

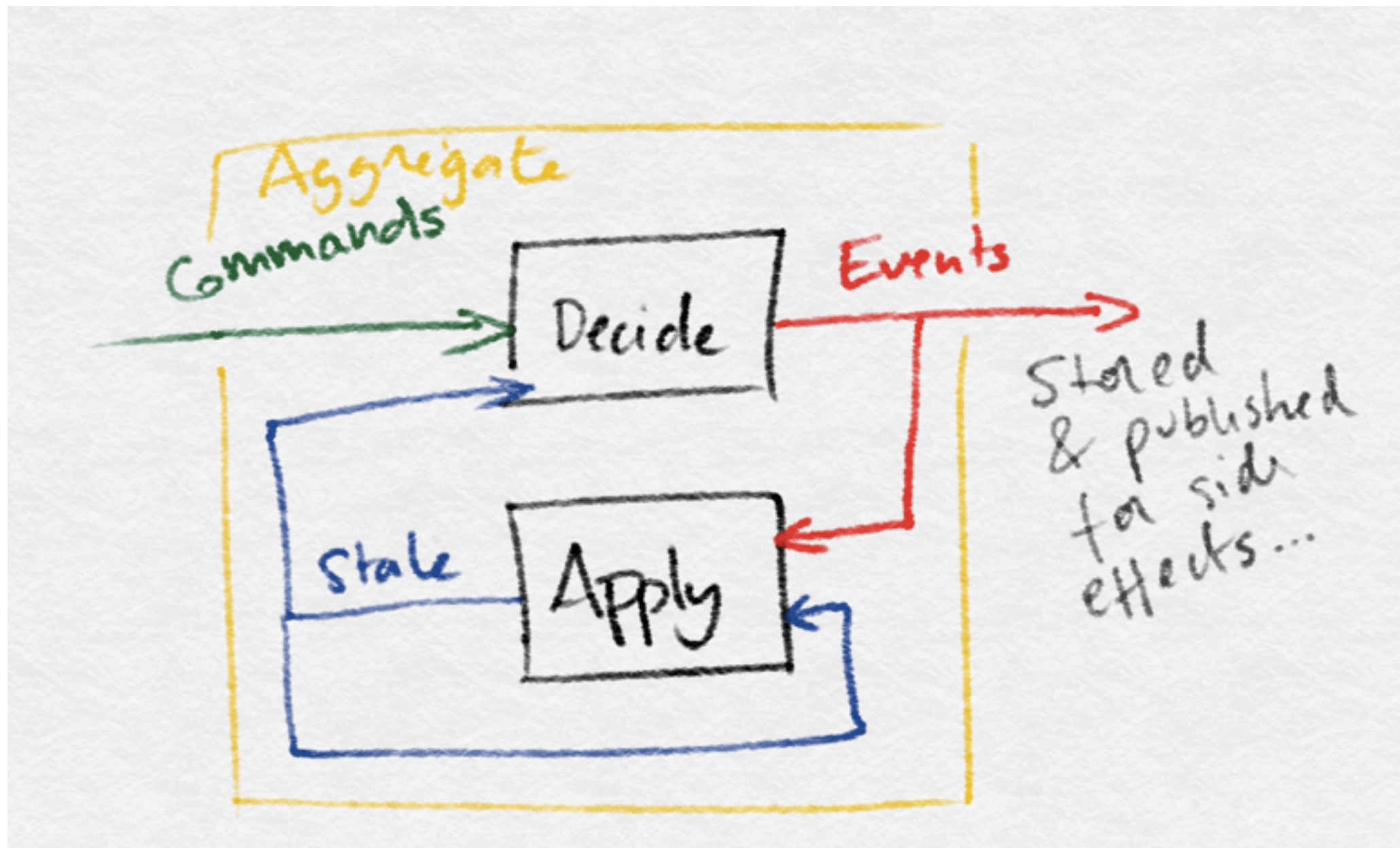
Event-Driven

Лайфхаки

Ринат Абдуллин

Intro

Паттерны последних проектов

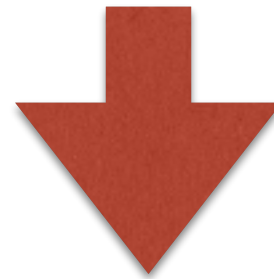


Кто знает про Event Sourcing?

```
update member.info  
set balance=money(4100)  
where id=2;
```

