

EventSourcing. Глубокое погружение



АНТОН ЖУКОВ

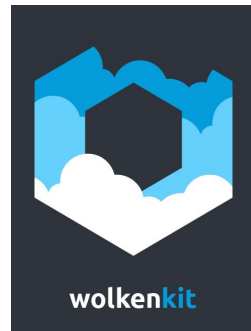
Контрол Системс



Источник опыта



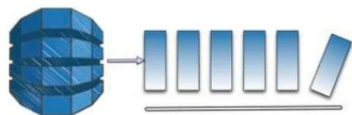
Axon



eventuate

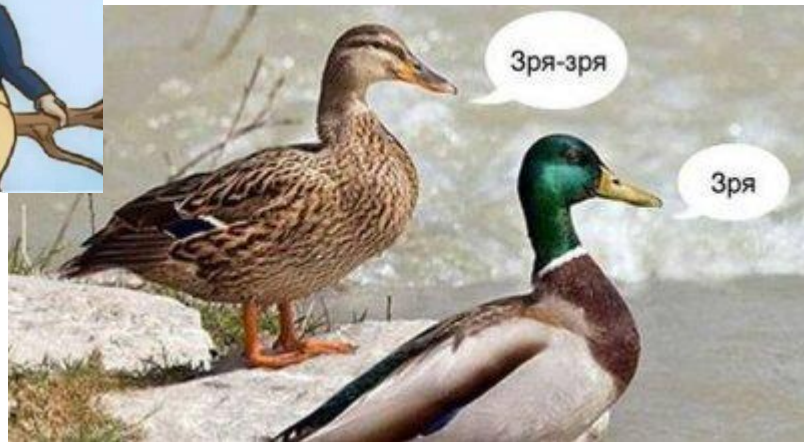


reSolve



DynamoDB

DynamoDB Streams



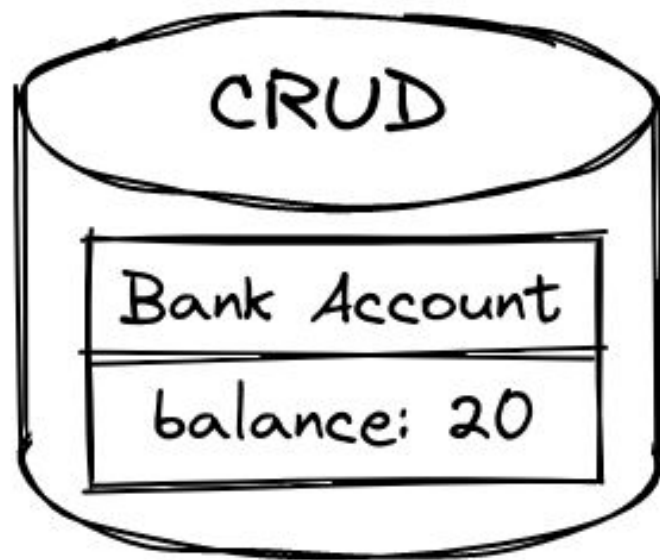
**Event
Sourcing**

VS

CRUD

CRUD и проблемы классического веб-приложения

- теряем знания о работе нашей системы - какие данные изменились? почему они изменились?
- необходимость вручную масштабировать, дробить на микросервисы, продумывать масштабирование БД
- сложно агрессивно развивать функционала, т.к. необходимы ручные миграции данных при изменении в схемах БД

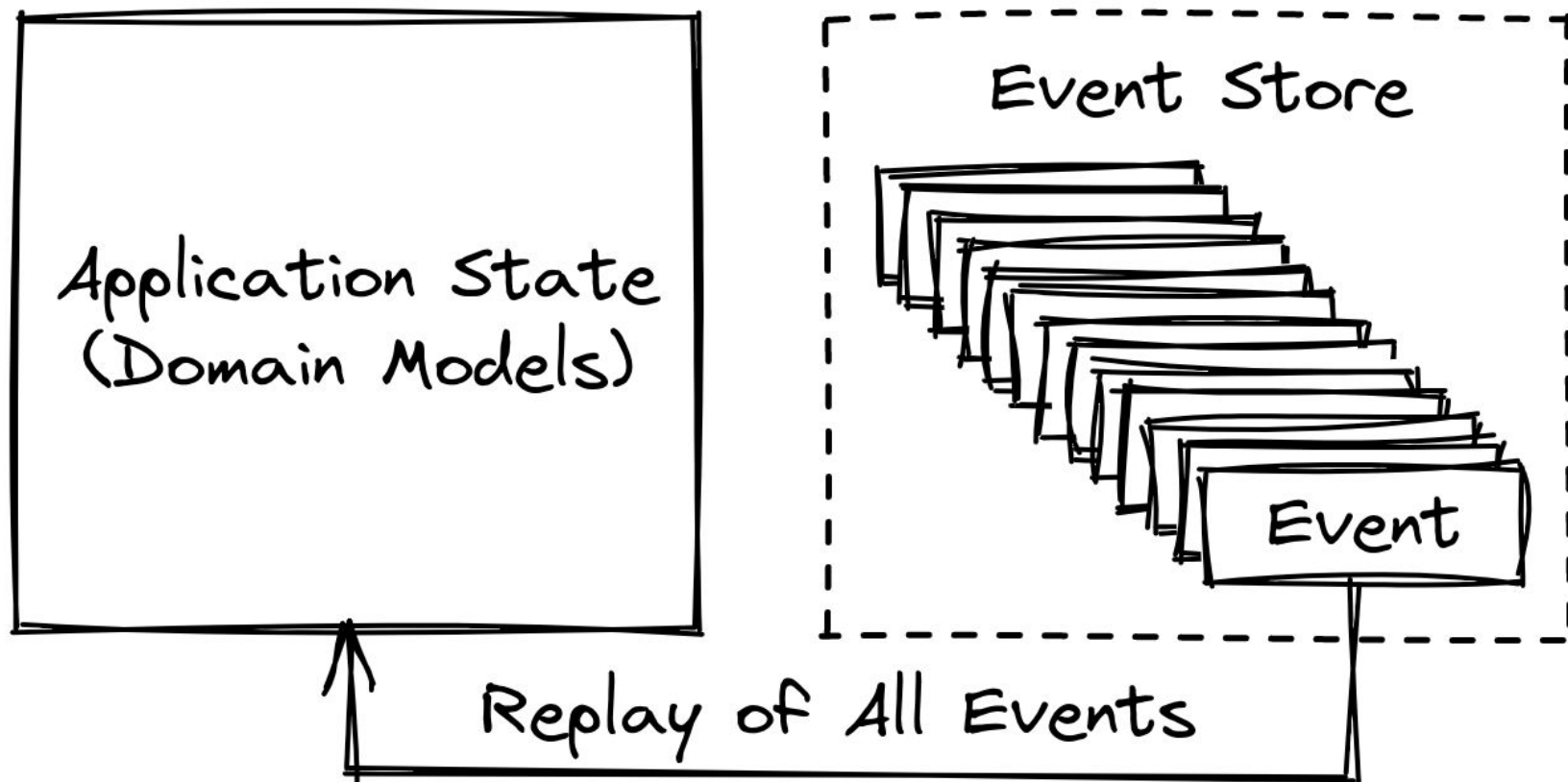


Преимущества Event Sourcing

- легко вносить изменения и развивать функционал
- есть история всех изменений и возможность их реинтерпретировать по-новому
- хорошо держит нагрузку вида (95% READ / 5% WRITE)
- быстрые и гибкие выборки данных
- легко масштабируется
- может быть географически распределено
- скорость и отзывчивость интерфейса



