

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

Krzysztof Ślusarski

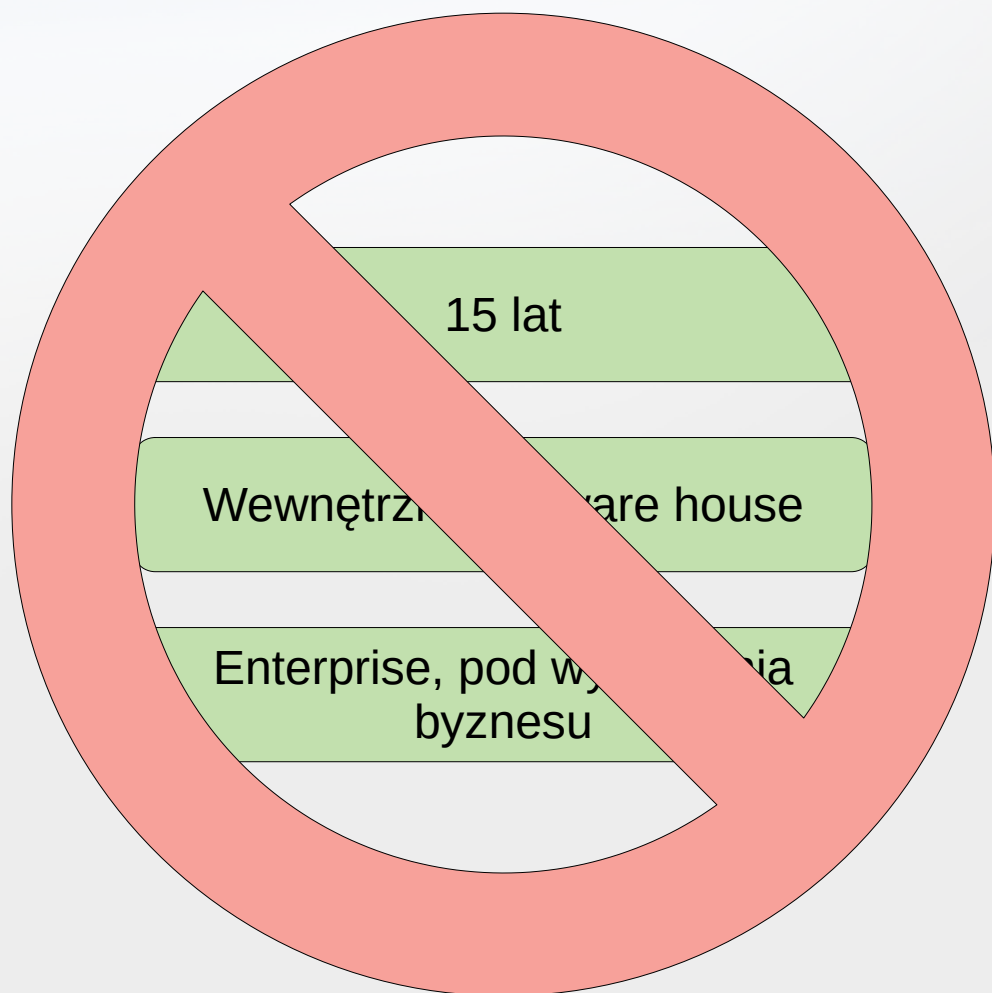
Krótko o mnie

Programowanie – od 1992

Zawodowo (JVM) – od 2006

Szkolenia, Profiling, Diagnoza awarii

Zawodowo



Zawodowo

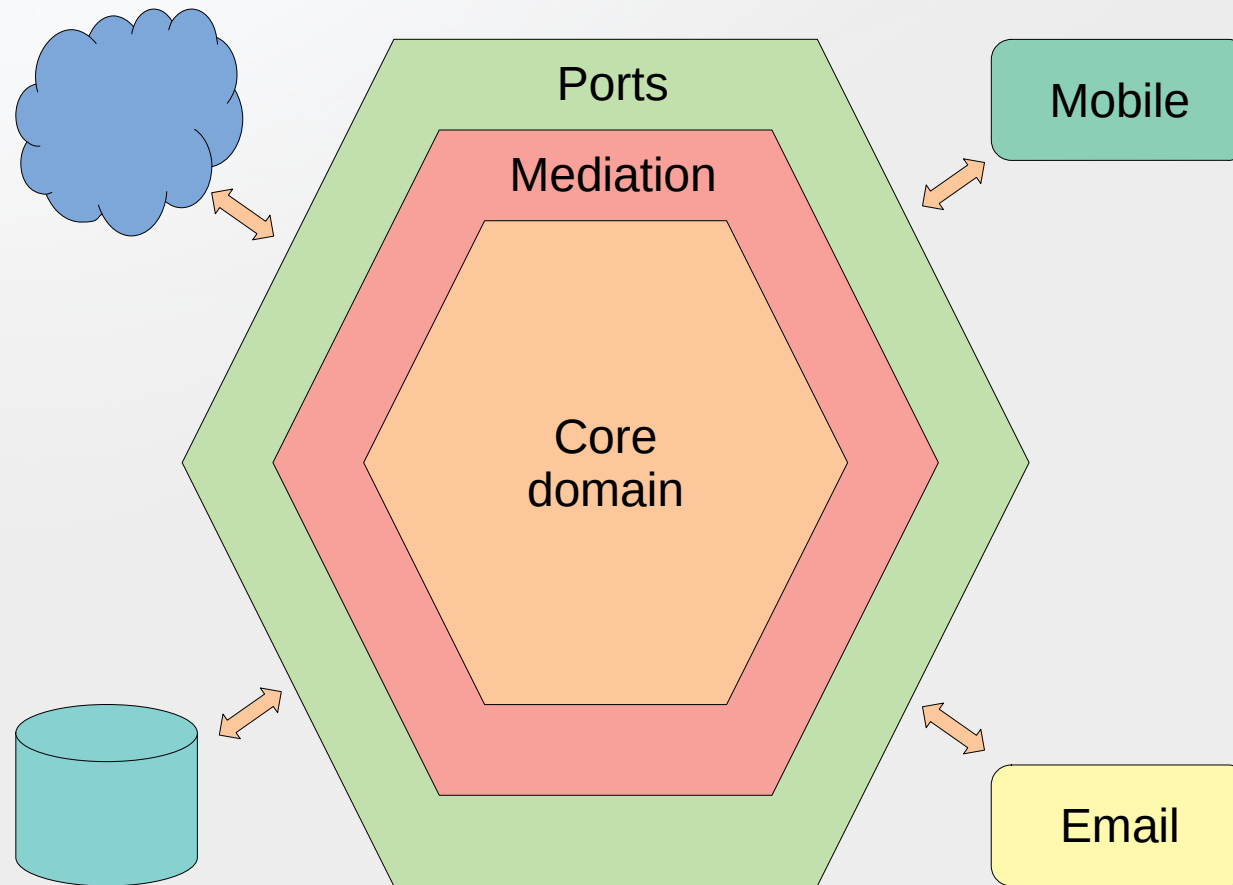


Zawodowo

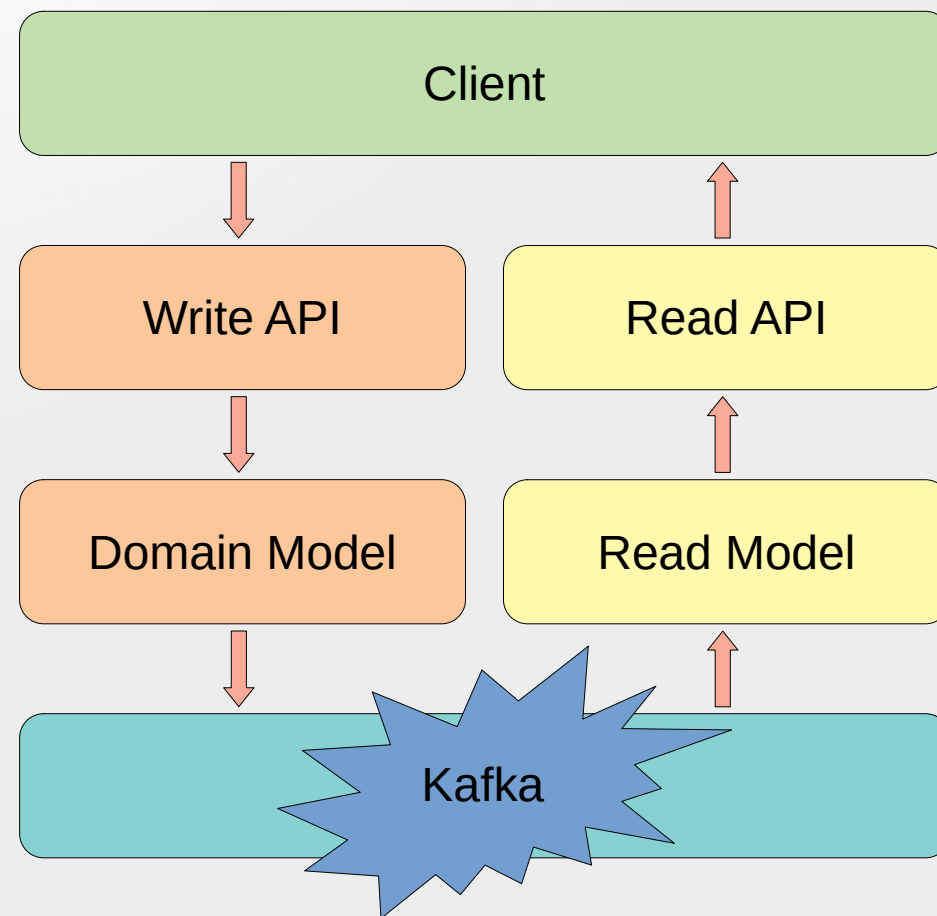


hazelcast

