

XII международная конференция
CEE-SECR / РАЗРАБОТКА ПО

28 - 29 октября, Москва



Apache Storm



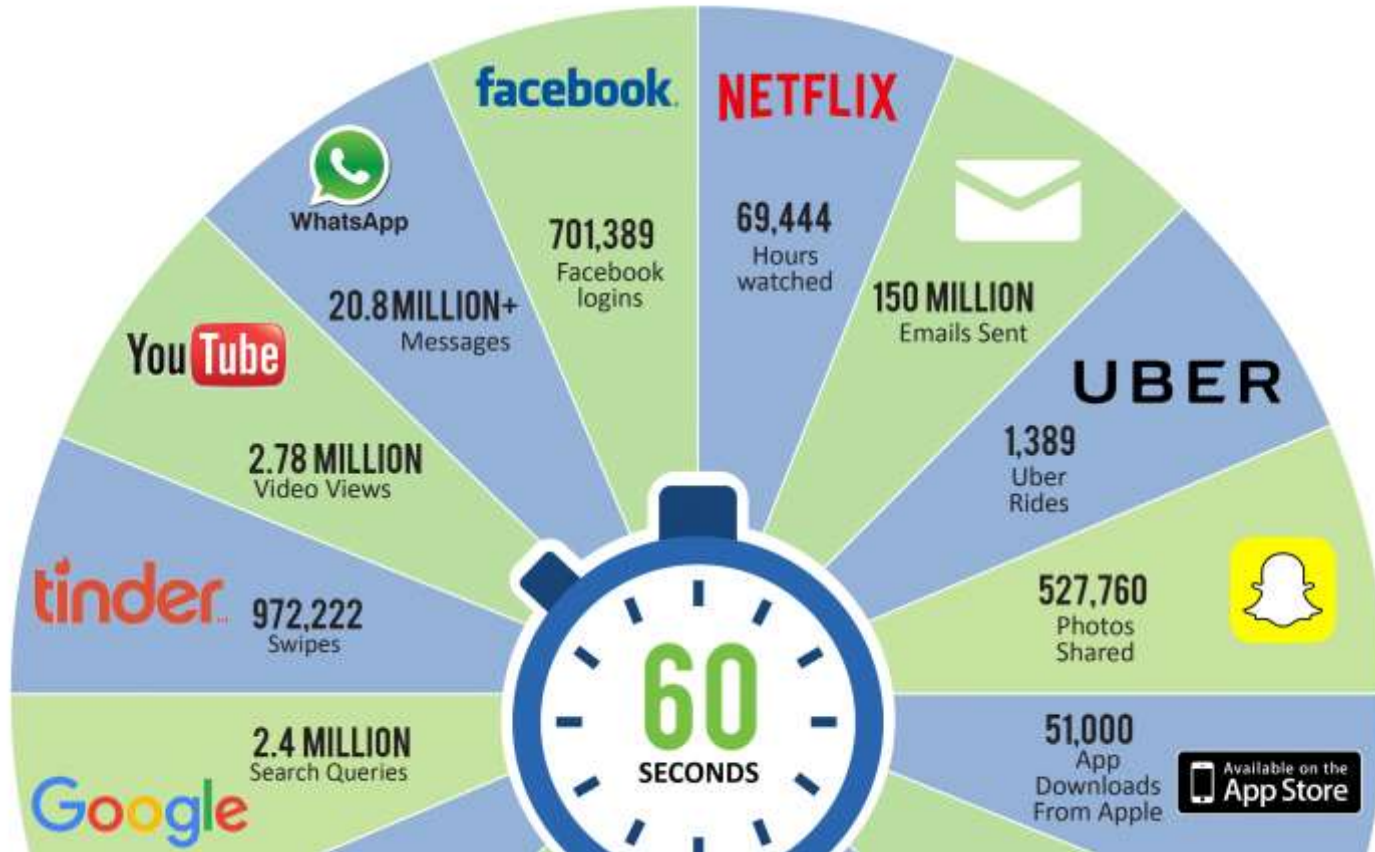
Kirill Shirokov

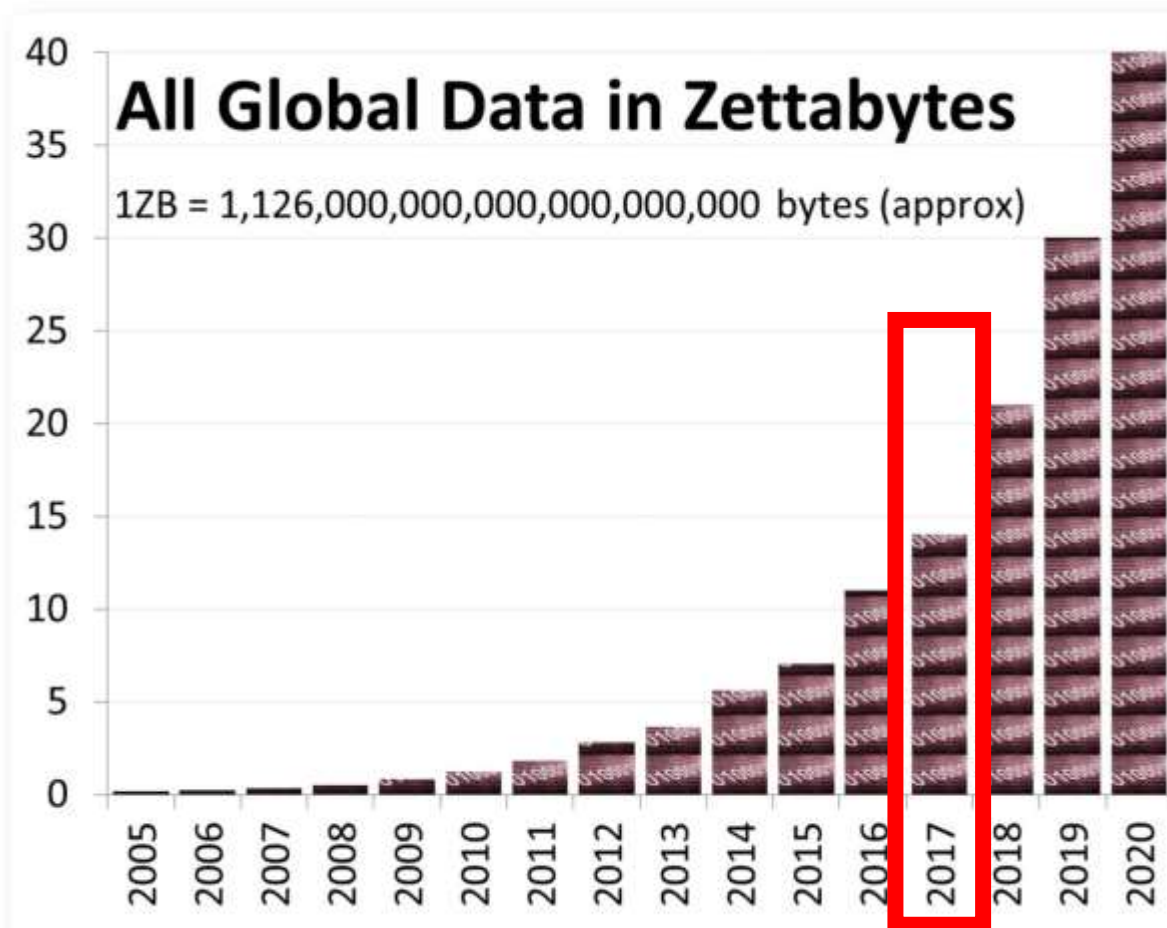
Application architect



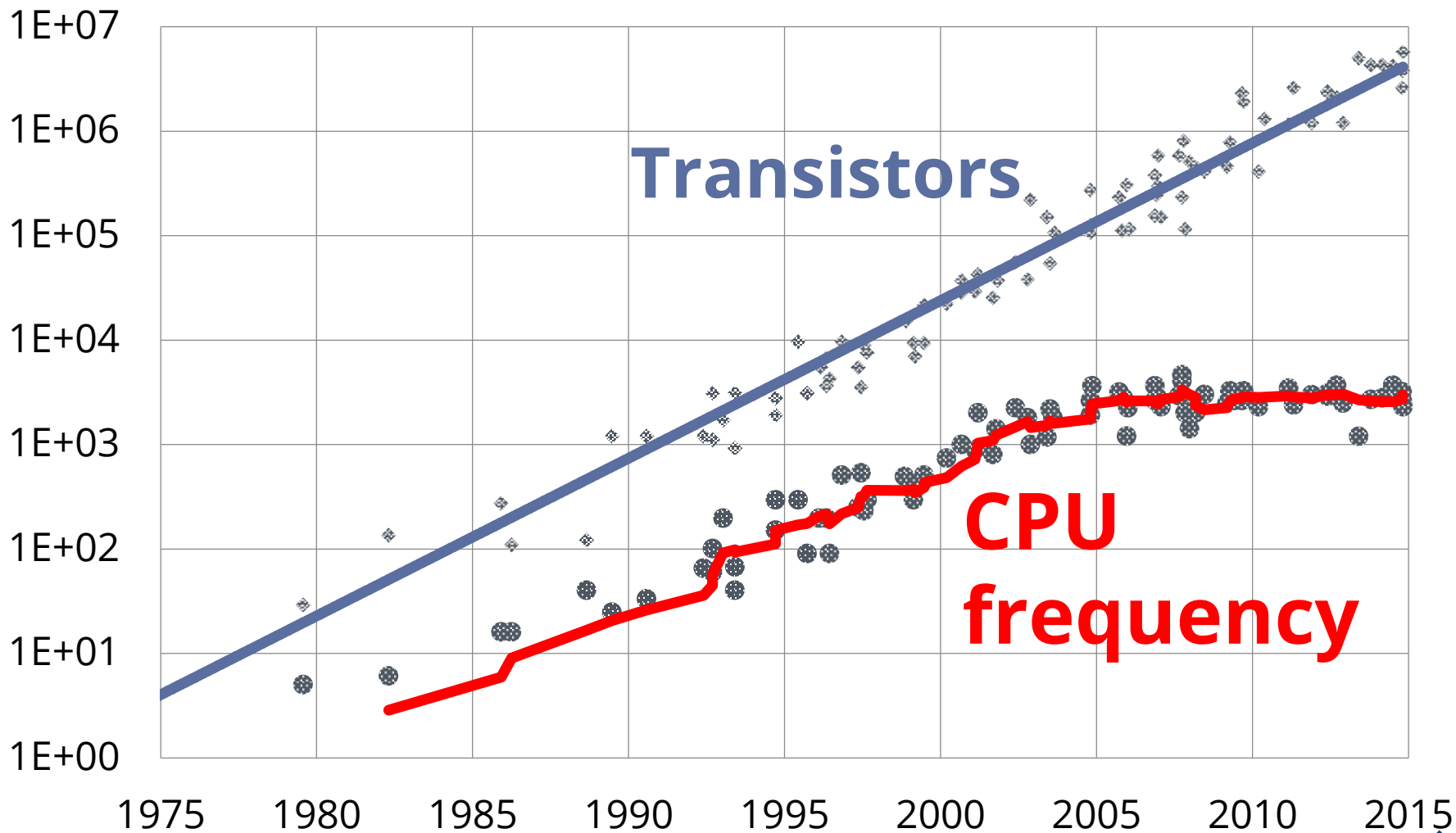
2016 What happens in an INTERNET MINUTE?

<http://www.excelacom.com/resources/blog/2016-update-what-happens-in-one-internet-minute>





<http://www1.unece.org/stat/platform/display/msis/Big+Data>



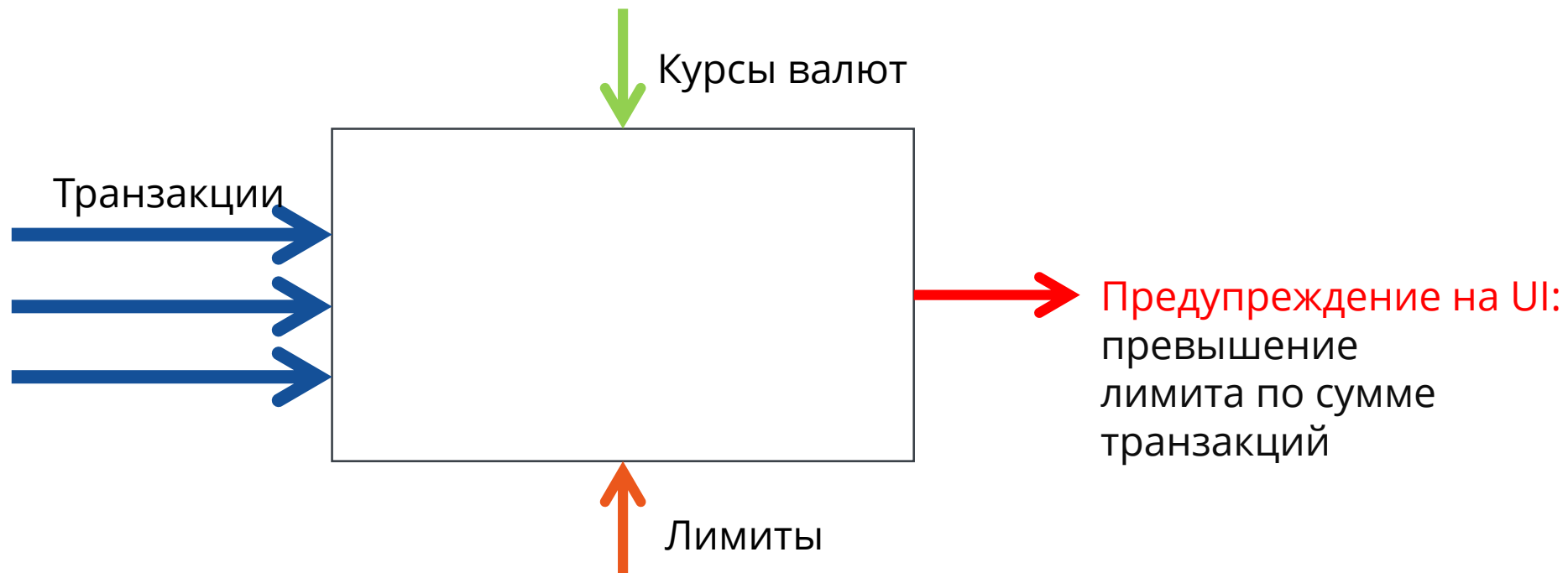






**А тем временем
в одном банке...**







0.5 сек

Курсы валют

Транзакции



Лимиты

Предупреждение на UI:
превышение
лимита по сумме
транзакций

1,000,000
документов/час



5 документов/сутки

Технология?

Технология?

- Единая для разных проектов (support)

Технология?

- Единая для разных проектов (support)
- Сложная бизнес-логика

Технология?

- Единая для разных проектов (support)
- Сложная бизнес-логика
- Жесткие NFR

Технология?

- Единая для разных проектов (support)
- Сложная бизнес-логика
- Жесткие NFR
- Низкий порог вхождения

Технология?

- Единая для разных проектов (support)
- Сложная бизнес-логика
- Жесткие NFR
- Низкий порог вхождения
- Мониторинг и настройка

Технология?

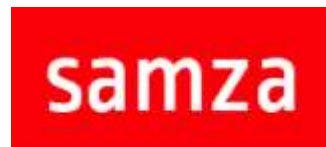
- Единая для разных проектов (support)
- Сложная бизнес-логика
- Жесткие NFR
- Низкий порог вхождения
- Мониторинг и настройка
- Внезапно: память – дорогой ресурс

И все
такое
вкусное...

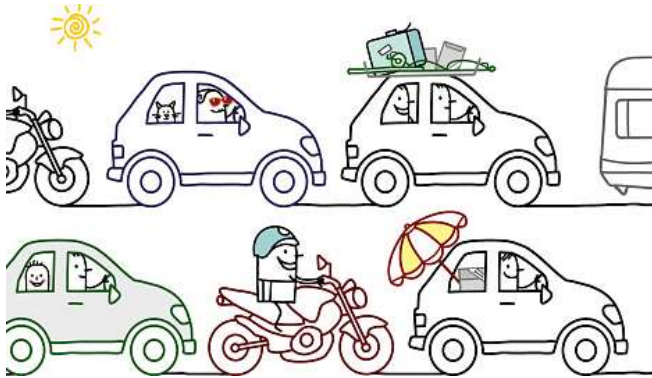
FIRSTMARK



APACHE
STORM™
Distributed • Resilient • Real-time



streaming



batch



Stream поверх batch: **НЕТ**

не верьте Spark Streaming!

Stream поверх batch: **НЕТ**

не верьте Spark Streaming!

Batch поверх stream: **ДА**

Сферический Flink в вакууме

```
DataStream<Tuple2<String, Integer>> dataStream = env  
    .socketTextStream("localhost", 9999)  
    .flatMap(new Splitter())  
    .myTransformation()  
    .keyBy(0)  
    .timeWindow(Time.seconds(5))  
    .sum(1);
```

+ жестокий энтерпрайз!

DataStream<Tuple2<String, Integer>

.socketTextStream("localhost")

.flatMap(new Splitter())

.myTransformation()

.keyBy(0)

.timeWindow(Time.seconds)

.sum(1);

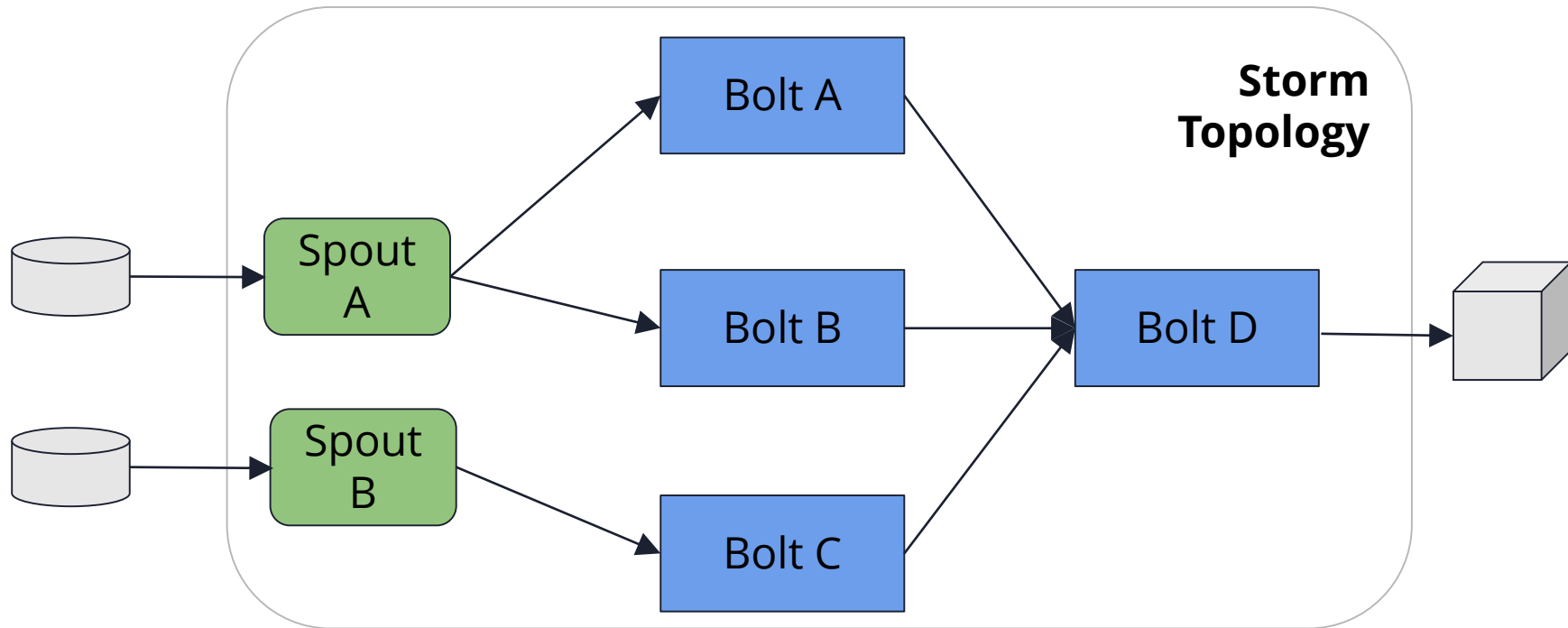
```
KTO_TAM_CAMOE_TAKOE^.XPEH_EMy:=  
uMEHHO_ETO_MOXHATOE_OHO_CAMOE; end;  
end; procedure XMb1Pb.B_nyTb; var  
ETOT_PAXuT: TAM_ABTOMAT; begin  
ETOT_PAXuT:= KTO_TAM_CAMOE_TAKOE;  
while not WyXEP do with ETOT_PAXuT^  
do begin Bb1B0g; ETOT_PAXuT:=  
XPEH_EMy; rge_EMy_KBACuTb(1, 10);  
TOPM03HyTb(10); end; end; procedure  
BAM_XOPOW0; begin BAM_TEMH0;  
WriteLN('OH TOPM03 v1.0 Copyright  
The Future Hackers, 1993');  
WriteLN(' Written by Kirill Sh...
```



