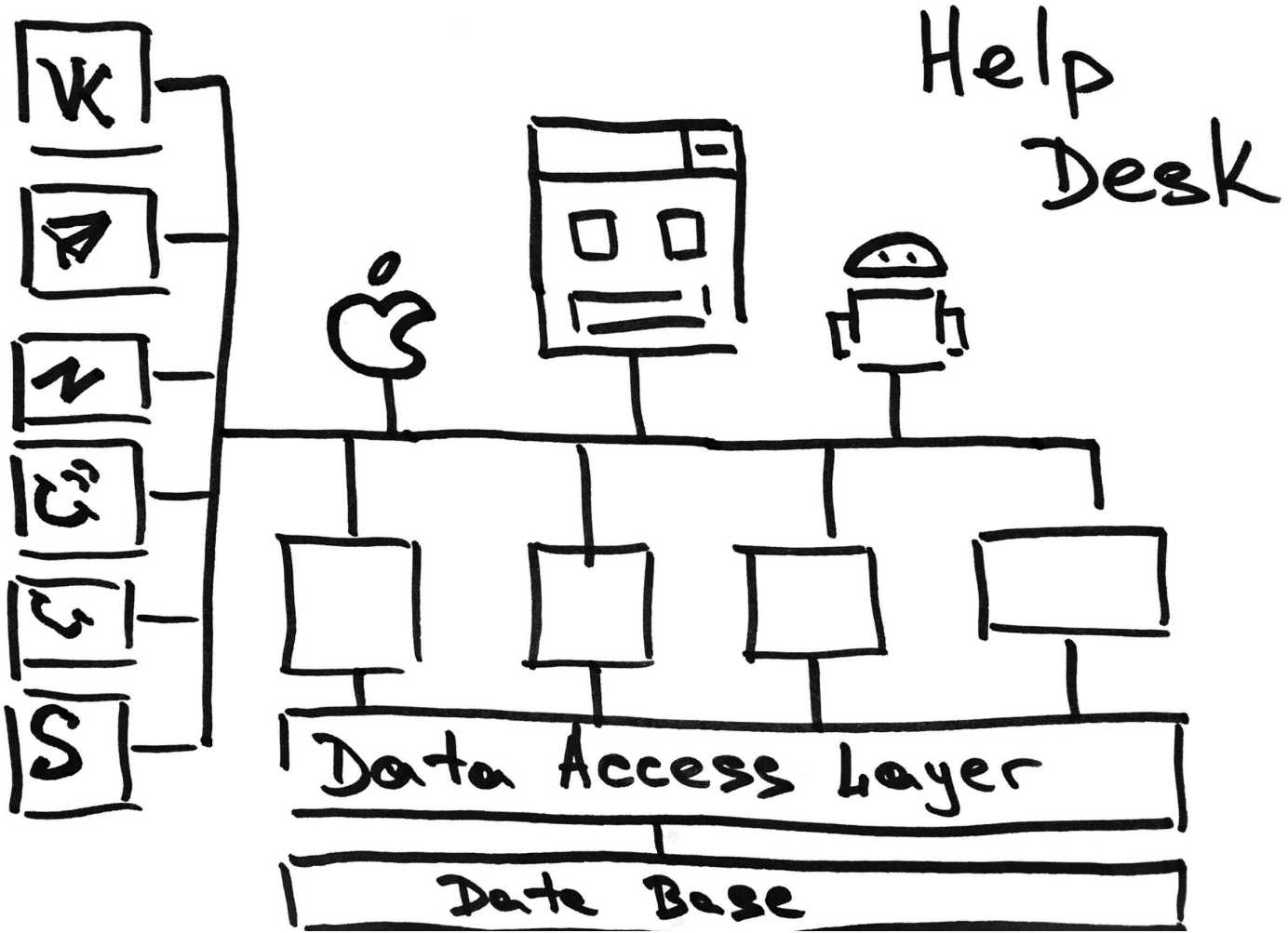
На пути к распределенной системе

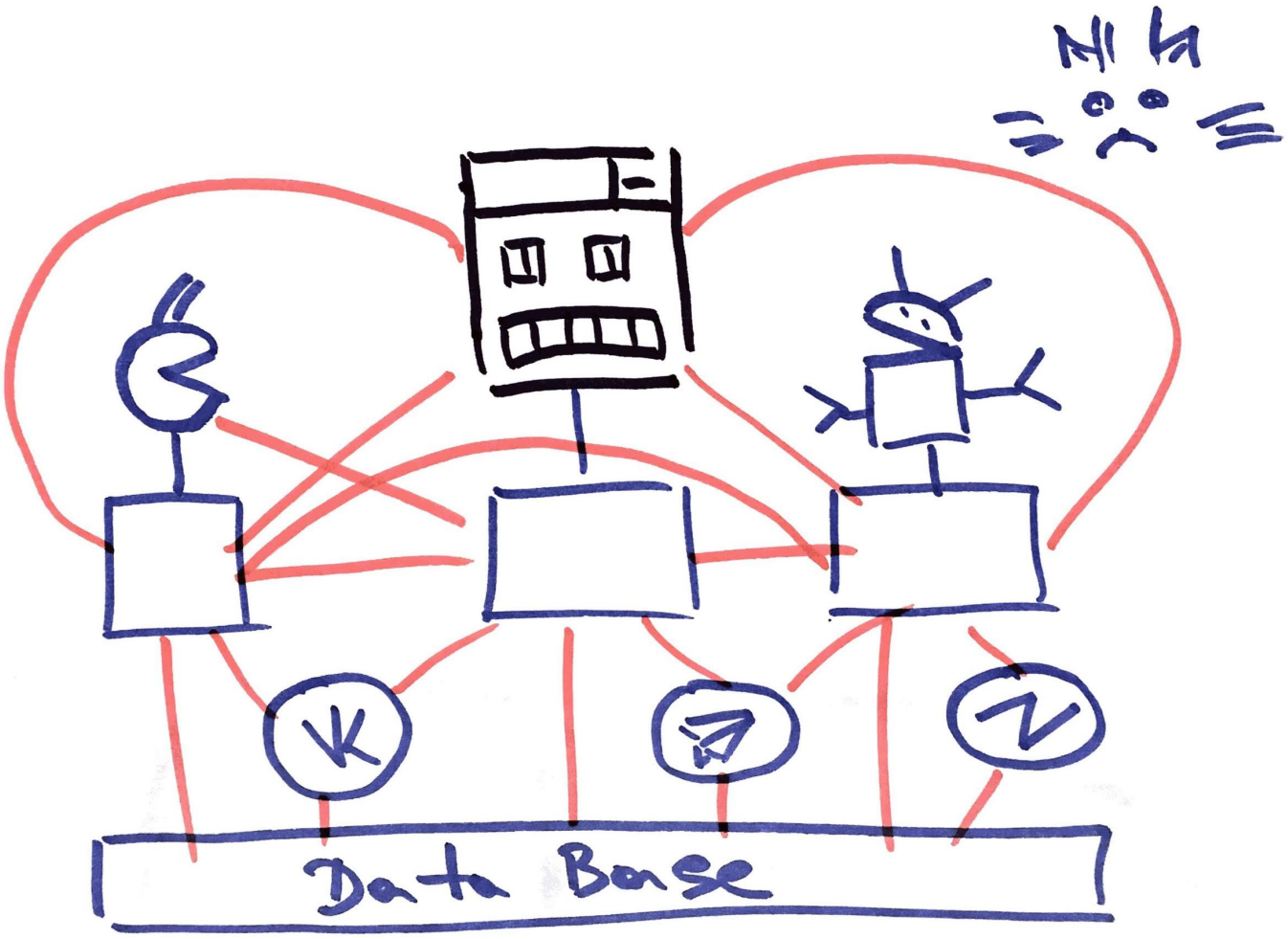
Роман Гордеев

Как и любая подобная история, эта история начинается с приложения...

"Такого обаянья ты исполнен..."


Help
Desk
Data Access Layer
Data Base

“Но мы отдёрнем занавес и покажем вам картину…”

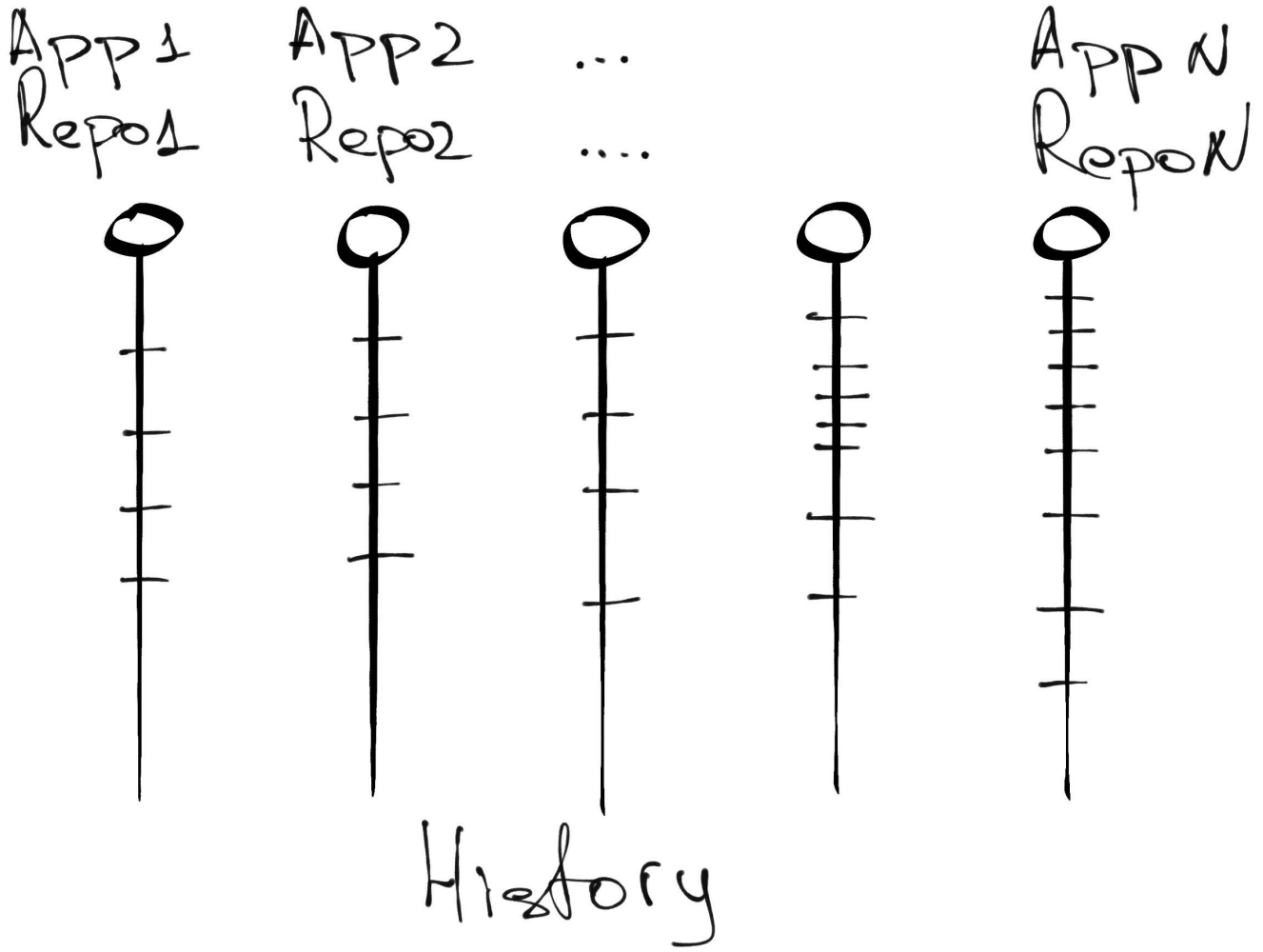

Data Base

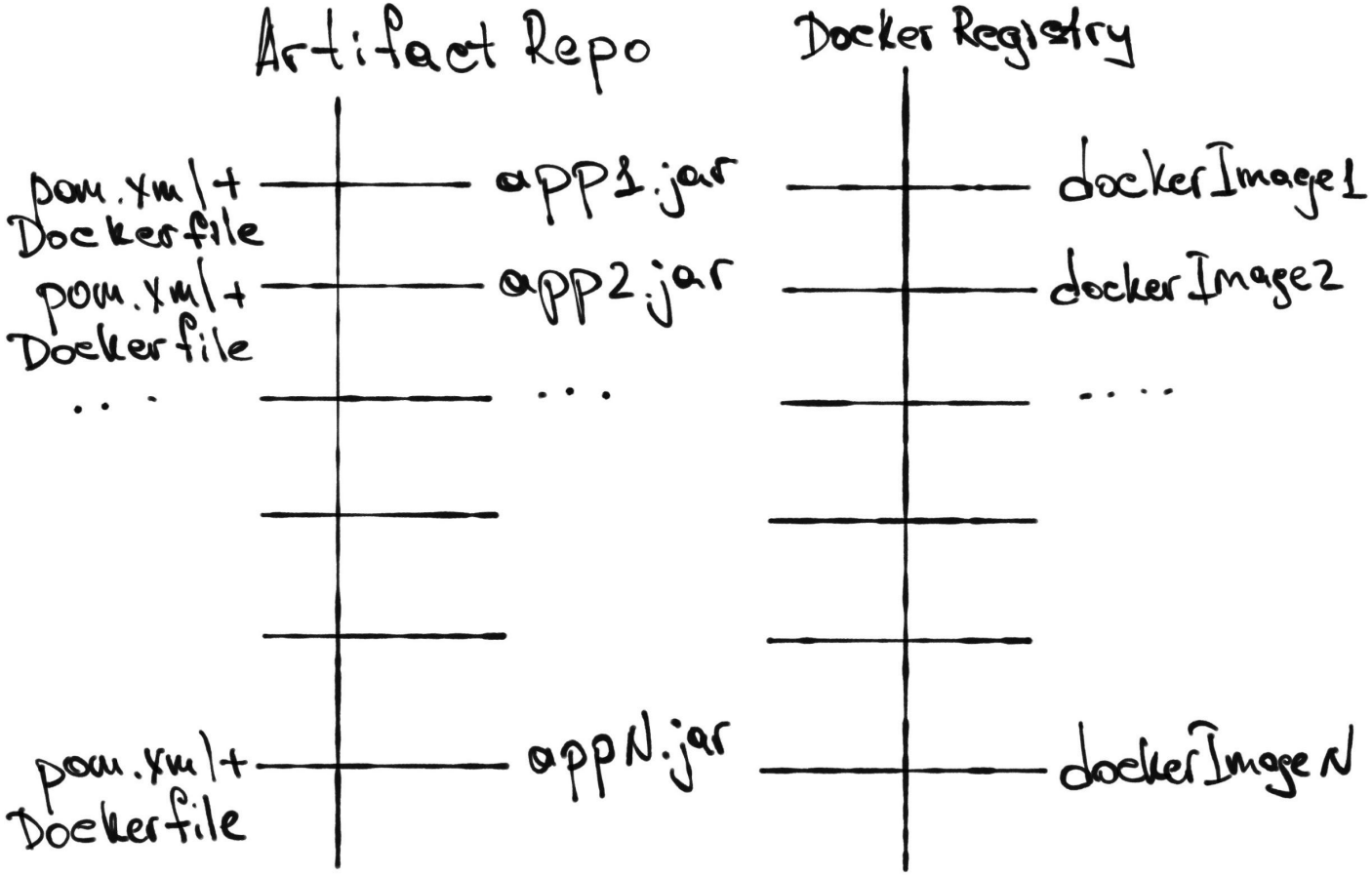
Цели:

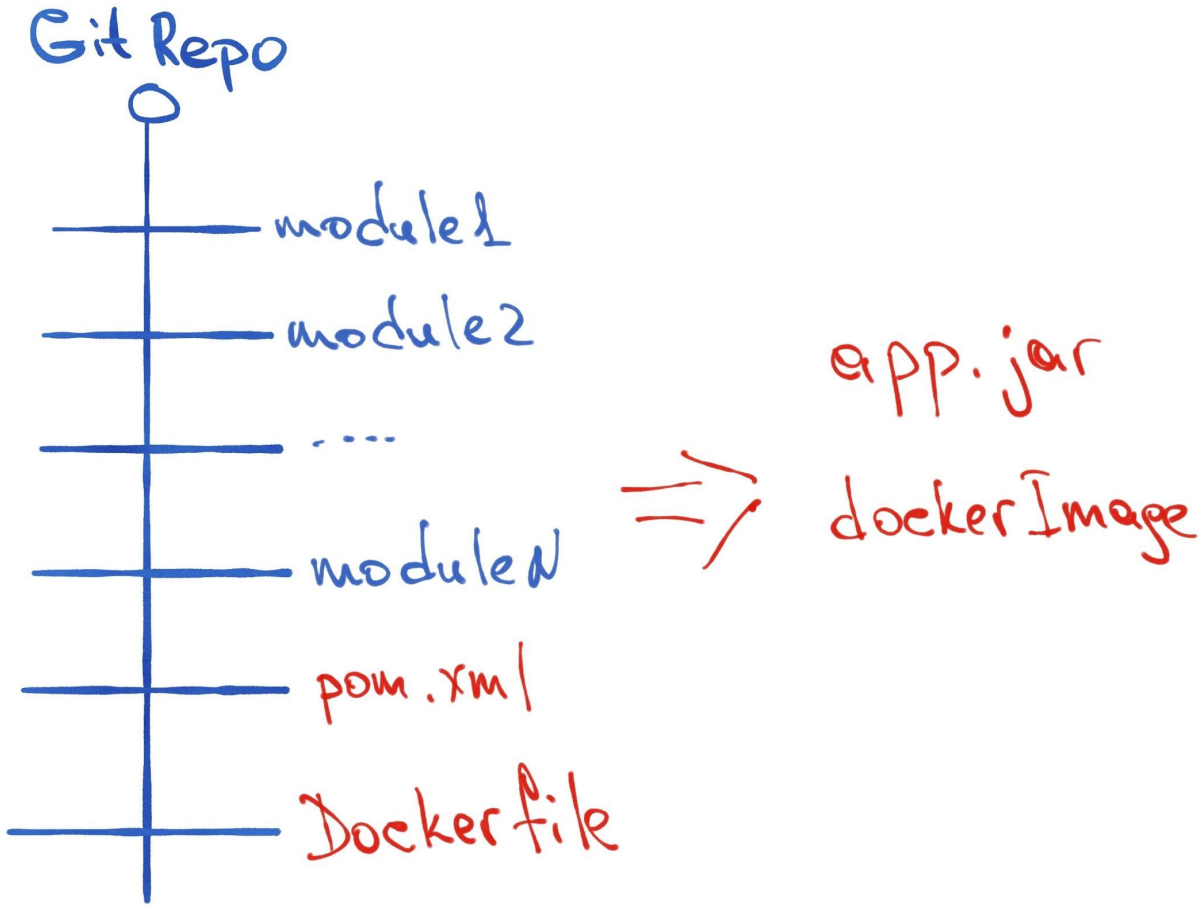
1. Как работать с унаследованным кодом
2. Чеклист для ORM
3. Простые решения для непростых ситуаций
4. Наивное проектирование
5. Инструментарий

Как организовать код?

- Монрепозиторий?
- Репозиторий на каждый подпроект?


App 1
Repo 1
App 2
Repo 2
...
....
App N
Repo N
History


Artifact Repo
Docker Registry
pom.xml +
Dockerfile
app1.jar
dockerImage1
pom.xml +
Dockerfile
app2.jar
dockerImage2
...
...
...
pom.xml +
Dockerfile
appN.jar
dockerImageN


Git Repo
module1
module2
....
moduleN
pom.xml
Dockerfile
app.jar
dockerImage

PROS

- Не нужна инфраструктура
- Просто собирать
- Просто запускать
- Возможны вариации на тему связывания
- Целостность

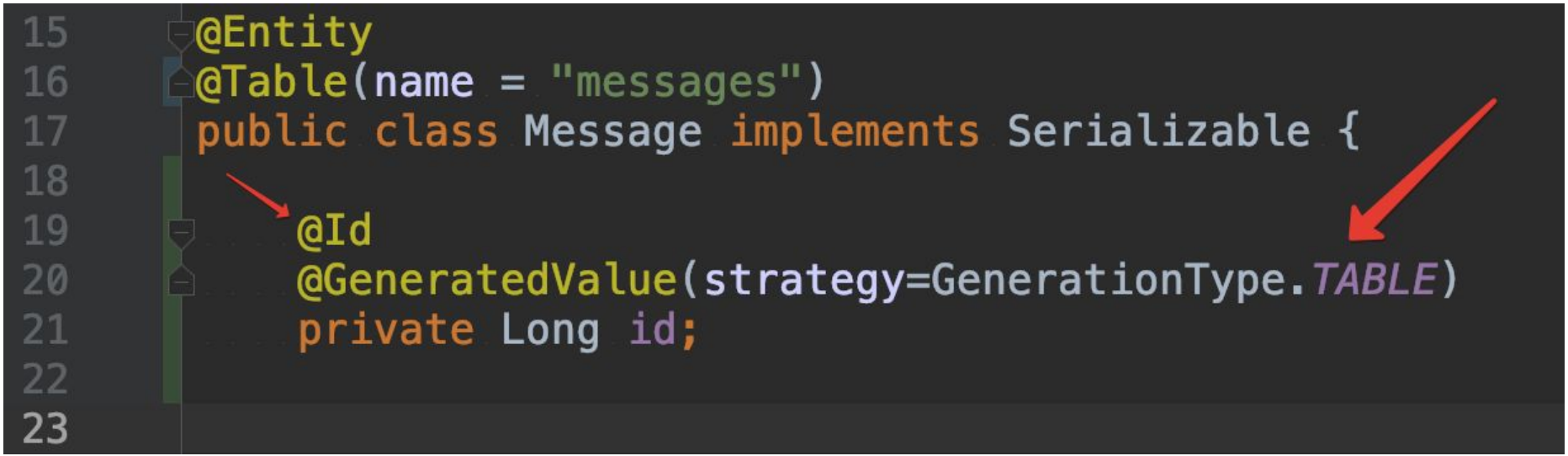
PROS

- Развитая инфраструктура
- Гибкое управление артефактами
- Ограничение связей
- Гибкое управление поставками

Простые трудности...

Чек-лист пользователя ORM

Никогда не используйте Table Generator!

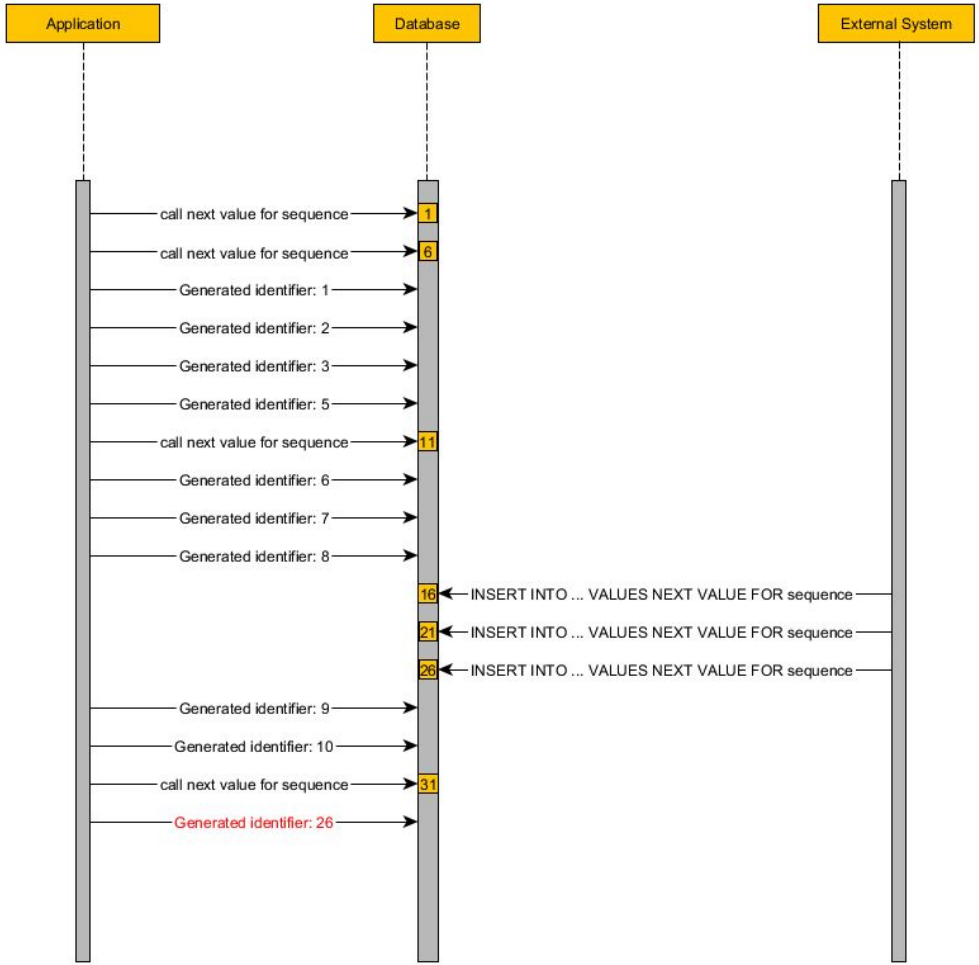

15 @Entity
16 @Table(name = "messages")
17 public class Message implements Serializable {
18
19 @Id
20 @GeneratedValue(strategy=GenerationType.TABLE)
21 private Long id;
22
23

```
SELECT tbl.next_val
FROM hibernate_sequences tbl
WHERE tbl.sequence_name=default
FOR UPDATE

INSERT INTO hibernate_sequences (sequence_name, next_val)
VALUES (default, 1)

UPDATE hibernate_sequences SET next_val=2
WHERE next_val=1 AND sequence_name=default

SELECT tbl.next_val
FROM hibernate_sequences tbl
WHERE tbl.sequence_name=default
FOR UPDATE

UPDATE hibernate_sequences SET next_val=3
WHERE next_val=2 AND sequence_name=default

INSERT INTO messages (id) values (1, 2)
```

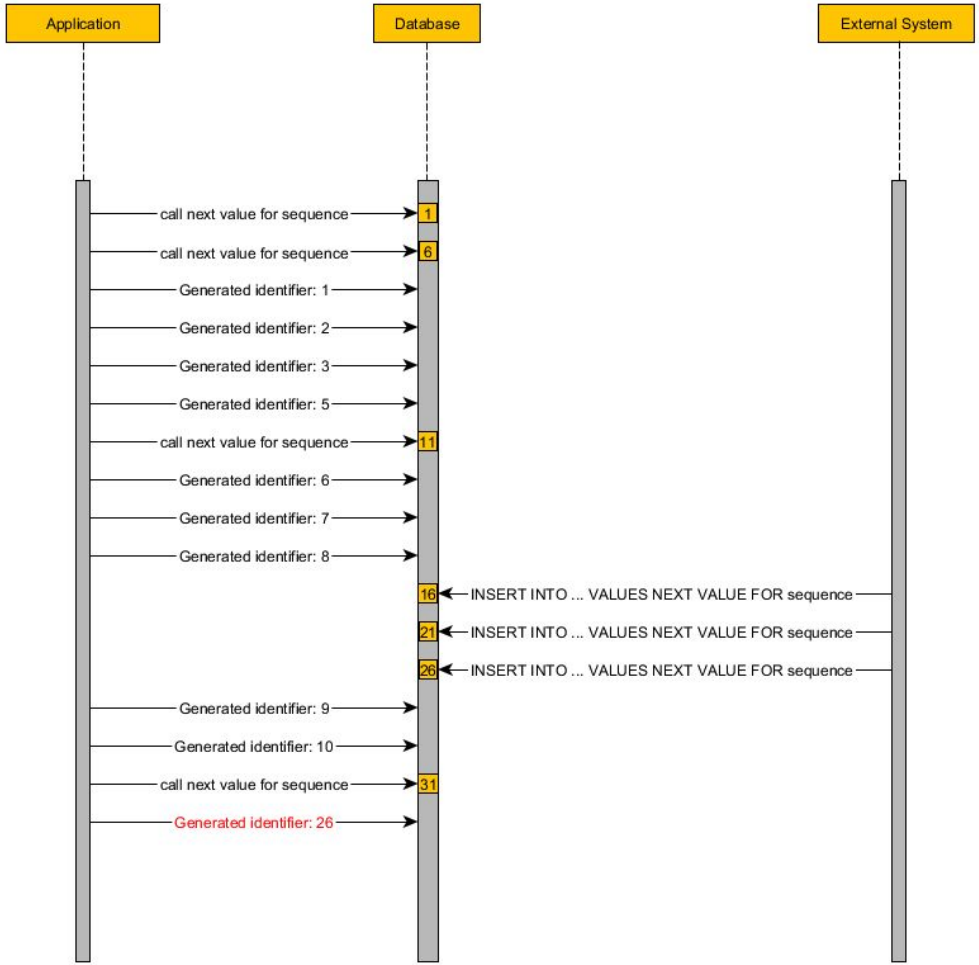
Используйте генераторы "последнего поколения"

```
92
93      properties.put("hibernate.id.new_generator_mappings", "true");
94
```


@Entity
@Table(name = "messages")
public class Message implements Serializable {
@Id
@GenericGenerator(
name = "sequenceGenerator",
strategy = "enhanced-sequence",
parameters = {
@org.hibernate.annotations.Parameter(
name = "optimizer",
value = "pooled"
),
@org.hibernate.annotations.Parameter(
name = "initial_value",
value = "1"
),
@org.hibernate.annotations.Parameter(
name = "increment_size",
value = "5"
)
}
)
@GeneratedValue(
strategy = GenerationType.SEQUENCE,
generator = "sequenceGenerator"
)
private Long id;


Application
Database
External System
call next value for sequence
1
call next value for sequence
6
Generated identifier: 1
Generated identifier: 2
Generated identifier: 3
Generated identifier: 5
call next value for sequence
11
Generated identifier: 6
Generated identifier: 7
Generated identifier: 8
16
INSERT INTO ... VALUES NEXT VALUE FOR sequence
21
INSERT INTO ... VALUES NEXT VALUE FOR sequence
26
INSERT INTO ... VALUES NEXT VALUE FOR sequence
Generated identifier: 9
Generated identifier: 10
call next value for sequence
31
Generated identifier: 26


@Entity
@Table(name = "messages")
public class Message implements Serializable {
@Id
@GenericGenerator(
name = "sequenceGenerator",
strategy = "enhanced-sequence",
parameters = {
@org.hibernate.annotations.Parameter(
name = "optimizer",
value = "pooled-lo"
),
@org.hibernate.annotations.Parameter(
name = "initial_value",
value = "1"
),
@org.hibernate.annotations.Parameter(
name = "increment_size",
value = "5"
)
}
)
@GeneratedValue(
strategy = GenerationType.SEQUENCE,
generator = "sequenceGenerator"
)
private Long id;


Application
Database
External System
call next value for sequence
1
call next value for sequence
6
Generated identifier: 1
Generated identifier: 2
Generated identifier: 3
Generated identifier: 5
call next value for sequence
11
Generated identifier: 6
Generated identifier: 7
Generated identifier: 8
16
INSERT INTO ... VALUES NEXT VALUE FOR sequence
21
INSERT INTO ... VALUES NEXT VALUE FOR sequence
26
INSERT INTO ... VALUES NEXT VALUE FOR sequence
Generated identifier: 9
Generated identifier: 10
call next value for sequence
31
Generated identifier: 26

Сконфигурируйте Batching

```
64
65  properties.put("hibernate.jdbc.batch_size", 400);
66  properties.put("hibernate.order_inserts", "true");
67  properties.put("hibernate.order_updates", "true");
```