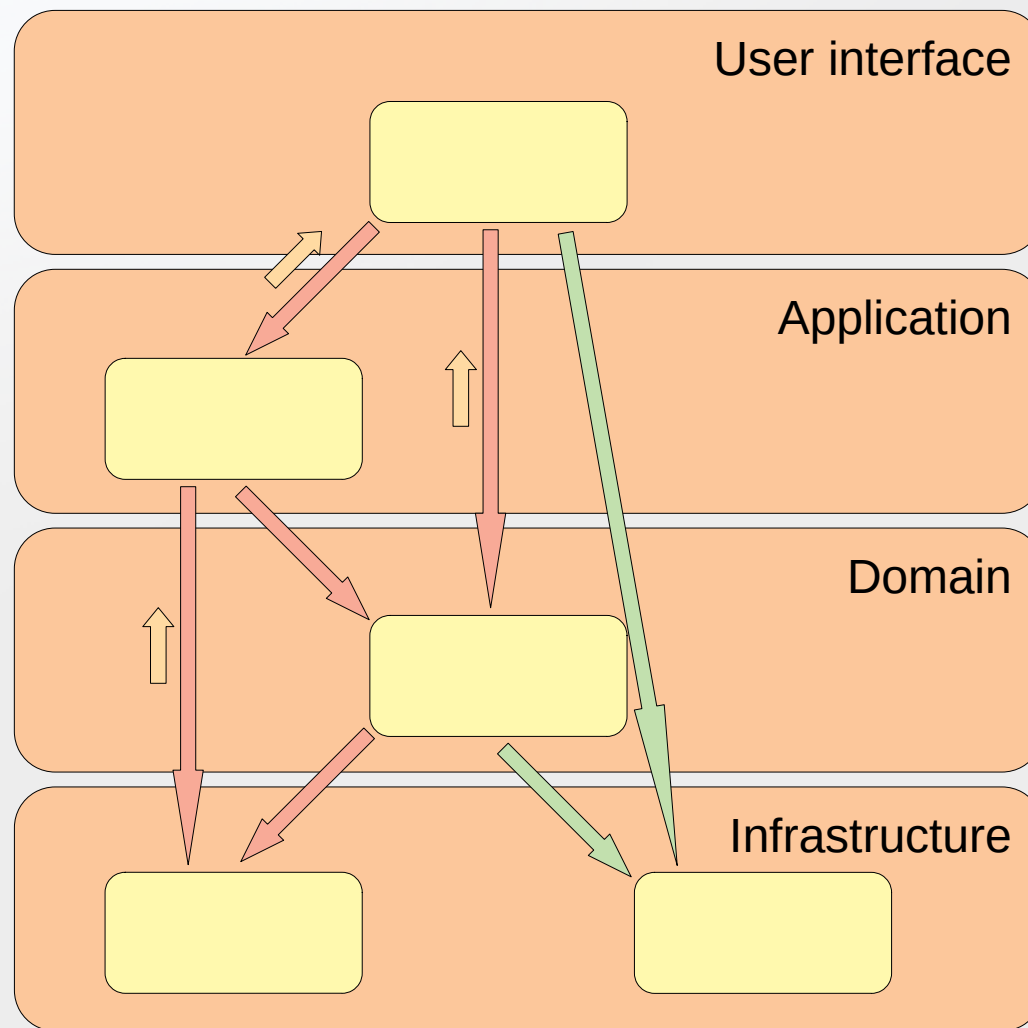
Cel



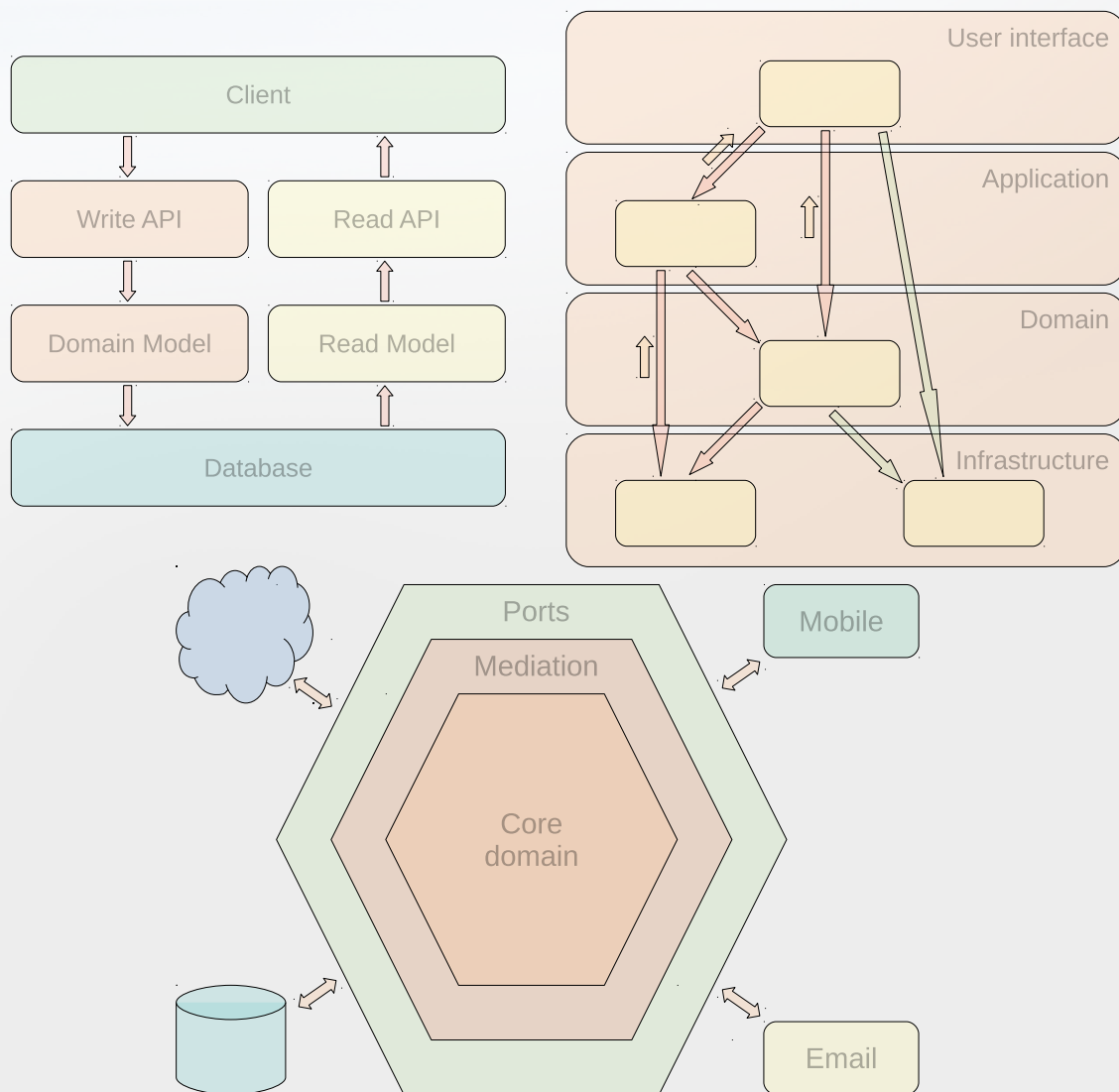
Cel



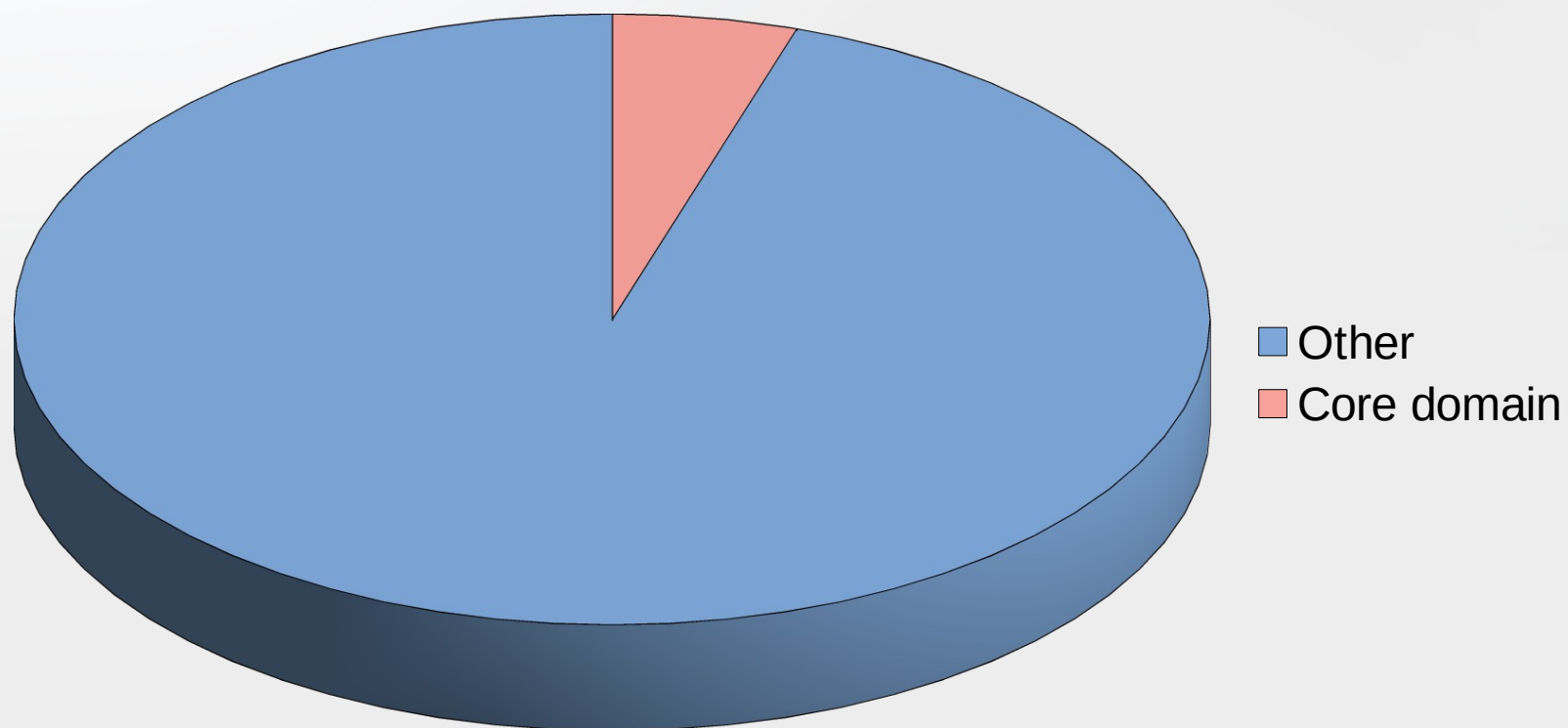
Cel



Cel



Cel



Domena



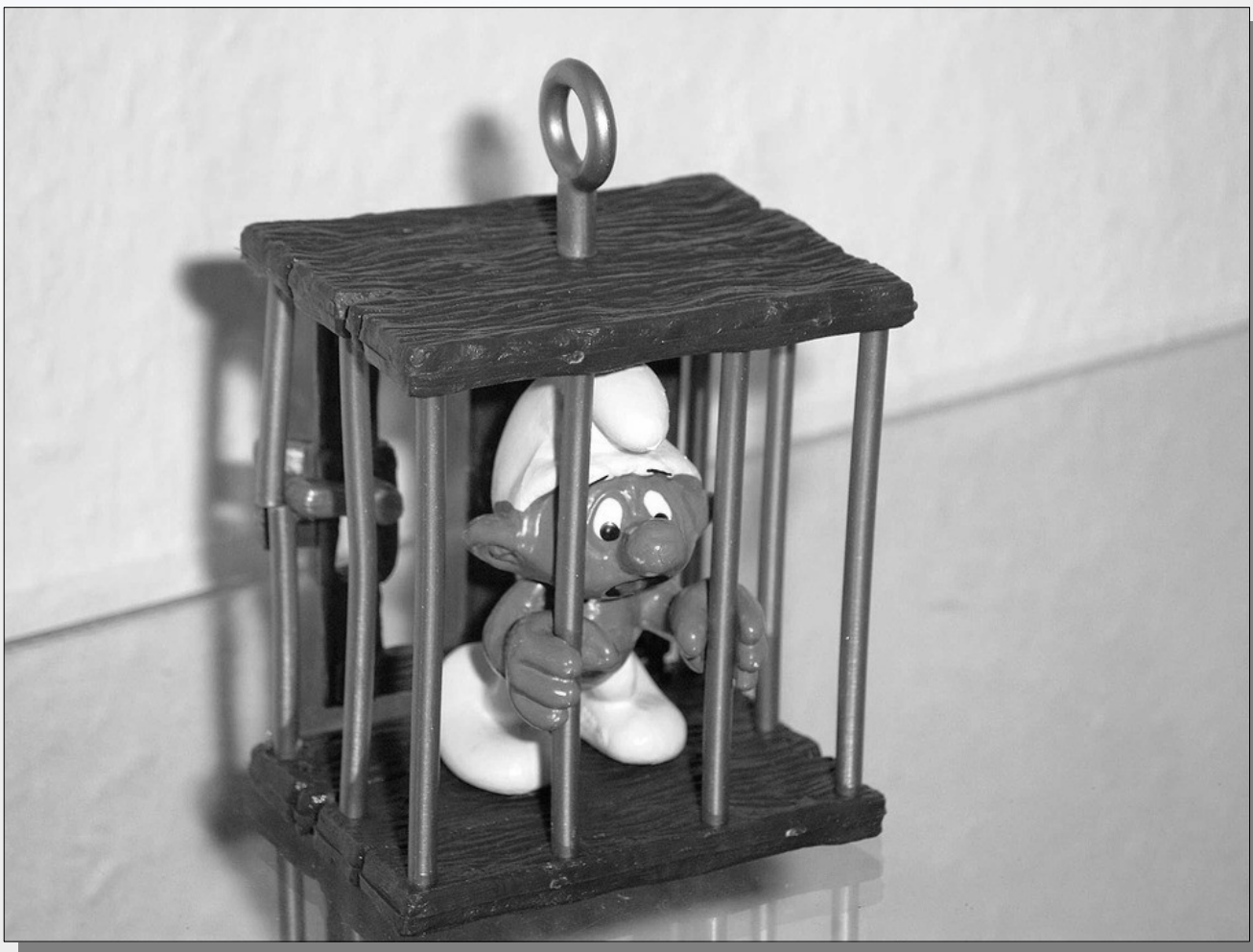
Domena

Wytwórz moją wyobraźnię

Wszelkie podobieństwa co do osób i zdarzeń
są przypadkowe

Brak politycznej poprawności

Domena



Domena



