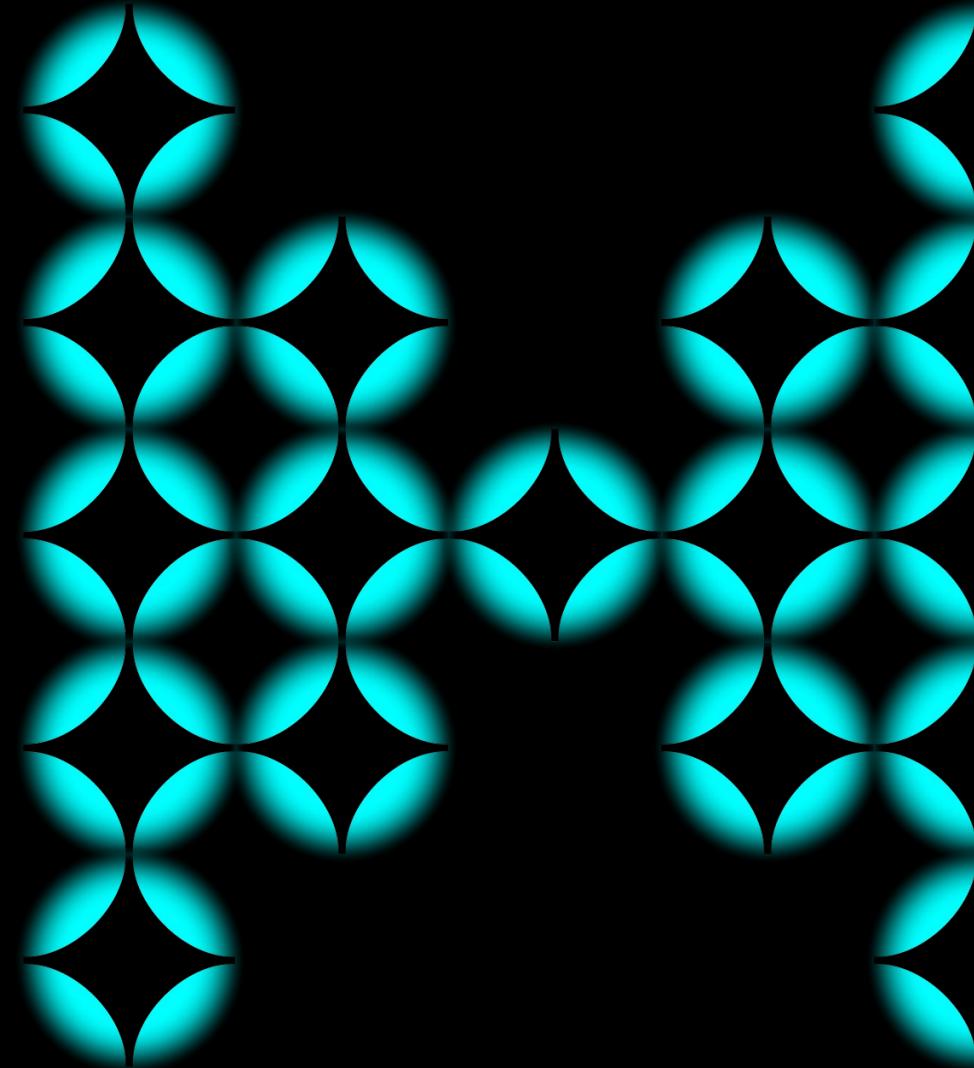


Porty i adaptery, CQRS, Event Sourcing, DDD ... Jak to zaimplementować w Springu?