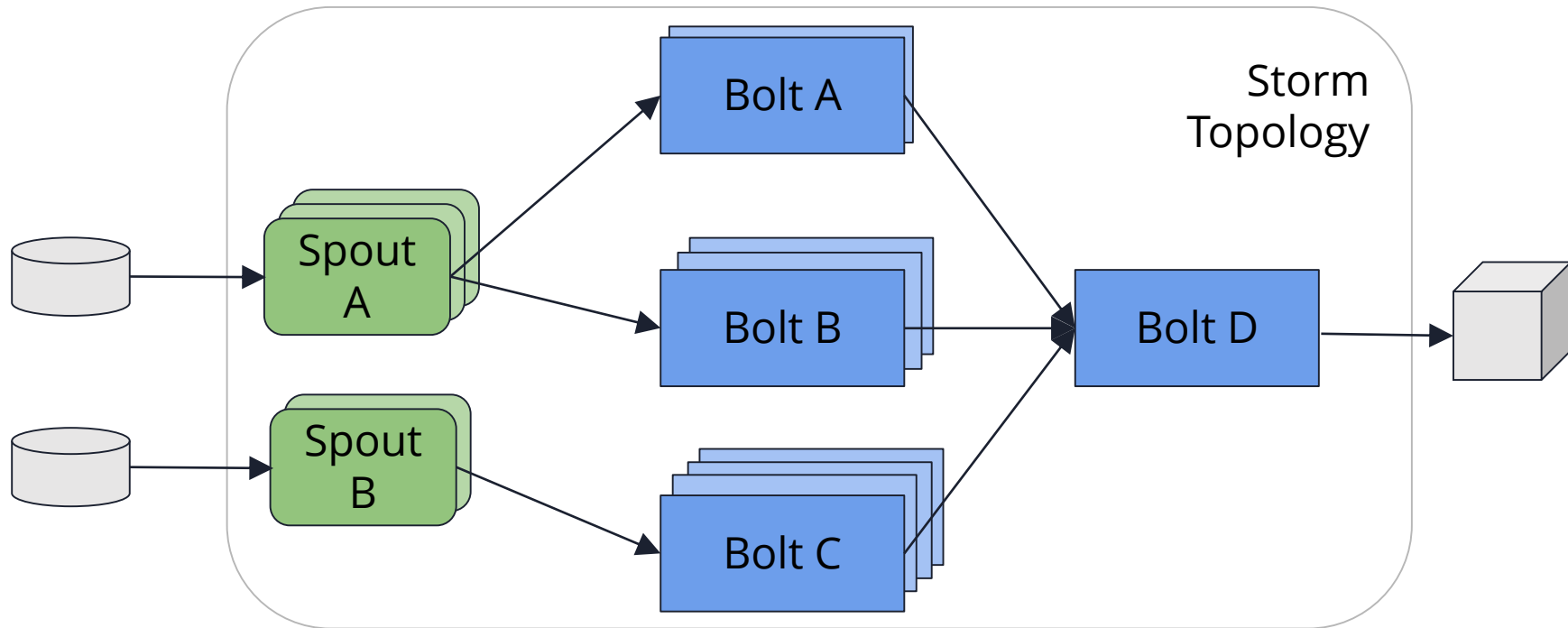
APACHE
STORMTM

Distributed • Resilient • Real-time

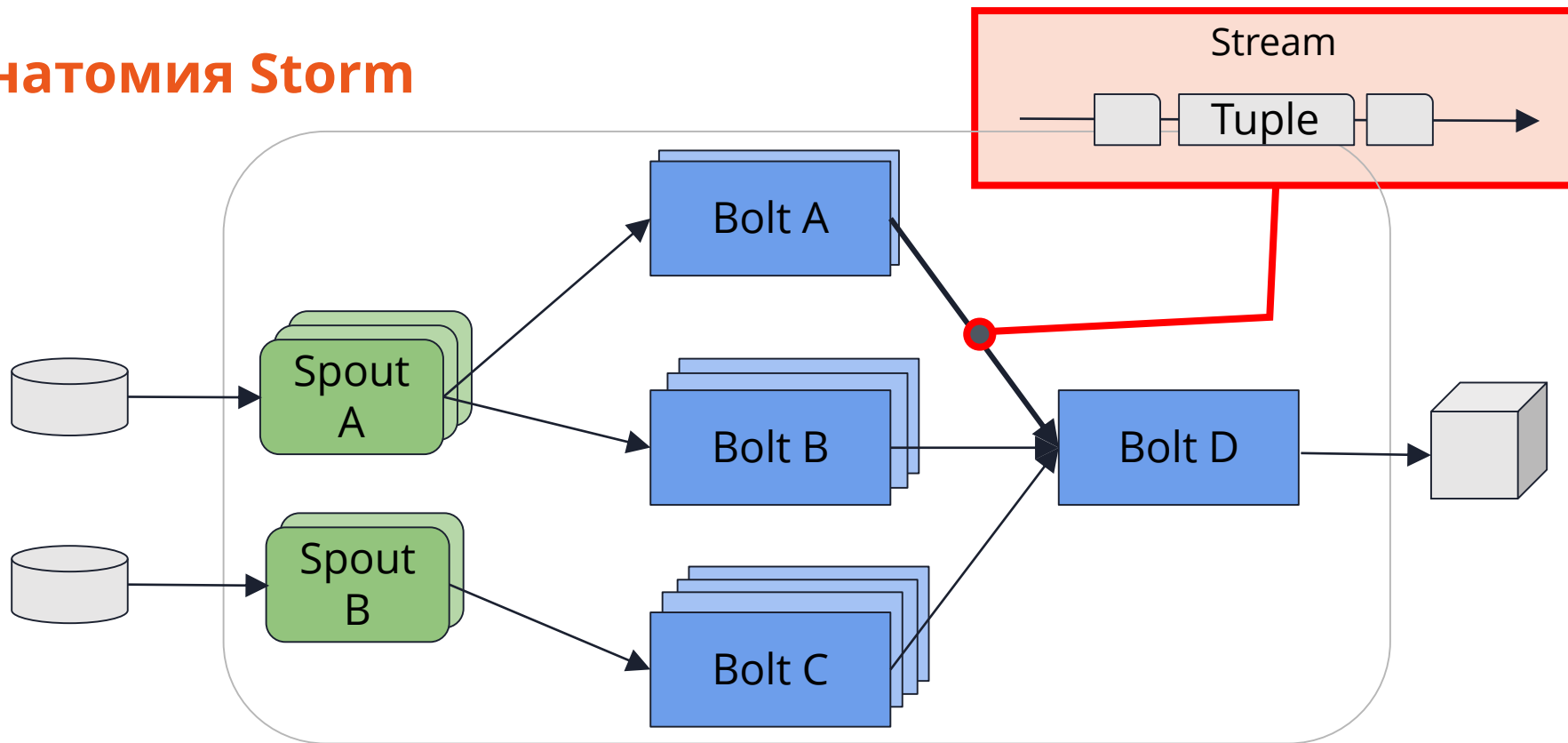
Анатомия Storm



Анатомия Storm



Анатомия Storm





Простота API

Storm: источник данных

```
class Spout extends BaseRichSpout {  
  
    void nextTuple() {
```

Storm: ИСТОЧНИК ДАННЫХ

```
class Spout extends BaseRichSpout {  
  
    void nextTuple() {  
        Values tuple = new Values(...);  
        collector.emit(tuple);  
    }  
}
```

Storm: обработчик

```
class Bolt extends BaseRichBolt {  
  
    void execute(Tuple input) {
```

Storm: обработчик

```
class Bolt extends BaseRichBolt {  
  
    void execute(Tuple input) {  
        Values output = ...(input);  
        collector.emit(output);  
    }  
}
```

Storm: создание топологии

```
class MyTopology { void main(...) {  
  
    builder = new TopologyBuilder();
```


Storm: создание топологии

```
class MyTopology { void main(...) {  
  
    builder = new TopologyBuilder();  
    builder.addSpout("spout", new MySpout(), 2);
```

Storm: создание топологии

```
class MyTopology { void main(...) {  
  
    builder = new TopologyBuilder();  
    builder.addSpout("spout", new MySpout(), 2);  
    builder.addBolt("bolt", new MyBolt(), 4)
```

Storm: создание топологии

```
class MyTopology { void main(...) {  
  
    builder = new TopologyBuilder();  
    builder.addSpout("spout", new MySpout(), 2);  
    builder.addBolt("bolt", new MyBolt(), 4)  
        .shuffle("spout");  
}
```

Storm: создание топологии

```
class MyTopology { void main(...) {  
  
    builder = new TopologyBuilder();  
    builder.addSpout("spout", new MySpout(), 2);  
    builder.addBolt("bolt", new MyBolt(), 4)  
                .shuffle("spout");  
    StormSubmitter.submit("topology", config,  
                           builder.buildTopology());  
}  
}
```

— Ваш телевизор в порядке! Идет репортаж с художественной выставки.



Мониторинг и настройка

Bolts (All time)

Id	Executors	Tasks	Emitted	Transferred	Capacity (last 10m)	Execute latency (ms)	Executed	Process latency (ms)	Acked	Failed
2shortServiceCallBolt	1	32	640	640	0.025	11.462	520	10.815	540	0
3multiplyBolt	1	32	5420	5420	0.000	0.185	540	0.111	540	0
4cpuBolt	1	32	5200	5200	0.574	25.291	5360	25.300	5340	0
5longServiceCallBolt	2	32	4520	4520	0.939	100.686	4400	100.482	4400	0
6consumerBolt	1	32	0	0	0.001	0.027	4400	0.018	4400	0

Bolts (10m 0s)

Id	Executors	Tasks	Emitted	Transferred	Capacity (last 10m)	Execute latency (ms)	Executed	Process latency (ms)	Acked	Failed
2shortServiceCallBolt	1	32	1640	1640	0.085	10.446	1660	10.482	1660	0
3multiplyBolt	1	32	16440	16440	0.000	0.060	1660	0.193	1660	0
4cpuBolt	2	32	16340	16340	0.998	25.448	15980	25.428	15980	0
5longServiceCallBolt	8	32	15700	15700	0.996	100.694	15740	100.551	15720	0
6consumerBolt	1	32	0	0	0.001	0.011	15760	0.016	15760	0

Spout A
instance 1

Spout A
instance 2

Spout B
instance 1

Executor 1 (*thread*)

Spout A
instance 1

Spout A
instance 2

Executor 2 (*thread*)

Spout B
instance 1

Executor 1 (*thread*)

Spout A
instance 1

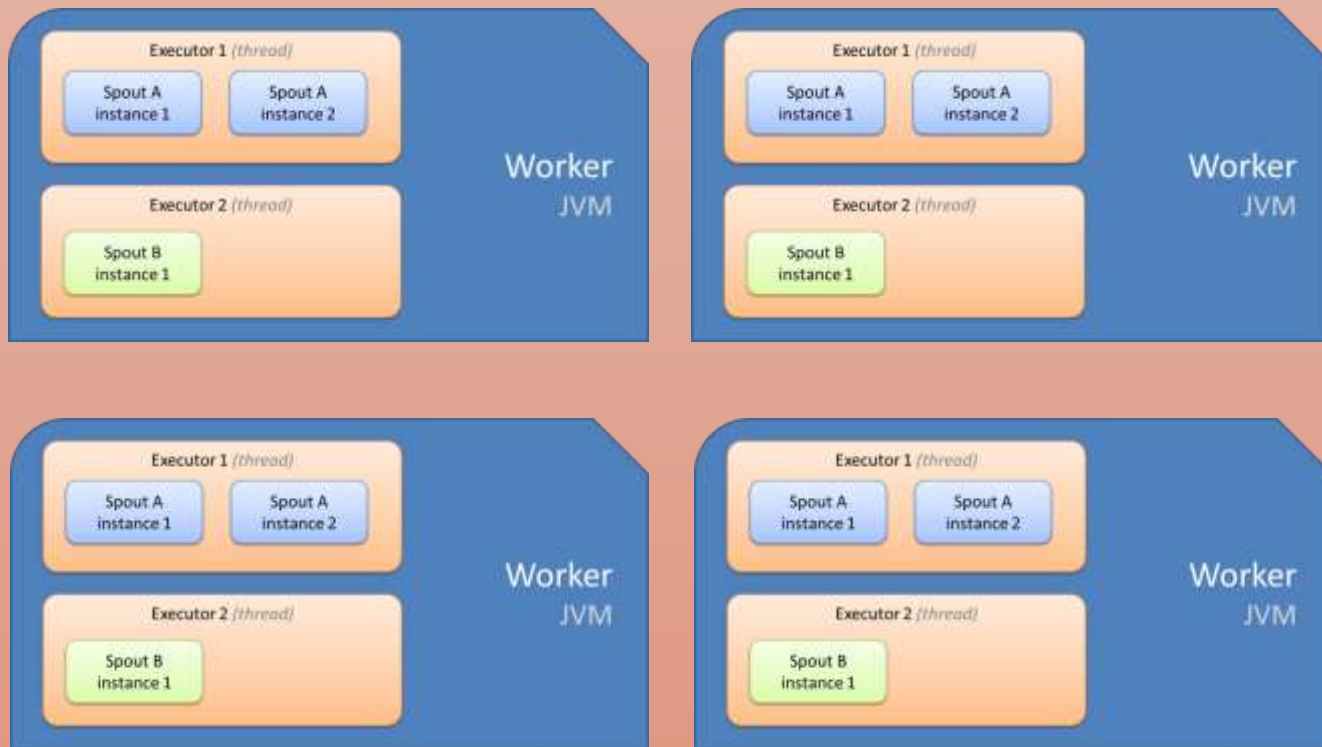
Spout A
instance 2

Executor 2 (*thread*)

Spout B
instance 1

Worker
JVM

Host





Подводные камни

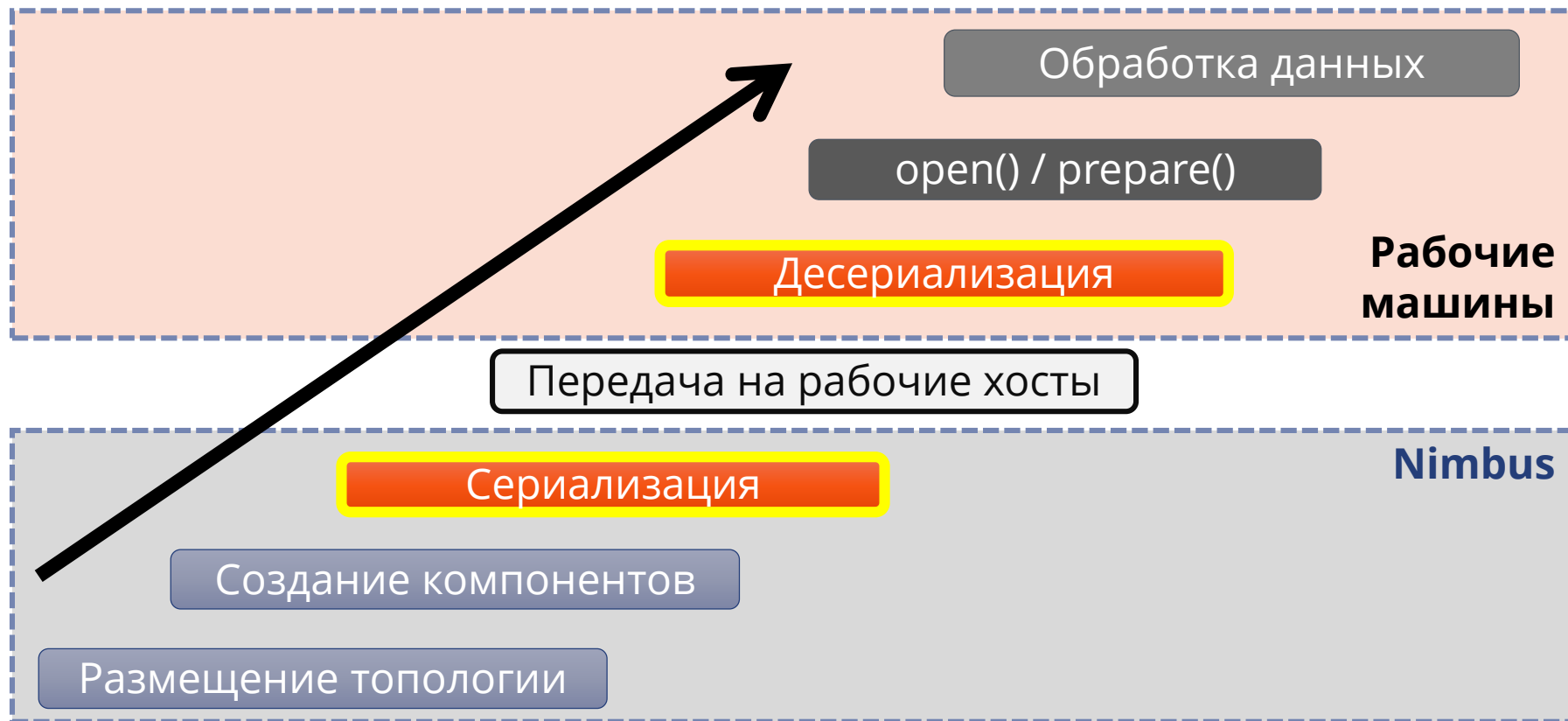


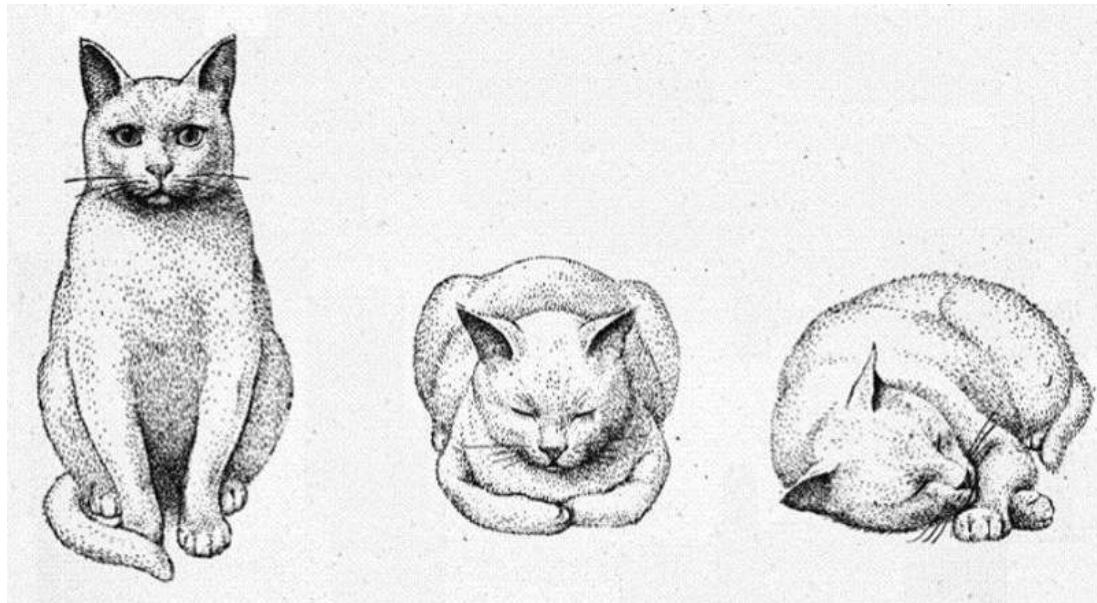
Проблема №1: грабли сериализации

Проблема №1: сериализация данных

```
class MyDataSupplier { // Not serializable!  
    public MyData getData() {  
  
        // Анонимный класс  
        // Хотя: MyData implements Serializable  
        return new MyData() { ... }  
    }  
}
```

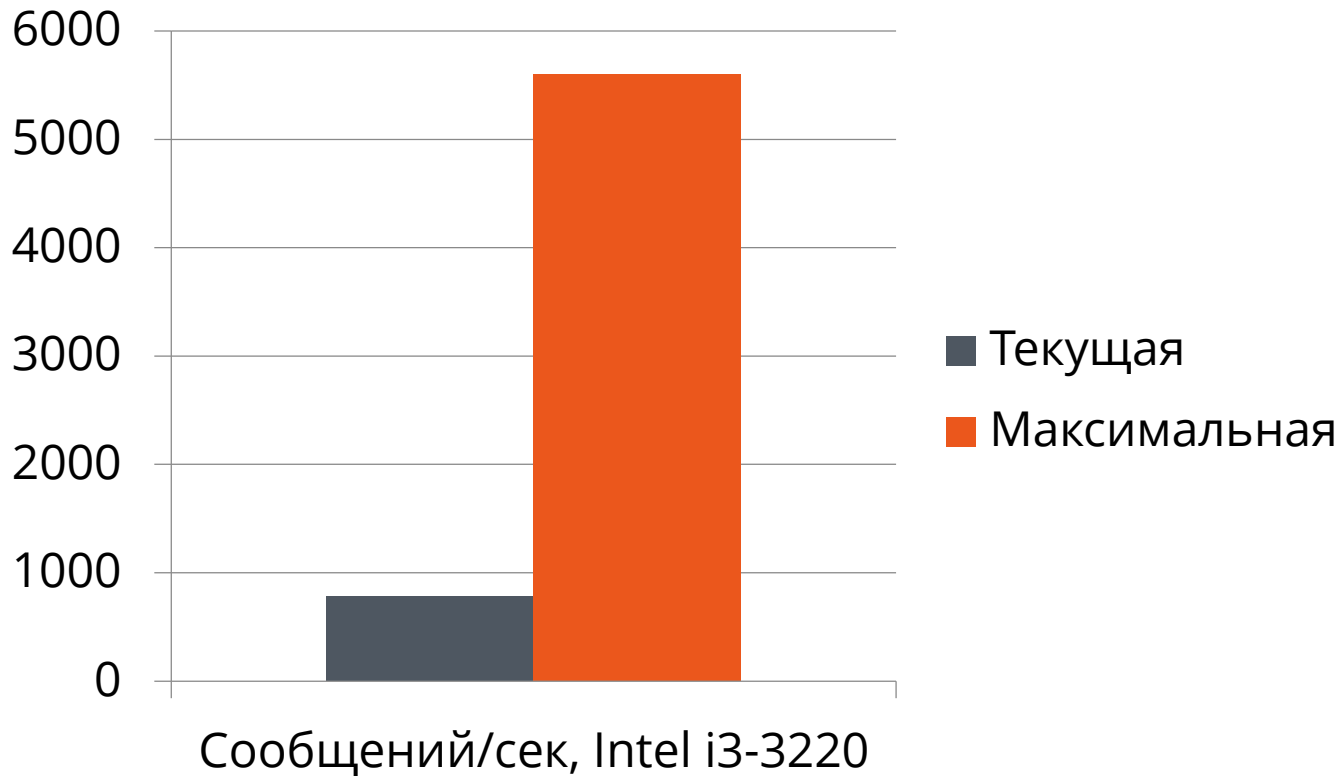
Сериализация кода: взлетим?