Легко вносить изменения и развивать функционал



Легко вносить изменения и развивать функционал



```
const events = [ { id: 'id-1', type: 'ORDER_CREATED' },  
                  { id: 'id-1', type: 'ORDER_APPROVED' },  
                  { id: 'id-1', type: 'ORDER_SHIPPED' } ]  
  
const currentState = events.reduce(  
  (state, event) => {  
    switch(event.type) {  
      case 'ORDER_CREATED':  
        state.activeOrders.add(event.id)  
        break  
      case 'ORDER_SHIPPED':  
        state.activeOrders.delete(event.id)  
        break  
    }  
    return state  
  },  
  { activeOrders: new Set() }  
)
```

Set(0) {}

{ id: 'id-1', type: 'ORDER_CREATED' }

Set(1) { 'id-1' }

{ id: 'id-1', type: 'ORDER_APPROVED' }

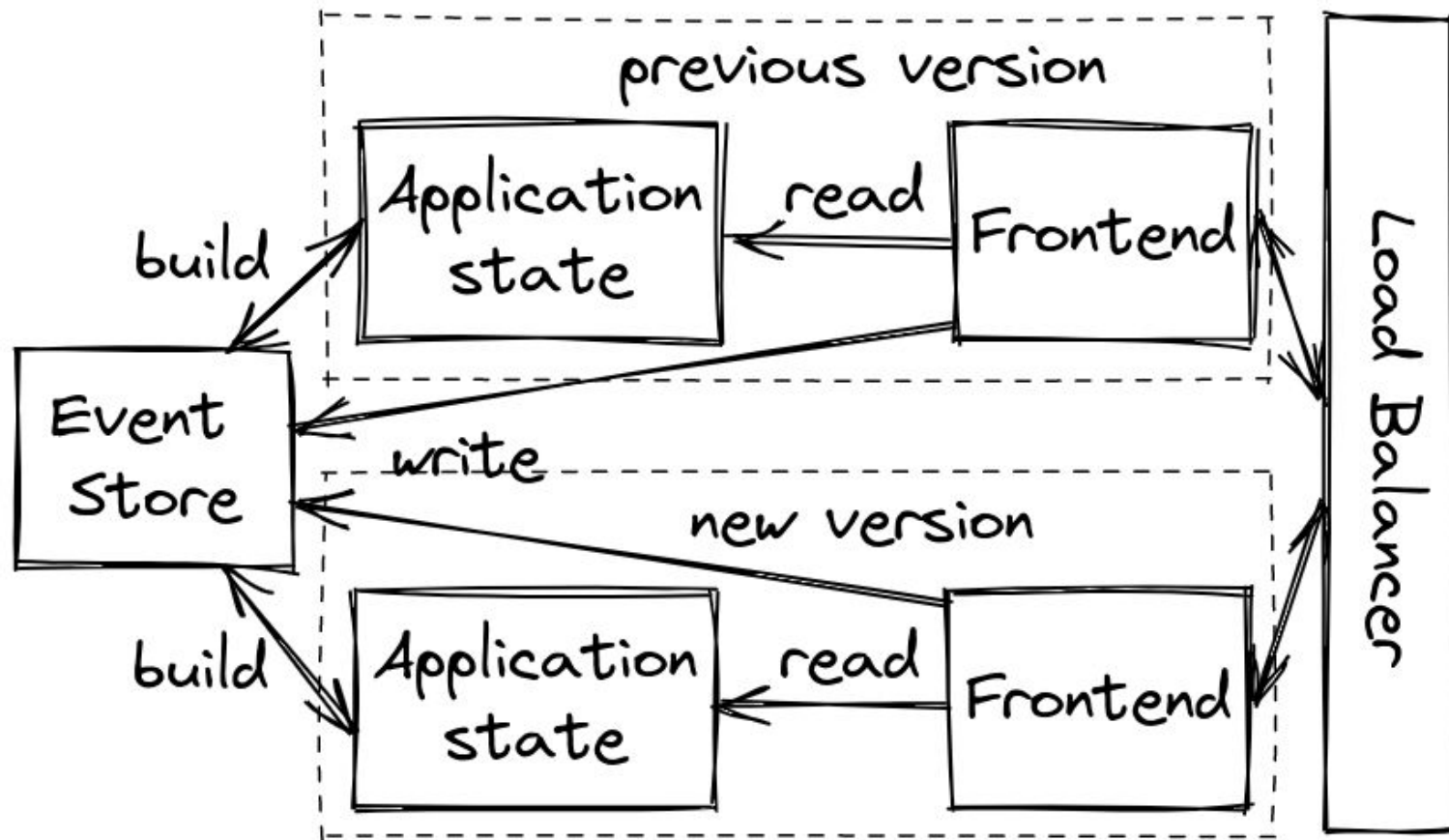
Set(1) { 'id-1' }

{ id: 'id-1', type: 'ORDER_SHIPPED' }

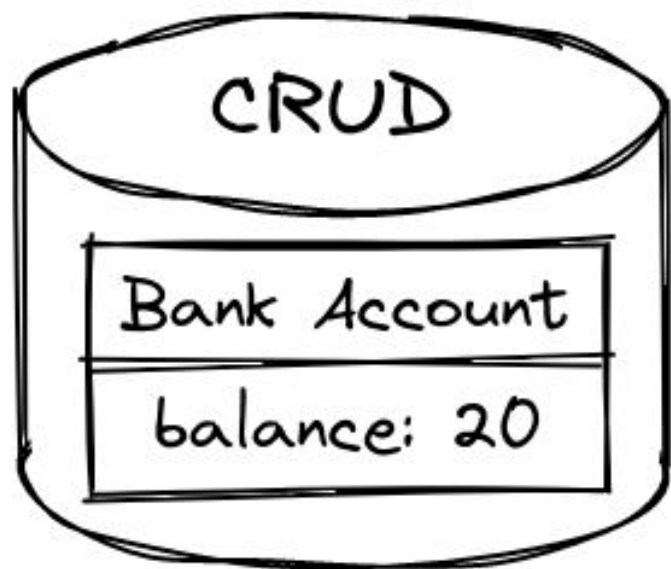
Set(0) {}



Легко вносить изменения и развивать функционал