Domena



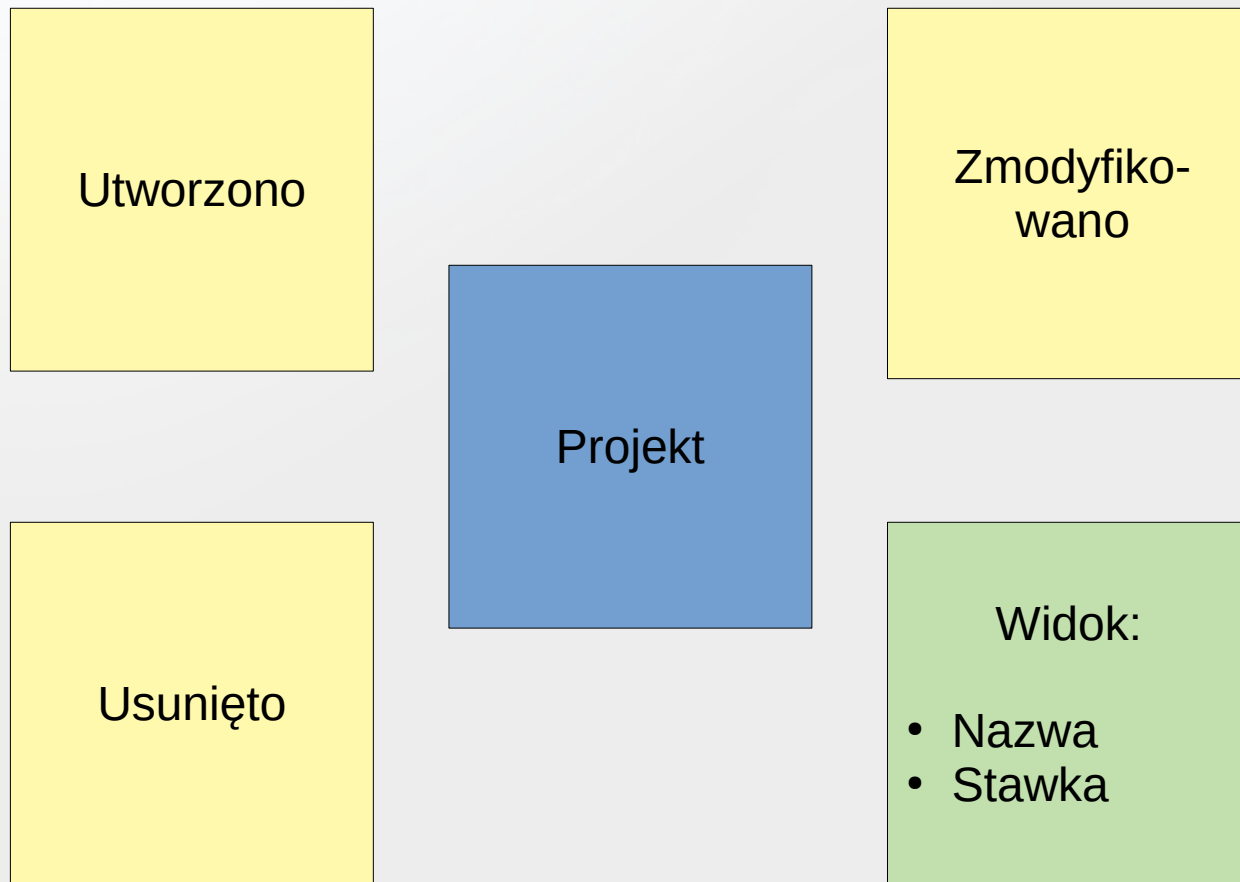
Janusz

Lat 40

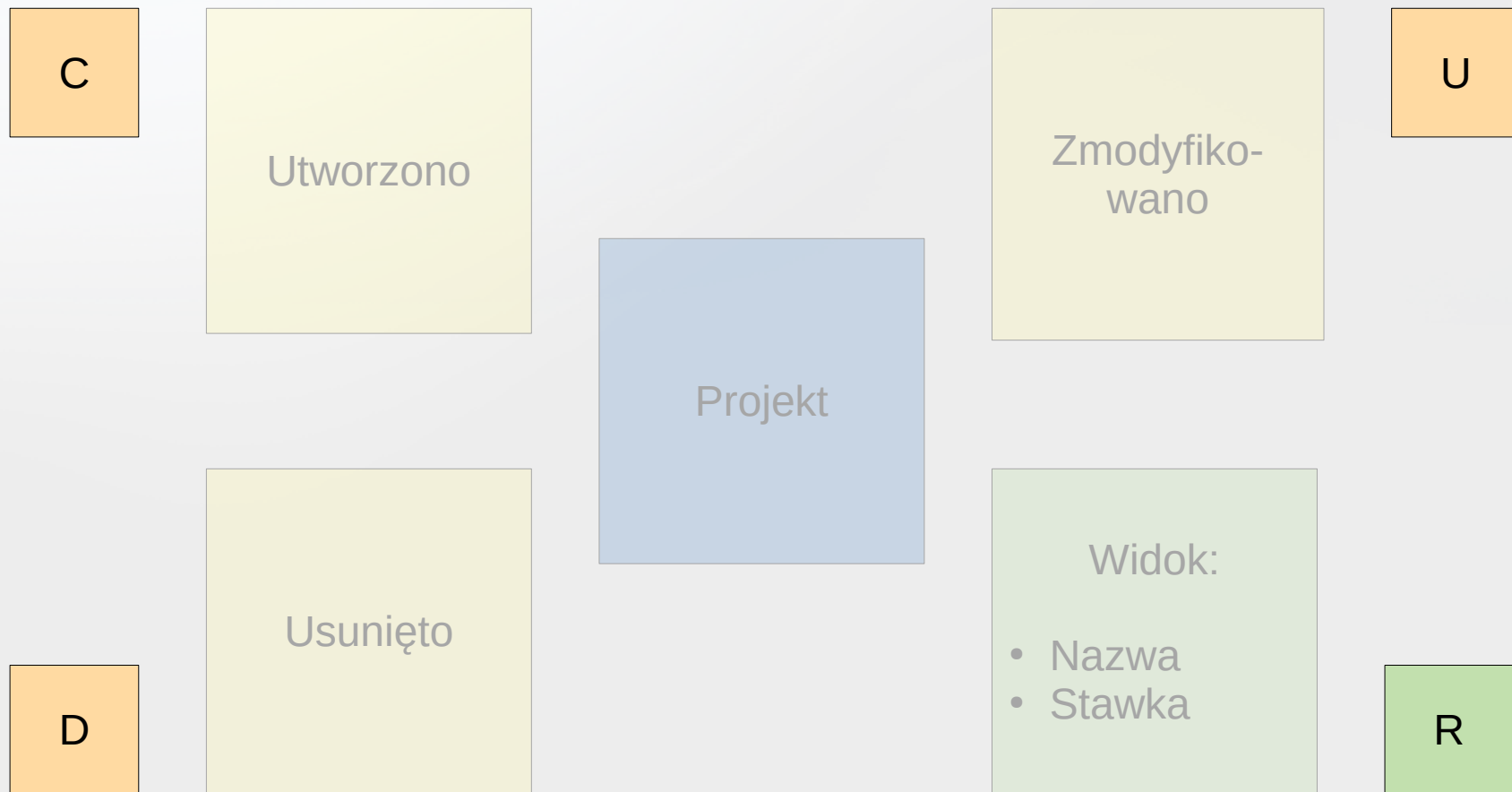
Prezes firmy BodyLeasingowej od 15 lat

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

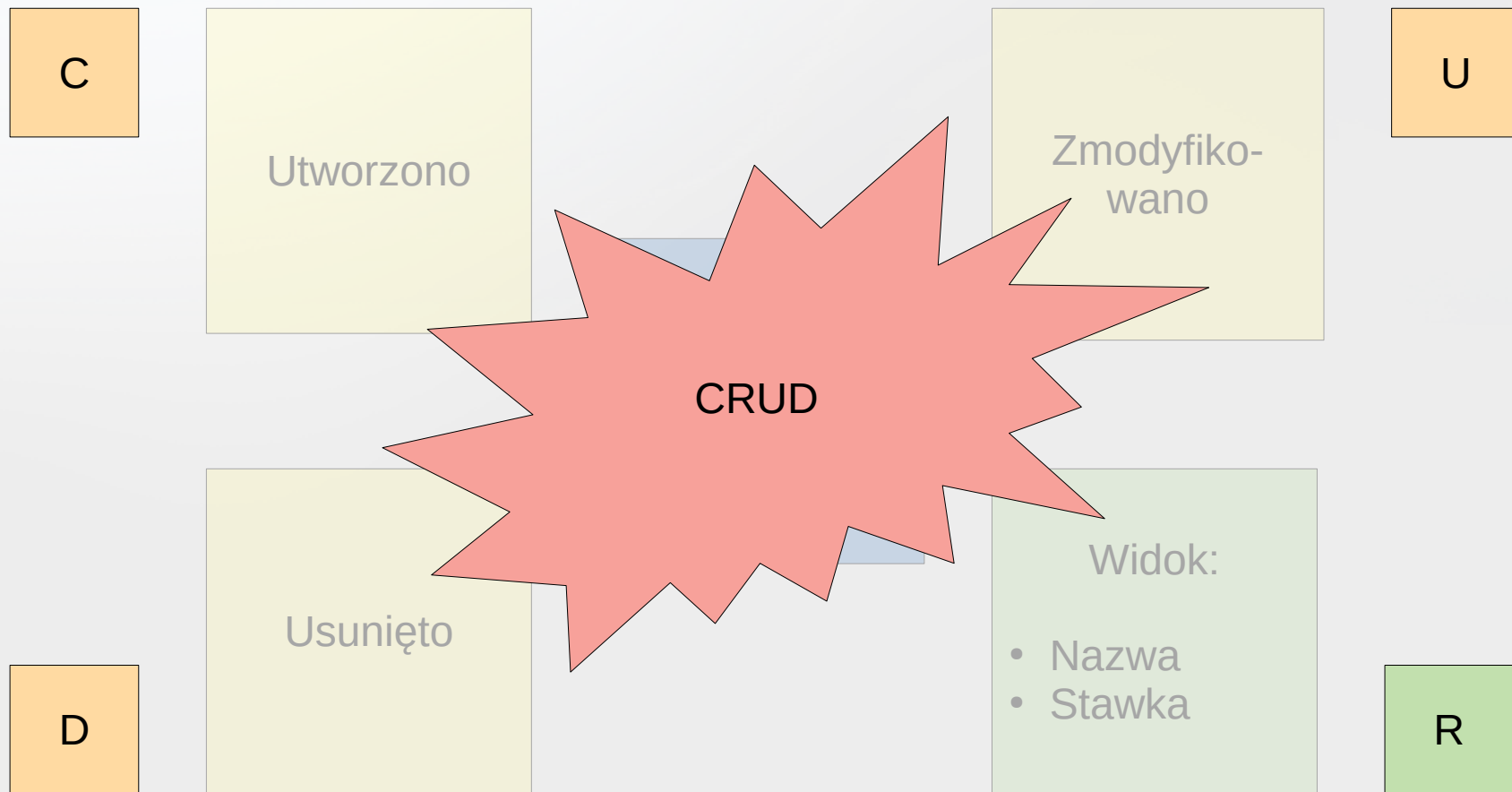
Domena



Domena



Domena



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

