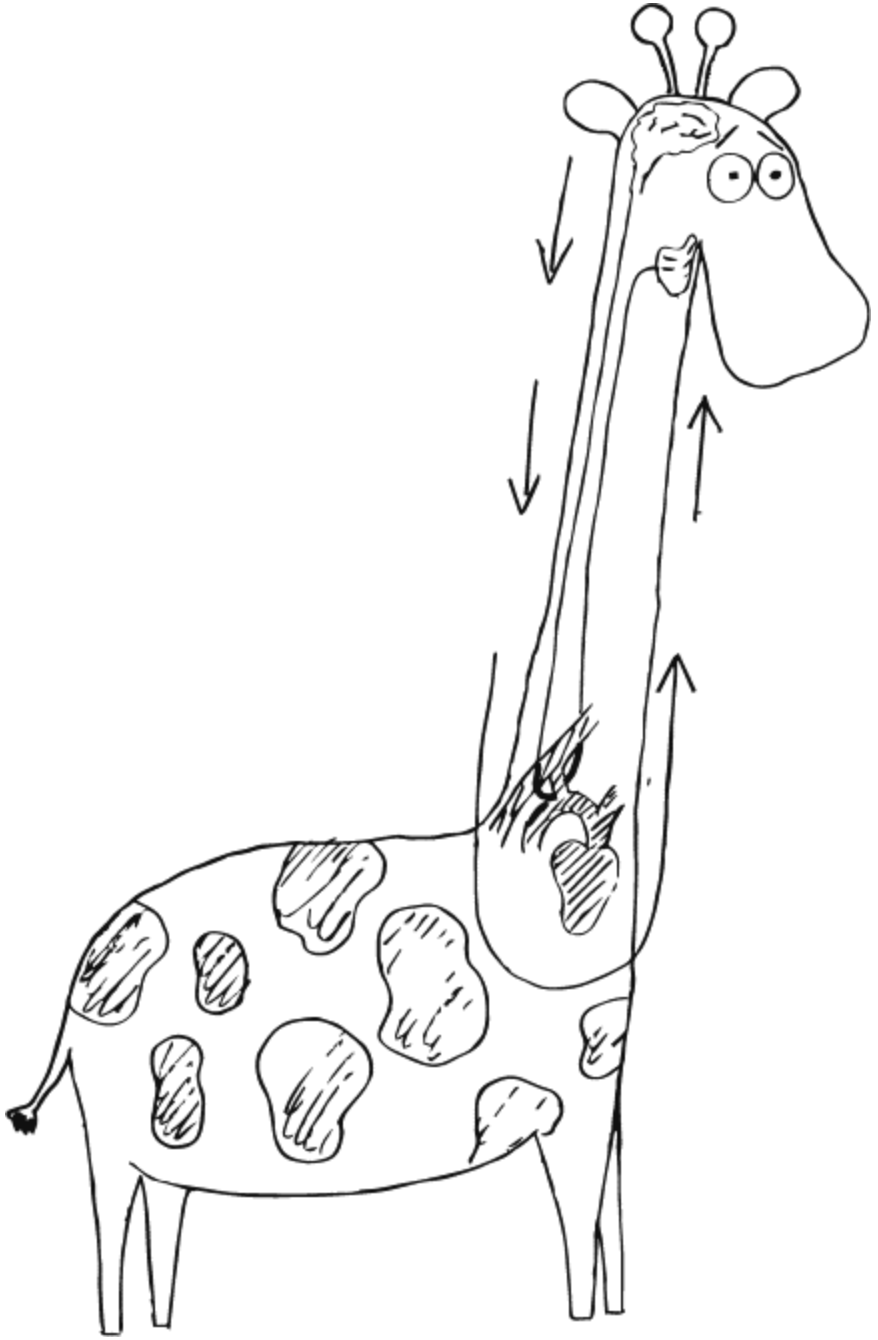
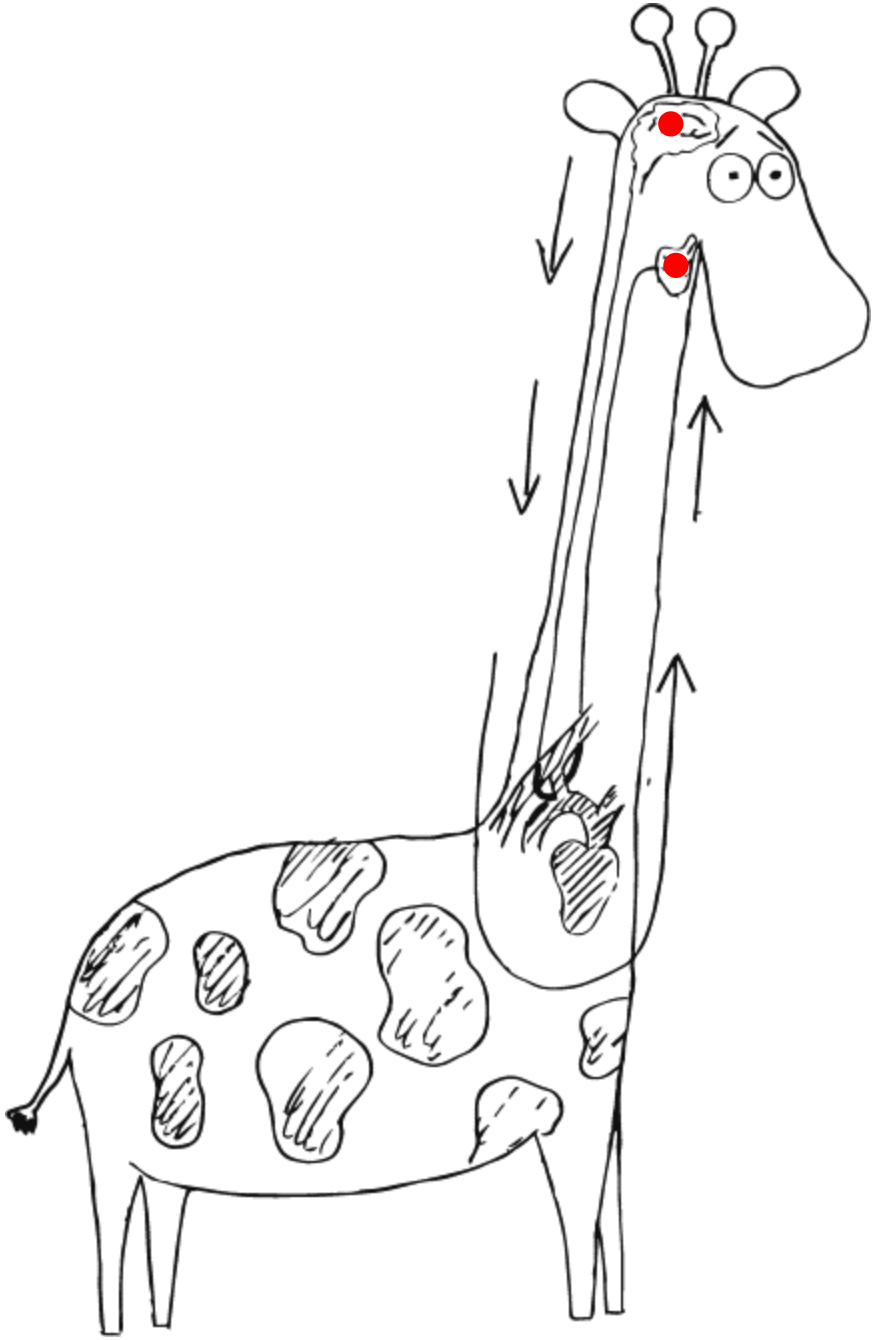
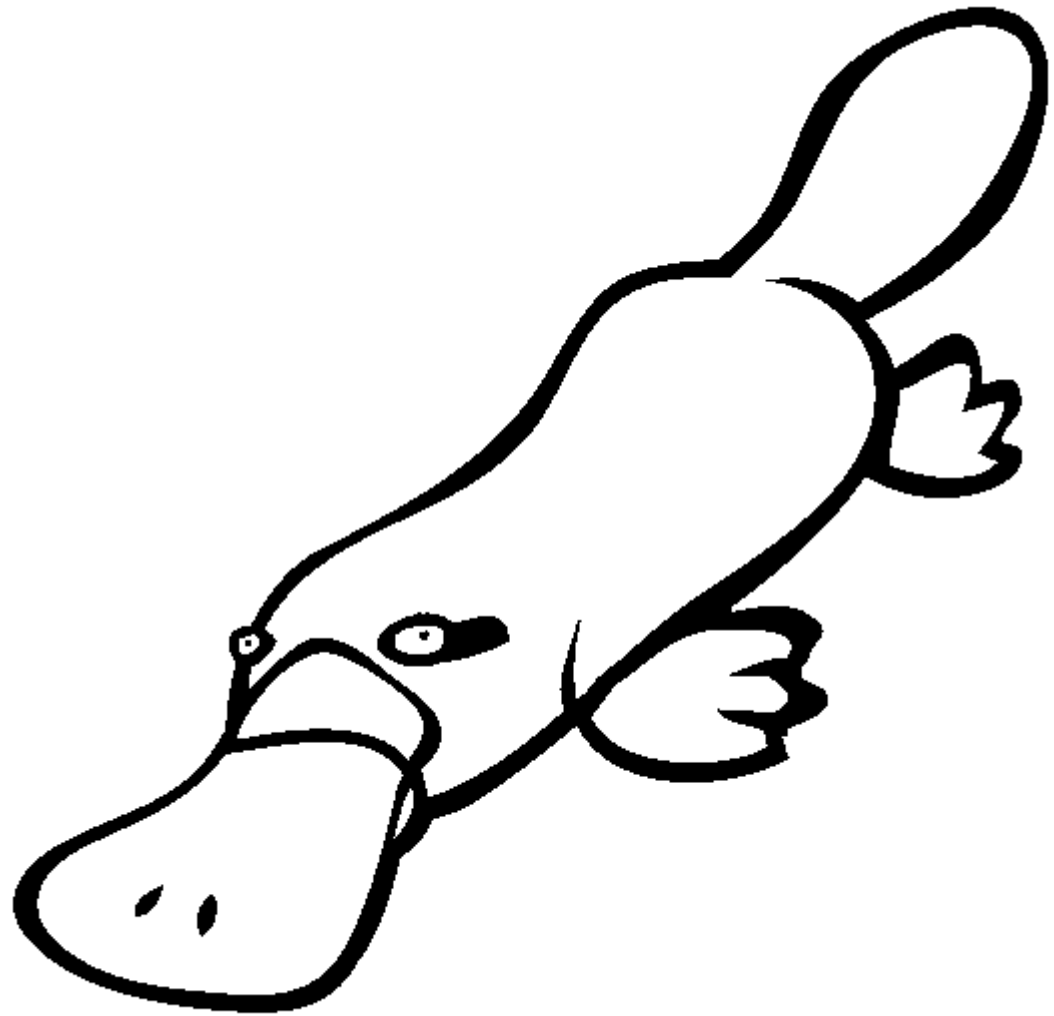
Domain Driven Design: рецепт для прагматиков

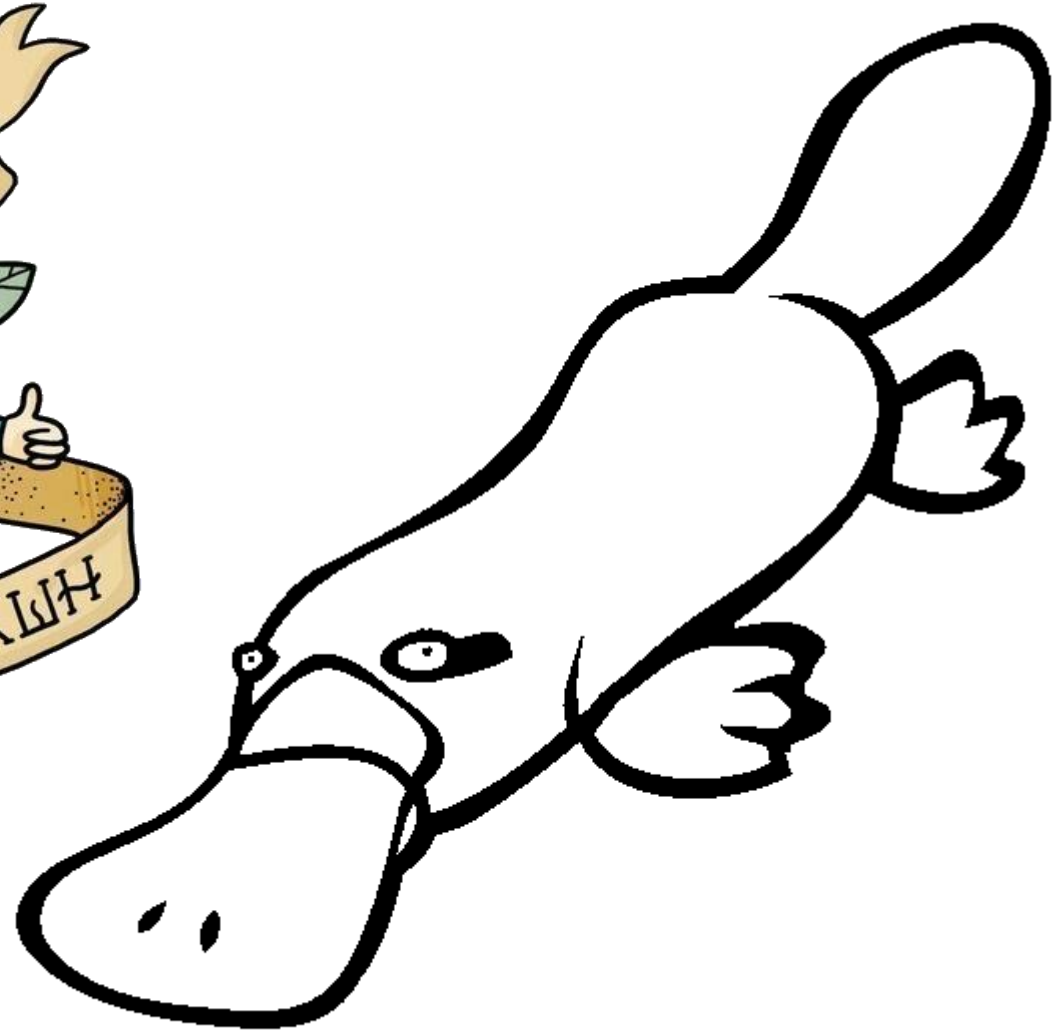
Алексей Мерсон
alexey.merson@gmail.com



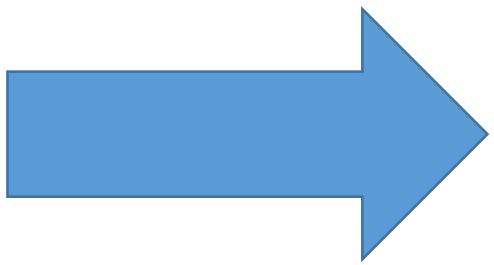


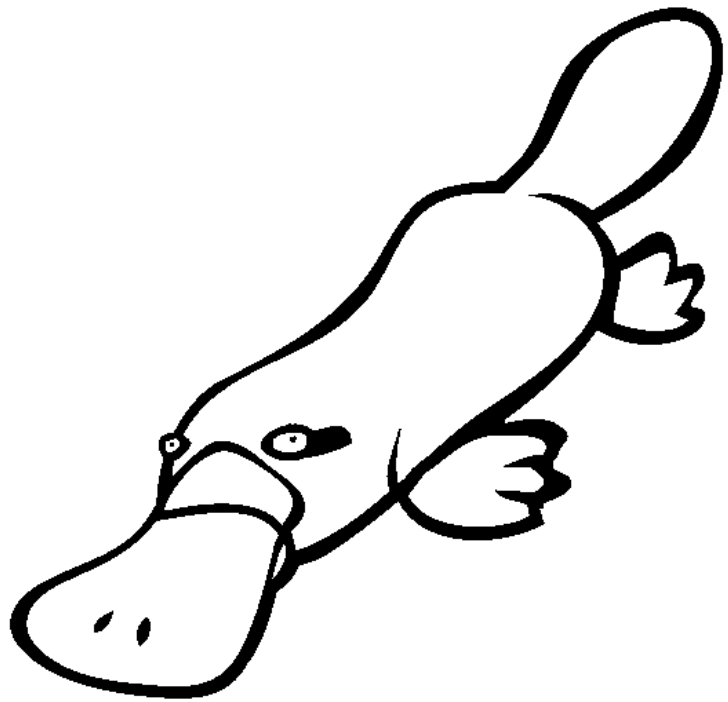



ХУЯ УЯК
И В ПРОДАКШН

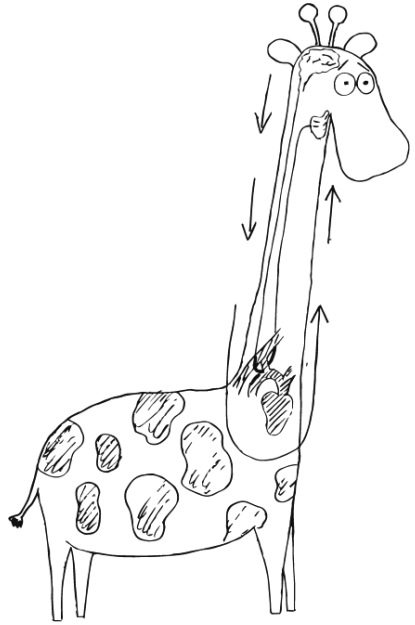


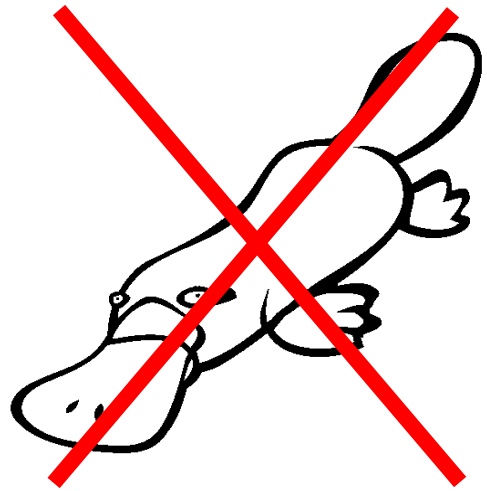

Но при чем здесь, черт побери,
Domain Driven Design?