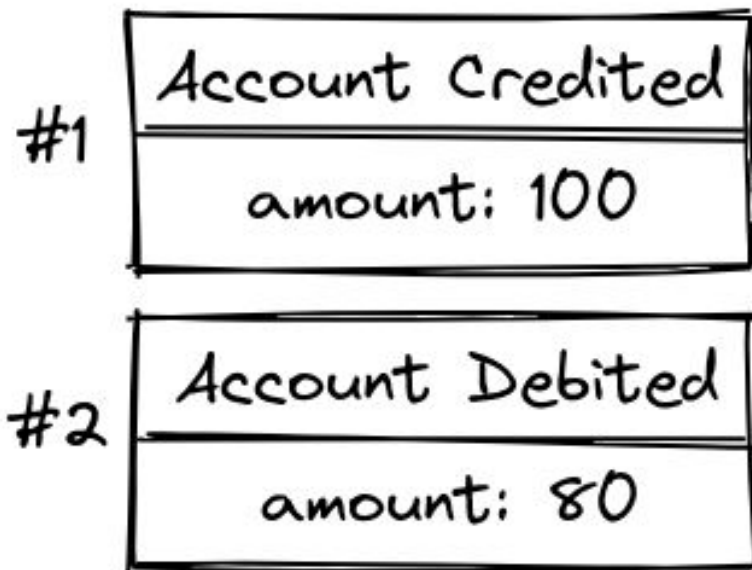


Есть история всех изменений



VS

Event Store

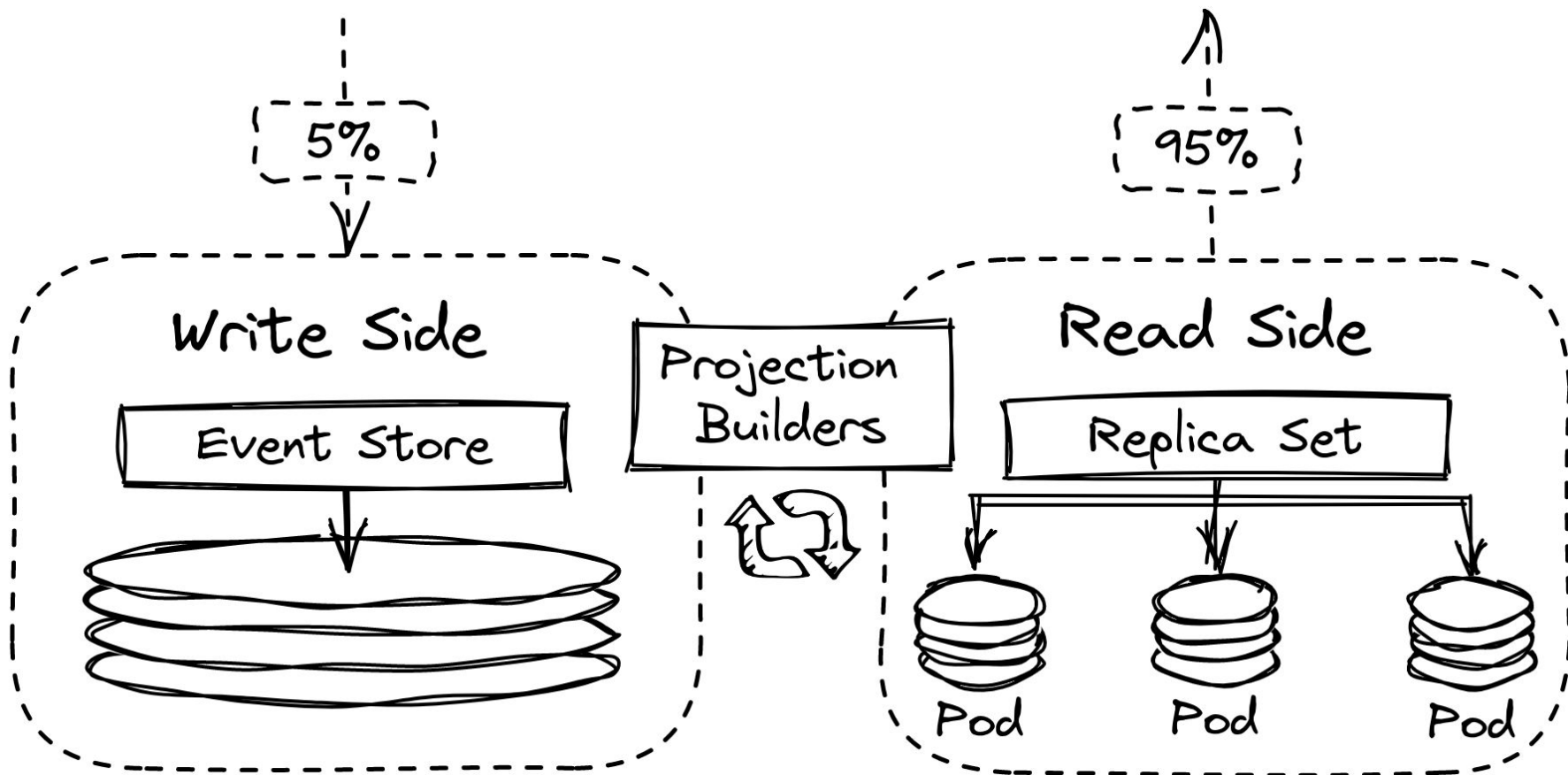


Преимущества Event Sourcing

- легко вносить изменения и развивать функционал ✓
- есть история всех изменений и возможность их реинтерпретировать по-новому ✓
- хорошо держит нагрузку вида (95% READ / 5% WRITE)
- быстрые и гибкие выборки данных
- легко масштабируется
- может быть географически распределено
- скорость и отзывчивость интерфейса



Нагрузка 95% READ / 5% WRITE

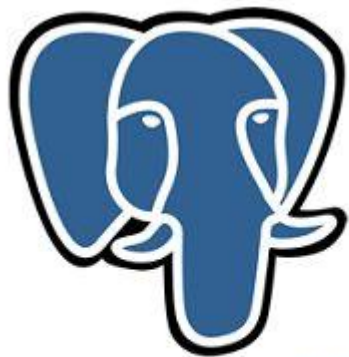


Преимущества Event Sourcing

- легко вносить изменения и развивать функционал ✓
- есть история всех изменений и возможность их реинтерпретировать по-новому ✓
- хорошо держит нагрузку вида (95% READ / 5% WRITE) ✓
- быстрые и гибкие выборки данных
- легко масштабируется
- может быть географически распределено
- скорость и отзывчивость интерфейса