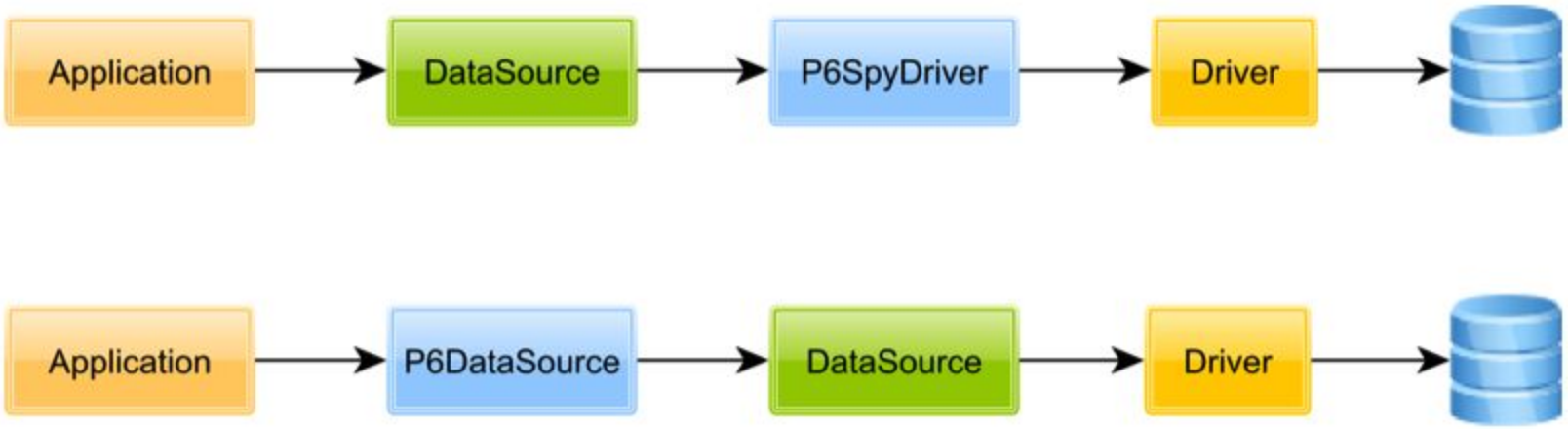
По умолчанию используйте Lazy Fetching

```
Association type | Default fetching policy
=================|=======================
@OneTMany        | LAZY
@ManyToMany      | LAZY
@ManyToOne       | EAGER
@OneToOne        | EAGER
```

Для отладки используйте механизмы логирования ORM и Proxy DataSource/Driver

```
78
79    <logger name="org.hibernate.SQL" level="debug"/>
80
81    INSERT INTO messages (id, payload, type) VALUES (?, ?, ?)
82
```

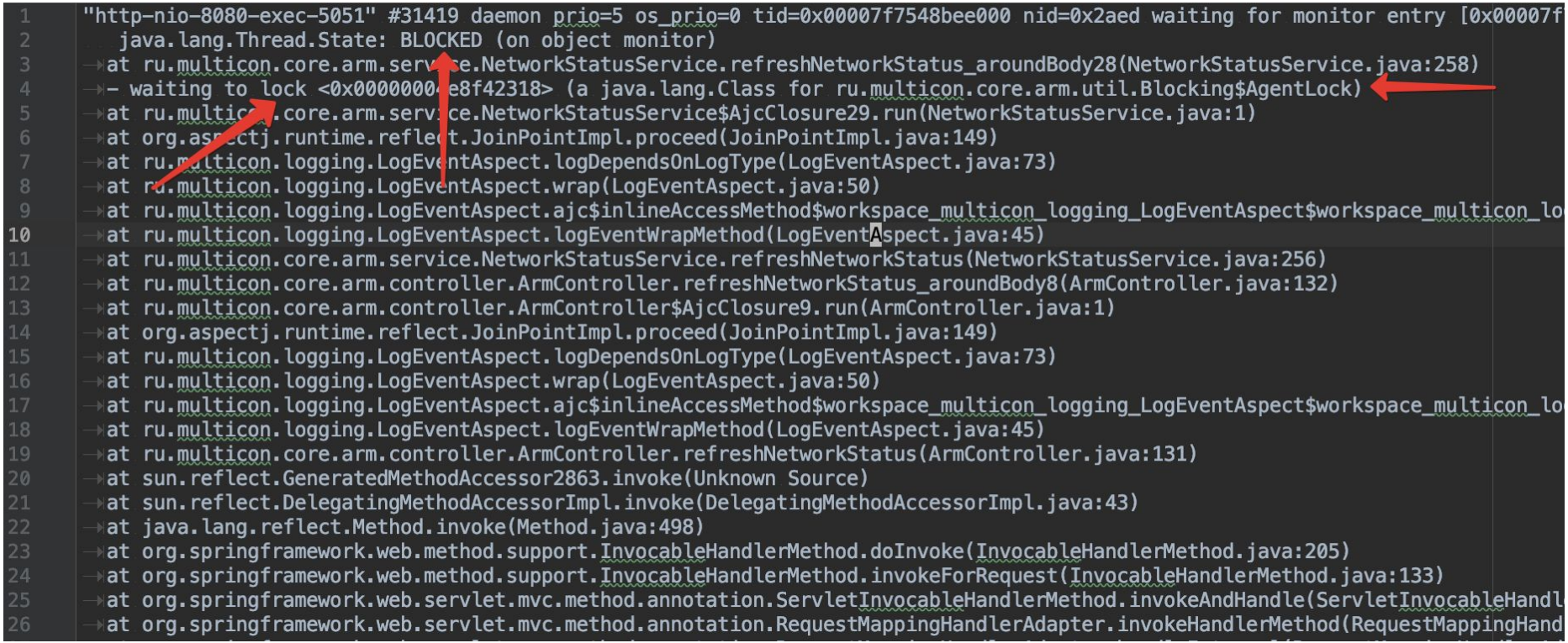
P6Spy


Application
DataSource
P6SpyDriver
Driver
Application
P6DataSource
DataSource
Driver

Что делать, если приложение "тормозит"?

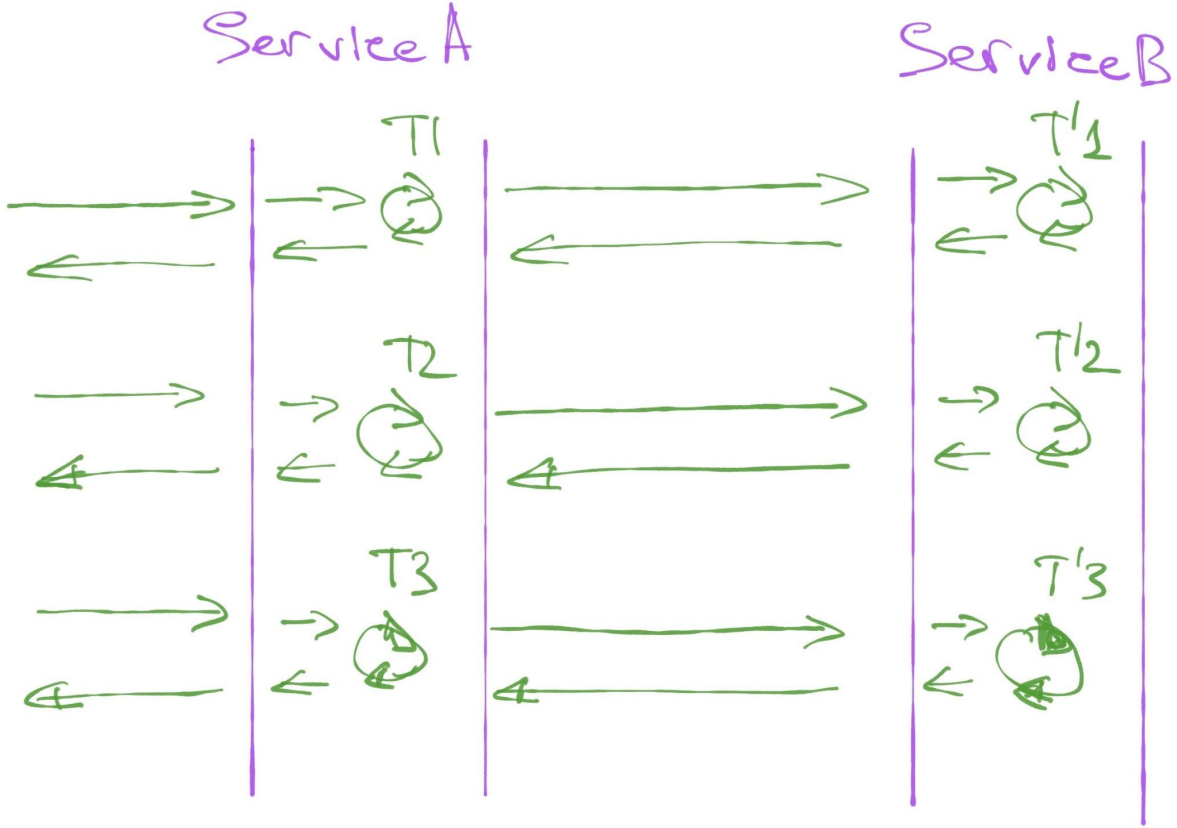
Используйте JMX!

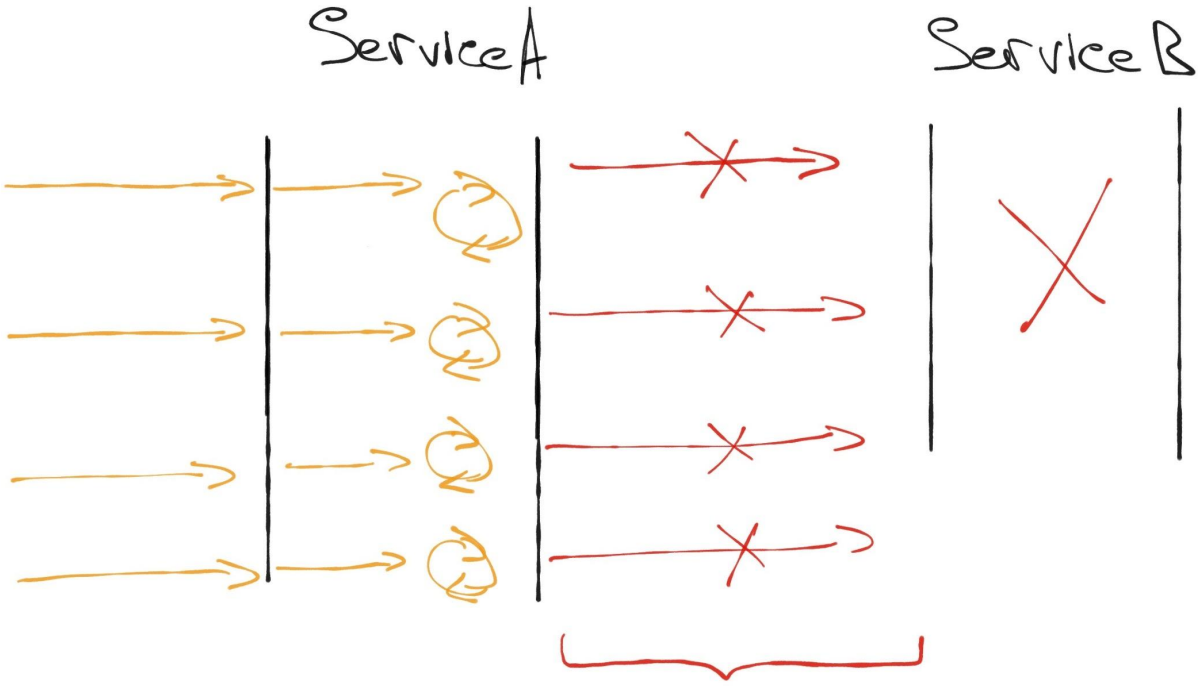
```
83
84  -Dcom.sun.management.jmxremote
85
86  jvisualvm
87
88  kill -s QUIT <pid>
89
90  docker kill -s QUIT <container-id>
```

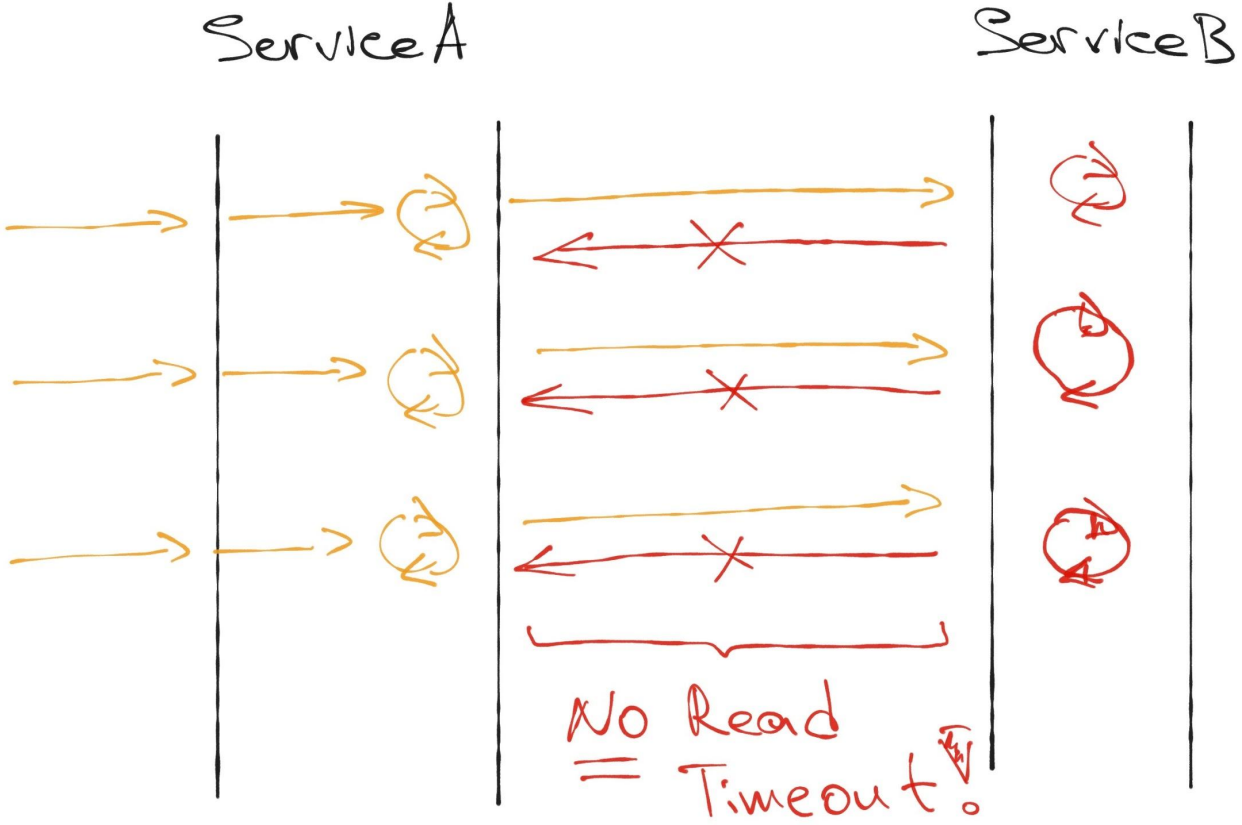

"http-nio-8080-exec-5051" #31419 daemon prio=5 os_prio=0 tid=0x00007f7548bee000 nid=0x2aed waiting for monitor entry [0x00007f
java.lang.Thread.State: BLOCKED (on object monitor)
at ru.multicon.core.arm.service.NetworkStatusService.refreshNetworkStatus_aroundBody28(NetworkStatusService.java:258)
- waiting to lock <0x00000004e8f42318> (a java.lang.Class for ru.multicon.core.arm.util.Blocking$AgentLock)
at ru.multicon.core.arm.service.NetworkStatusService$AjcClosure29.run(NetworkStatusService.java:1)
at org.aspectj.runtime.reflect.JoinPointImpl.proceed(JoinPointImpl.java:149)
at ru.multicon.logging.LogEventAspect.logDependsOnLogType(LogEventAspect.java:73)
at ru.multicon.logging.LogEventAspect.wrap(LogEventAspect.java:50)
at ru.multicon.logging.LogEventAspect.ajc$inlineAccessMethod$workspace_multicon_logging_LogEventAspect$workspace_multicon_lo
at ru.multicon.logging.LogEventAspect.logEventWrapMethod(LogEventAspect.java:45)
at ru.multicon.core.arm.service.NetworkStatusService.refreshNetworkStatus(NetworkStatusService.java:256)
at ru.multicon.core.arm.controller.ArmController.refreshNetworkStatus_aroundBody8(ArmController.java:132)
at ru.multicon.core.arm.controller.ArmController$AjcClosure9.run(ArmController.java:1)
at org.aspectj.runtime.reflect.JoinPointImpl.proceed(JoinPointImpl.java:149)
at ru.multicon.logging.LogEventAspect.logDependsOnLogType(LogEventAspect.java:73)
at ru.multicon.logging.LogEventAspect.wrap(LogEventAspect.java:50)
at ru.multicon.logging.LogEventAspect.ajc$inlineAccessMethod$workspace_multicon_logging_LogEventAspect$workspace_multicon_lo
at ru.multicon.logging.LogEventAspect.logEventWrapMethod(LogEventAspect.java:45)
at ru.multicon.core.arm.controller.ArmController.refreshNetworkStatus(ArmController.java:131)
at sun.reflect.GeneratedMethodAccessor2863.invoke(Unknown Source)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at org.springframework.web.method.support.InvocableHandlerMethod.doInvoke(InvocableHandlerMethod.java:205)
at org.springframework.web.method.support.InvocableHandlerMethod.invokeForRequest(InvocableHandlerMethod.java:133)
at org.springframework.web.servlet.mvc.method.annotation.ServletInvocableHandlerMethod.invokeAndHandle(ServletInvocableHandl
at org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter.invokeHandlerMethod(RequestMappingHand