Krzysztof Ślusarski
17.05.2022



Krótko o mnie

Krótko o mnie

Programowanie – od 1992

Krótko o mnie

Programowanie – od 1992

Zawodowo (JVM) – od 2006

Krótko o mnie

Programowanie – od 1992

Zawodowo (JVM) – od 2006

Szkolenia, Profiling, Diagnoza awarii

Zawodowo

Zawodowo

15 lat

Zawodowo

15 lat

Wewnętrzny software house

Zawodowo

15 lat

Wewnętrzny software house

Enterprise, pod wymagania
byznesu

Zawodowo

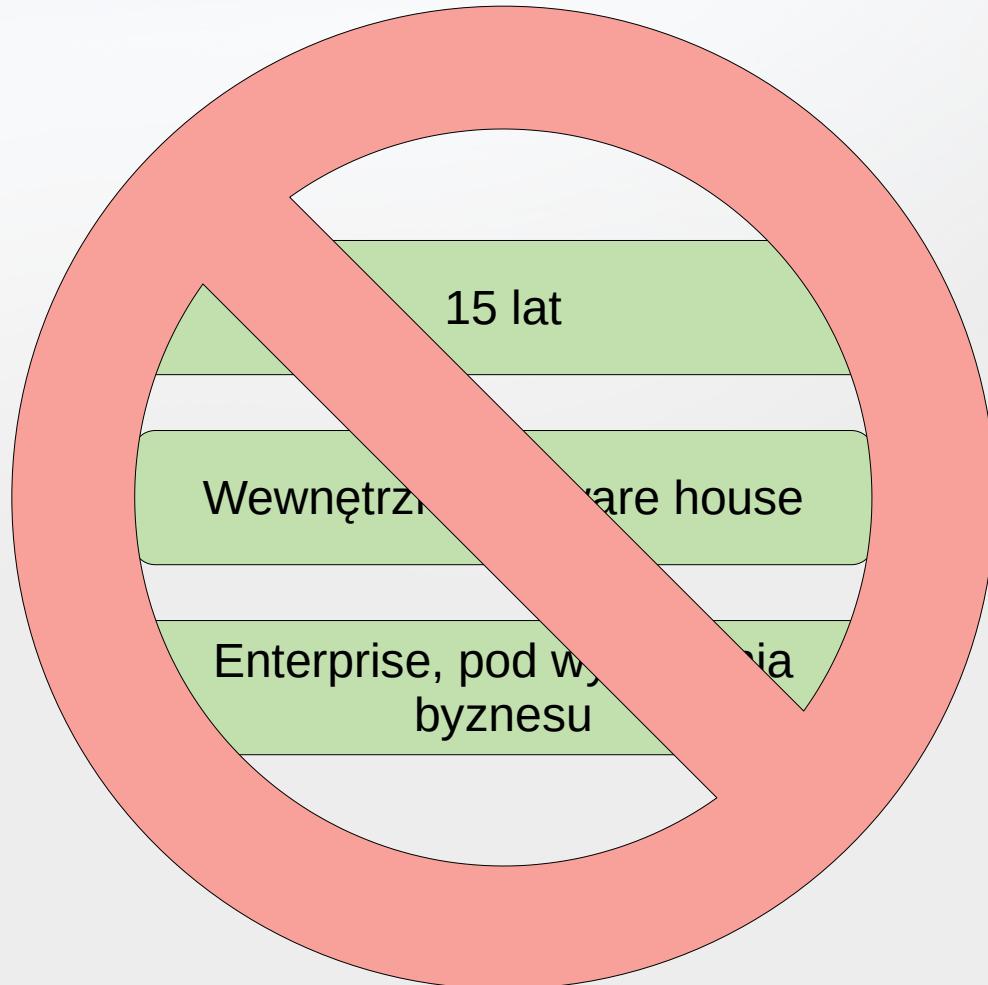
15 lat

Wewnętrzny software house

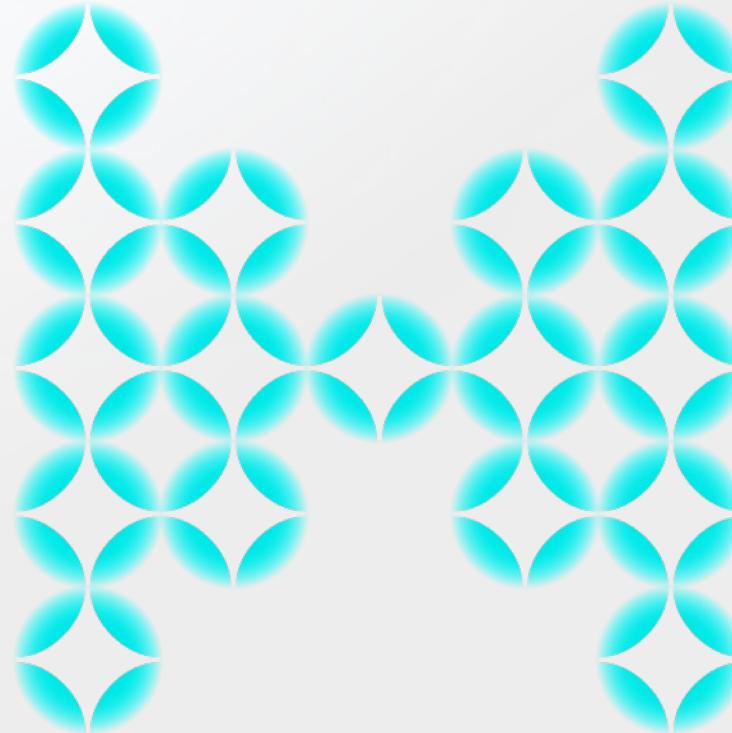
Enterprise, pod wymagania
byznesu



Zawodowo

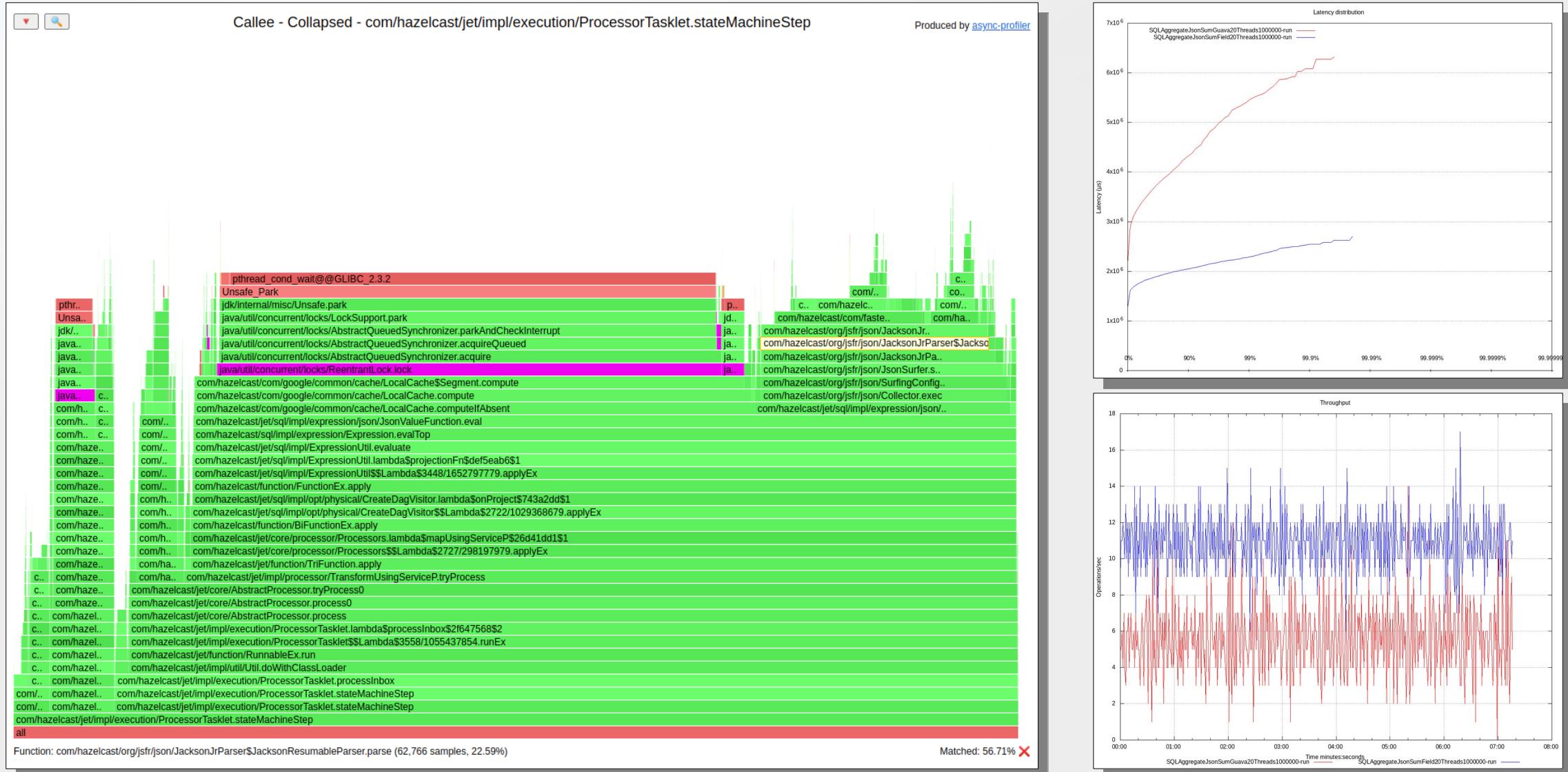


Zawodowo

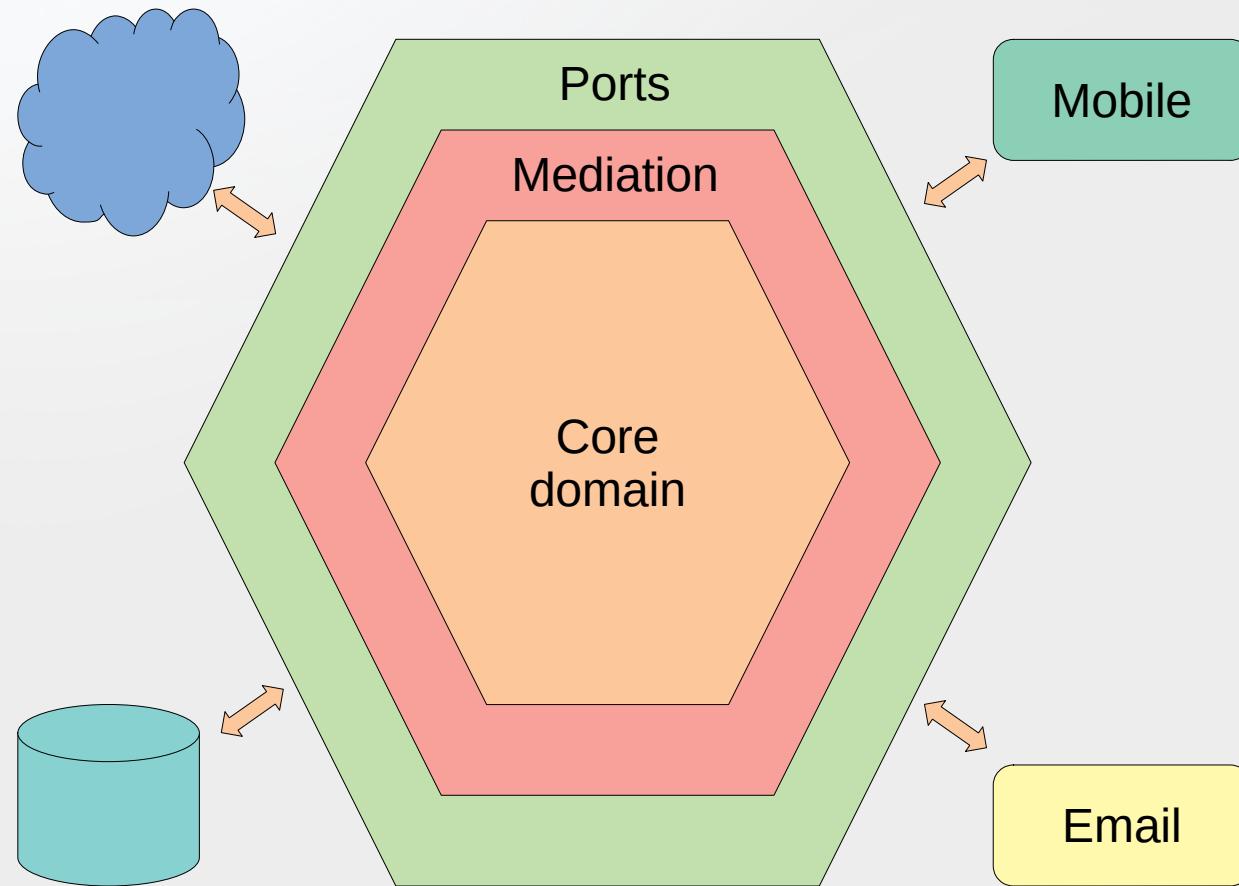


HAZELCAST

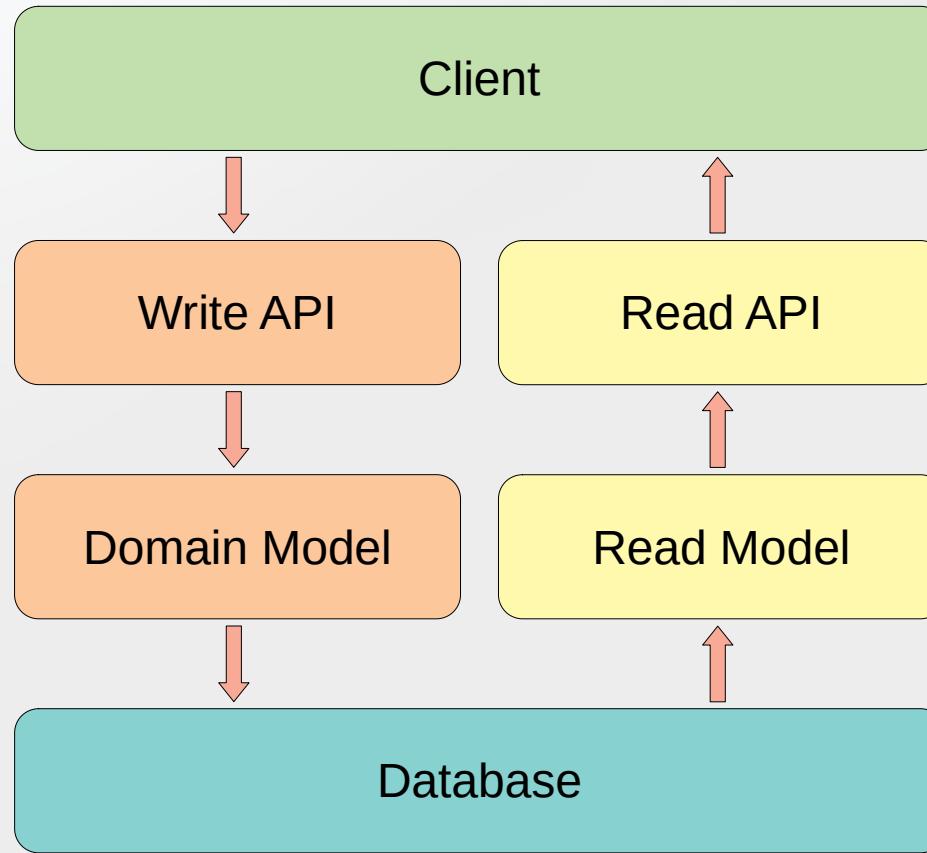
Zawodowo



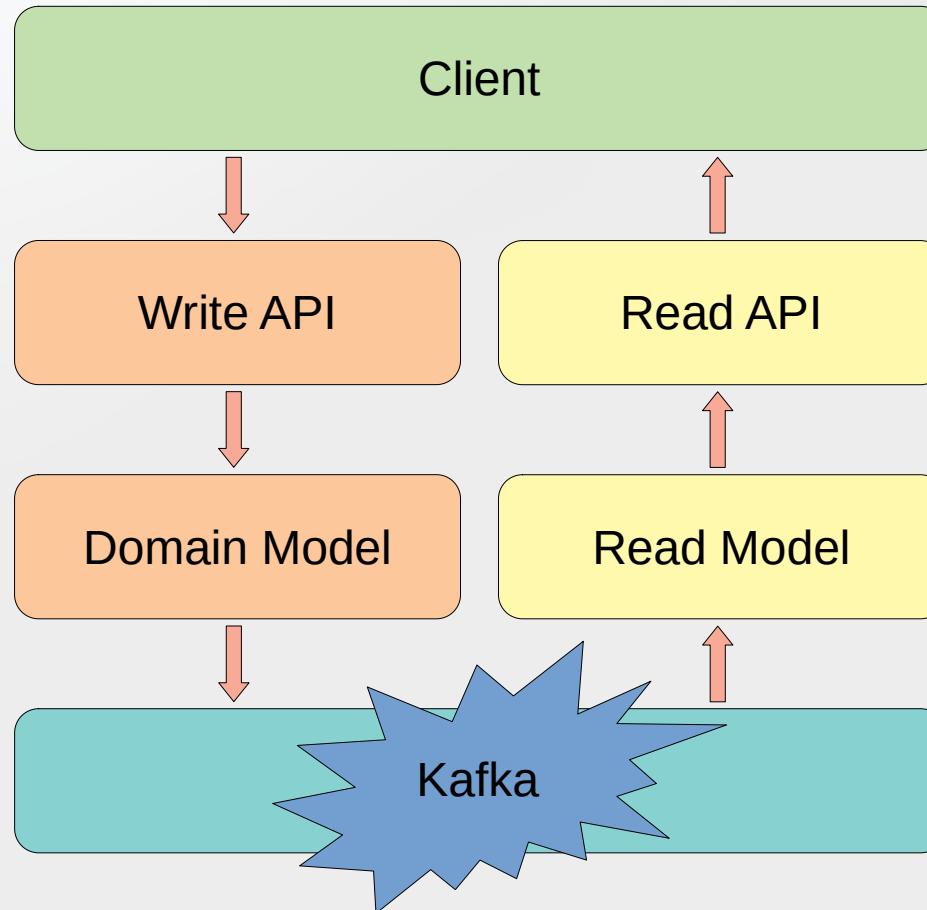
Cel



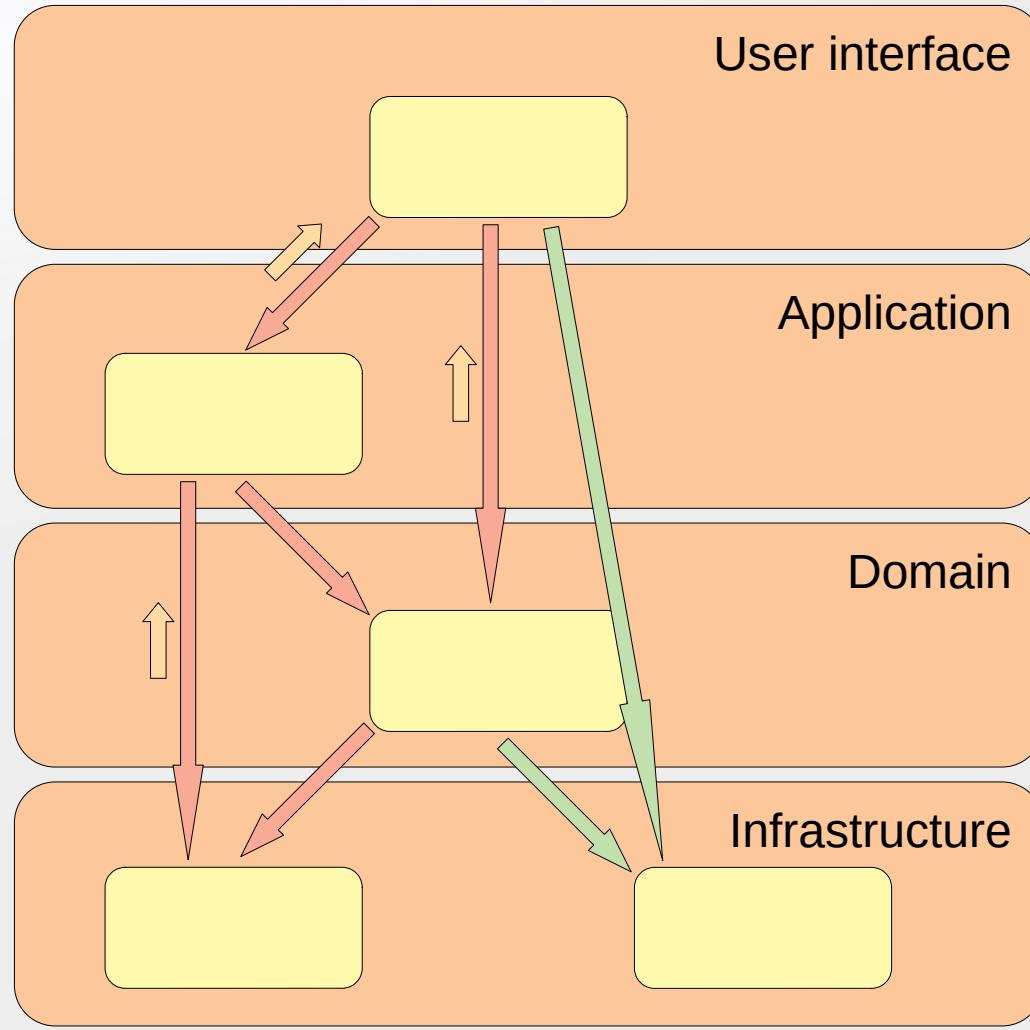
Cel



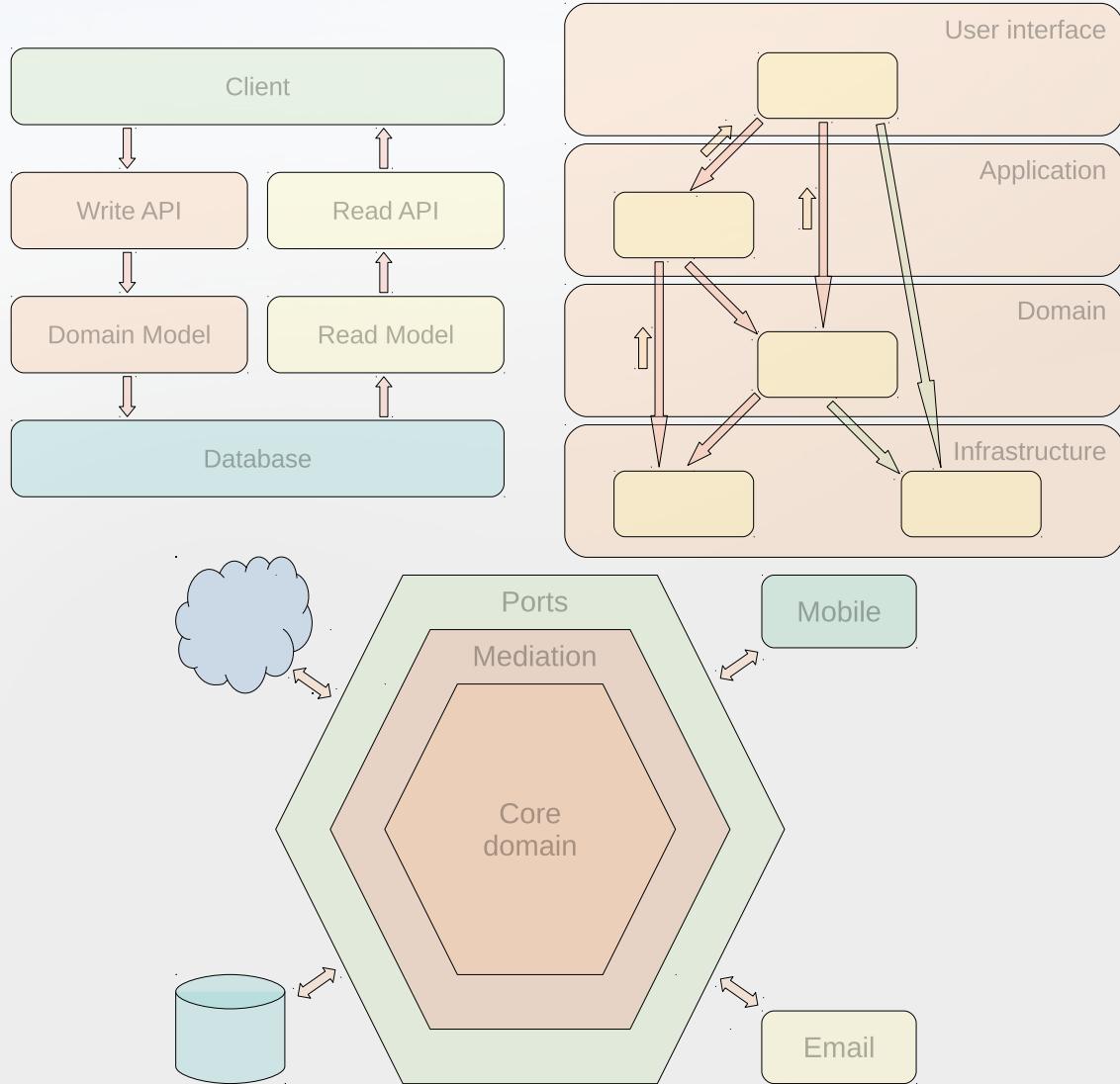
Cel



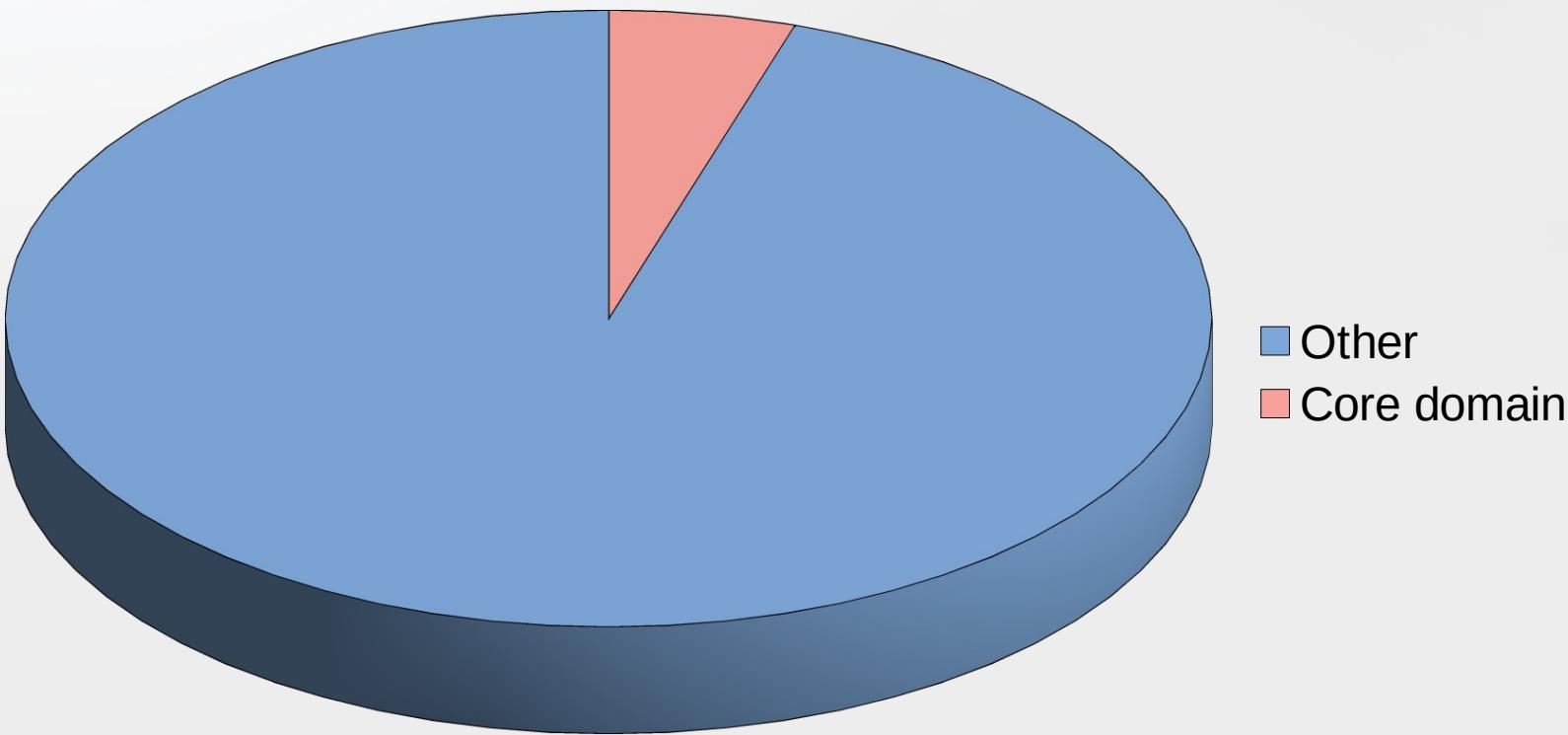
Cel



Cel



Cel



Domena

Domena



Domena

Wytwórz mojej wyobraźni

Domena

Wytwórz mojej wyobraźni

Wszelkie podobieństwa co do osób i zdarzeń
są przypadkowe

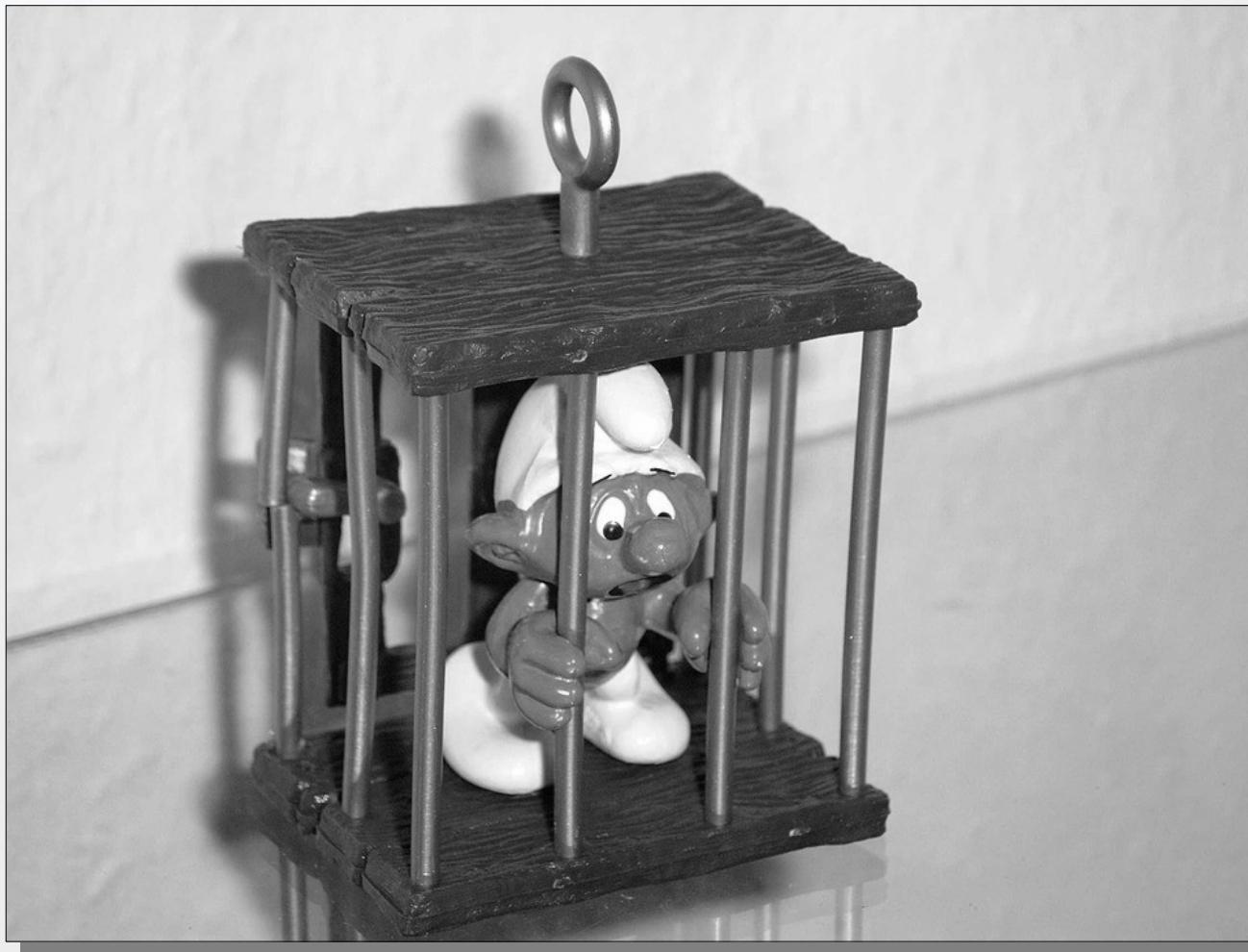
Domena

Wytwór mojej wyobraźni

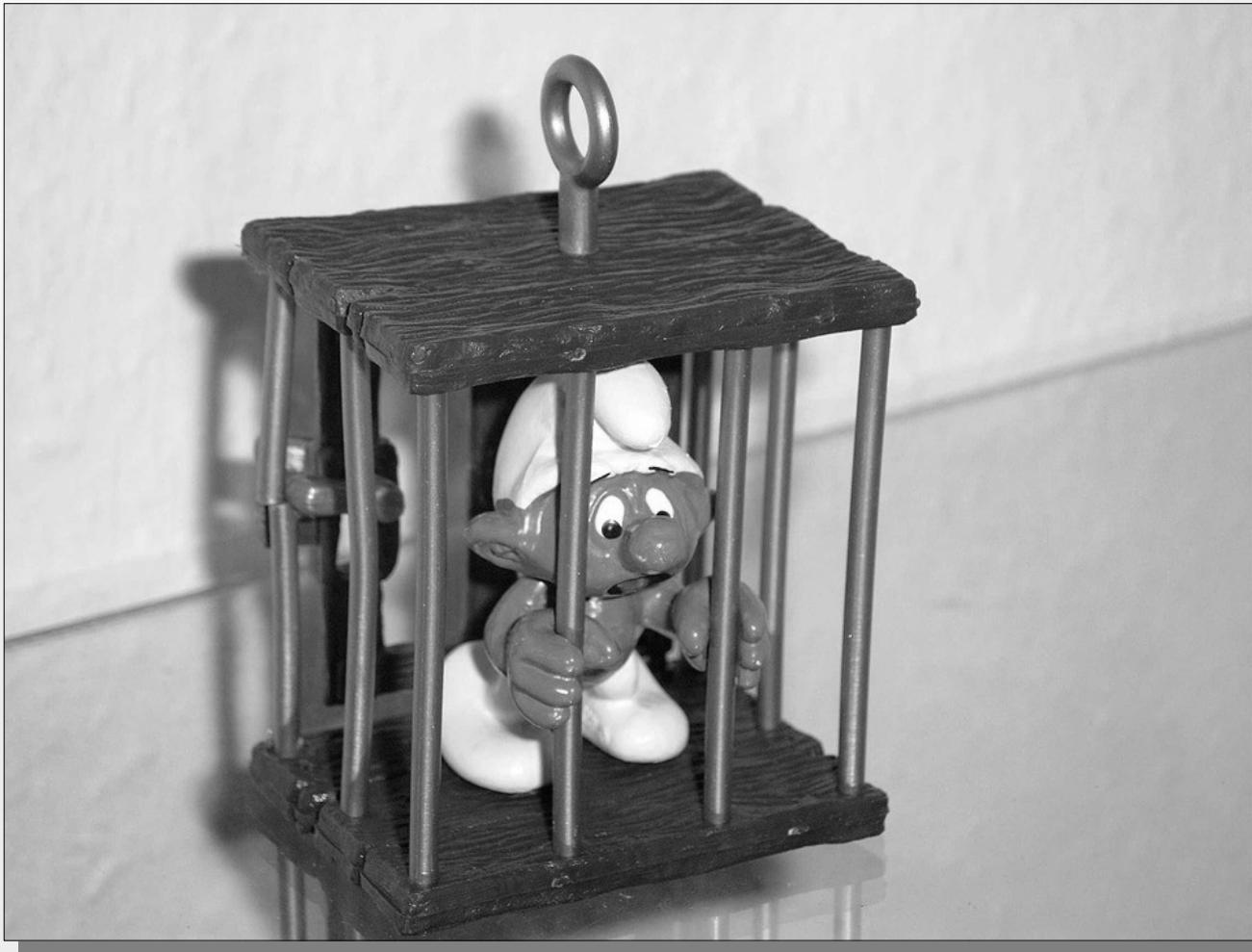
Wszelkie podobieństwa co do osób i zdarzeń
są przypadkowe

Brak politycznej poprawności

Domena



Domena



Domena



Domena



Domena



Janusz

Domena



Janusz

Lat 40

Domena



Janusz

Lat 40

Prezes firmy BodyLeasingowej od 15 lat

Domena



Janusz

Lat 40

Prezes firmy BodyLeasingowej od 15 lat

„Jedyne co trzeba robić dobrze w firmie BodyLeasingowej to wiedzieć kiedy zwolnić już zatrudnionego pracownika”

Domena

Projekt

Domena

Utworzono

Projekt

Domena

Utworzono

Projekt

Zmodyfiko-
wano

Domena

Utworzono

Projekt

Usunięto

Zmodyfiko-
wano

Domena

Utworzono

Zmodyfiko-
wano

Usunięto

Projekt

Widok:

- Nazwa
- Stawka

Domena

C

Utworzono

Projekt

Usunięto

Zmodyfiko-
wano

Widok:

- Nazwa
- Stawka

Domena

C

Utworzono

Projekt

Usunięto

Zmodyfiko-
wano

Widok:

- Nazwa
- Stawka

R

Domena

C

Utworzono

Projekt

Usunięto

Zmodyfiko-
wano

U

Widok:

- Nazwa
- Stawka

R

Domena

C

Utworzono

D

Usunięto

Projekt

Zmodyfiko-
wano

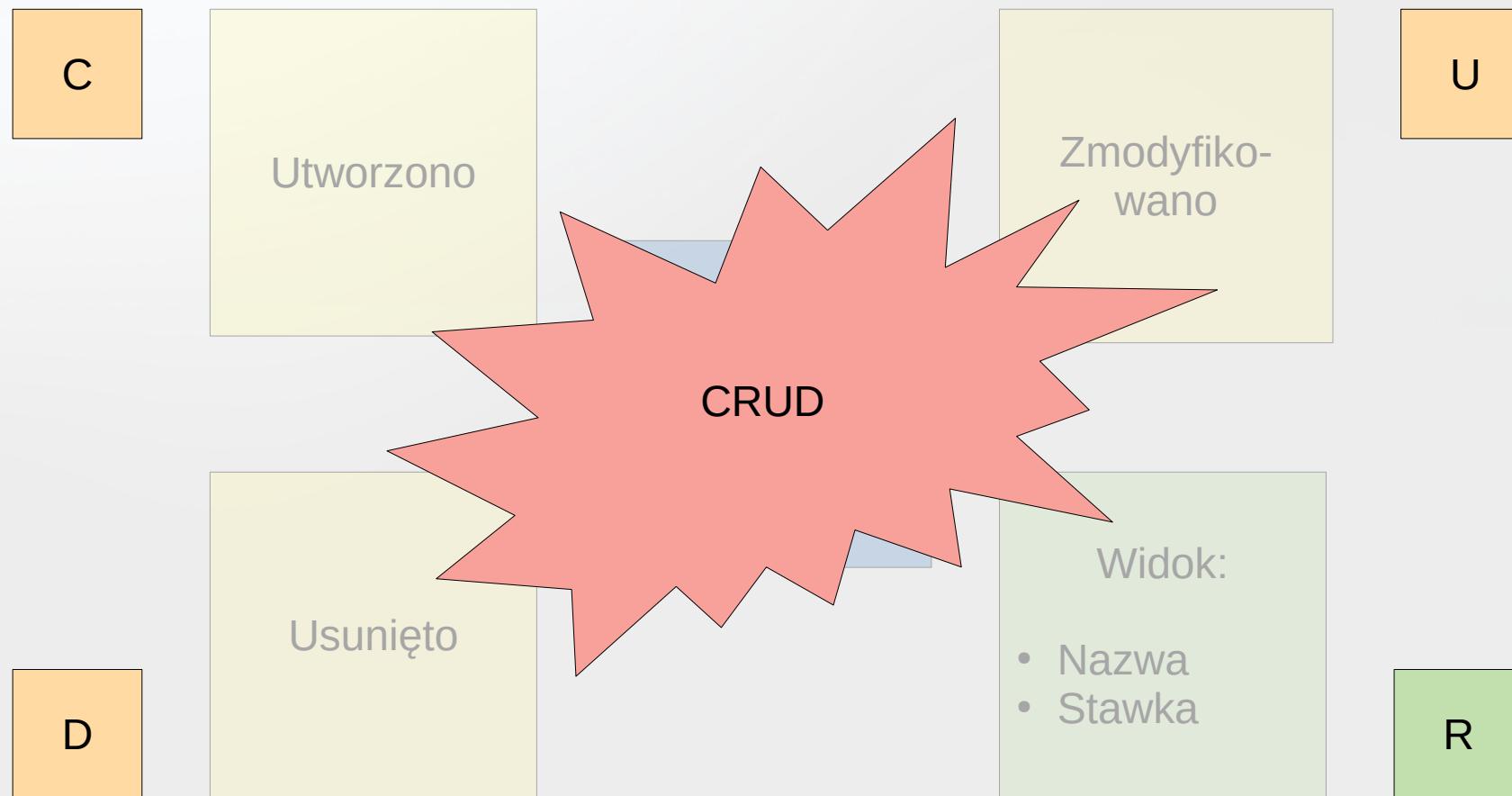
U

Widok:

- Nazwa
- Stawka

R

Domena



Domena

Parobek

Domena

Zatrudniono

Parobek

Domena

Zatrudniono

Parobek

Zwolniono

Domena

Zatrudniono

Zaraportowano
czas pracy

Parobek

Zwolniono

Domena

Zatrudniono

Zwolniono

Parobek

Zaraportowano
czas pracy

Wystawiono
fakturę

Domena

Zatrudniono

Zwolniono

Parobek

Zaraportowano
czas pracy

Wystawiono
fakturę

Widok:
• Ile firma
zarobiła
(kwota)

Domena

Reguły biznesowe:

- Nie można zaraportować czasu pracy jeżeli nie wystawiono faktury za poprzedni raport
- Faktura musi być zgodna ze stawką godzinową i raportem czasu pracy.
- Parobek przed zatrudnieniem musi być zweryfikowany pod kątem niekaralności (WS zewnętrzny)
- ...