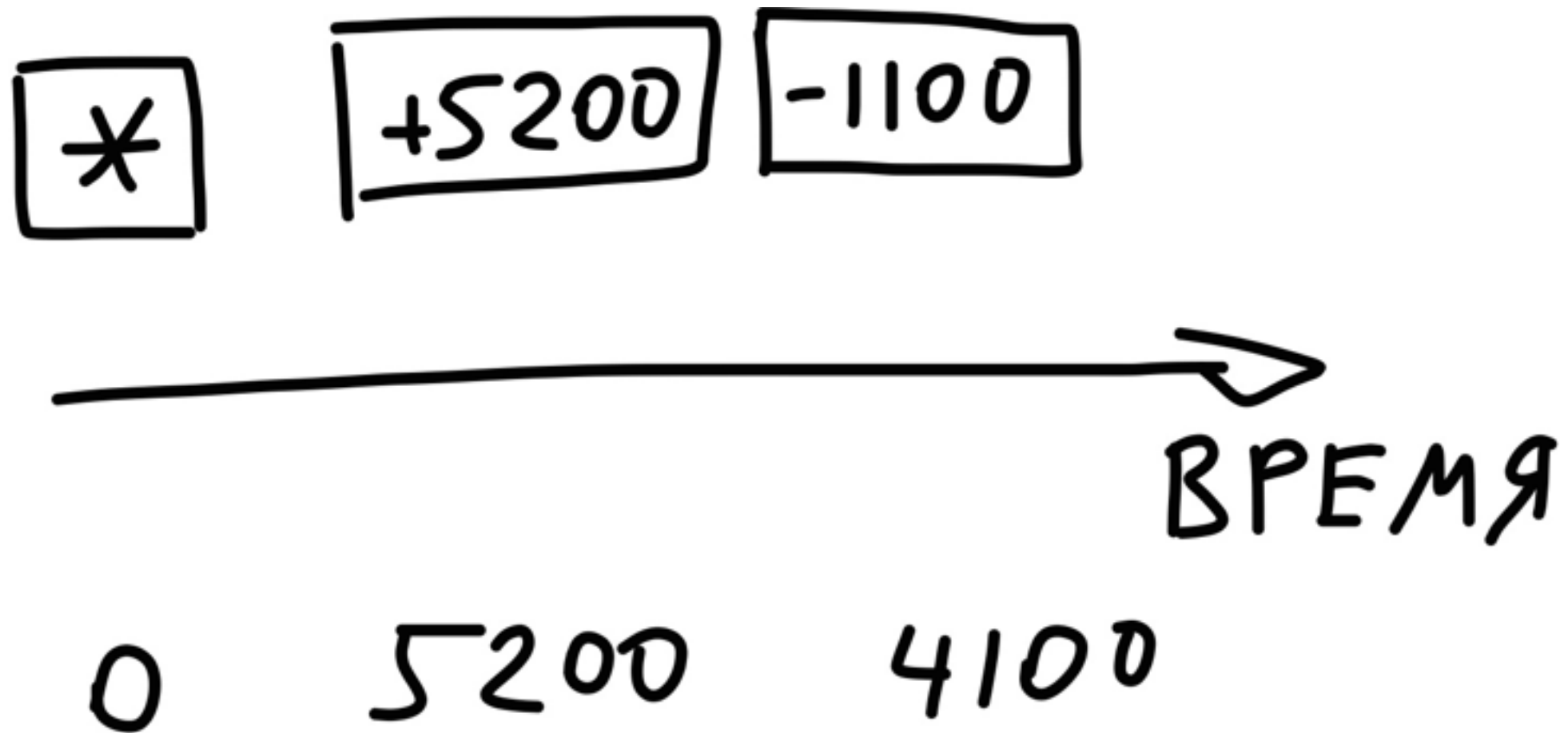
```
-- Query returned successfully: one row affected, 88 ms execution time.
```

