

DDDamn



good

!

m²tech

Ахтям Сакаев <m2.ru>



О чём

Лучшие практики для
командной разработки



m²tech





Цель



Вдохновить применять DDD

m²tech





0 ce6e

lead software engineer
scala developer
OSS contributor

m²tech



«Метр квадратный»



технологическая компания на стыке
недвижимости и финансового сектора

java, scala, typescript

zio, cats

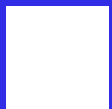
kafka, pg, mongodb, elasticsearch



m²tech

Что будет

Domain-Driven Design
Functional programming
Unix philosophy



m²tech



Чего не будет



Microservice Architecture
Hexagonal Architecture
Event Sourcing



m²tech



Agenda

1

**DDD
Event Storming**



2

**DDD Aggregate
Bounded context**



3

**Distributed system
Unix pipes**



m²tech



История

Пользователь узнаёт
о новых квартирах быстро

m²tech





Задача

content-based subscription

m²tech



Детали

Подписка на объявления

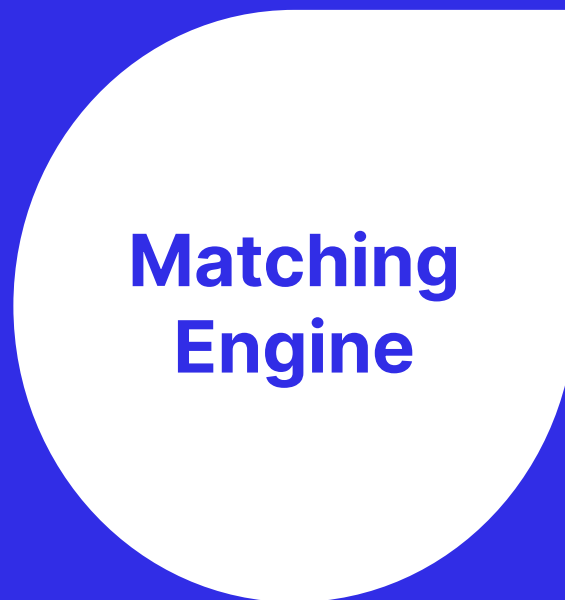
Можно добавлять и убирать емейлы

Это настоящая задача 🤖

m²tech



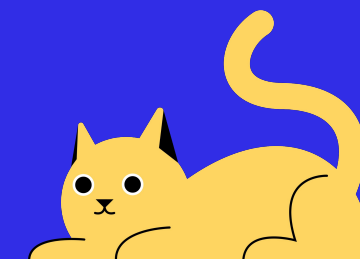
Команды



Объявление



видовая



<5 млн



Подписаться (< 5 млн)

Удалить e-mail (black@hole.com)



m²tech

Вопросы

Все одинаково поняли задачу?

Получится сделать систему понятной?

Как поддерживать решение через год?



m²tech



Domain-Driven Design

Концепт (Eric Evans, 2003)

Упрощение систем

m²tech



Use-cases

User



AddEmail

User



Subscribe



Модель

AddEmail(Email) => Unit

RemoveEmail(Email) => Unit

Subscribe(Subscription, Email) => Unit

GetSubscriptions => List[Subscription]



Domain knowledge

```
AddEmail(Email) => Unit
```

```
RemoveEmail(Email) => Unit
```

```
Subscribe(Subscription, Email) => Unit
```

```
GetSubscriptions => List[Subscription]
```



Domain-Driven Design

Бизнес процессы в явном виде в коде

m²tech





Alberto Brandolini

@ziobrando

Replying to @JeromeAvoustin

"It's not stakeholder knowledge but developers' ignorance that gets deployed into production"
[#QuoteServer](#)

11:59 AM · May 27, 2016 · Twitter for iPhone

m²tech



Пример без DDD

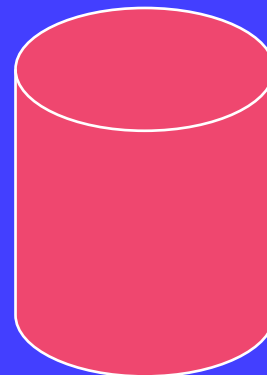


Subscribe



UpdateSubscription

api



user_data



Ubiquitous

Language

Термин из DDD («Унифицированный язык»)

Упрощает общение внутри команды 🌿

m²tech



Ubiquitous Language

👉 Не общий язык на всю компанию

a. name clash

b. scopes

Проблема с 🇷🇺



Как моделировать в команде?