Проблема №2: низкая производительность sprout

Проблема 1: низкая производительность sprout



Остерегайтесь ISpoutWaitStrategy

```
class MySpout ... { void nextTuple() {  
  
    if (iHaveTupleToSend()) {  
        collector.emit(tuple);  
    } else {  
        // ISpoutWaitStrategy: 1ms delay  
    }  
}  
}
```

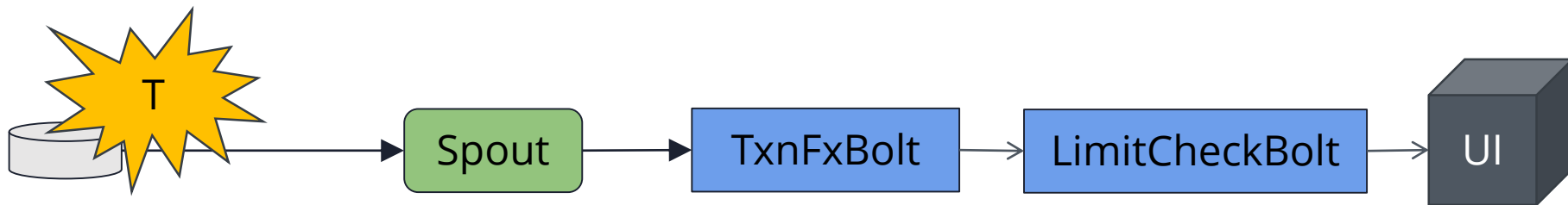
MySpoutWaitStrategy

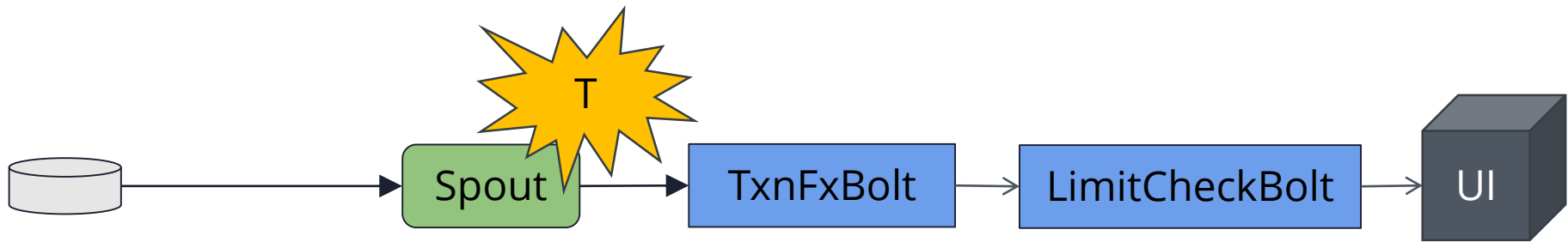
- ◆ `config.put(
 Config.TOPOLOGY_SLEEP_SPOUT_WAIT_STRATEGY_TIME_MS,
 1000);`
- ◆ `class MyStrategy implements ISpoutWaitStrategy {
 void emptyEmit(long streak)
 { /* Спите, сколько хотите! */ } }`
- `Config.put(TOPOLOGY_SPOUT_WAIT_STRATEGY, "MyStrategy");`

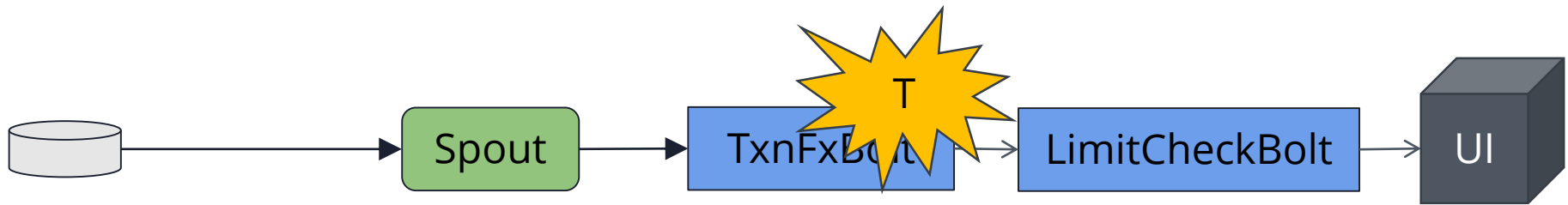
Проблема №3: «летучий голландец»

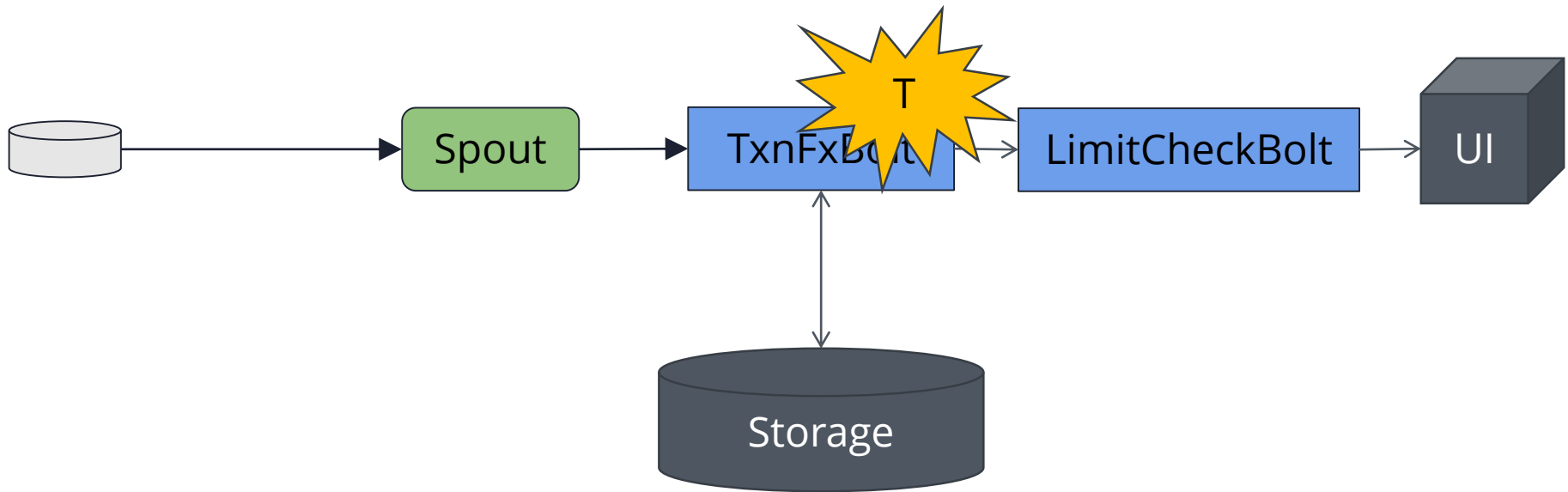


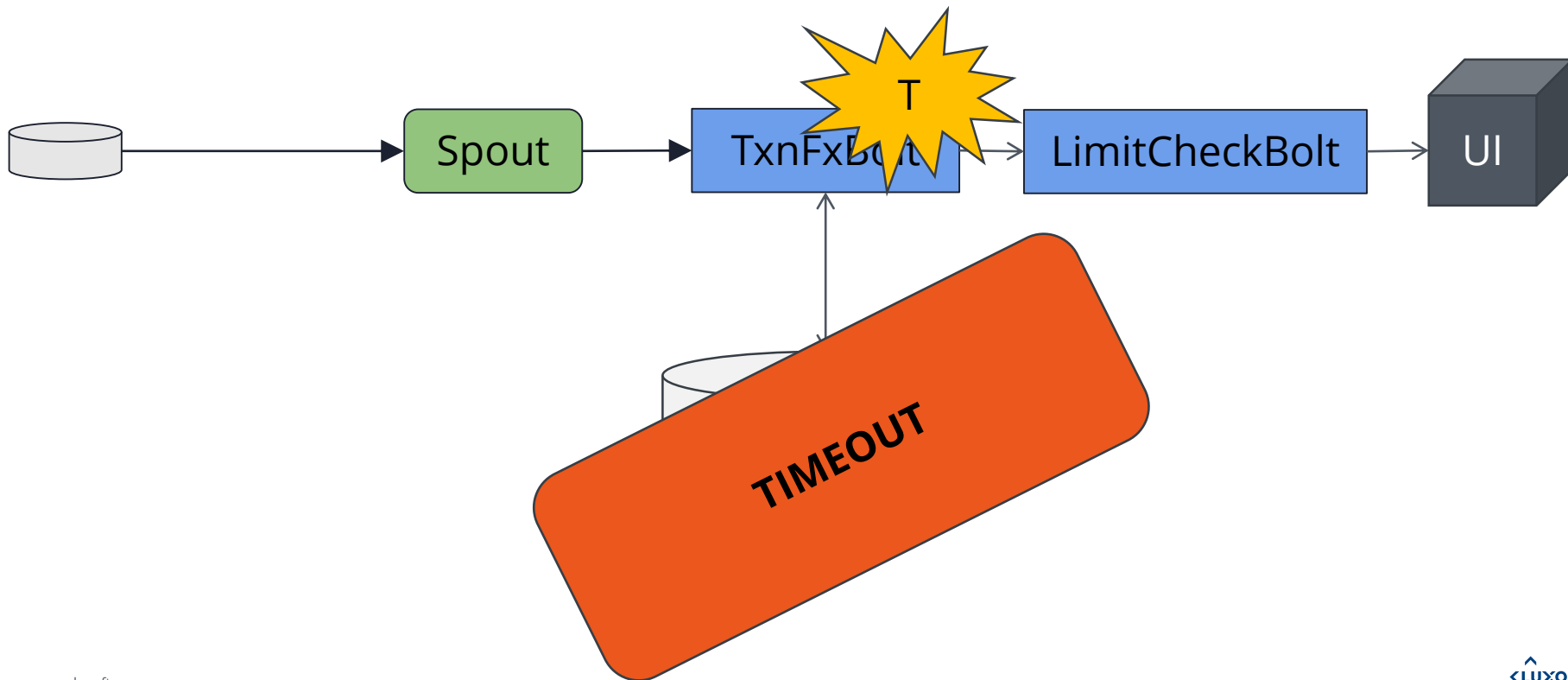












Модели доставки (delivery)

Модели доставки (delivery)

- At **most** once

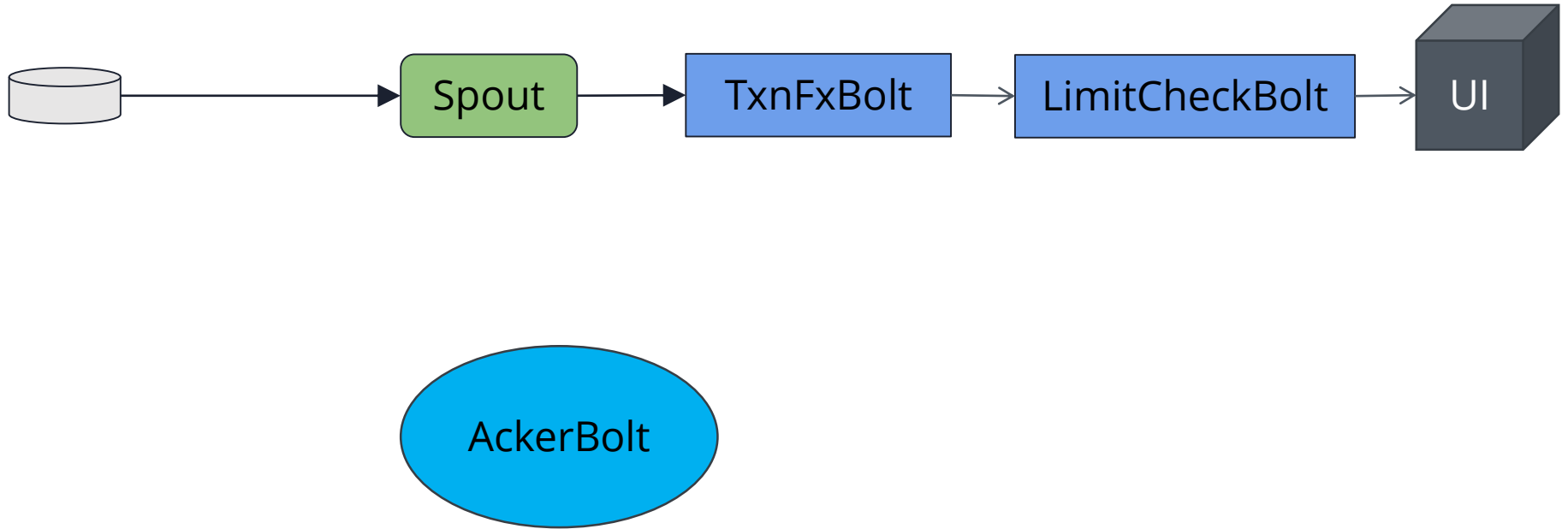
Модели доставки (delivery)

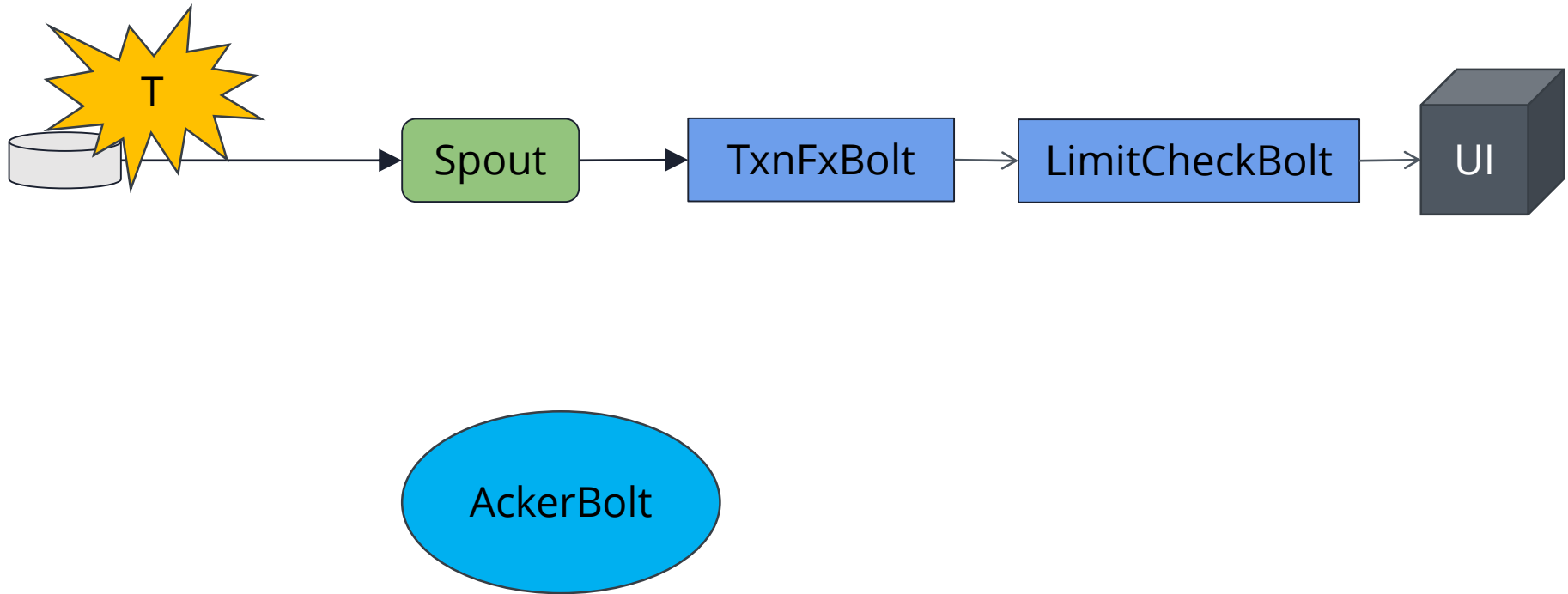
- At **most** once
- At **least** once

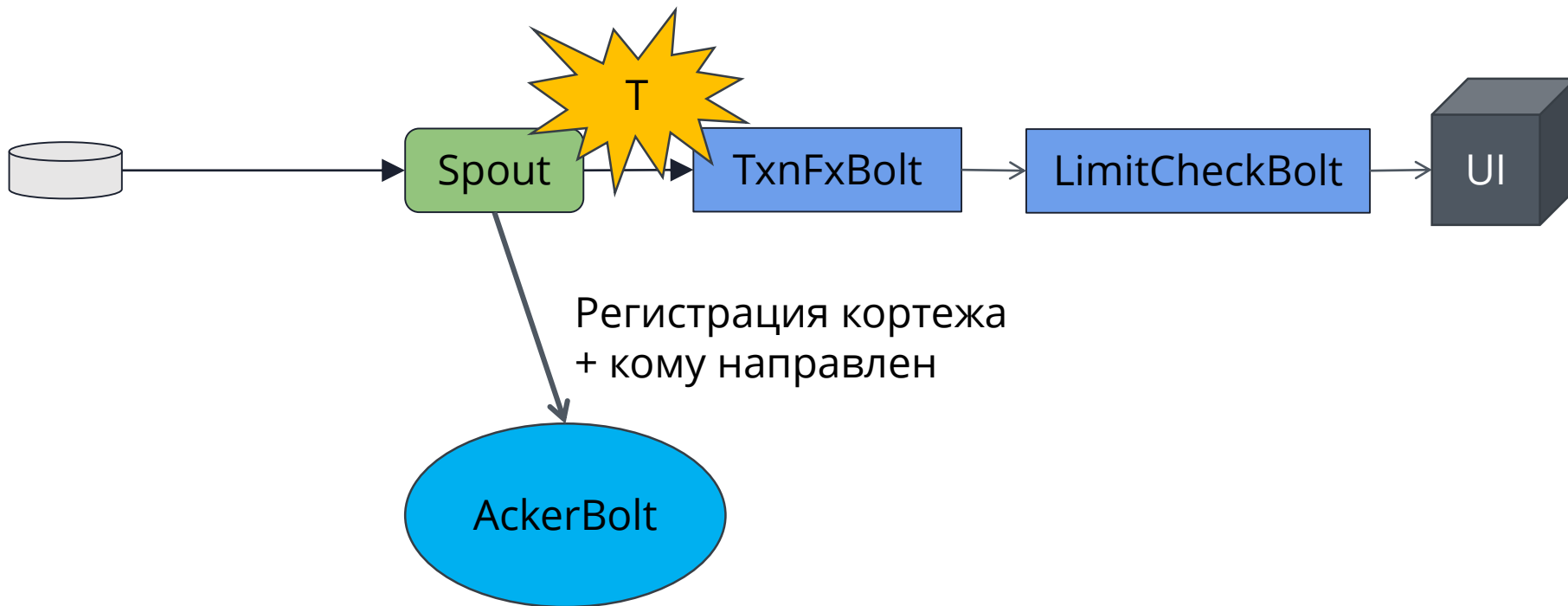
Модели доставки (delivery)

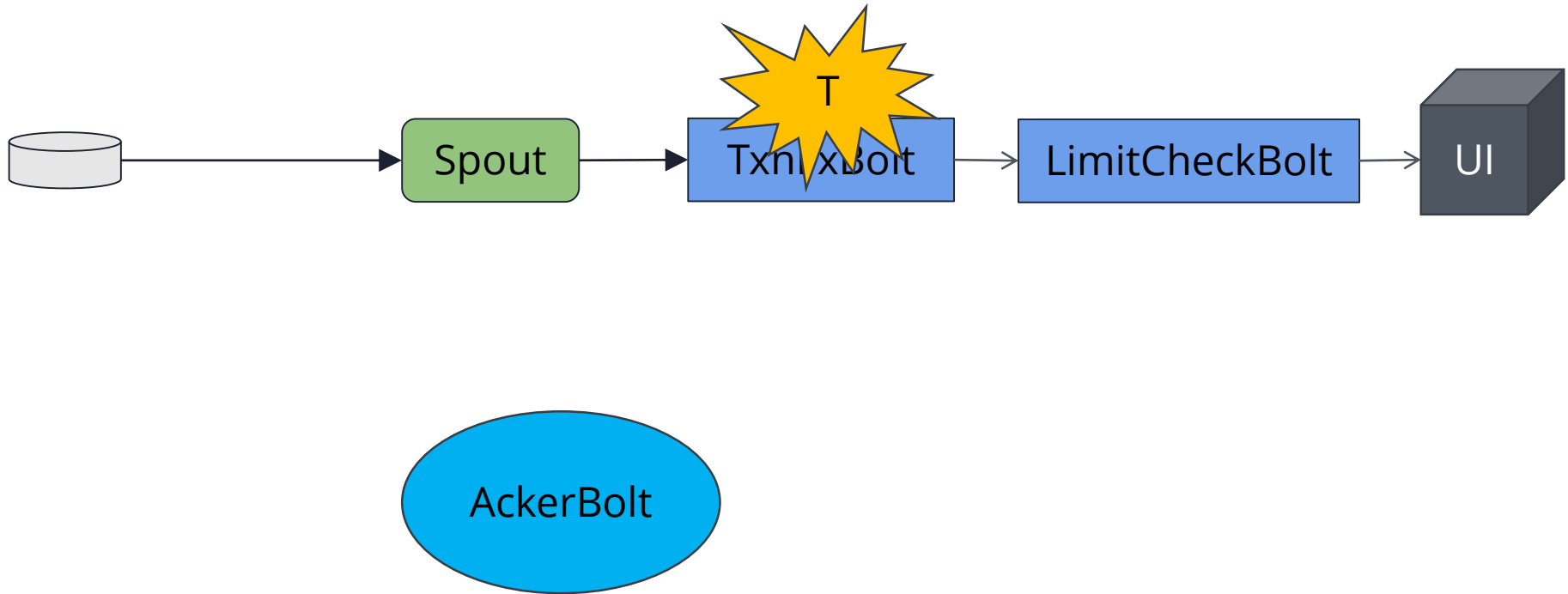
- At **most** once
- At **least** once
- **Exactly** once: **МИФ!**