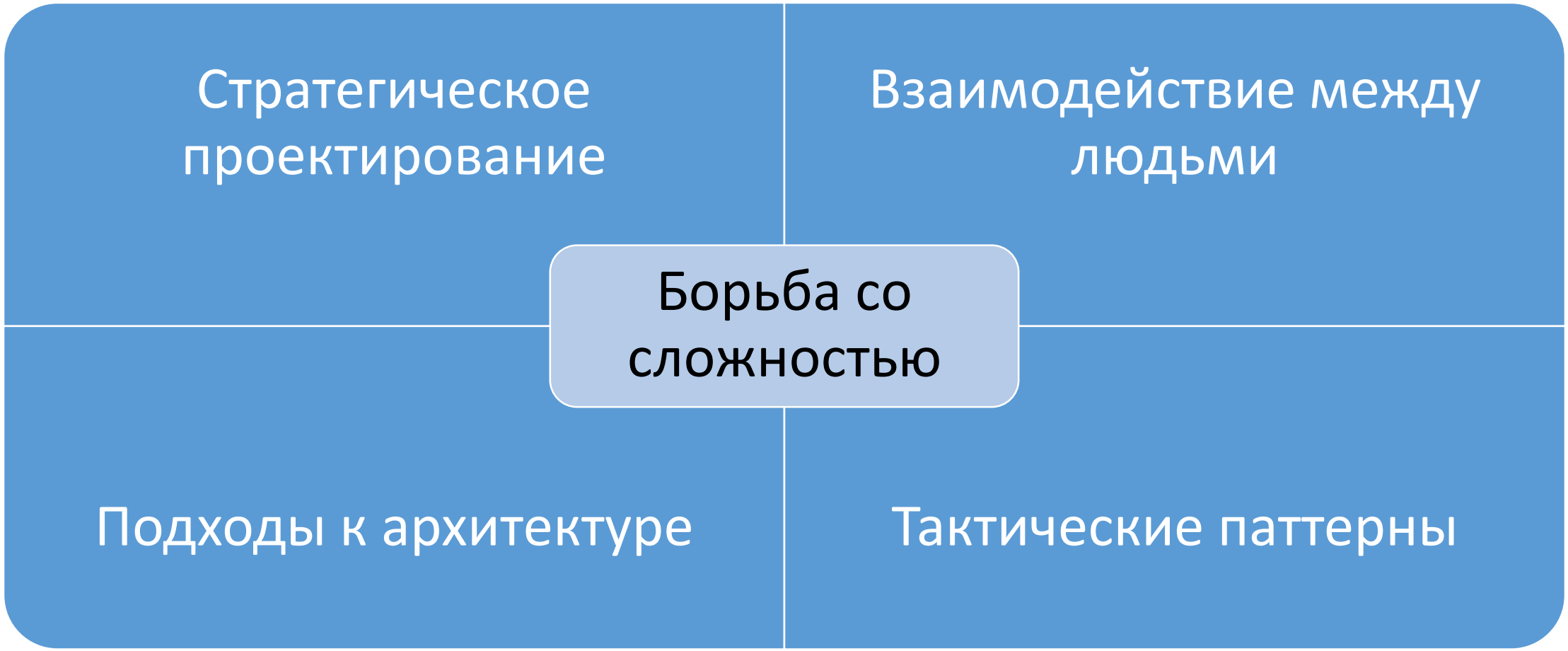


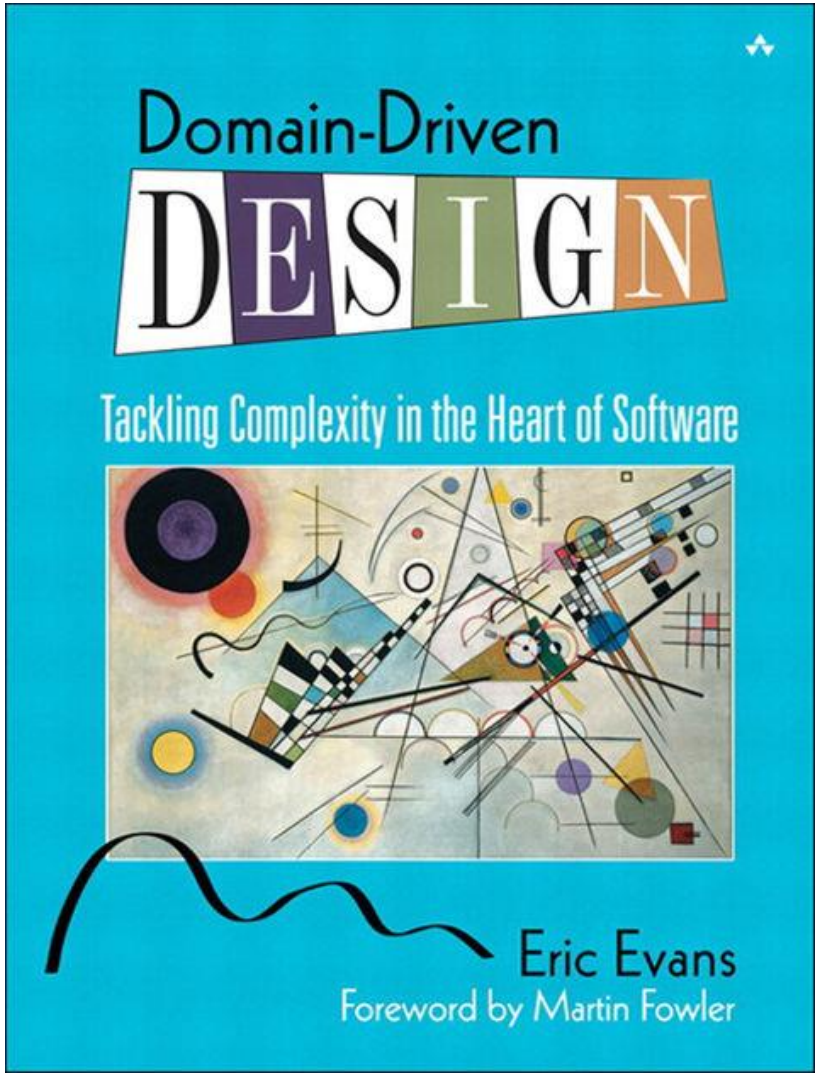
DDD

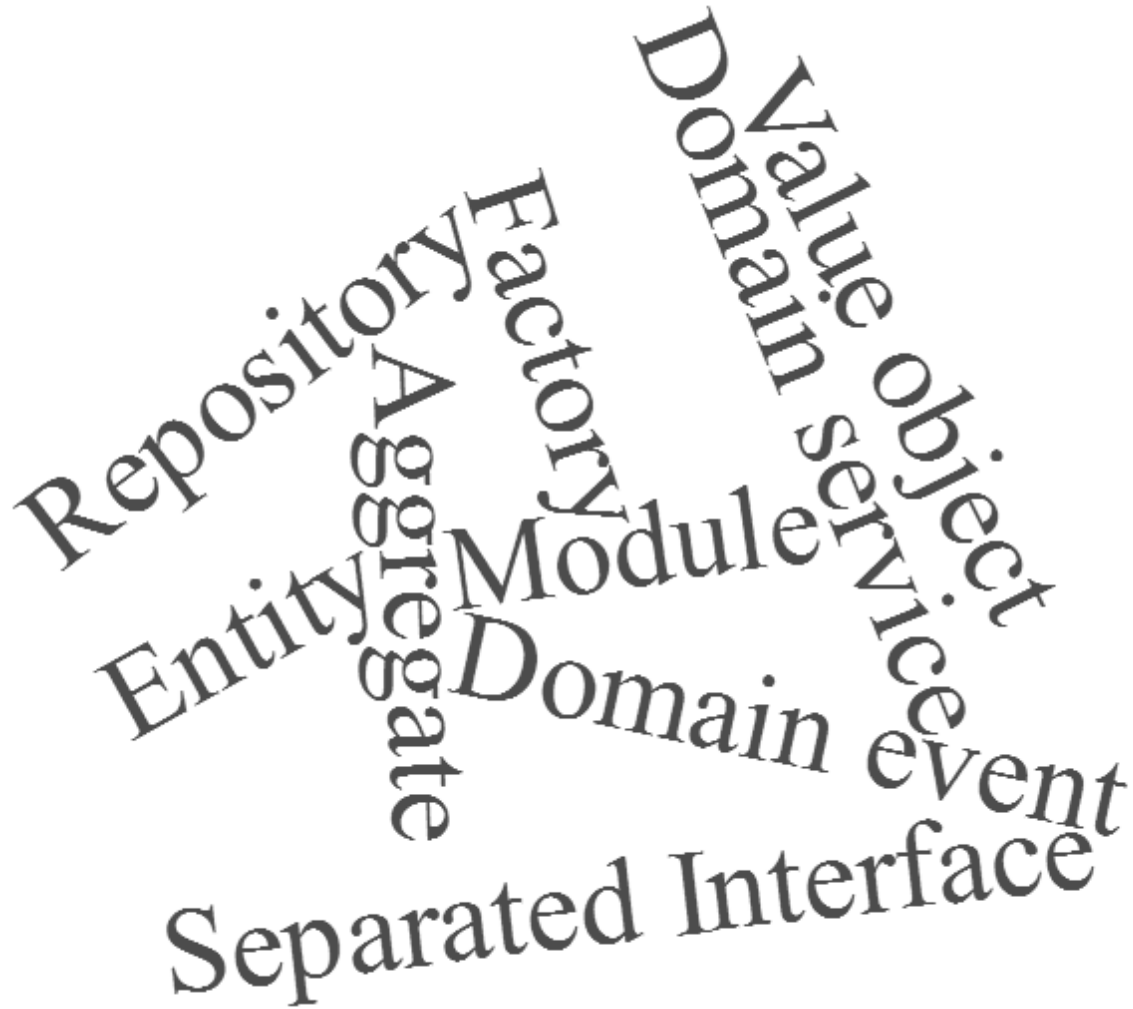




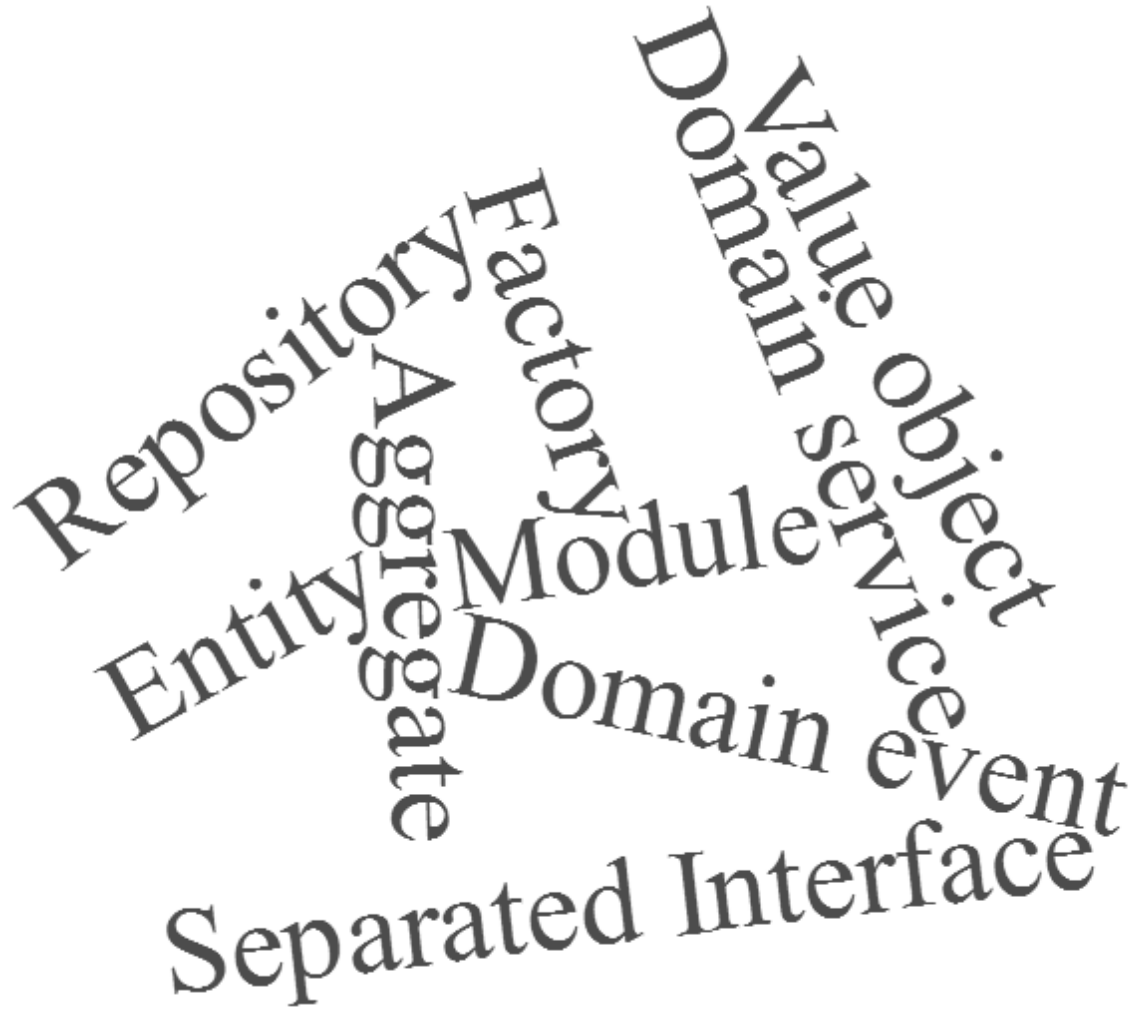
Domain Driven Design:

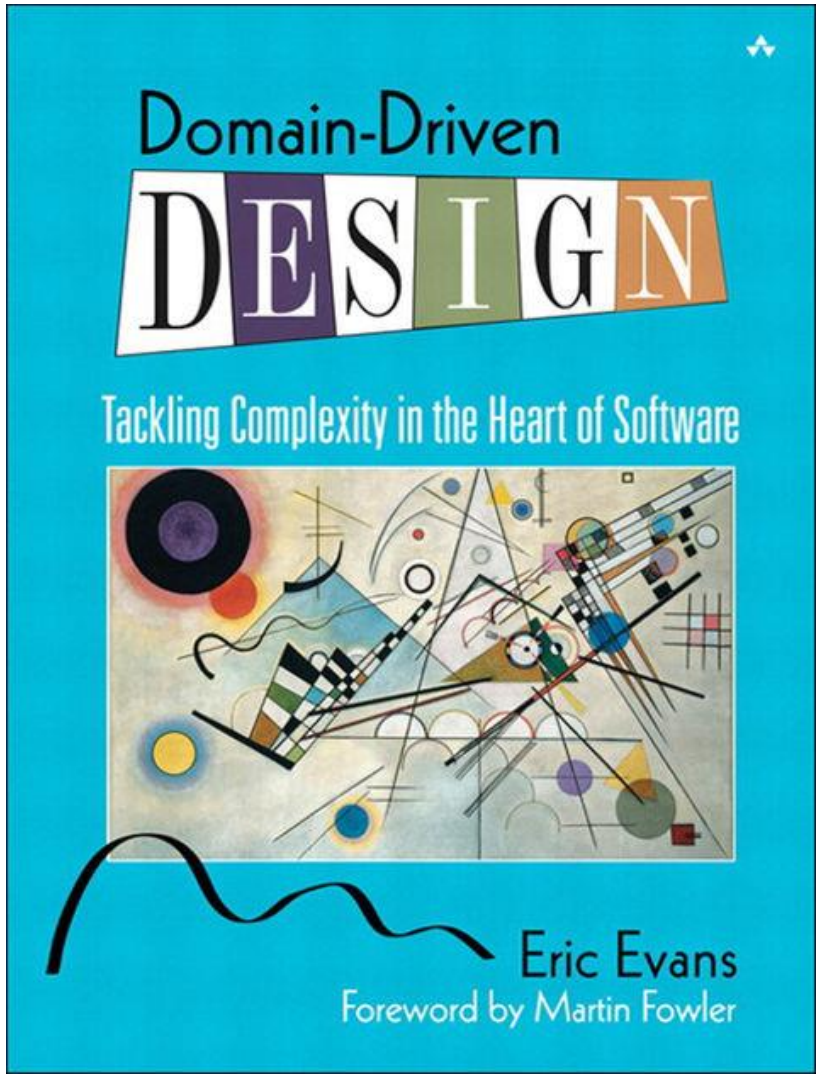


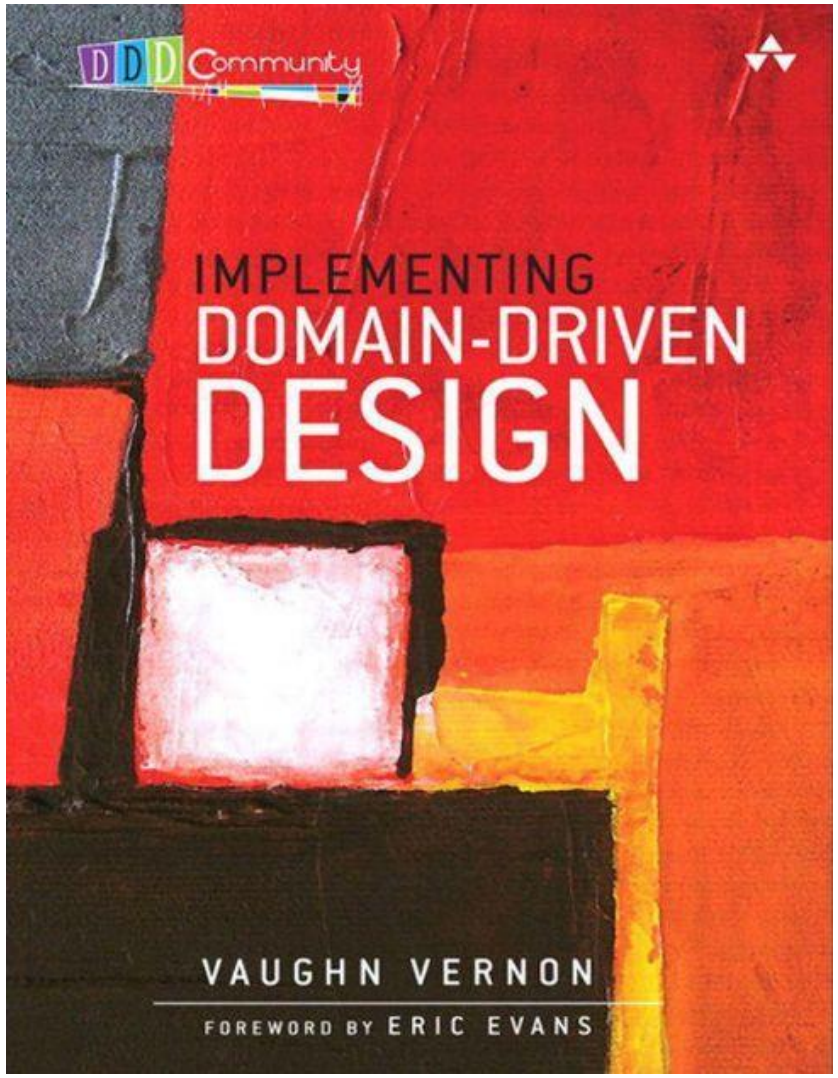

Domain-Driven
DESIGN
Tackling Complexity in the Heart of Software
Eric Evans
Foreword by Martin Fowler