Быстрые и гибкие выборки данных



PostgreSQL



ClickHouse



elasticsearch

Преимущества Event Sourcing

- легко вносить изменения и развивать функционал ✓
- есть история всех изменений и возможность их реинтерпретировать по-новому ✓
- хорошо держит нагрузку вида (95% READ / 5% WRITE) ✓
- быстрые и гибкие выборки данных ✓
- легко масштабируется
- может быть географически распределено
- скорость и отзывчивость интерфейса

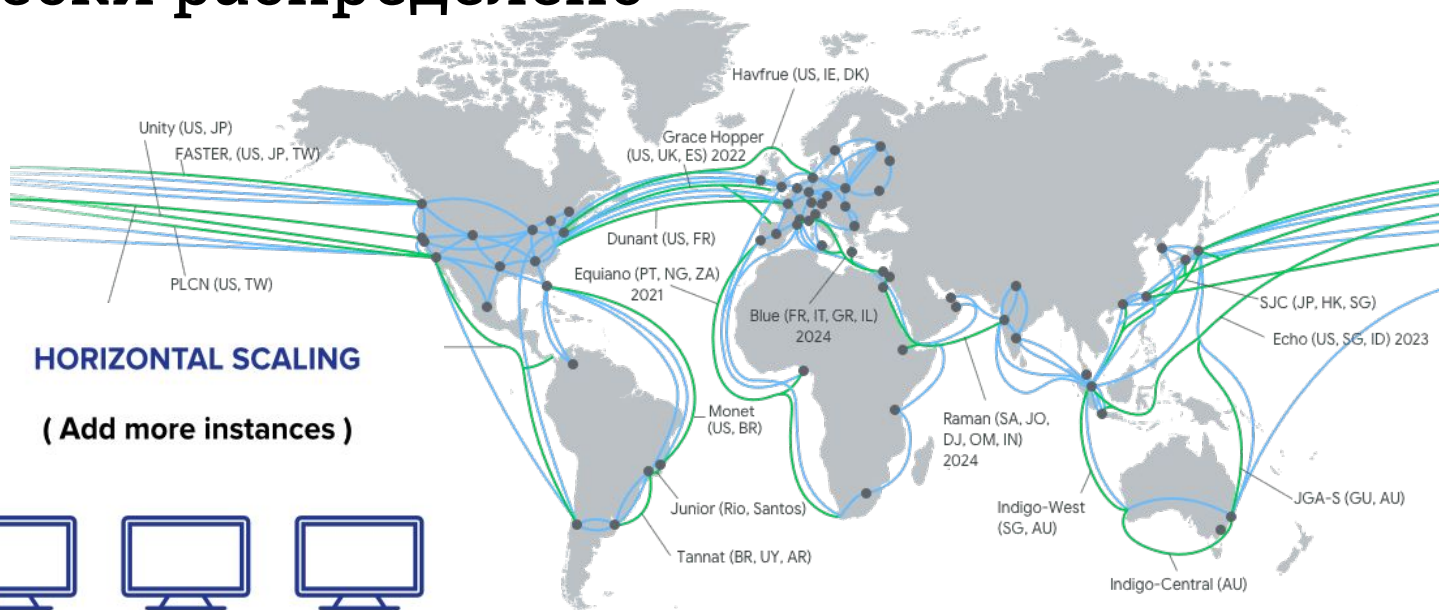


Легко масштабируется и может быть географически распределено

VERTICAL SCALING
Increase size of instance
(RAM, CPU etc.)



HORIZONTAL SCALING
(Add more instances)

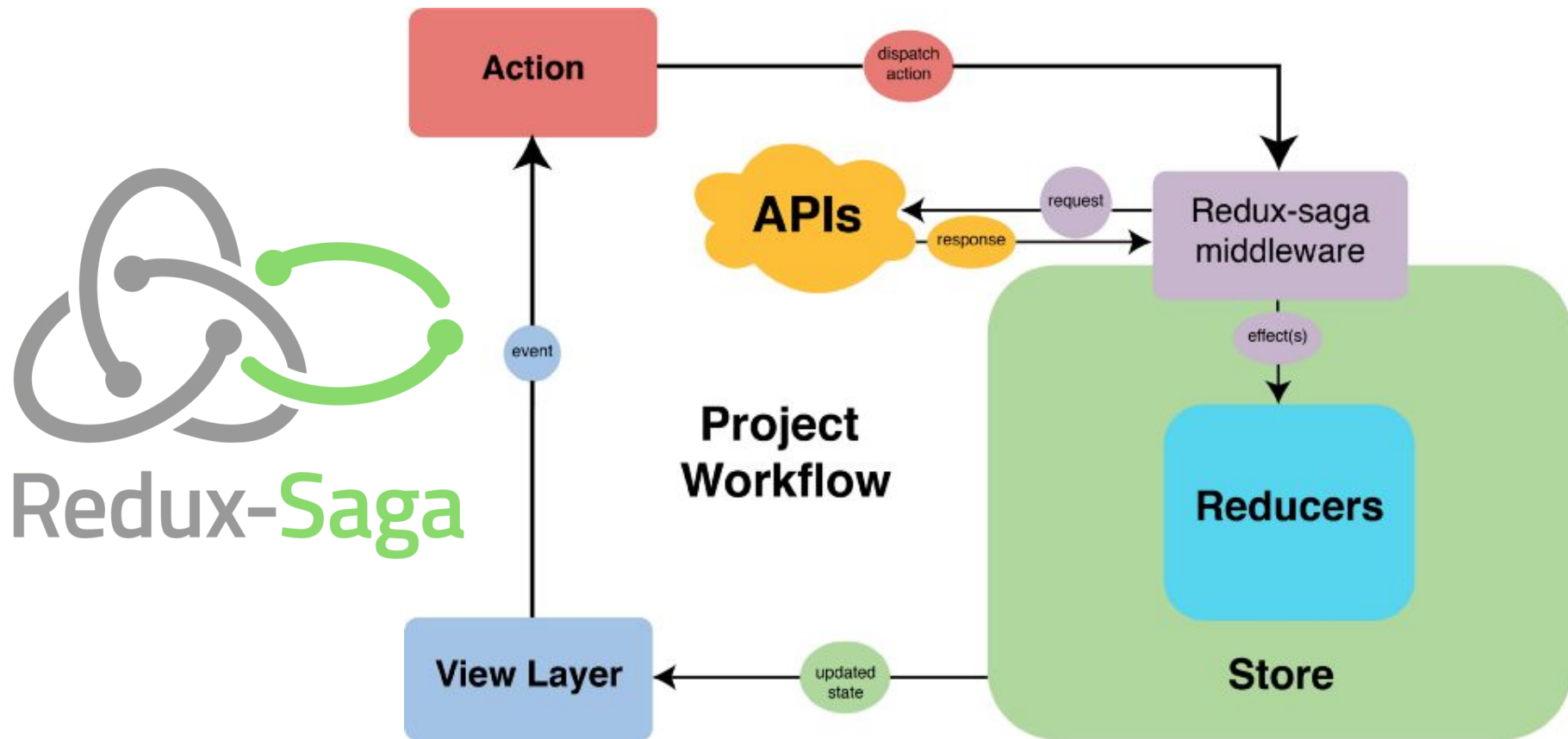


Преимущества Event Sourcing

- легко вносить изменения и развивать функционал ✓
- есть история всех изменений и возможность их реинтерпретировать по-новому ✓
- хорошо держит нагрузку вида (95% READ / 5% WRITE) ✓
- быстрые и гибкие выборки данных ✓
- легко масштабируется ✓
- может быть географически распределено ✓
- скорость и отзывчивость интерфейса