Zatrudniono

Zwolniono

Parobek

Zaraportowano
czas pracy

Wystawiono
fakturę

Widok:

- Ile firma zarobiła (kwota)

Struktura na dziś



Struktura na dziś



Struktura na dziś



Struktura na dziś



Struktura na dziś



Jak tego nie robić?



Co w tym kodzie jest najgorsze?

Co w tym kodzie jest najgorsze?



Co w tym kodzie jest najgorsze?



public

Co w tym kodzie jest najgorsze?



public



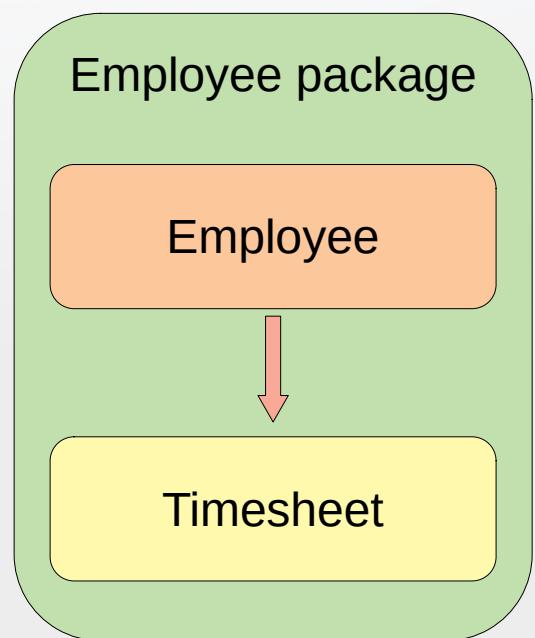
Anemiczny model domeny



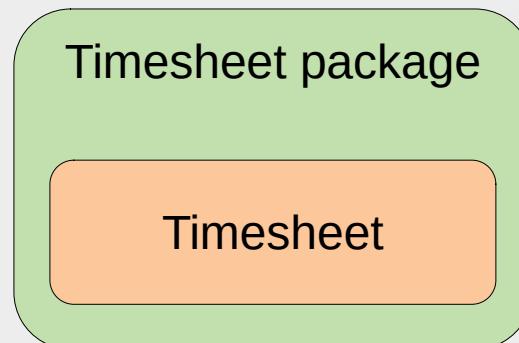
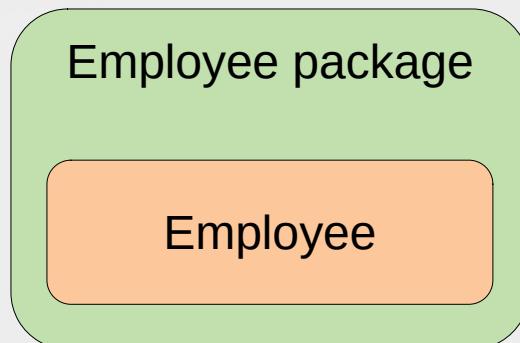
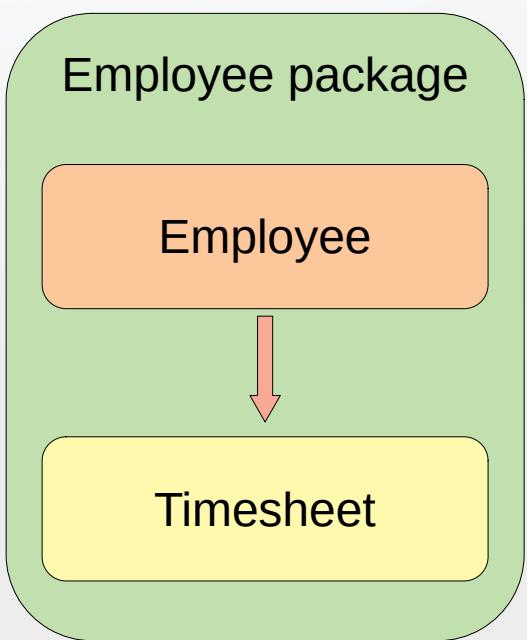
Jak to poprawić?



Archeo vs ORM



Archeo vs ORM



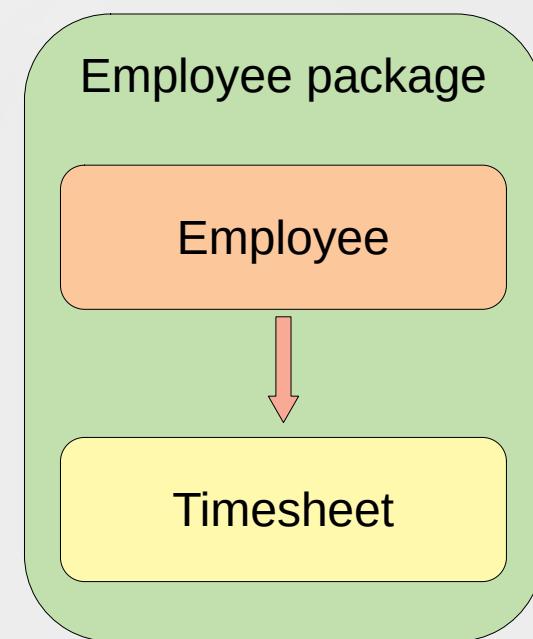
Archeo vs ORM

```
@Entity
public class Employee {
    @JoinColumn("fk_employee")
    @OneToMany
    private Set<Timesheet> timesheets;

    // ...
}
```

Archeo vs ORM

```
@Entity  
public class Employee {  
    @JoinColumn("fk_employee")  
    @OneToMany  
    private Set<Timesheet> timesheets;  
  
    // ...  
}
```



Archeo vs ORM

```
@Entity
public class Timesheet {
    @AttributeOverrides({
        @AttributeOverride(
            name = "value",
            column = @Column(name = "fk_employee")
        )
    })
    private EmployeeId employeeId;

    // ...
}
```

Archeo vs ORM

```
@Entity  
public class Timesheet {  
    @AttributeOverrides({  
        @AttributeOverride(  
            name = "value",  
            column = @Column(name = "fk_employee")  
    })  
    private EmployeeId employeeId;  
  
    // ...  
}
```

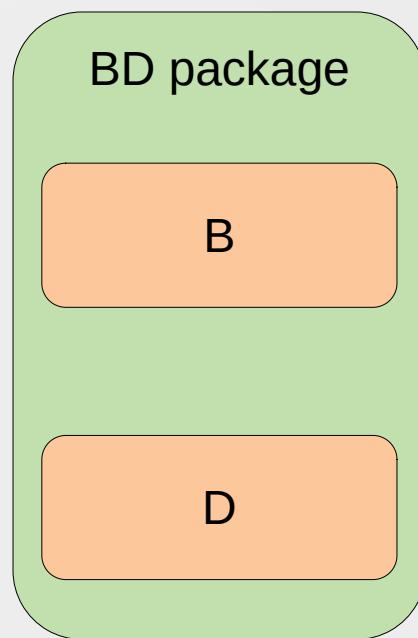
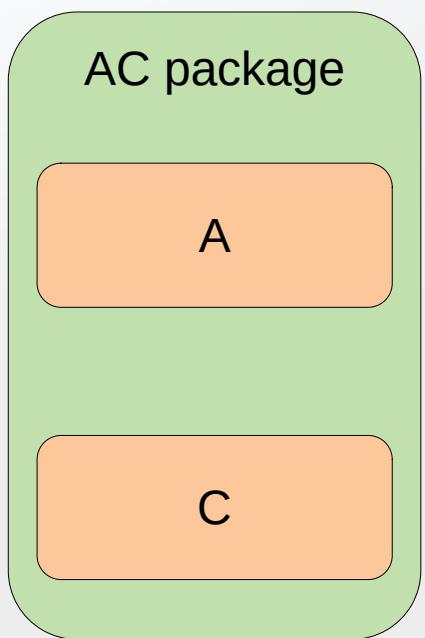
Employee package