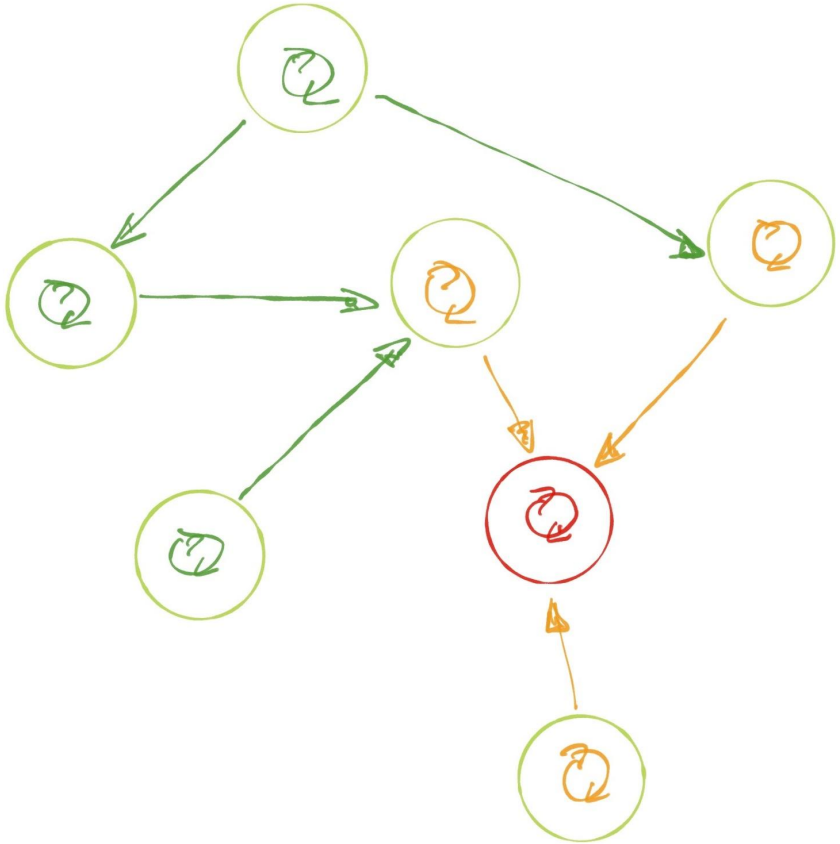
Майкл Нейгард. Release it! Проектирование и дизайн ПО для тех, кому не все равно

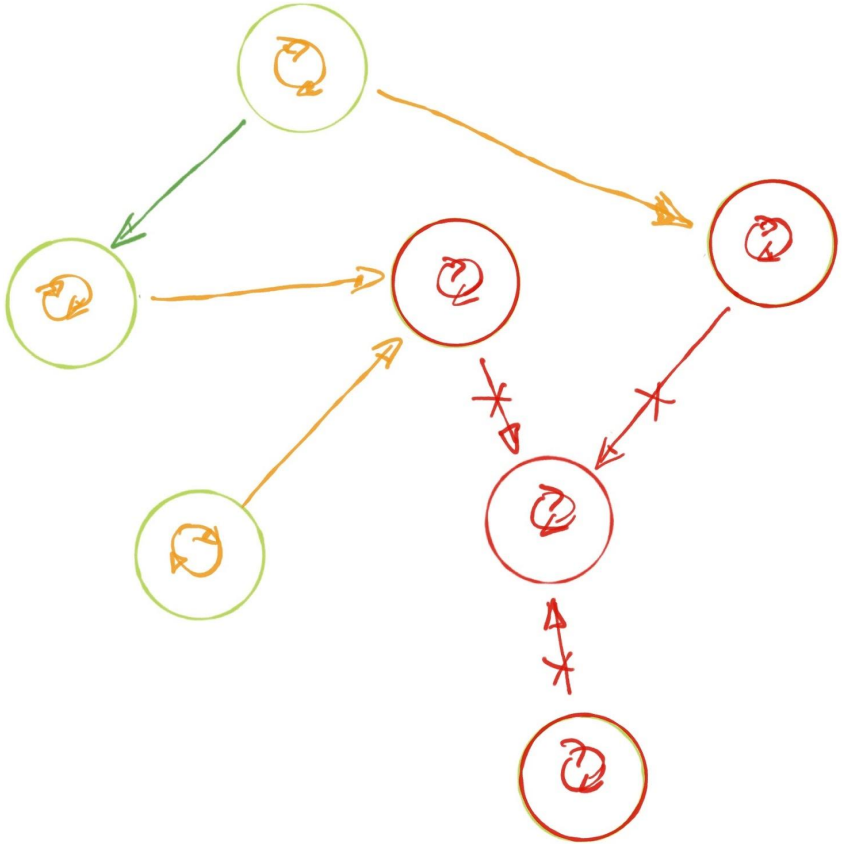

TIMEOUTS

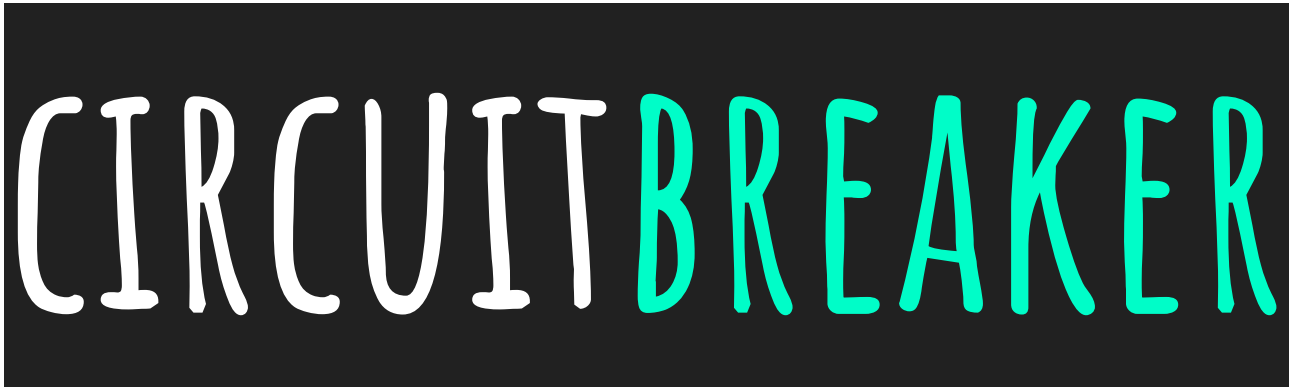

Service A
Service B
T1
T'1
T2
T'2
T3
T'3

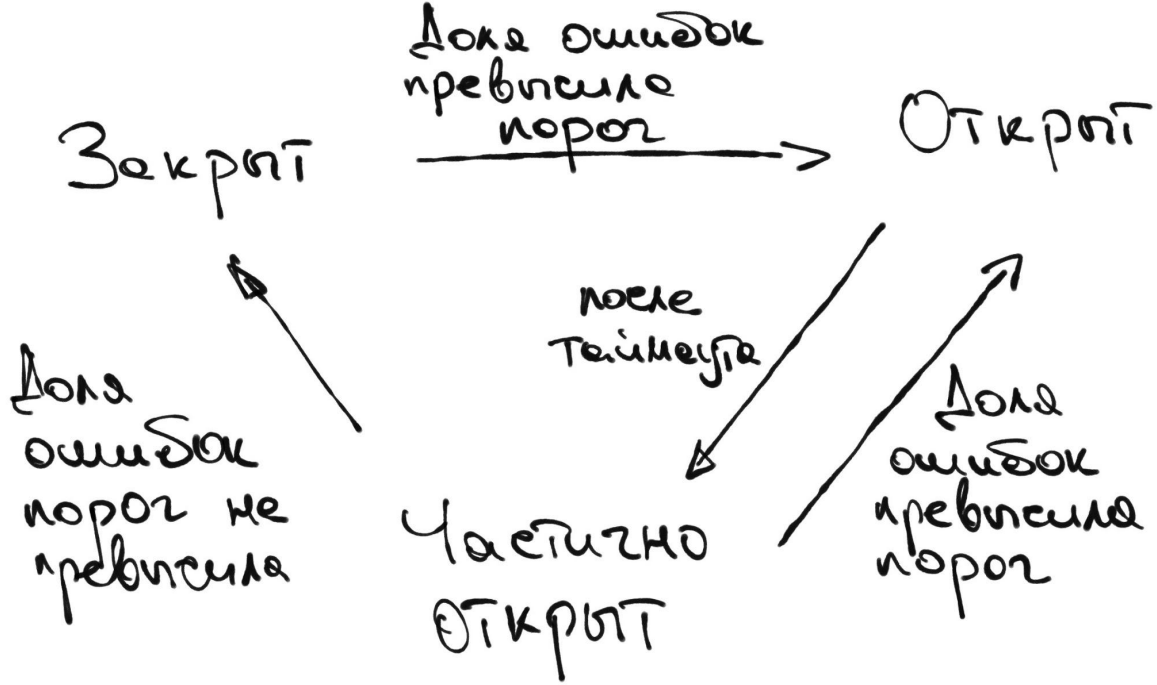

Service A
Service B
No Connection Timeout


Service A
Service B
No Read
= Timeout!

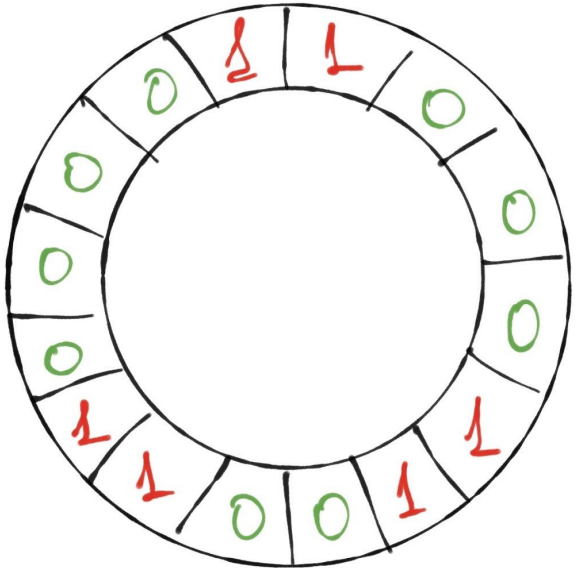




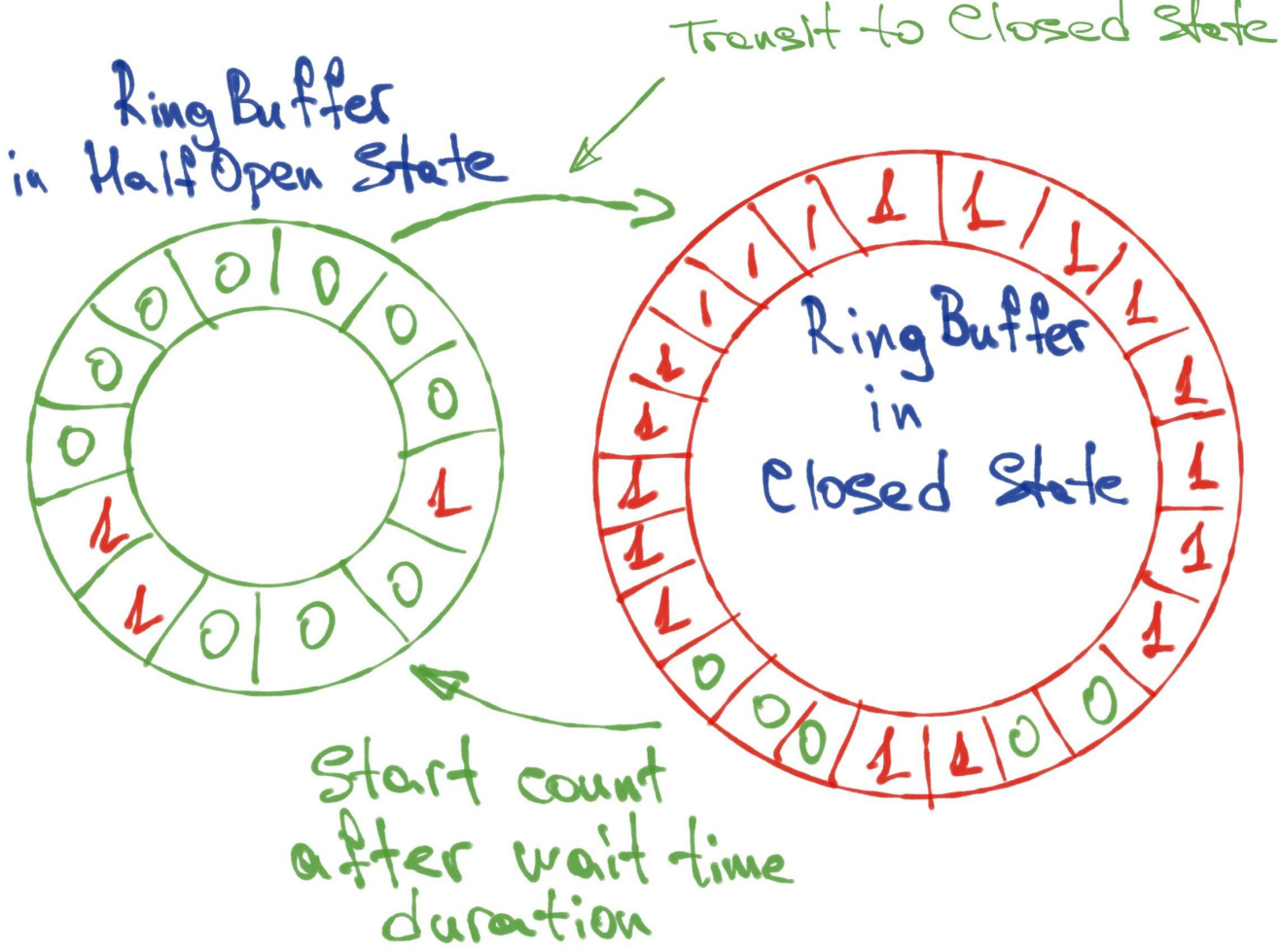

CIRCUITBREAKER

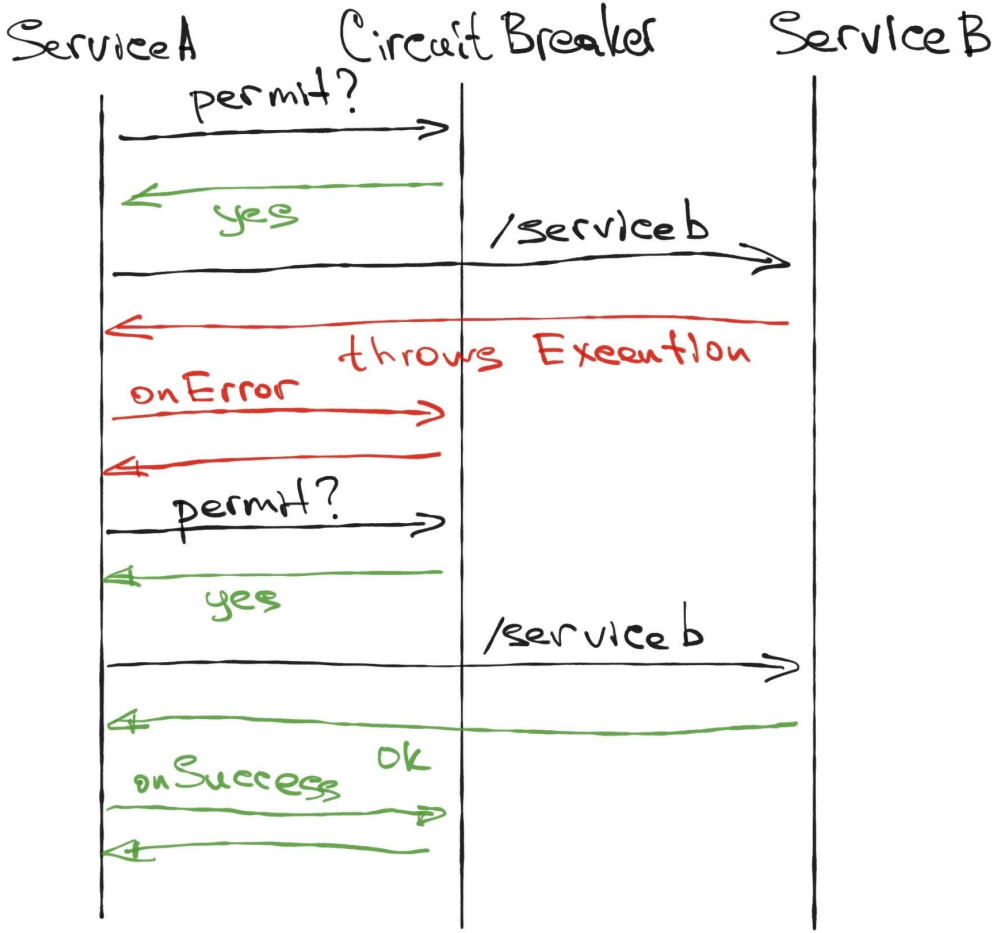

Доля ошибок превысила порог
Закрыт
Открыт
после таймаута
Доля ошибок порог не превысила
Частично открыт
Доля ошибок превысила порог

