

Масштабируем Spring Boot микросервисы

Александр Коженков





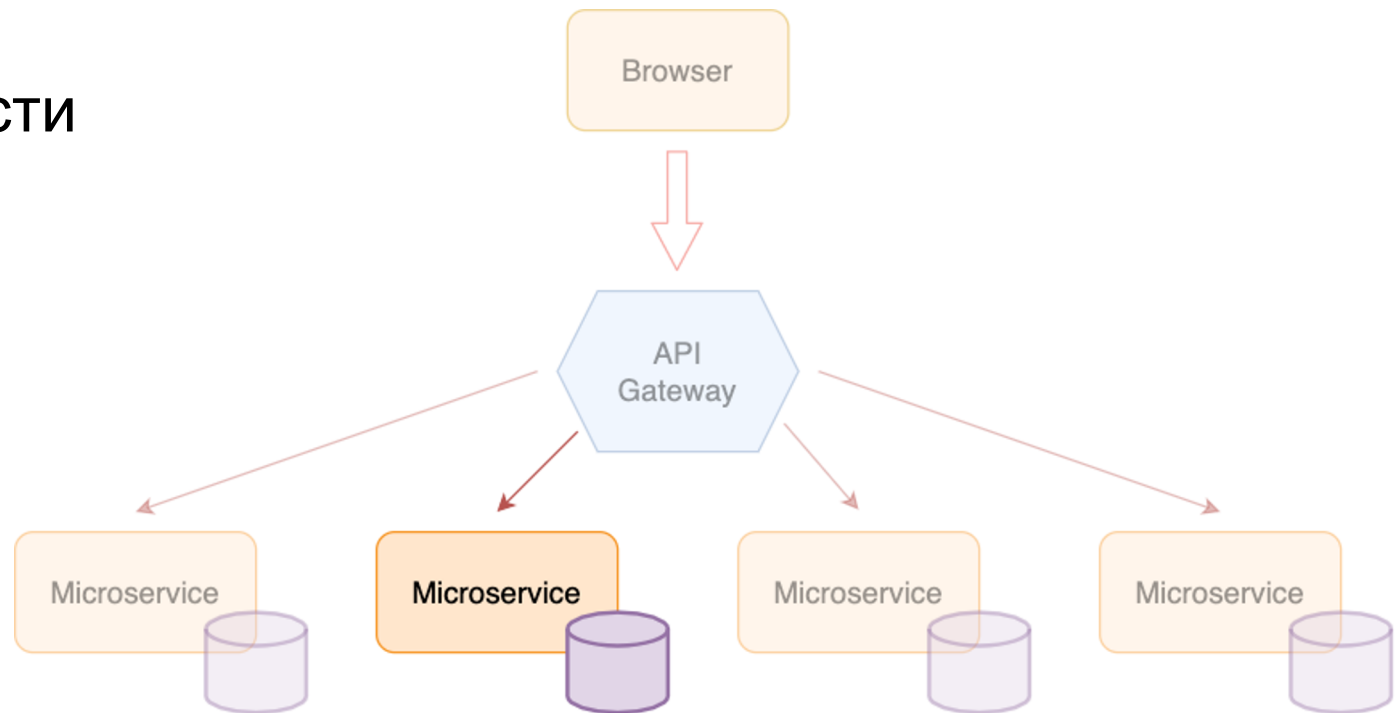
Александр Коженков

Team Lead в Cloud/Tooling команде компании
GridGain

Занимаюсь разработкой GridGain Control Center
- системы для мониторинга и менеджмента
Apache Ignite и GridGain кластеров.