- Хаос
- Разногласия
- Безразличие



Event Storming

Воркшоп (Alberto Brandolini, 2013)

Коллаборативный процесс

Вовлечены инженеры, менеджеры, дизайнеры...

m²tech



Event Storming



Event Storming

Моделирование времени
В ЯВНОМ ВИДЕ

EmailAdded

Subscribed

m²tech



Event Storming

Sync ментальной модели 🌿

«Полная картина» для бизнеса

Эволюционный процесс

m²tech



Новые требования



Удаление емейла деактивирует подписки
...однажды

m²tech



Event Storming

RemoveEmail

Emails and
Subscriptions

Unsubscribed

EmailRemoved



m²tech

Получится написать такой
же простой код? 🤔

m²tech



Итоги

Ubiquitous Language делает модель понятнее
Моделируем в команде — Event Storming

m²tech



1

**DDD
Event Storming**



2

**DDD Aggregate
Bounded context**



3

**Distributed system
Unix pipes**

m²tech





Дискуссия с **экспертом**

m²tech



Удаление емейла деактивирует подписки 🧑💻

m²tech



Модель

```
case class User(  
  id: UserId,  
  emails: Set[Email],  
  subscriptions: Set[Subscription]  
)  
def removeEmail(u: User, e: Email): User = ???
```



Тестирование

```
val u = User(
    1,
    Set("black@hole.com"),
    Set(Subscription(1, "black@hole.com"))
)
val expected = User(1, Set(), Set())
val result = removeEmail(s, "black@hole.com")
result.shouldBe(expected)
```



Поддержали новые
требования 🙌

m²tech



Что потребовалось?

1. Знать текущее состояние User
2. Изменить состояние User
3. Вернуть изменённое состояние User



Reducer (FSM, fold)



$(S, A) \Rightarrow S$

```
def removeEmail(u: User, e: Email): User = ???
```



DDD Aggregate

Is a cluster of associated objects that we treat as a unit for the purpose of data changes.

— Eric Evans (Domain-Driven Design)