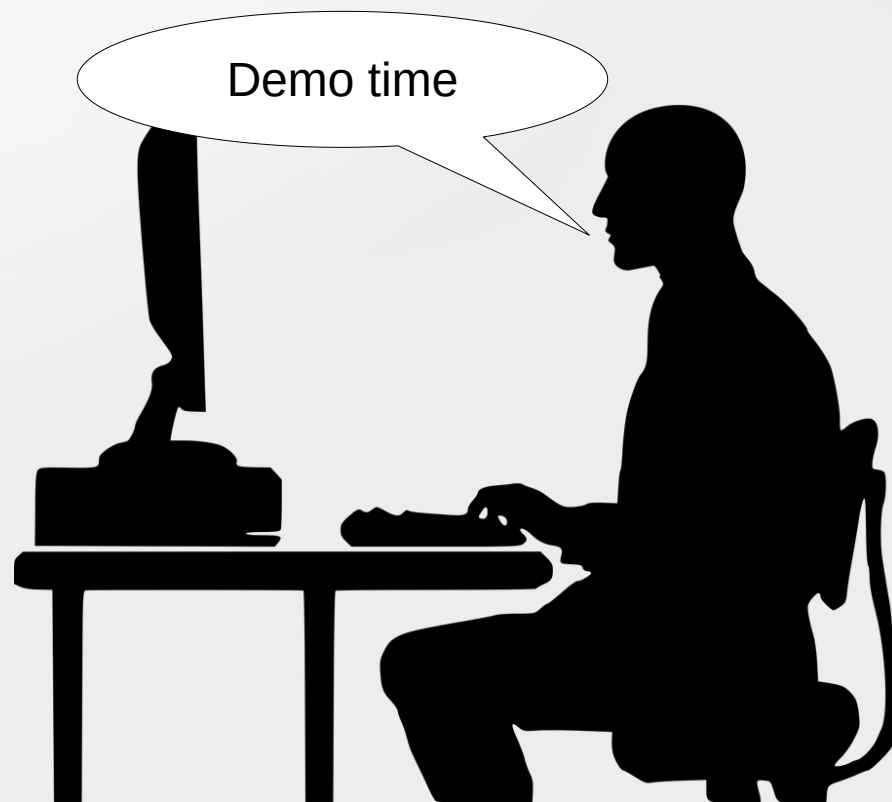
Widok:

- Ile firma zarobiła (kwota)

Struktura na dziś



Jak tego nie robić?



Co w tym kodzie jest najgorsze?



public



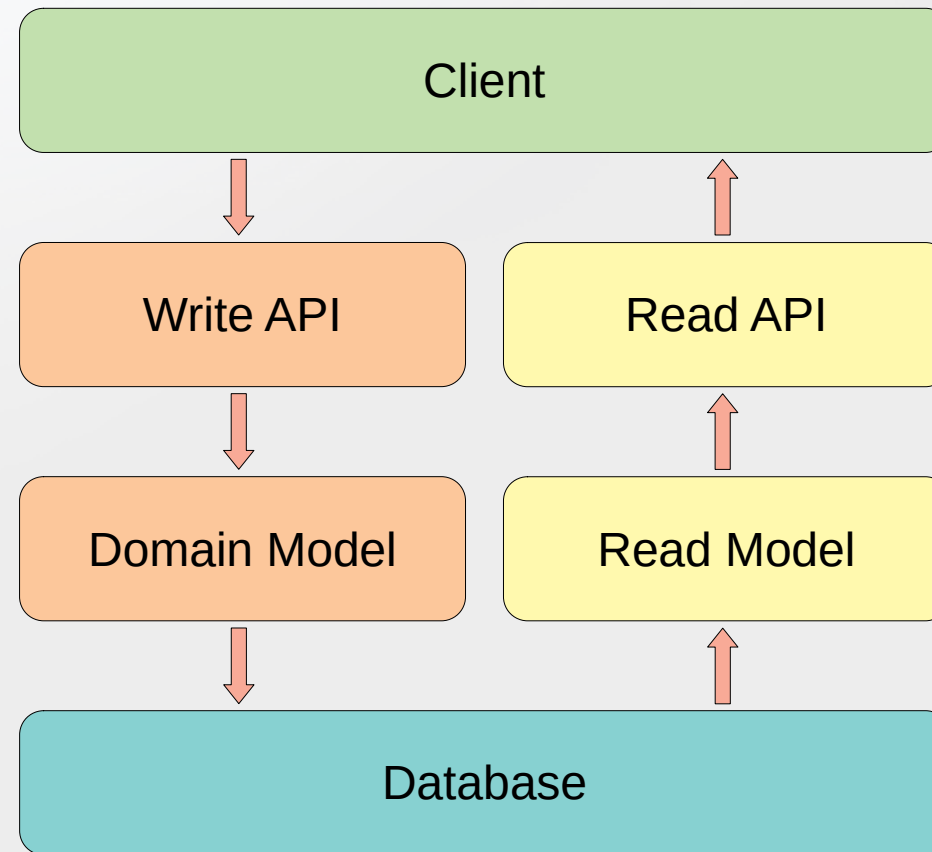
Anemiczny model domeny



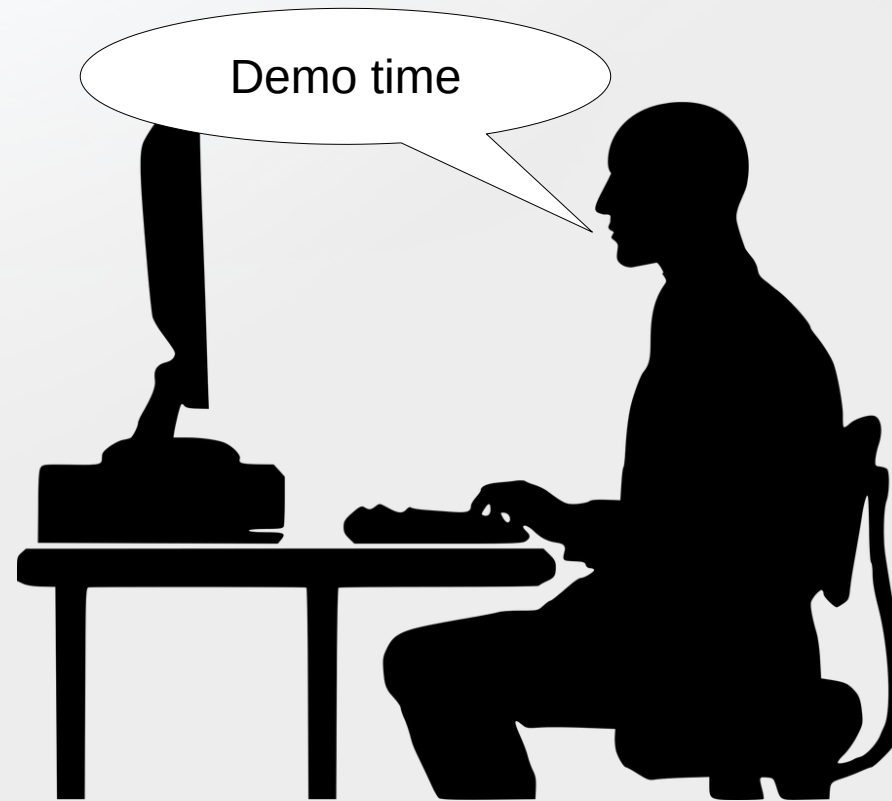
Jak to poprawić?



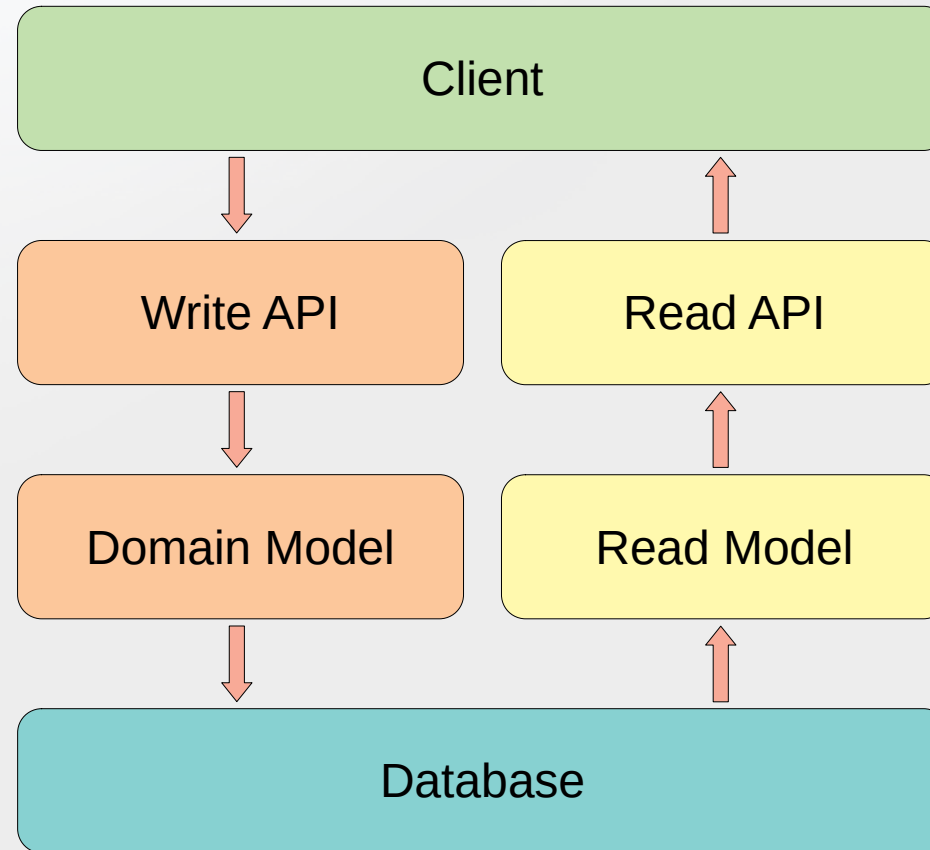
Czas na:



CQRS



Jest?