Ситуация

Spring Boot приложение
Не думали о масштабируемости
Срочно замасштабировать



Потенциальные проблемы



Schedulers

Sessions

Cache

WebSockets



Demo



Application overview

Потенциальные проблемы



Schedulers

Sessions

Cache

WebSockets



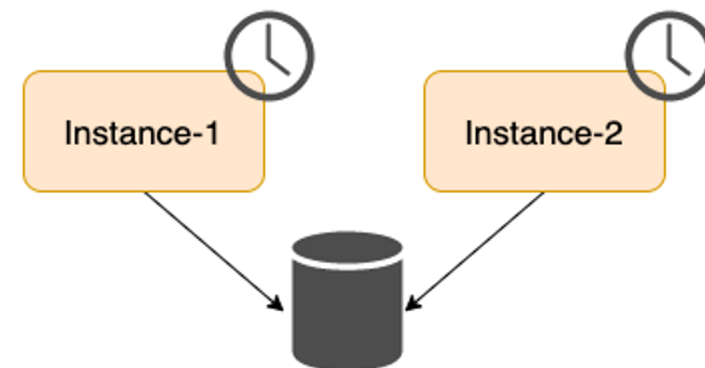
Schedulers



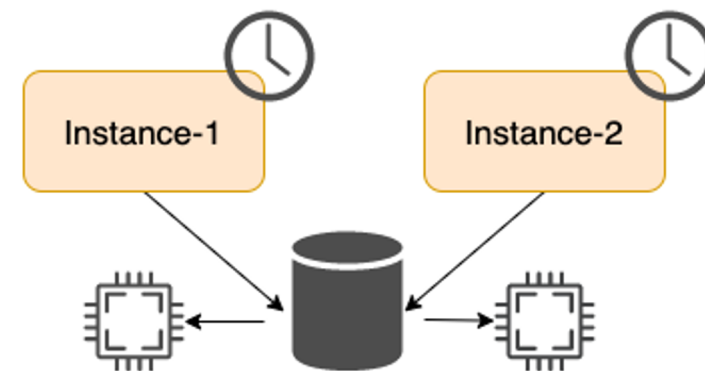
Schedulers



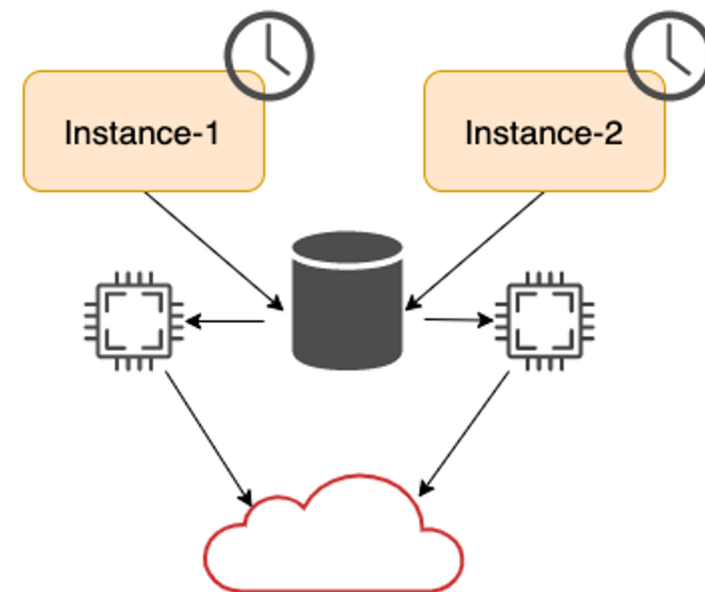
Schedulers



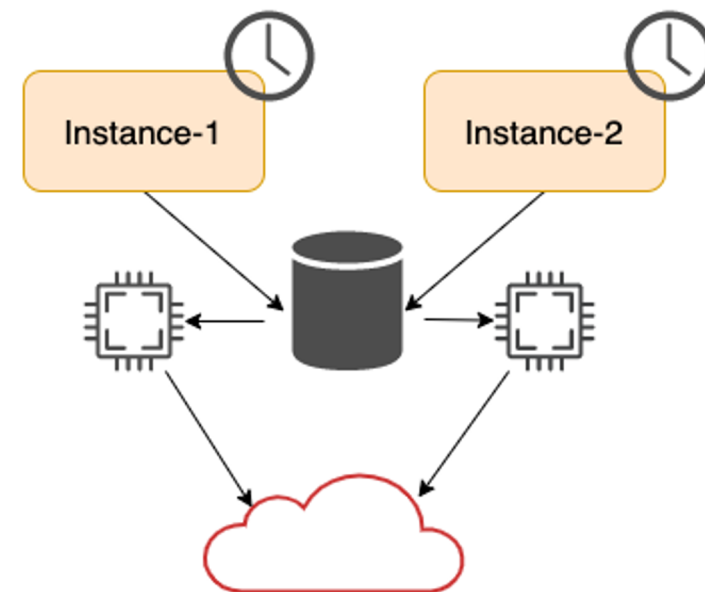
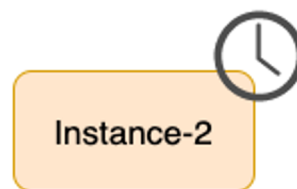
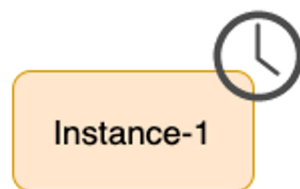
Schedulers



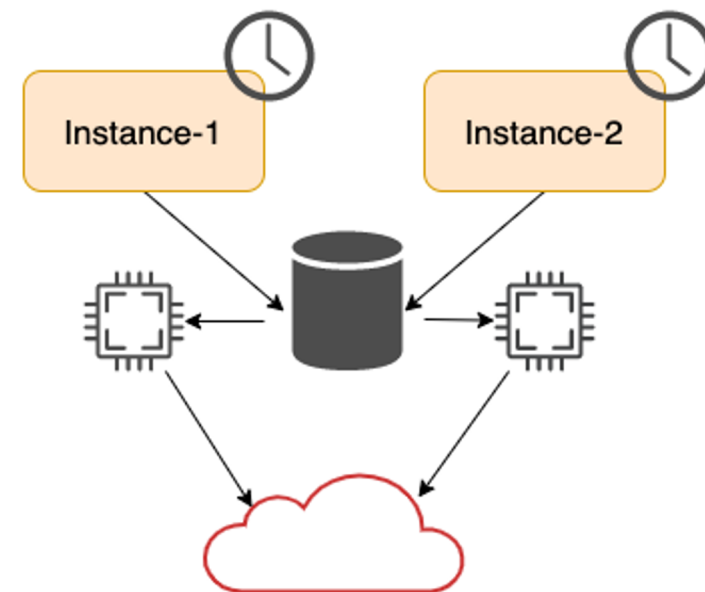
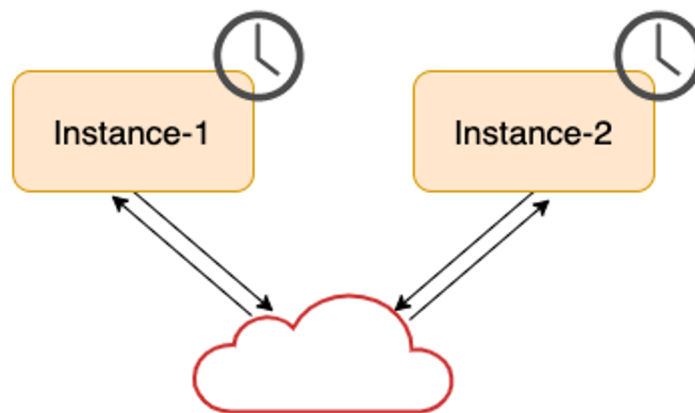
Schedulers