DDD Aggregate

Паттерн DDD с сигнатурой $(S, A) \Rightarrow S$



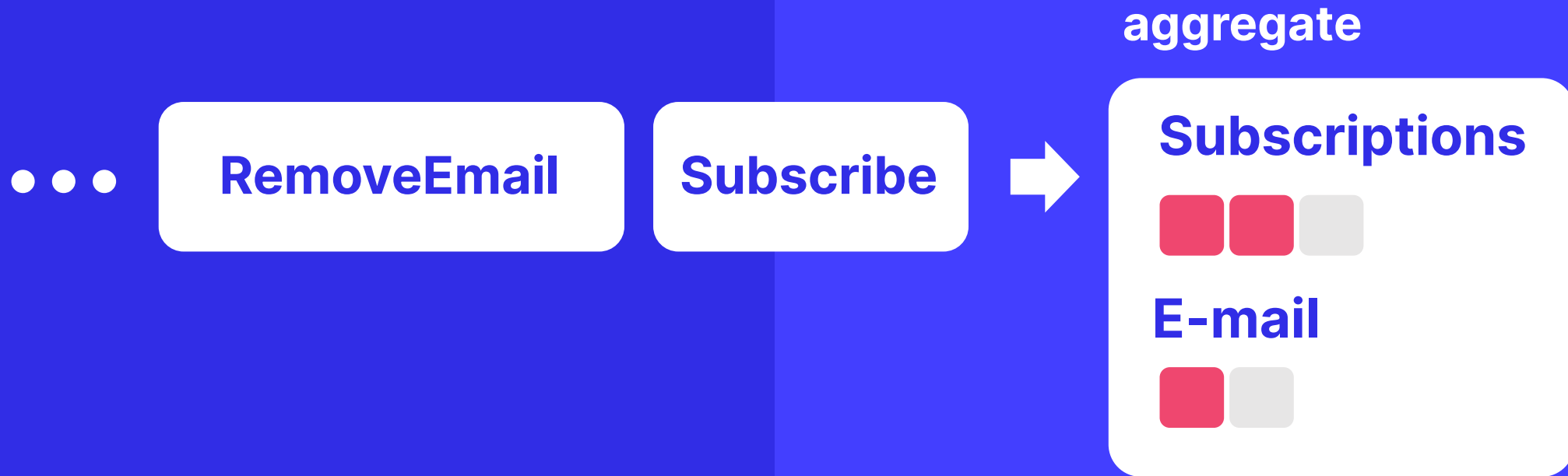
DDD Aggregate

Гарантирует сохранение инварианта
«Удаление емейла деактивирует
подписки».

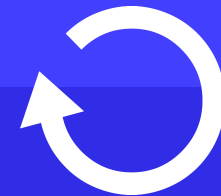
Но как? 🤔



Сериализация



lock / mutex, queue



m²tech

DDD Aggregate

Атомарность read-modify-write

Transactional consistency (не eventual)

m²tech



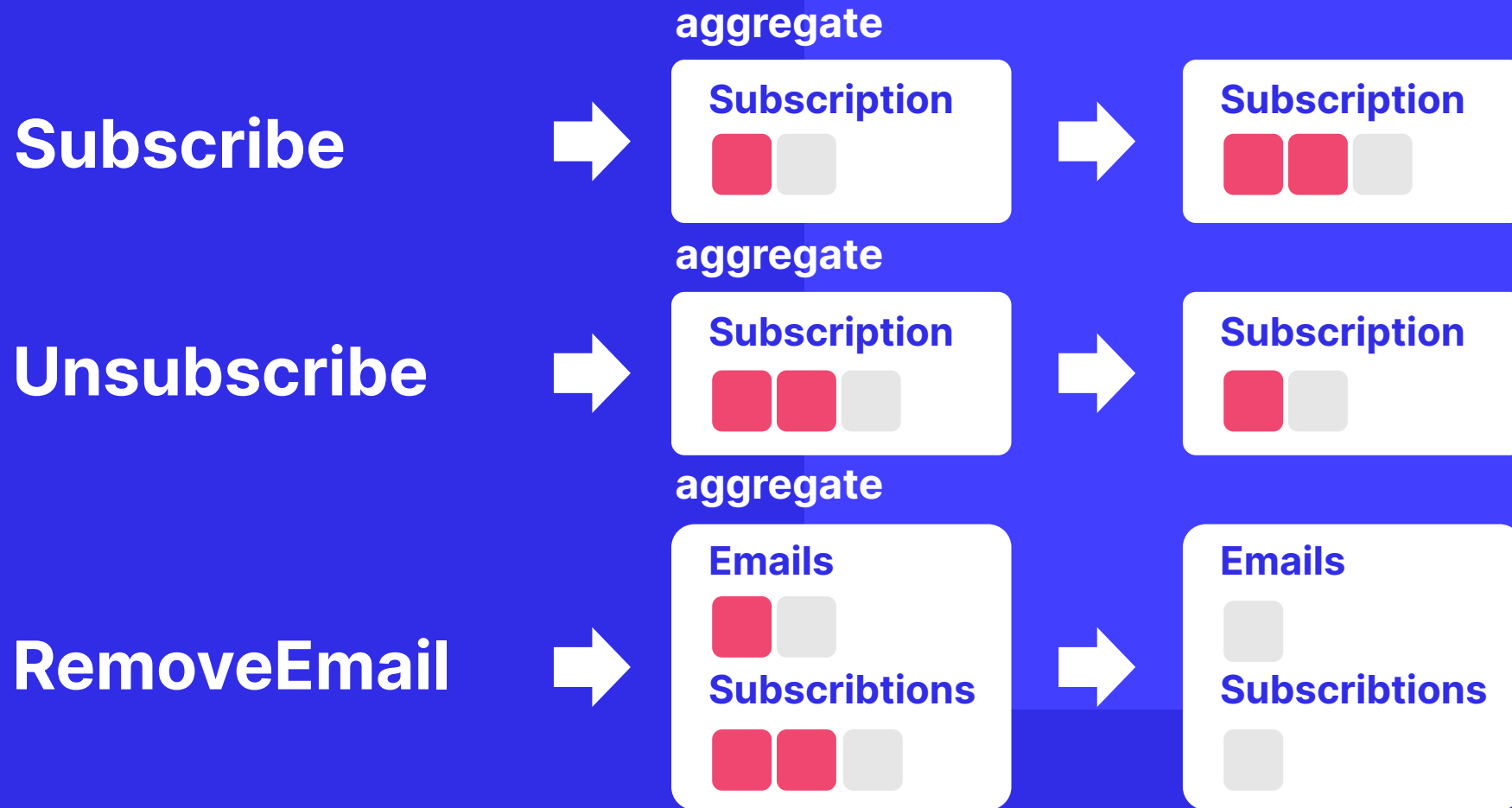
Зачем?