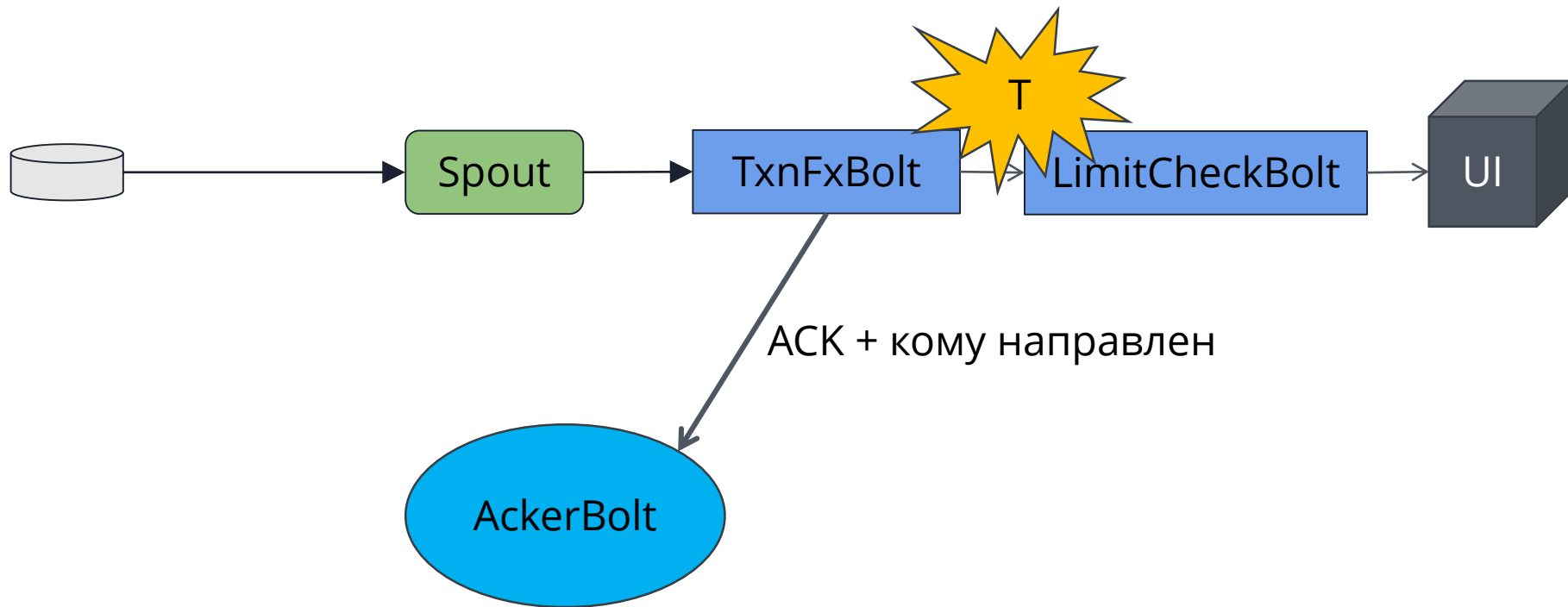


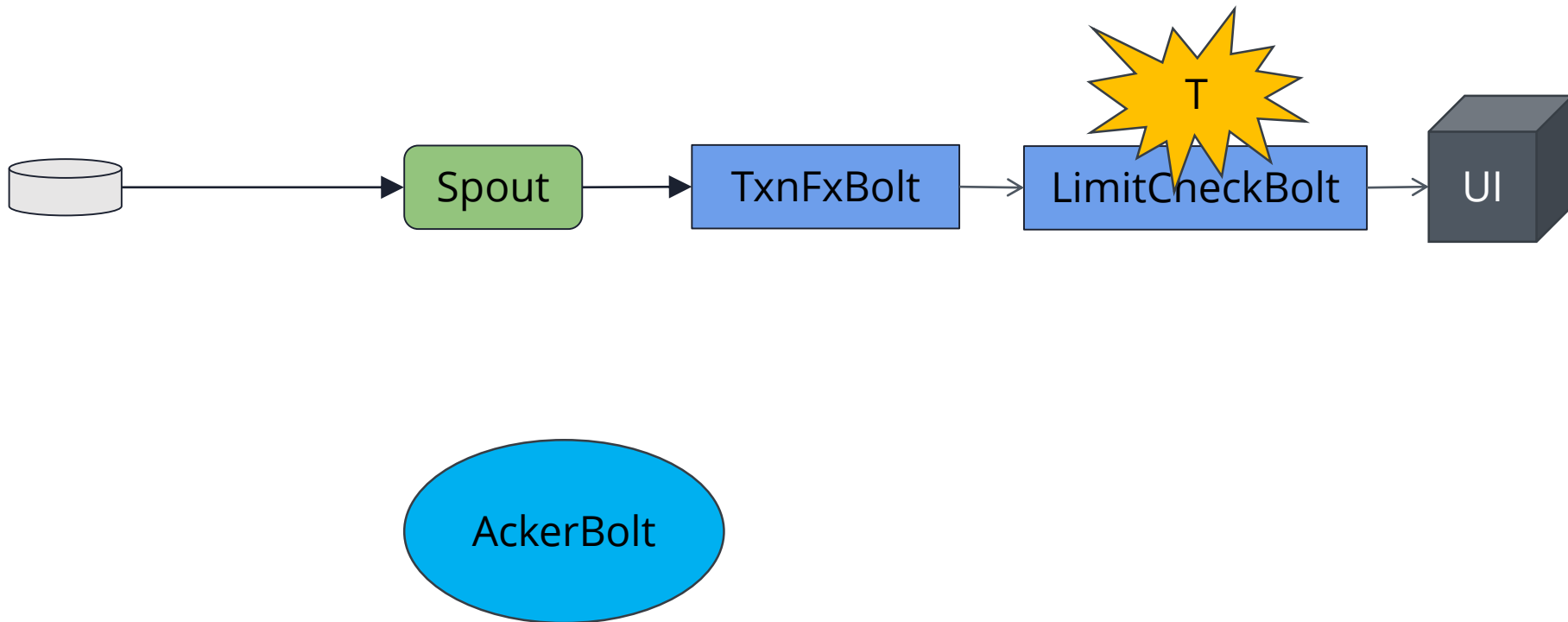


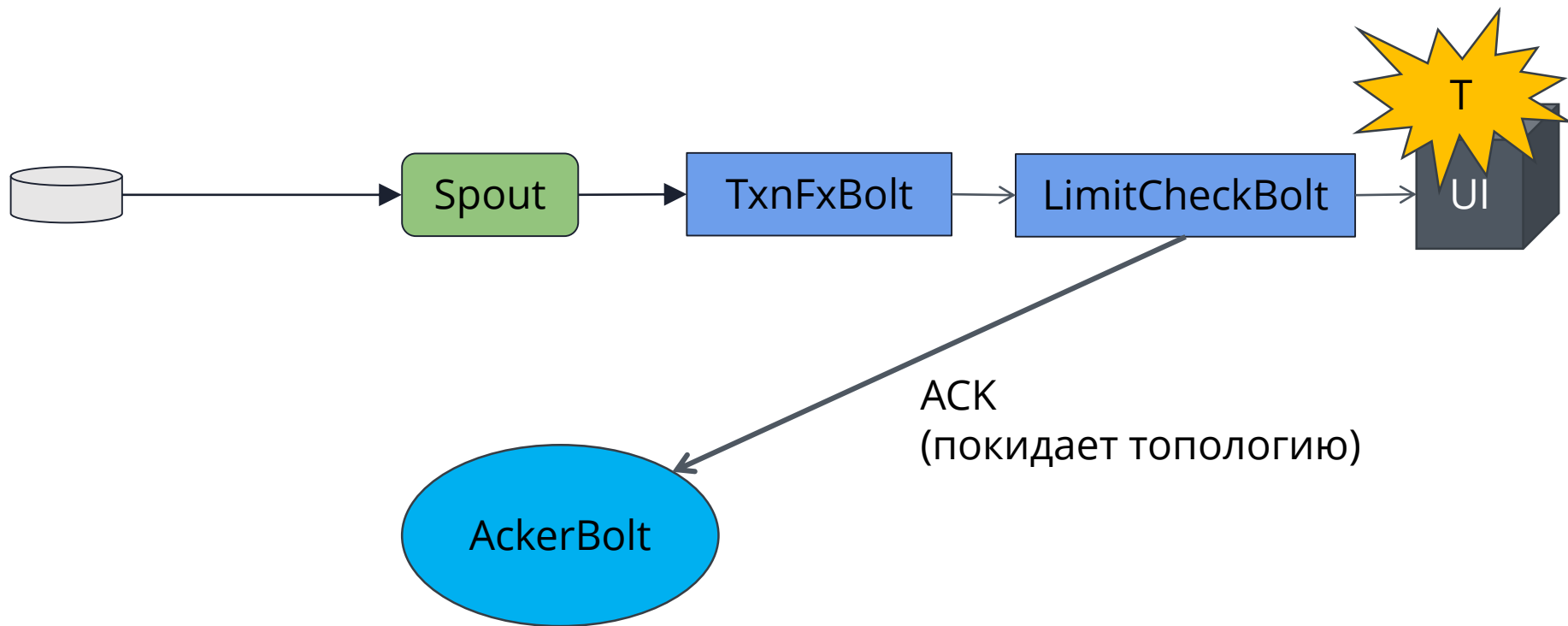


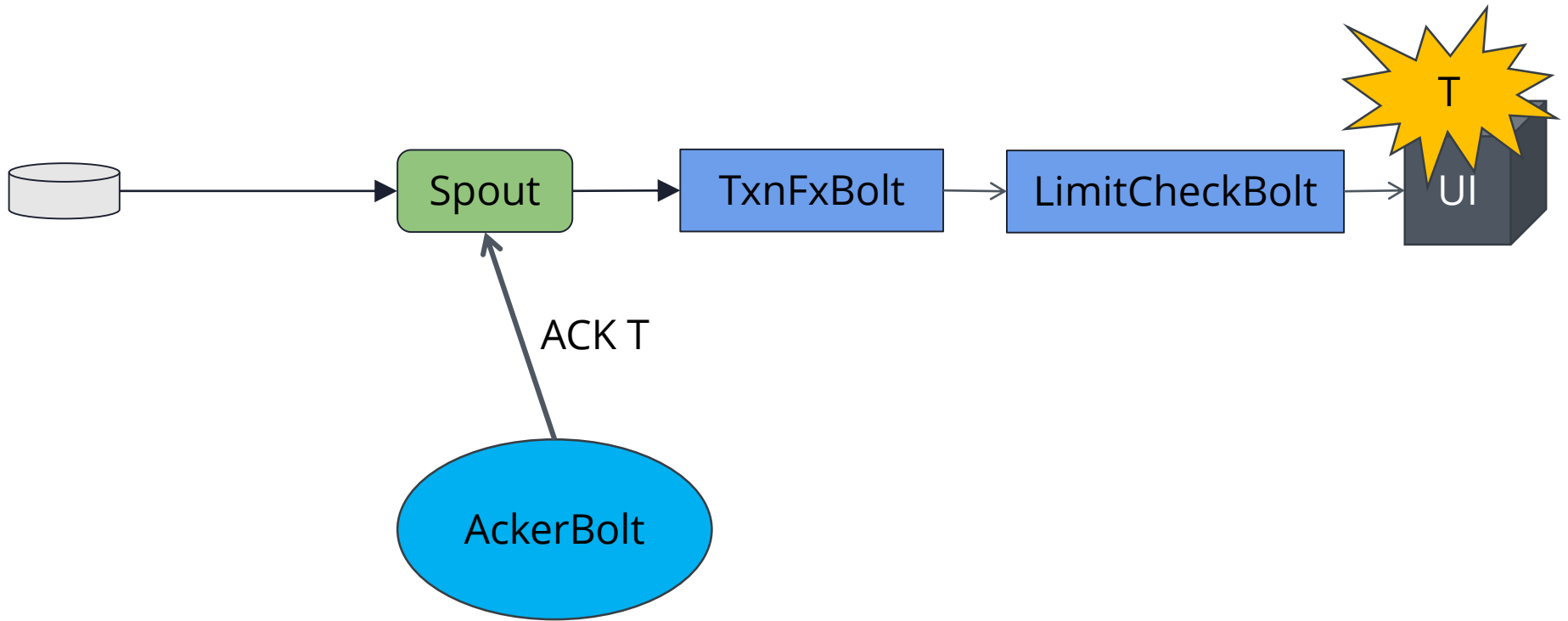


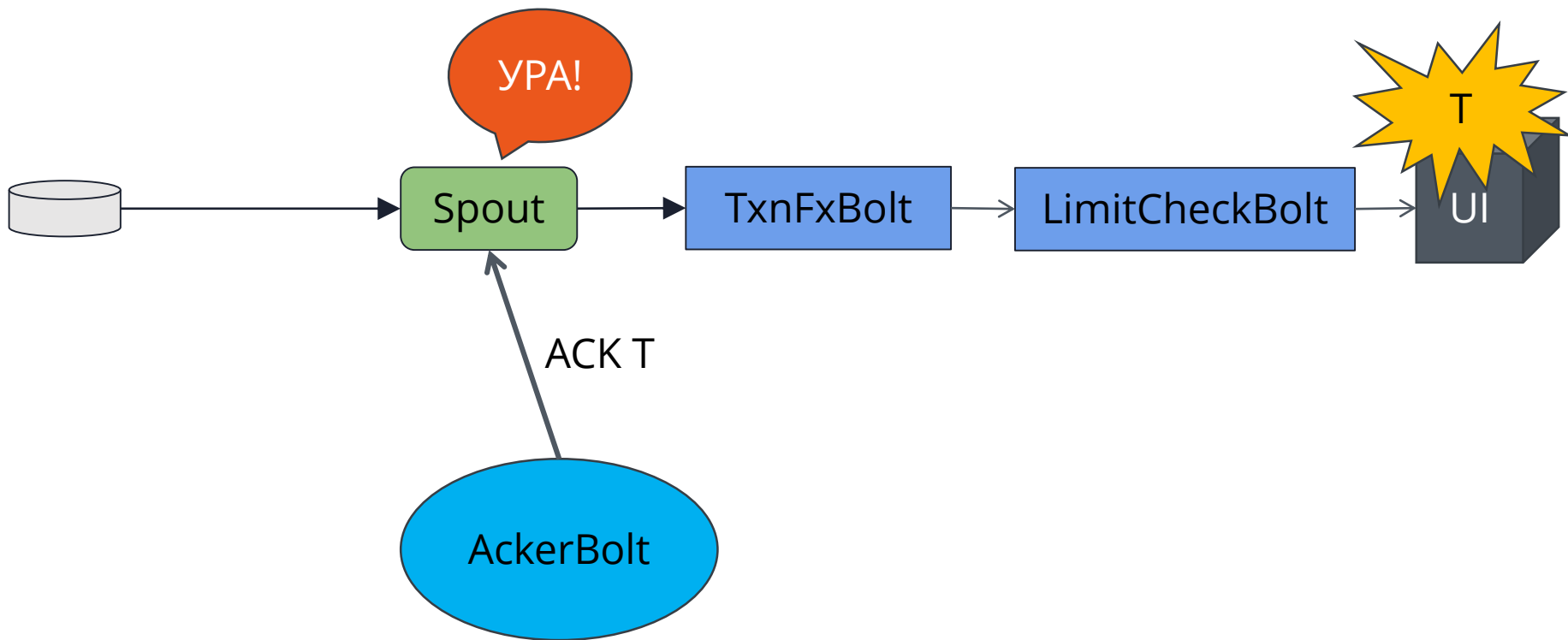


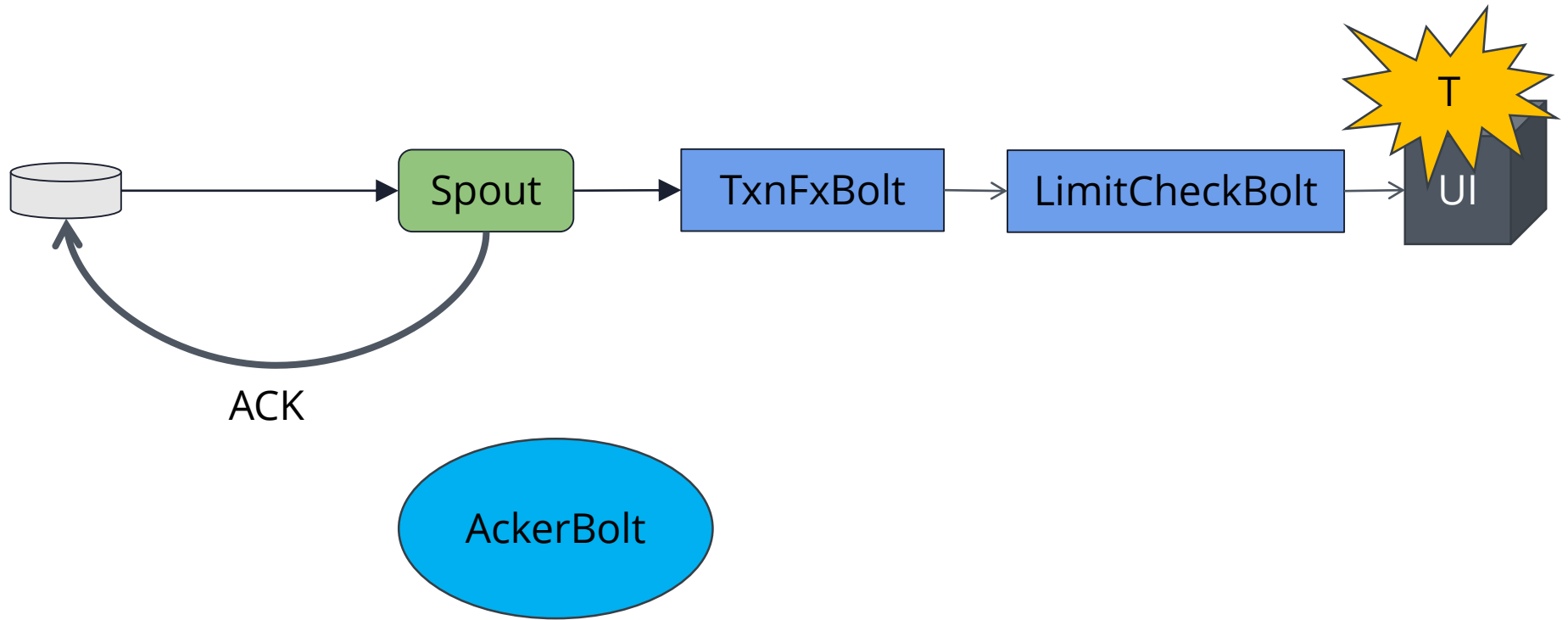


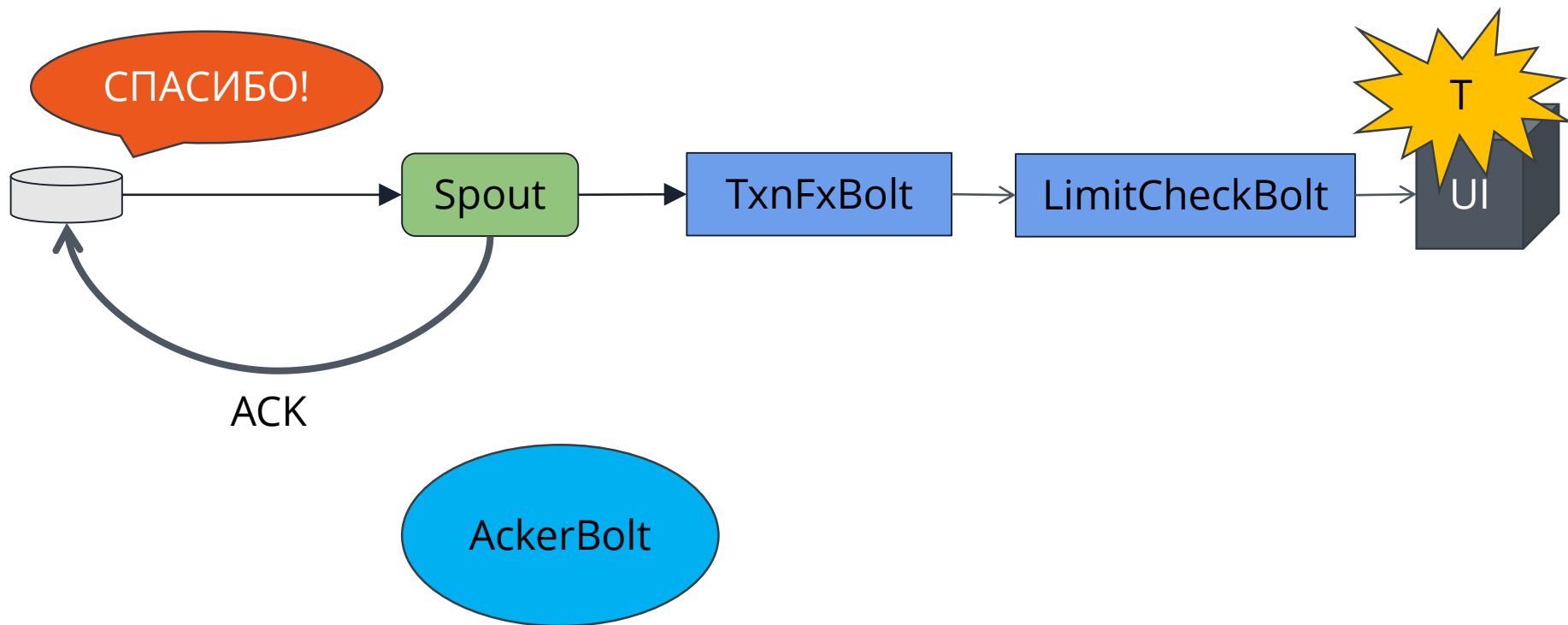






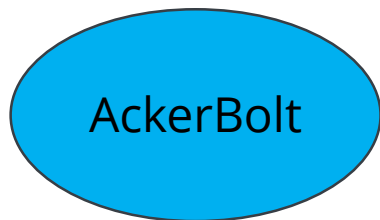
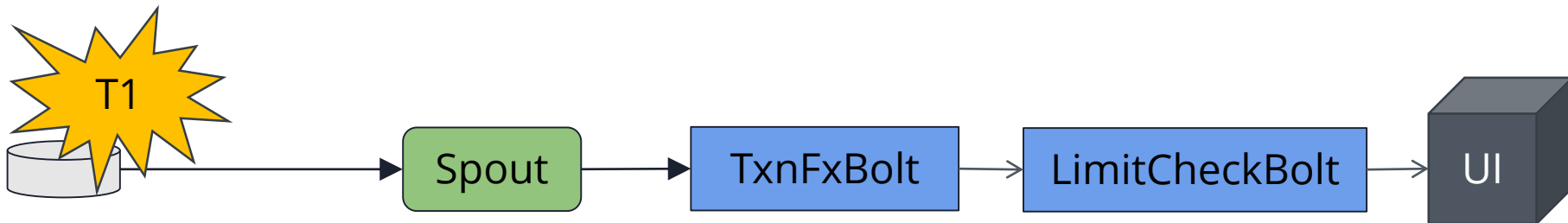