Скорость и отзывчивость интерфейса



Преимущества Event Sourcing

- легко вносить изменения и развивать функционал ✓
- есть история всех изменений и возможность их реинтерпретировать по-новому ✓
- хорошо держит нагрузку вида (95% READ / 5% WRITE) ✓
- быстрые и гибкие выборки данных ✓
- легко масштабируется ✓
- может быть географически распределено ✓
- скорость и отзывчивость интерфейса ✓



Роль Грега Янга и Мартина Фаулера в теме EventSourcing



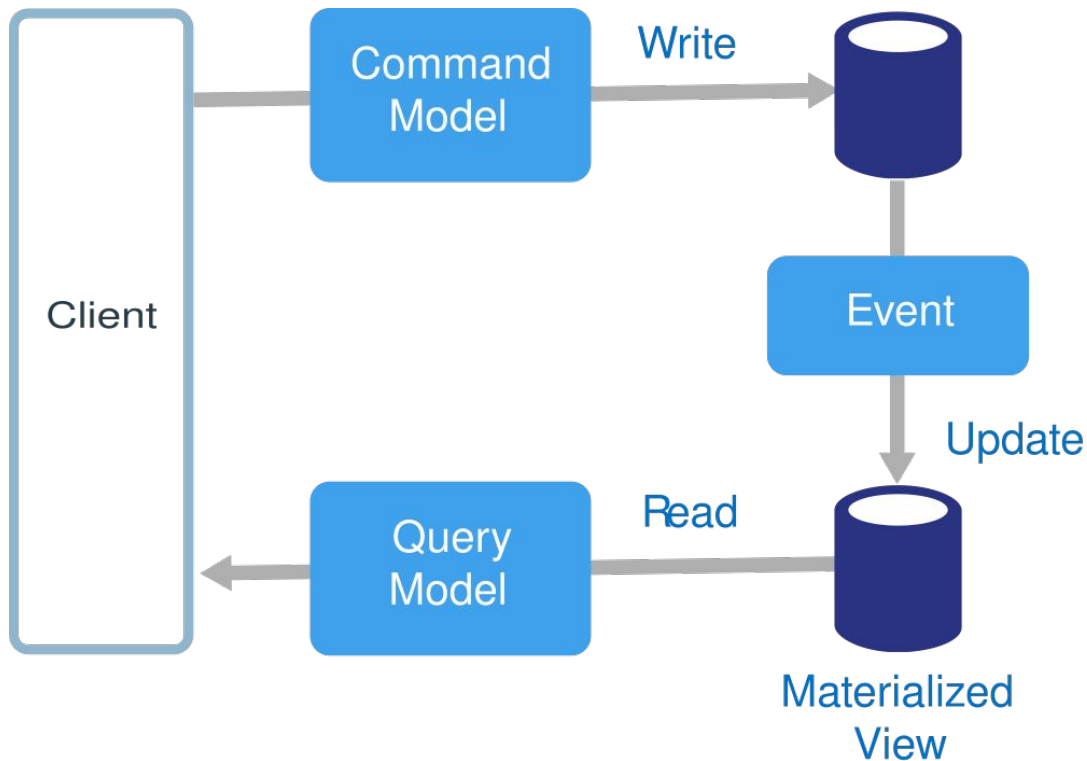
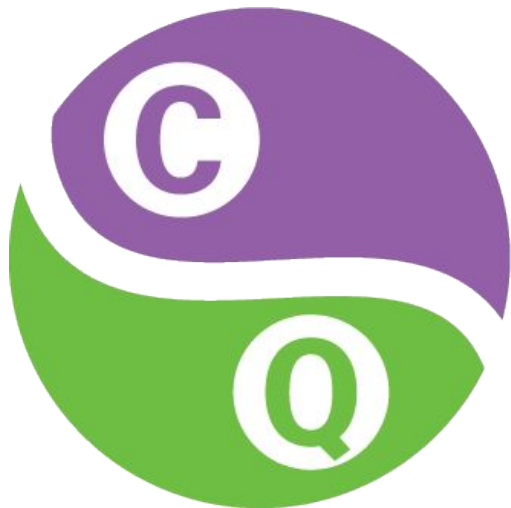
Роль Грегга Янга и Мартина Фаулера в теме
EventSourcing

