



Vacations
Visitors

MrDev.pl
After.conf
Then



Dev Advent Calendar.pl

Wiele
odslon CORS-a

@k_owsiany



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

@k_owsiany



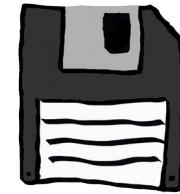
```
public class DirtyClass {  
    private bool _clean;  
    private int _cleanCount = 0;  
  
    public bool CleanAndCheck() {  
        _clean = true;  
        return _clean;  
    }  
  
    public int UpdateAndGetCleanCount() {  
        return ++_cleanCount;  
    }  
}
```

SRP

@k_owsiany



Command



Query



Separation

©1986 Bertrand Meyer

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

@k_owsiany



Command
Query
Separation

Clean Code
Testy
Pracq v responde

```
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;  
}
```

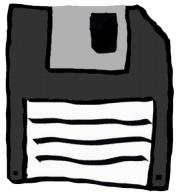
@k_owsiany

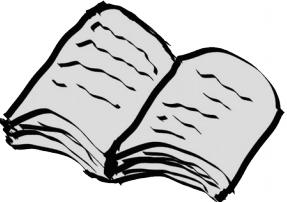


```
public class ComplexCleanClass {  
    private bool _clean;  
    private int _cleanCount = 0;  
  
    public void Clean() { _clean = true; }  
    public void Dirty() { _clean = false; }  
    public void CleanOnlyEverySecond() { if (IsSecond) Clean(); }  
    public void UpdateCleanCount() { _cleanCount++; }  
    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



Command 

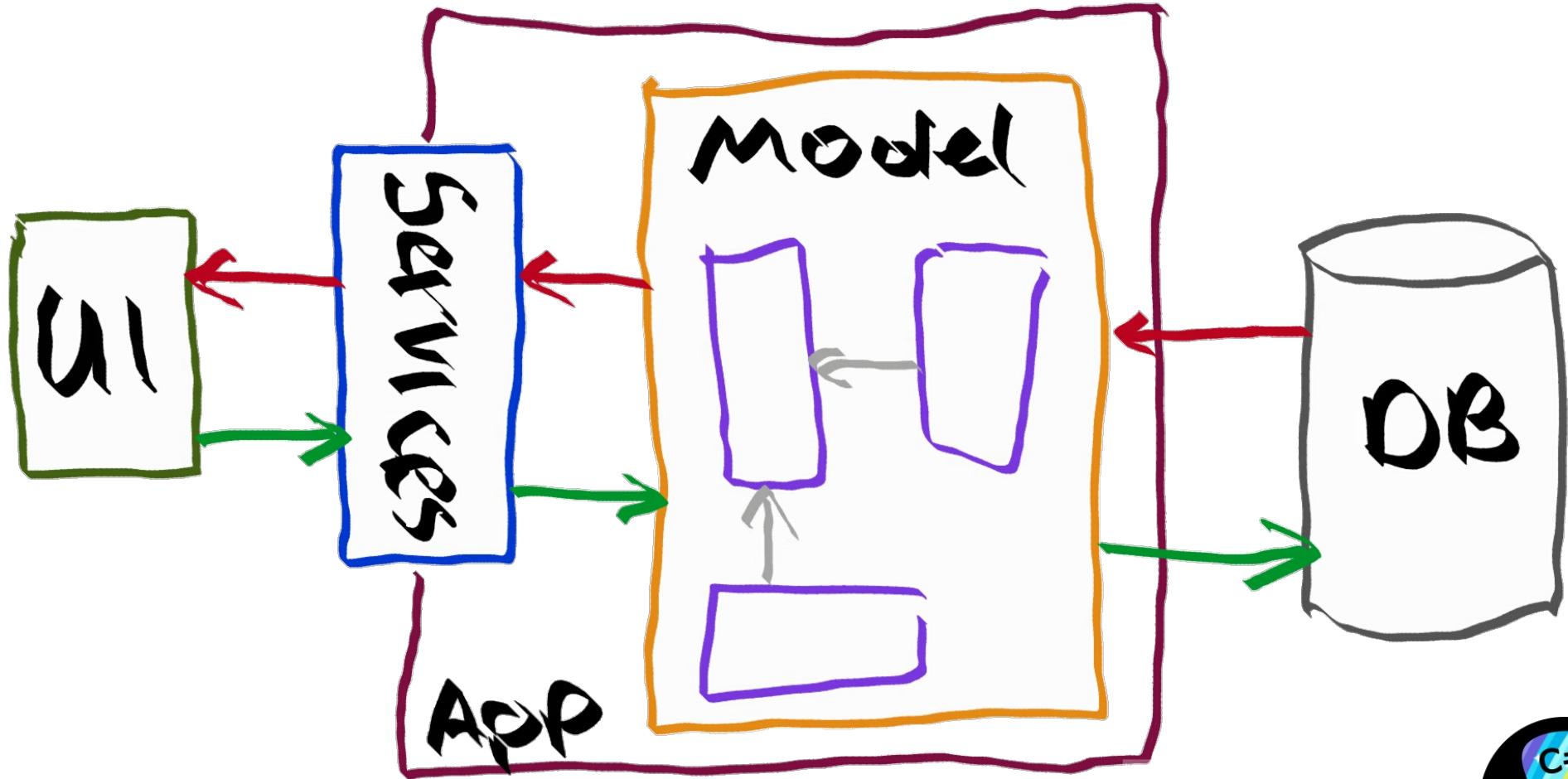
Query 

Responsibility

Segregation

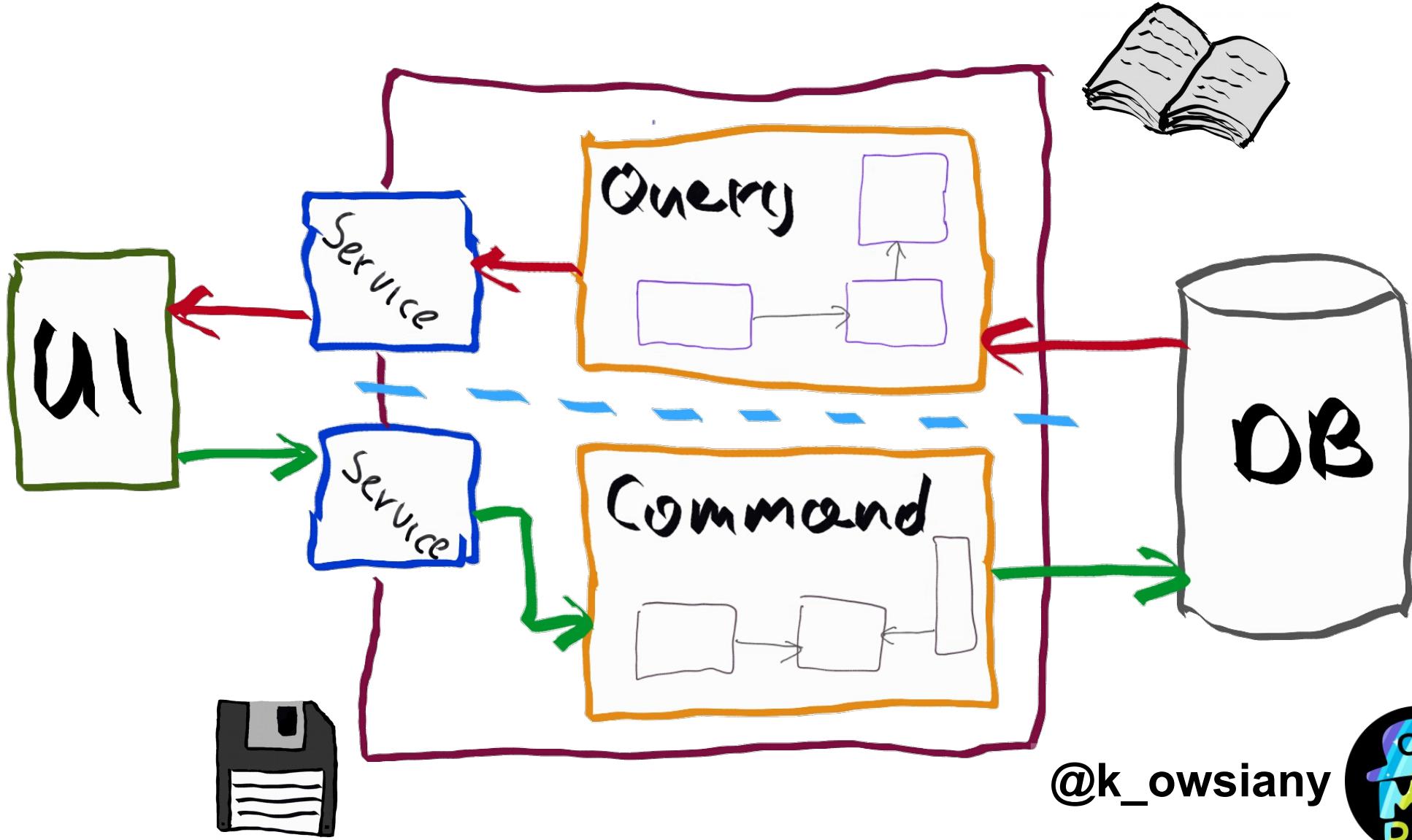
@k_owsiany





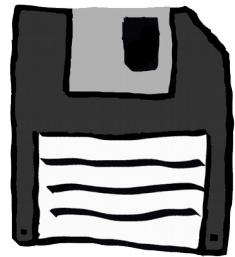
@k_owsiany





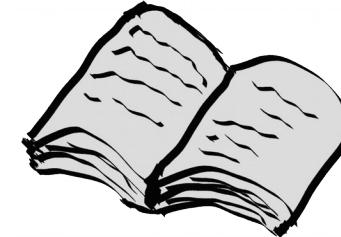
@k_owsiany





Command
Handler
Bus

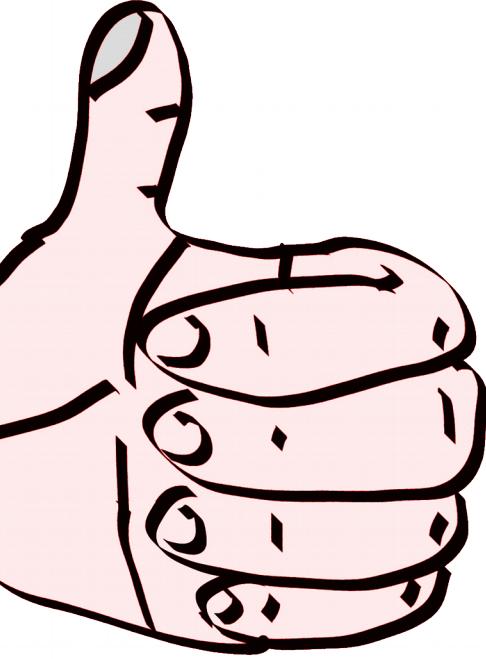
Event
Sourcing



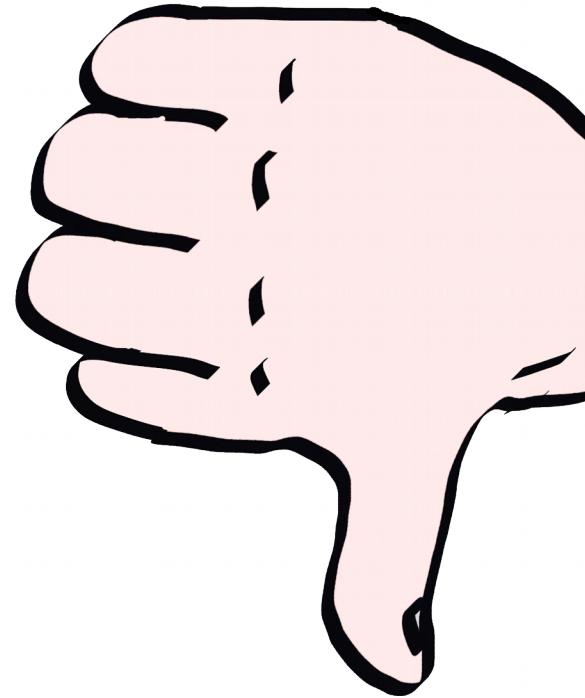
Query
Handler
Bus

@k_owsiany





cqrs



@k_owsiany



Legacy

@k_owsiany



Login

Libraries.io

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

1 - 30 of 1.57K packages

Sort: Relevance ▾

cqrs

cqrs for javascript

Latest release 3.2.1-pre - Updated Feb 18, 2018 - 231 stars

Cqrs

A lightweight enterprise framework to write CQRS, event-sourced and micro-service applications in...

License

- MIT 750 projects
- Apache-2.0 205 projects
- BSD-3-Clause 66 projects
- ISC 37 projects
- LGPL-2.1 36 projects
- Other 34 projects
- MPL-2.0 28 projects
- GPL-3.0 22 projects

@k_owsiany



walkenkit

,

js

,

nestjs | cgrs

@k_owsiany



prooph Honeybadger

php

|

ukcgvsp

@k_owsiany



sequent

|

verbly

|

verses

@k_owsiany



redshift
go
event horizon

@k_owsiany



axon
|

java

jester

@k_owsiany



recall

|

python

|

C#

@k_owsiany



? - Pascal - ?
|
?

@k_owsiany



Exceptional
Mediator
C#
|
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);
    }
}
```