Employee

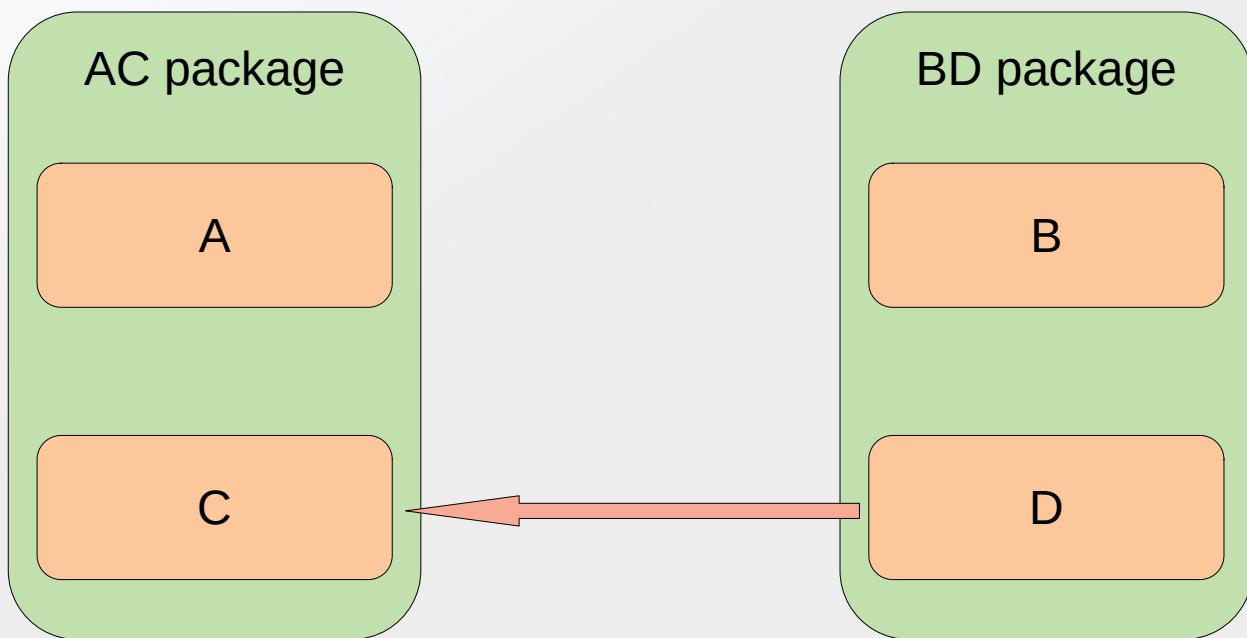
Timesheet package

Timesheet

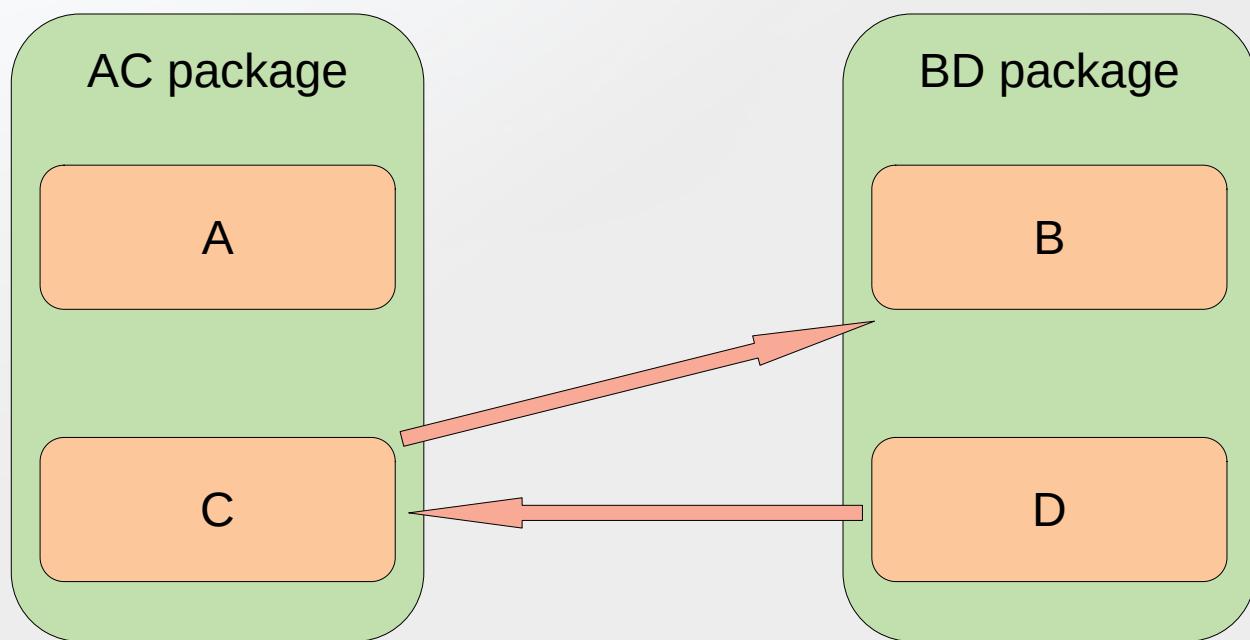
Cykl



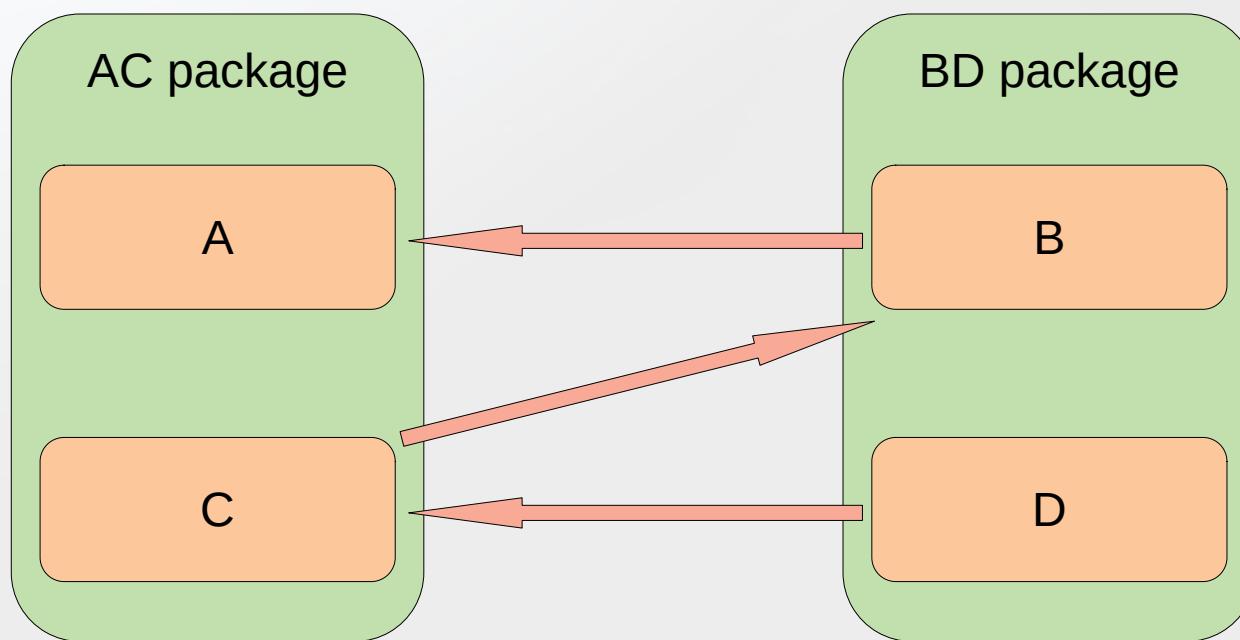
Cykl



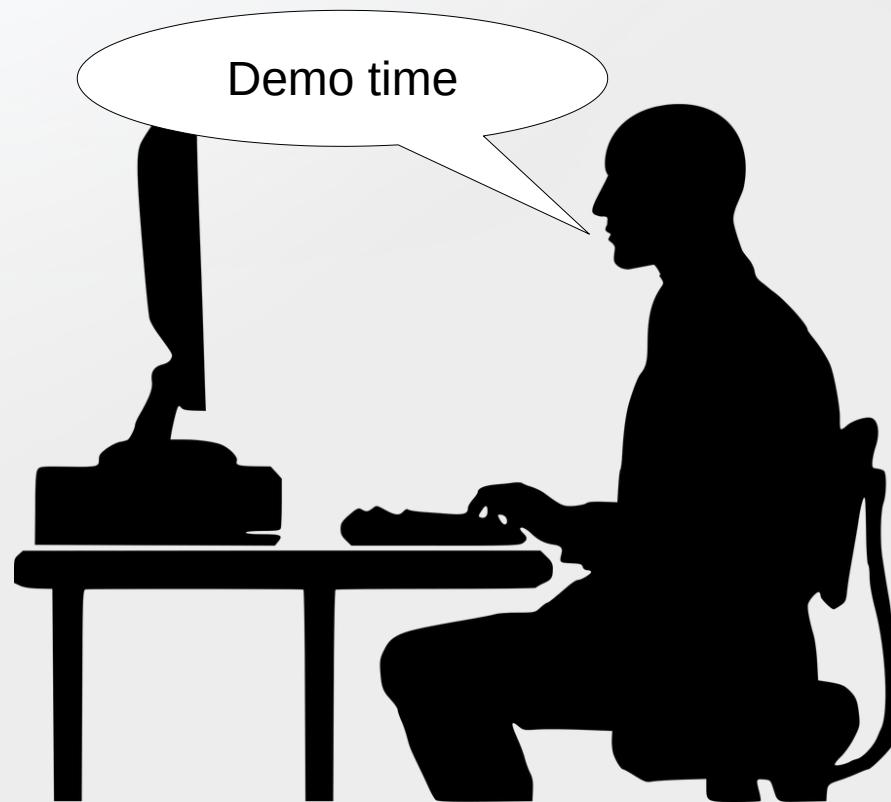
Cykl



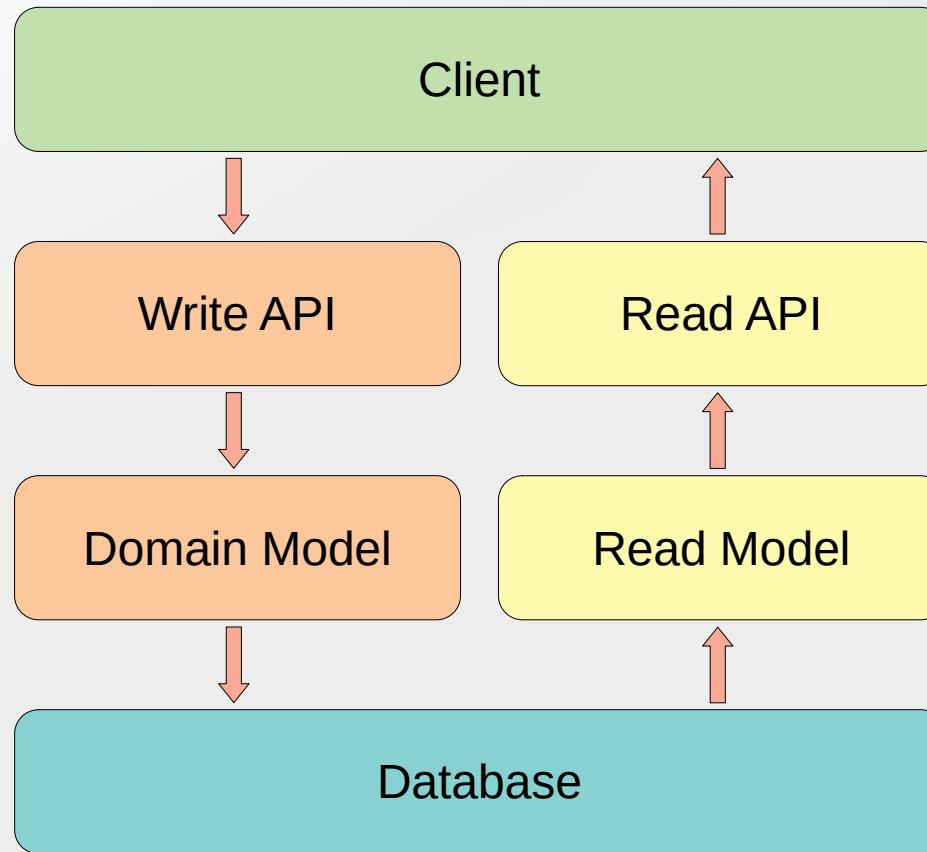
Cykl



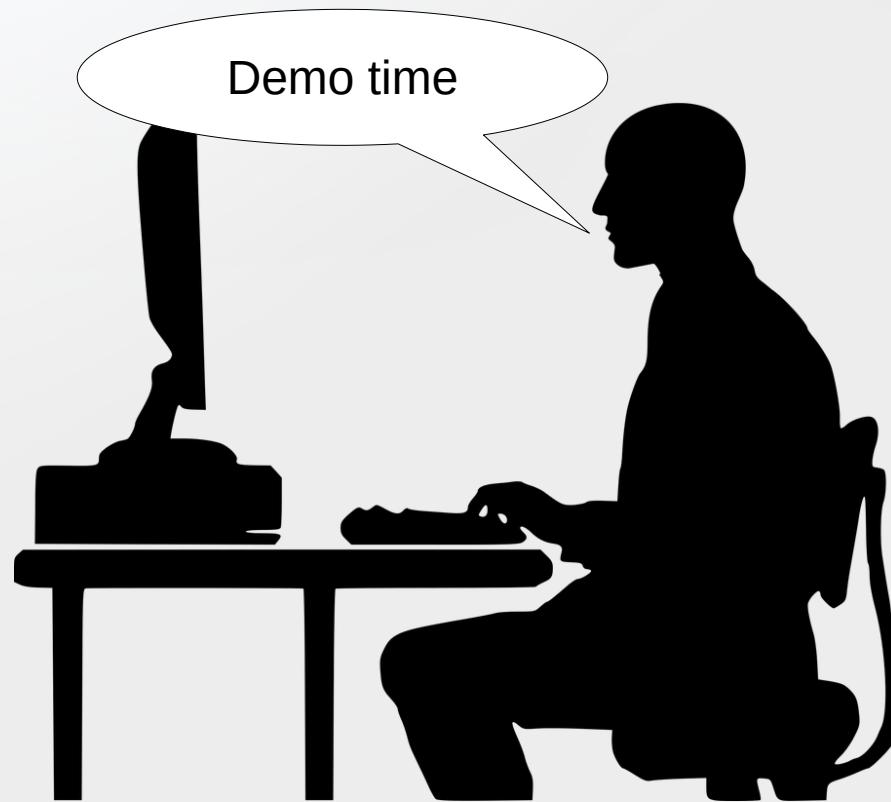
Cykl



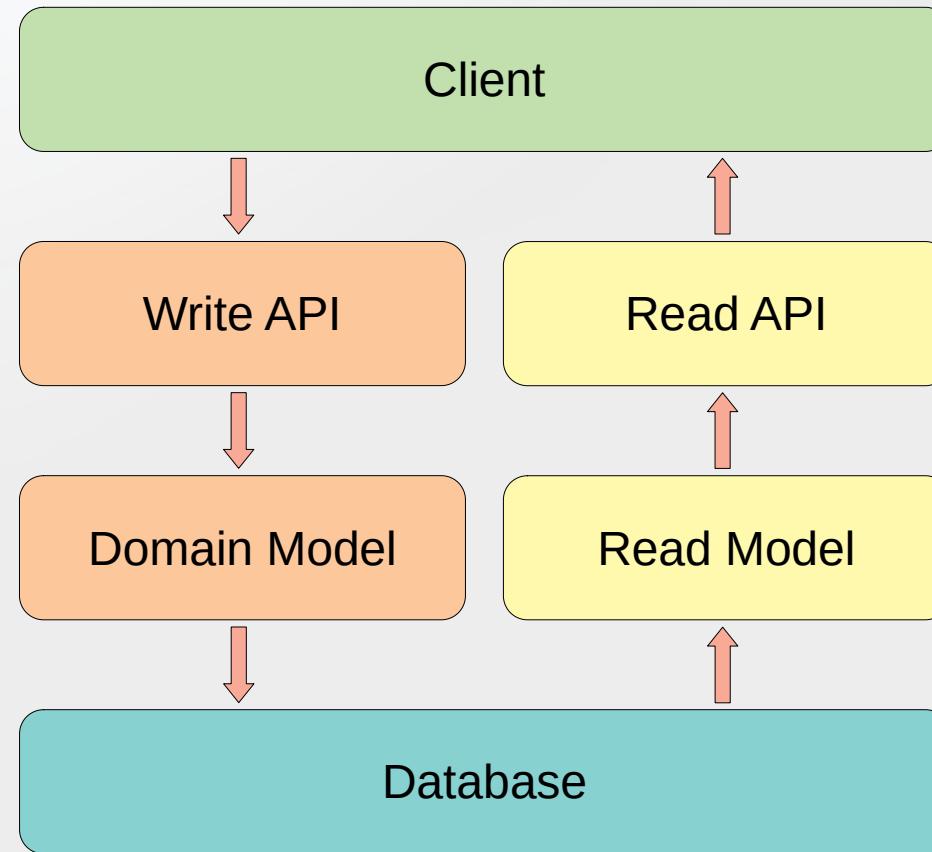
Czas na:



CQRS



Jest?



Jeszcze inaczej?

```
@Query(new Employee(...))
```

Jeszcze inaczej?

```
@Query(new Employee(...))
```

```
@Query(new EmployeeDto(...))
```

Jeszcze inaczej?

```
@Query(new Employee(...))
```

```
@Query(new EmployeeDto(...))
```

```
@Subselect / @Synchronize
```

Jeszcze inaczej?

`@Query(new Employee(...))`

`@Query(new EmployeeDto(...))`

`@Subselect / @Synchronize`

`JDBC`

Co w tym kodzie jest jeszcze złe?

Co w tym kodzie jest jeszcze złe?

O O P? 

Co w tym kodzie jest jeszcze złe?

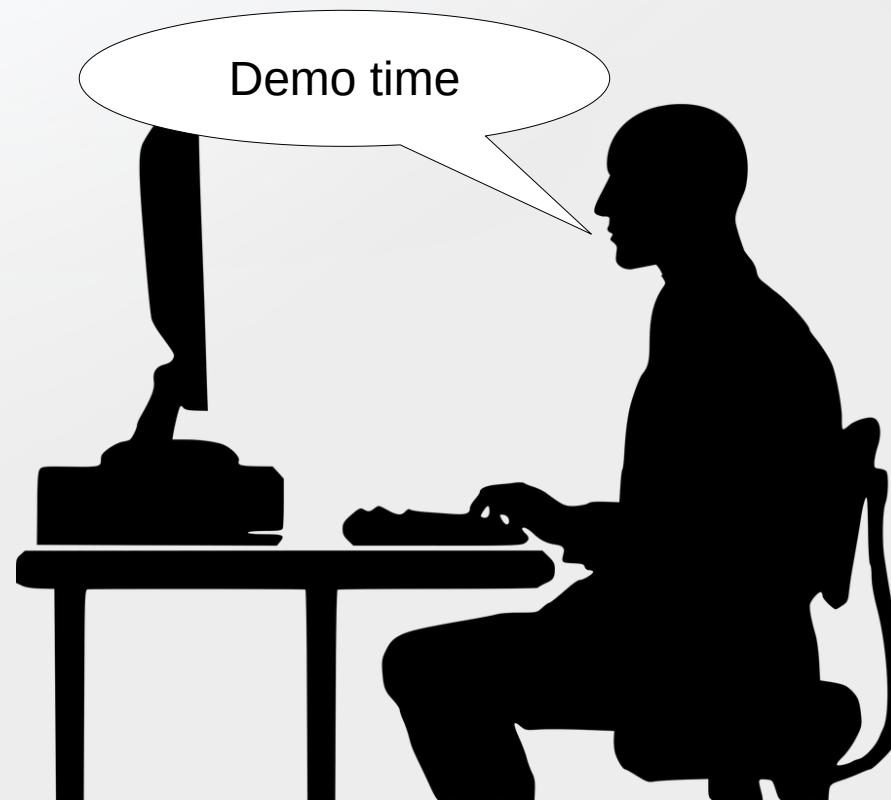
```
class Employee {  
    @{...}  
    private EmployeeId id;  
  
    @Version  
    private Long version;  
  
    @{...}  
    private FirstName firstName;  
  
    @{...}  
    private LastName lastName;  
  
    @{...}  
    private Money hourlyEarnings;  
  
    @{...}  
    private WorkHours lastNotSettledTimesheetWorkTime;  
}
```

```
@RequiredArgsConstructor  
class EmployeeServiceImpl implements EmployeeService {  
    private final EmployeeRepository employeeRepository;  
    private final EmployeeSubService employeeSubService;  
  
    @Override  
    public EmployeeId createNew(FirstName firstName, LastName lastName, Money hourlyEarnings) {...}  
  
    @Override  
    public void addNewTimesheet(EmployeeId employeeId, WorkHours hours) {...}  
  
    @Override  
    public void settleNewInvoice(EmployeeId employeeId, Money payment) {...}  
}
```

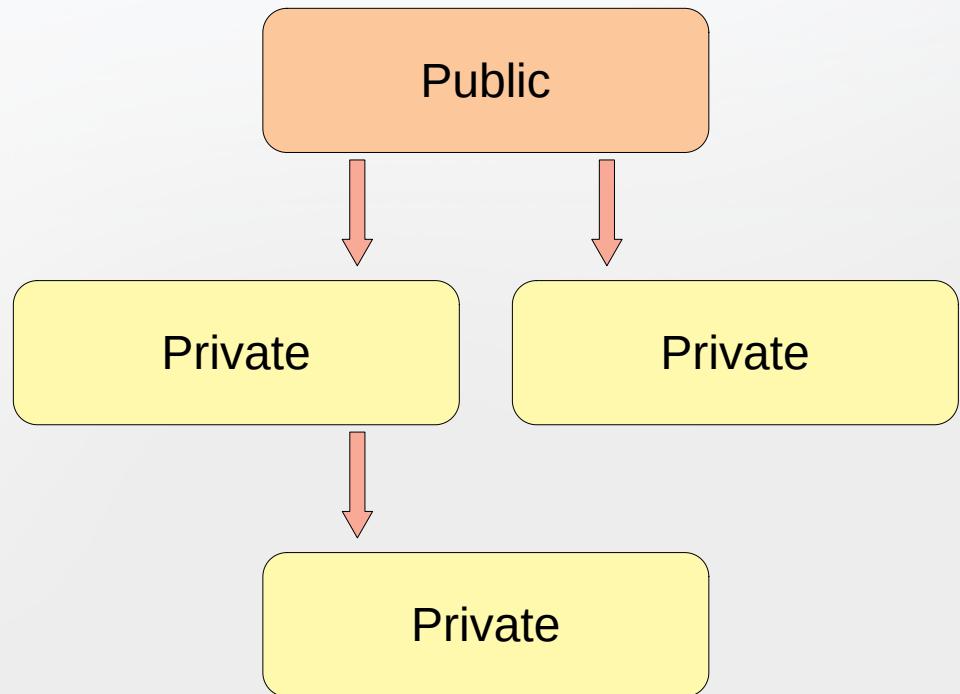
Jaka jest zaleta tego kodu?



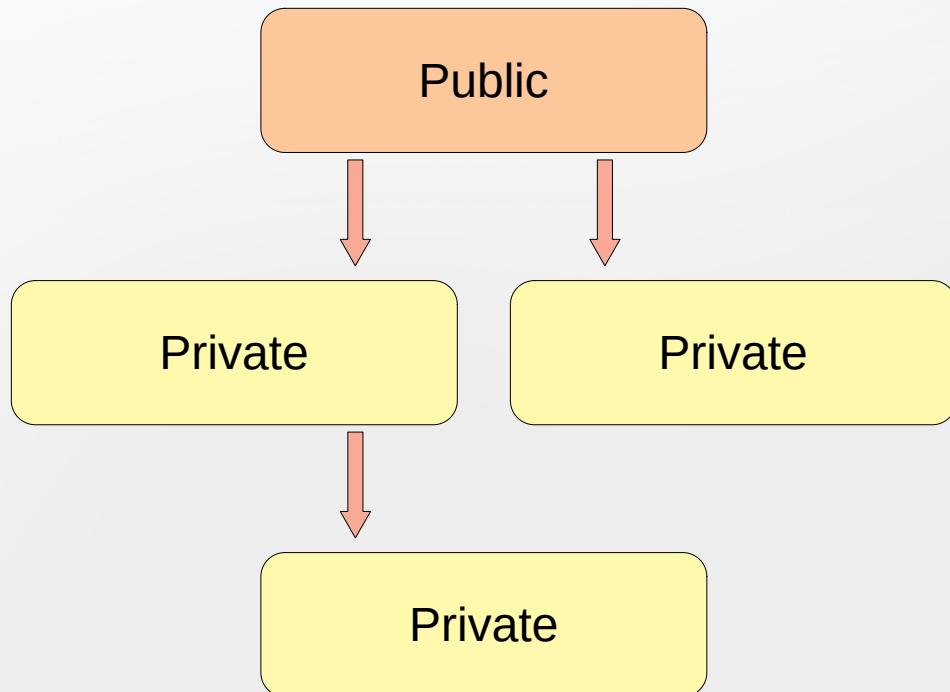
Jedziemy dalej



OOP

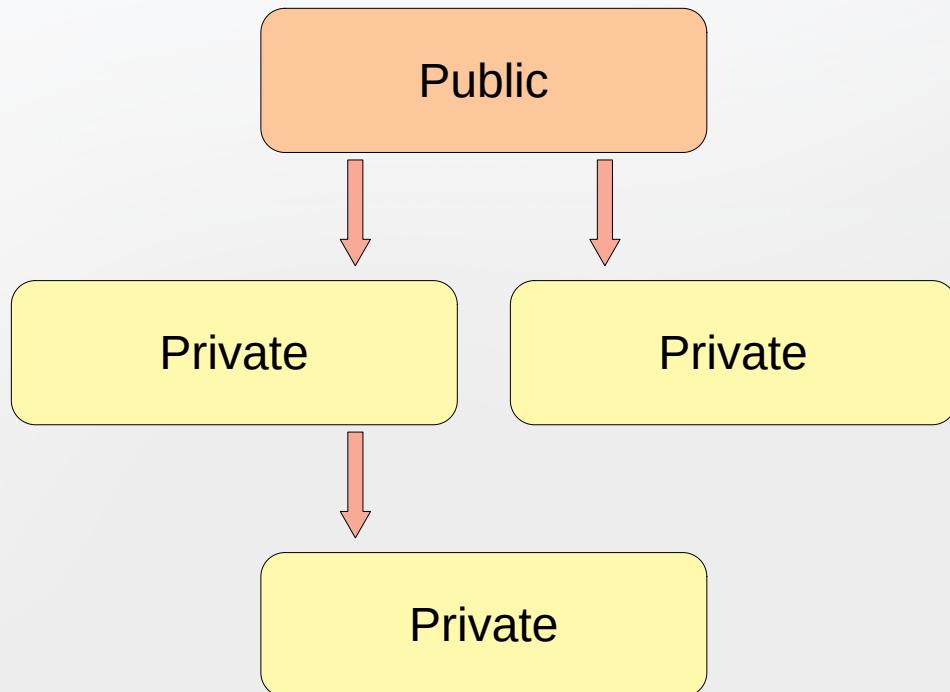


OOP

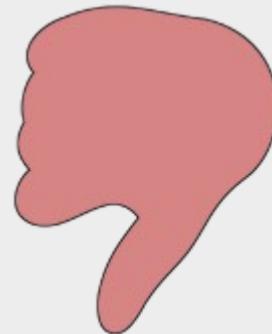


```
public class Employee {  
    //...  
  
    public void addHoursToTimesheet(TimesheetId timesheetId, WorkHours workHours) {  
        if (!timesheets.containsKey(timesheetId)) {  
            throw new IllegalArgumentException("Invalid timesheet ID");  
        }  
  
        Timesheet timesheet = timesheets.get(timesheetId);  
        timesheet.setReportedHours(timesheet.getReportedHours().plus(workHours));  
    }  
  
    //...  
}
```

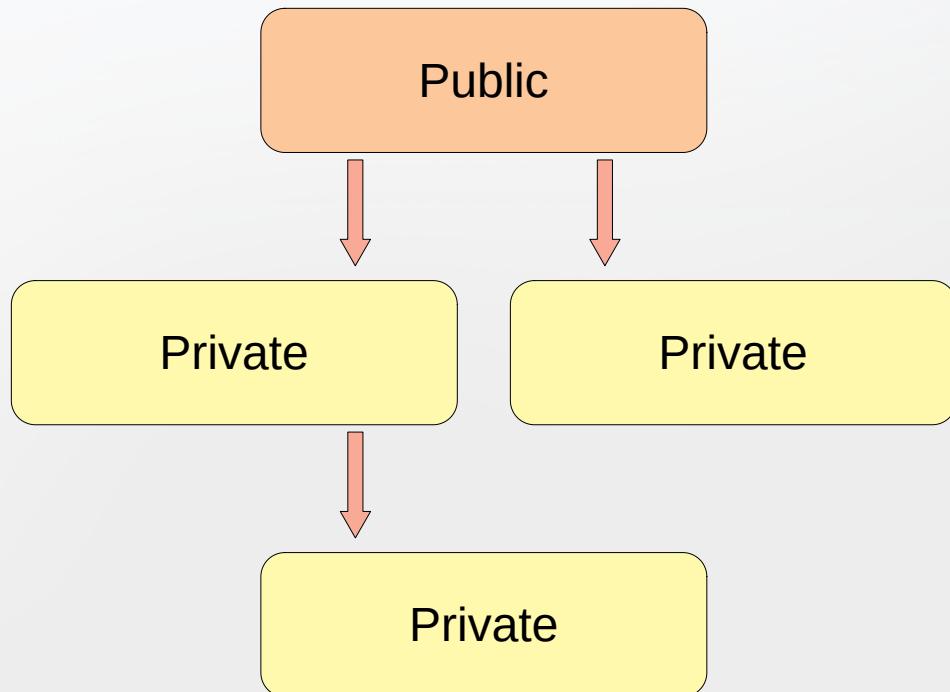
OOP



```
public class Employee {  
    //...  
  
    public void addHoursToTimesheet(TimesheetId timesheetId, WorkHours workHours) {  
        if (!timesheets.containsKey(timesheetId)) {  
            throw new IllegalArgumentException("Invalid timesheet ID");  
        }  
  
        Timesheet timesheet = timesheets.get(timesheetId);  
        timesheet.setReportedHours(timesheet.getReportedHours().plus(workHours));  
    }  
  
    //...  
}
```

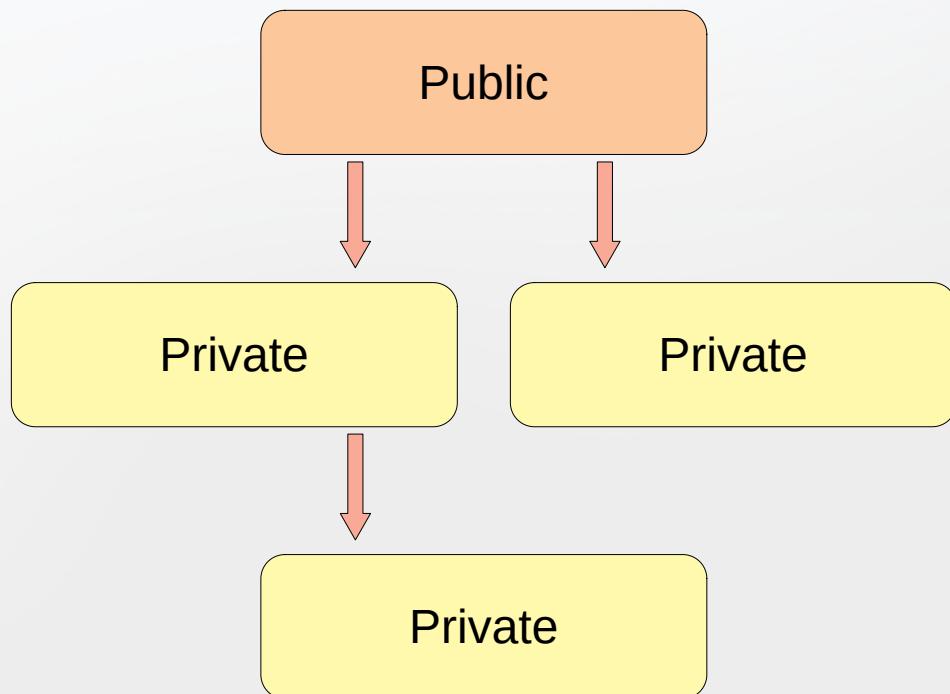


OOP



```
public class Employee {  
    //...  
  
    public void addHoursToTimesheet(TimesheetId timesheetId, WorkHours workHours) {  
        if (!timesheets.containsKey(timesheetId)) {  
            throw new IllegalArgumentException("Invalid timesheet ID");  
        }  
  
        Timesheet timesheet = timesheets.get(timesheetId);  
        timesheet.addHours(workHours);  
    }  
  
    //...  
}
```