EventSourcing = CQRS = DDD

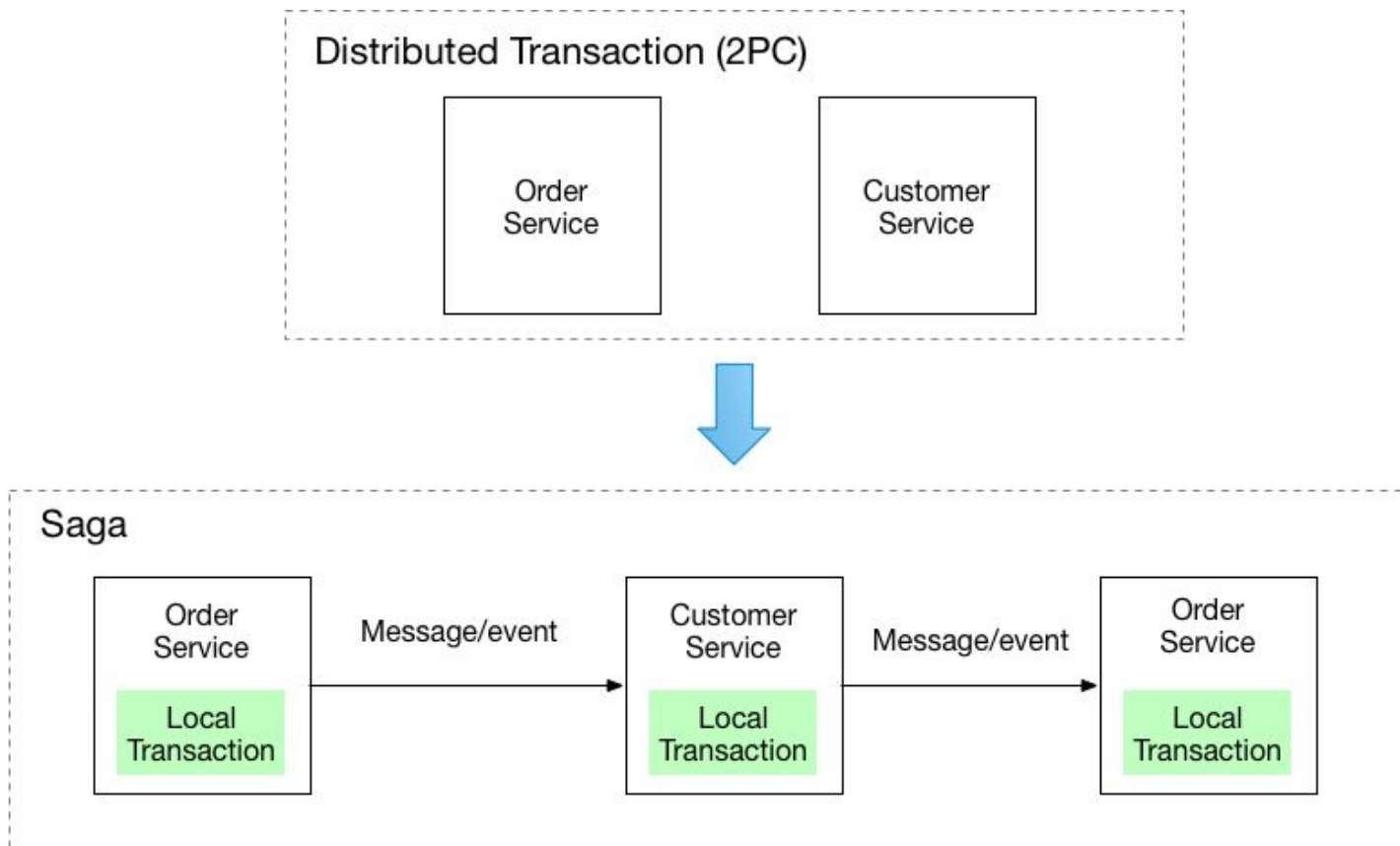
*** это не так!**

Что такое CQRS

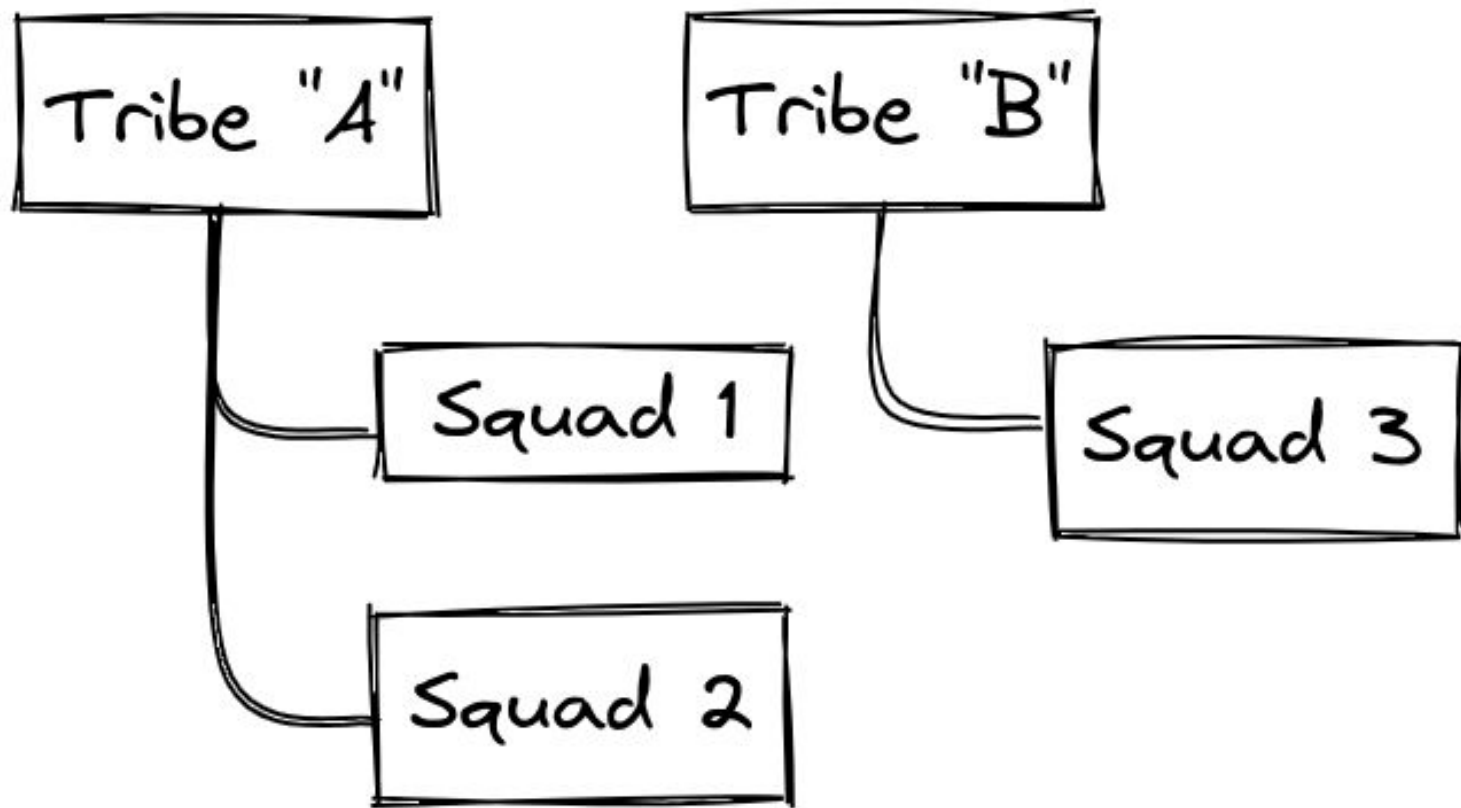
CQRS
Command Query Responsibility Segregation