Threshold = 50% FRate = $\frac{6}{15}$

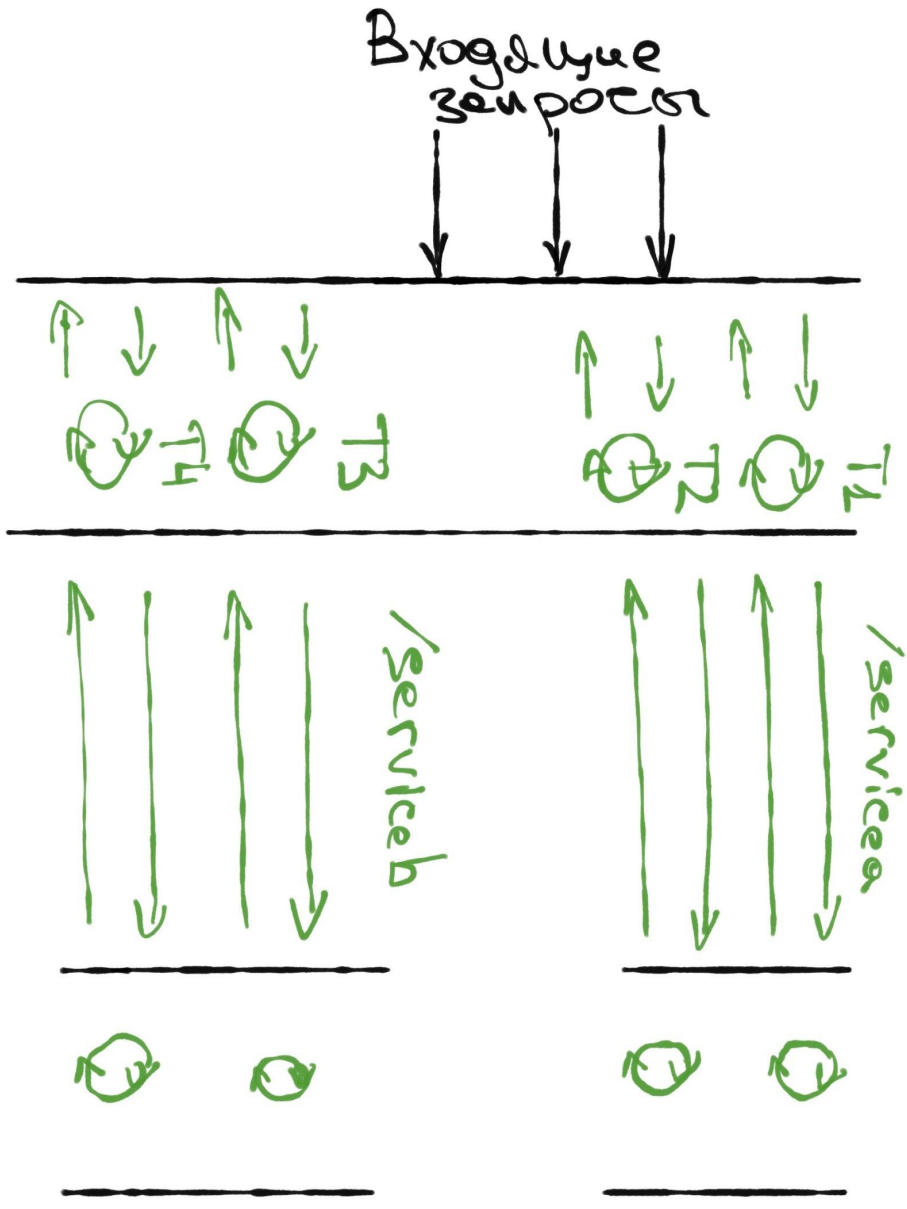


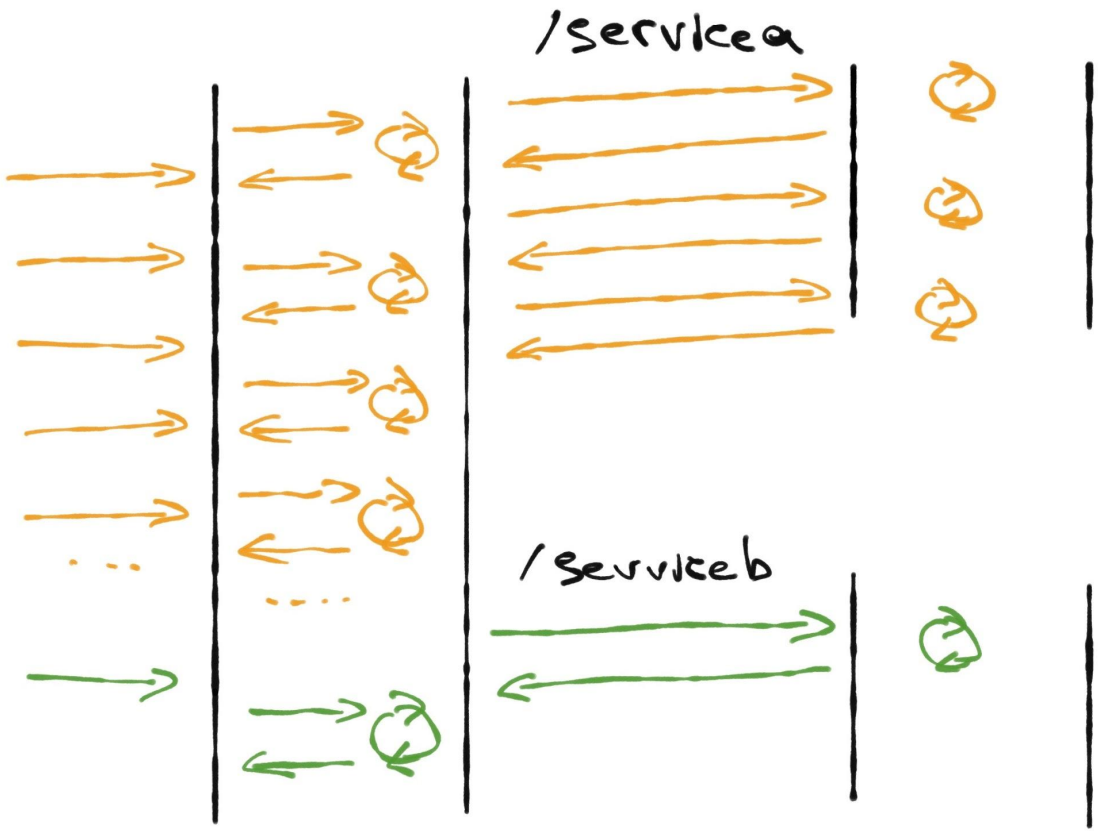
$6/15 < 1/2 \Rightarrow$ Closed

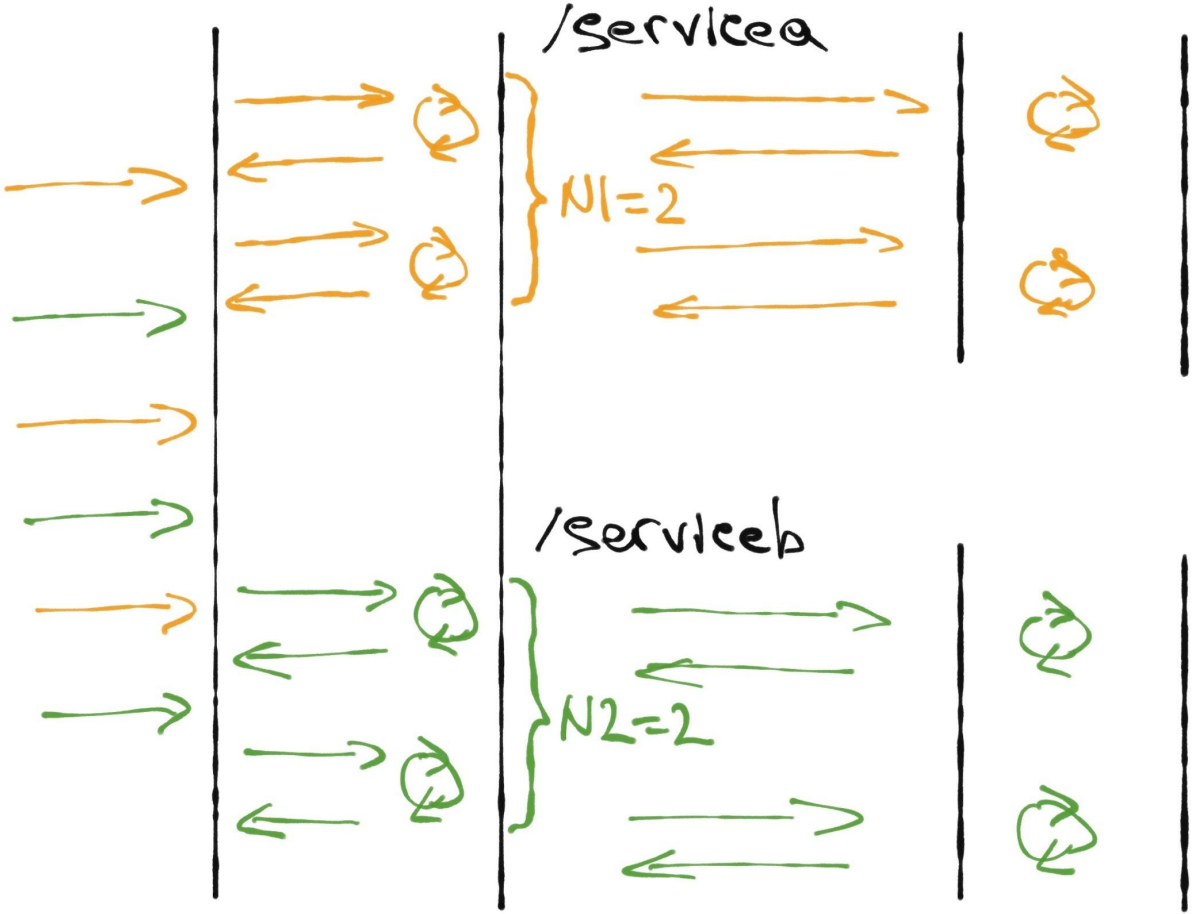

Transit to Closed State
Ring Buffer
in Half Open State
Ring Buffer
in
Closed State
Start count
after wait time
duration


Service A
Circuit Breaker
Service B
permit?
yes
/service b
throws Exception
onError
permit?
yes
/service b
ok
onSuccess


BULKHEAD


Входящие запросы
T4
T3
T2
T1
/serviceb
/servicea


/servicea
/serviceb

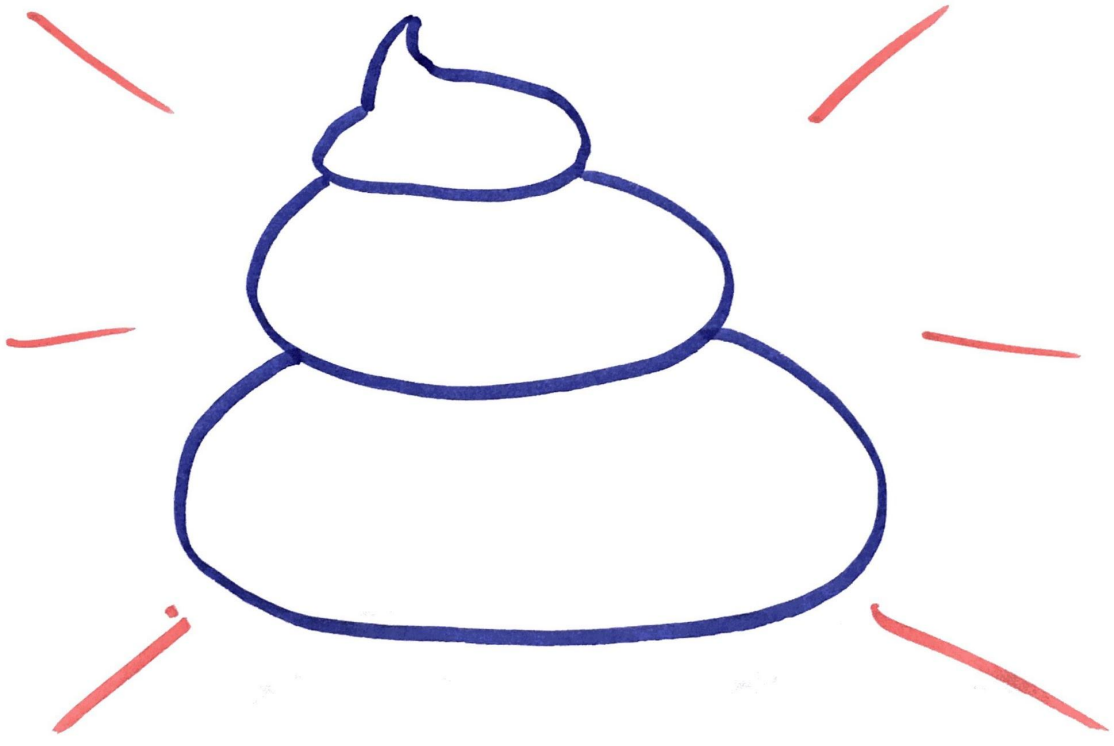

/servicea
N1=2
/serviceb
N2=2

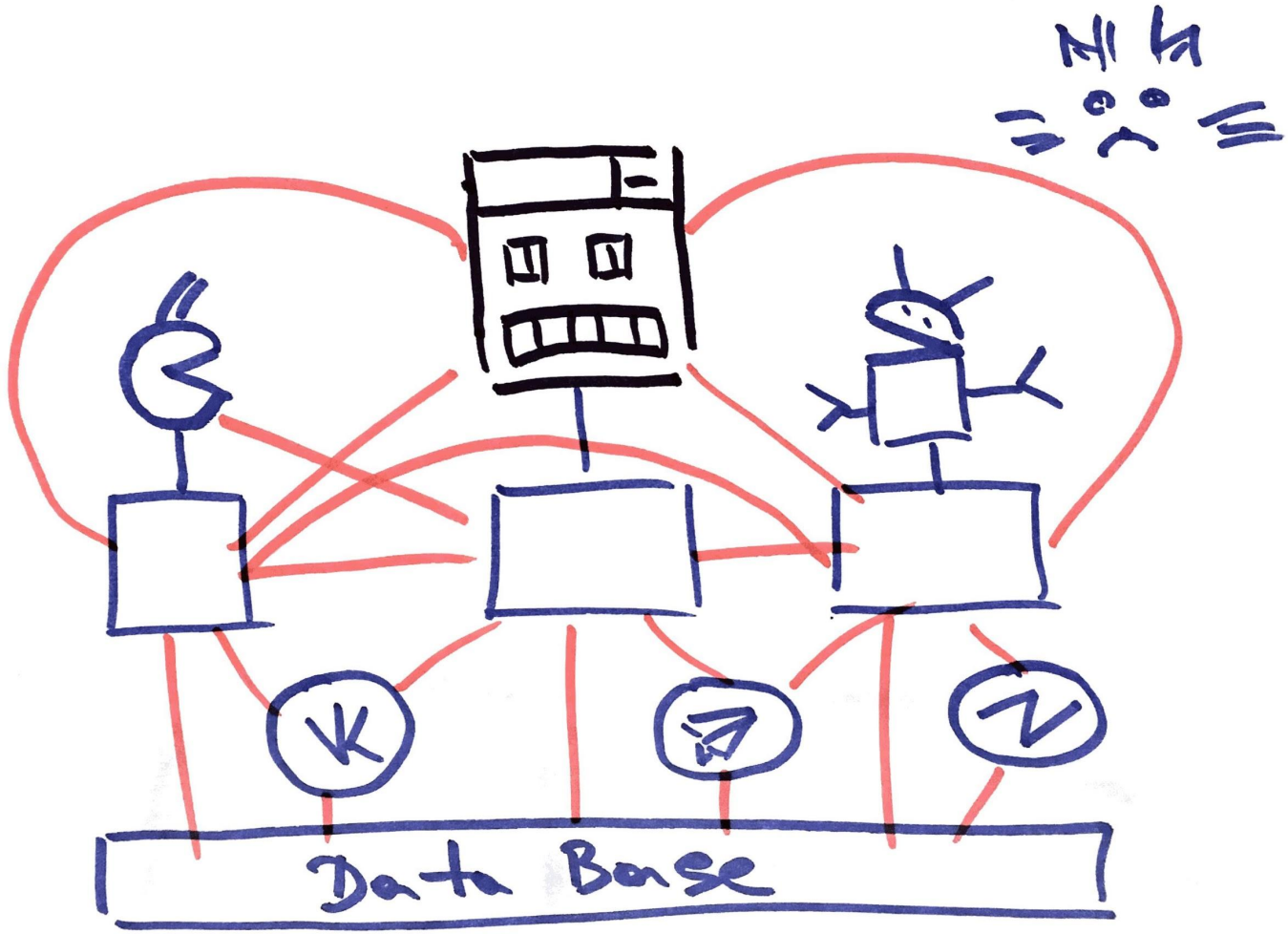

RESILIENCE4J

Наивное проектирование...