	id [PK] serial	name text	balance money	bio text
1	1	Bob	\$3,922.00	Likes to travel
2	2	Mary	\$5,200.00	Likes to watch movies
3	3	Dave	\$404.00	Codes like maniac



	id [PK] serial	name text	balance money	bio text
1	1	Bob	\$3,922.00	Likes to travel
2	2	Mary	\$4,100.00	Likes to watch movies
3	3	Dave	\$404.00	Codes like maniac

\$5,200.00 Потерины навсегда



История событий

Лог Транзакций

У бухгалтеров
такое есть



Event Handler

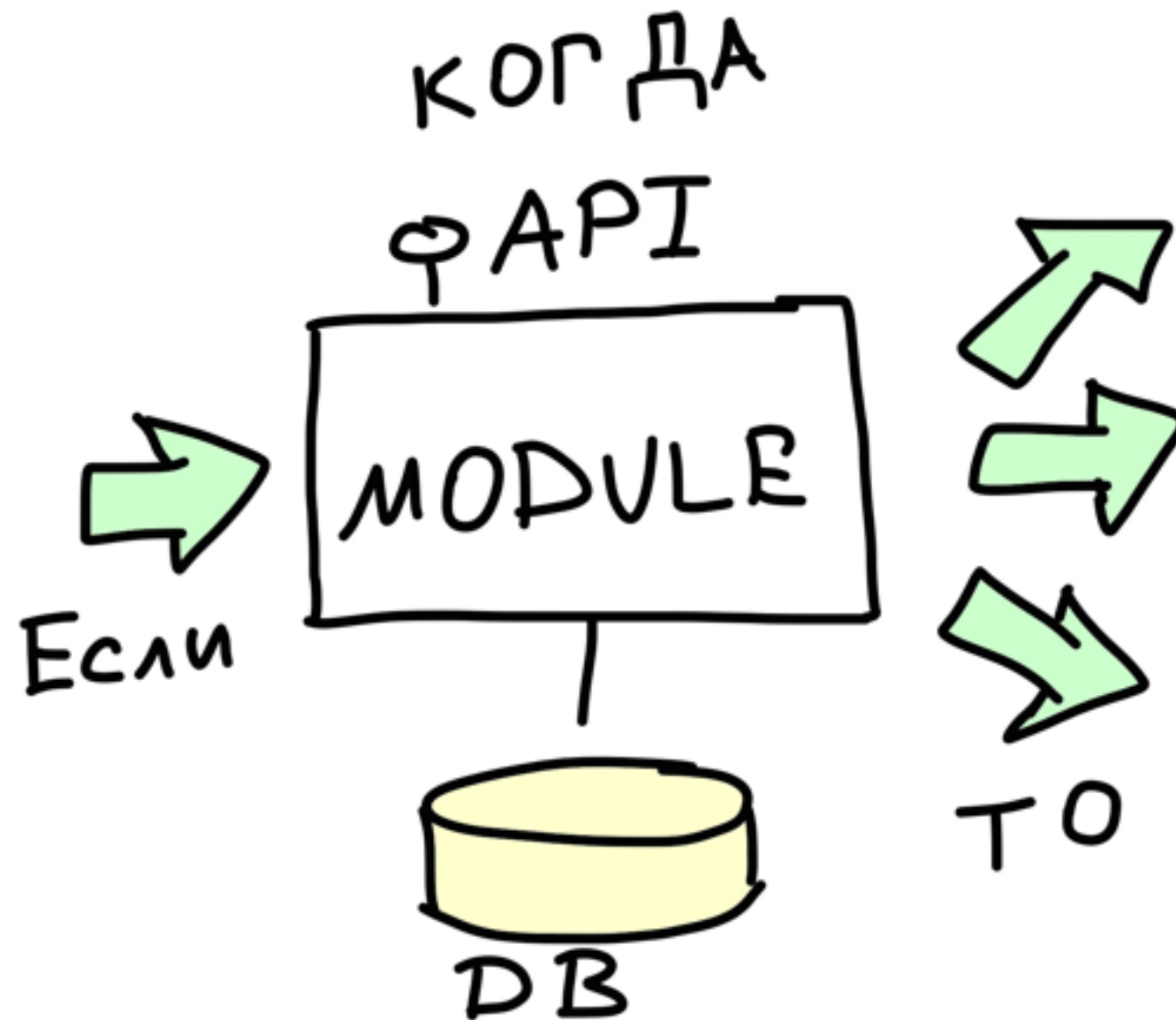
```
public delegate void MyEventHandler(string foo);  
  
public event MyEventHandler SomethingHappened;  
  
void HandleSomethingHappened(string foo)  
{  
    //Do some stuff  
}  
  
myObj.SomethingHappened += new MyEventHandler(HandleSomethingHappened);
```



```
1 type PortraitApproved struct {  
2     Id          Id  
3     MemberId    Id  
4     MediaRef    *MediaRef  
5 }
```

```
1 {  
2     "Id": "137f83d88b21c43d0000004400886534",  
3     "MemberId": "1371c6b236de5f8000000000a000c2907",  
4     "MediaRef": {  
5         "Id": "1371c6b236de5f8000000000a000c2907",  
6         "Ext": ".jpg"  
7     }  
8 }
```