Co w tym kodzie jest jeszcze złe?

OOP?

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) {...}  
}
```

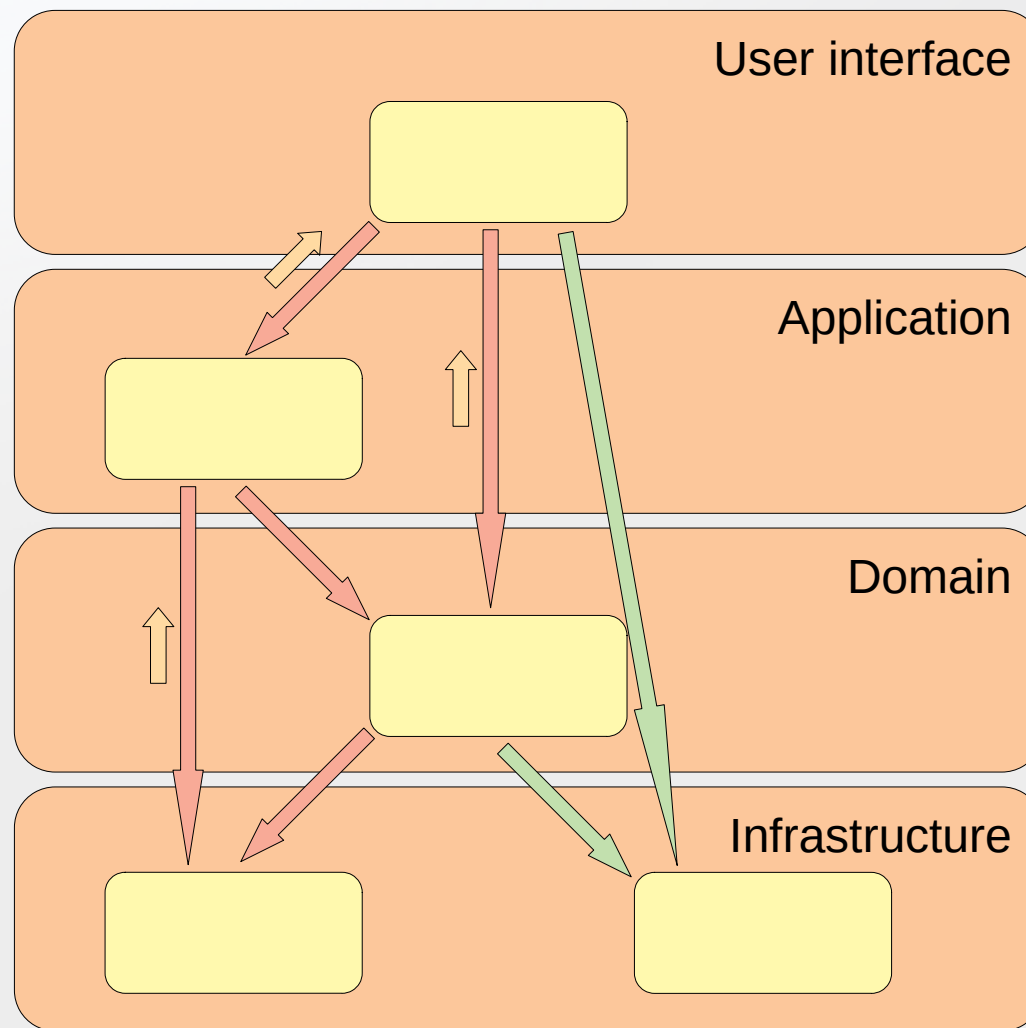
Jaka jest zaleta tego kodu?



Jedziemy dalej



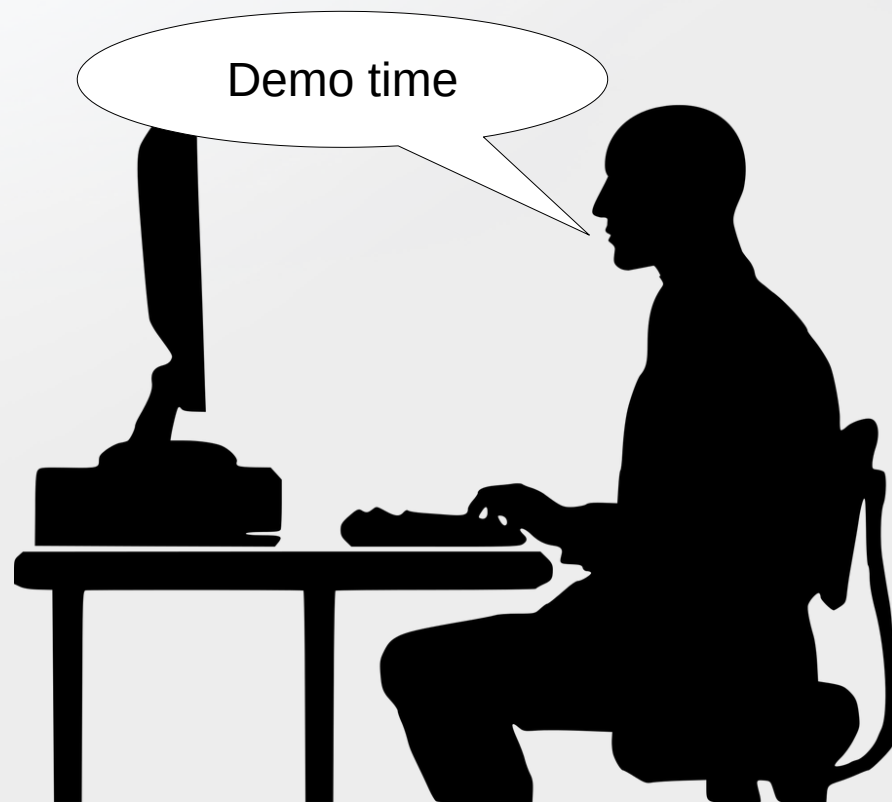
Jest?



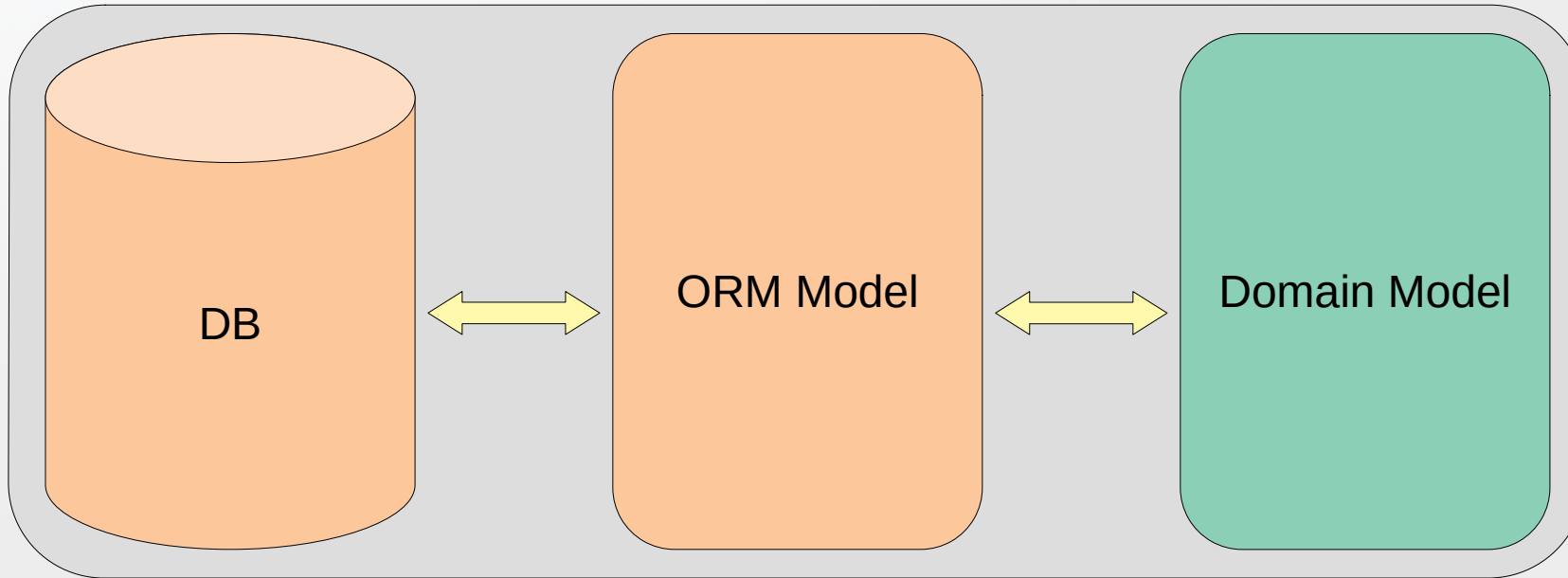
Q/A?



Jeszcze czyściej?



Kłopotnik



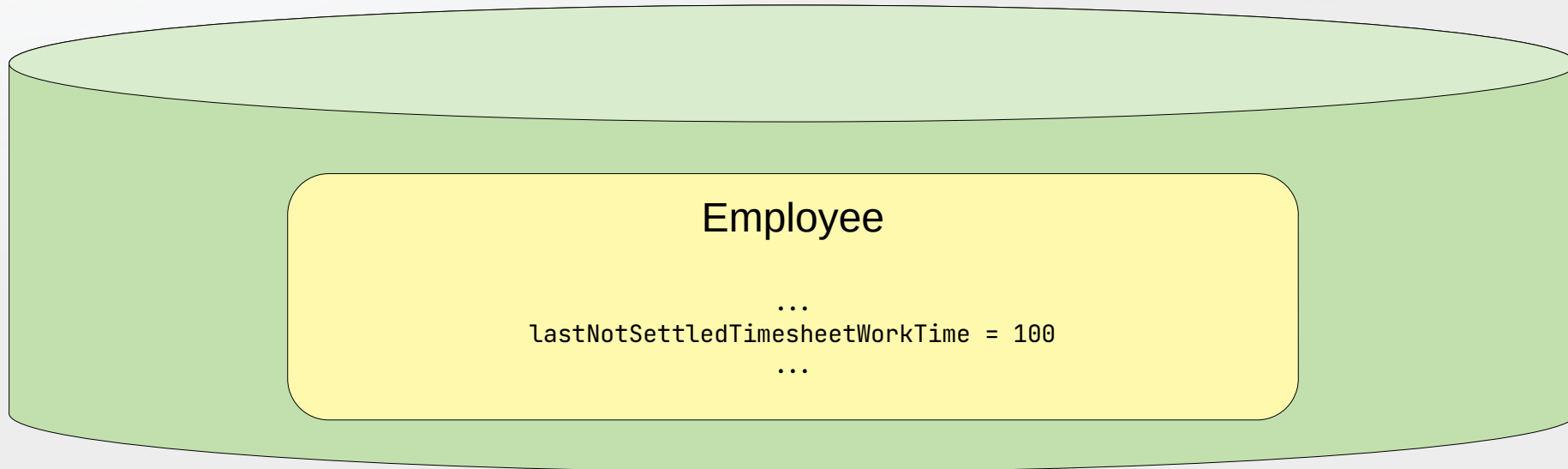
Połączyć ORM z Domain

Odtwarzać ORM na
podstawie Domain