Основная гипотеза

SHARED DB + RPC =

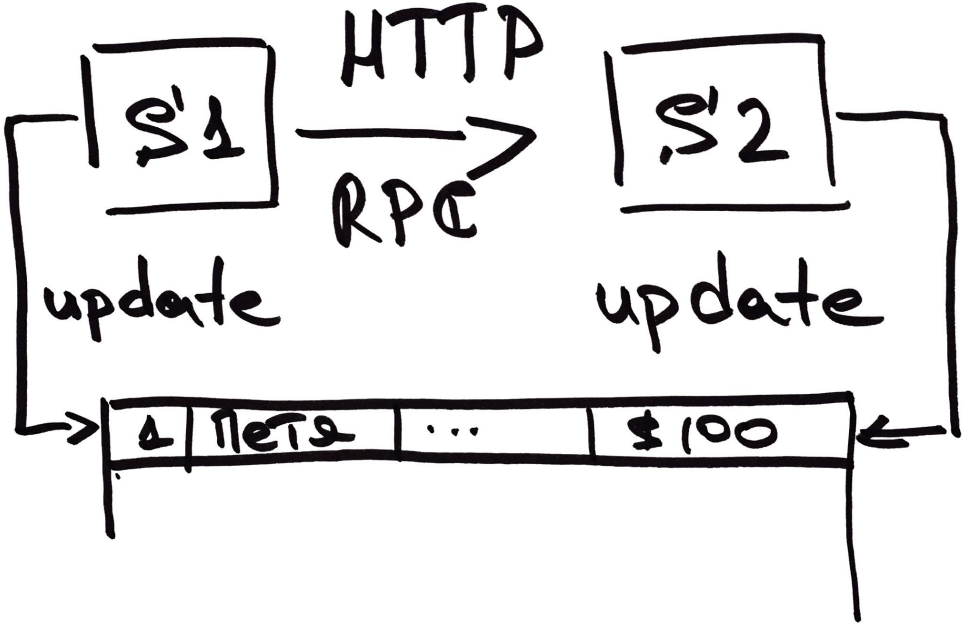



Data Base

Хьюстон, у нас проблемы?! "Лучшее" из двух миров

- Блокировки и даже ... deadlocks
- Связанность во время разработки
- Нет выбора в организации данных
- Ограниченные варианты взаимодействия (request/reply)
- Потребность знать всех своих визави

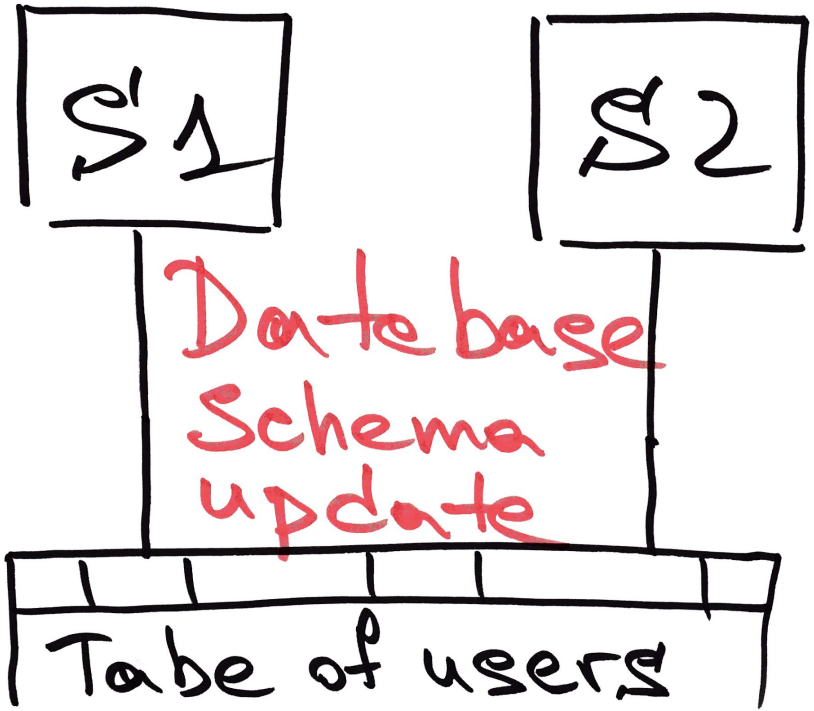
Блокировки и дедлоки



Хьюстон, у нас проблемы?! "Лучшее" из двух миров

- Блокировки и даже ... deadlocks
- Связанность во время разработки
- Нет выбора в организации данных
- Ограниченные варианты взаимодействия (request/reply)
- Потребность знать всех своих визави

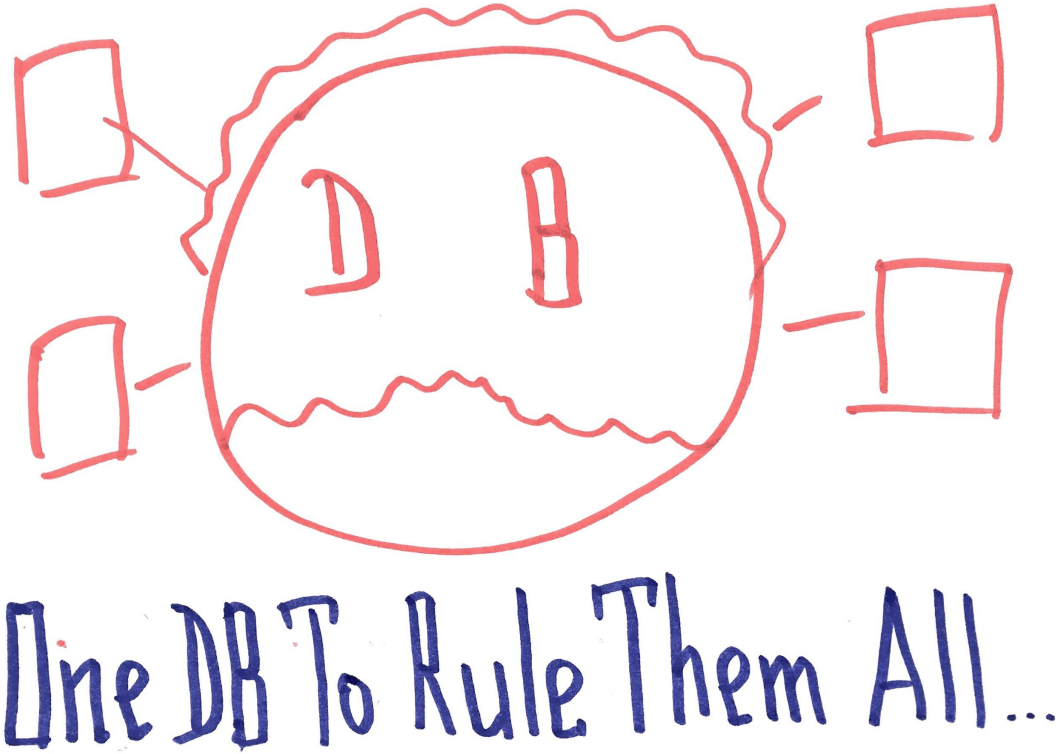
Конфликты обновления схемы данных



Хьюстон, у нас проблемы?! "Лучшее" из двух миров

- Блокировки и даже ... deadlocks
- Связанность во время разработки
- Нет выбора в организации данных
- Ограниченные варианты взаимодействия (request/reply)
- Потребность знать всех своих визави

Единая БД на все



Хьюстон, у нас проблемы?! "Лучшее" из двух миров

- Блокировки и даже ... deadlocks
- Связанность во время разработки
- Нет выбора в организации данных
- Ограниченные варианты взаимодействия (request/reply)
- Потребность знать всех своих визави

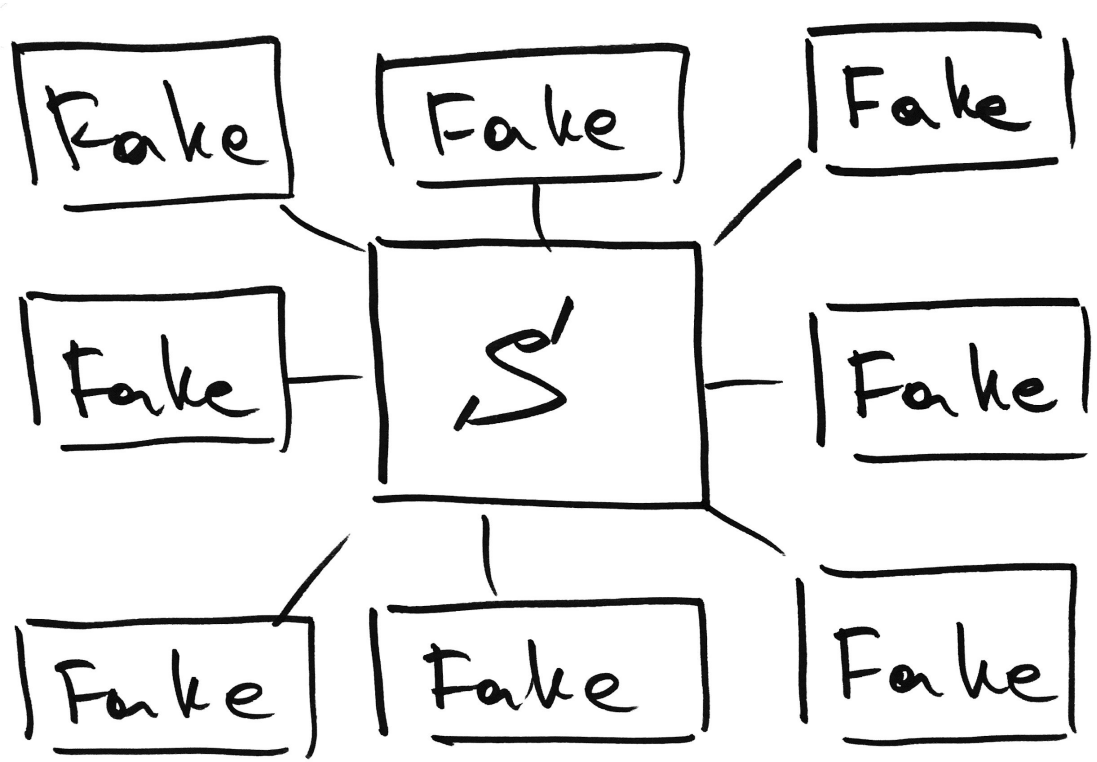
Сложные сервисы


Фильтрация

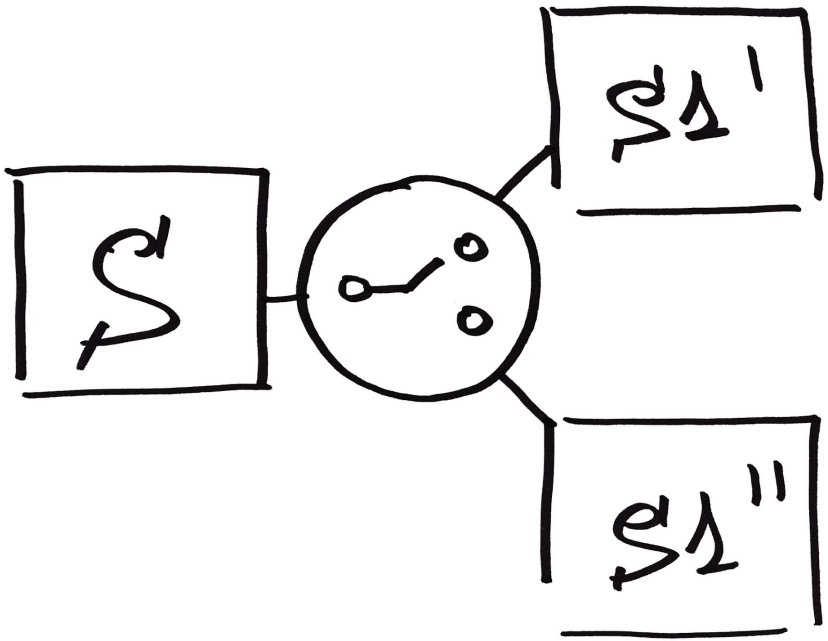

Агрегирование


Упорядочение

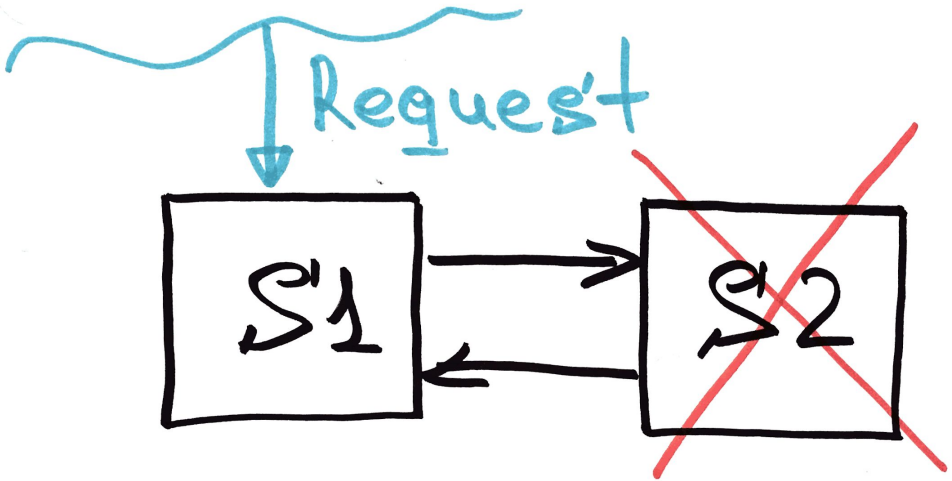
Плохо поддающиеся тестированию



... и масштабированию



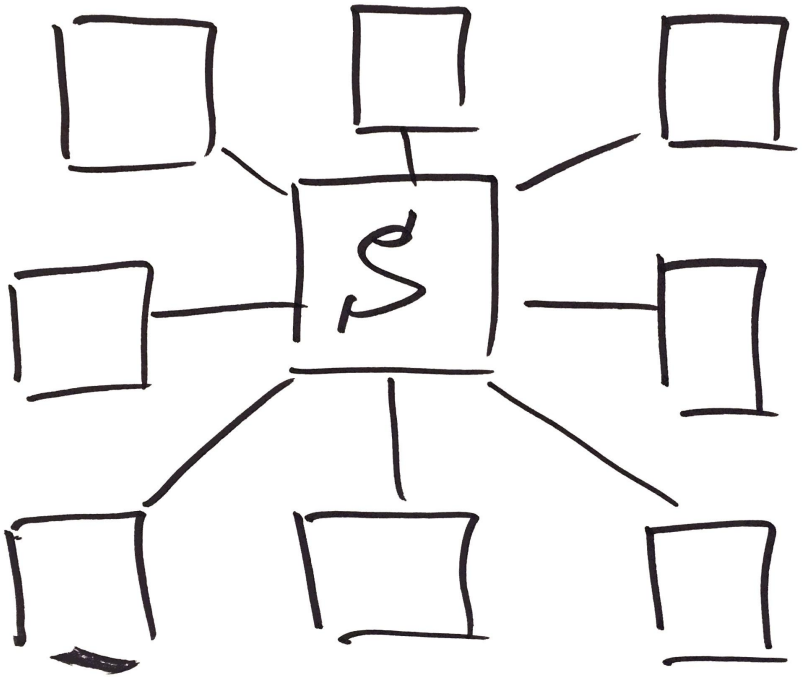
... И НЕНАДЕЖНЫЕ



Хьюстон, у нас проблемы?! "Лучшее" из двух миров

- Блокировки и даже ... deadlocks
- Связанность во время разработки
- Нет выбора в организации данных
- Ограниченные варианты взаимодействия (request/reply)
- Потребность знать всех своих визави