BDD



BDD

Если-Когда-То

ЕСЛИ · ПОЛЬЗОВАТЕЛЬ СОЗДАН
· ПОЛЬЗОВАТЕЛЮ КИНУЛИ 5200

КОГДА · СНИМАЕМ ПЛАТУ ЗА ИНЕТ

ТО · У ПОЛЬЗОВАТЕЛЯ СНИЛОСЬ
1100 И СТАЛО 4100

Сценарий

```
1 func (x *context) TestGivenLoveStoryWhenBobChecksNews(c *C) {  
2     s := x.given_bob_and_nancy_love_story()  
3  
4     r := x.GetJson(s.bobId, "/news")  
5     c.Assert(r.Code, Equals, http.StatusOK)  
6  
7     var m model  
8     r.Unmarshal(&m)  
9     c.Check(m.Title, Equals, "News")  
10    c.Check(m.HasMore, Equals, false)  
11  
12    c.Assert(m.Items, NotNil)    // shouldn't be nil  
13    c.Assert(m.Items, HasLen, 1) // bob should see nancy in news  
14}  
15
```

BDD Test in Go

```
public void given_existing_customer_with_balance()
{
    Given = new IEvent[]
    {
        new CustomerCreated {
            Id = new CustomerId(2),
            Name = "Microsoft",
            Currency = Currency.Eur
        },
        new CustomerPaymentAdded {
            Id = new CustomerId(2),
            NewBalance = 1000m.Eur(),
            Payment = 1000m.Eur(),
            PaymentName = "Bonus",
            Transaction = 1
        }
    };
    When = c => c.Charge("Sales forecast fee", 200m.Eur(), new DateTime(2012, 3, 2));
    Then = new IEvent[]
    {
        new CustomerChargeAdded {
            Charge = 200m.Eur(),
            ChargeName = "Sales forecast fee",
            Id = new CustomerId(2),
            NewBalance = 800m.Eur(),
            TimeUtc = new DateTime(2012, 3, 2),
            Transaction = 2
        }
    };
}
```