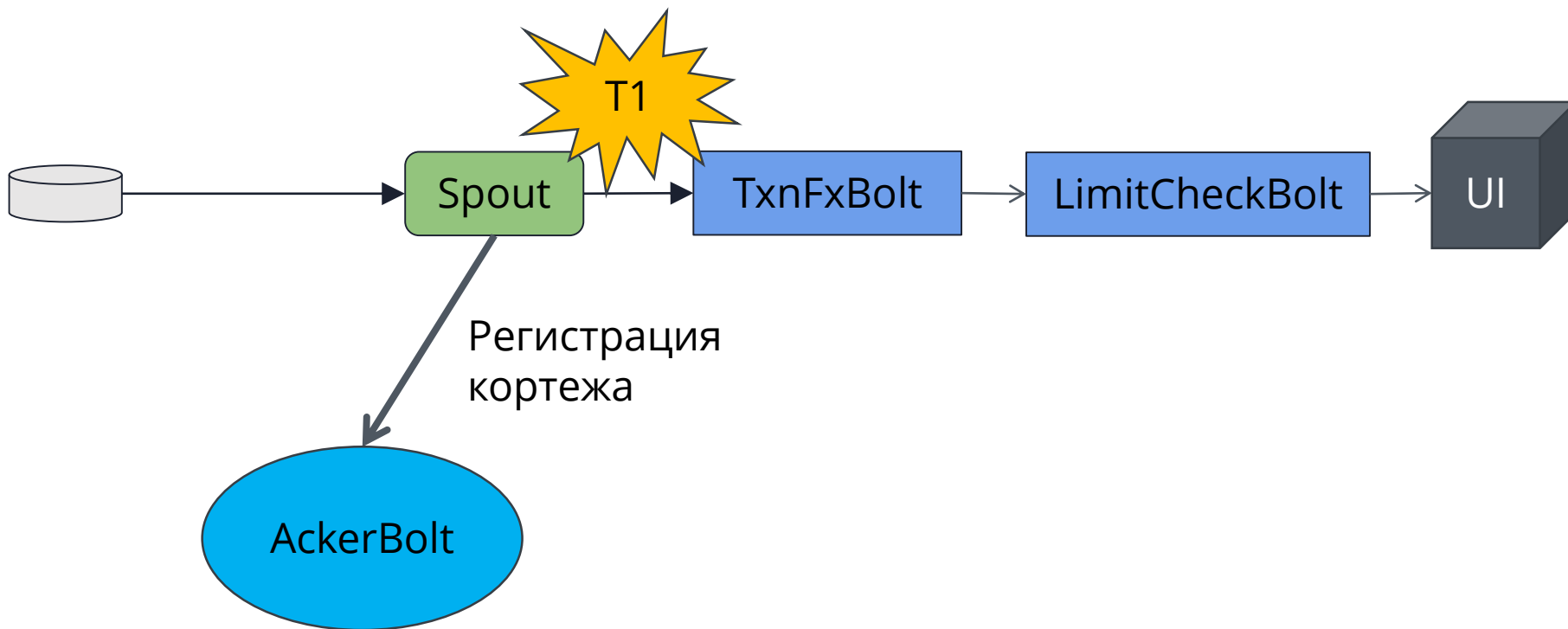


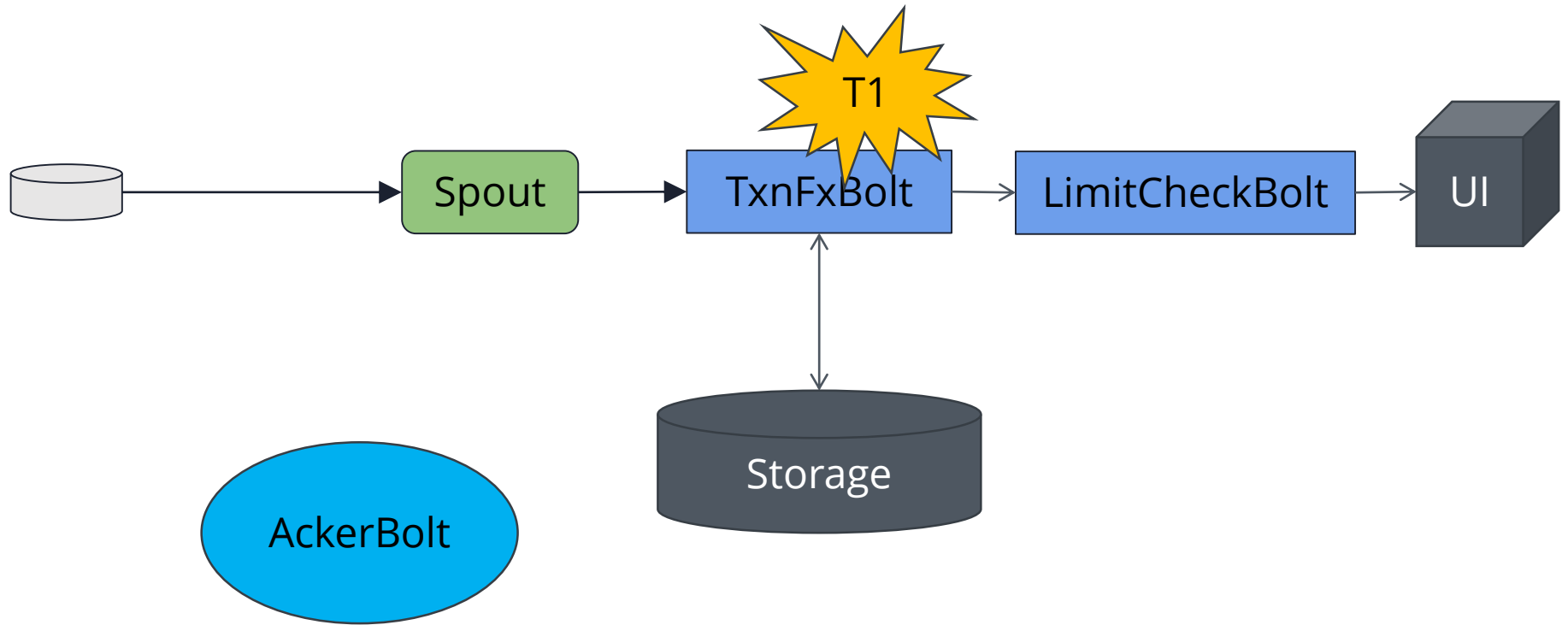


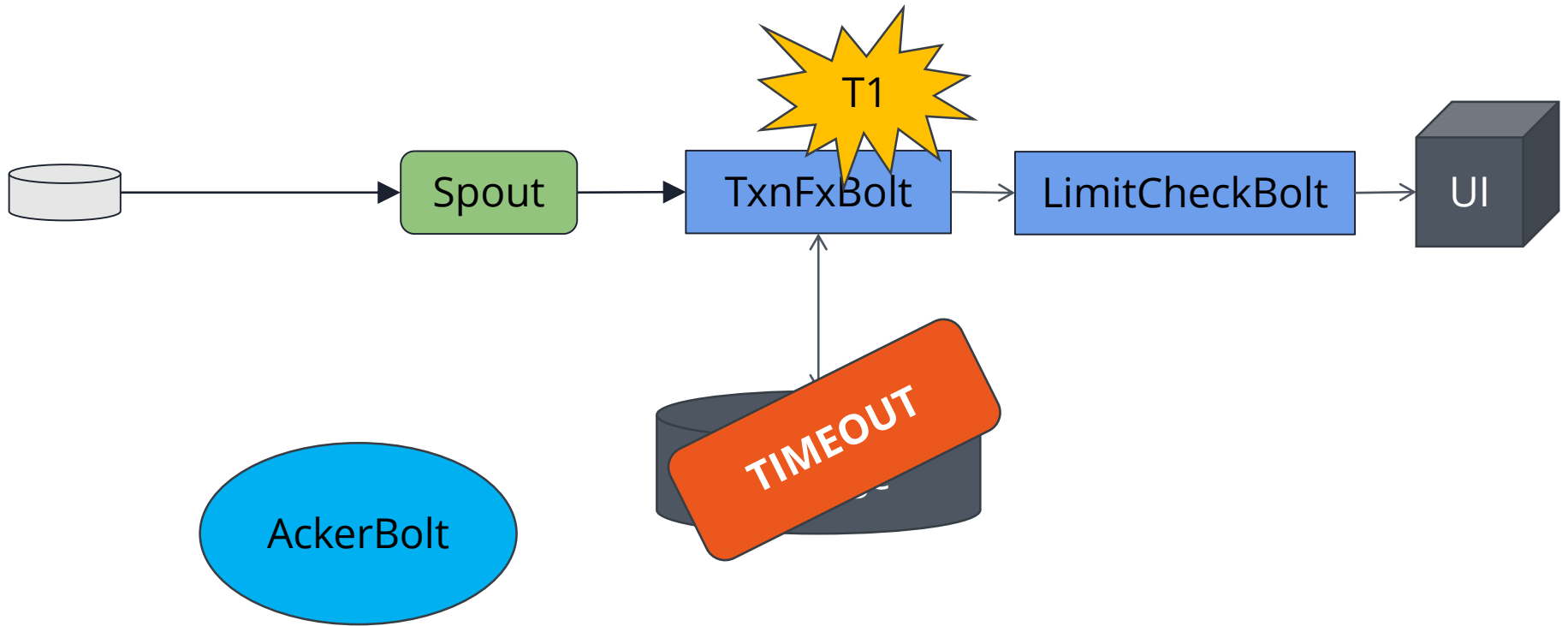
**И в чём
проблема?**

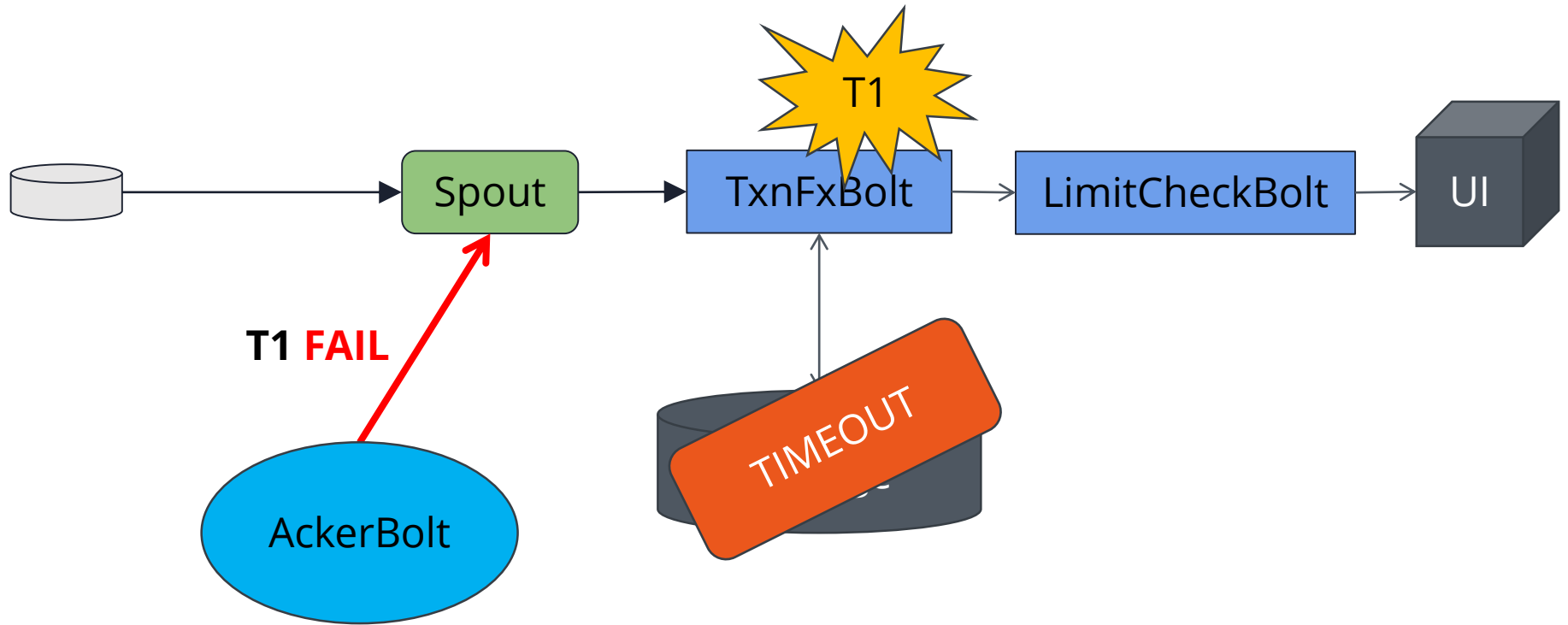


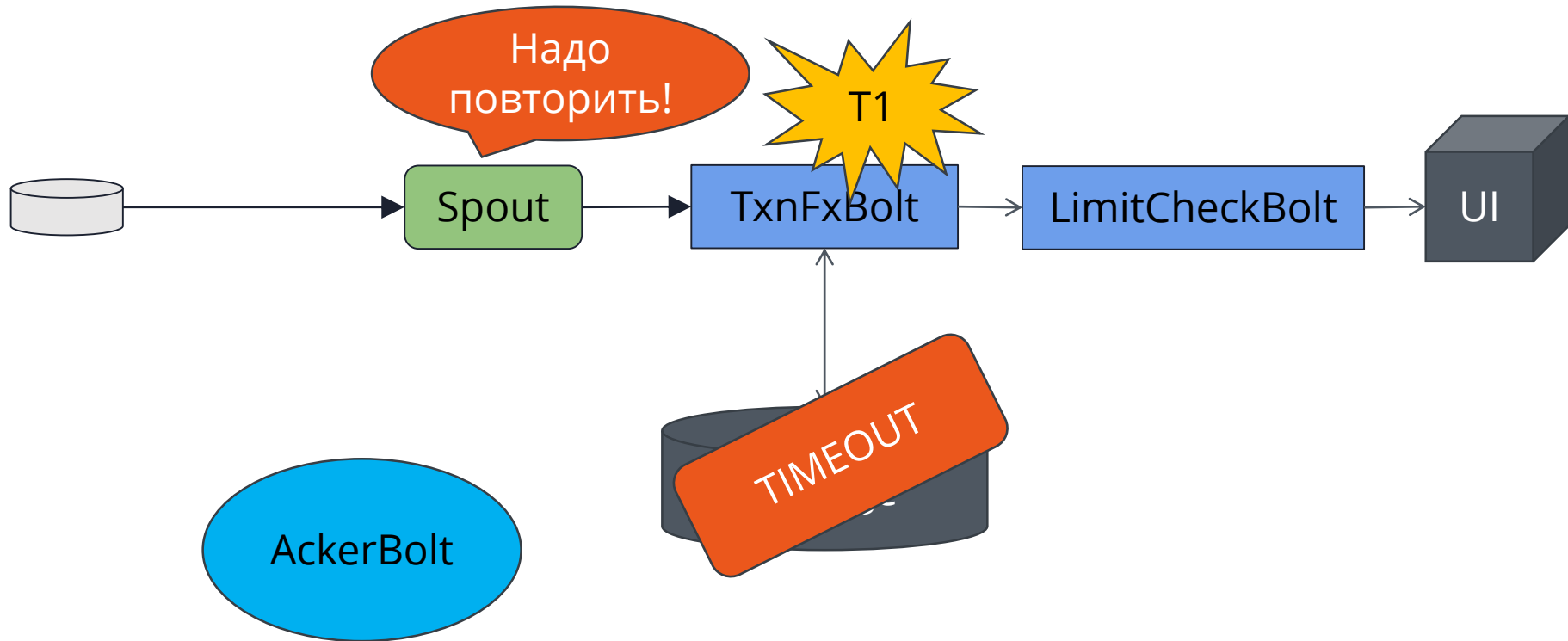


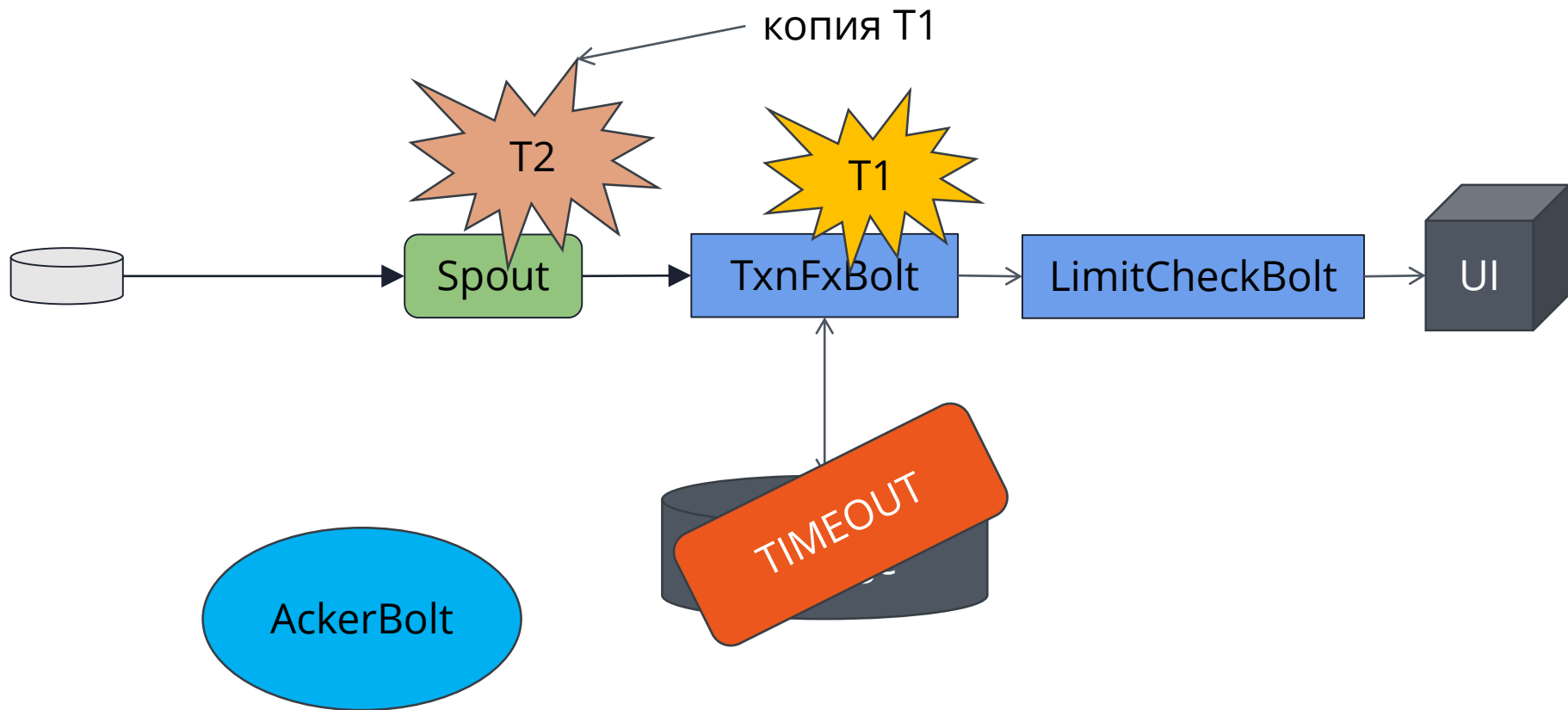


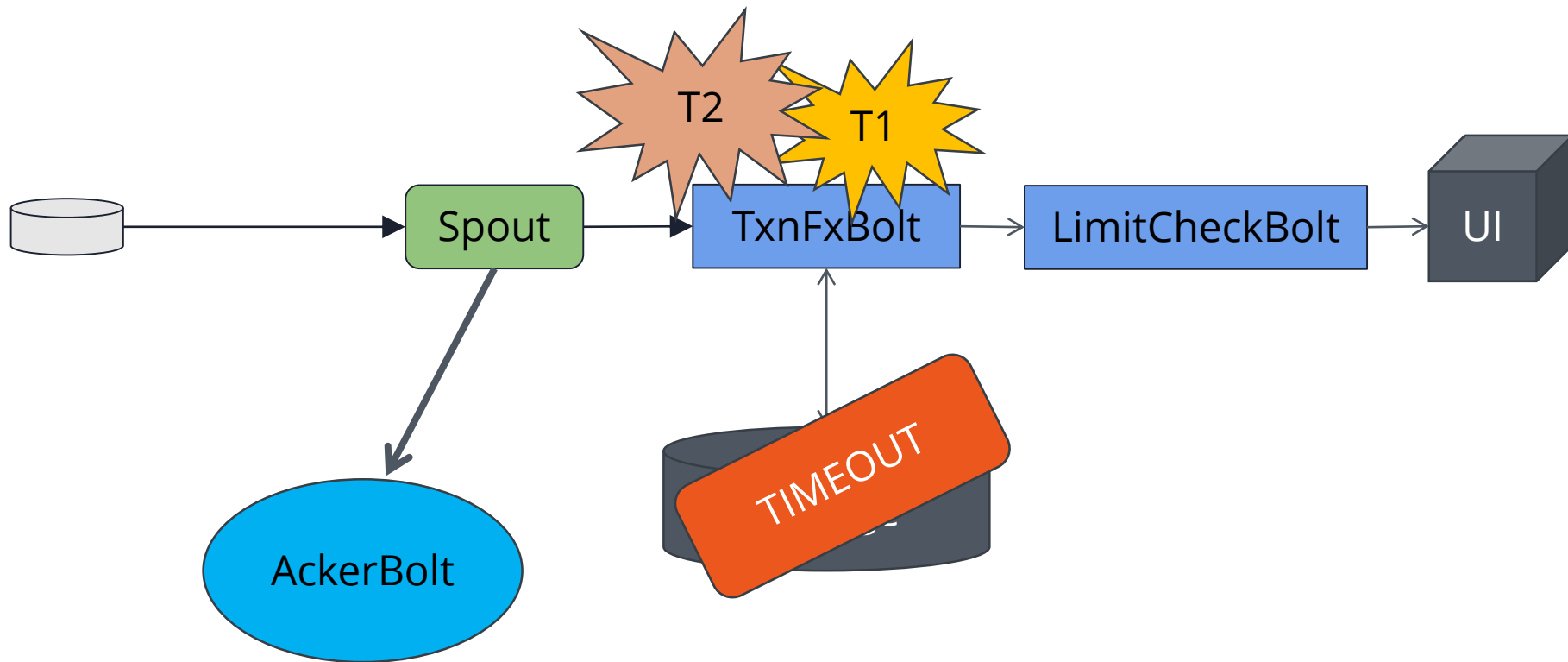