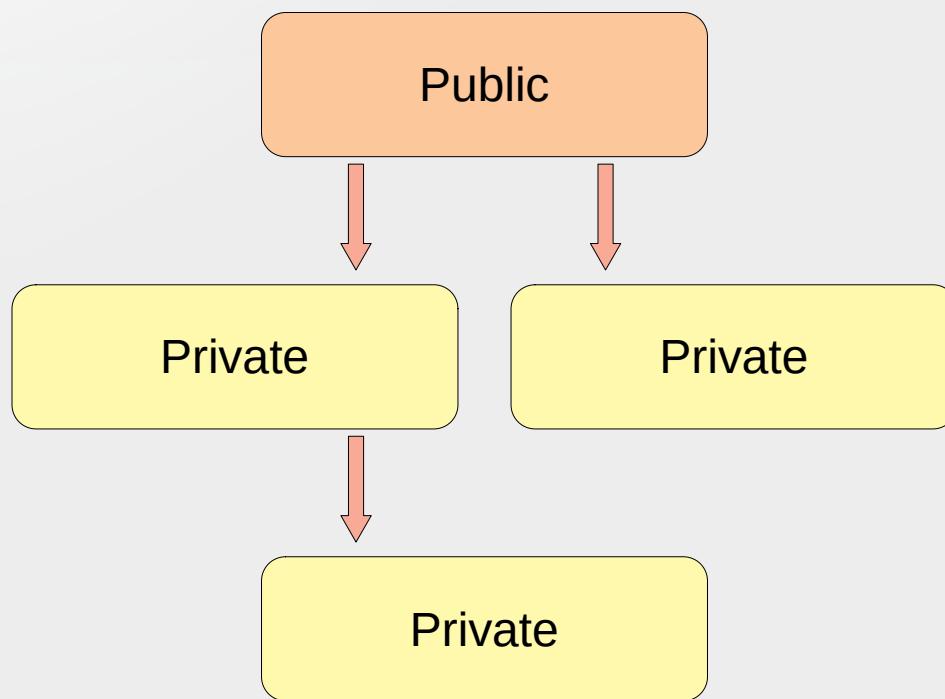
OOP



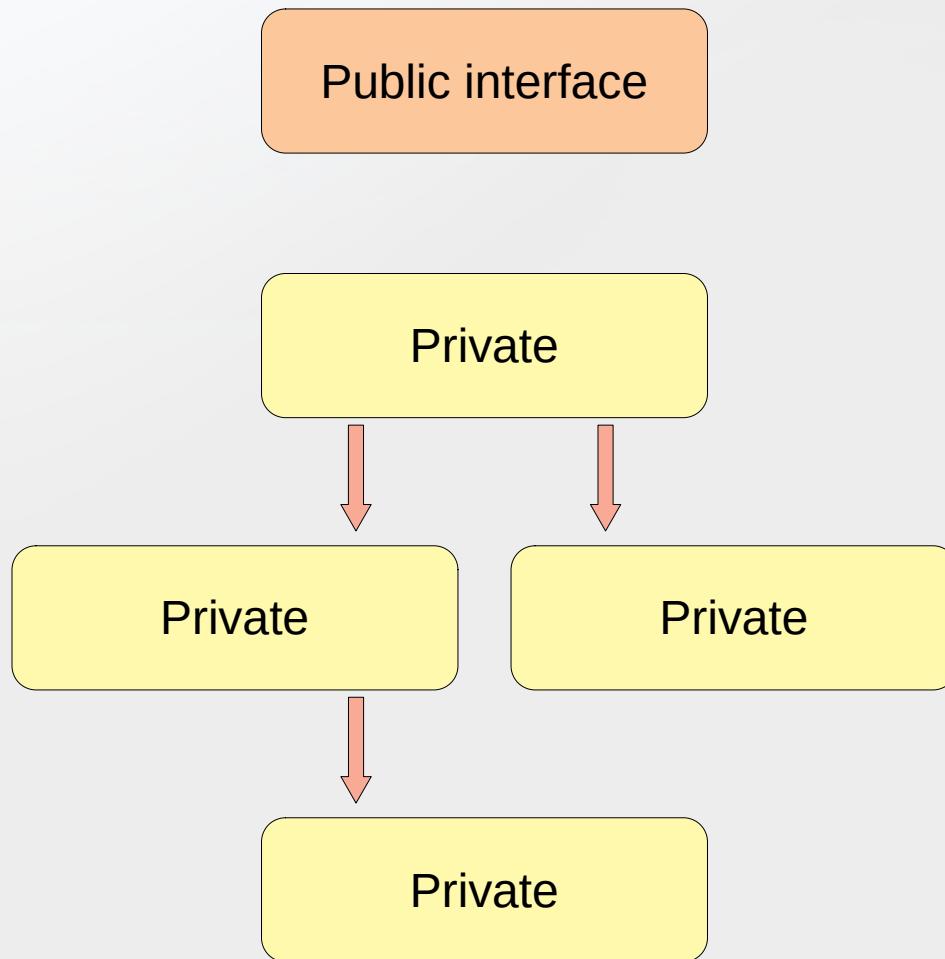
```
public class Employee {  
    //...  
  
    public void addHoursToTimesheet(TimesheetId timesheetId, WorkHours workHours) {  
        if (!timesheets.containsKey(timesheetId)) {  
            throw new IllegalArgumentException("Invalid timesheet ID");  
        }  
  
        Timesheet timesheet = timesheets.get(timesheetId);  
        timesheet.addHours(workHours);  
    }  
  
    //...  
}
```



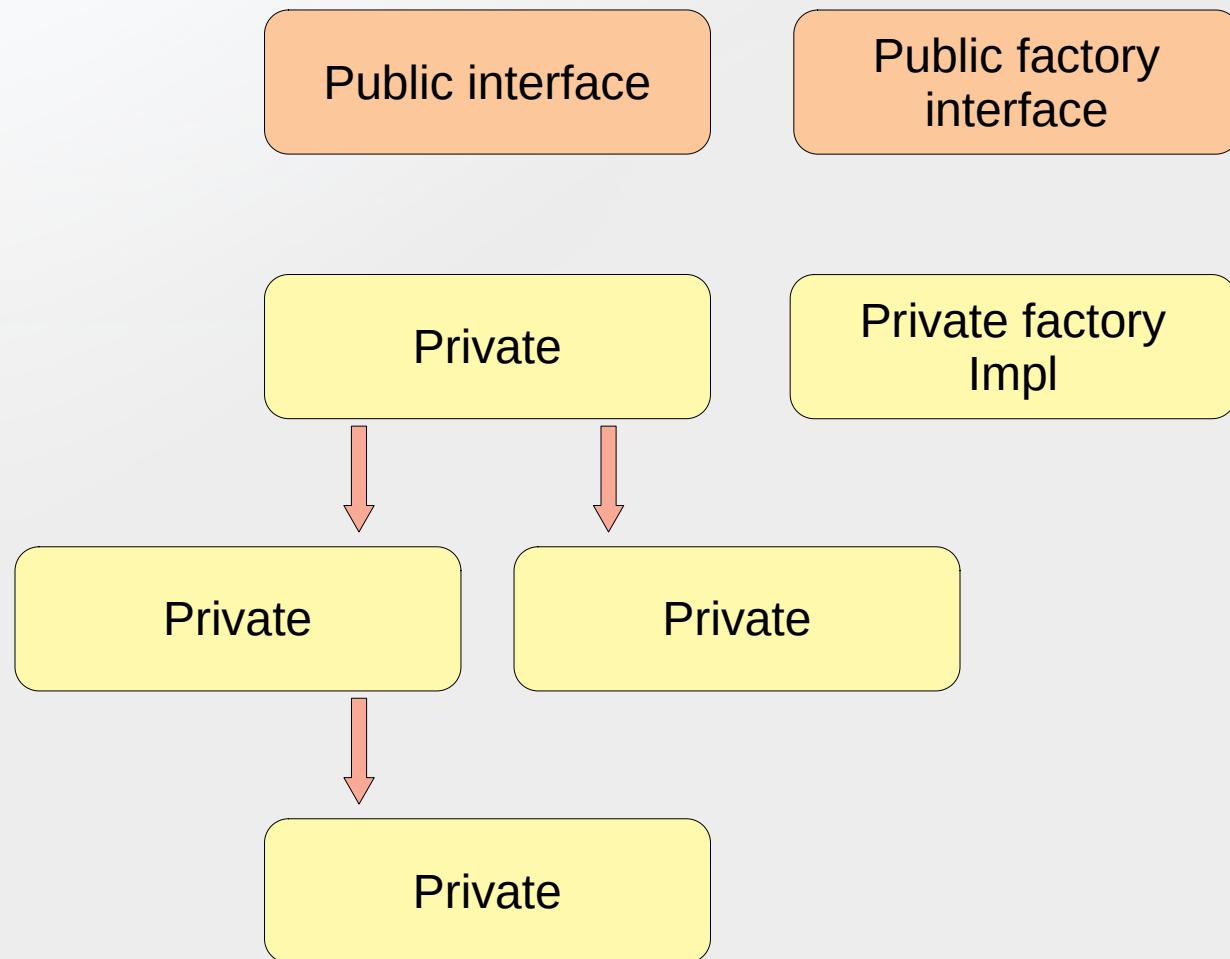
OOP



OOP



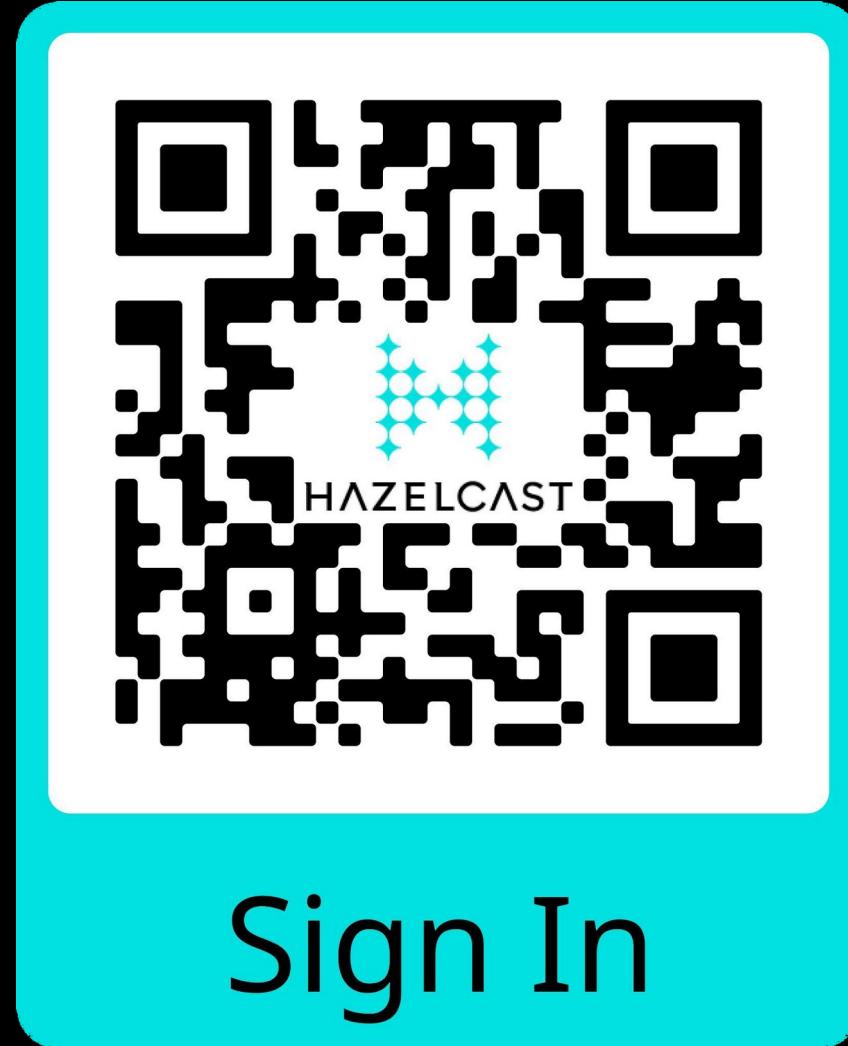
OOP



Q/A?

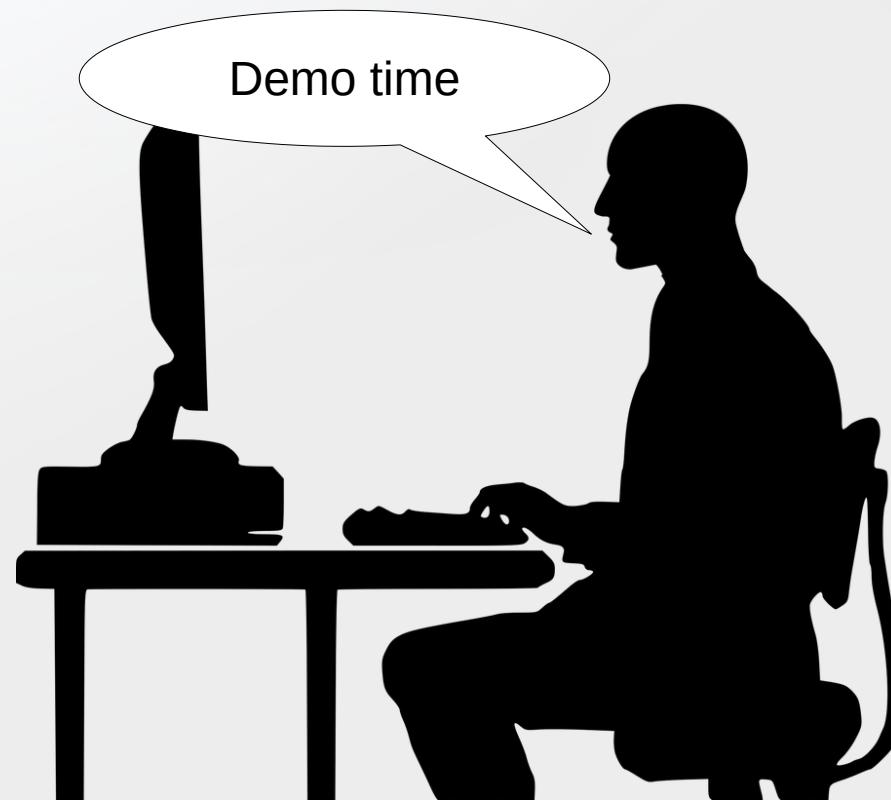


Scan the QR Code for your chance to win a €100 Amazon Voucher

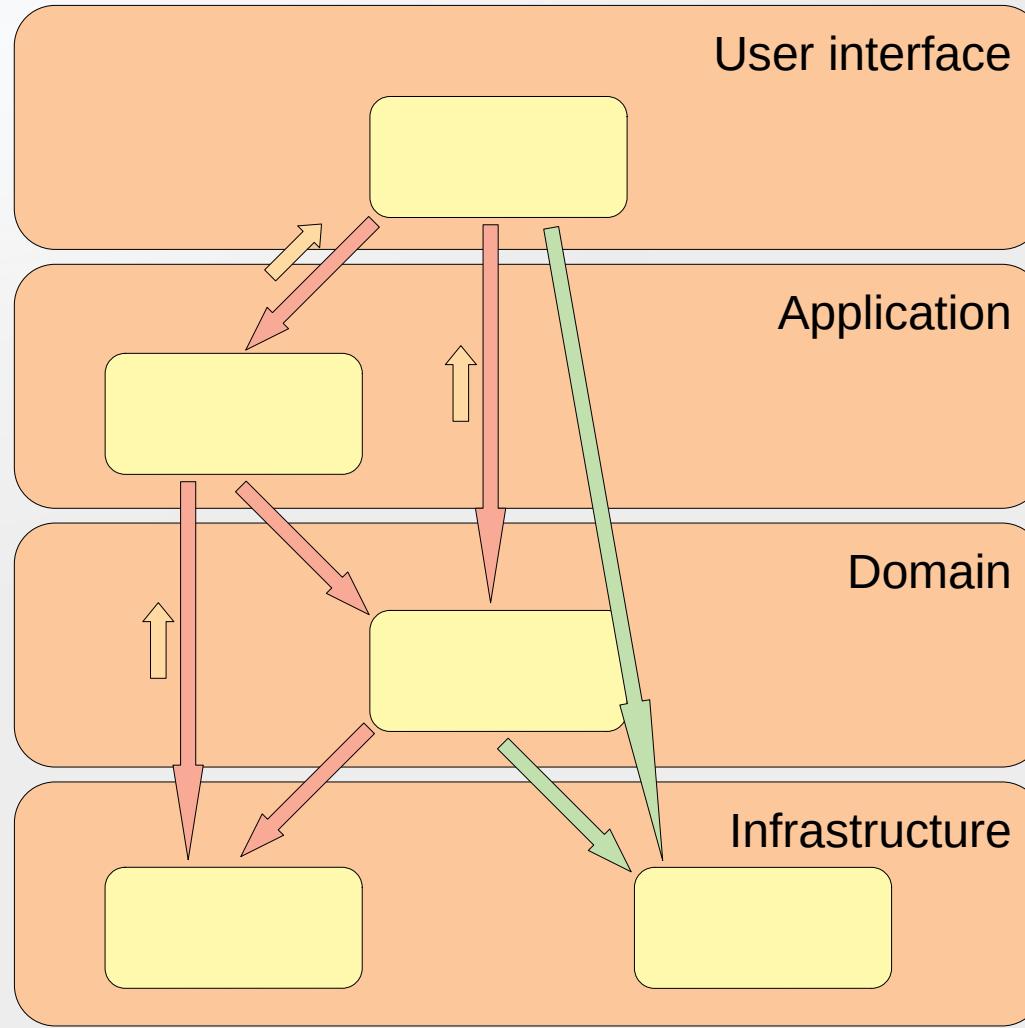


The winner will be notified after the session

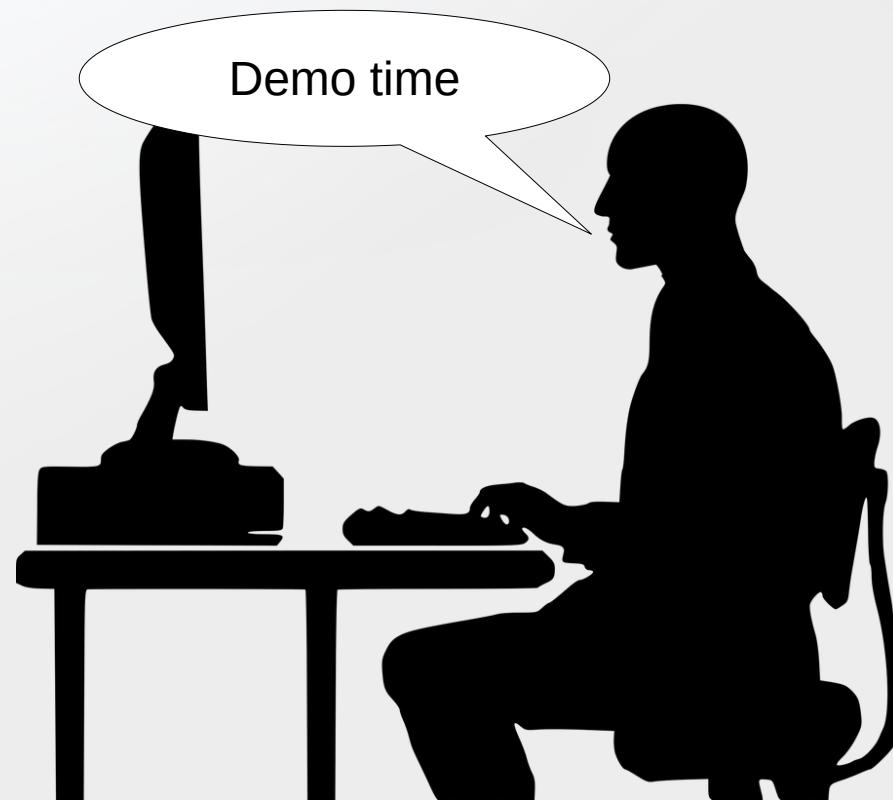
Jedziemy dalej



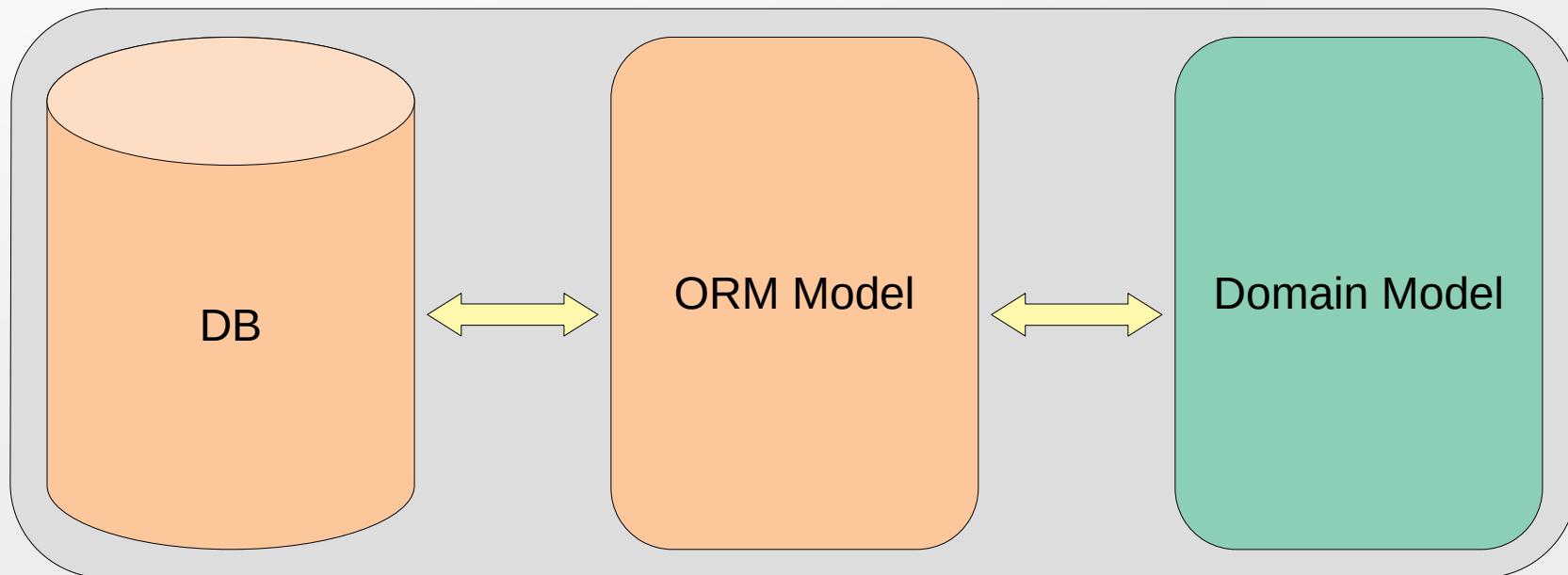
Jest?