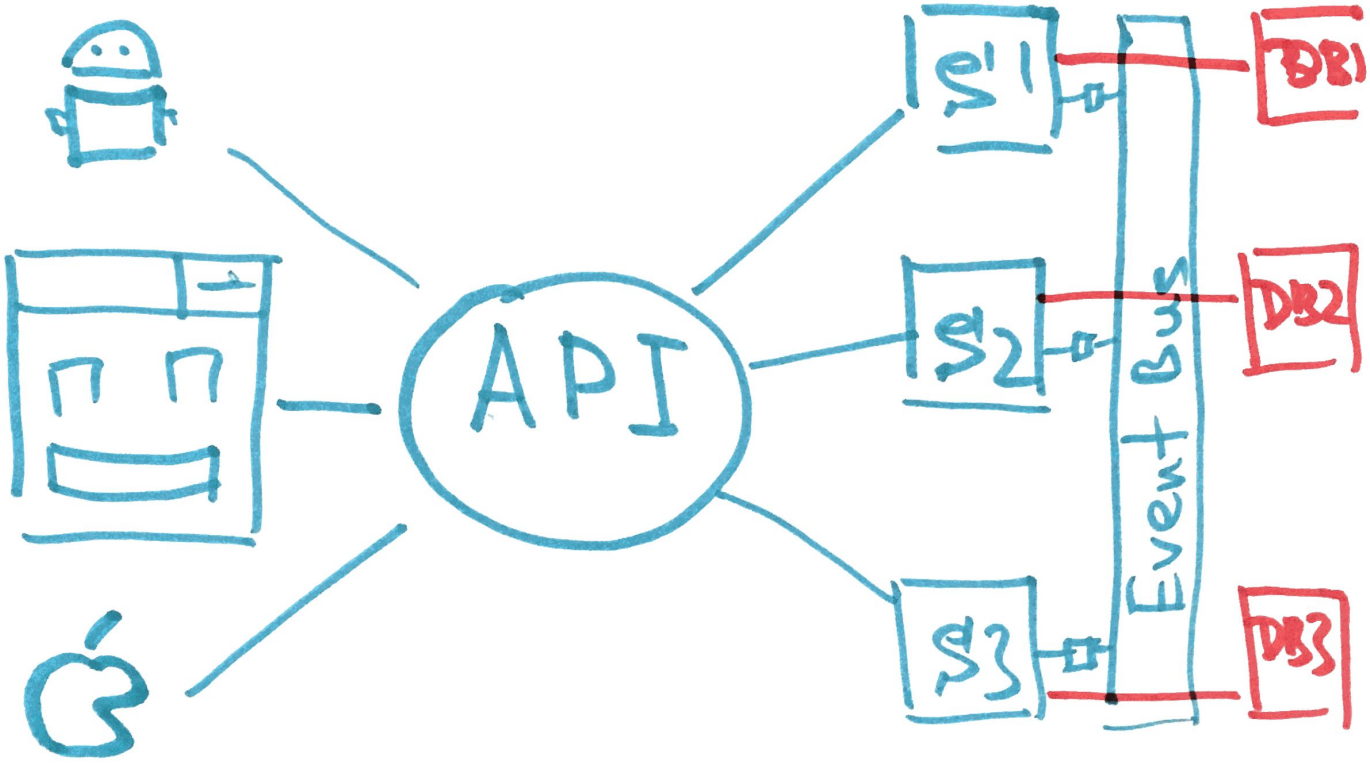
Сложная структура взаимодействия



ЧТО ДЕЛАТЬ?

НЕ УСЛОЖНЯТЬ!

И ПЕРЕЕХАТЬ НА МИКРОСЕРВИСЫ ;)...


API
S1
S2
S3
DB1
DB2
DB3
Event Bus

СЕРВИСОВ БОЛЬШЕ,

НО ОНИ ПРОЩЕ!

РАЗДЕЛЬНЫЕ БД

- НЕТ КОНФЛИКТОВ
- ЛЕГКО ОБНОВЛЯТЬСЯ

EVENT BUS

- УНИФИКАЦИЯ ПРОТОКОЛОВ ОБЩЕНИЯ
- ШИРОКИЙ НАБОР ВАРИАНТОВ ОБЩЕНИЯ
- МАЛЕНЬКИЕ КОМПОНЕНТЫ
- ОТНОСИТЕЛЬНО НЕЗАВИСИМЫЕ
- ТОЛЕРАНТНЫЕ К ОШИБКАМ И СТЕКУ ТЕХНОЛОГИЙ

ГОМОГЕННАЯ СРЕДА

- ПРОЩЕ ТЕСТИРОВАТЬ
- ПРОЩЕ ОТЛАЖИВАТЬ

РЕЦЕПТ

"Это зависит от удачи. Все зависит от удачи..."

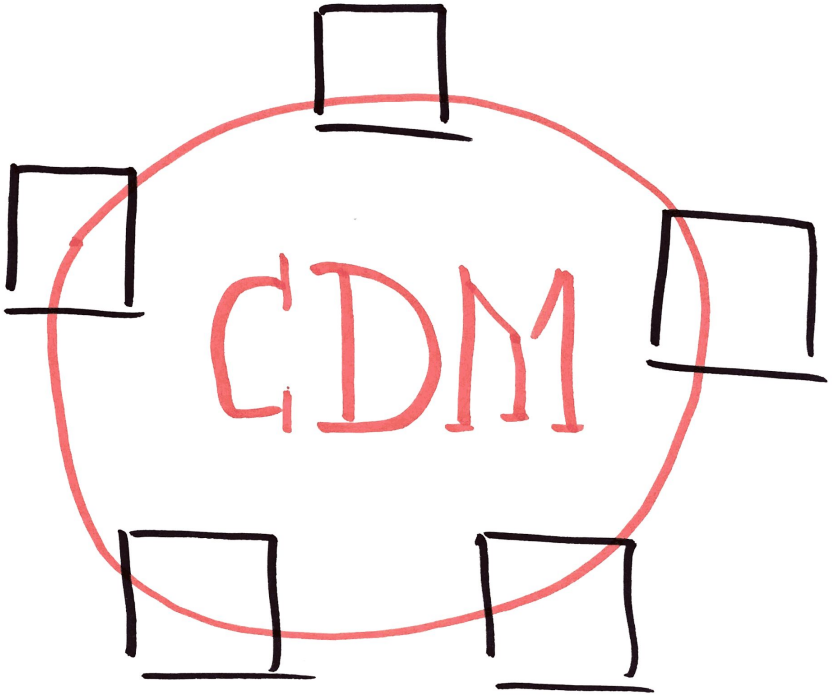
Как делать?

- разделять
- упрощать

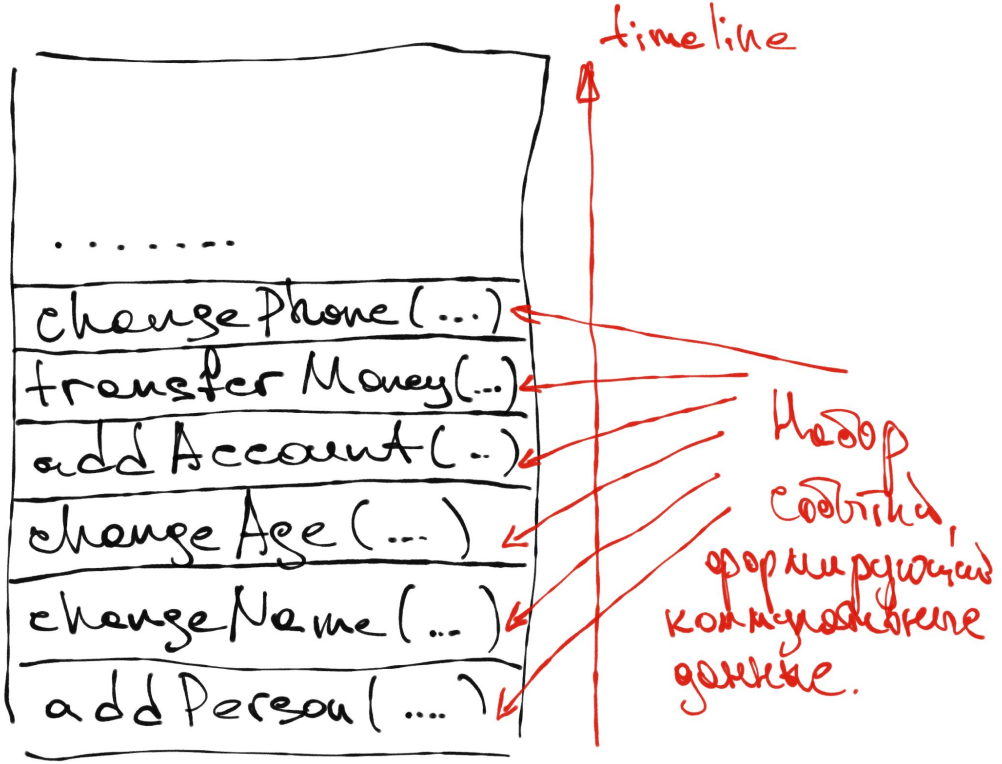
Что использовать?

- Каноническая модель?!
- EventSoursing?!
- CQRS?!
- Тесты!
- Мониторинг, трасировщики, etc.?!