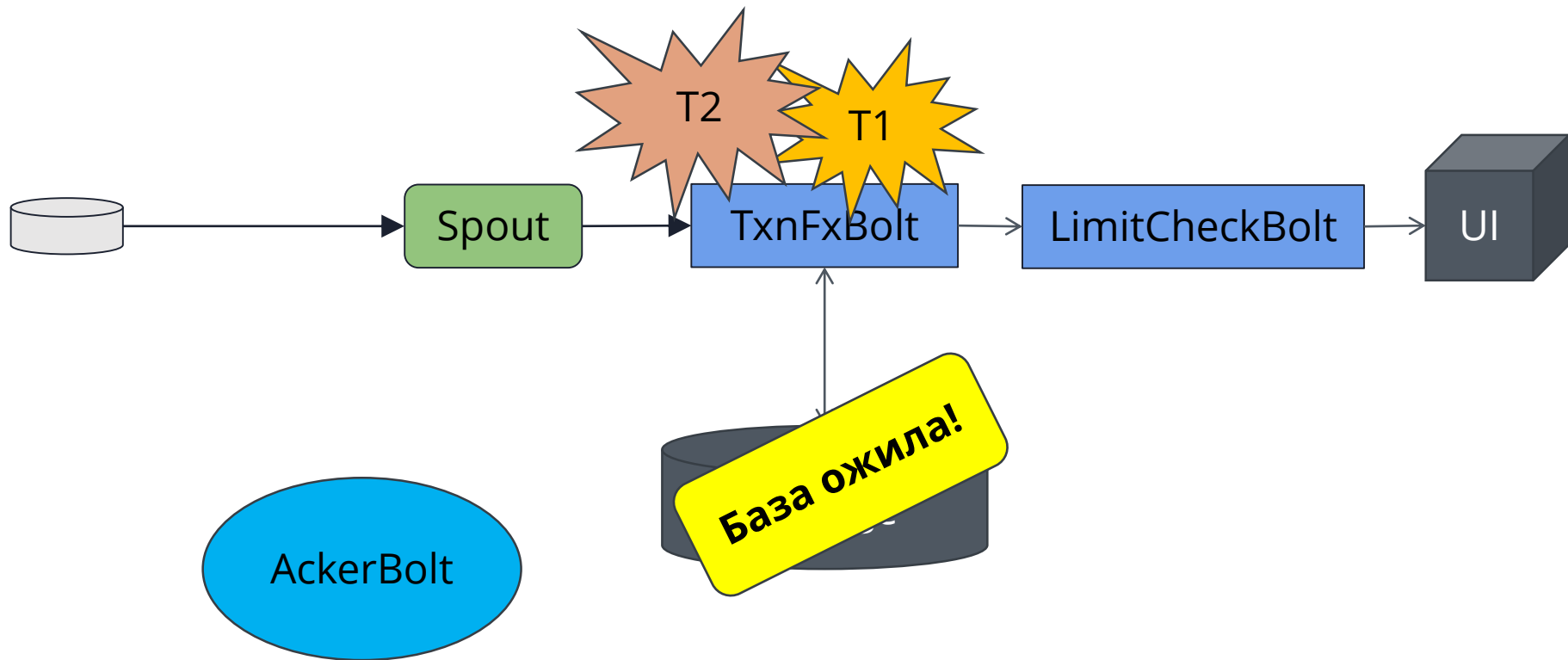




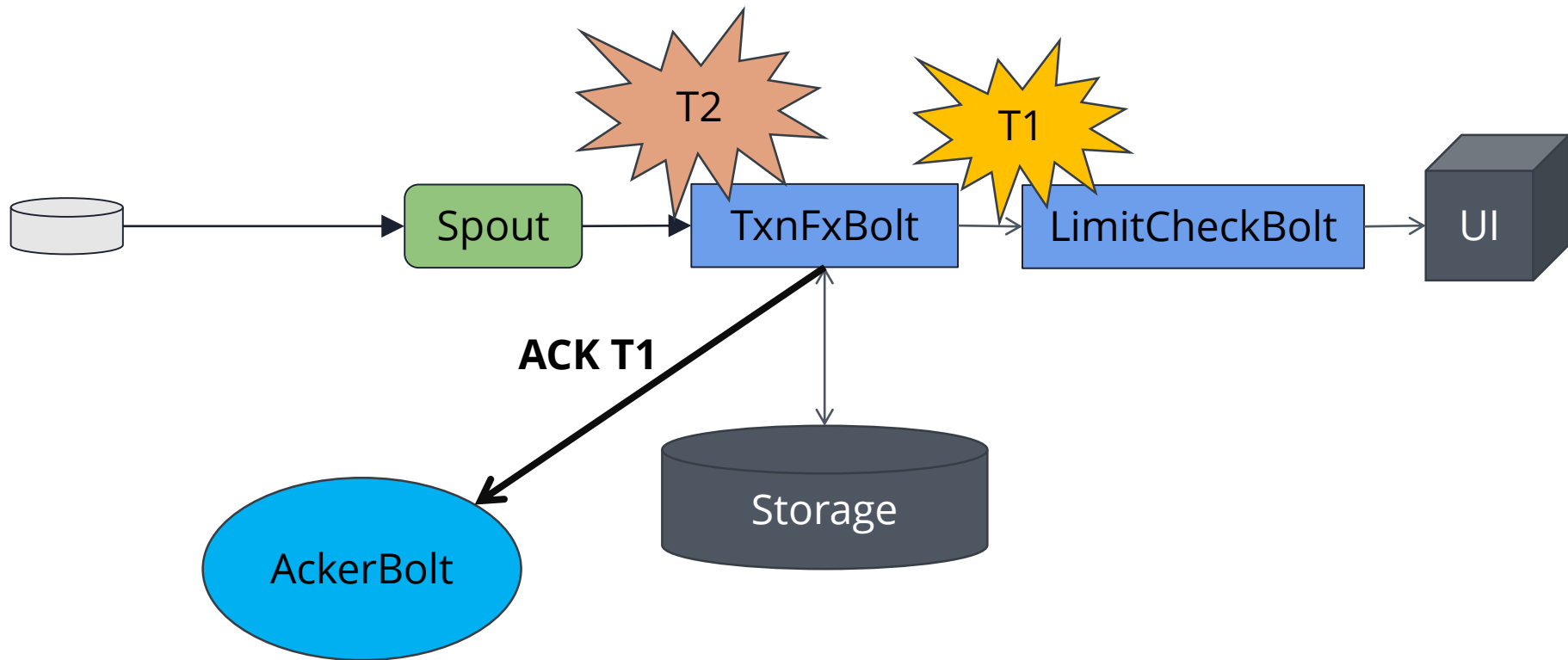




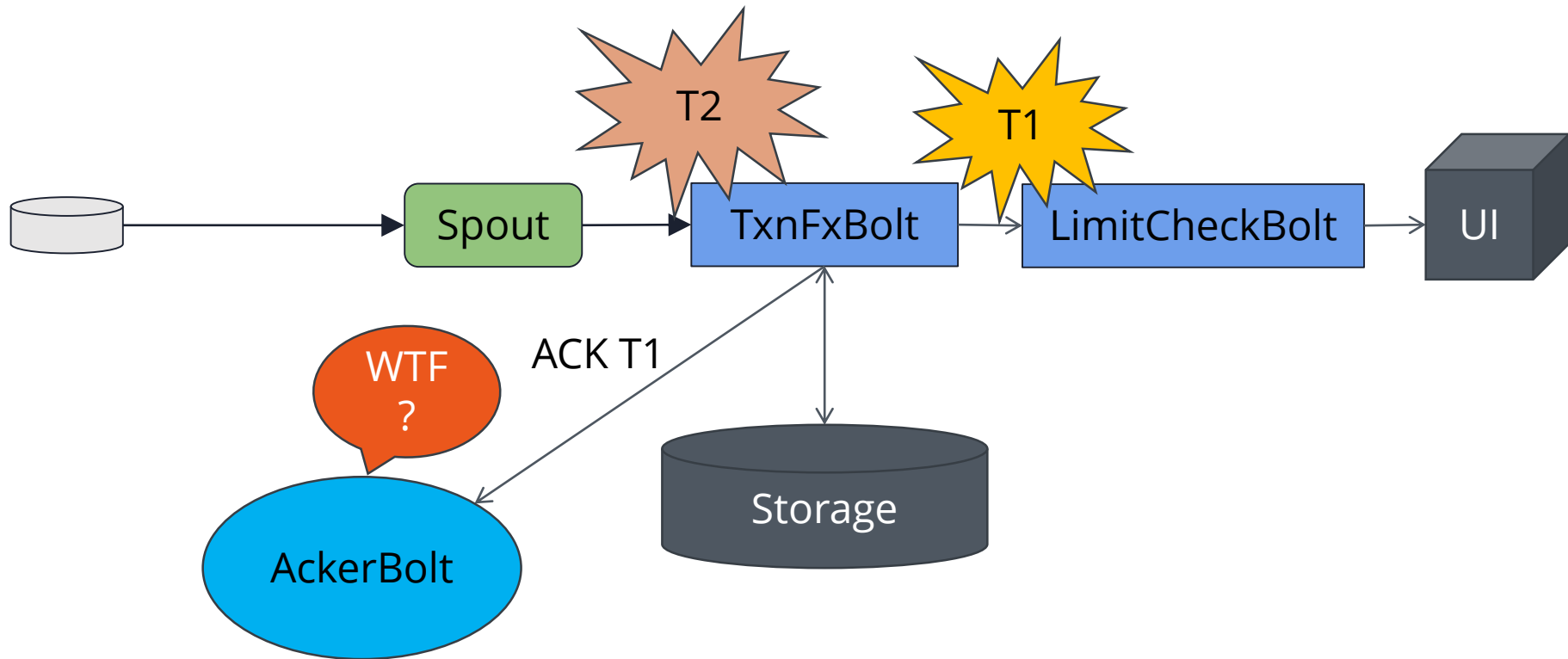


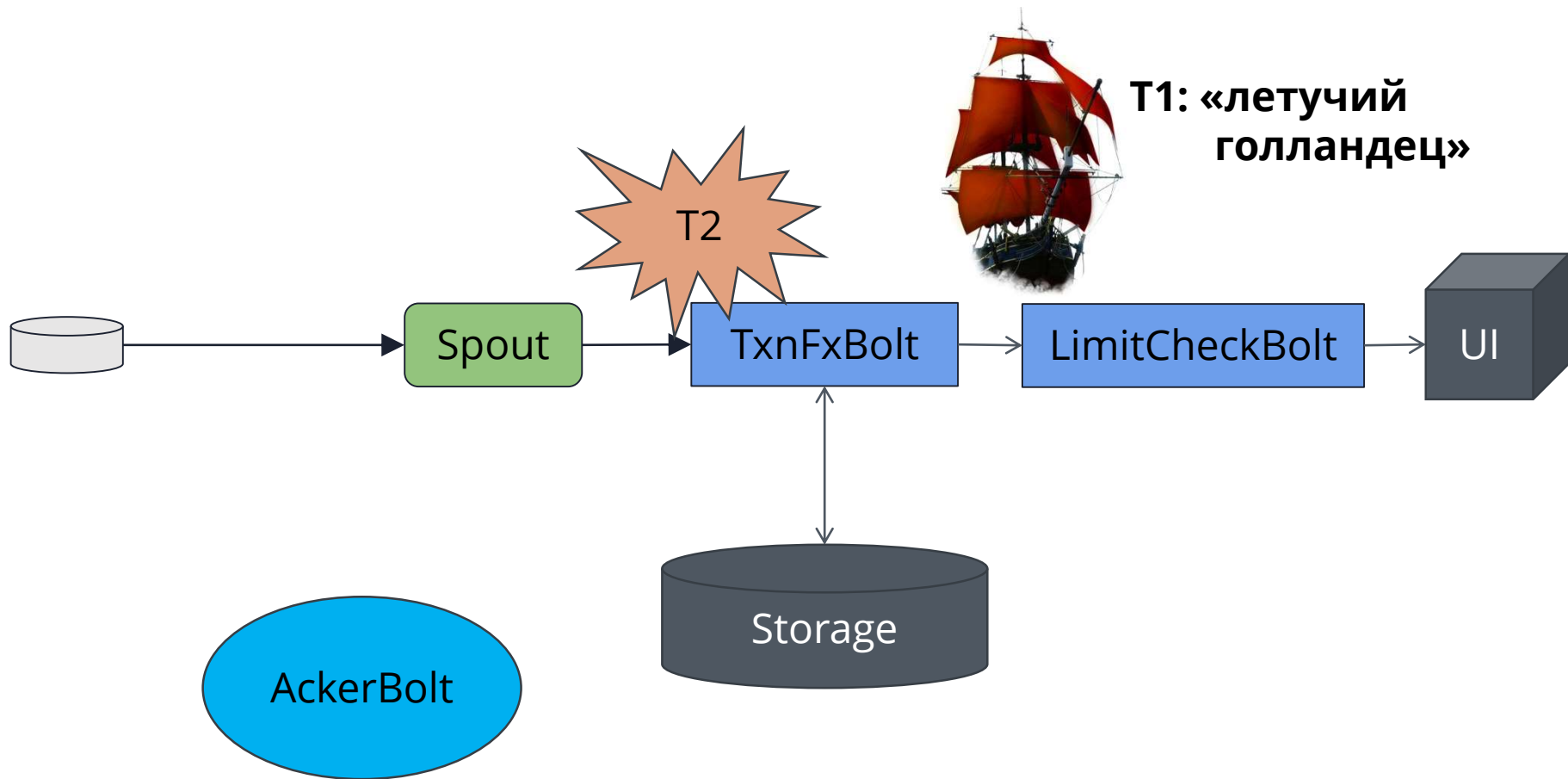


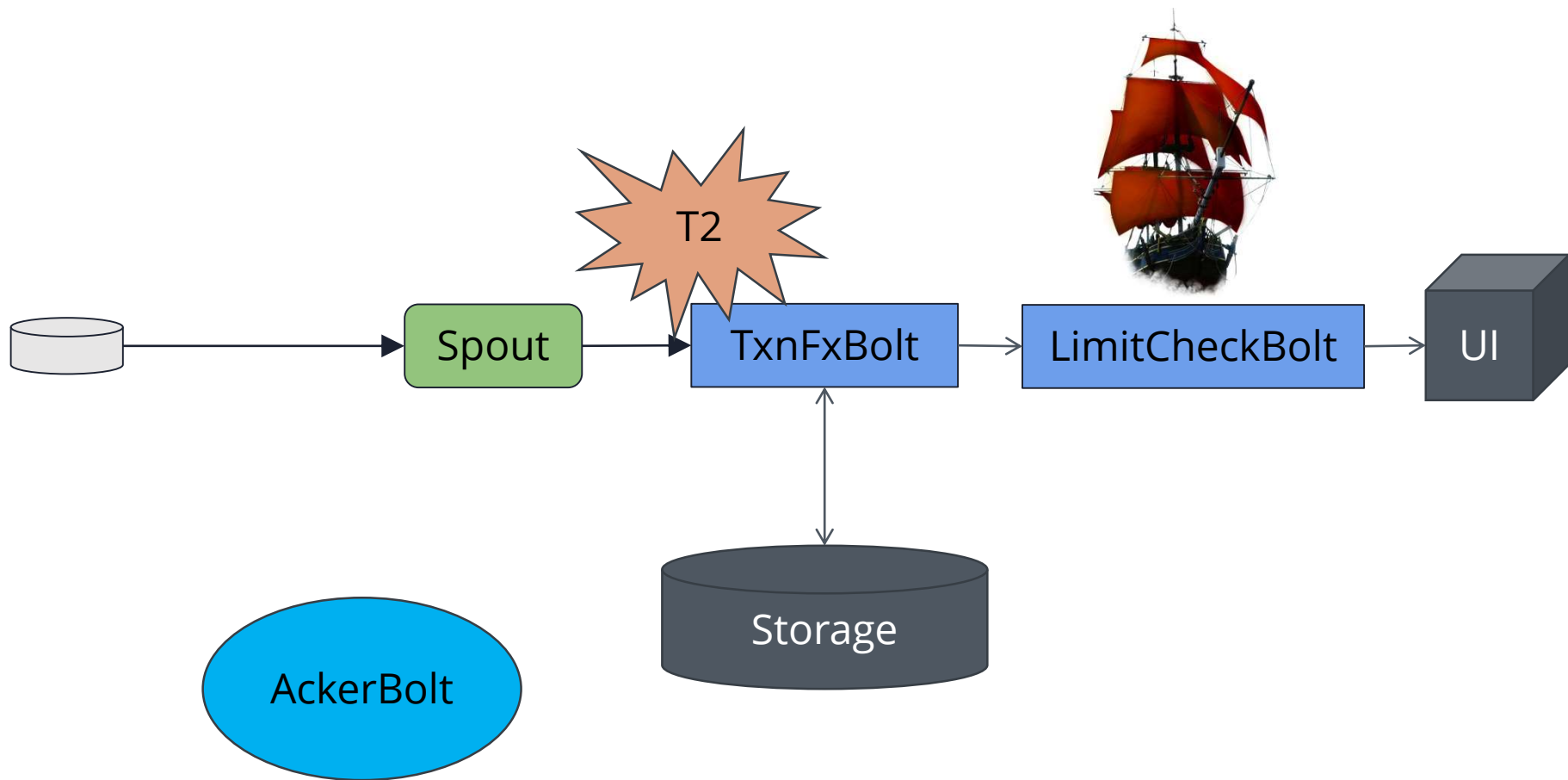
Проблема 2: летучий голландец

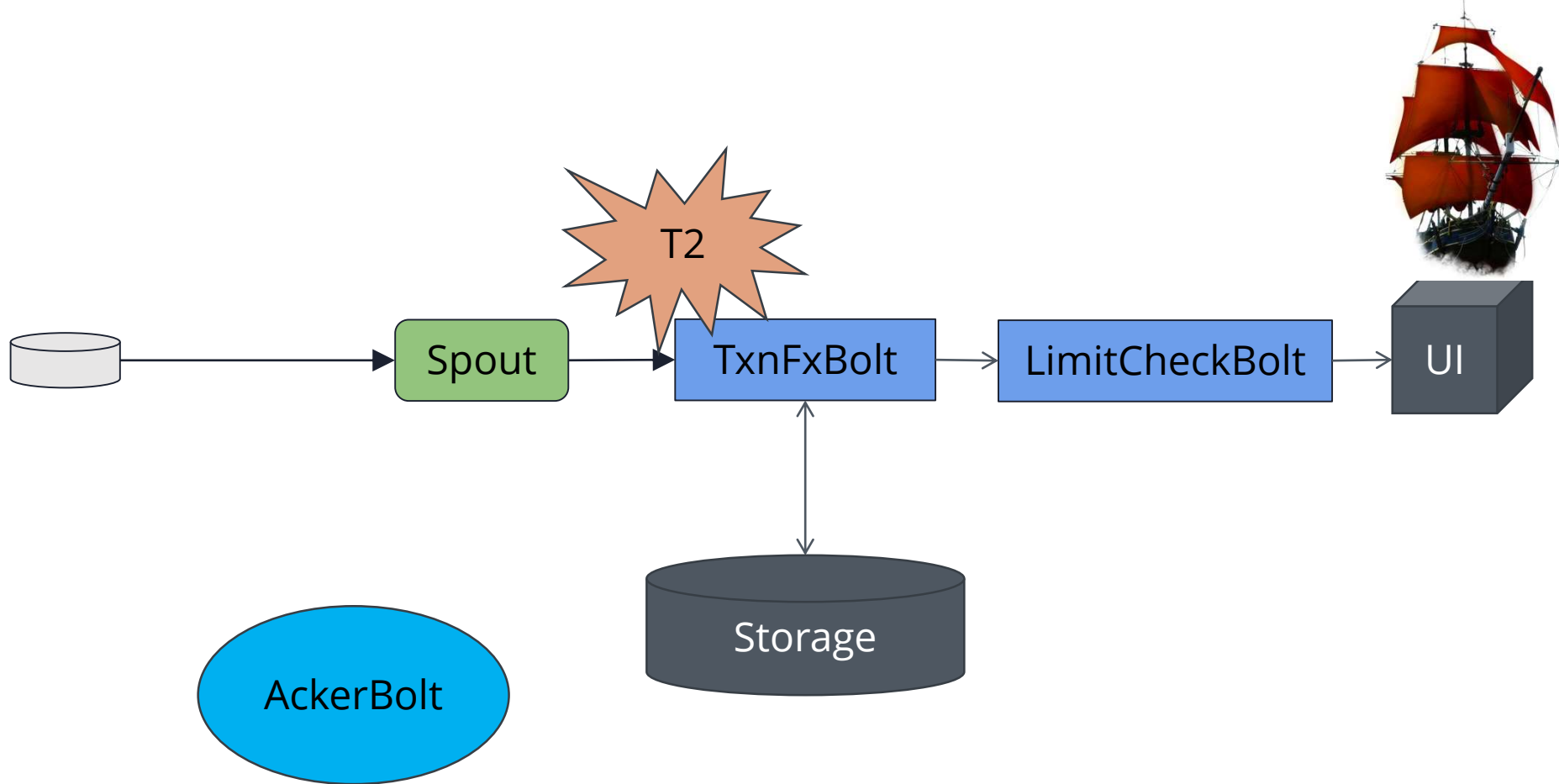


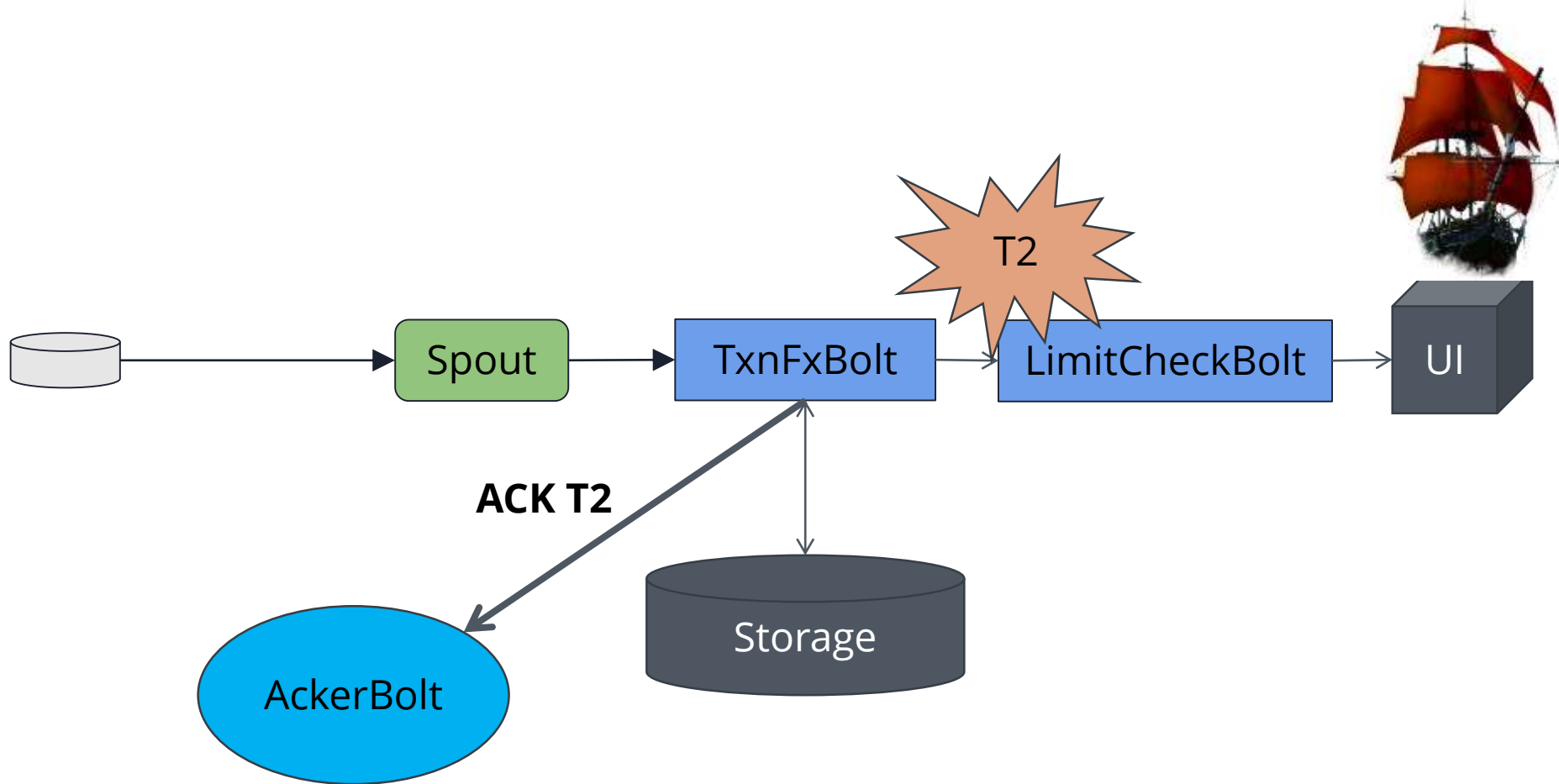