@k_owsiany



```
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);  
        }  
    }  
}
```

@k_owsiany



```
public interface ICommand { }

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

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

@k_owsiany



```
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);  
    }  
}
```

@k_owsiany



```
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);  
        }  
}
```

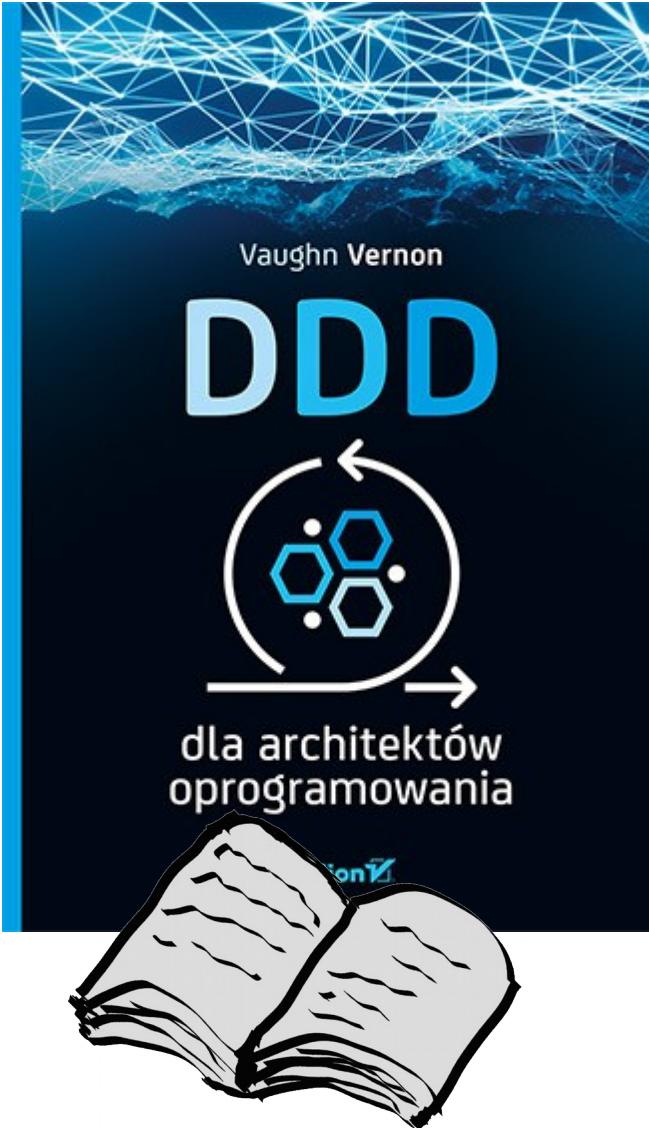


@k_owsiany

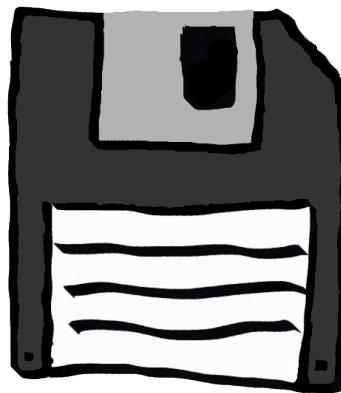
```
...  
  
_commandBus.SendCommand(new DeleteDataCommand("klucz"));  
  
...  
  
_queryBus.Process<GetDataQuery, string>(new GetDataQuery("filterek"));
```

@k_owsiany





cqrs
cqrs
ES



@k_owsiany