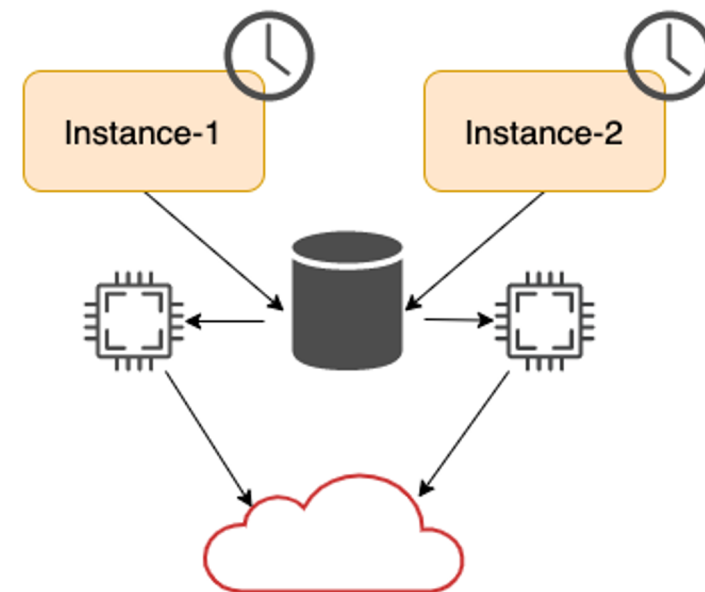
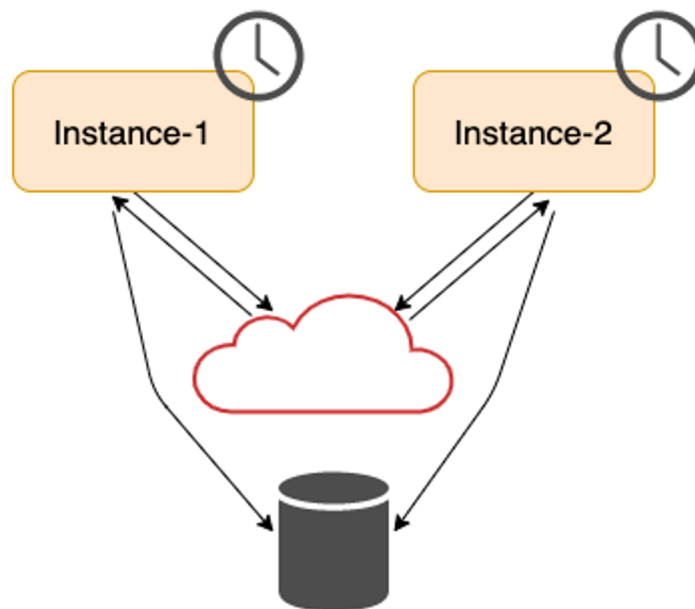
Schedulers



Schedulers

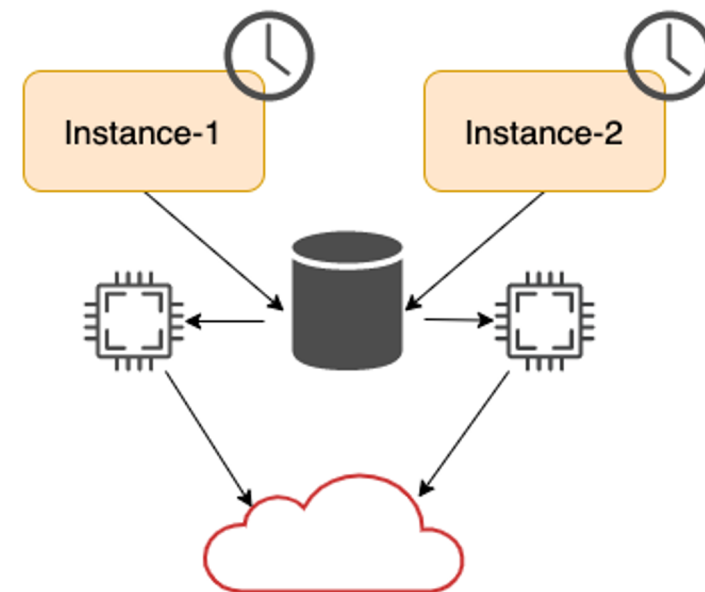
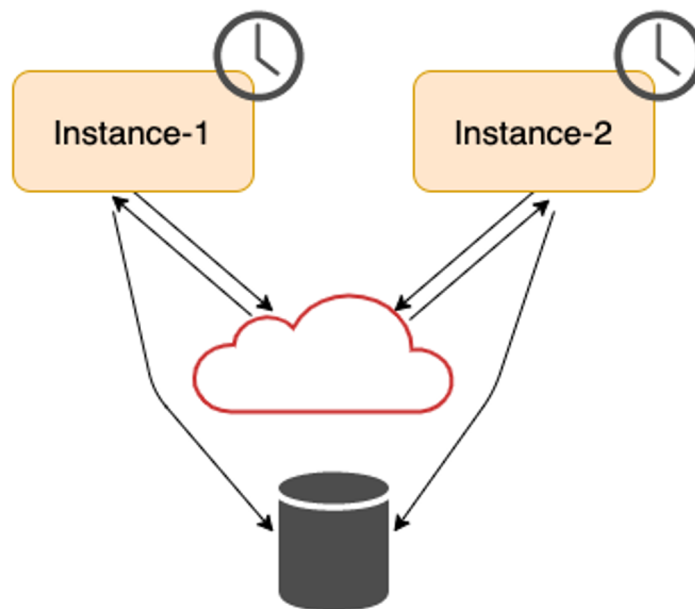