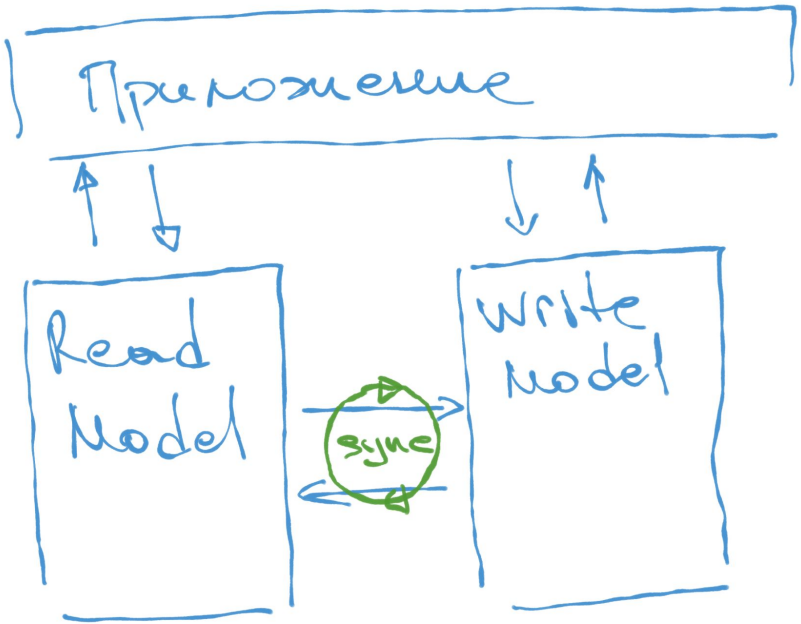
Каноническая модель данных



EventSourcing



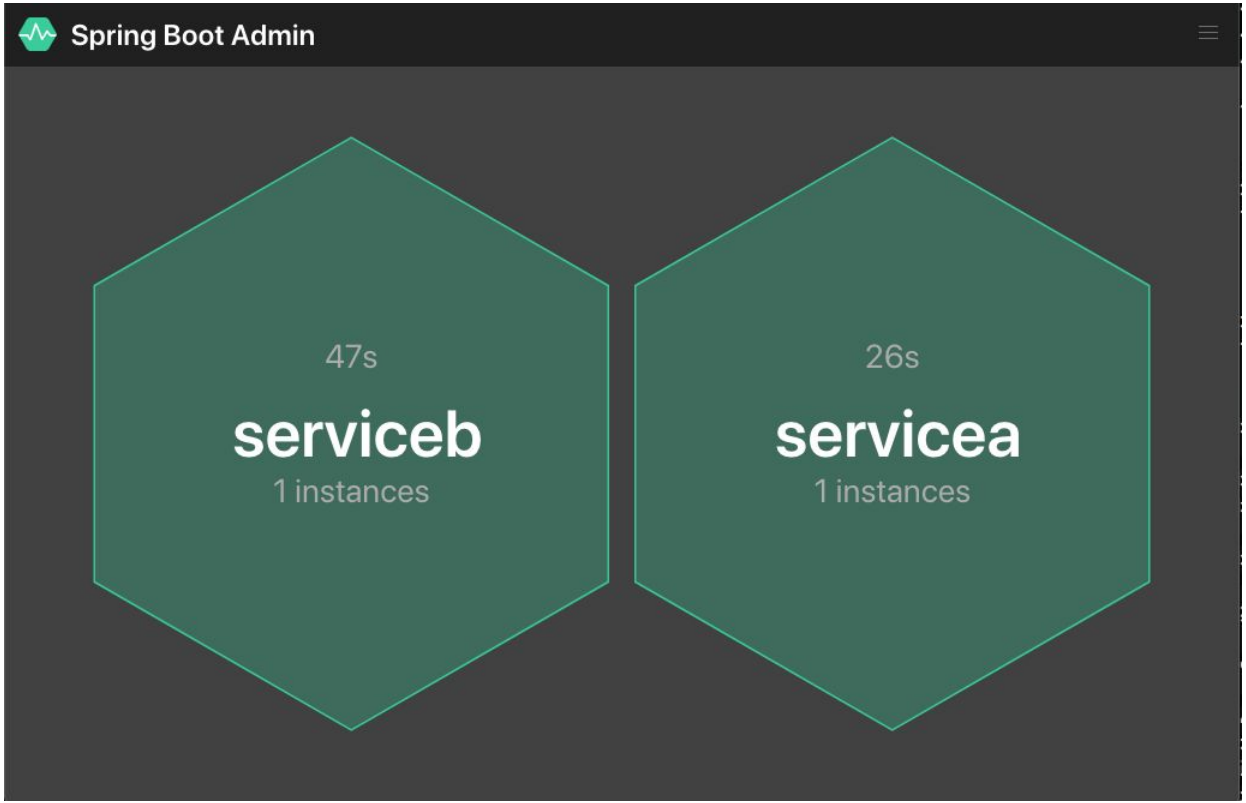
CQRS



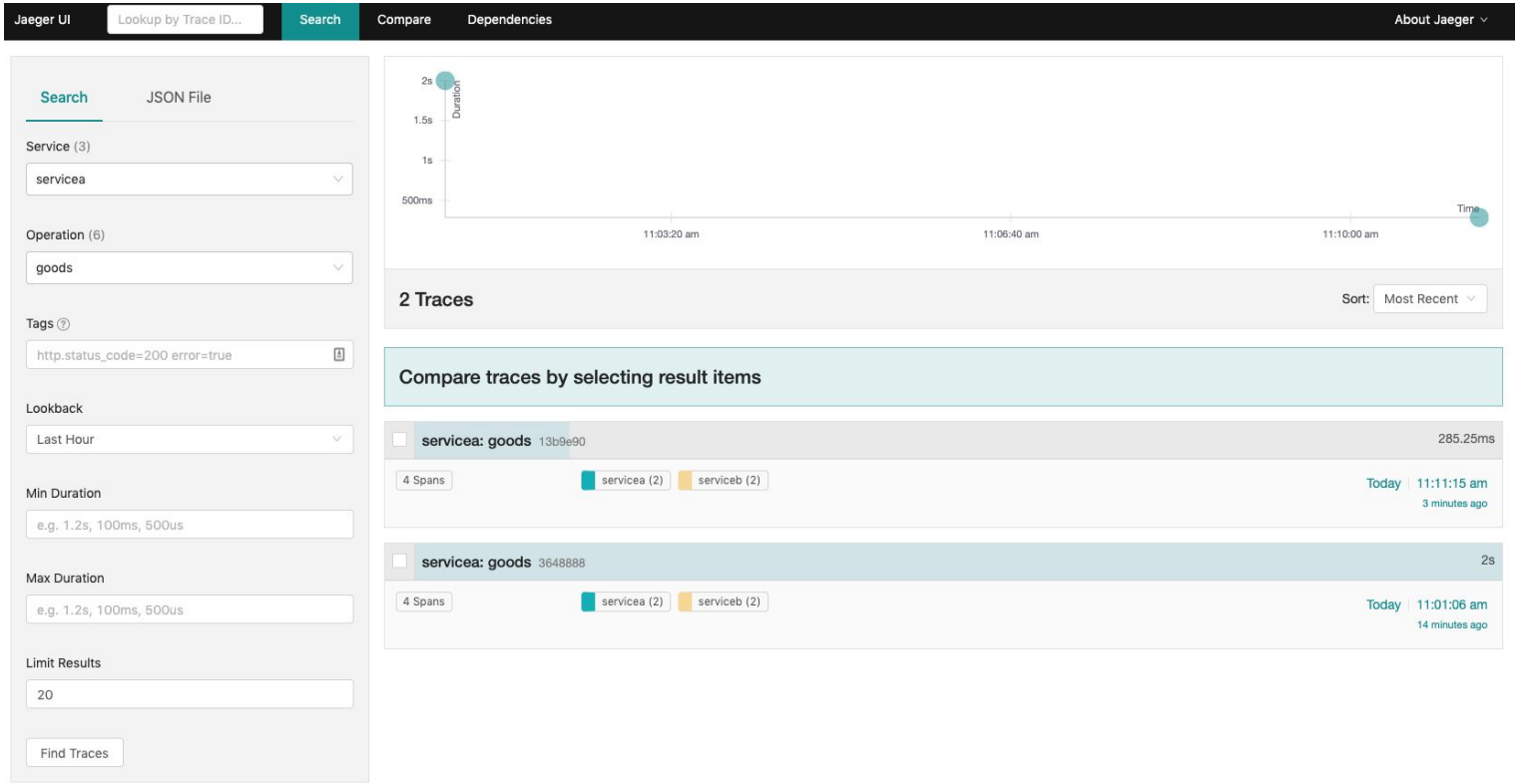
Тестирование

- @SpringBootTest
- @TestConfiguration
- @MockBean, @SpyBean

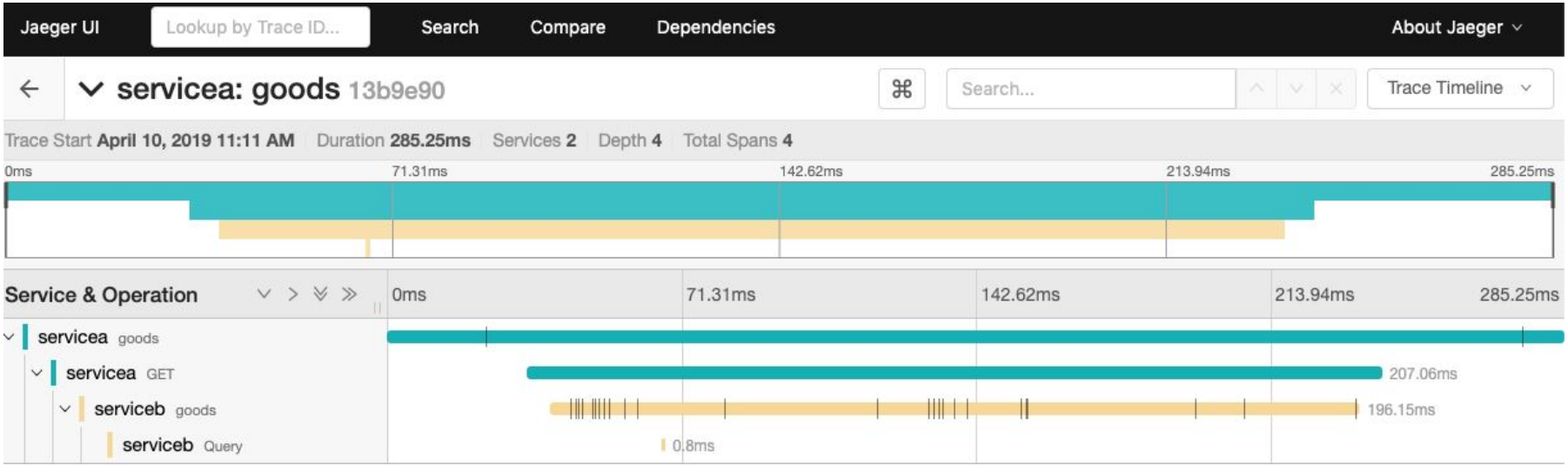
Мониторинг


Spring Boot Admin
47s
serviceb
1 instances
26s
servicea
1 instances

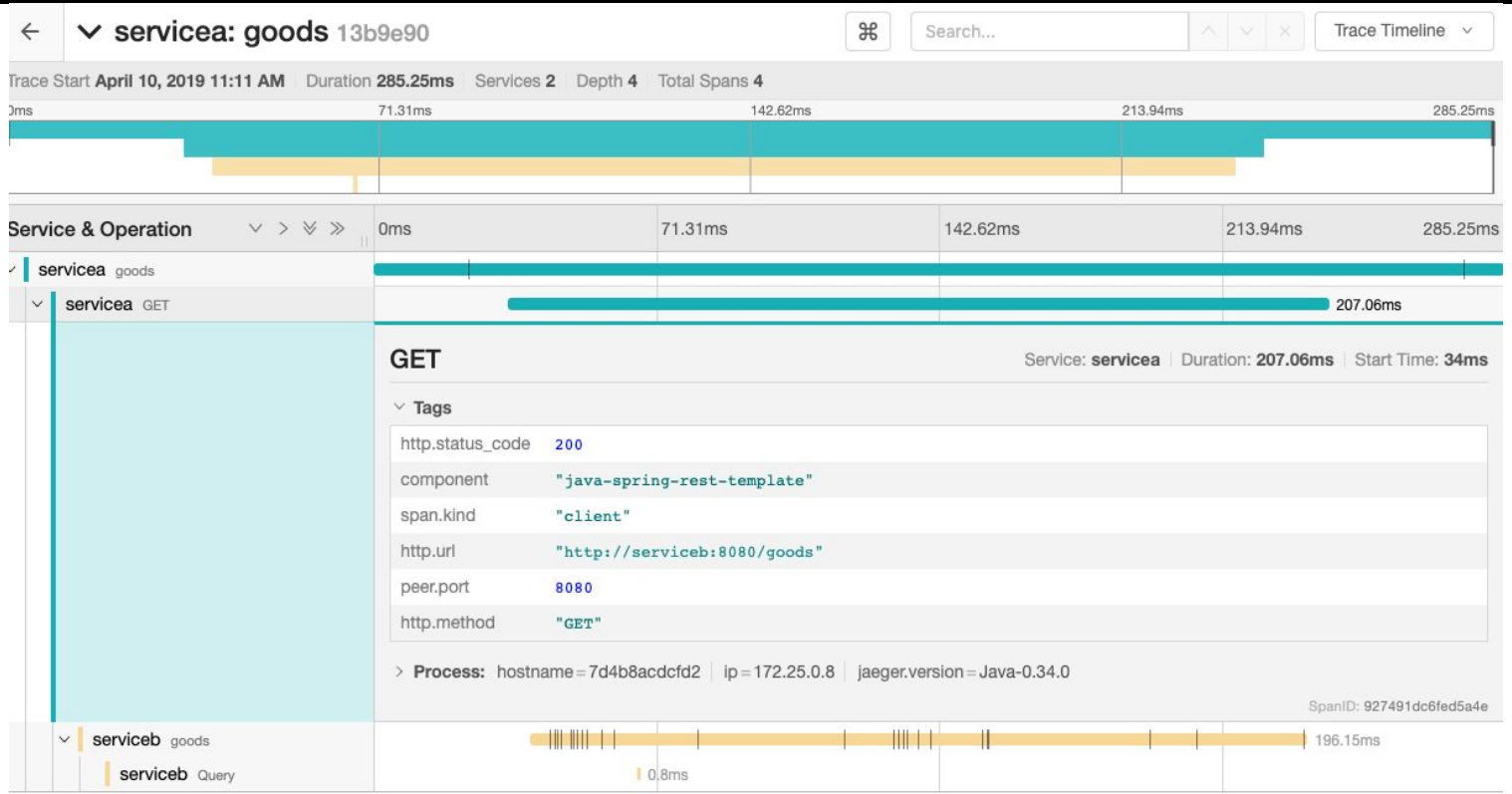
Трасировка


Jaeger UI
Lookup by Trace ID...
Search
Compare
Dependencies
About Jaeger
Search
JSON File
Service (3)
servicea
Operation (6)
goods
Tags
http.status_code=200 error=true
Lookback
Last Hour
Min Duration
e.g. 1.2s, 100ms, 500us
Max Duration
e.g. 1.2s, 100ms, 500us
Limit Results
20
Find Traces
2s
1.5s
1s
500ms
Duration
Time
11:03:20 am
11:06:40 am
11:10:00 am
2 Traces
Sort:
Most Recent
Compare traces by selecting result items
servicea: goods 13b9e90
285.25ms
4 Spans
servicea (2)
serviceb (2)
Today 11:11:15 am
3 minutes ago
servicea: goods 3648888
2s
4 Spans
servicea (2)
serviceb (2)
Today 11:01:06 am
14 minutes ago

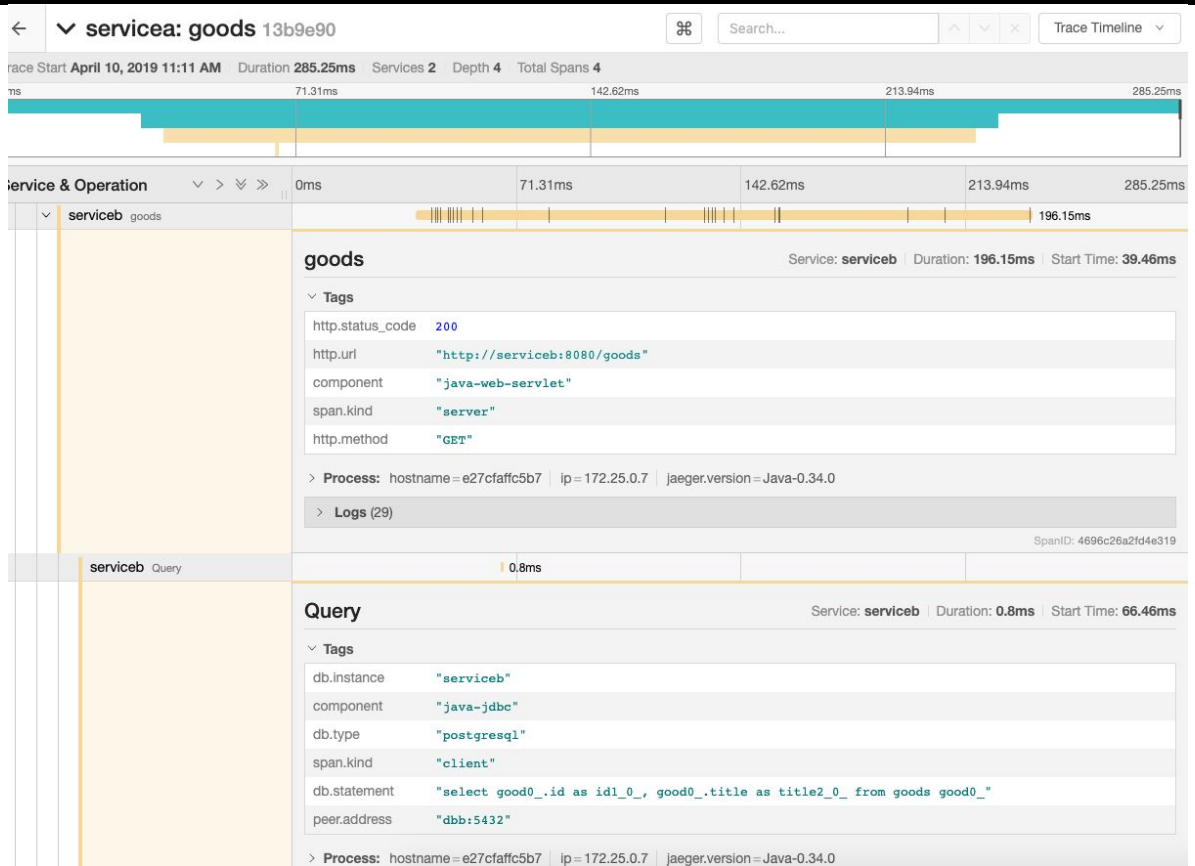
Трасировка


Jaeger UI
Lookup by Trace ID...
Search
Compare
Dependencies
About Jaeger
servicea: goods 13b9e90
Search...
Trace Timeline
Trace Start April 10, 2019 11:11 AM
Duration 285.25ms
Services 2
Depth 4
Total Spans 4
0ms
71.31ms
142.62ms
213.94ms
285.25ms
Service & Operation
servicea goods
servicea GET
207.06ms
serviceb goods
196.15ms
serviceb Query
0.8ms

Трасировка


servicea: goods 13b9e90
Search...
Trace Timeline
Trace Start April 10, 2019 11:11 AM | Duration 285.25ms | Services 2 | Depth 4 | Total Spans 4
0ms
71.31ms
142.62ms
213.94ms
285.25ms
Service & Operation
servicea goods
servicea GET
207.06ms
GET
Service: servicea | Duration: 207.06ms | Start Time: 34ms
Tags
http.status_code 200
component "java-spring-rest-template"
span.kind "client"
http.url "http://serviceb:8080/goods"
peer.port 8080
http.method "GET"
Process: hostname = 7d4b8acdcfd2 | ip = 172.25.0.8 | jaeger.version = Java-0.34.0
SpanID: 927491dc6fed5a4e
serviceb goods
196.15ms
serviceb Query
0.8ms

Трасировка


servicea: goods 13b9e90
Search...
Trace Timeline
Start April 10, 2019 11:11 AM
Duration 285.25ms
Services 2
Depth 4
Total Spans 4
71.31ms
142.62ms
213.94ms
285.25ms
Service & Operation
0ms
71.31ms
142.62ms
213.94ms
285.25ms
serviceb goods
196.15ms
goods
Service: serviceb
Duration: 196.15ms
Start Time: 39.46ms
Tags
http.status_code 200
http.url "http://serviceb:8080/goods"
component "java-web-servlet"
span.kind "server"
http.method "GET"
Process: hostname = e27cfaffc5b7 | ip = 172.25.0.7 | jaeger.version = Java-0.34.0
Logs (29)
SpanID: 4696c26a2fd4e319
serviceb Query
0.8ms
Query
Service: serviceb
Duration: 0.8ms
Start Time: 66.46ms
Tags
db.instance "serviceb"
component "java-jdbc"
db.type "postgresql"
span.kind "client"
db.statement "select good0_.id as id1_0_, good0_.title as title2_0_ from goods good0_"
peer.address "dbb:5432"
Process: hostname = e27cfaffc5b7 | ip = 172.25.0.7 | jaeger.version = Java-0.34.0

Итого

- Читайте документацию
- Следите за пулами соединений и потоков
- тестируйте самые худшие сценарии
- оставайтесь любознательными ;)

References

- Gregor Hohpe. Enterprise Integration Patterns
- Chris Richardson. Microservices Patterns
- Sebastian Daschner. Architecting Modern Java EE Applications
- Martin Kleppmann. Designing Data-Intensive Applications
- Николай Голов. Целостность данных в микросервисной архитектуре
- https://microservices.io
- Максим Гореликов. Дизайн реактивной системы на Spring 5

Критика приветствуется, но не является обязательной ;)