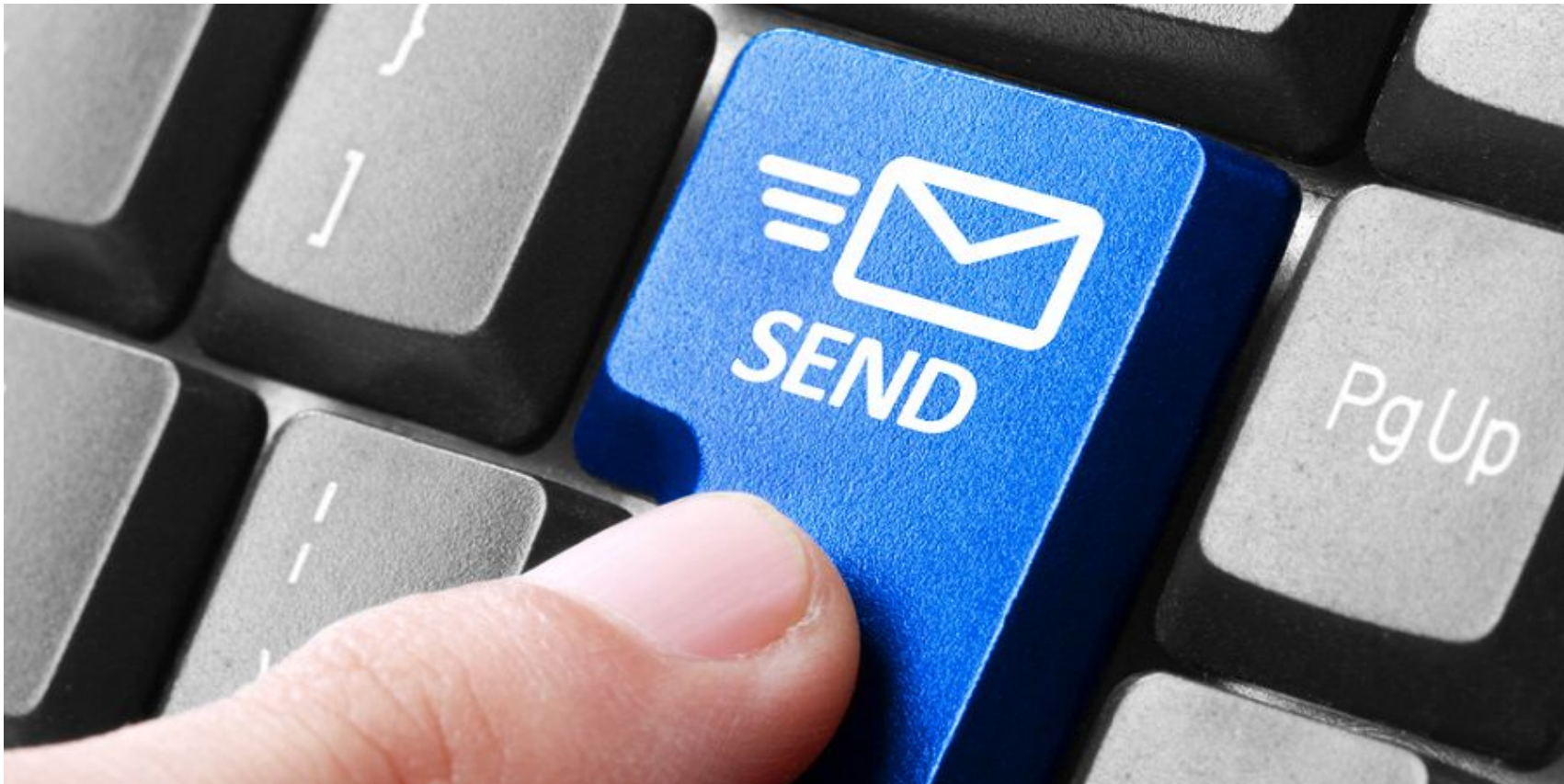
Что такое CQRS. Сага



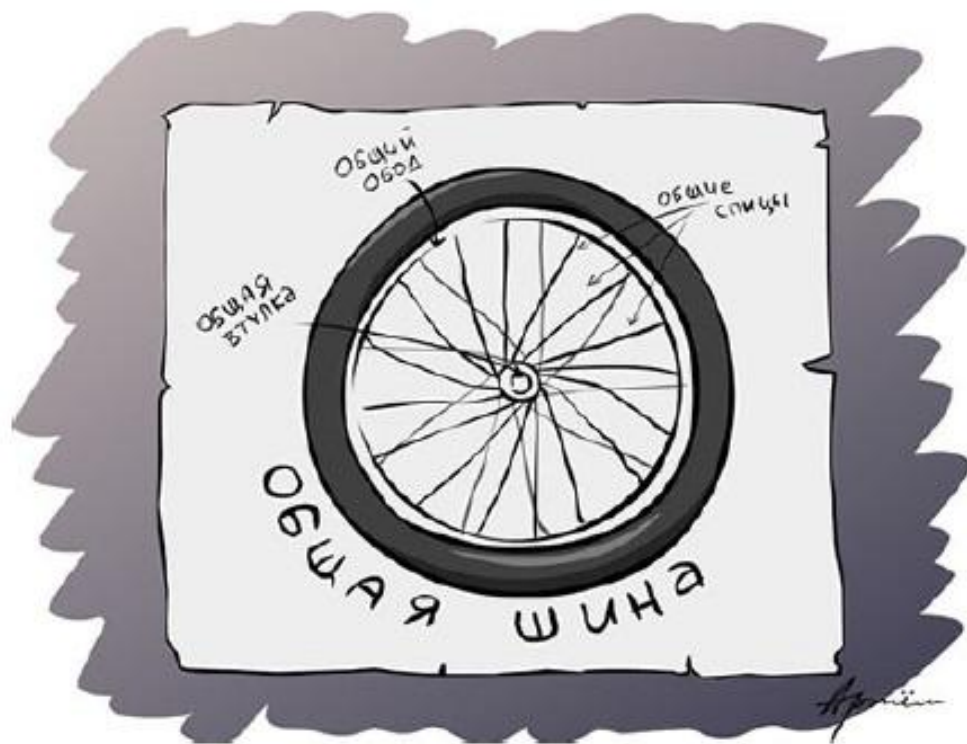
Что такое CQRS. Сага



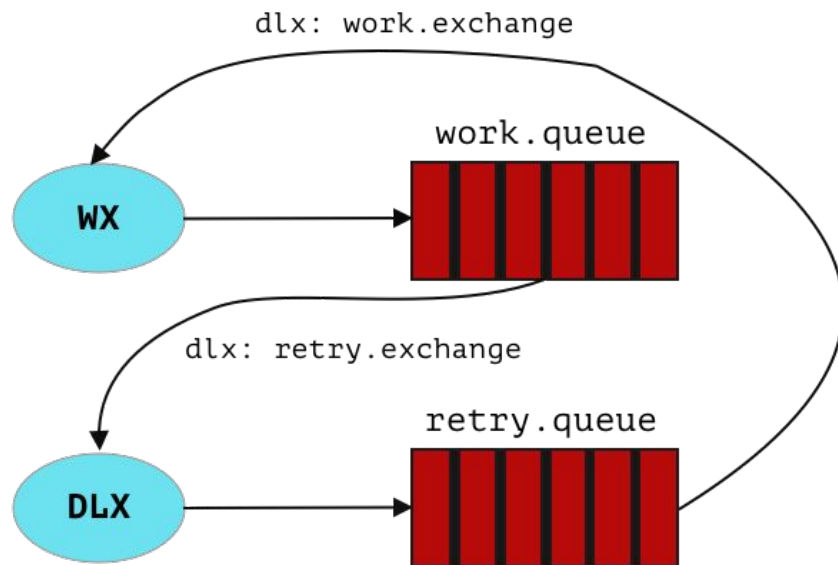
Что такое CQRS. Сага



Что такое CQRS. Шина



dlx = x-dead-letter-exchange

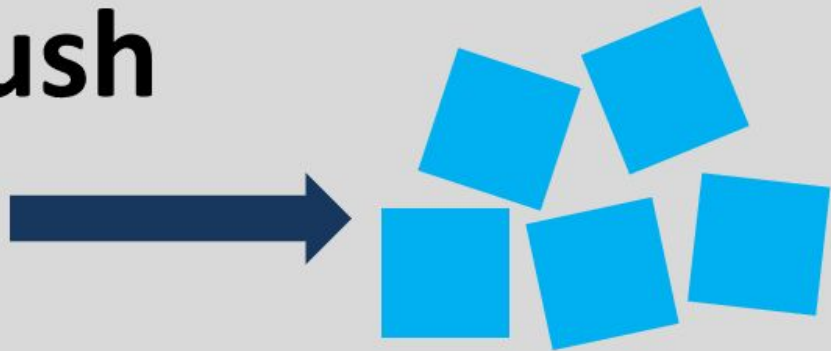


Роль Грега Янга и Мартина Фаулера в теме EventSourcing

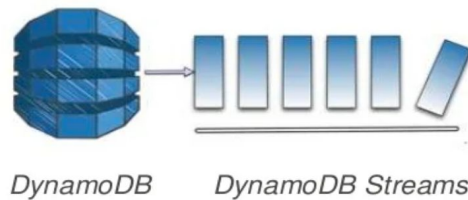


Стратегия доставки событий

Push



Pull



Pull-стратегия сборки и Ledger / DocumentVersion



Событие / Event

```
{  
    streams: Array<{entityName: String, entityId:  
String, entityVersion: BigInt}>,  
    tags: Record<String, any>,  
    timestamp: Date,  
    type: String,  
    payload: Object  
}
```

*** entityVersion - монотонно растущая, без дыр**

Виды проекций

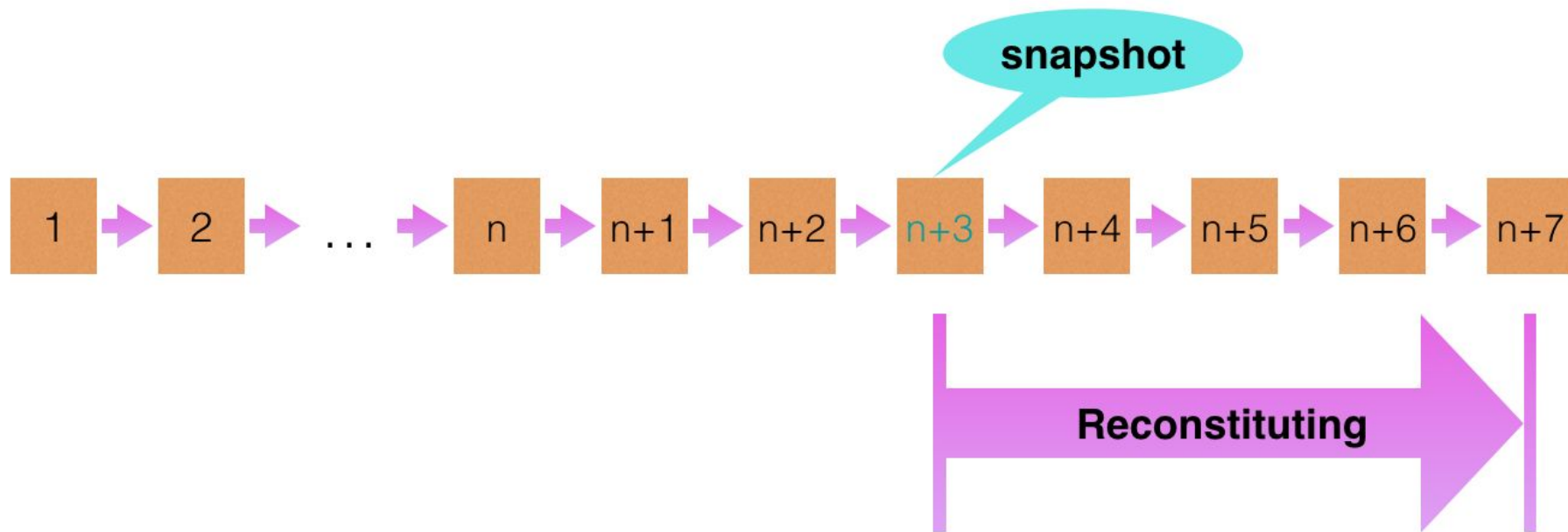
- Строгая / Нестрогая
- Синхронная / Асинхронная
- Реактивная / Ленивая
- На уровне документа / коллекции /
схемы / БД

Что хранит EventStore

- События
- Методы информации и правила её формирования
- Файлы
- Ключи шифрования

*** GDPR полиция следит за тобой**

Проекции и снимоты



Версионирование Event-ов