Schedulers



Schedulers

Двойная работа



Schedulers. Вариант 1



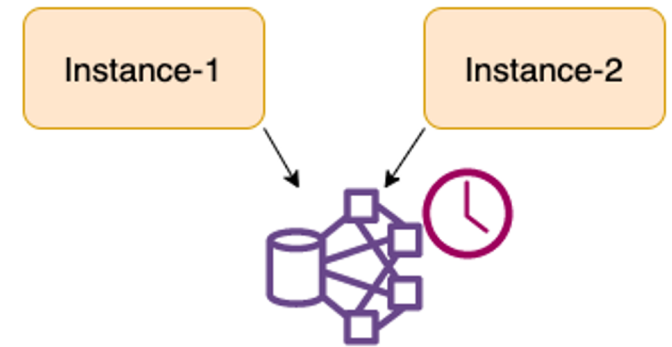
Распределенный планировщик

Failover

Load balancing

Требует настройки

Переписывание кода



Schedulers. Вариант 2

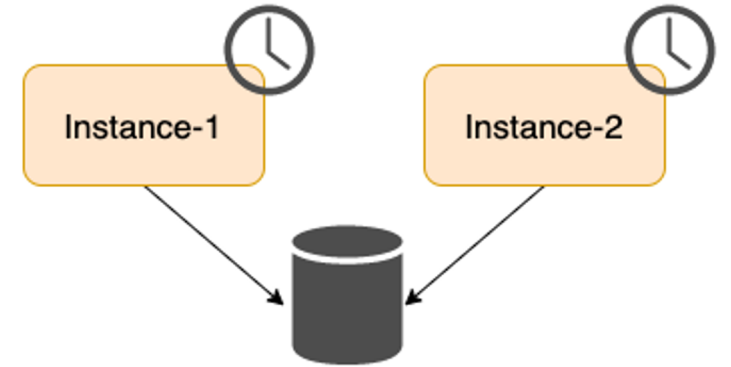


Атомарные изменения в БД

Select-for-update / CAS

~~Pull из сторонних сервисов~~

Параллельная обработка



Schedulers. Select-For-Update

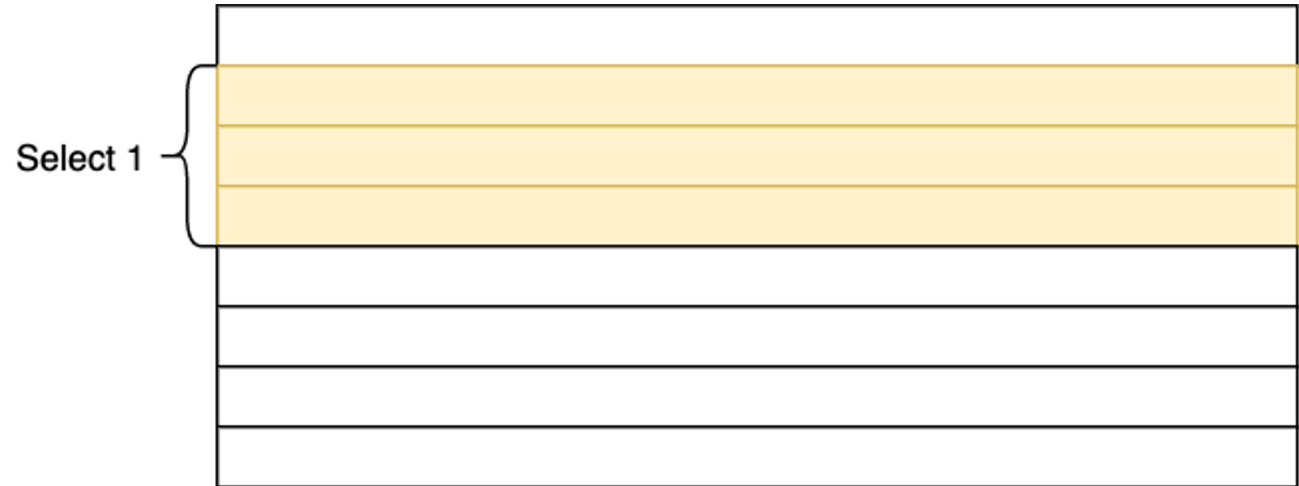


```
UPDATE <table>
SET <column> = :value
WHERE id in (
    SELECT t.id FROM <table> t
    WHERE <condition>
    LIMIT :limit
    FOR UPDATE SKIP LOCKED)
RETURNING *
```

Schedulers. Select-For-Update



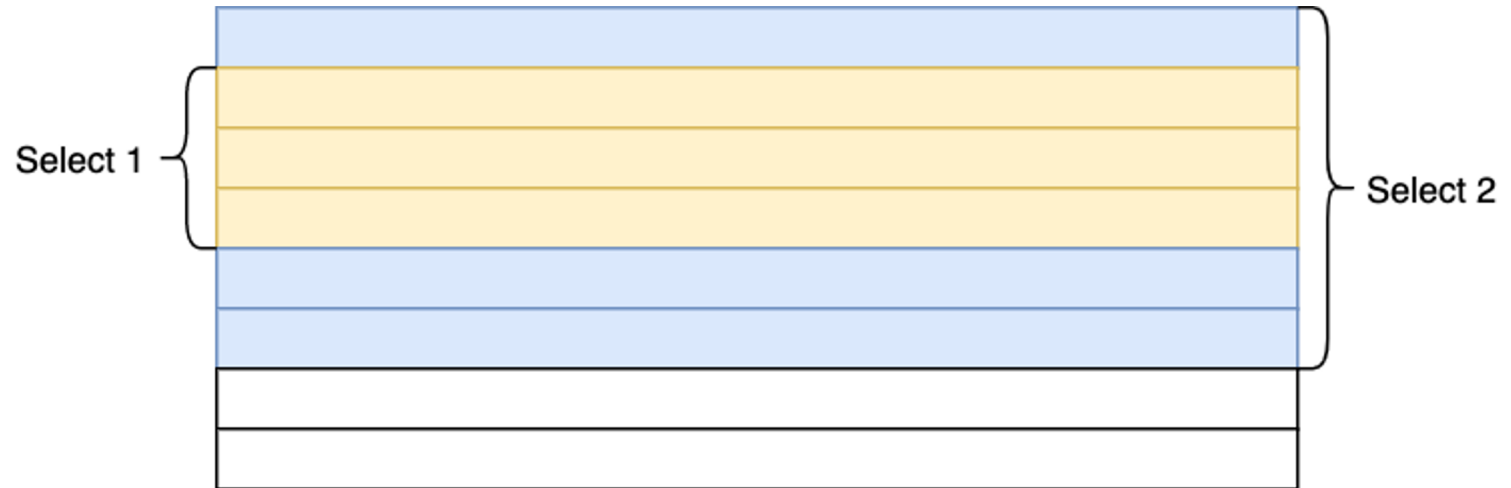
```
UPDATE <table>
SET <column> = :value
WHERE id in (
    SELECT t.id FROM <table> t
    WHERE <condition>
    LIMIT :limit
    FOR UPDATE SKIP LOCKED)
RETURNING *
```



Schedulers. Select-For-Update



```
UPDATE <table>
SET <column> = :value
WHERE id in (
    SELECT t.id FROM <table> t
    WHERE <condition>
    LIMIT :limit
    FOR UPDATE SKIP LOCKED)
RETURNING *
```