Repository
Factory
Value object
Domain service
Entity
Aggregate
Module
Domain event
Separated Interface

Context Map
Bounded Context
Anticorruption Layer
Ubiquitous Language
Domain model
Subdomains

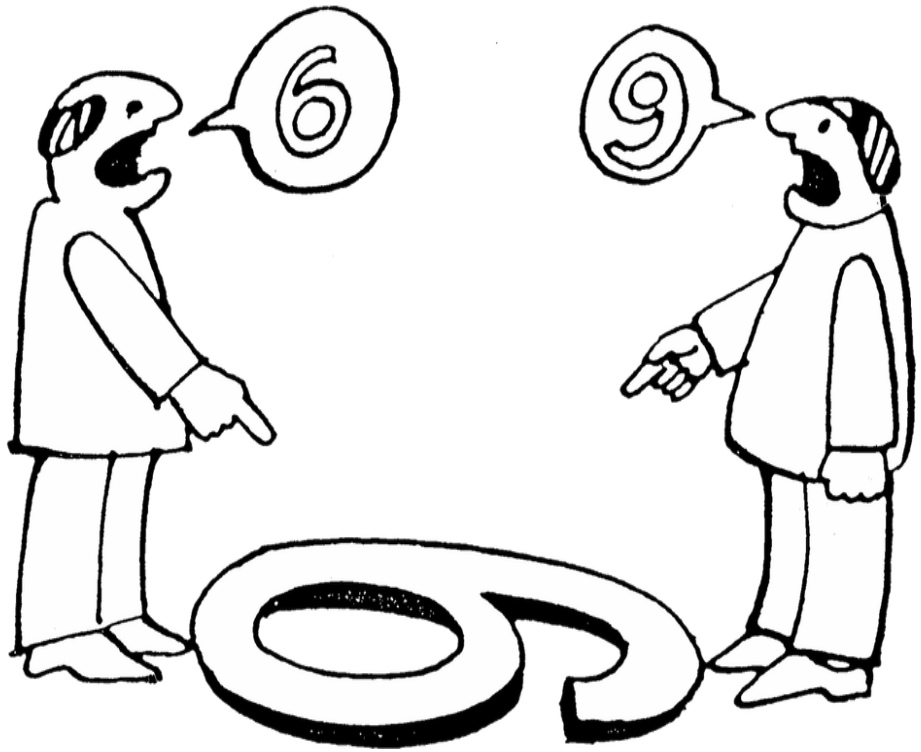



Domain-Driven
DESIGN
Tackling Complexity in the Heart of Software
Eric Evans
Foreword by Martin Fowler