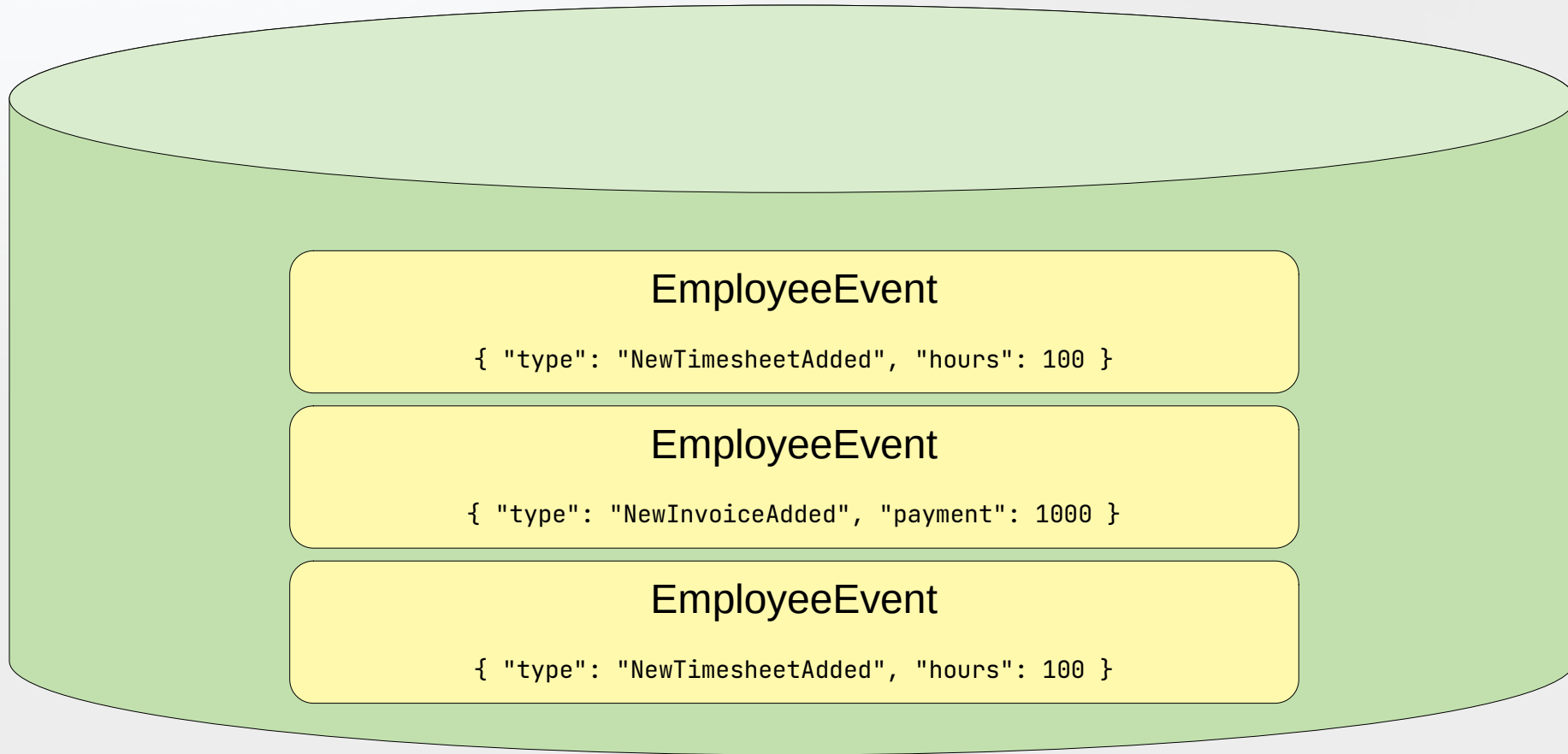
Nakładać zdarzenia z
Domain na ORM

Event sourcing

Event sourcing



Event sourcing



Event sourcing w życiu



Event sourcing



Geneza



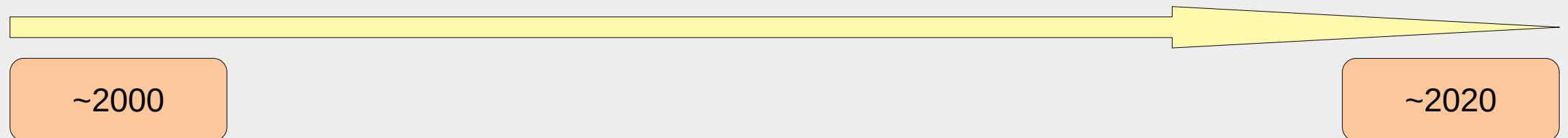
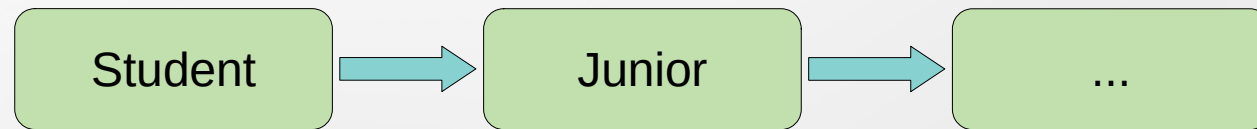
Geneza

1997 – JavaBeans Spec

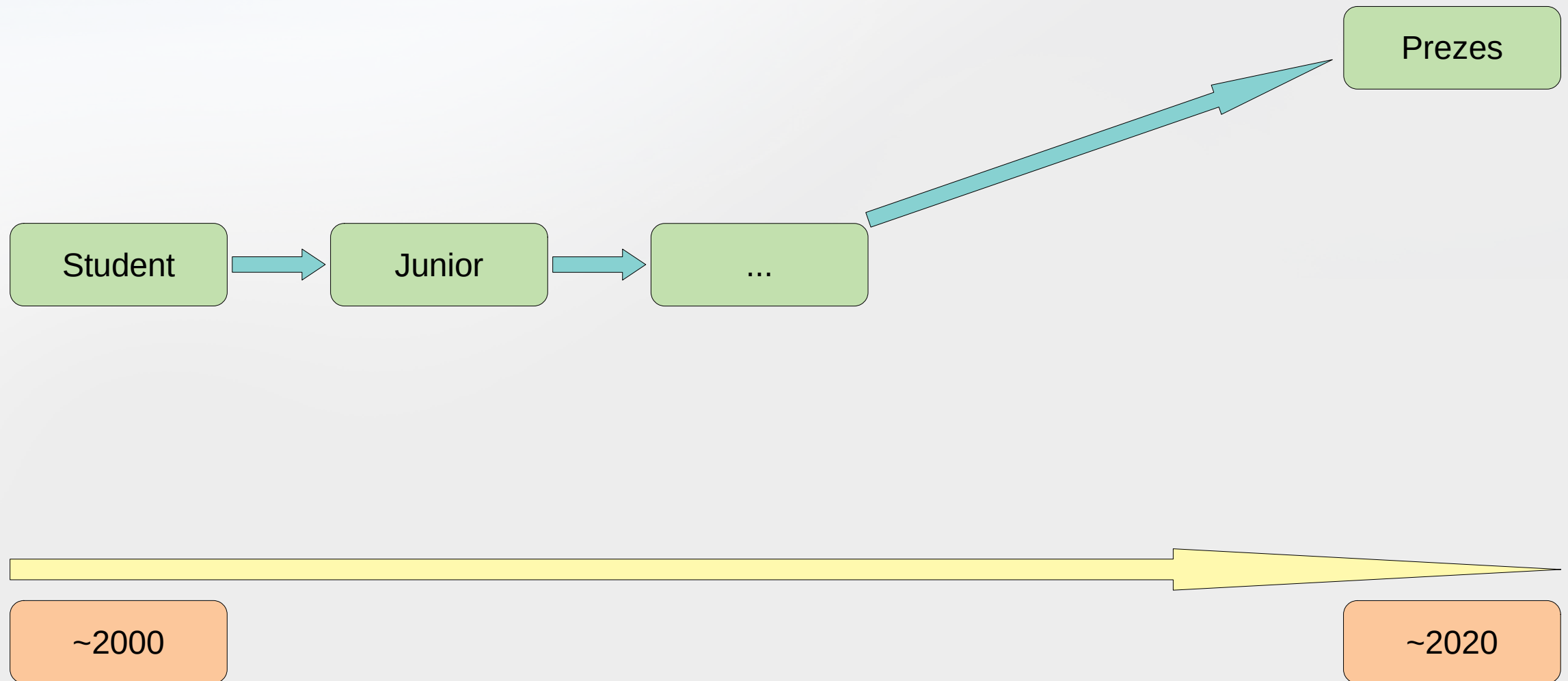
1997-200x – Frameworks

2009 – Lombok :)

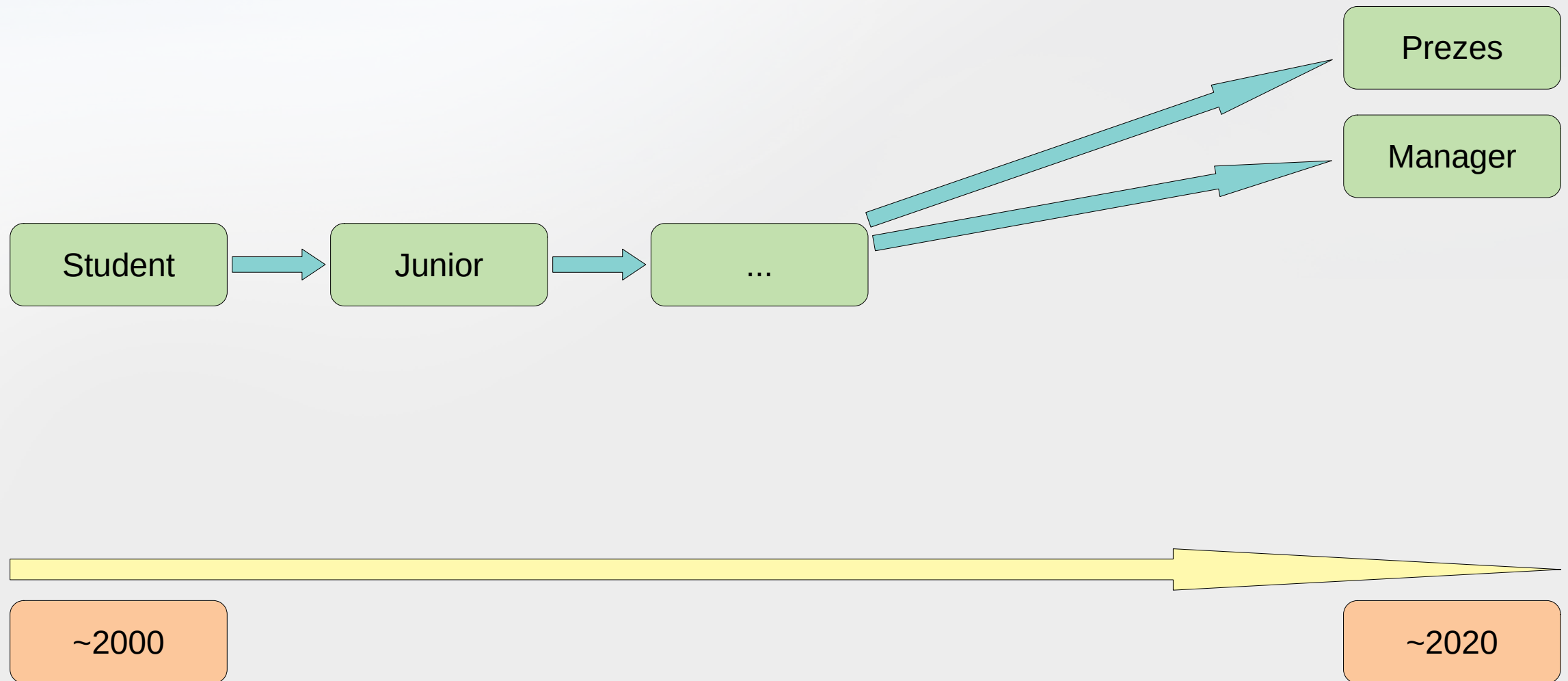