Schedulers. Вариант 3

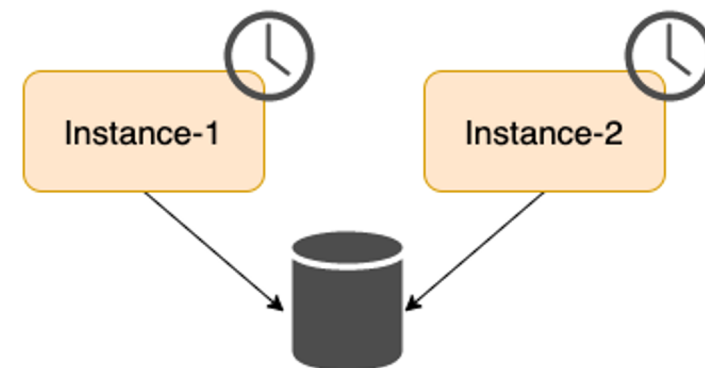


ShedLock

~~Failover~~

Не трогаем код

Синхронизация через БД

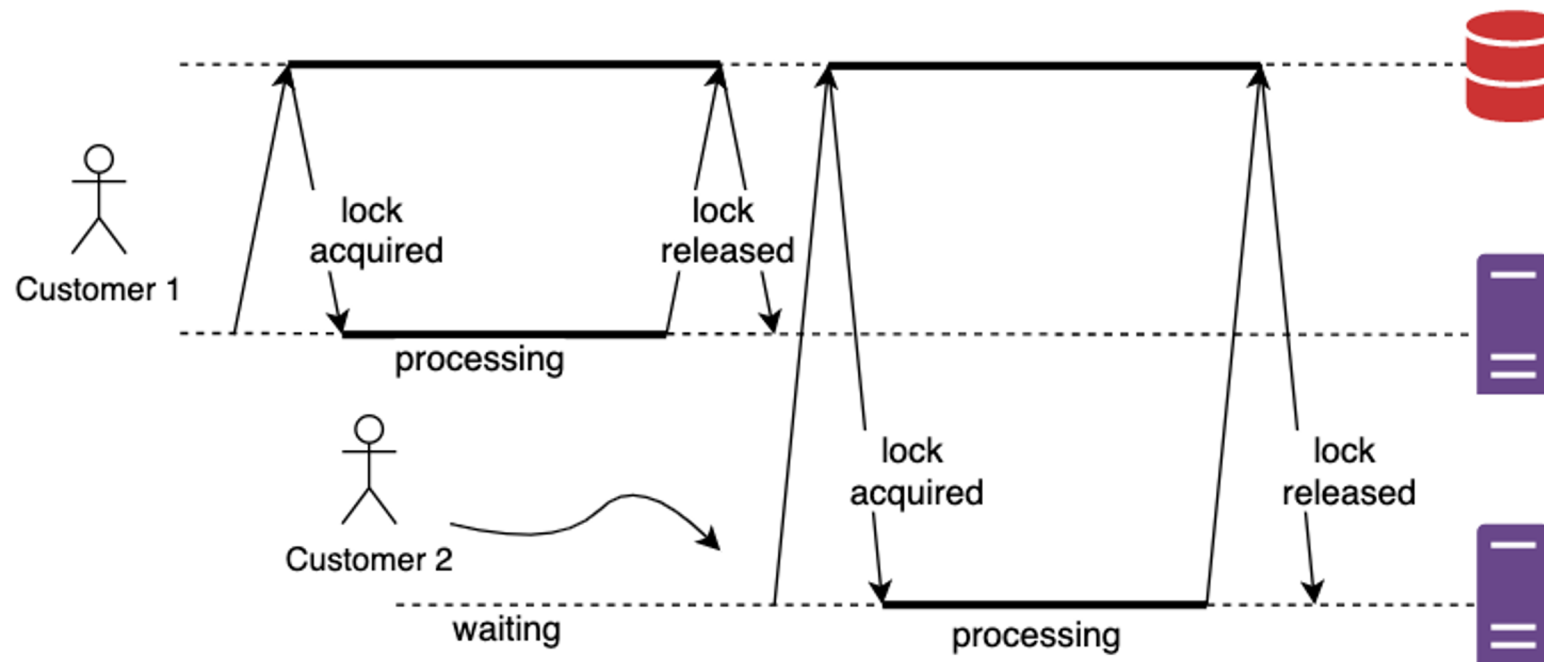


Shedlock

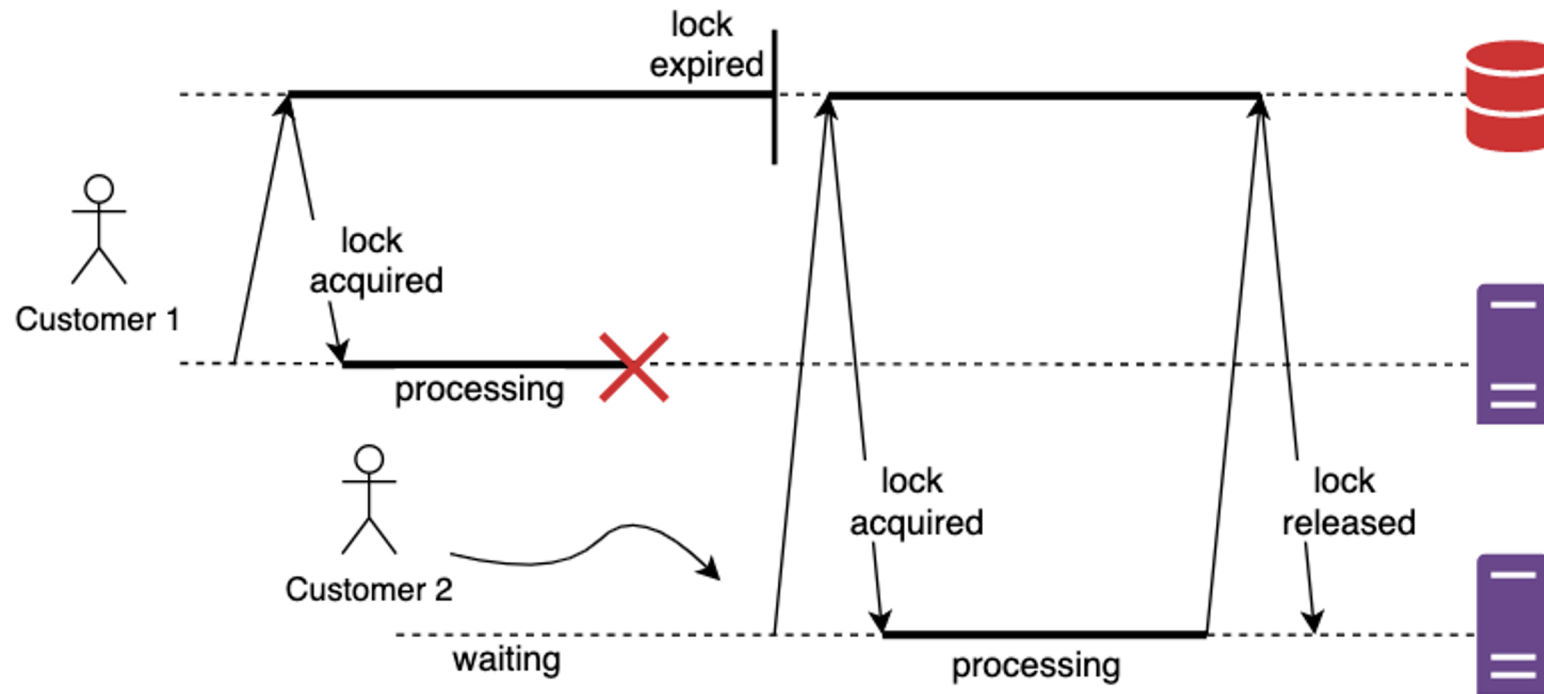


name	lock_until	locked_at	locked_by
schedulerName1	2021-08-15 14:49:40.008956	2021-08-15 14:40:12.020511	appInstance5
schedulerName2	2021-08-15 14:57:13.104003	2021-08-15 14:52:05.862010	appInstance3