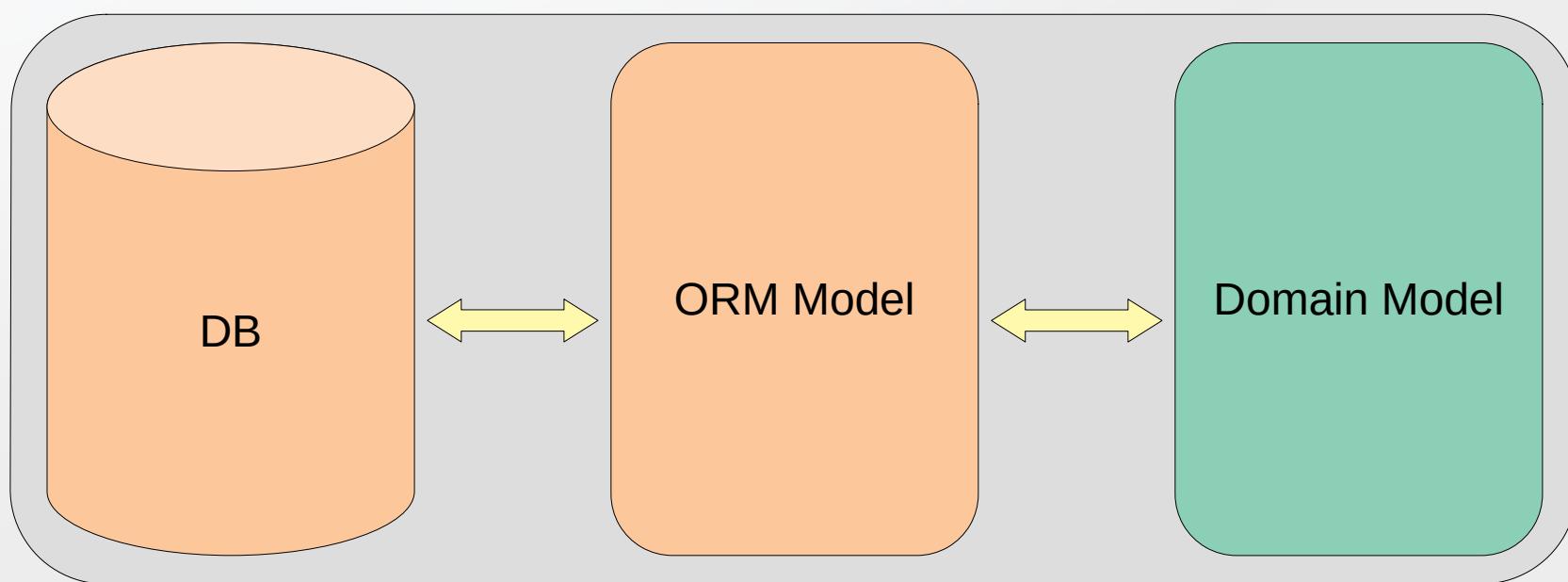
Jeszcze czyściej?



Kłopocik

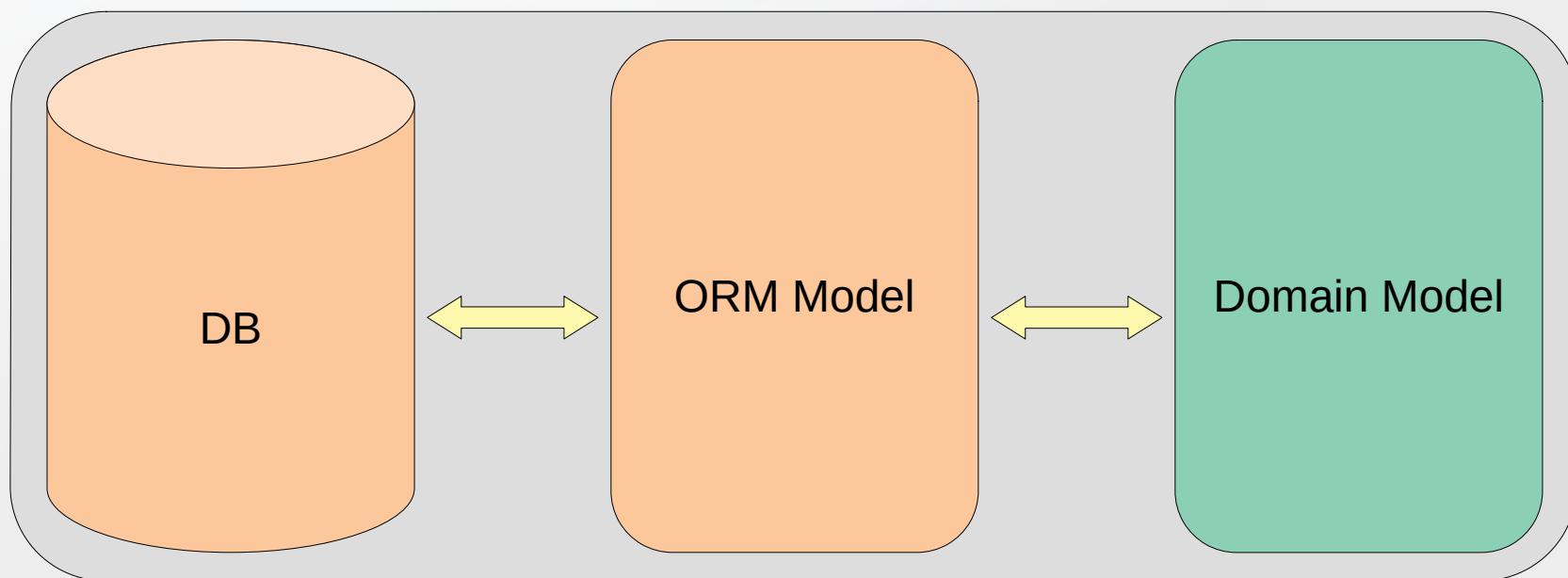


Kłopotek



Połączyć ORM z Domain

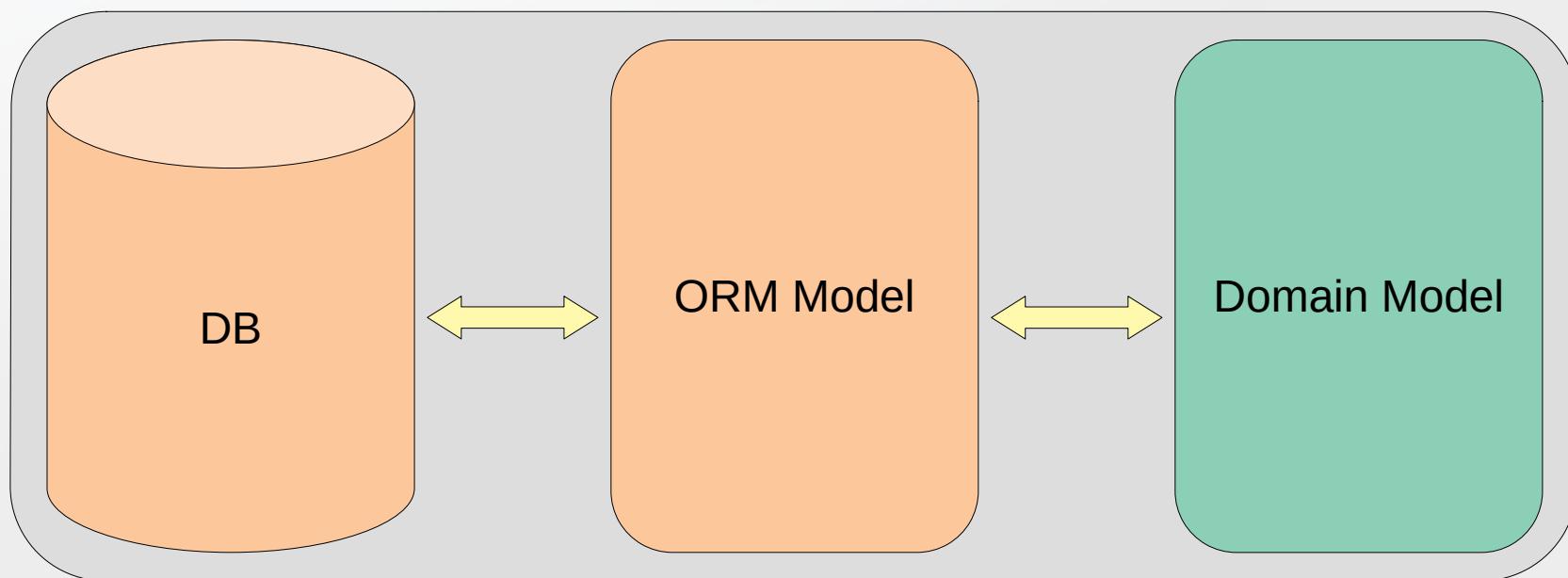
Kłopotek



Połączyć ORM z Domain

Odtwarzać ORM na podstawie Domain

Kłopotek

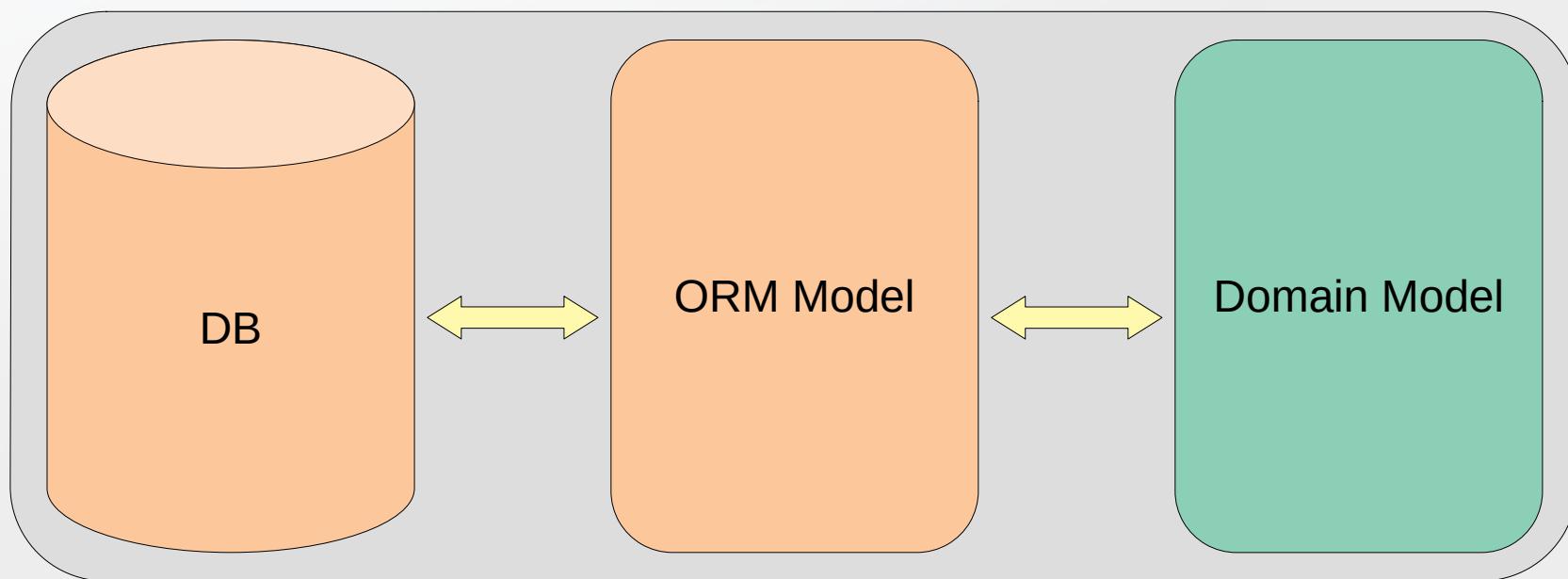


Połączyć ORM z Domain

Odtwarzać ORM na podstawie Domain

Nakładać zdarzenia z Domain na ORM

Kłopotek



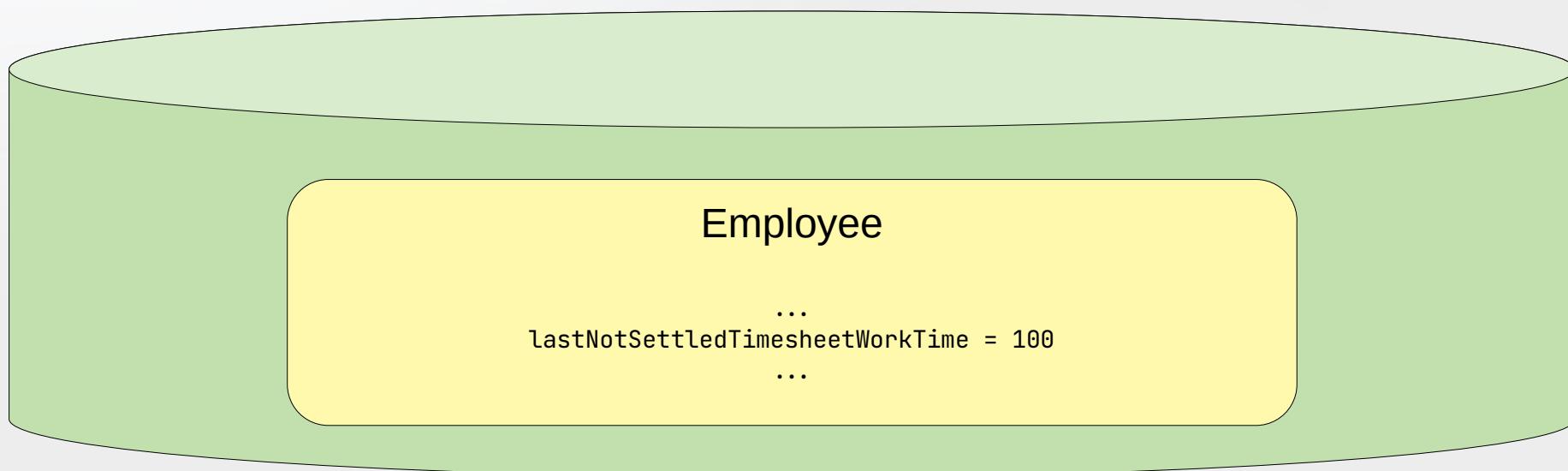
Połączyć ORM z Domain

Odtwarzać ORM na podstawie Domain

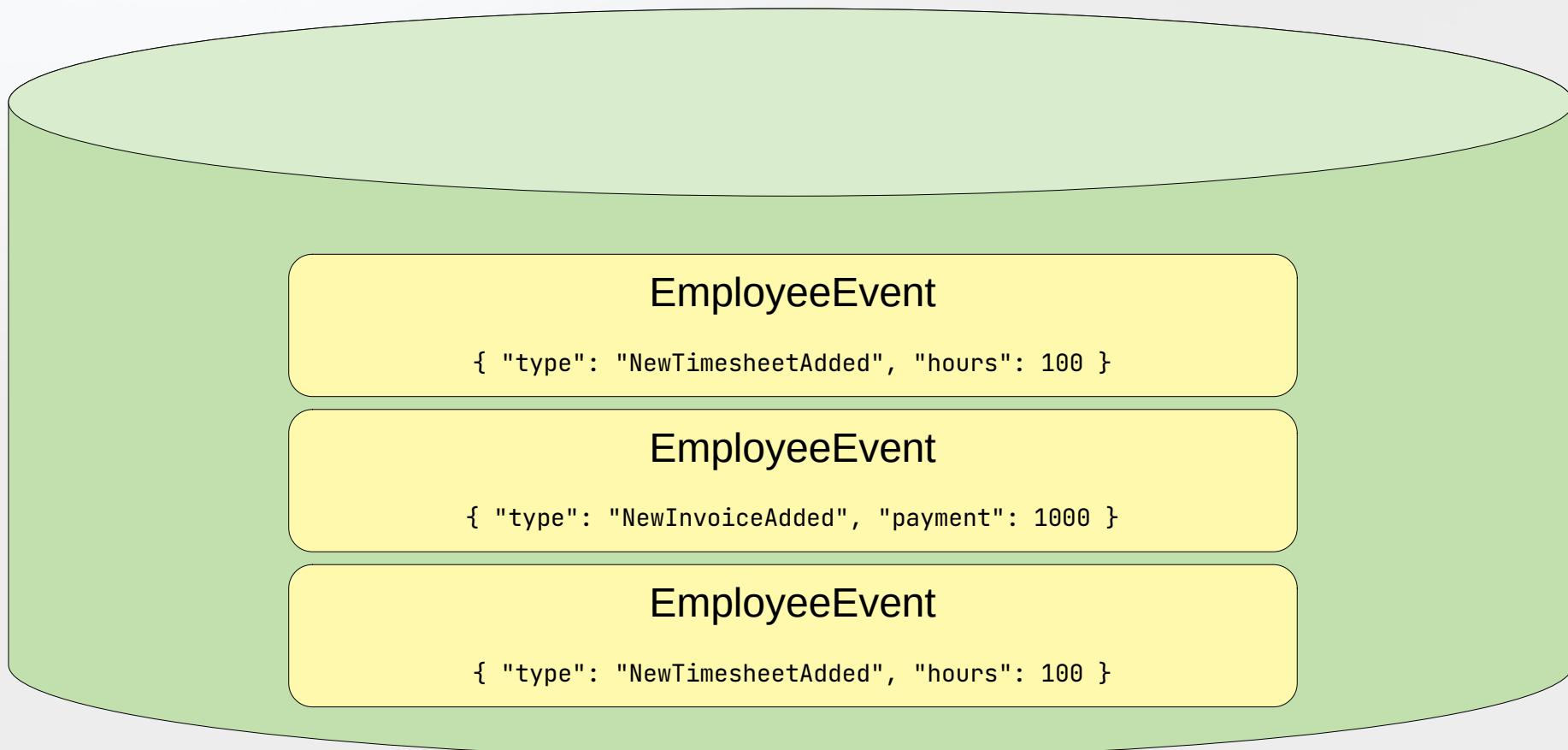
Nakładać zdarzenia z Domain na ORM

Event sourcing

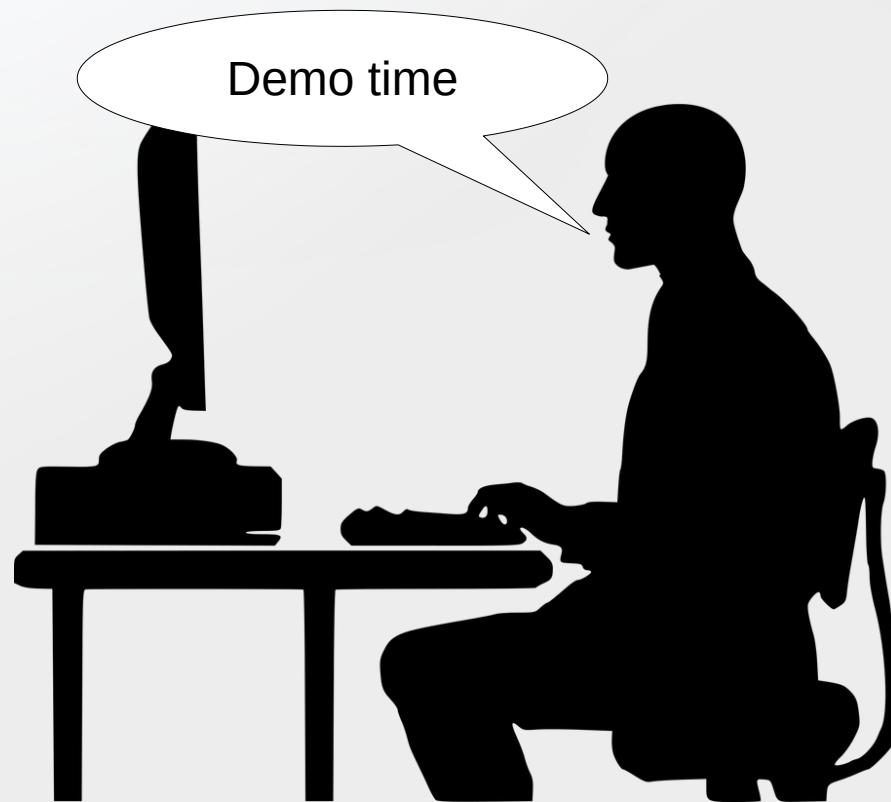
Event sourcing



Event sourcing



Event sourcing



Odwrocenie kontroli

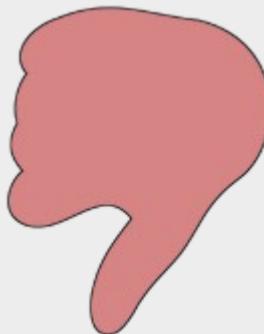
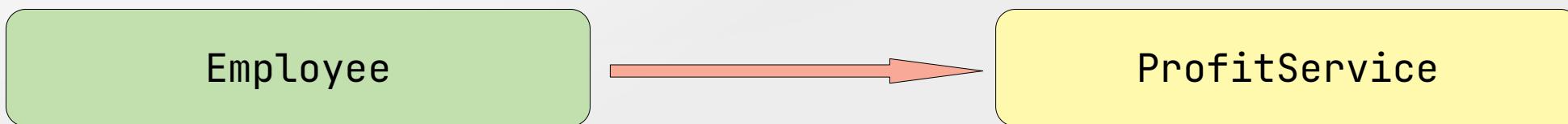
Employee