Проблема 2: летучий голландец

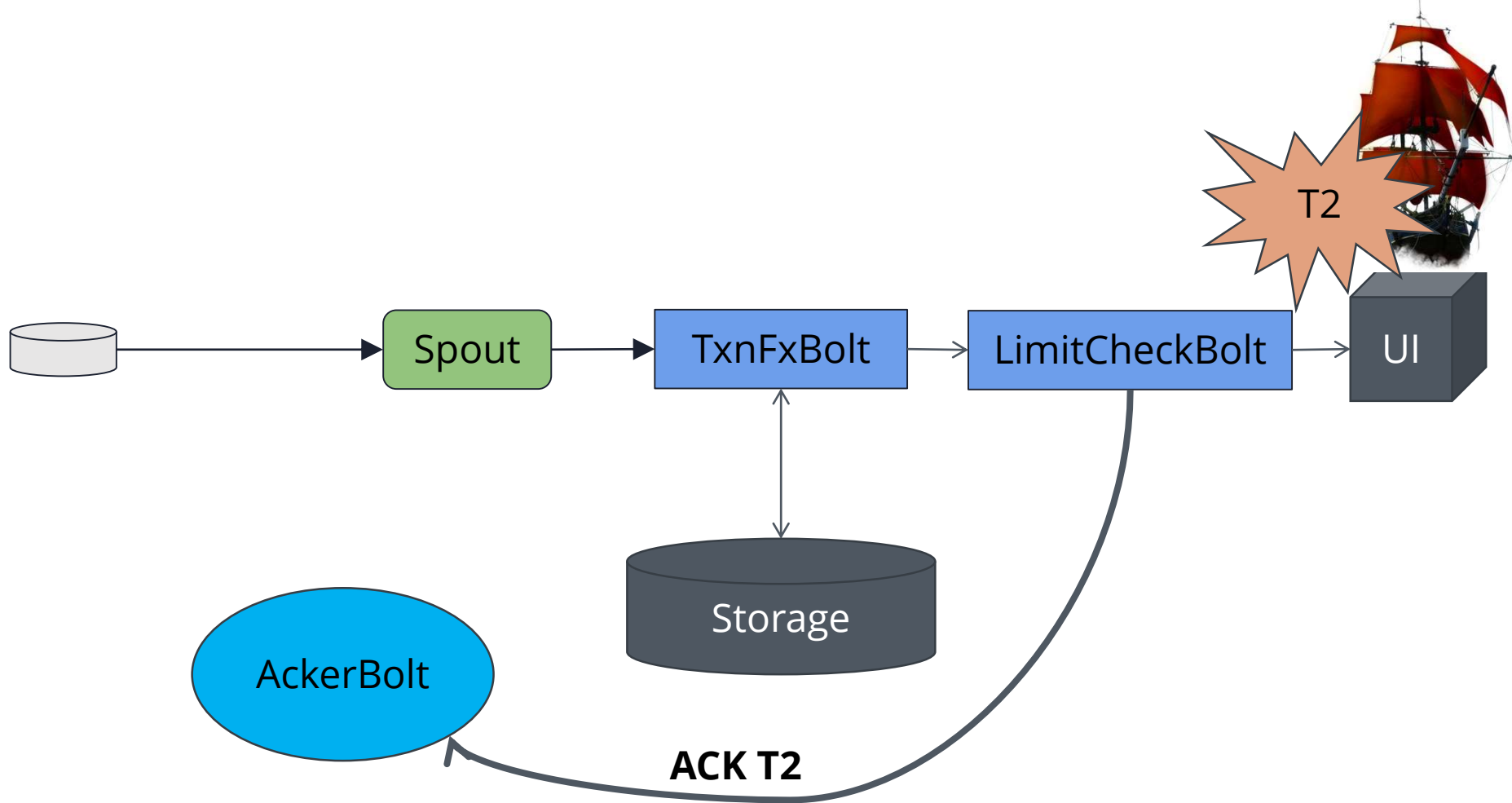


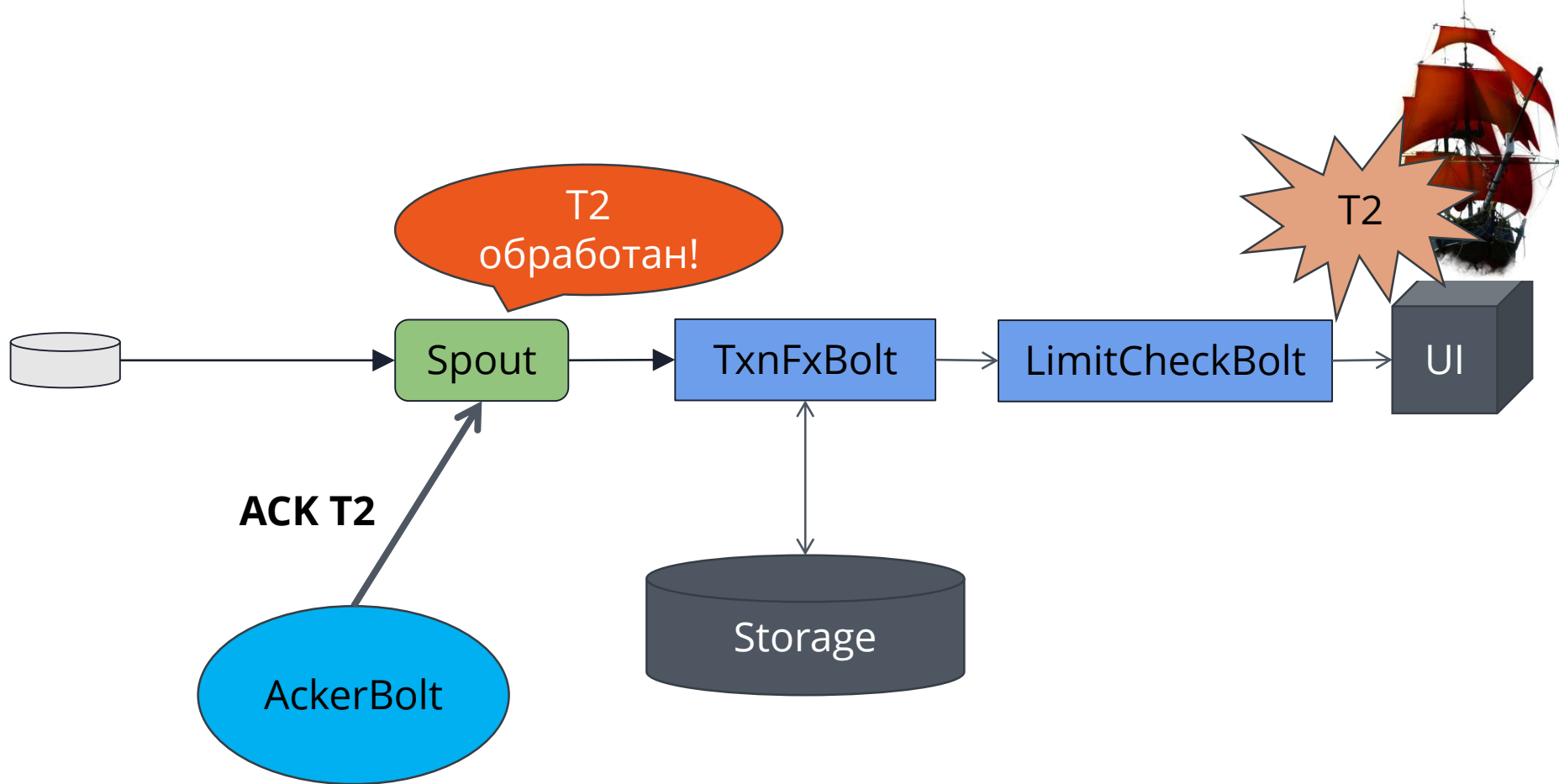








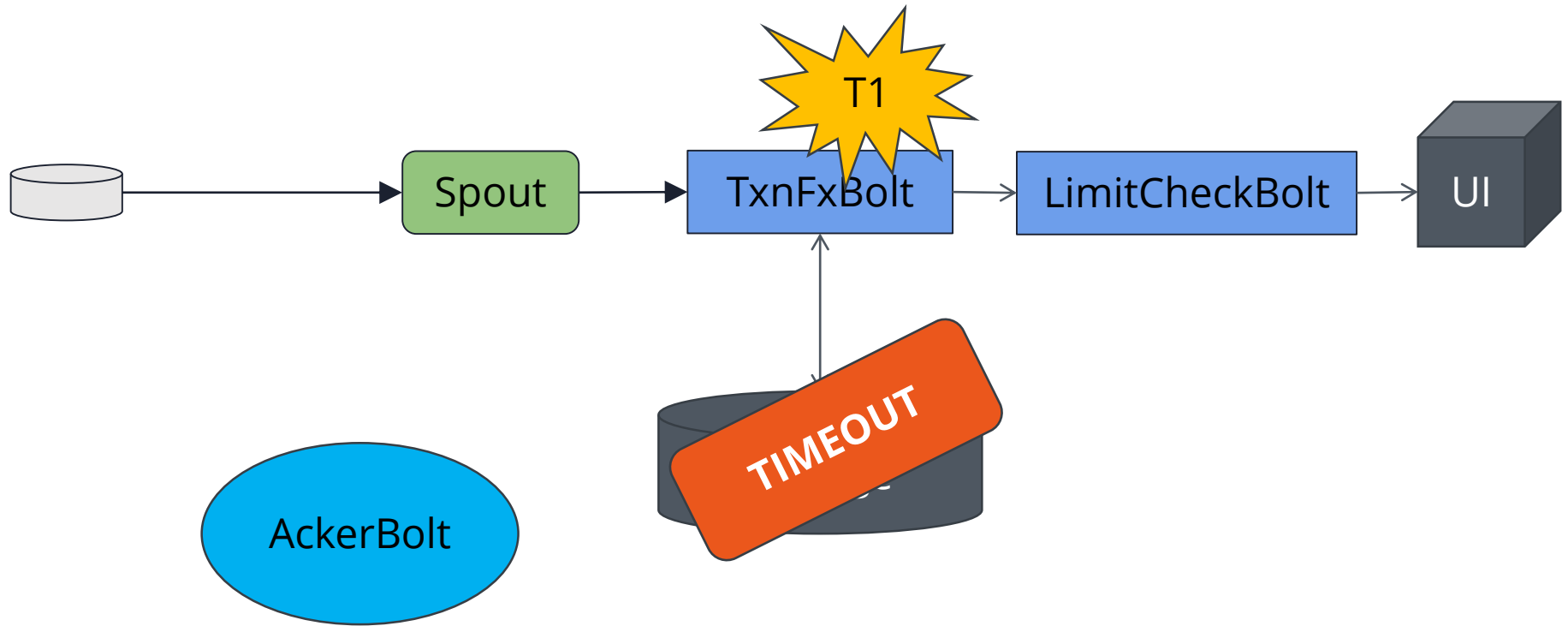


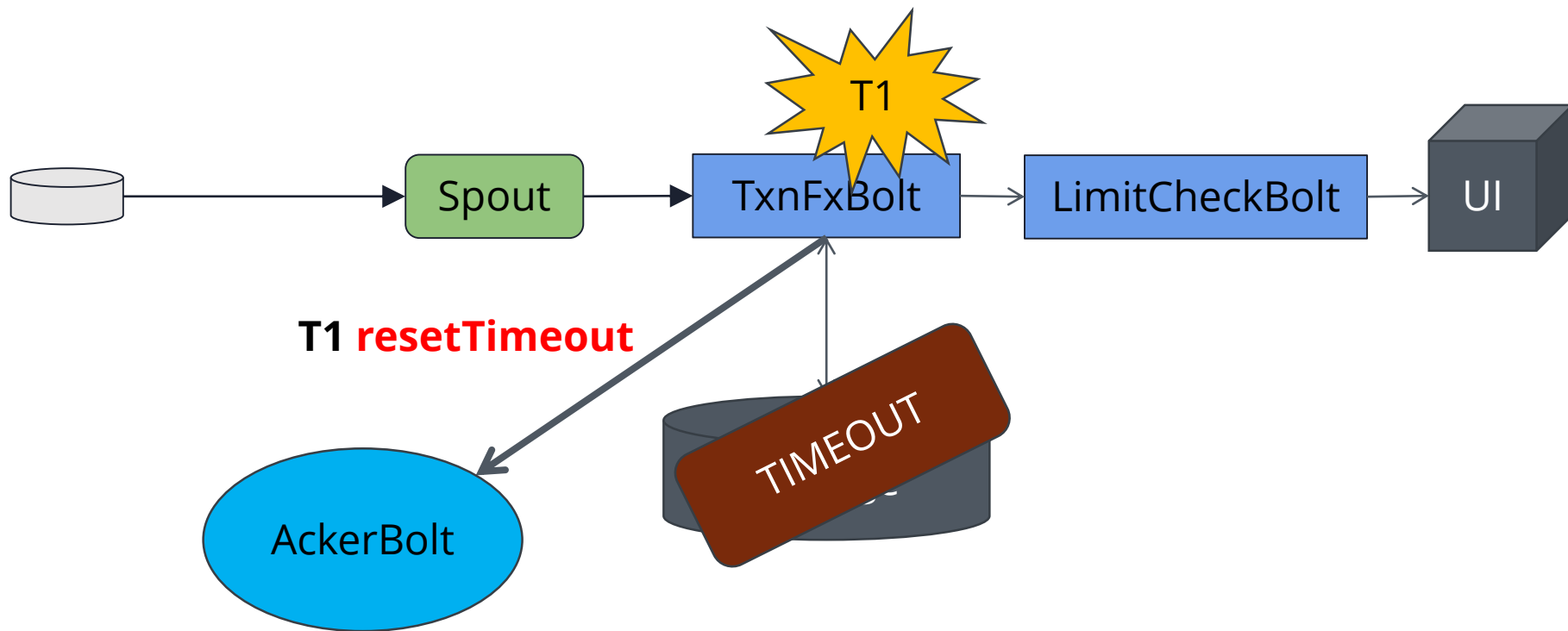


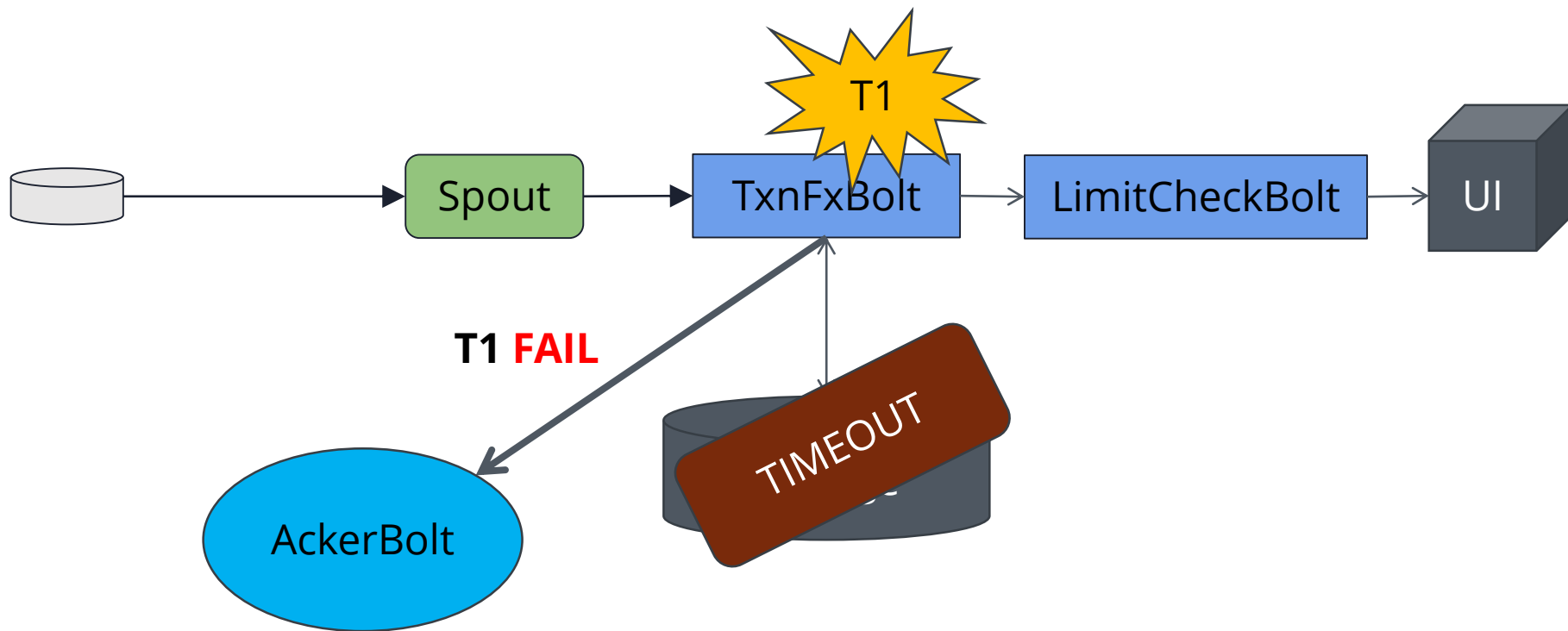
Решение?

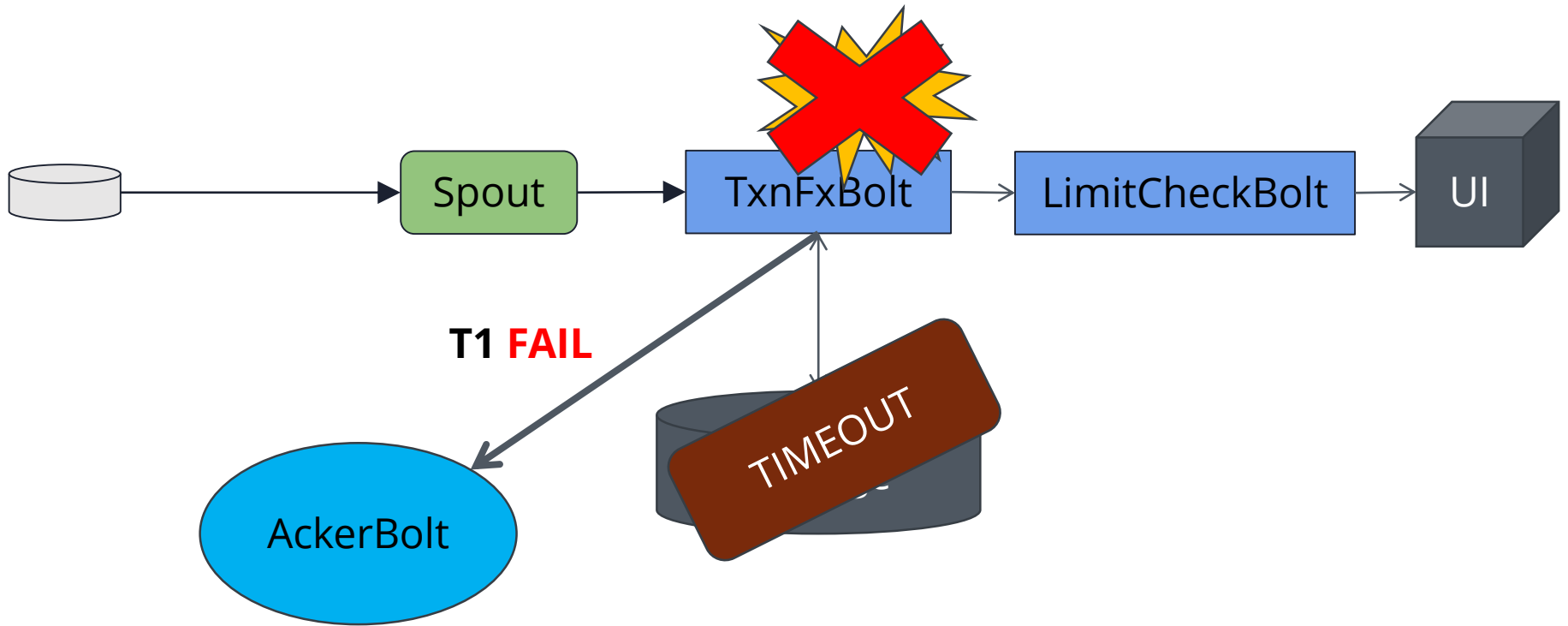
Решение?