Pessimistic blocking



Pessimistic blocking



Demo



ShedLock

Select-for-update

Schedulers



1. Distributed schedulers
2. Select-for-update / CAS
3. ShedLock

Потенциальные проблемы



Schedulers

Sessions

Cache

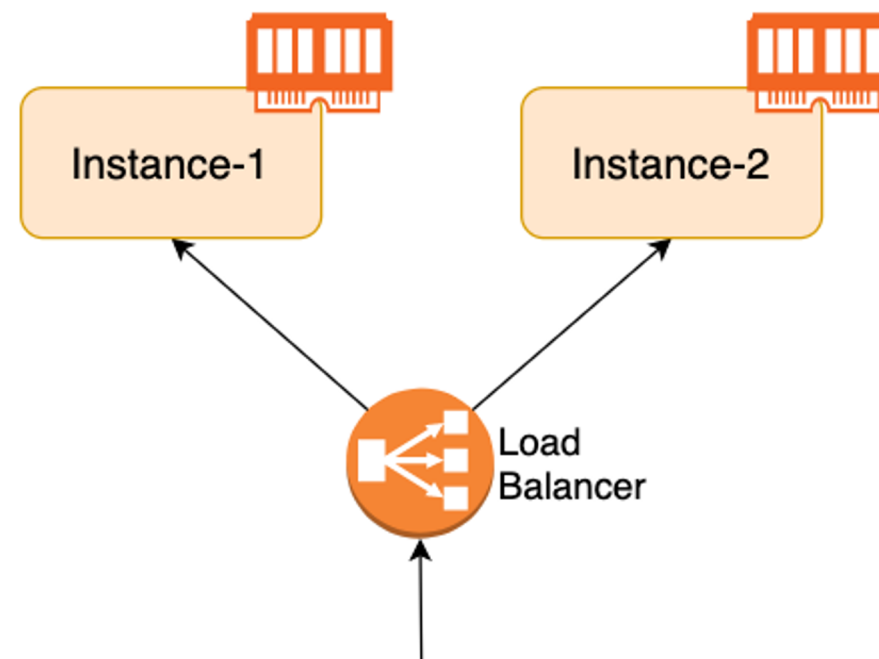
WebSockets



Sessions



1. Повторный логин
2. Перезапуск инстанса

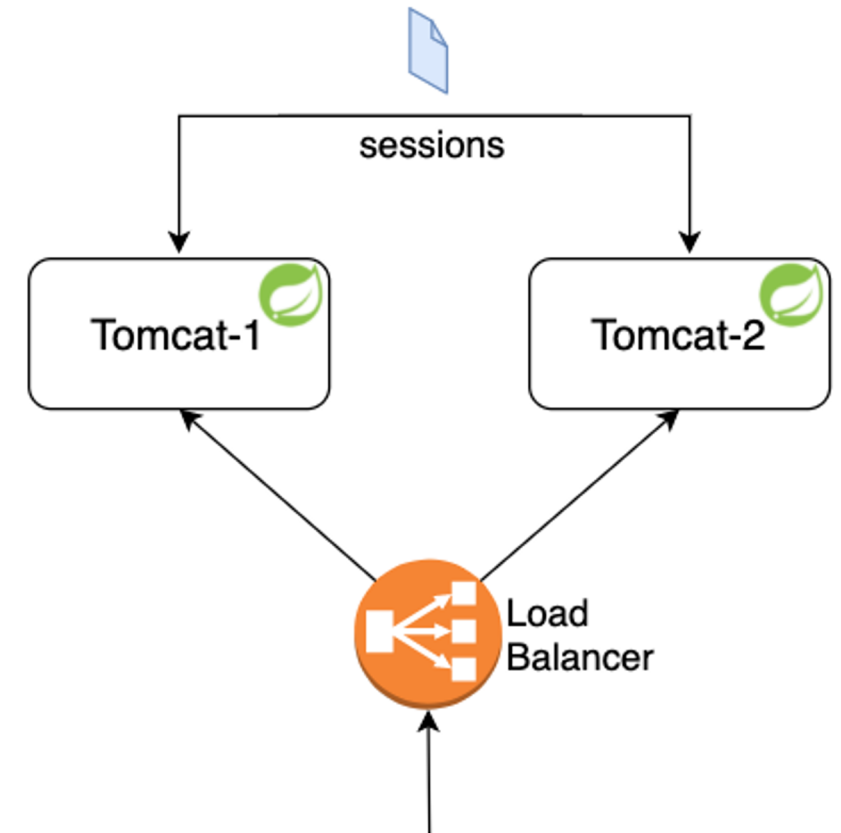


Sessions. Вариант 1

Tomcat clustering

Multicast ping

Sessions replication



Sessions. Вариант 2

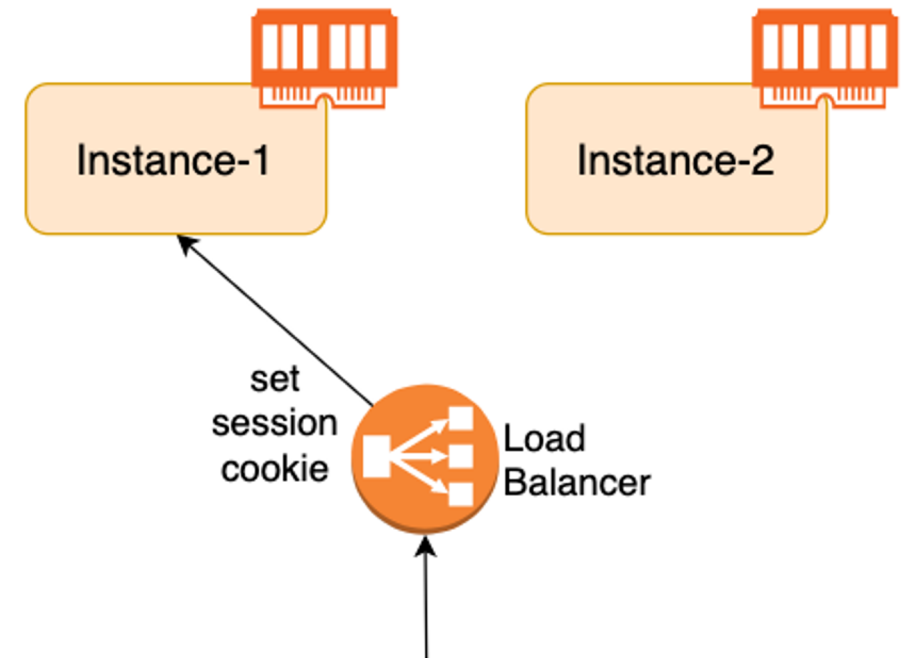


Sticky sessions

Sticky Cookie

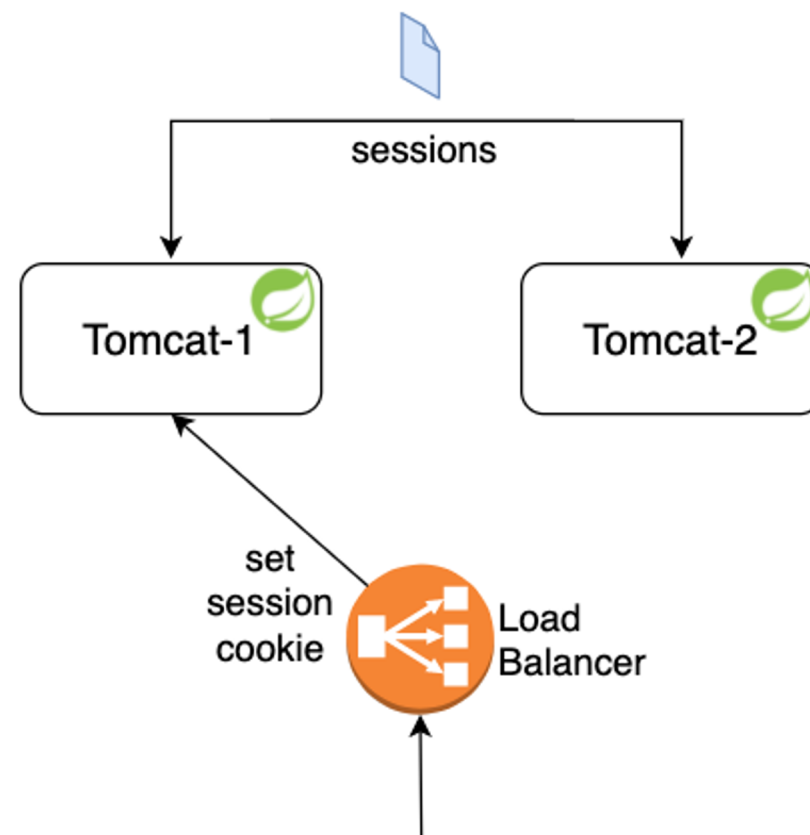
Тяжелые сессии (Vaadin/JSF)