Race conditions don't exist.
Not in the business world anyway.
— Udi Dahan

m²tech



Поиск DDD Aggregate



DDD Aggregate

Про поведение

Не про модель данных

m²tech





Решили проблему



Что дальше?



m²tech



Разные команды и системы

AddEmail
RemoveEmail
Subscribe
GetSubscriptions

Subscription
management

Subscription
matching
???

subscriptions



m²tech

Bounded context

Is all about dealing with large
models and teams.

— Martin Fowler



Bounded context

- Основной DDD паттерн
- У каждого контекста свой Ubiquitous language
- Связь с командами разработки



Как определить

границы

Everything you can imagine is real.

— Pablo Picasso

m²tech



А серьёзно?

1. Компетенции, орг. структура, политика
2. Решение социотехнических проблем («проблема на вашей стороне», «коллеги отвечают раз в сутки»)



Размер Bounded context

Bounded context (subscriptions)

management



aggregate aggregate

matching



aggregate



m²tech

Итоги

- Поддержали новое бизнес правило (инвариант)
- Исключили баги с инвариантами (DDD Aggregate)
- Контекст связан с компетенциями команды



Но это только половина решения...



m²tech



Matching до сих пор не знает о подписках.
У management и matching разные хранилища.
DDD Aggregate — про локальные транзакции.

2PC? Тупик? 🤔

m²tech



1

DDD
Event Storming



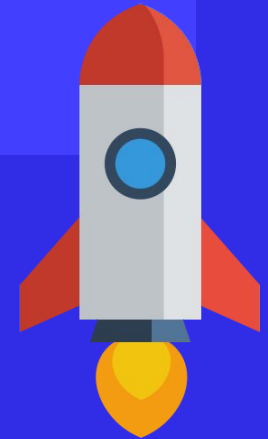
2

DDD Aggregate
Bounded context



3

Distributed system
Unix pipes



m²tech

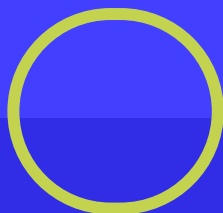


Дискуссия с **экспертом**



m²tech





Как заставить всё работать вместе?