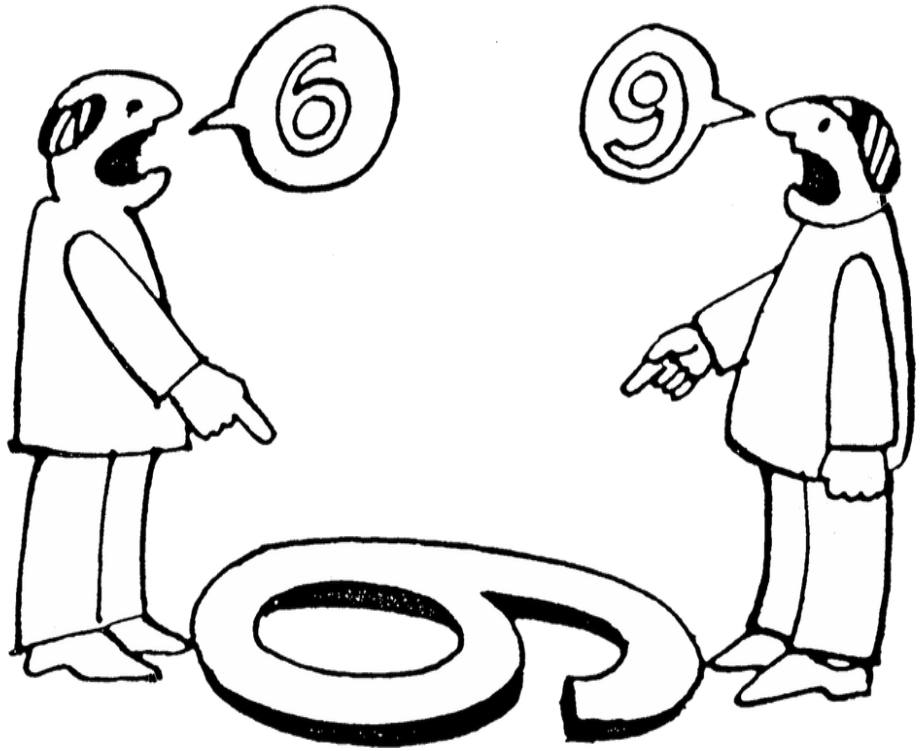

DDD Community
IMPLEMENTING
DOMAIN-DRIVEN
DESIGN
VAUGHN VERNON
FOREWORD BY ERIC EVANS

Ключевые идеи стратегического проектирования

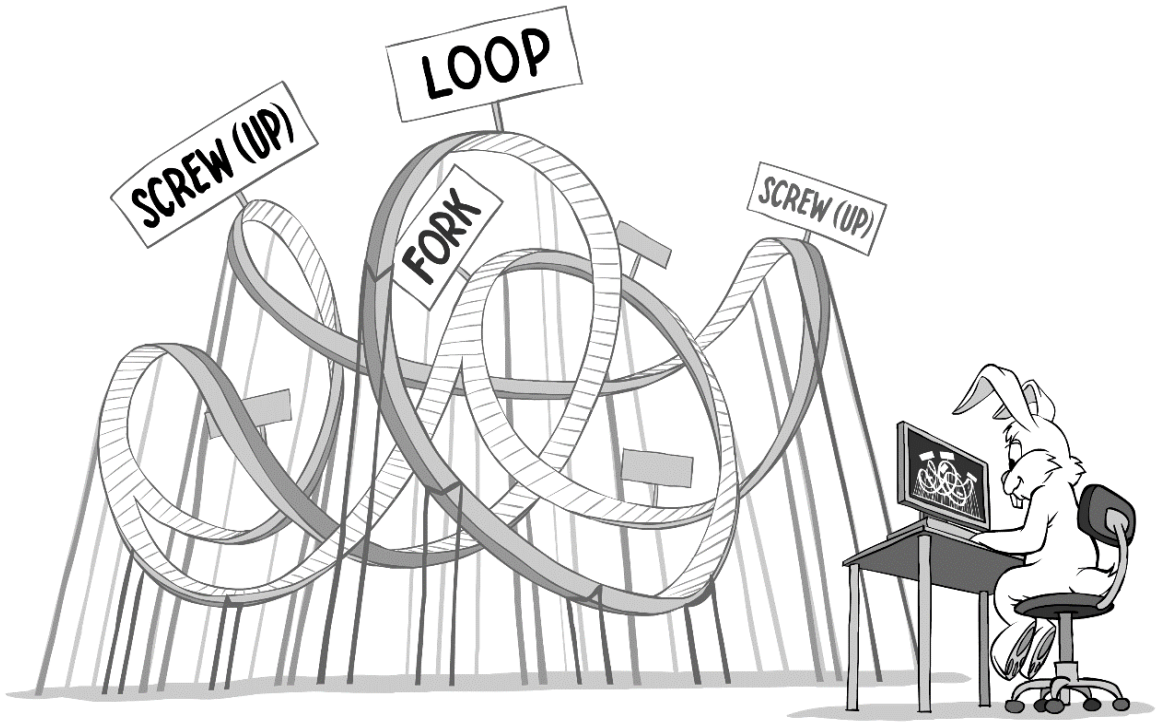
Доменные эксперты



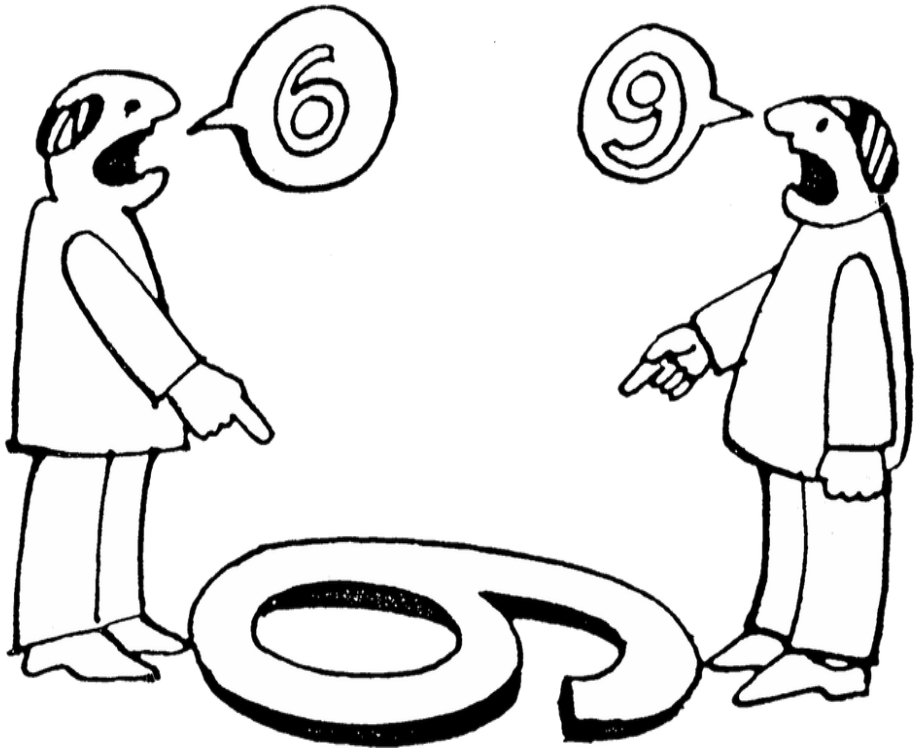
Доменные эксперты

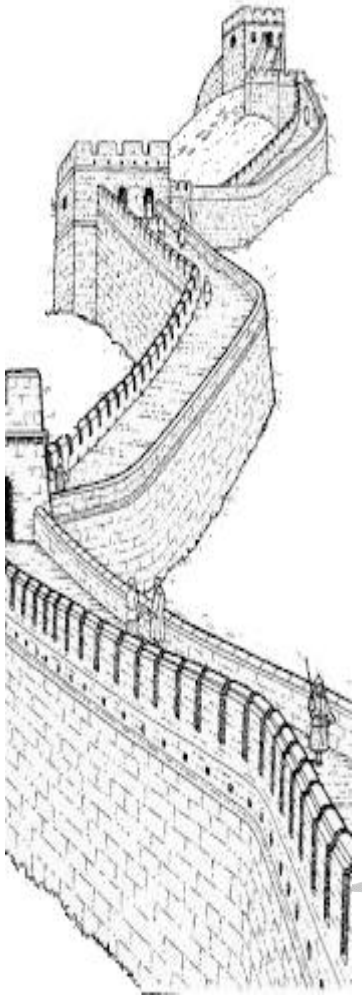


Технические специалисты

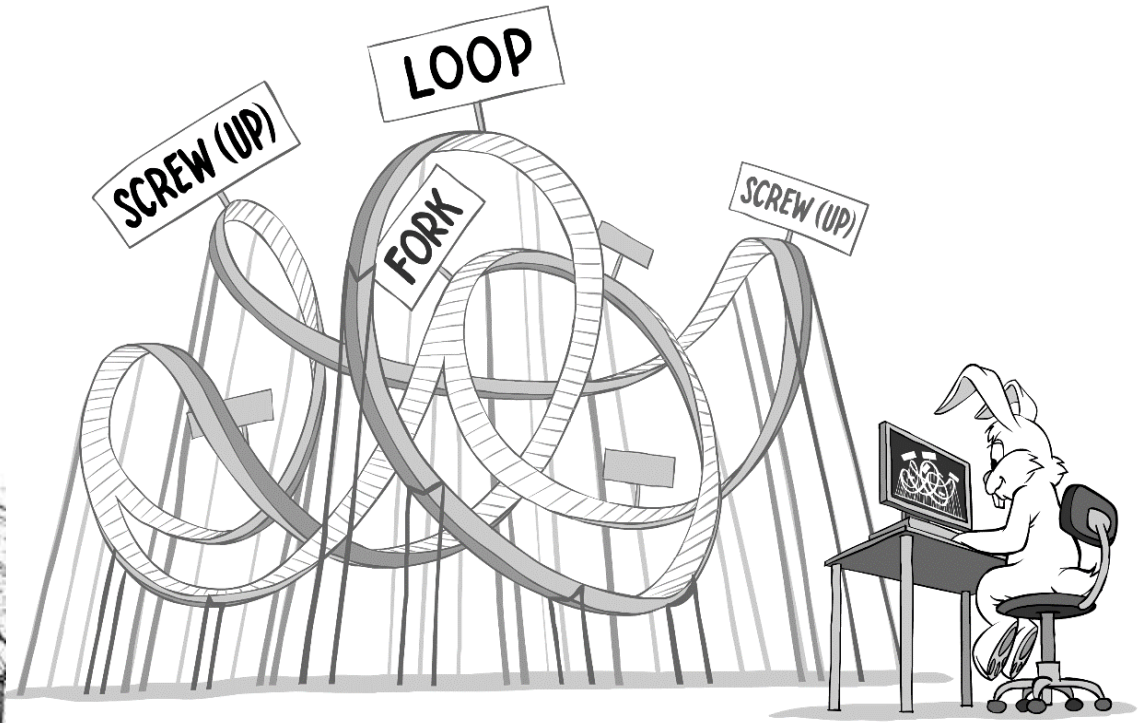


Доменные эксперты



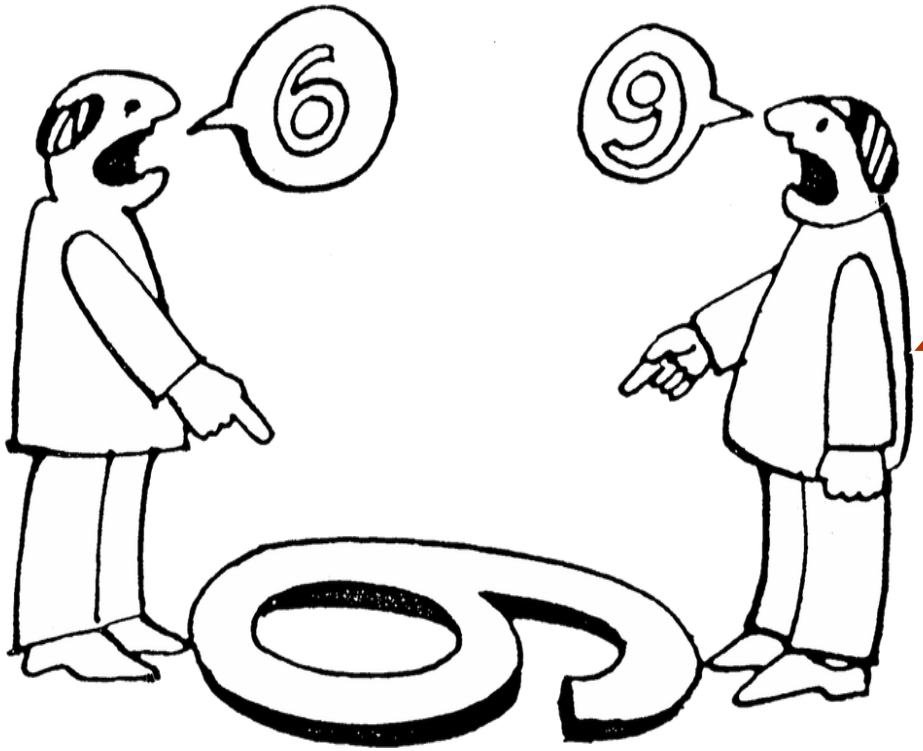


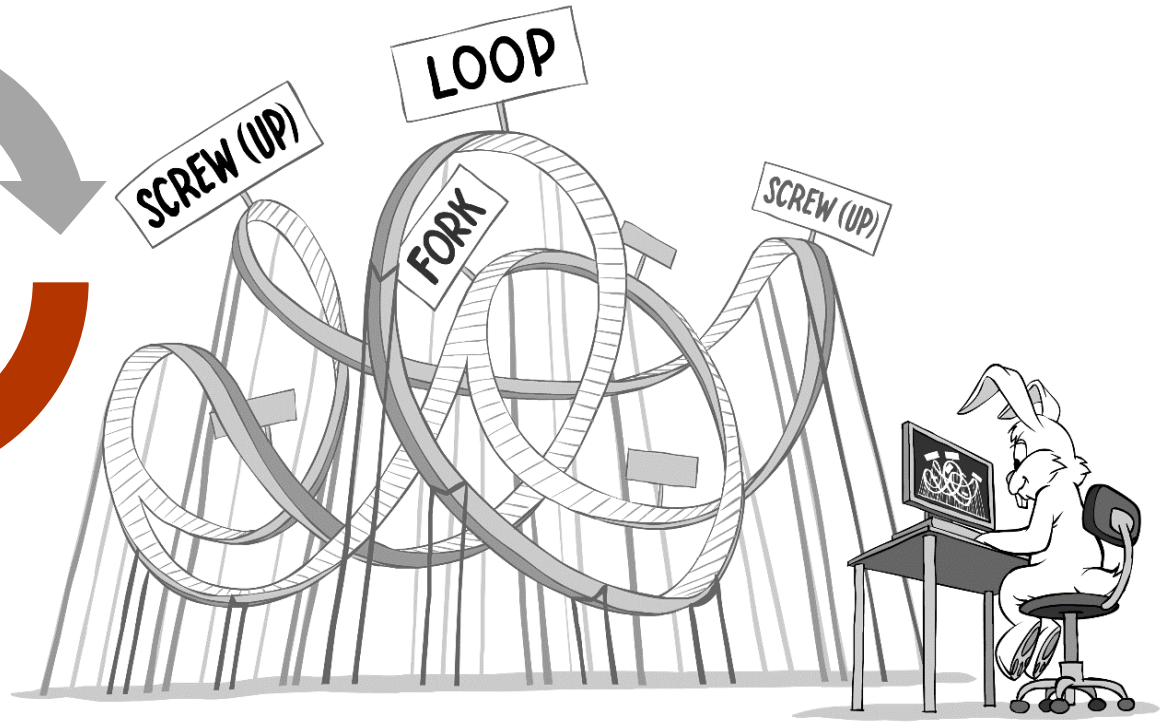
Технические специалисты



Доменные эксперты

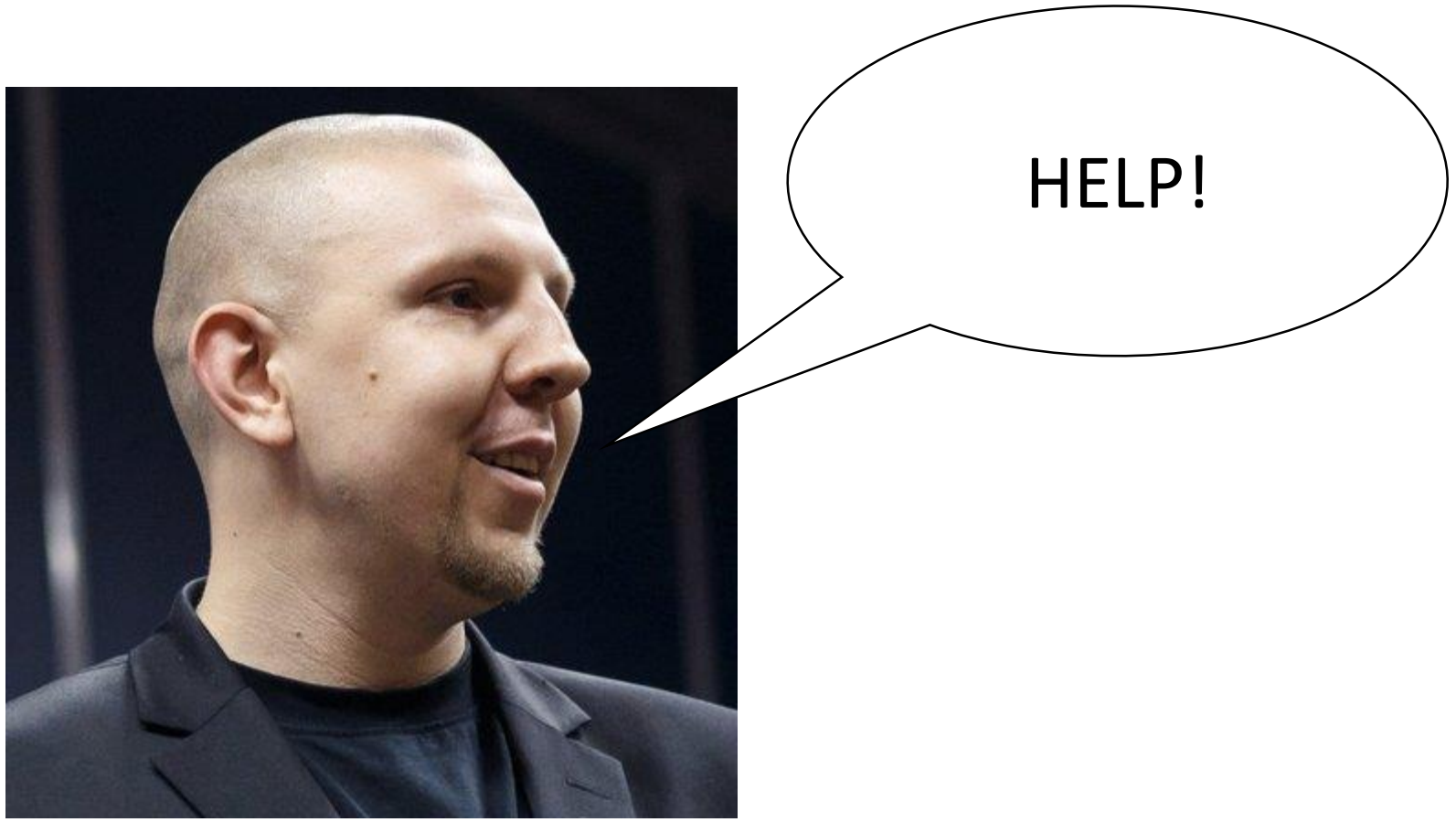
Технические специалисты





Общение между участниками проекта формирует **ubiquitous language**

Бизнес-сценарий


HELP!

HELP!

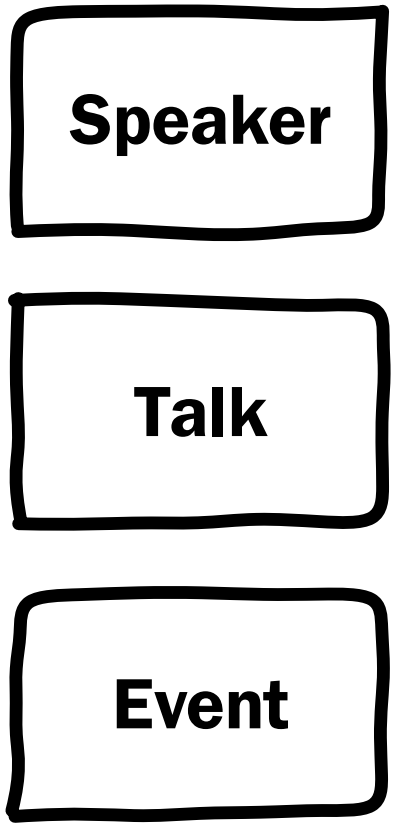
DOTNEXT

Management Software

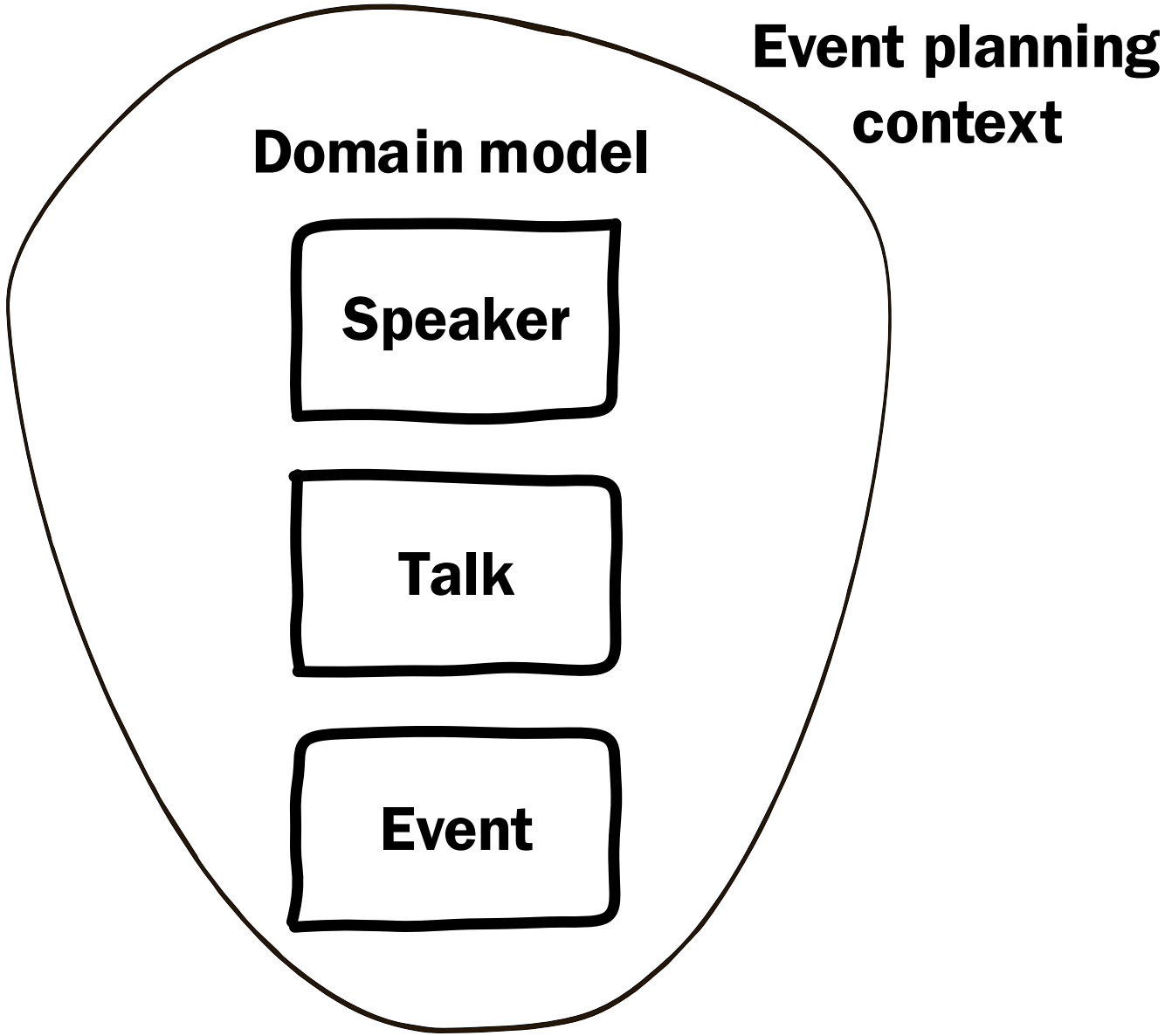
Сценарий:

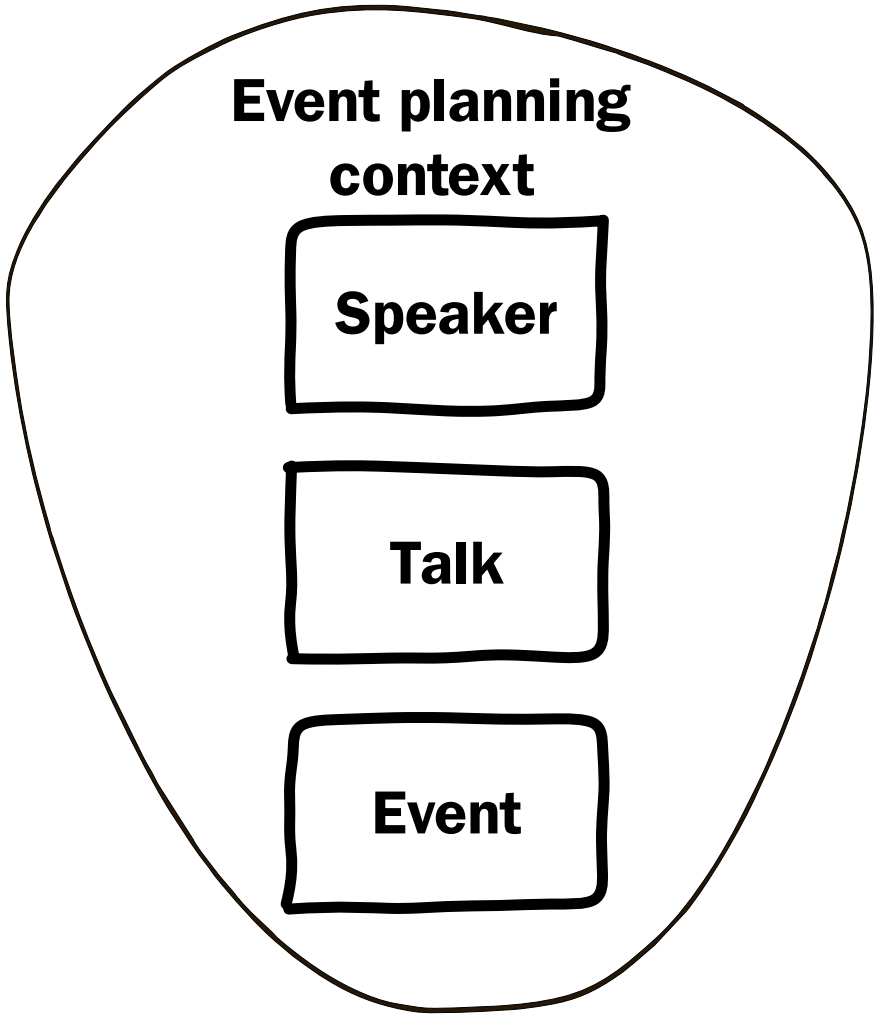
Докладчик регистрируется на событие и добавляет информацию о докладе

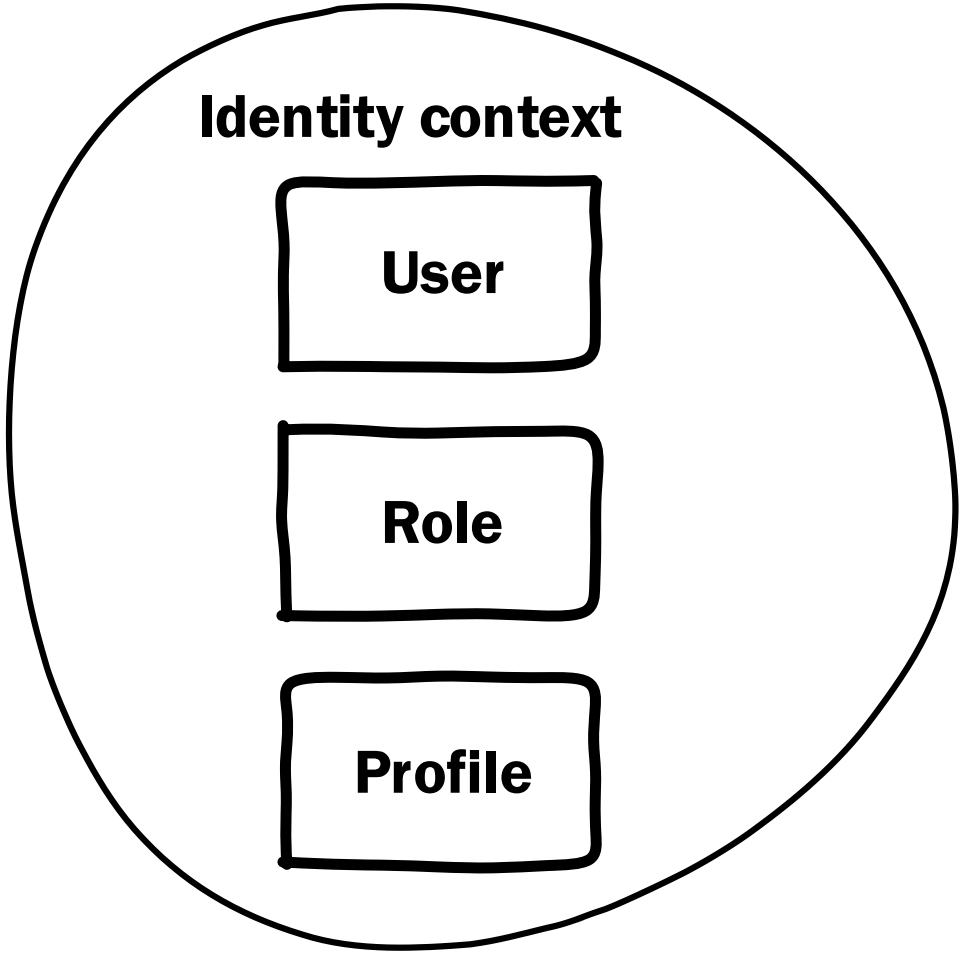
Domain model

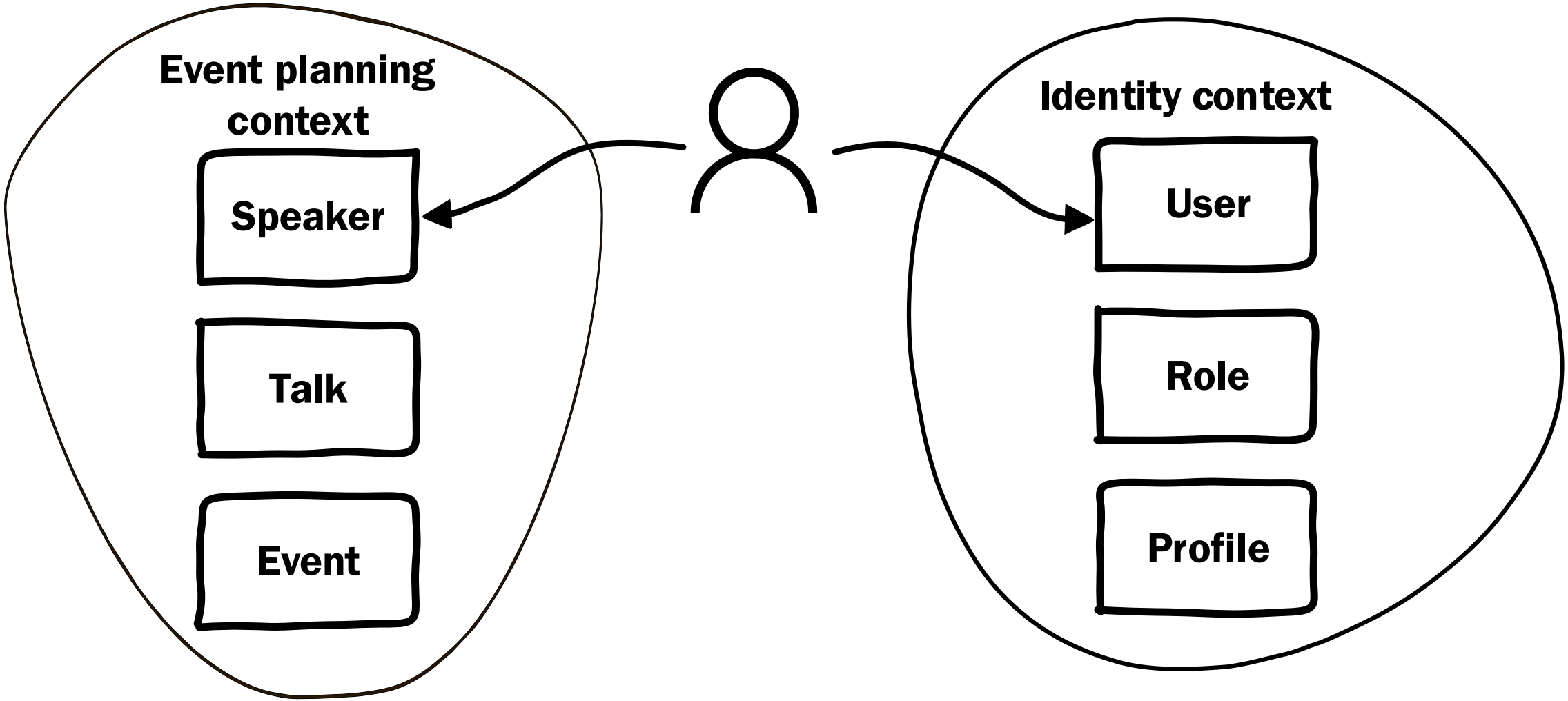


Доменная модель и ubiquitous language ограничены контекстом (**bounded context**)