ProfitService

Odwrocenie kontroli



Odwrocenie kontroli



Odwrocenie kontroli

Employee

ProfitService

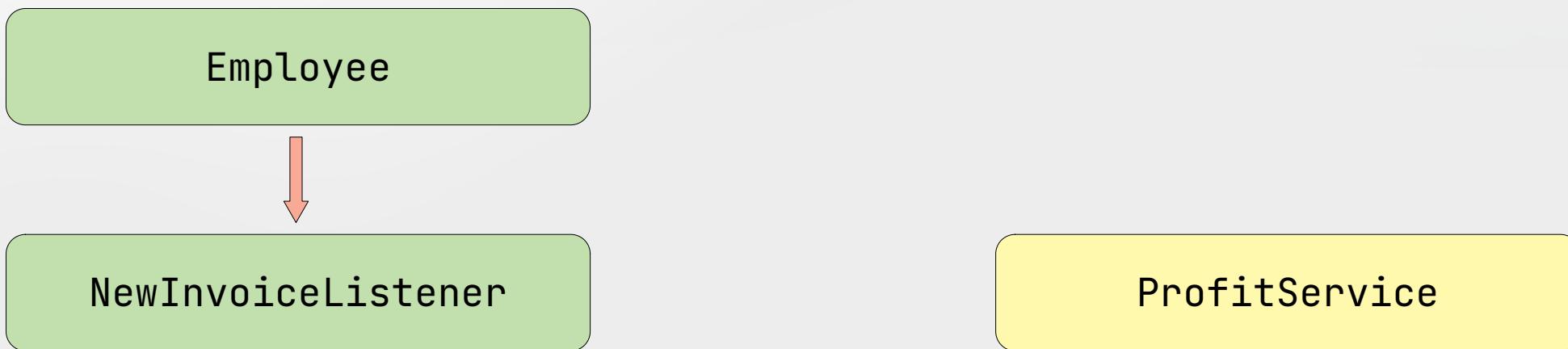
Odwrocenie kontroli

Employee

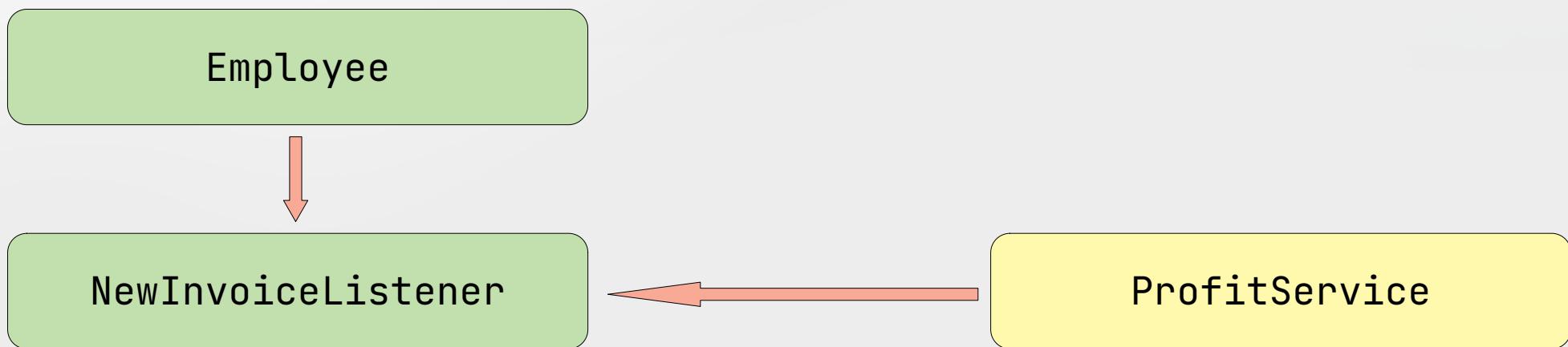
NewInvoiceListener

ProfitService

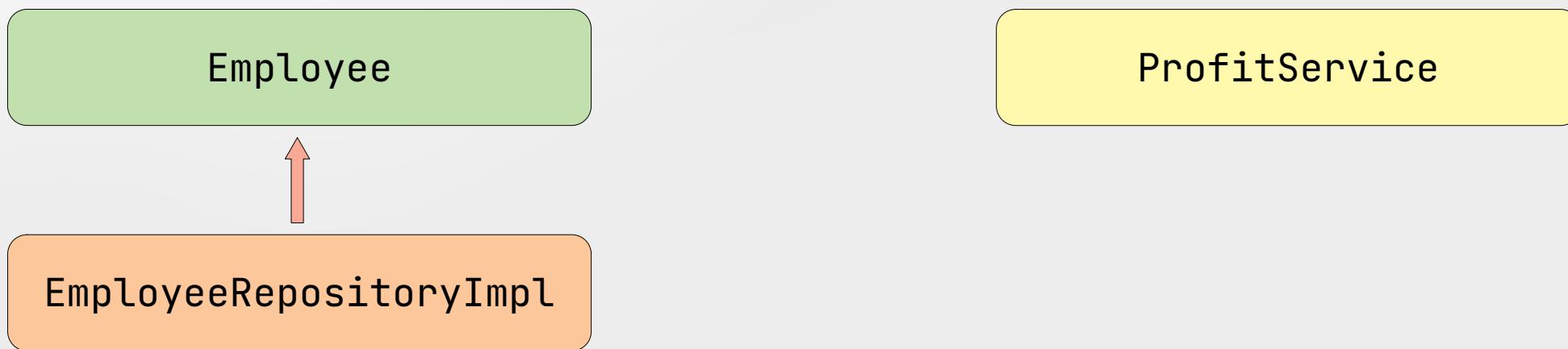
Odwrocenie kontroli



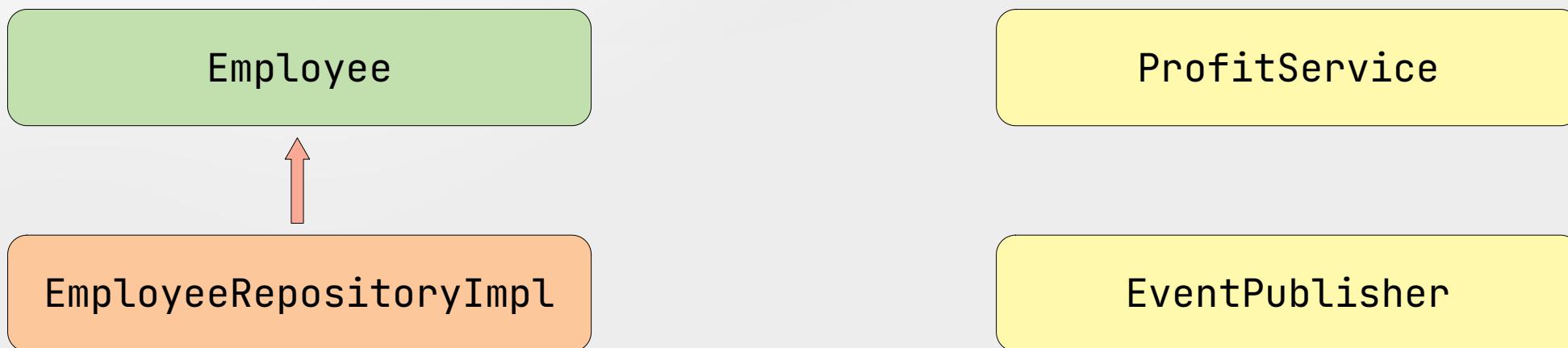
Odwrocenie kontroli



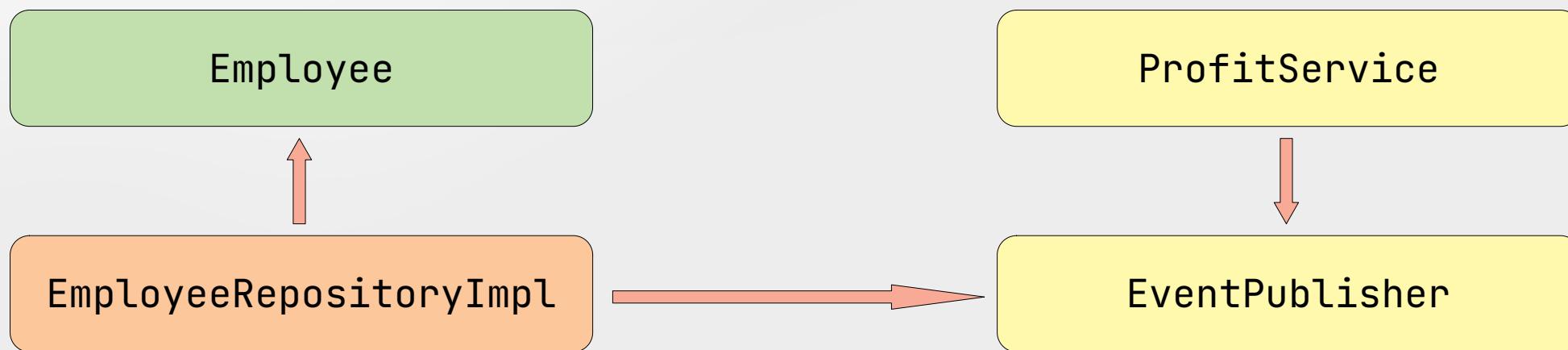
Odwrocenie kontroli



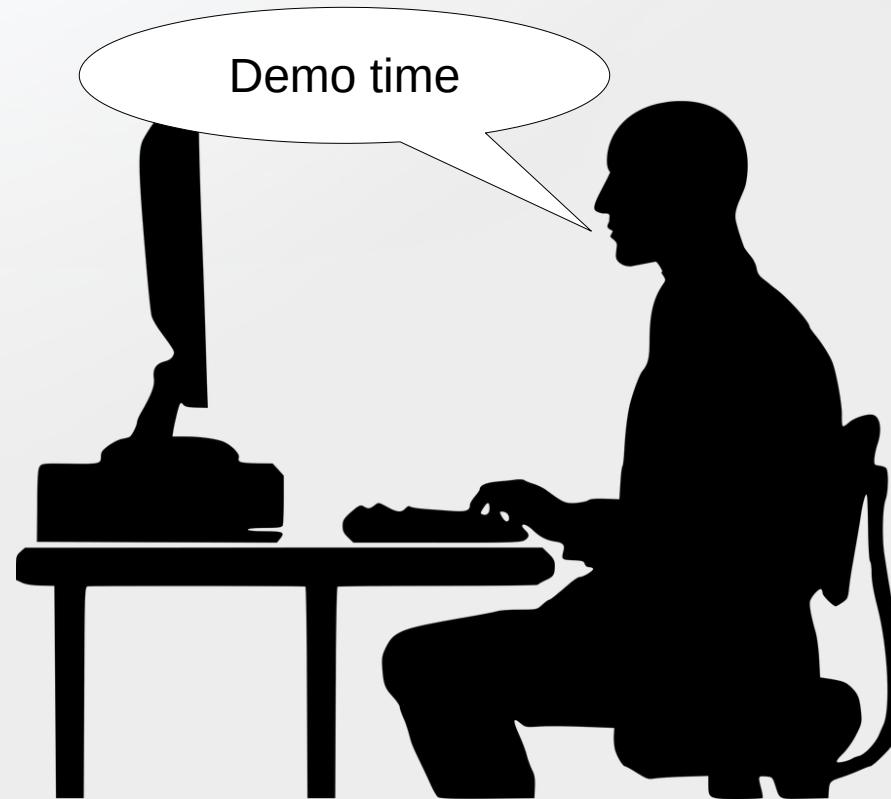
Odwrocenie kontroli



Odwrocenie kontroli



Odwrócenie kontroli



Geneza



Geneza

1997 – JavaBeans Spec

Geneza

1997 – JavaBeans Spec

1997-200x – Frameworks

Geneza

1997 – JavaBeans Spec

1997-200x – Frameworks

2009 – Lombok :)