Management + Matching



m²tech





➔ **Subscribe** ➔

management

Emails

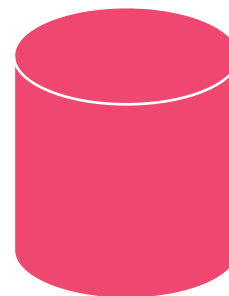


Subscriptions



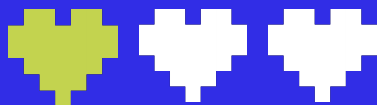
1

matcher



**Subscriptions
Index**

2



m²tech



Команда Subscribe меняет состояние:

- В разных системах
- С разными хранилищами

m²tech



Интеграция **систем**

Matcher должен знать о подписках

m²tech



Схема компонентов

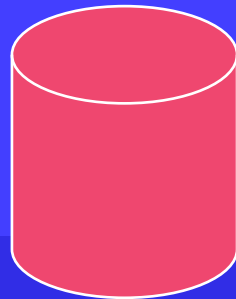


➔ **Subscribe**

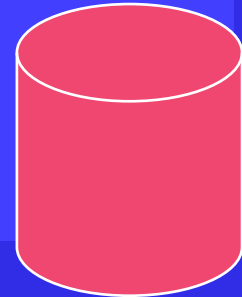
**Subscription
management**

Subscribed

Matching



Users



Subscriptions



m²tech

Eventual consistency





Как выразить ивенты в коде? 🤔



m²tech



Модель

```
def removeEmail(u: User, email: Email)  
: (User, List[Unsubscribed]) = ???
```

```
// IntegrationEvent (public api)
```



Тестирование

```
val u = User(
    1,
    Set("black@hole.com"),
    Set(Subscription(2, "black@hole.com"))
)
val expected = List(Unsubscribed(2))
val result = removeEmail(s, "black@hole.com")
result._2.shouldBe(expected)
```



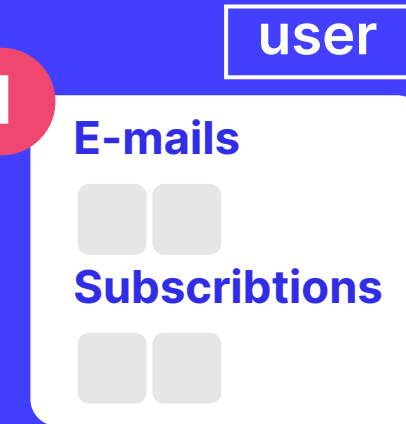
Решение



RemoveEmail



1



2

Unsubscribed



Unsubscribed



matcher index



Remove subscription



m²tech



Сохранить данные и отправить сообщение.
Одновременно.
Как? 🤔

m²tech



Outbox pattern

История
CDC (MongoChangeStream)



Outbox pattern



subscription	event
s1	ev1
	ev2

```
{  
  "subscription": {},  
  "events": [ ]  
}
```



Инфраструктура



Unix pipes

management | matcher



m²tech



Data-flow programming



subscribe



ev1

SubscribedA

ev2

SubscribedB

User+

[SubscribedEventA
SubscribedEventB]



SubscribedEventB

SubscribedEventA



matcher



subscriptions



match?



m²tech

Итоги

Перенос use-case 1:1 в решение
Минимальный когнитивный оверхед

m²tech





Стройте надежные и понятные системы! 🚀

m²tech





Introducing EventStorming
Alberto Brandolini 2013

Functional and Reactive Domain Modeling
Debasish Ghosh 2016

Designing Data-Intensive Applications
Martin Kleppmann 2017



m²tech





Life beyond Distributed Transactions
Pat Helland 2007

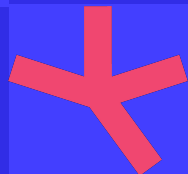
Effective Aggregate Design
Vaughn Vernon 2011

Your Server as a Function
Marius Eriksen 2013



m²tech





 github.com/asakaev

 t.me/techm2

m²tech