Event planning context
Domain model
Speaker
Talk
Event


Event planning context
Speaker
Talk
Event


Identity context
User
Role
Profile


Event planning context
Speaker
Talk
Event
Identity context
User
Role
Profile

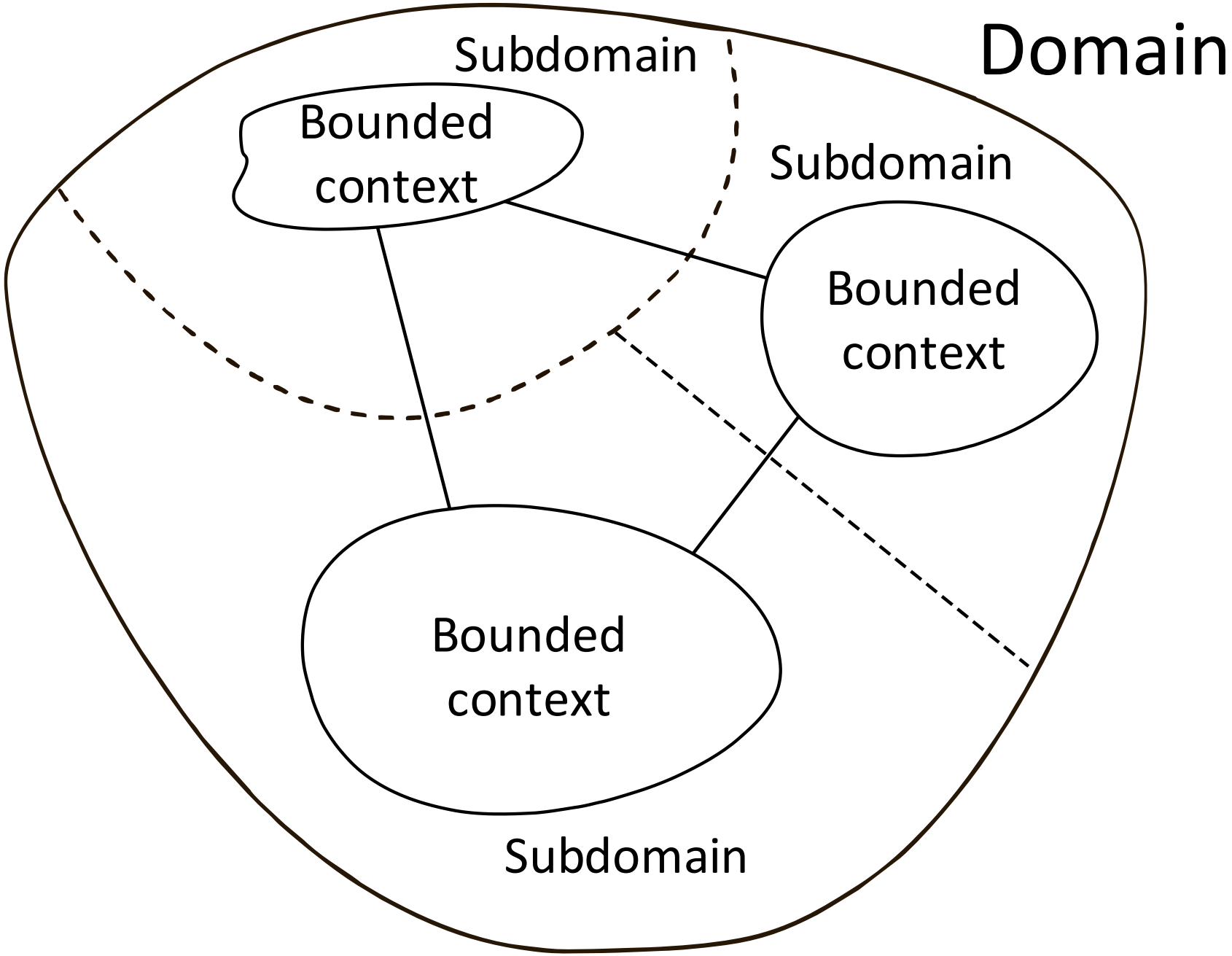
Демо

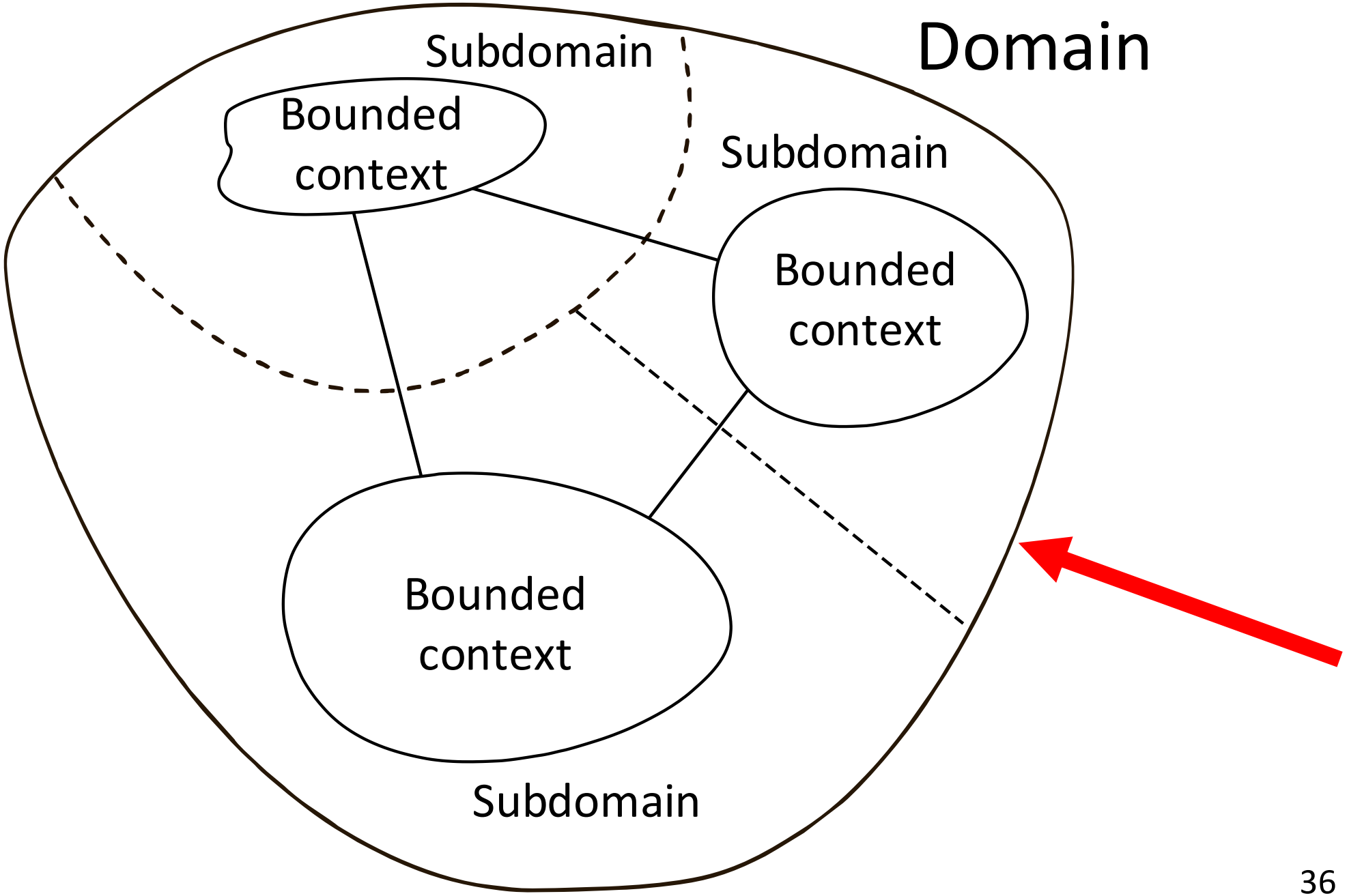
Sales service

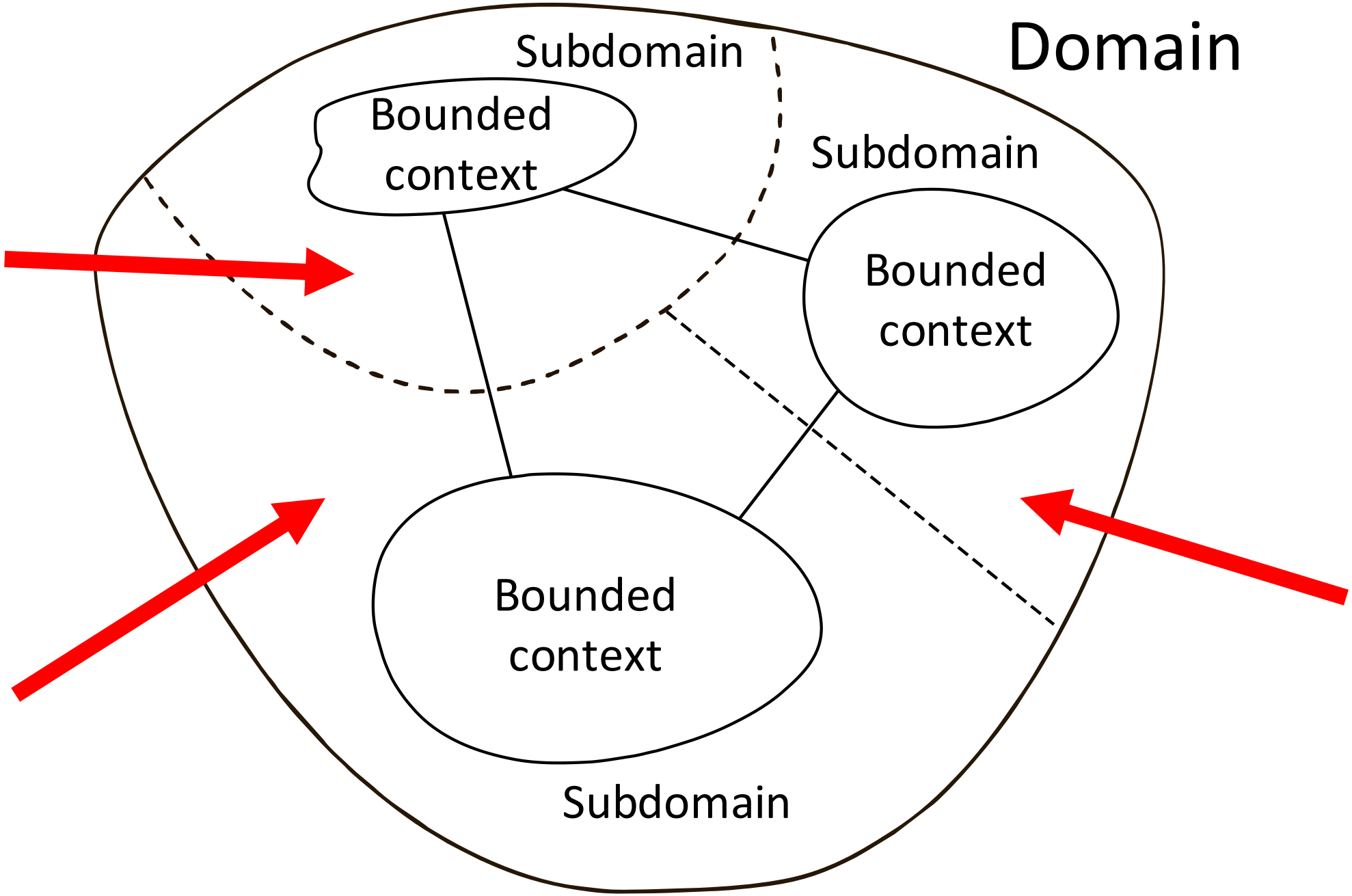
Сценарий:

По нажатию на Checkout считается окончательная цена заказа с учетом скидки постоянного клиента, и заказ переходит в статус «Ожидание оплаты»

Доменная модель и бизнес-логика должны использовать ubiquitous language


Domain
Subdomain
Bounded context
Subdomain
Bounded context
Bounded context
Subdomain


Domain
Subdomain
Bounded context
Subdomain
Bounded context
Bounded context
Subdomain


Domain
Subdomain
Bounded context
Subdomain
Bounded context
Bounded context
Subdomain