Geneza

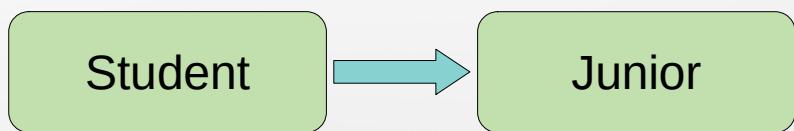
Student



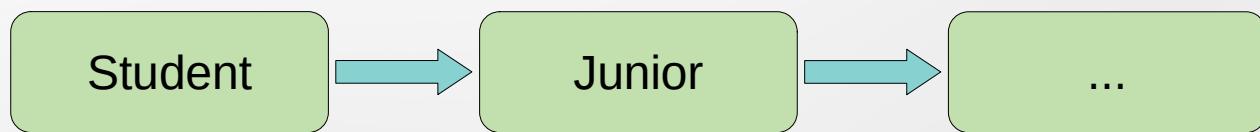
~2000

~2020

Geneza



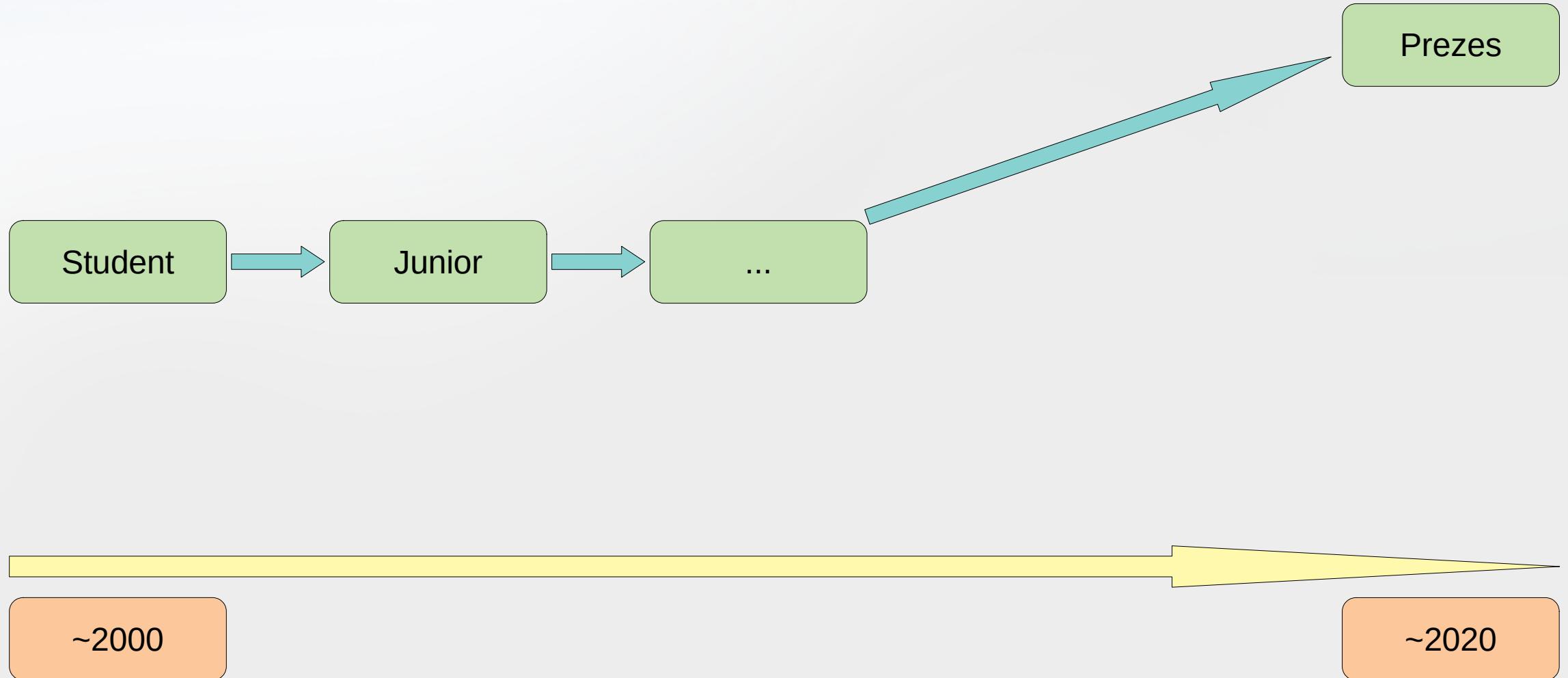
Geneza



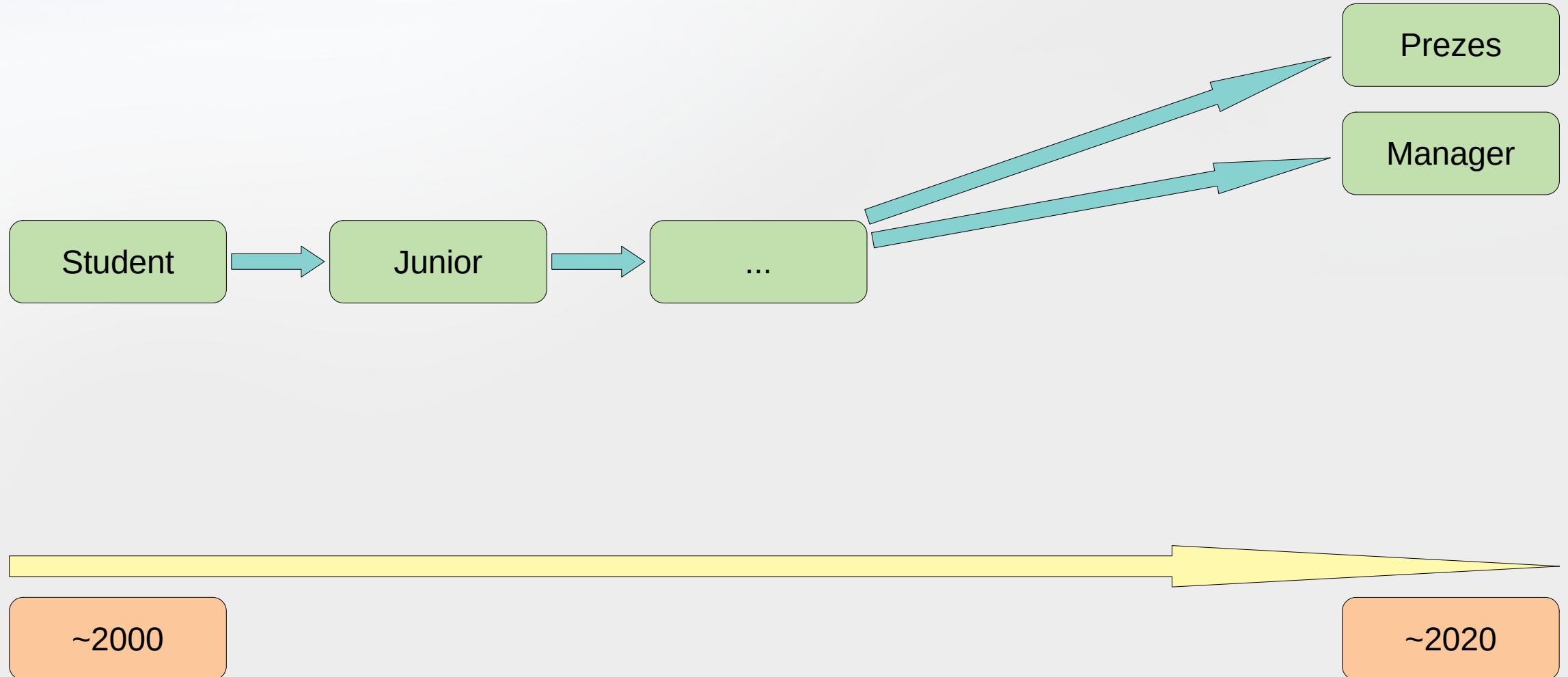
~2000

~2020

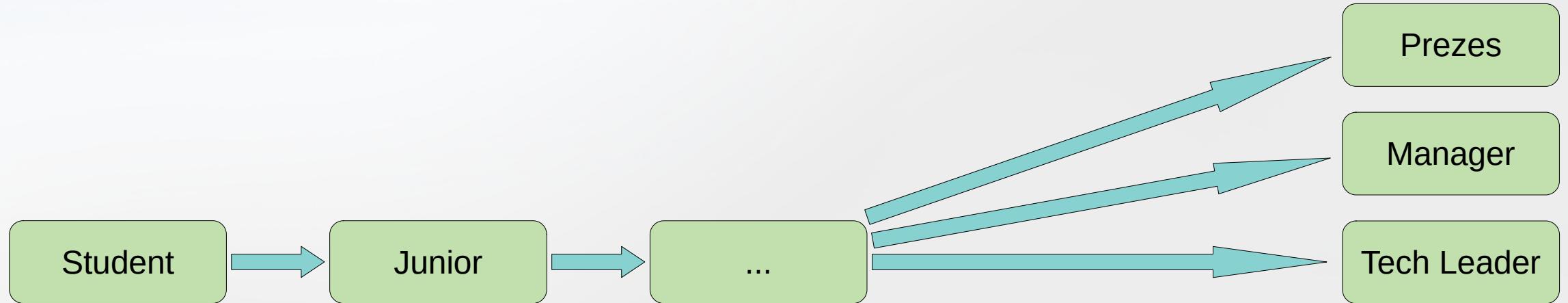
Geneza



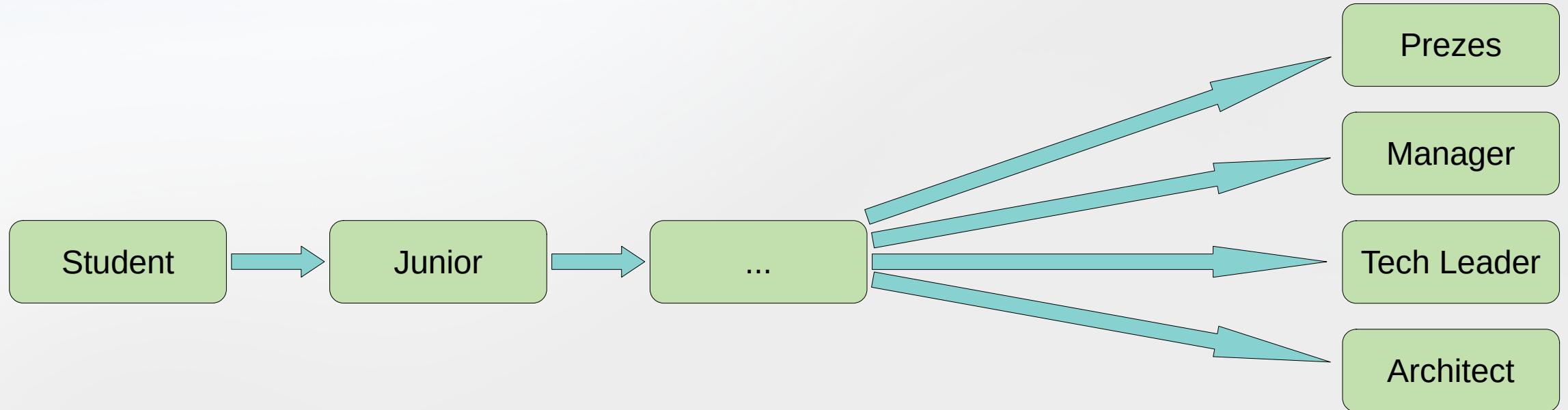
Geneza



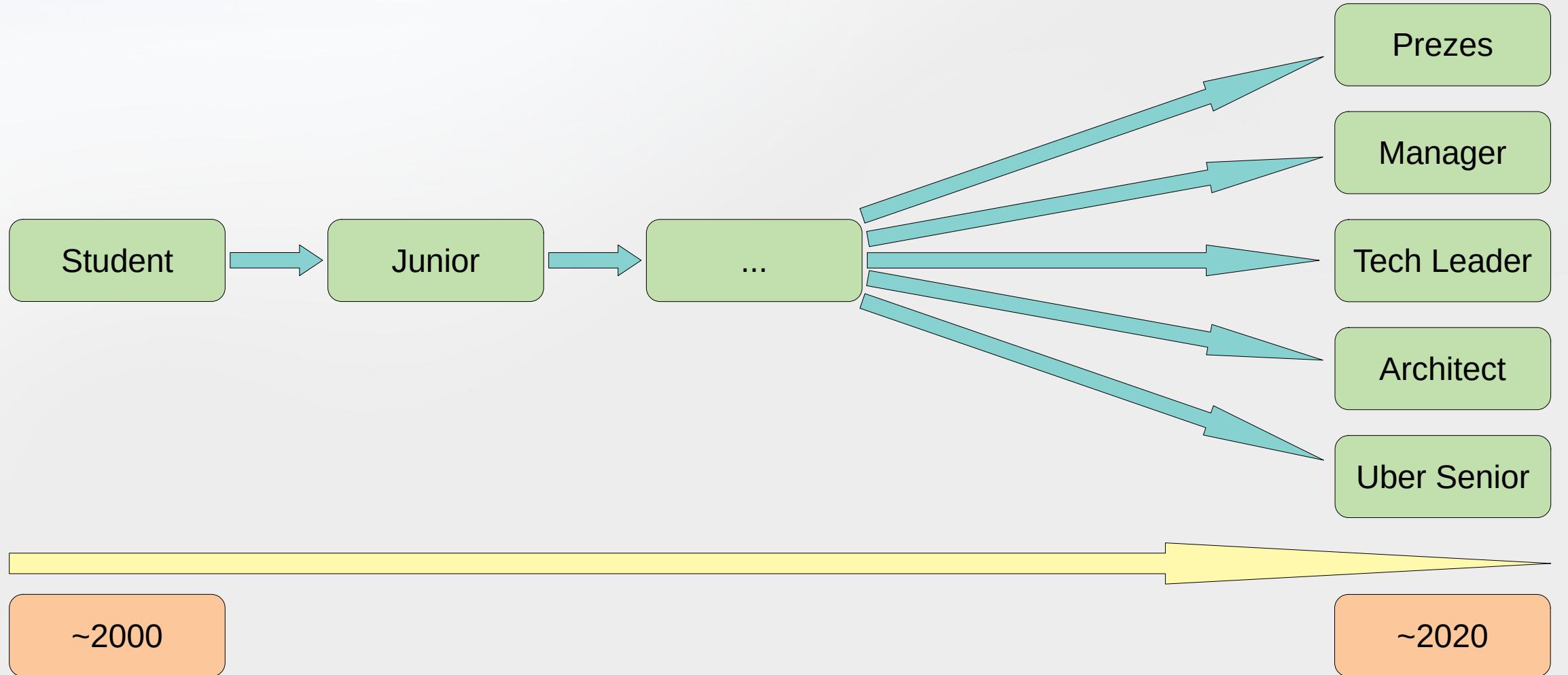
Geneza



Geneza



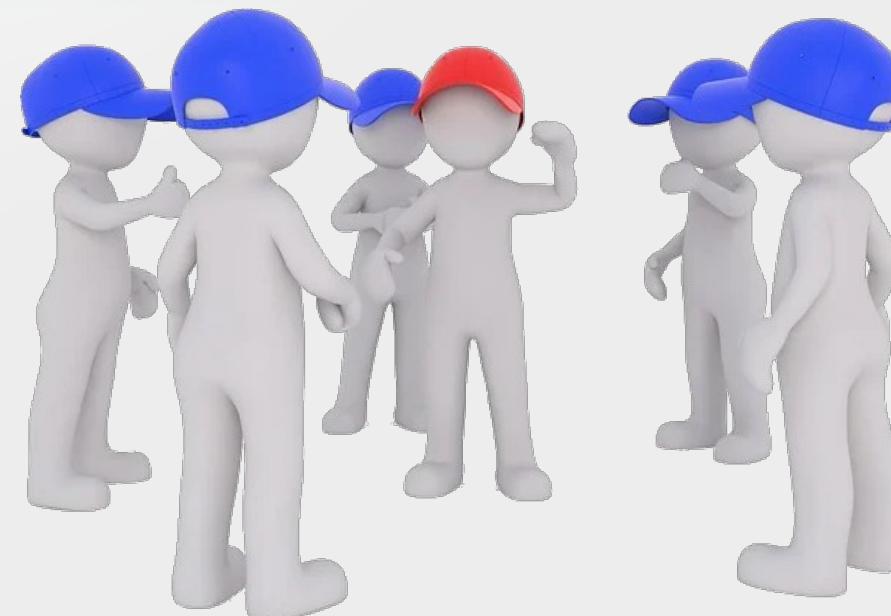
Geneza



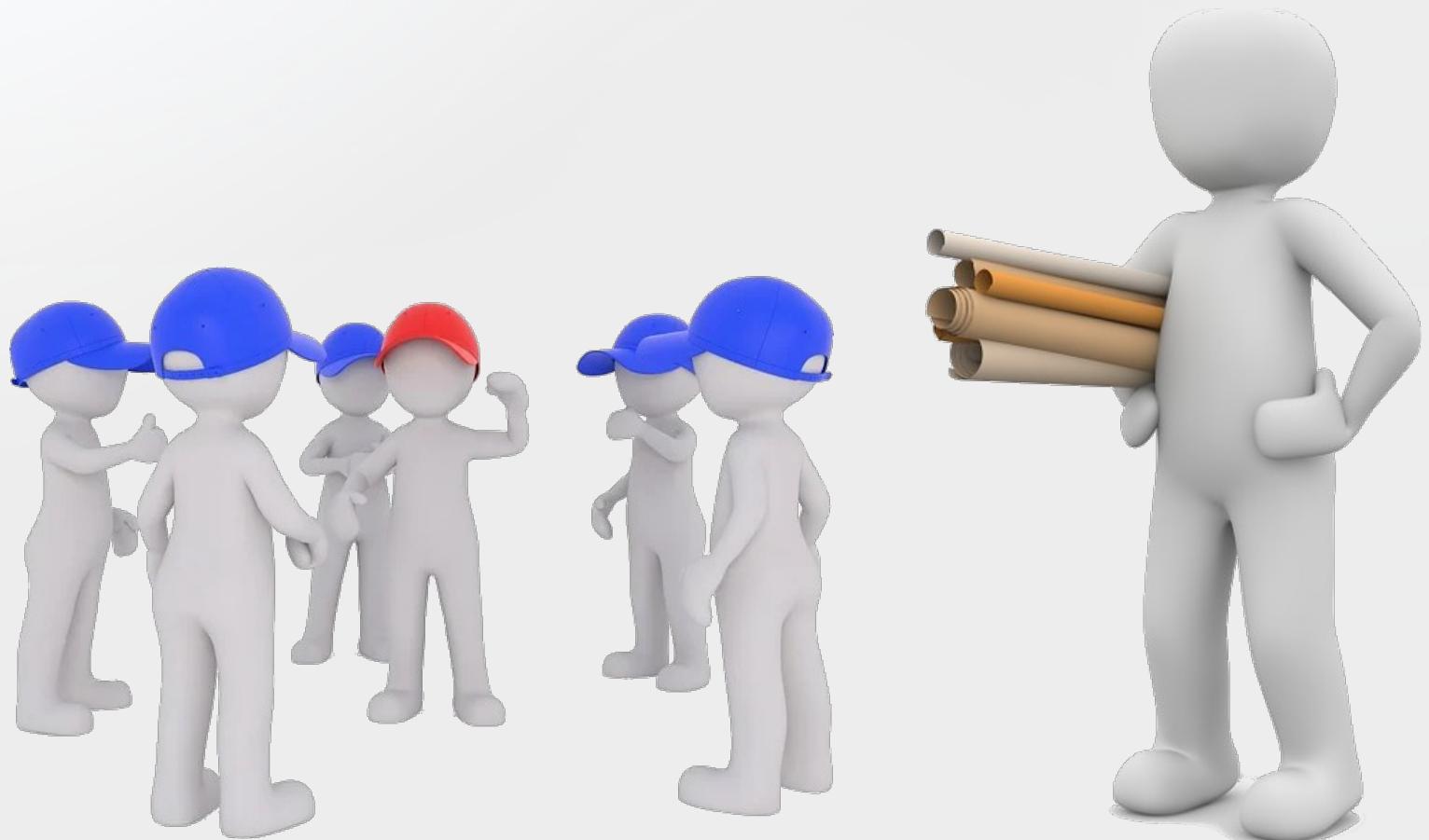
Geneza



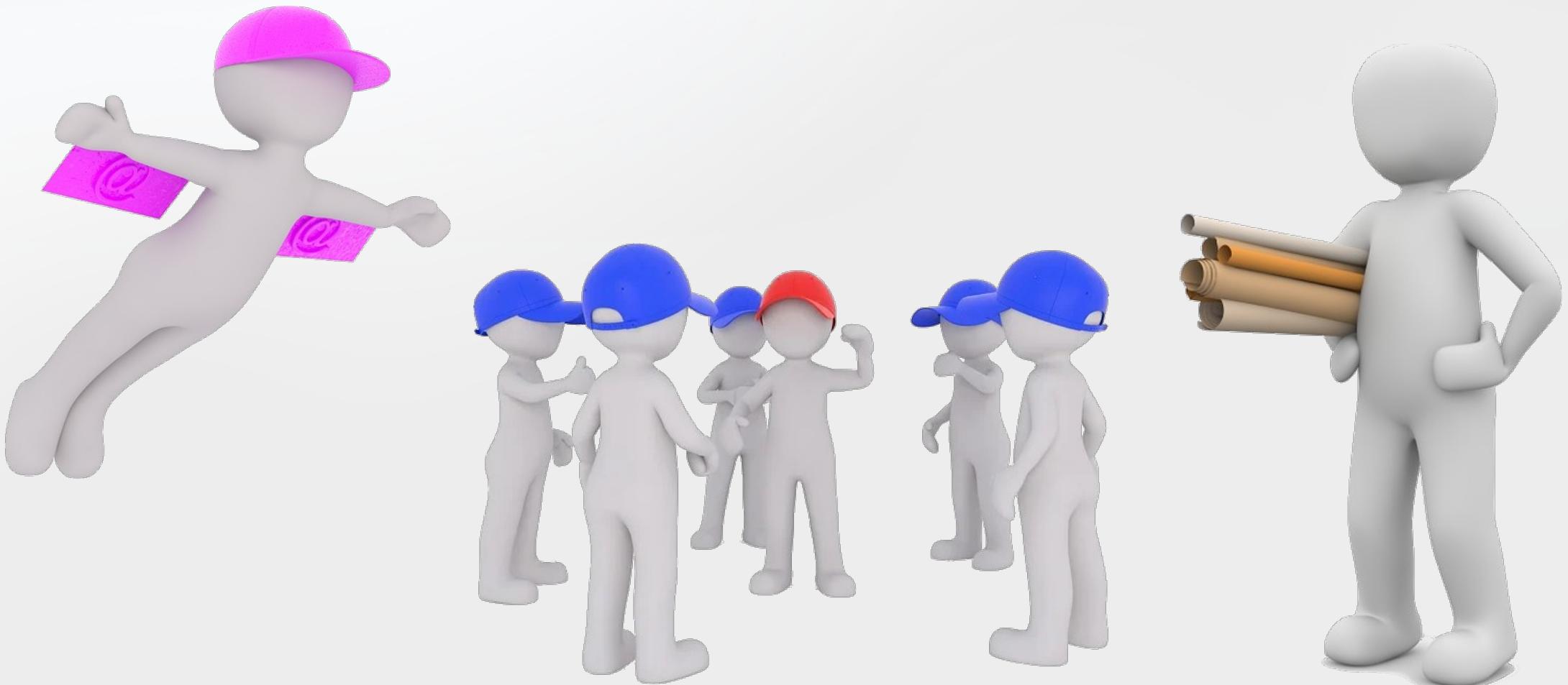
Przepis na sukces



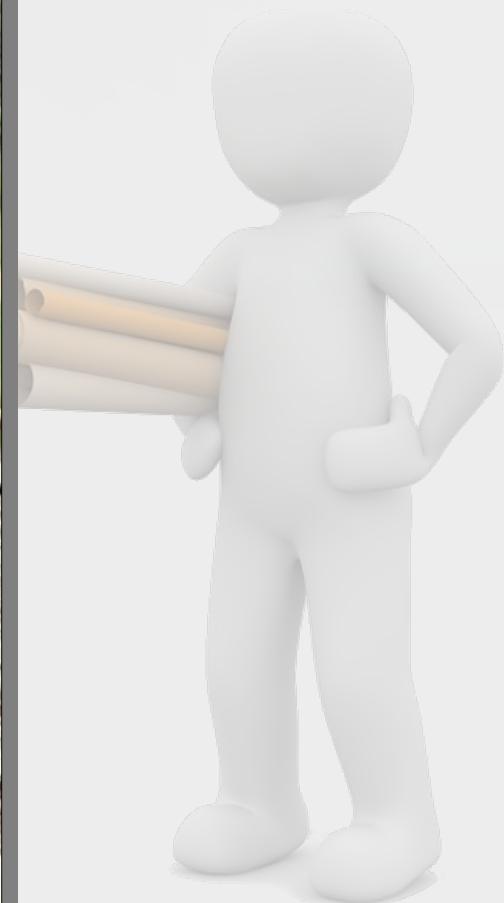
Przepis na sukces



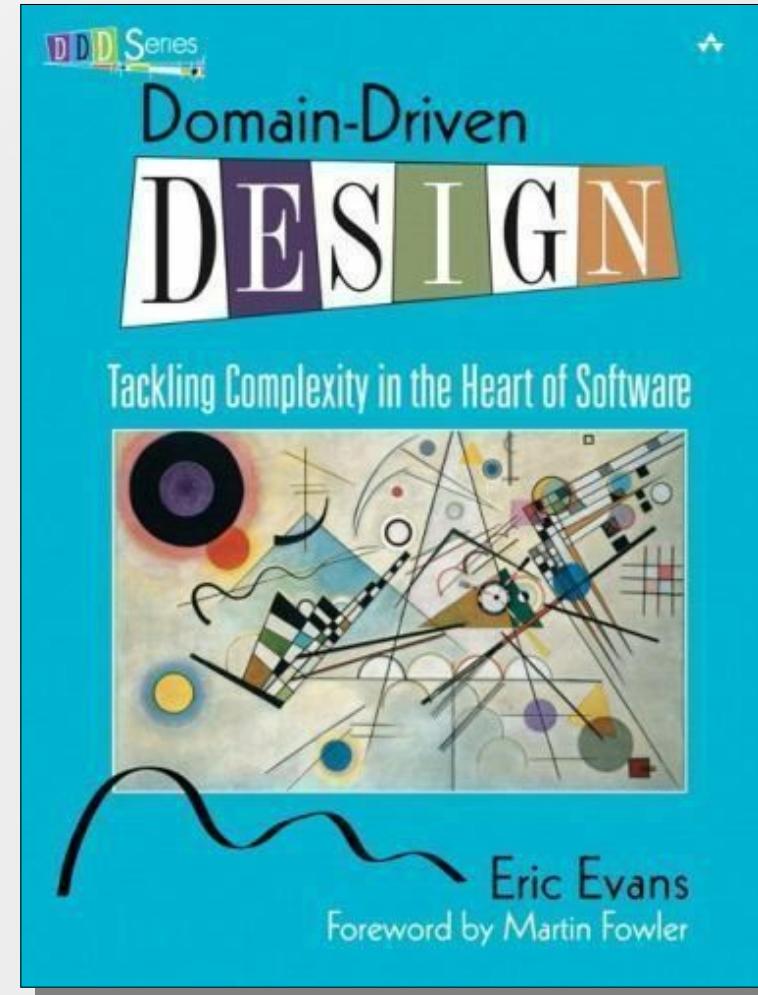
Przepis na sukces



Przepis na sukces



Hard part



keep IT clean

mid-sized building blocks

hexagonal architecture

by Jakub Nabrdalik

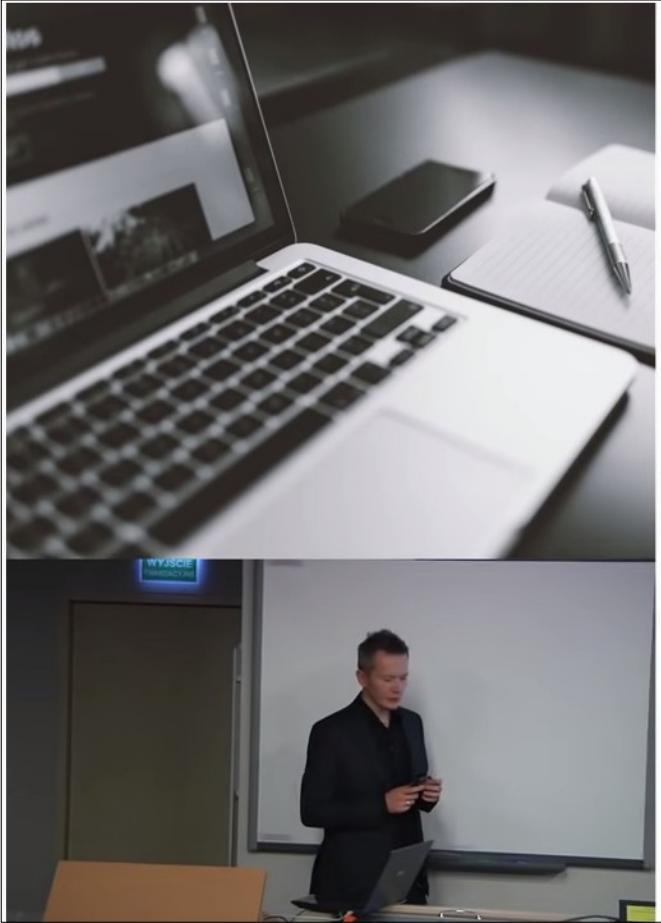
1 / 111

JAKUB NABRDALIK

Modularity and hexagonal architecture in real life

SPONSOR: **allegro**

WJUG #211 - [PL] Modularity and hexagonal architecture in real life: Jakub Nabrdalik



Domain Driven Design - Q&A

Co gryzie świadomego programistę/programistkę?

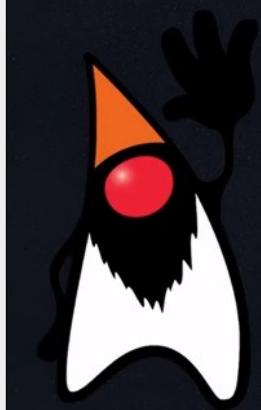
Sławomir Sobótka

BO·TT·EGA
IT minds

bottega.com.pl

BO·TT·EGA
IT minds

Sławomir Sobótka - DDD Q&A - wersja rozszerzona



Łukasz Szydło - Problemy wynikające z modularyzacji.

@WrocławJUG

156. WrocławJUG [MID, PL] - Problemy wynikające z modularyzacji. - Łukasz Szydło

MAKE **IT** GROW



ARCHITEKTONICZNE ANTYWZORCE

czyli na co chorują aplikacje



WWW.4DEVELOPERS.ORG.PL



4Developers 2017: Architecture antipatterns and how to beat them (Łukasz Szydło)

Złote rady wujka Krzyśka



Złote rady wujka Krzyśka



Najważniejsza zasada przy tworzeniu: YAGNI

Złote rady wujka Krzyśka



Najważniejsza zasada przy tworzeniu: YAGNI

Znajdź wady „technik” zanim ich użyjesz

Złote rady wujka Krzyśka



Najważniejsza zasada przy tworzeniu: YAGNI

Znajdź wady „technik” zanim ich użyjesz

Enkapsulacja / hermetyzacja

Złote rady wujka Krzyśka



Najważniejsza zasada przy tworzeniu: YAGNI

Znajdź wady „technik” zanim ich użyjesz

Enkapsulacja / hermetyzacja

Dobry moduł od złego odróżnia jego rozmiar

Złote rady wujka Krzyśka



Najważniejsza zasada przy tworzeniu: YAGNI

Znajdź wady „technik” zanim ich użyjesz

Enkapsulacja / hermetyzacja

Dobry moduł od złego odróżnia jego rozmiar

Testy – tak, ale tam gdzie są przydatne

Złote rady wujka Krzyśka



Najważniejsza zasada przy tworzeniu: YAGNI

Znajdź wady „technik” zanim ich użyjesz

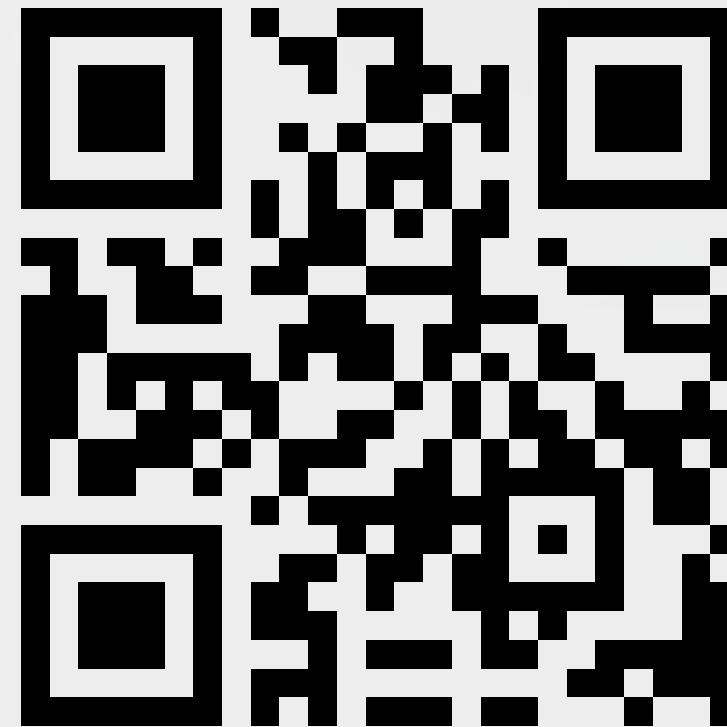
Enkapsulacja / hermetyzacja

Dobry moduł od złego odróżnia jego rozmiar

Testy – tak, ale tam gdzie są przydatne

CRUD – to jest dawno rozwiązyany problem

Feedback please



<https://bit.ly/wjug-292>

<http://kś.pl>
<http://gclogs.com>

conf@kś.pl
ks@gclogs.com

Dziękuję



<https://bit.ly/wjug-292>