~~Last seen~~



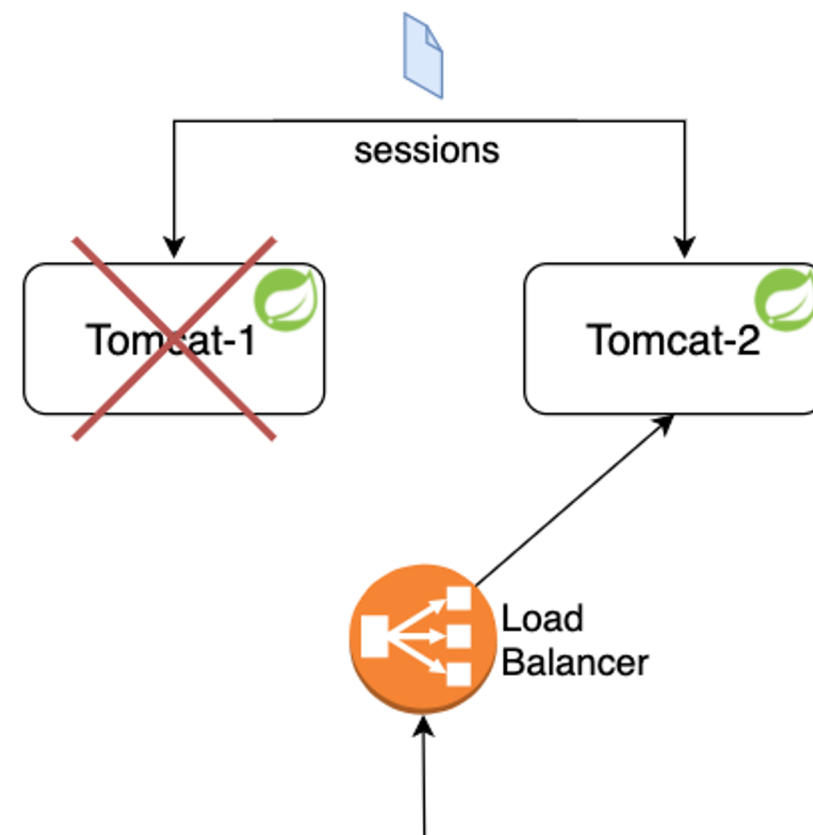
Sessions. Вариант 3

Tomcat clustering + Sticky sessions



Sessions. Вариант 3

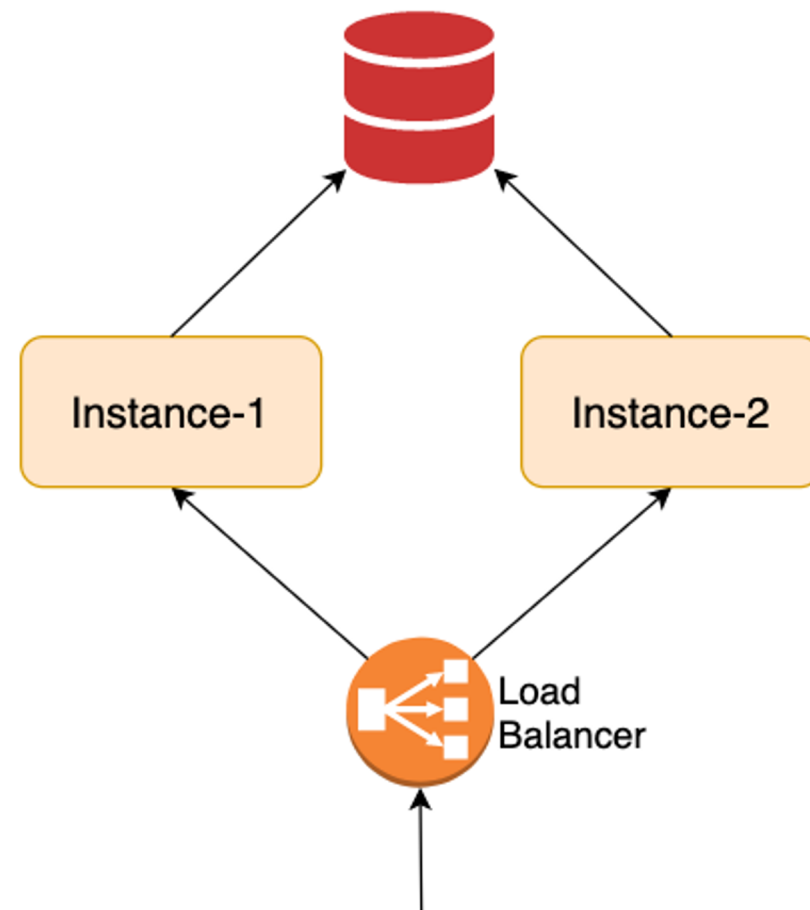
Tomcat clustering + Sticky sessions



Sessions. Вариант 4

Persistent

Любое хранилище
Переживает рестарт



Demo



JDBC sessions

Потенциальные проблемы

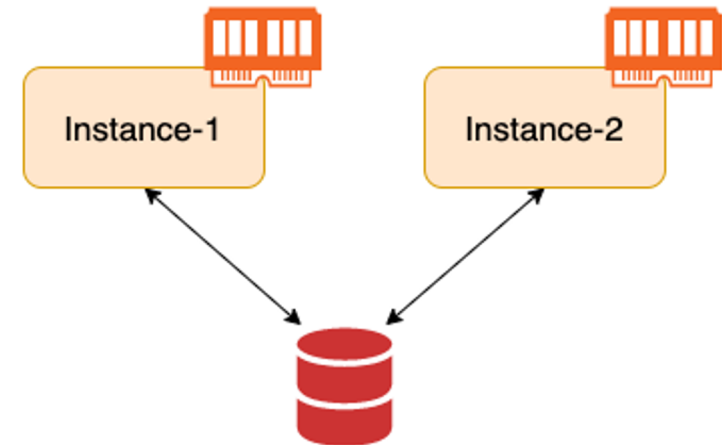