Преимущества

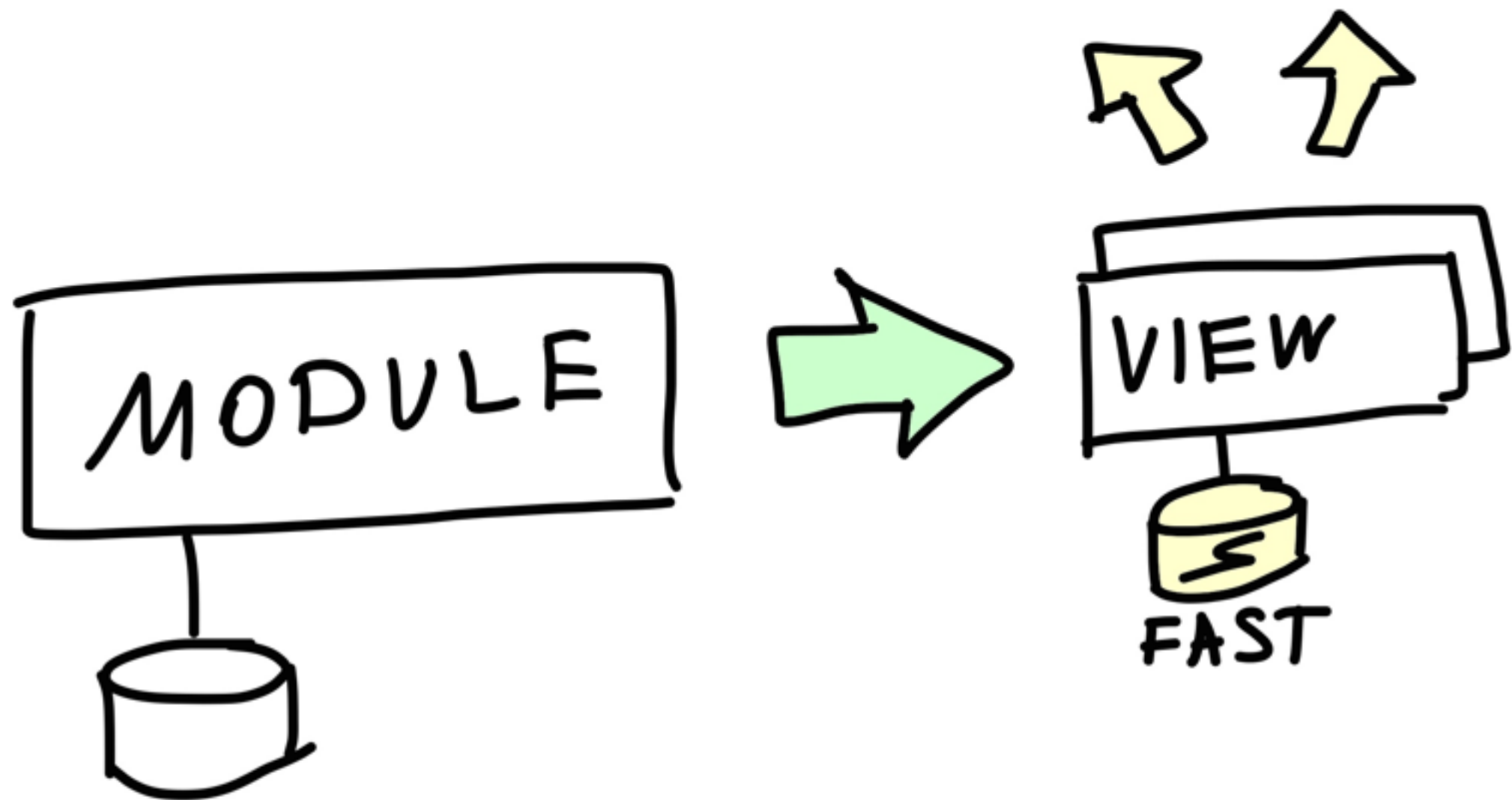
- Сценарии всегда актуальны
- Сценарии могут “сочинять” бизнес-люди
- Non-fragile код

