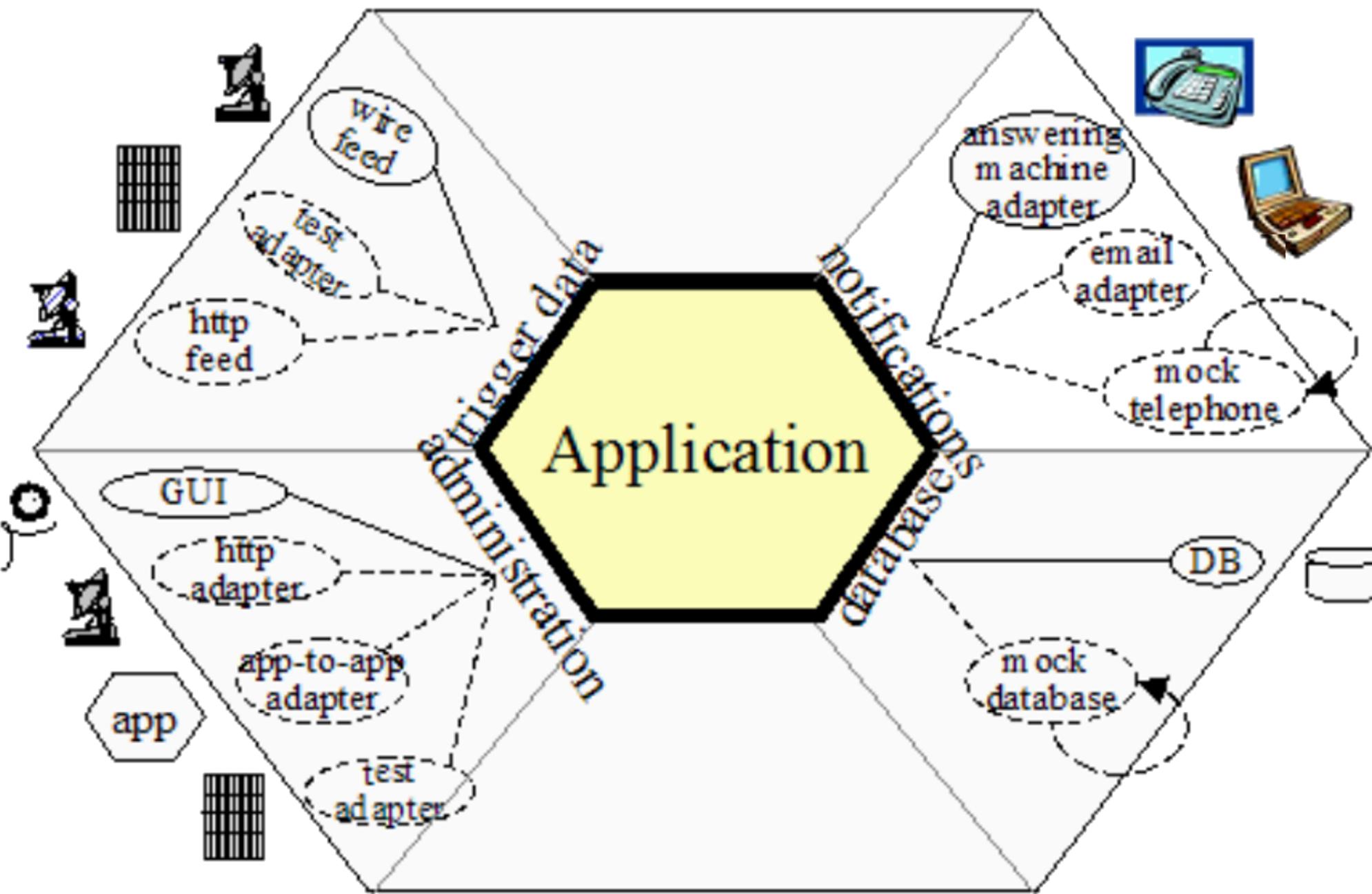


Ports & Adapters

Dr. Alistair Cockburn



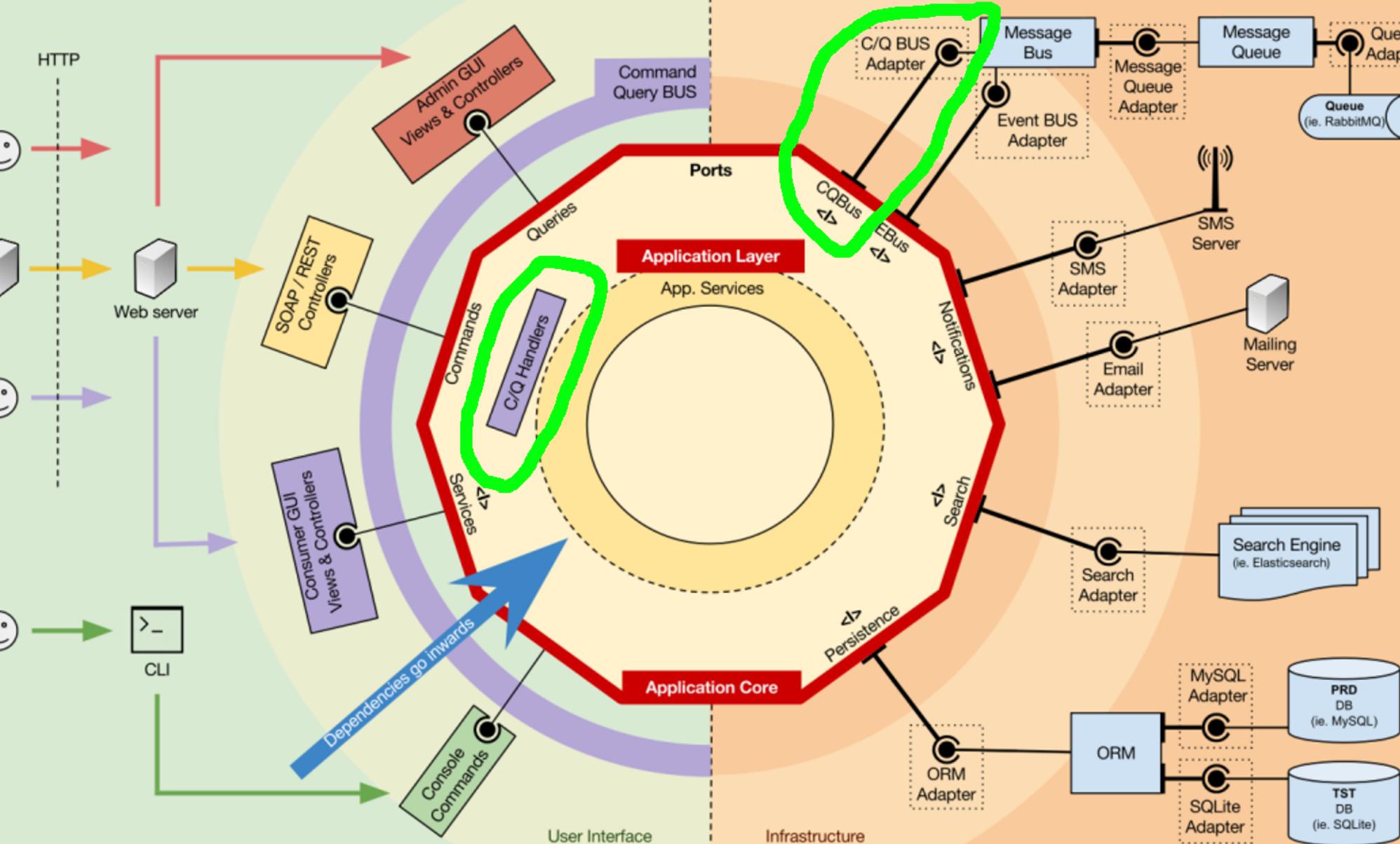
@k_owsiany



Herberto Graça

herbertograca.com

Primary/Driving Adapters



Understand all of this,
but use only what you need



Command
Handler



Query
Handler



Command
Handler

Bus



Query
Handler



Command
Handler
Bus



Query
Handler
Bus

Swój własny CQRS?



@k_owsiany

es.io

Search ▾

cqrs



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

7K packages

Sort: Relevance ▾

javascript

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

ight enterprise framework to write CQRS, event-sourced and
ice applications in...

License

- MIT 750 projects
- Apache-2.0 200 projects
- BSD-3-Clause 150 projects
- ISC 37 projects
- LGPL-2.1 36 projects
- Other 34 projects
- MPL-2.0 28 projects
- GPL-2.0 22 projects



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

ackages

Sort: **Relevance** ▾

cript

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

Enterprise Function as a Service (FaaS) framework to write
and serverless...

License

- MIT** 888 projects
- Apache-2.0** 253 projects
- BSD-3-Clause** 133 projects
- ISC** 48 projects
- GPL-3.0** 45 projects
- LGPL-2.1** 36 projects
- Other** 36 projects

JavaScript

wolkenkit

nestjs/cqrs



@k_owsiany

PHP

prooph

litecqrsphp

honeybee



@k_owsiany

Ruby

sequent

rcqrs



@k_owsiany

Go

rabbit

eventhorizon

rethinkdb



@k_owsiany

Python

recall

cq



@k_owsiany

JAVA

axon

jaster



@k_owsiany

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>;
}
```



@k_owsiany

```
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("filtererek"));
```



Swój własny CQRS?



@k_owsiany

Krzesztof Owsiany

MrDev.pl

After.conf



MrDevlog

SzkolaEventStormingu.pl

Thenv.pl



@k_owsiany