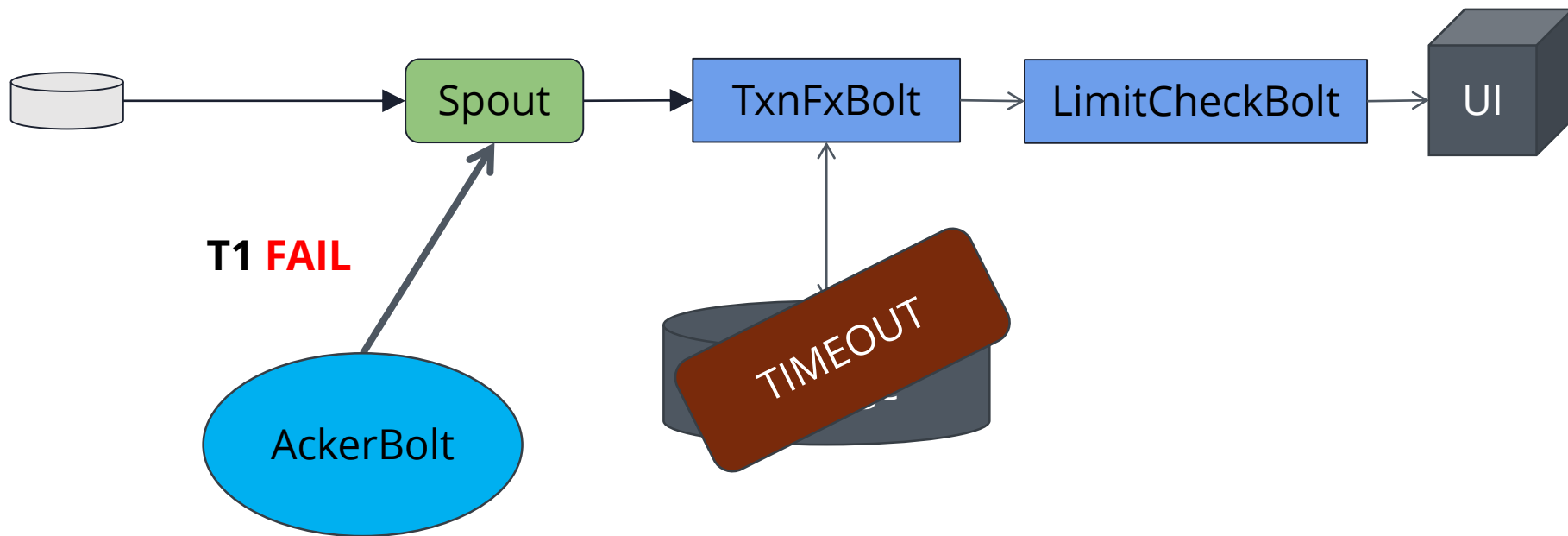
- ♦ `resetTimeout()`
- ♦ Отслеживать таймаут и посылать FAIL

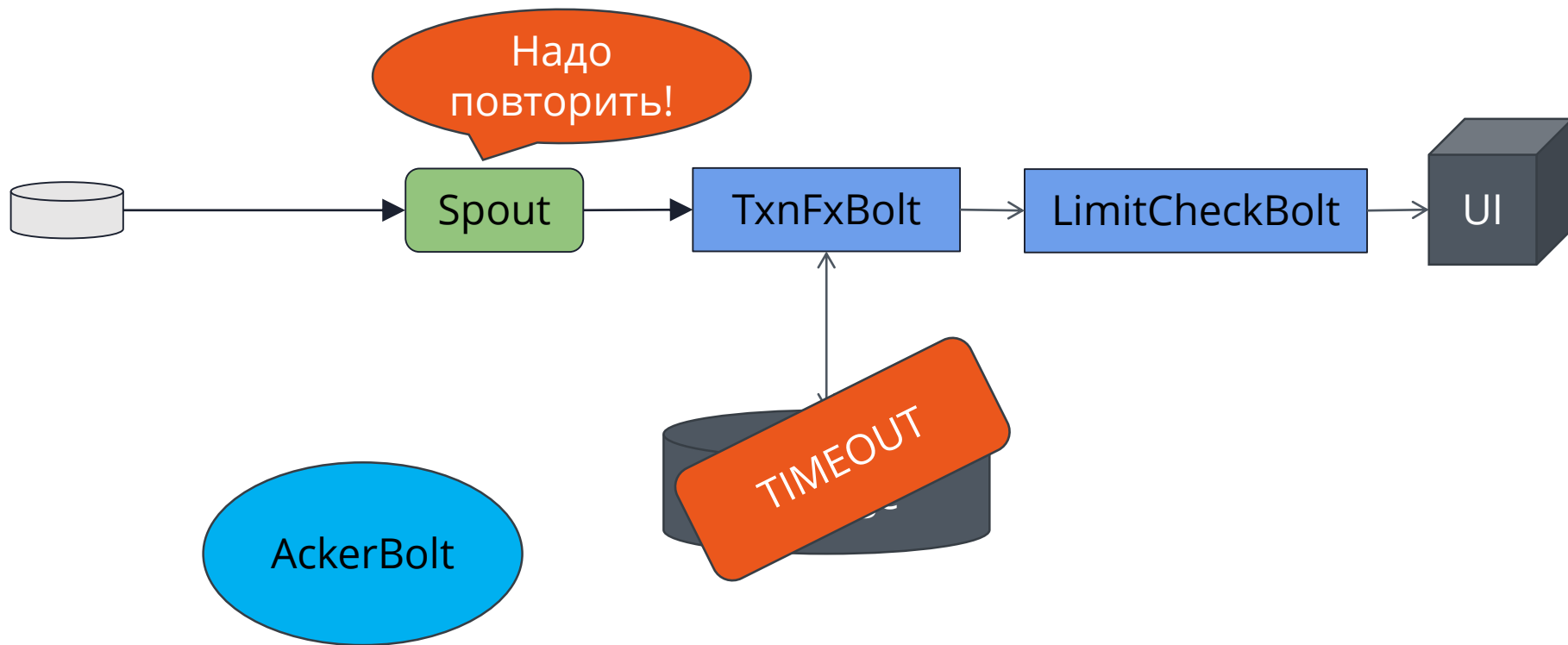


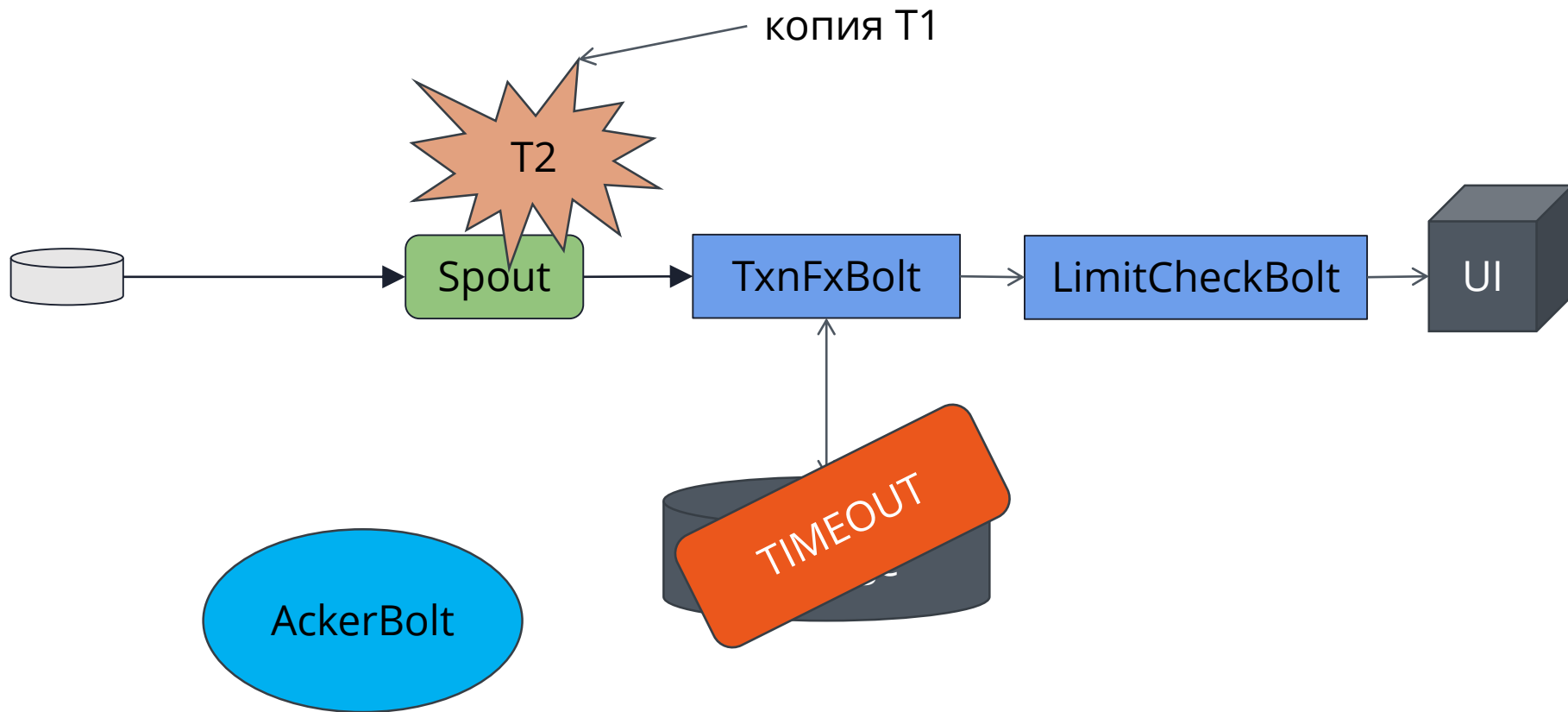


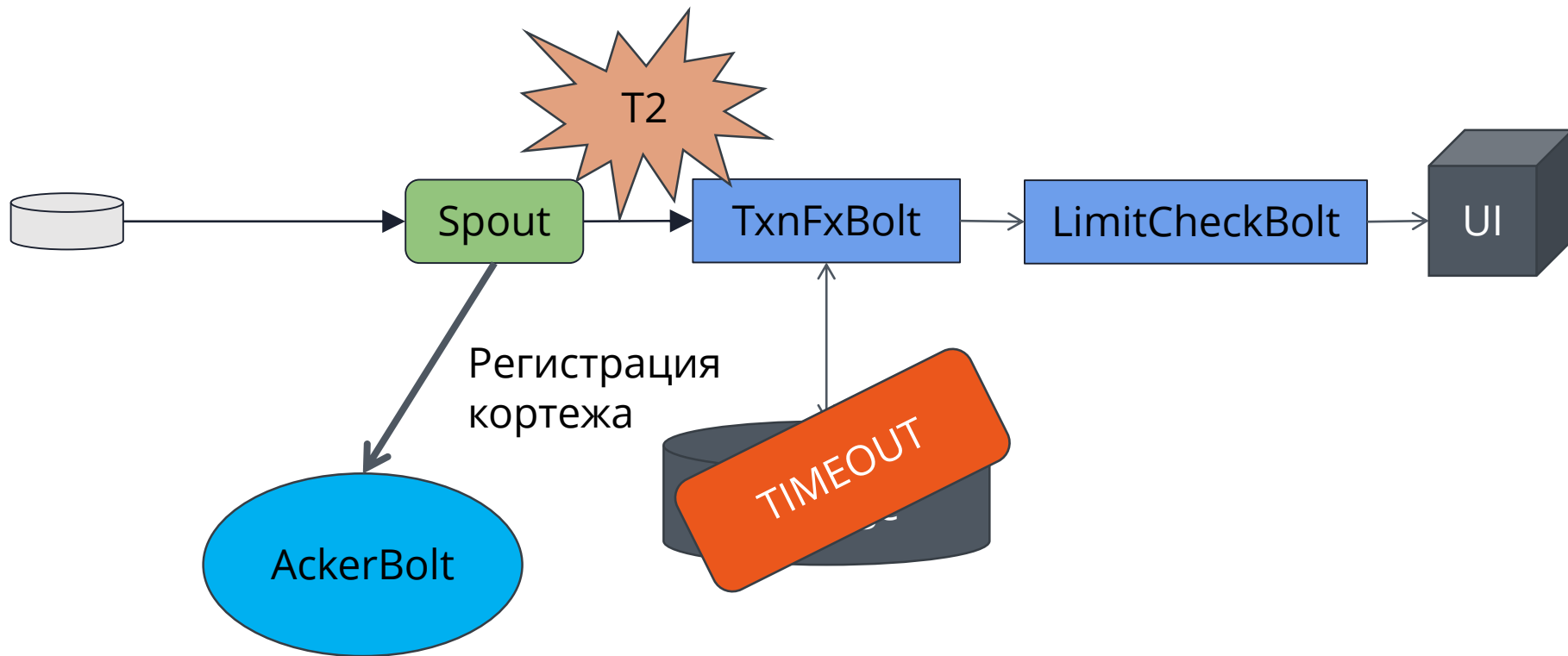


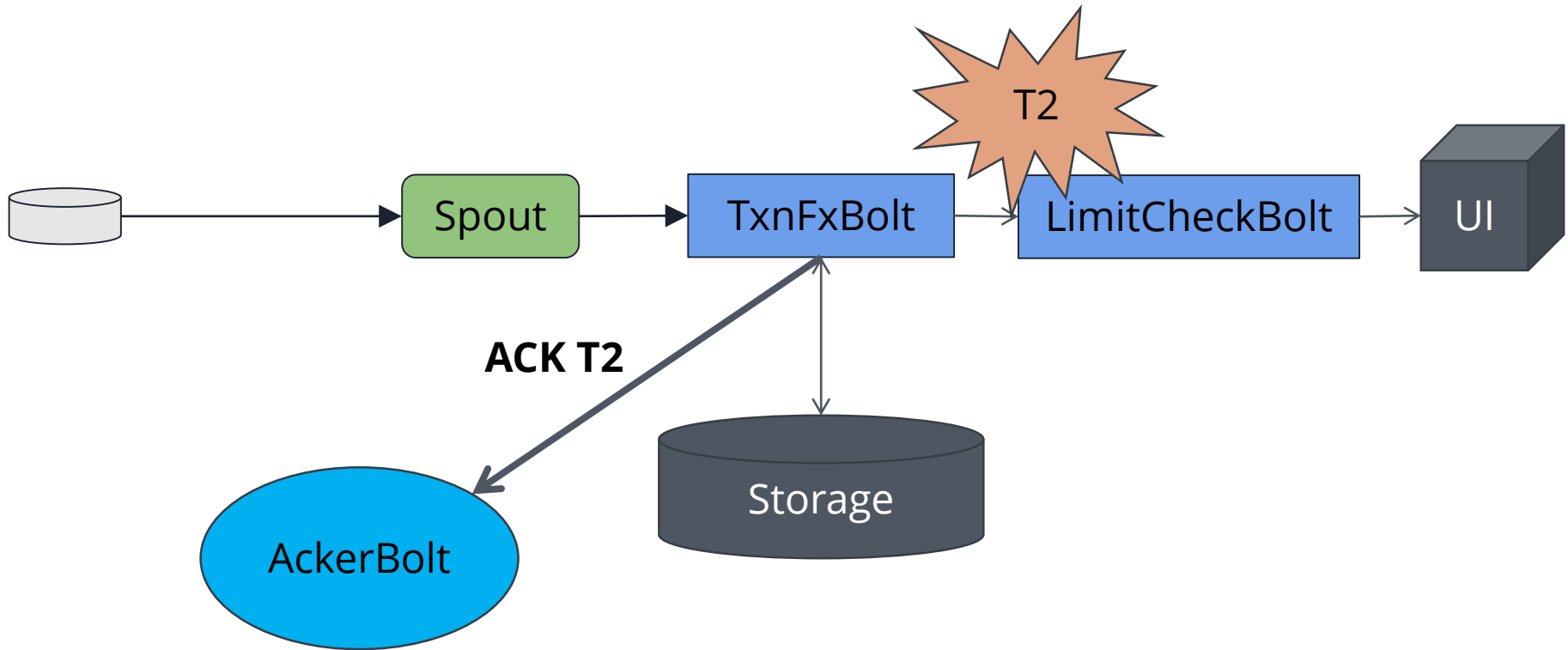


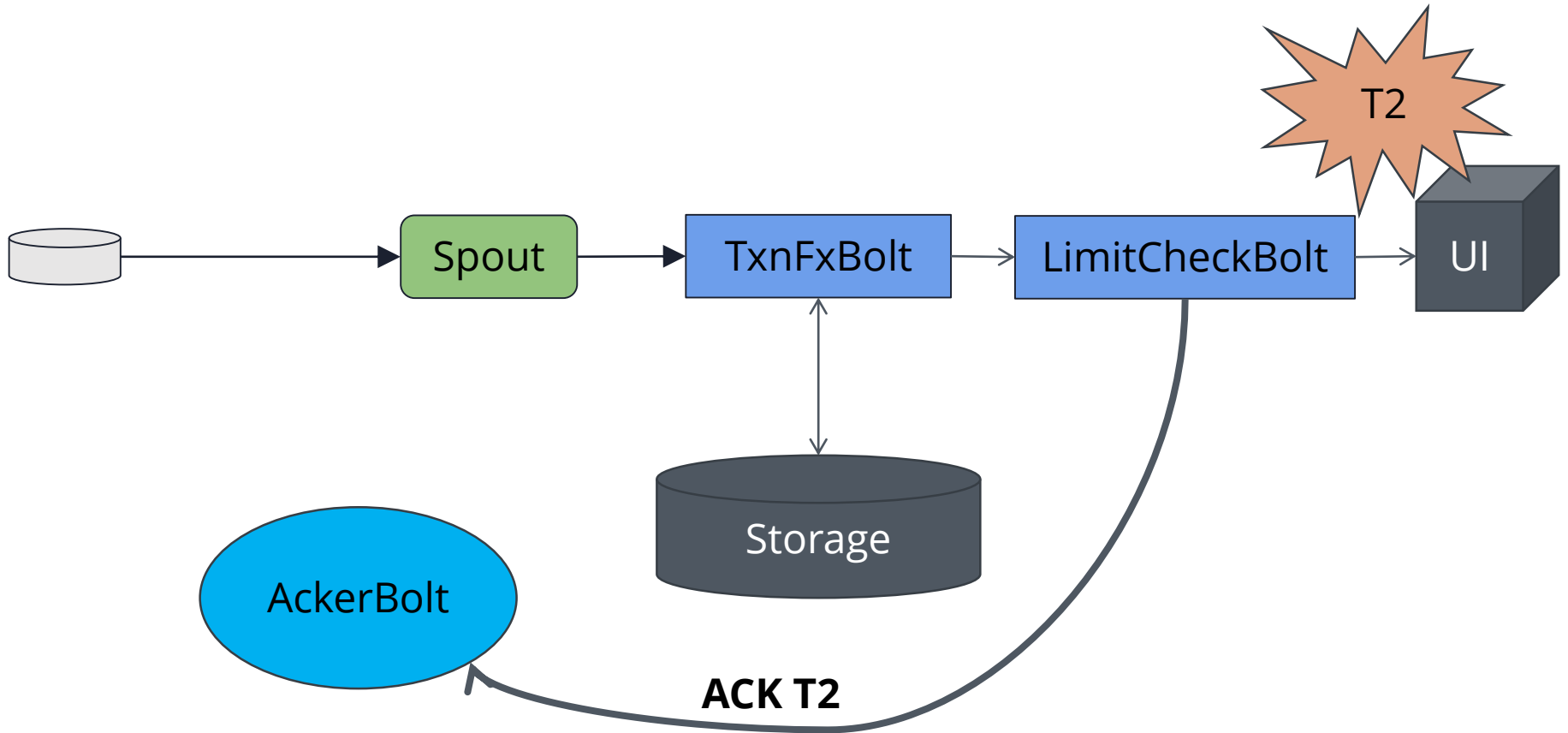


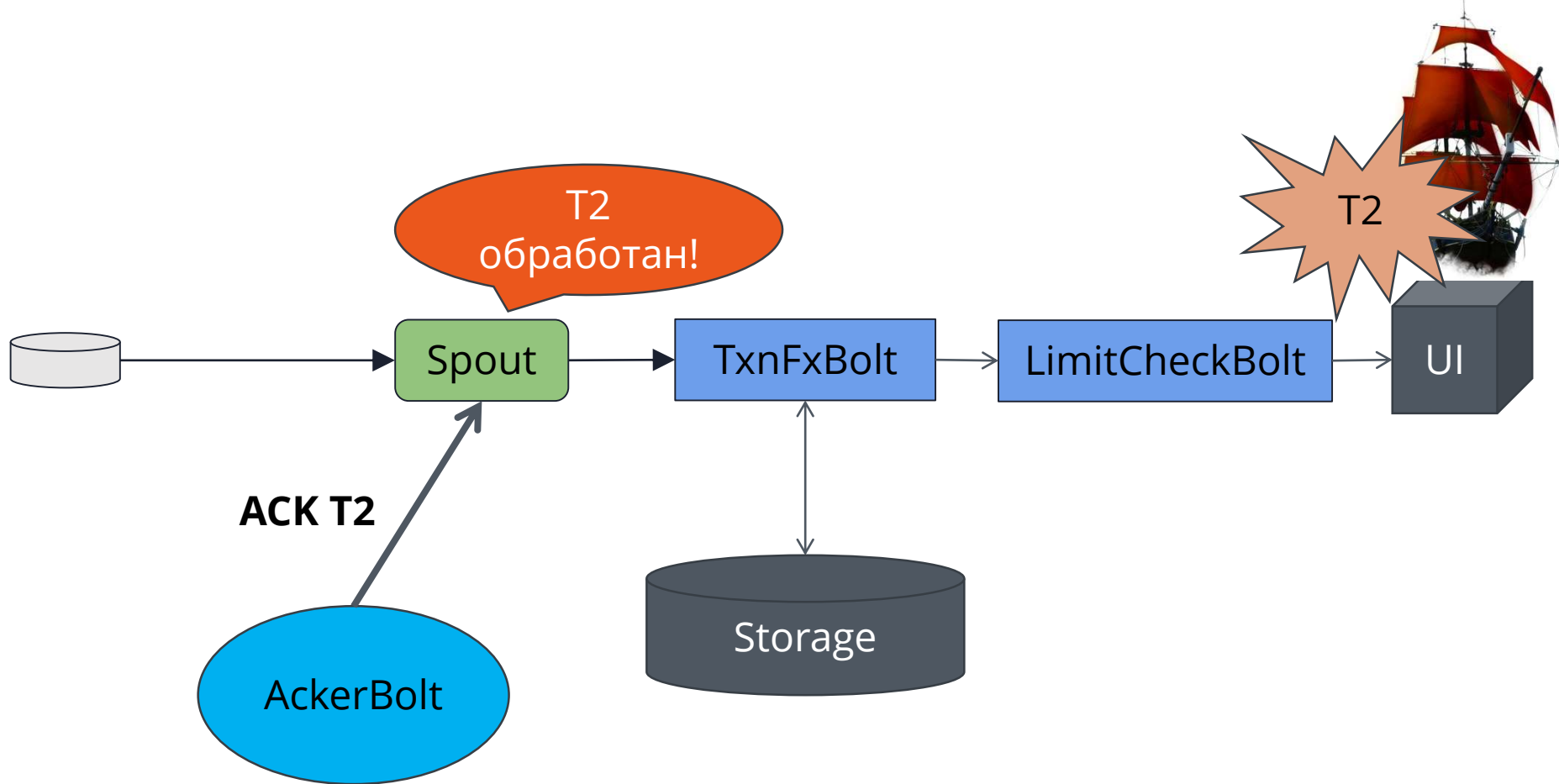












Конкуренты

Конкуренты

- ◆ Spark Streaming

Конкуренты

- ◆ Spark Streaming
- ◆ Heron by Twitter

Конкуренты

- ◆ Spark Streaming
- ◆ Heron by Twitter
- ◆ Apache Flink

Конкуренты

- ◆ Spark Streaming
- ◆ Heron by Twitter
- ◆ Apache Flink
- ◆ ...

СПАСИБО!