Geneza



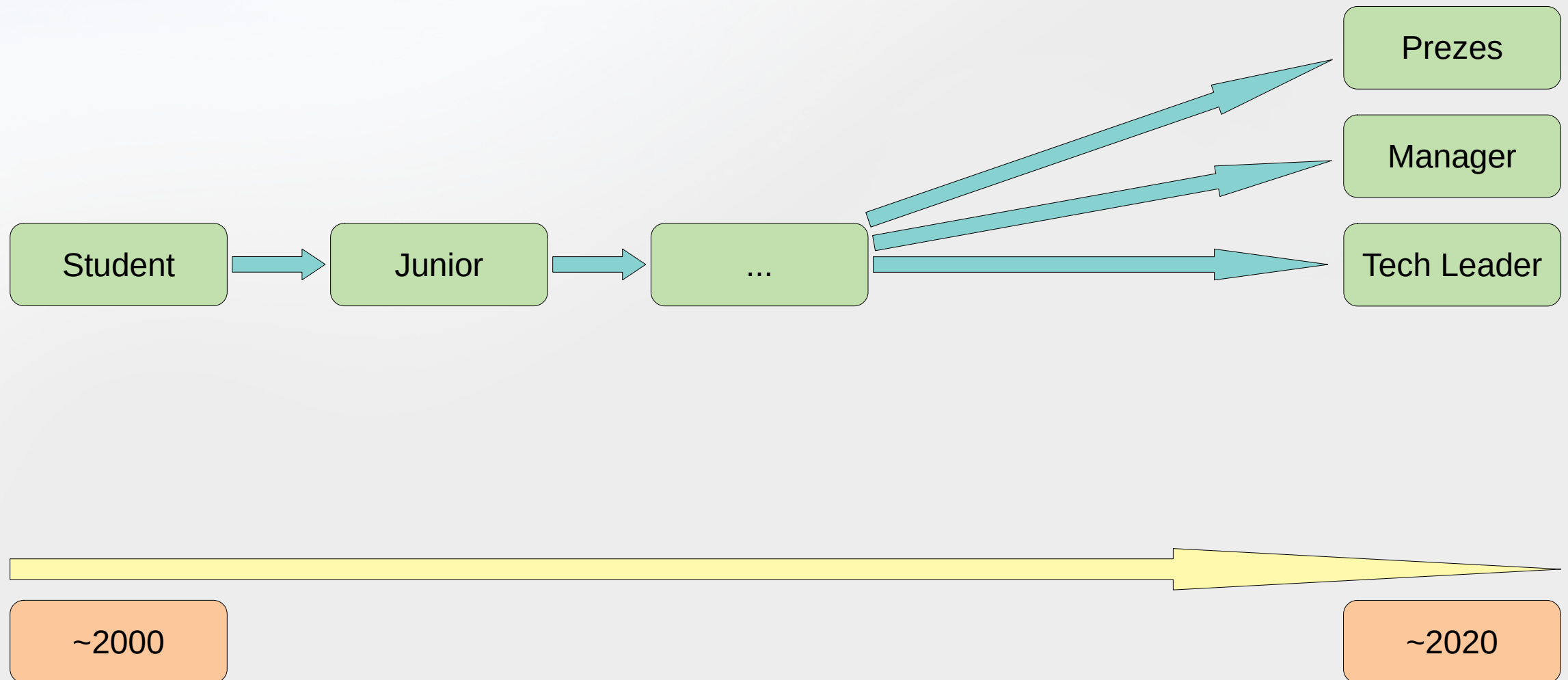
Geneza



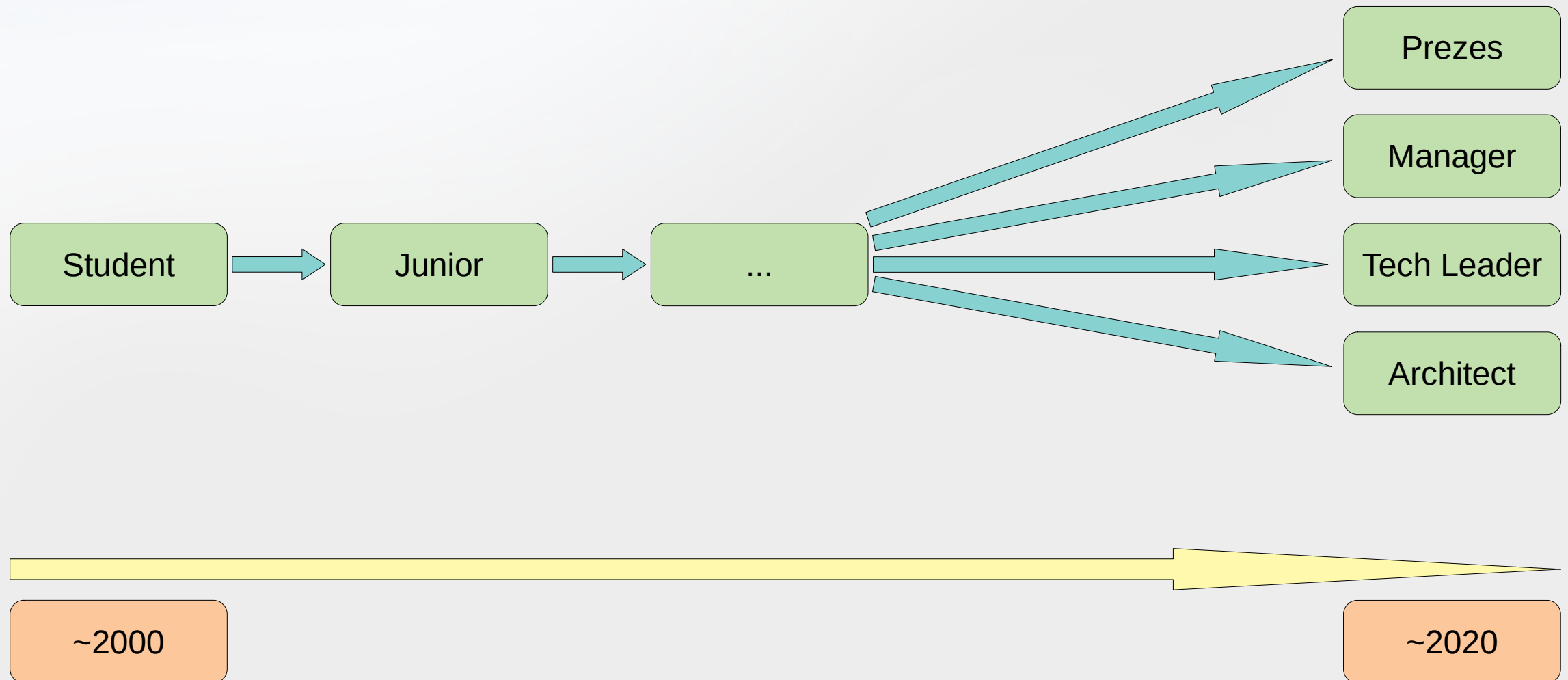
Geneza



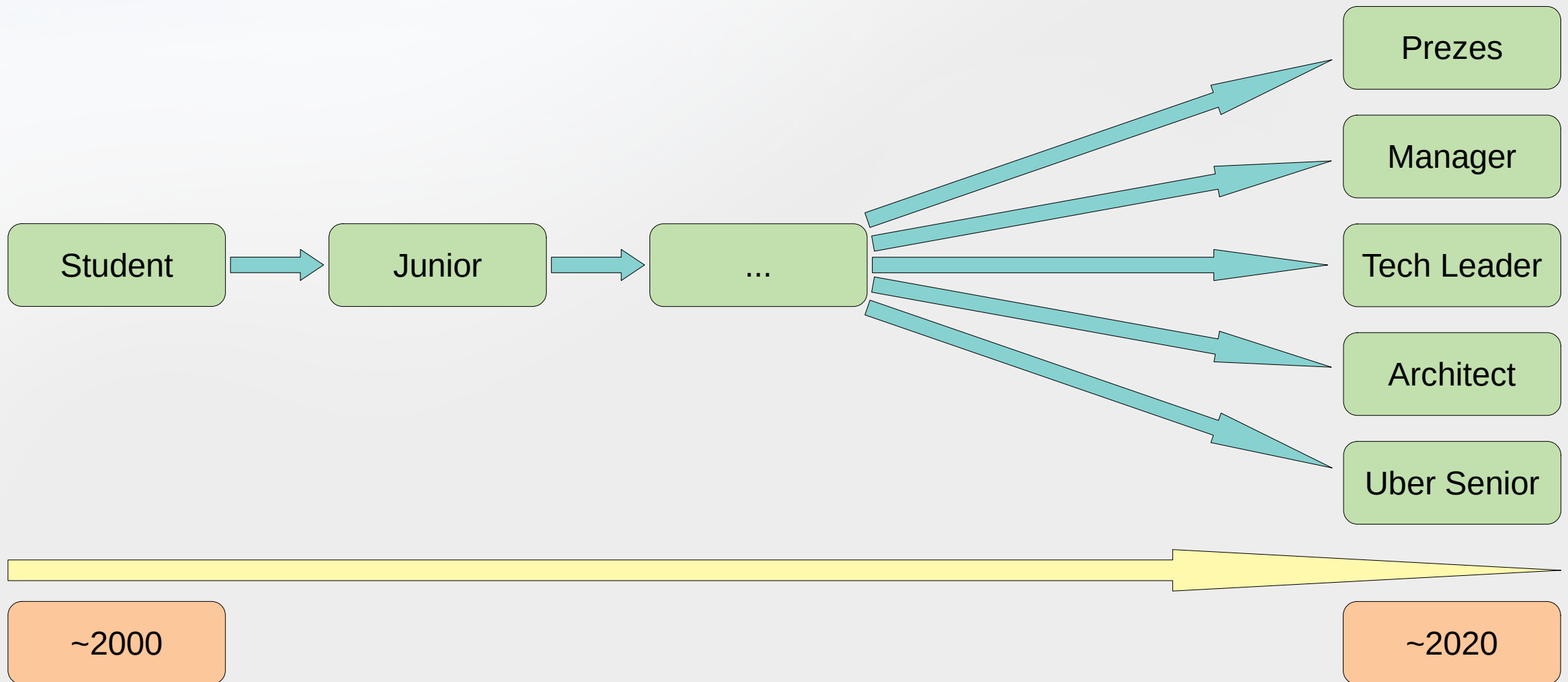
Geneza



Geneza



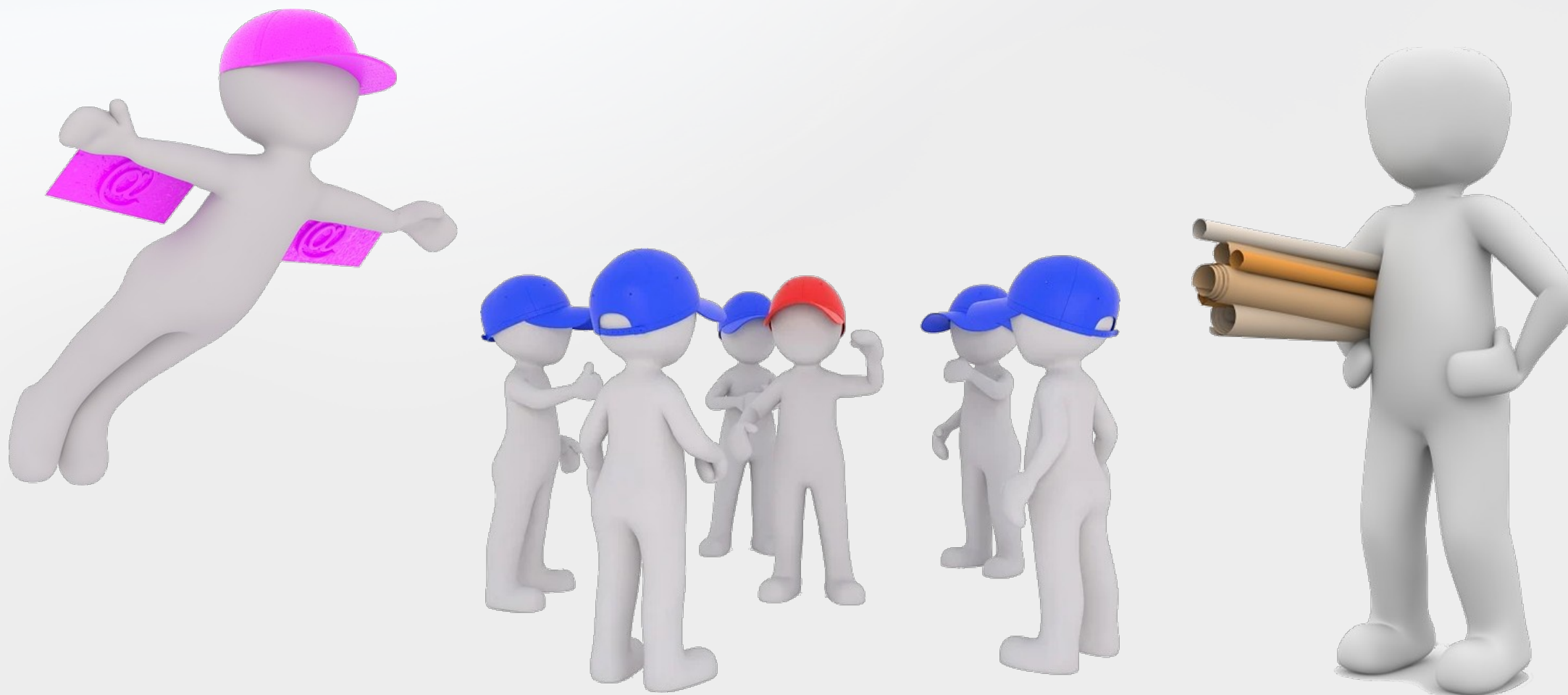
Geneza



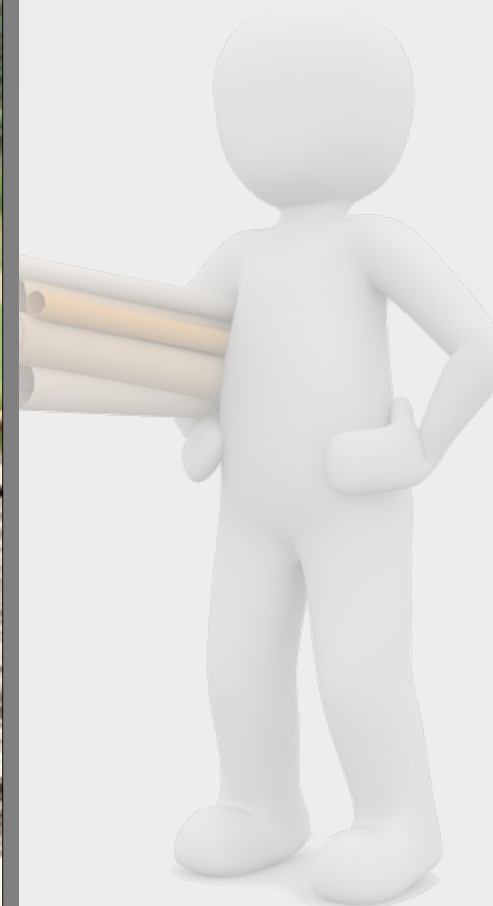
Geneza



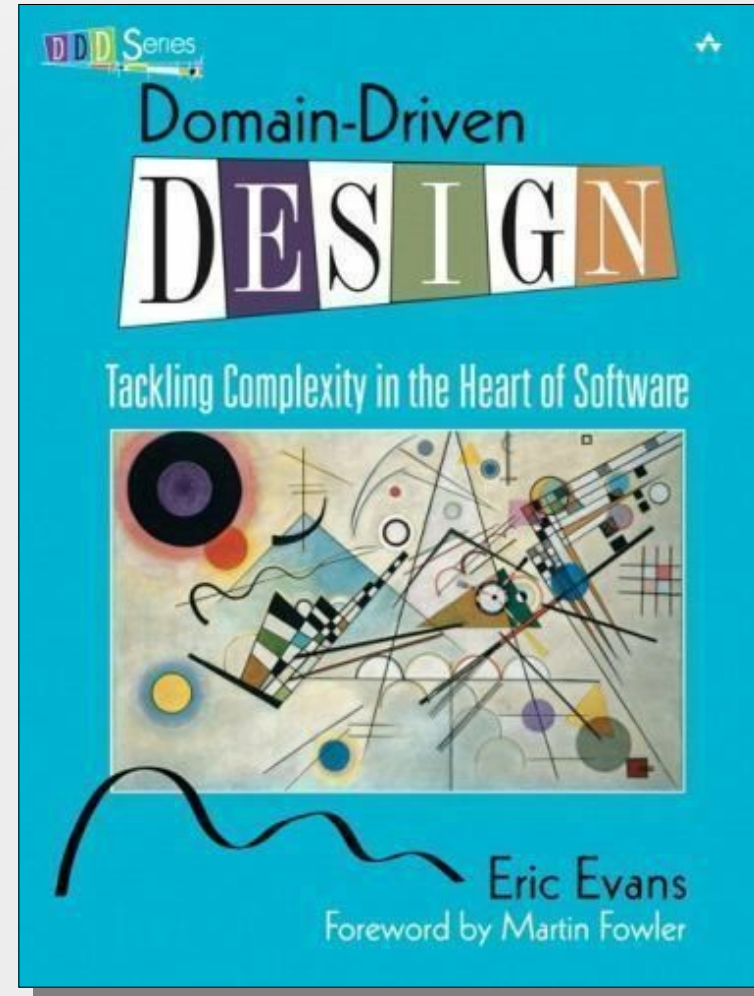
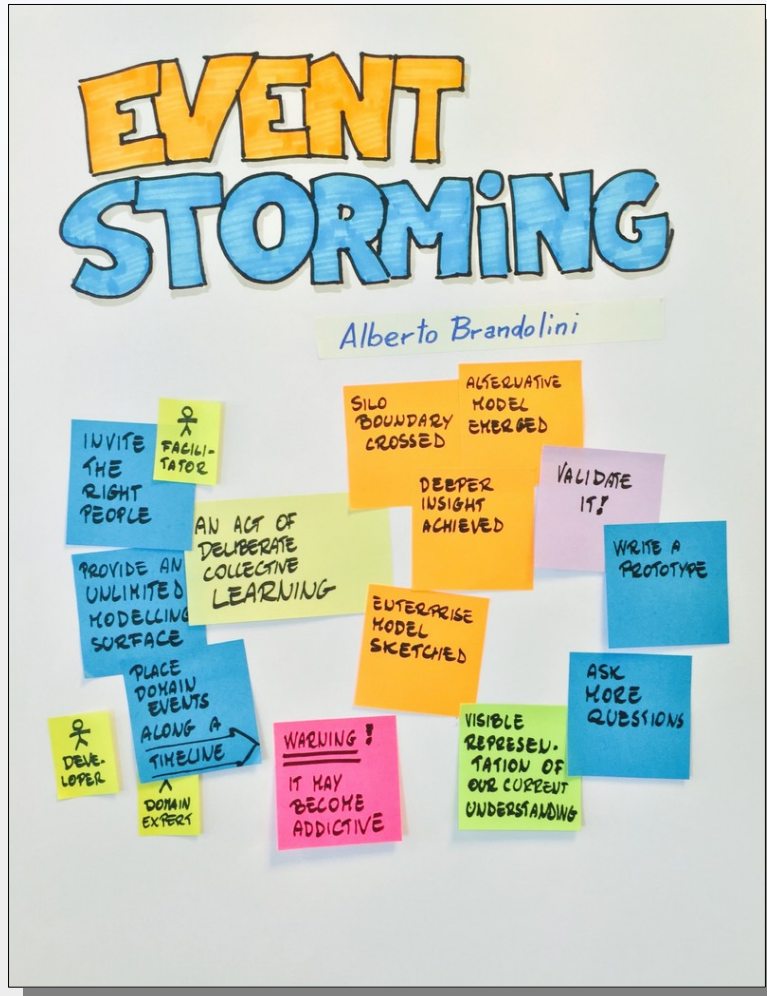
Przepis na sukces



Przepis na sukces



Hard part



Złote rady wujka Krzyśka