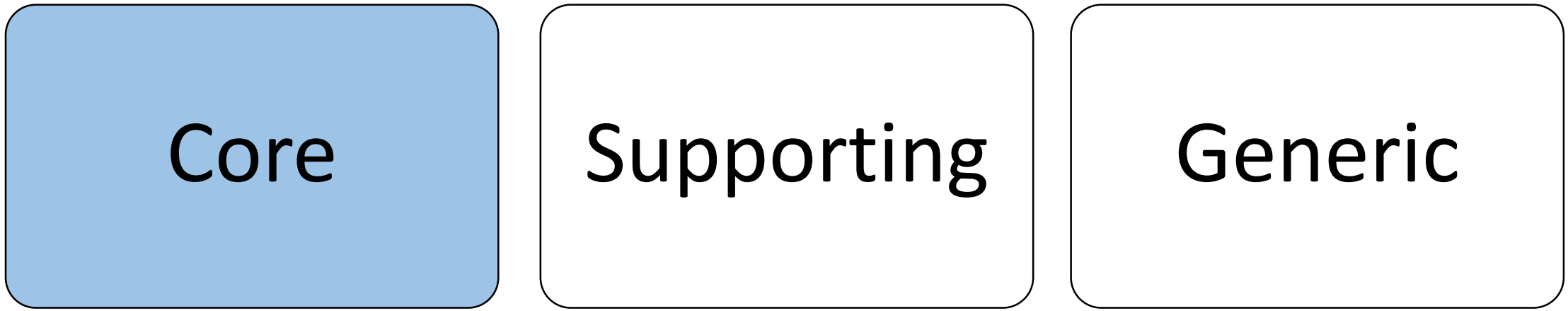
Subdomains

Core

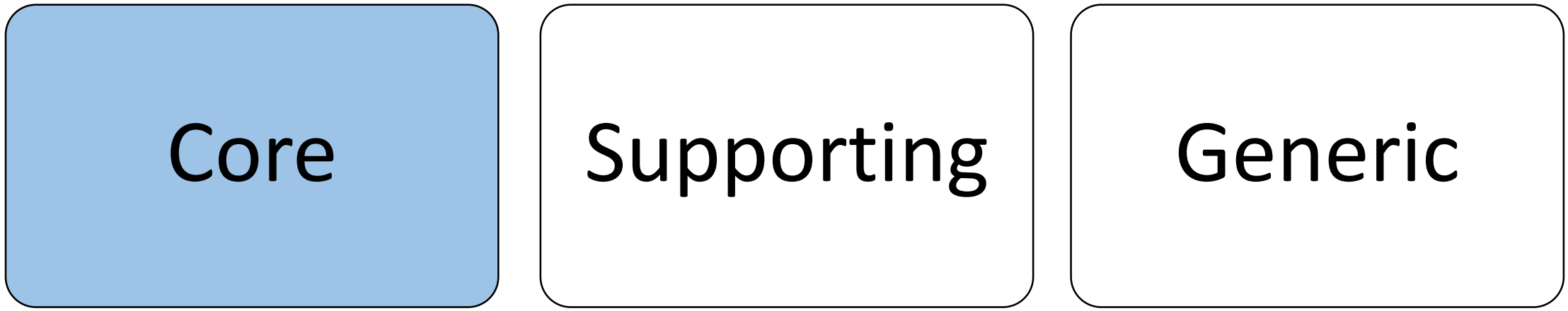
Supporting

Generic

Subdomains

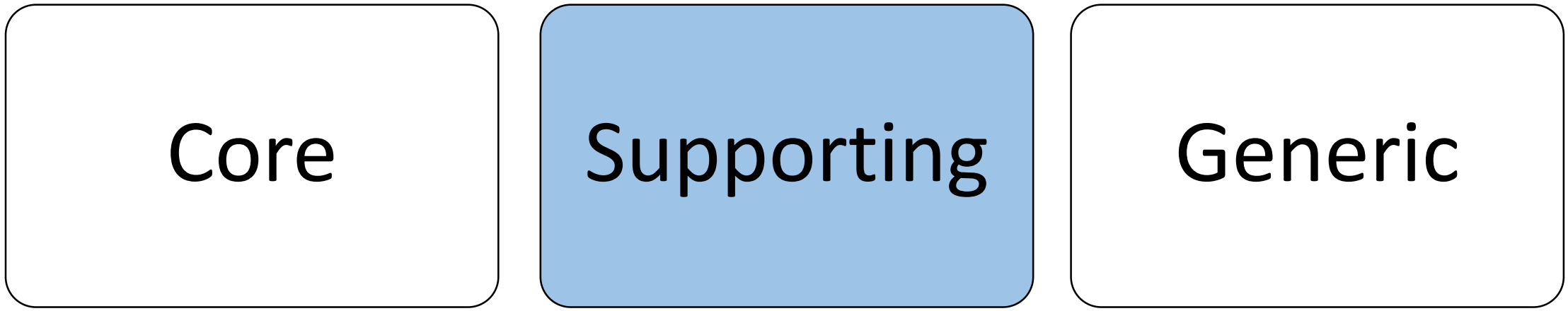


Subdomains

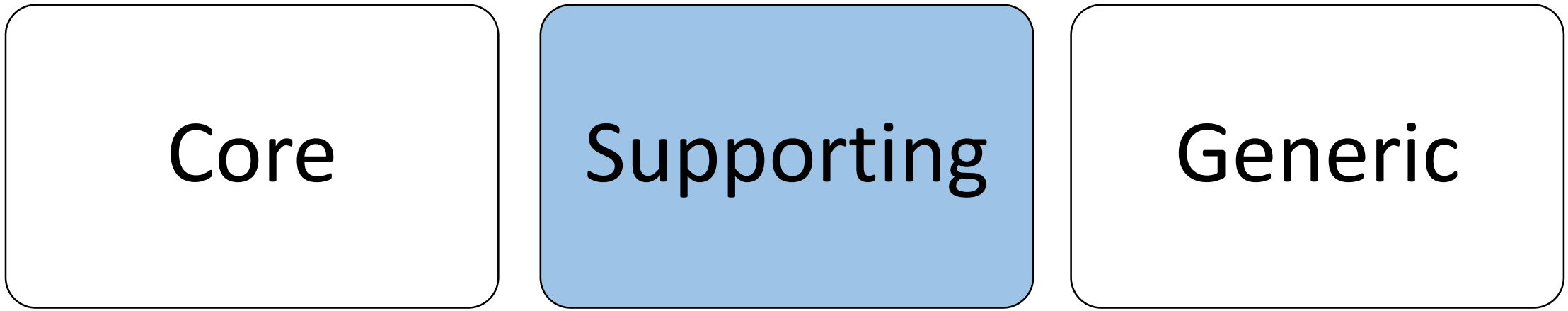


Контент

Subdomains

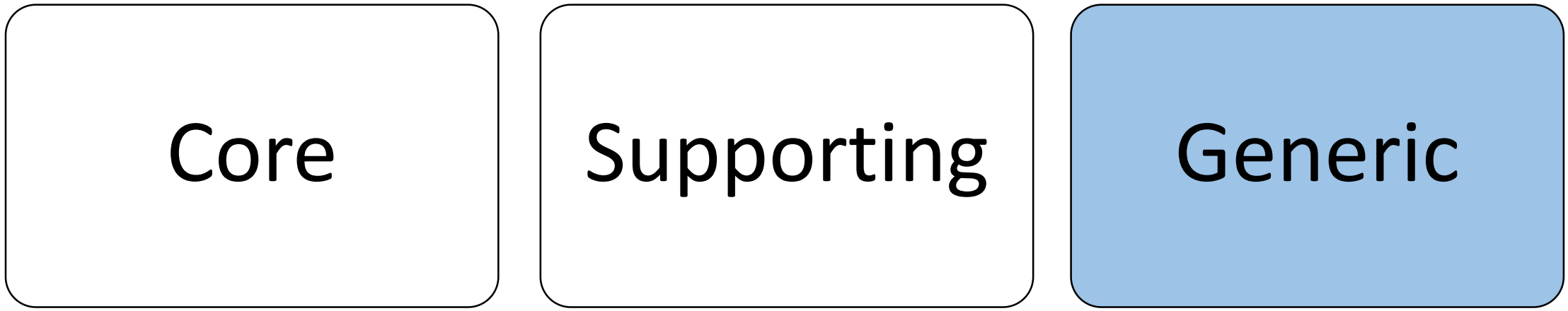


Subdomains

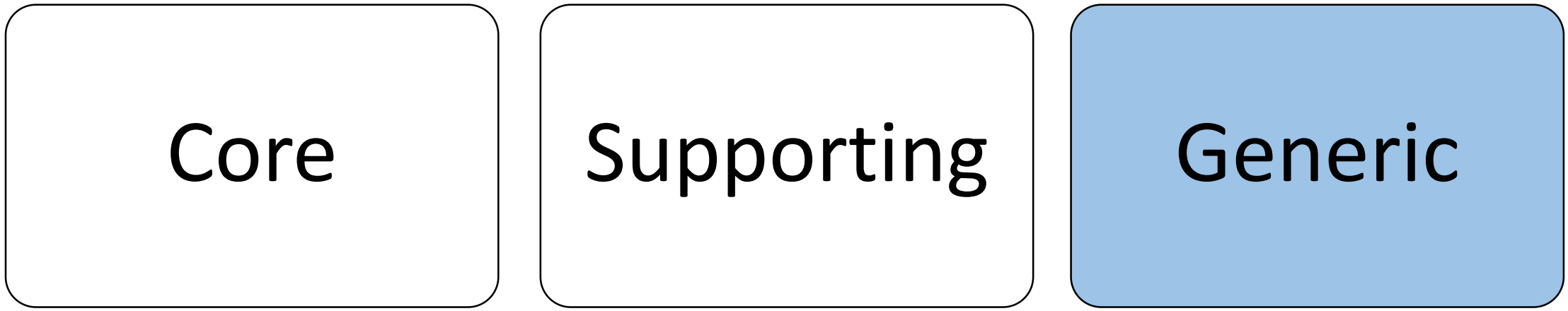


Маркетинг

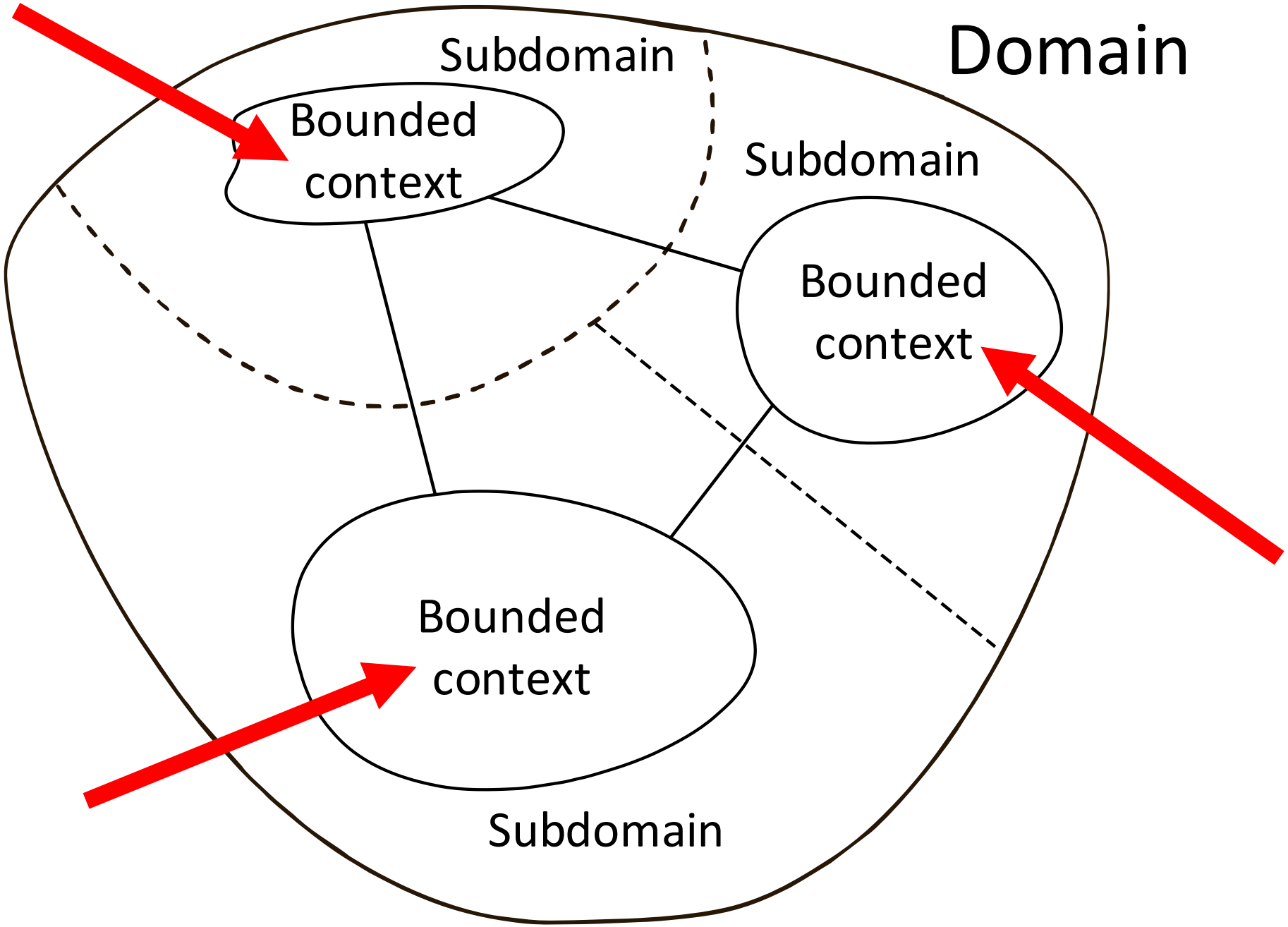
Subdomains



Subdomains



Билеты


Domain
Subdomain
Bounded context
Subdomain
Bounded context
Bounded context
Subdomain

Задачи бизнеса

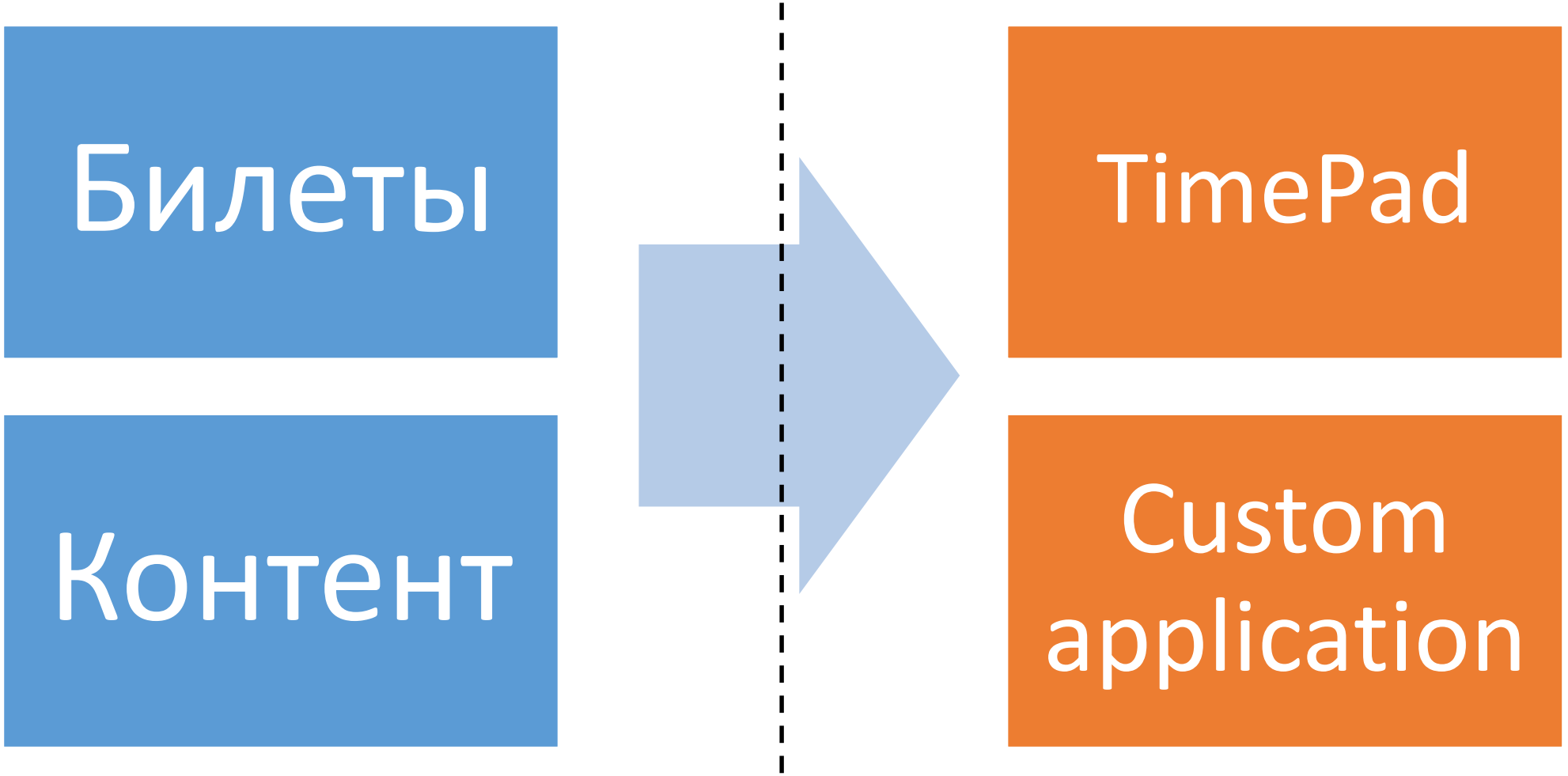
Программные решения

Subdomain

Bounded context

Бухгалтерия

Программные решения


Билеты
Контент
TimePad
Custom application

Bounded Context должен быть таким, чтобы Ubiquitous Language был полным и однозначным