Пусть код
документирует себя

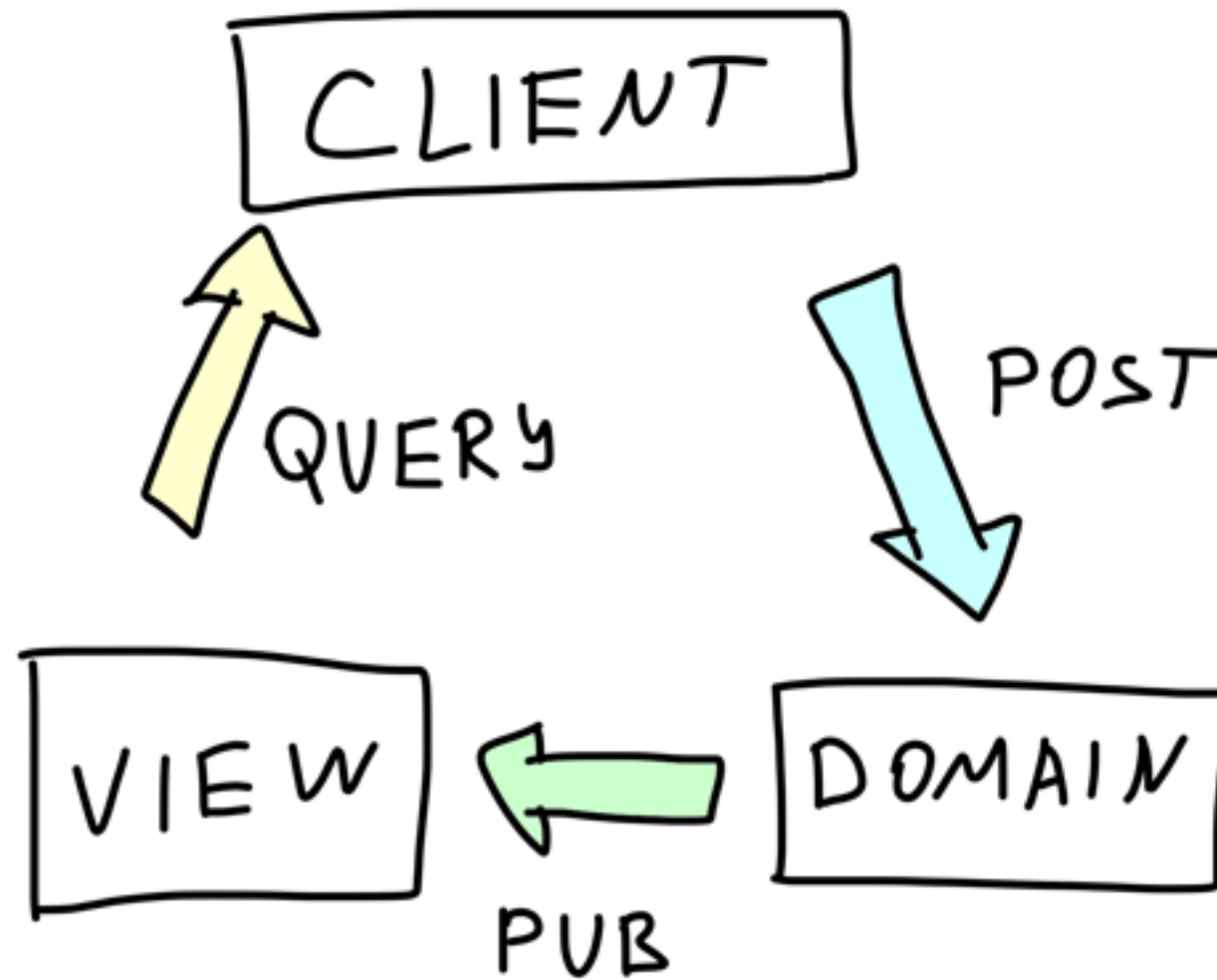

```
1 *Given*: Customer Microsoft created with Eur
2 *Given*: Added 'Bonus' 1000 EUR | Tx 1 => 1000 EUR
3
4 *When*: c => c.Charge("Sales forecast fee", 200.Eur(), Date(2012, 3, 2))
5
6 *Expect*: Charged 'Sales forecast fee' 200 EUR | Tx 2 => 800 EUR
7
```