Najważniejsza zasada przy tworzeniu: YAGNI

Znajdź wady „technik” zanim ich użyjesz

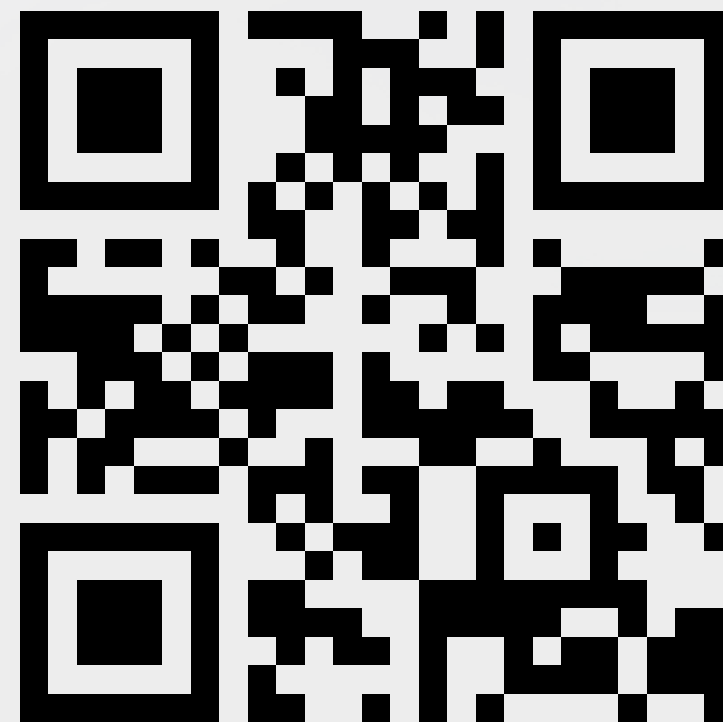
Enkapsulacja / hermetyzacja

Dobry moduł od złego odróżnia jego rozmiar

Testy – tak, ale tam gdzie są przydatne

CRUD – to jest dawno rozwiązany problem

Feedback please



<http://kielce.kś.pl/>

<http://kś.pl>
<http://gclogs.com>

conf@kś.pl
ks@gclogs.com

Dziękuję



<http://kielce.kś.pl/>