1 контекст = 1 поддомен

1 контекст = 1 поддомен

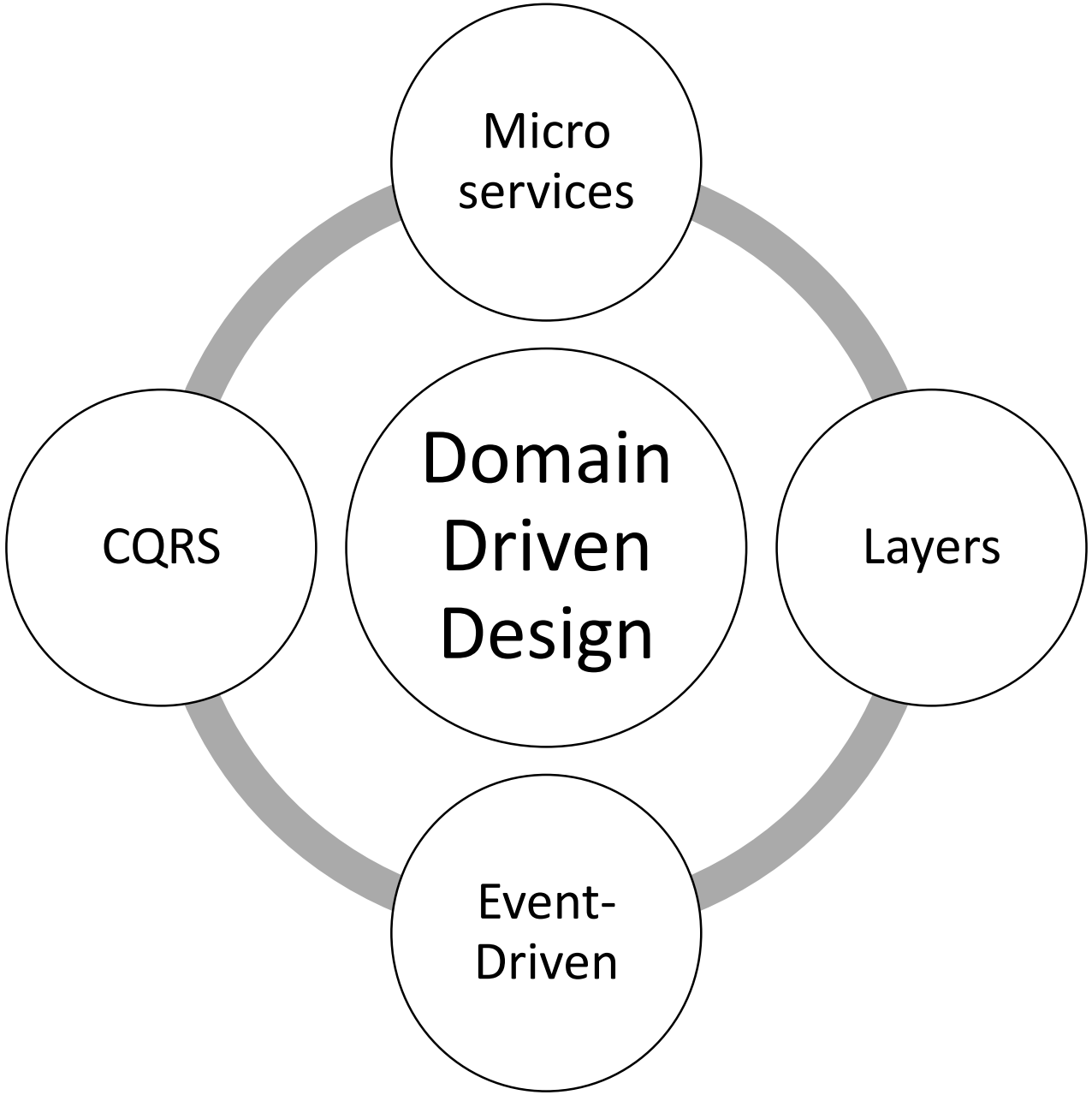
1 контекст = 1 микросервис

1 контекст = 1 поддомен

1 контекст = 1 микросервис

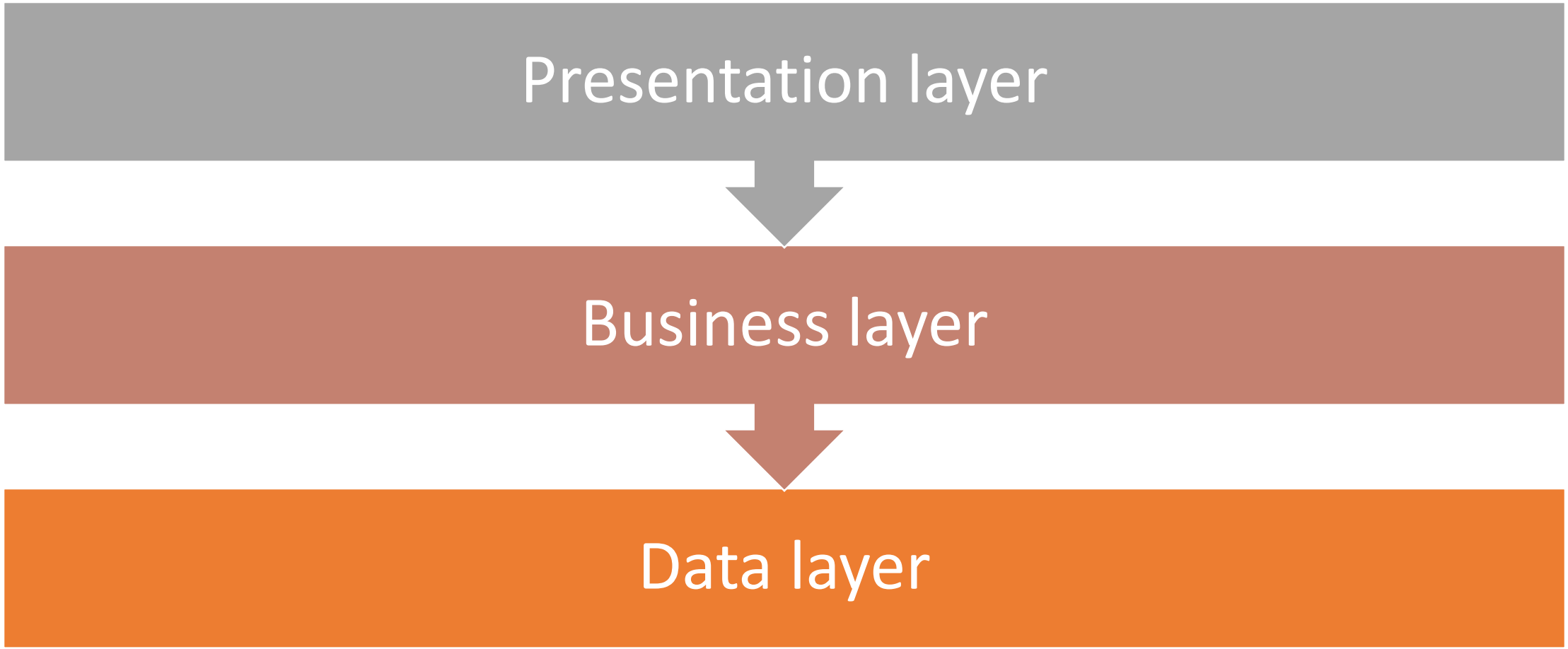
1 контекст = X человек

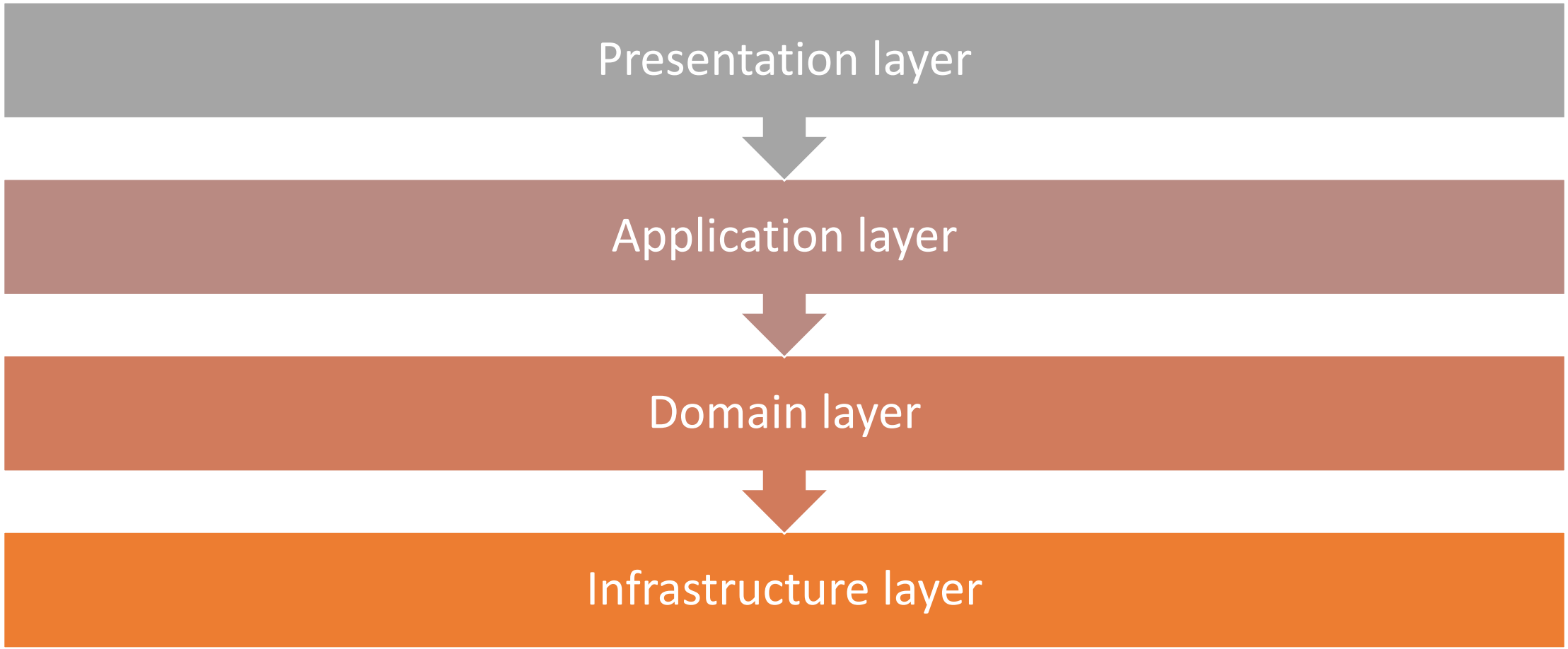
Архитектура и управление зависимостями

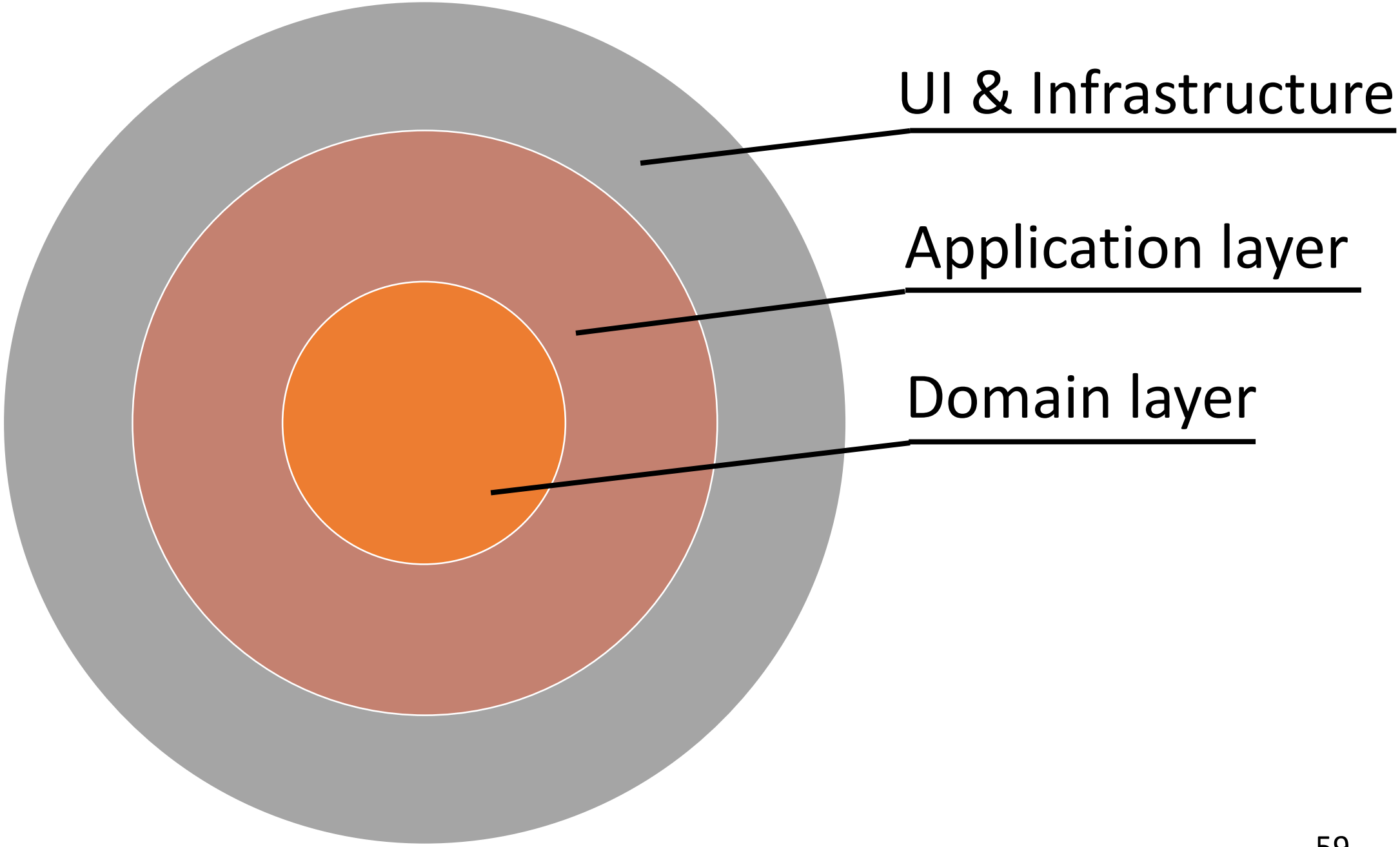

Micro services
CQRS
Domain Driven Design
Layers
Event-Driven

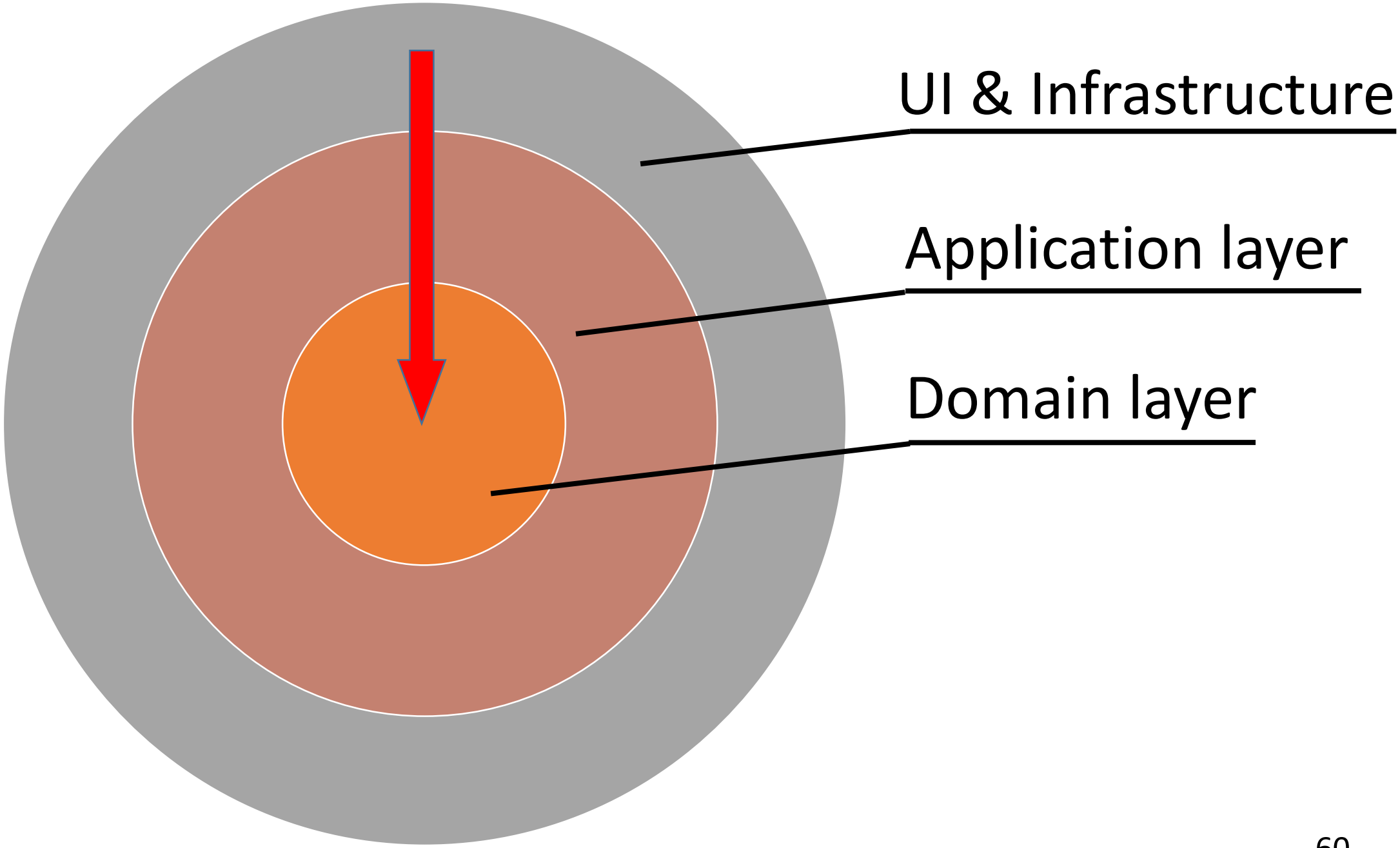
Цель:

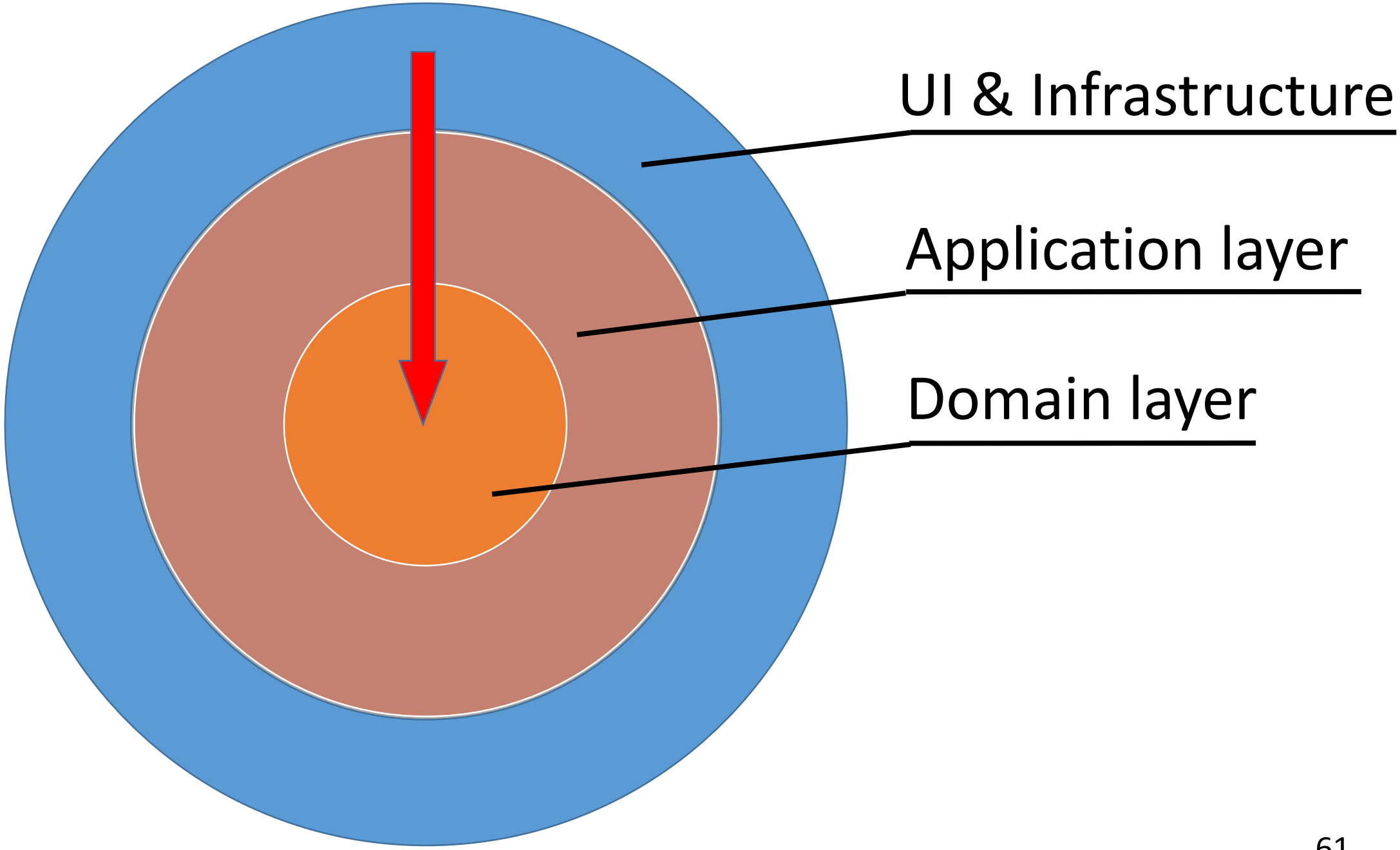
максимально избавить доменную логику от зависимостей