CQRS

Простое масштабирование



InMem + Multiplex

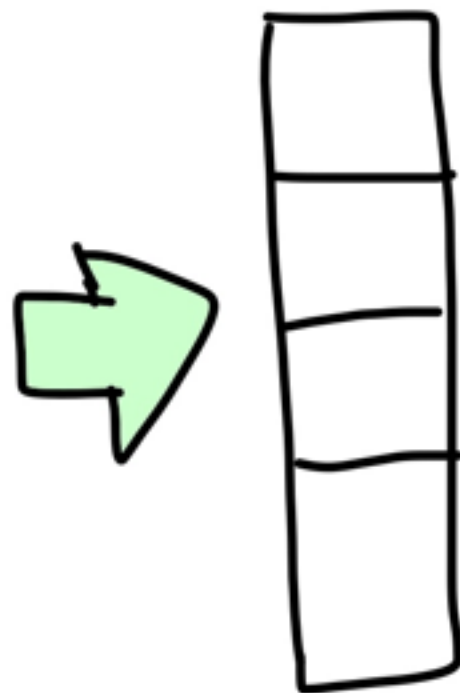


BIG CQRS

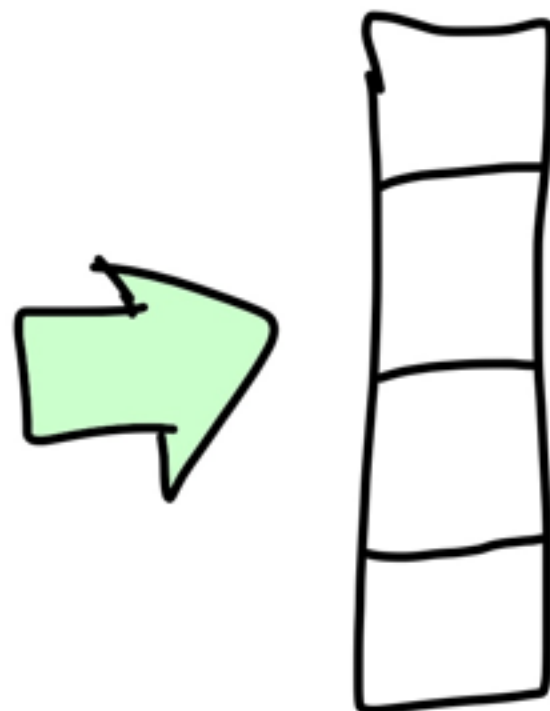
SEDA

Staged-Event-Driven Architecture

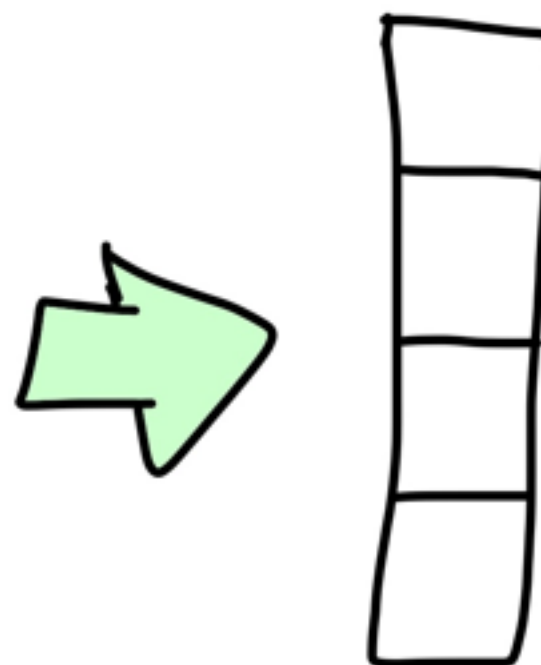
REQUEST



STORE



VIEW



SEDA

Event Storming



Моделирование Событий

Вопросы

@abdullin