version 1

```
{  
  type: 'USER_CREATED',  
  firstName: String  
}
```

version 2

```
{  
  type: 'USER_CREATED',  
  firstName: String,  
  lastName?: String  
}
```

Приемлемо

version 1

```
{  
  type: 'USER_CREATED',  
  name: String  
}
```

version 2

```
{  
  type: 'USER_CREATED_V2',  
  firstName: String,  
  lastName: String  
}
```

Общий случай

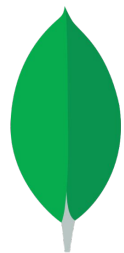
Выбор БД для EventStore



YDB



Azure
Cosmos DB



mongoDB®



DynamoDB



DynamoDB Streams

Что посмотреть/почитать

- Event Sourcing Greg Young <https://www.youtube.com/watch?v=8JKjvY4etTY>
- Pragmatic Event-Driven Microservices Allard Buijze
https://www.youtube.com/watch?v=vSd_0zGxslU&t=2s
- <https://ddd-cqrs-es.slack.com>
- Hands-On Domain-Driven Design with .NET Core: Tackling complexity in the heart of software by putting DDD principles into practice
- Versioning in an Event Sourced System (<https://leanpub.com/esversioning>) • The Dark Side of Event Sourcing: Managing Data Conversion • Designing Data-Intensive Applications (Kleppmann)
- <https://github.com/RBMHTechnology/eventuate>