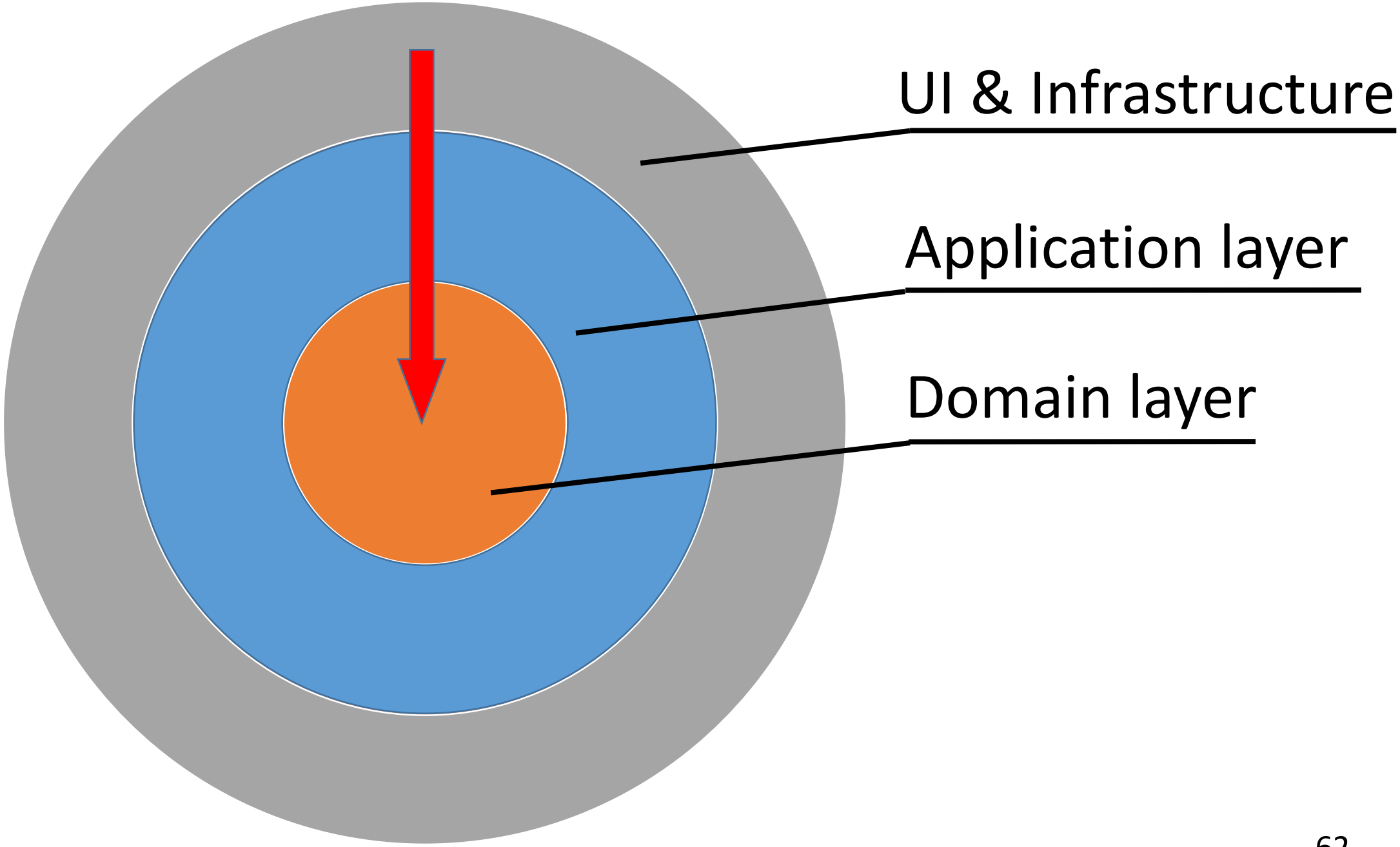

Presentation layer
Business layer
Data layer

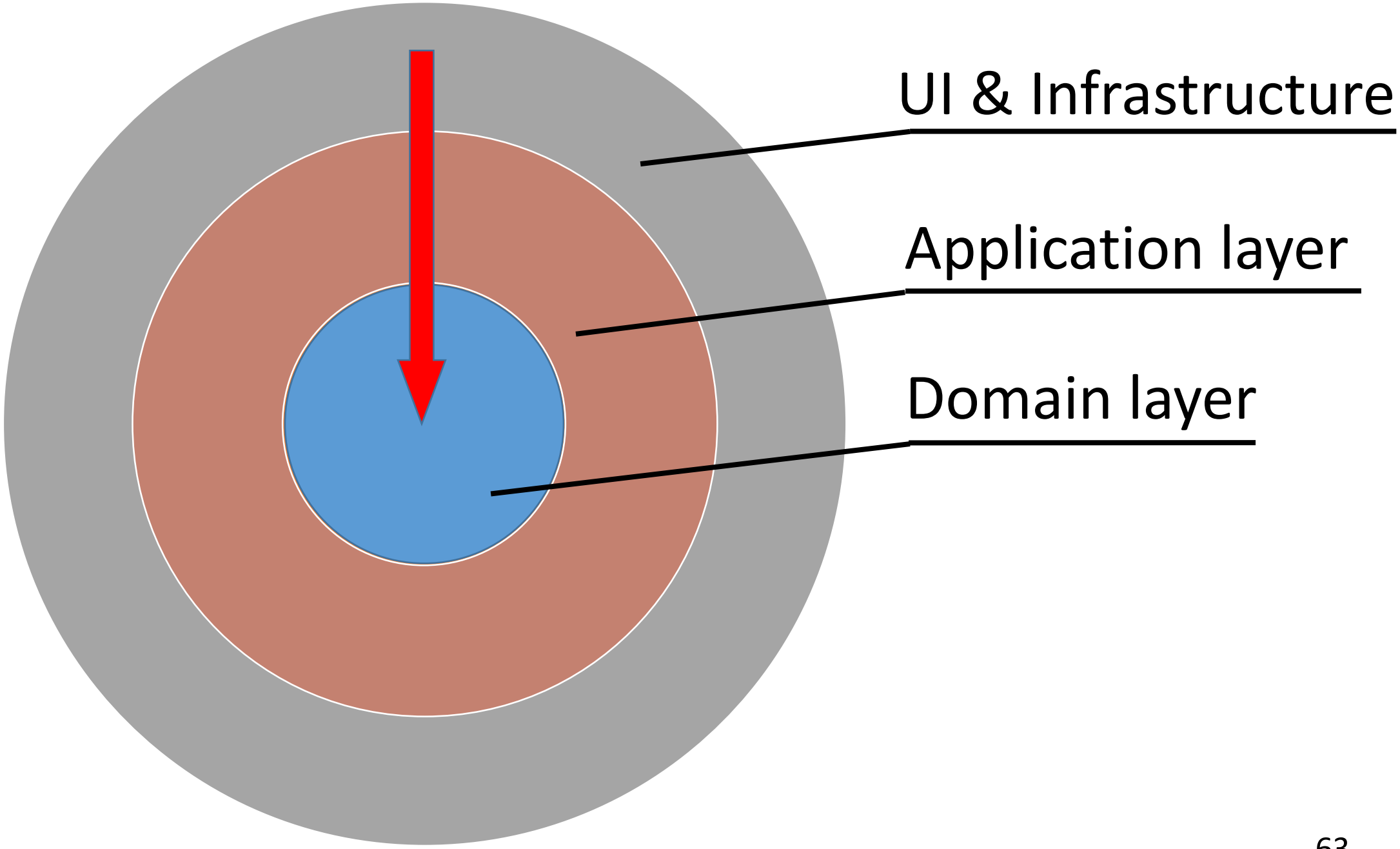

Presentation layer
Application layer
Domain layer
Infrastructure layer


UI & Infrastructure
Application layer
Domain layer


UI & Infrastructure
Application layer
Domain layer

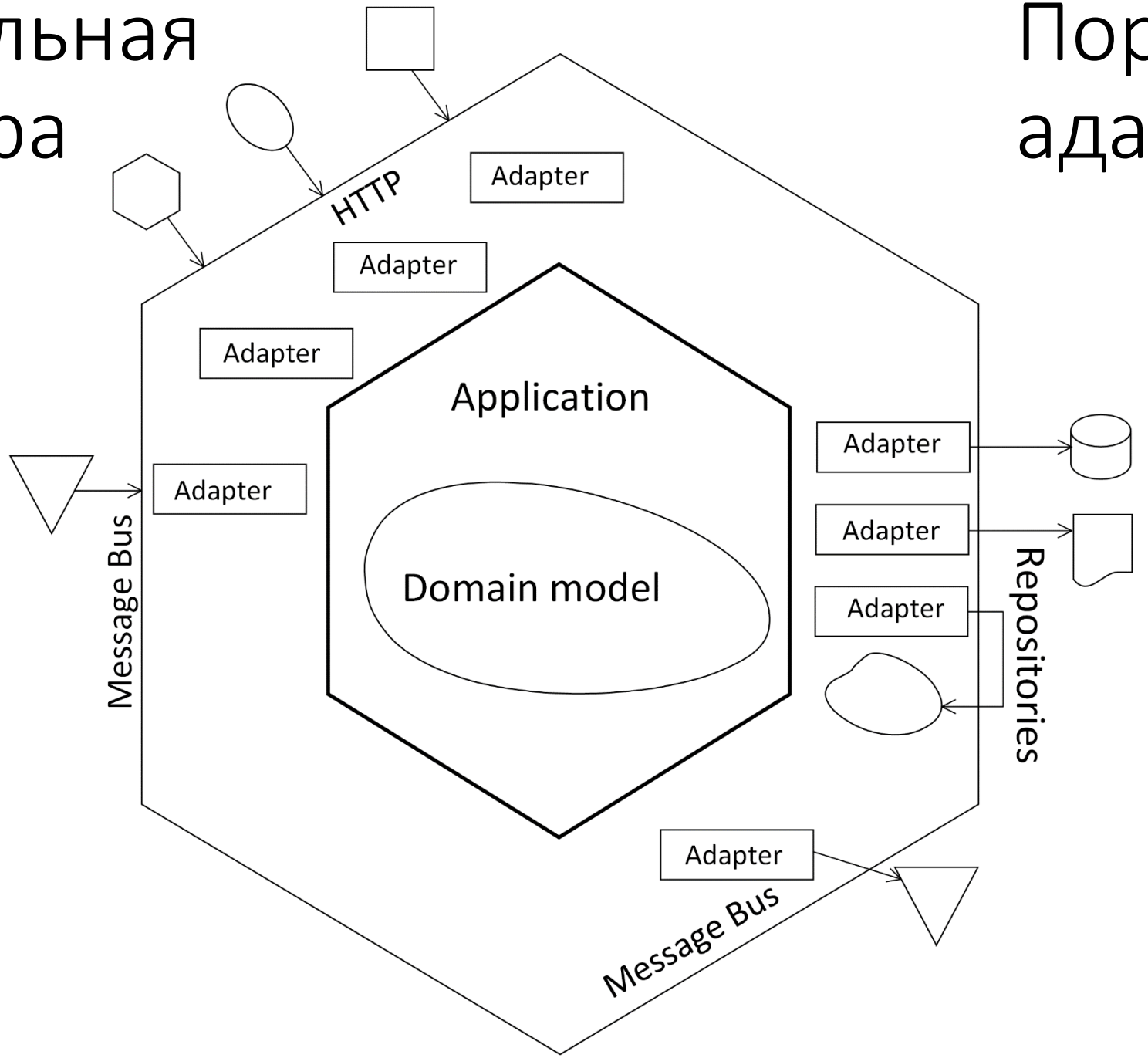

UI & Infrastructure
Application layer
Domain layer


UI & Infrastructure
Application layer
Domain layer


UI & Infrastructure
Application layer
Domain layer

Гексагональная архитектура

Порты и адаптеры

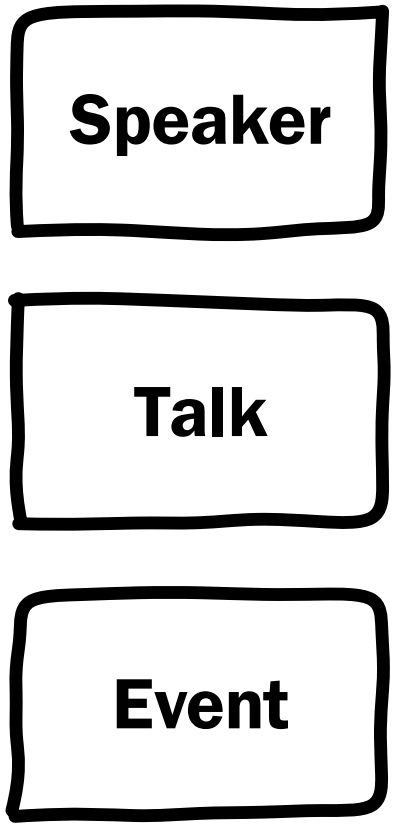


Немного про тактические паттерны:

Separated Interface

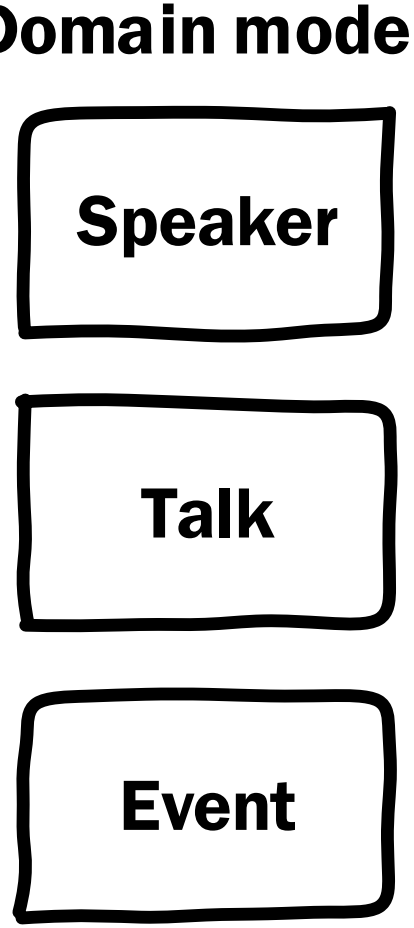
Еще раз о языке

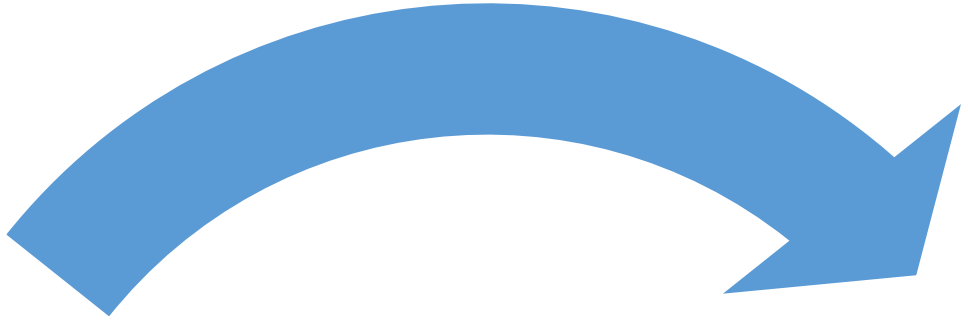
Domain model



Сценарий:

Докладчик регистрируется на событие и добавляет информацию о докладе





Русский English

```
// НДФЛ
ДатаОперацииПоНалогомИВзносам = Мин(Дата, КонецМесяца(МесяцНачисления));

УчетНДФЛ.СформироватьДоходыНДФЛПоНачислениямСПланируемойДатойВыплаты(
                                Движения,
                                Организация,
                                ДатаОперацииПоНалогомИВзносам,
                                МесяцНачисления);

УчетНДФЛ.СформироватьНалогиВычеты(
                                Движения,
                                Организация,
                                ДатаОперацииПоНалогомИВзносам,
                                ДанныеДляПроведения.НДФЛ);

УчетНДФЛ.СформироватьСоциальныеВычетыПоУдержаниям(
                                Ссылка,
                                Движения,
                                Организация,
                                ДатаОперацииПоНалогомИВзносам,
                                МесяцНачисления,
                                ДанныеДляПроведения.УдержанияПоСотрудникам);
```

```
        private readonly IРепозиторийЗаказов _репозиторийЗаказов;
        private readonly КалькуляторСкидки _калькуляторСкидки;
        public СервисПодтвержденияЗаказаV2(IРепозиторийЗаказов репозиторийЗаказов,
                                           КалькуляторСкидки калькуляторСкидки)
        {
            _репозиторийЗаказов = репозиторийЗаказов;
            _калькуляторСкидки = калькуляторСкидки;
        }

        public void Подтвердить(long кодЗаказа)
        {
            var заказ = _репозиторийЗаказов.ПолучитьЗаказ(кодЗаказа);
            var скидка = _калькуляторСкидки.РассчитатьСкидку(заказ.КодКлиента);
            заказ.ПрименитьСкидку(скидка);
            заказ.Статус = СтатусЗаказа.ОжидаетОплаты;
            _репозиторийЗаказов.СохранитьЗаказ(заказ);
        }
    }
}
```

```
        private readonly IРепозиторийЗаказов _репозиторийЗаказов;
        private readonly КалькуляторСкидки _калькуляторСкидки;
        public СервисПодтвержденияЗаказаV2(IРепозиторийЗаказов репозиторийЗаказов,
                                           КалькуляторСкидки калькуляторСкидки)
        {
            _репозиторийЗаказов = репозиторийЗаказов;
            _калькуляторСкидки = калькуляторСкидки;
        }

        public void Подтвердить(long кодЗак
        {
            var заказ = _репозиторийЗаказов
            var скидка = _калькуляторСкидки
            заказ.ПрименитьСкидку(скидка);
            заказ.Статус = СтатусЗаказа.Ожи
            _репозиторийЗаказов.СохранитьЗа
        }
    }
}
```

Подытожим

1. Общение + общение + общение = ubiquitous language

Подытожим

1. Общение + общение + общение = ubiquitous language
2. Модели ограничиваются контекстами

Подытожим

1. Общение + общение + общение = ubiquitous language
2. Модели ограничиваются контекстами
3. Минимум зависимостей для доменной модели

Подытожим

1. Общение + общение + общение = ubiquitous language
2. Модели ограничиваются контекстами
3. Минимум зависимостей для доменной модели
4. Максимально выразительный код бизнес-логики

Подытожим

1. Общение + общение + общение = ubiquitous language
2. Модели ограничиваются контекстами
3. Минимум зависимостей для доменной модели
4. Максимально выразительный код бизнес-логики
5. ?????
6. PROFIT

Полезные ссылки:

1. Habrahabr
   - https://habrahabr.ru/users/marshinov (например Как мы попробовали DDD, CQRS и Event Sourcing и какие выводы сделали)
2. Блоги
   - https://lostechies.com/jimmybogard/
   - https://blog.byndyu.ru
3. DDD и функциональное программирование
   - https://fsharpforfunandprofit.com/
4. Hexagonal Architecture (Ports&Adapters)
   - https://herbertograca.com/2017/09/14/ports-adapters-architecture/

Спасибо за внимание!

alexey.merson@gmail.com

https://github.com/a-merson/DotNext2018