- ✓ Schedulers
- ✓ Sessions
- Cache
- WebSockets



Cache

1. Несогласованный
2. Ответ зависит от инстанса



Cache. Вариант 1

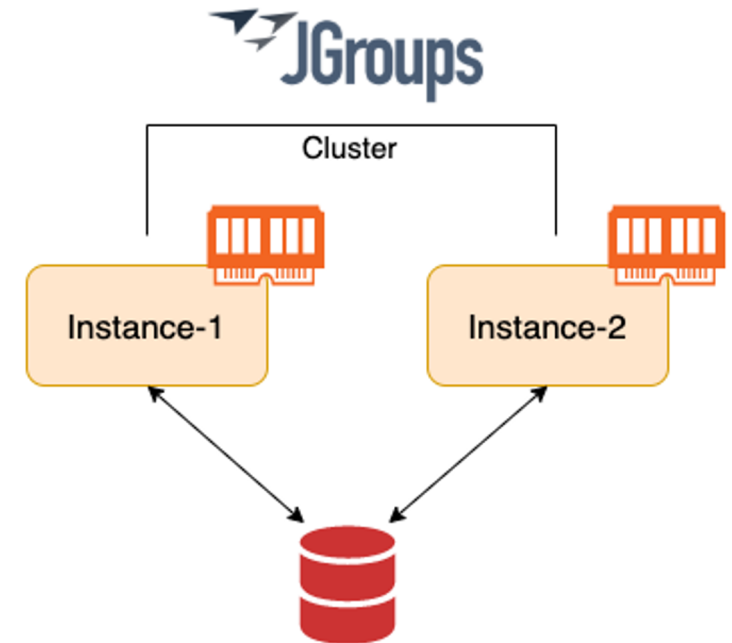


Nodes synchronization

Membership

Node-to-Node

Node-to-Cluster

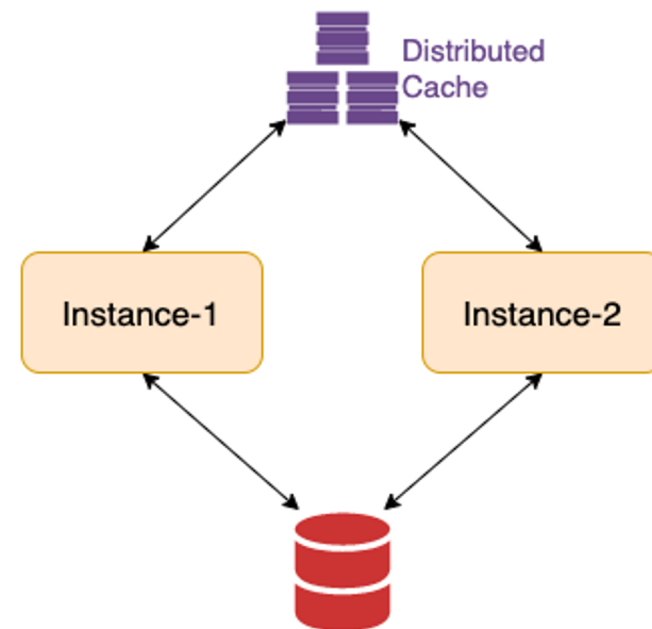


Cache. Вариант 2



Distributed cache

Redis
Hazelcast
Apache Ignite



Cache. Вариант 3

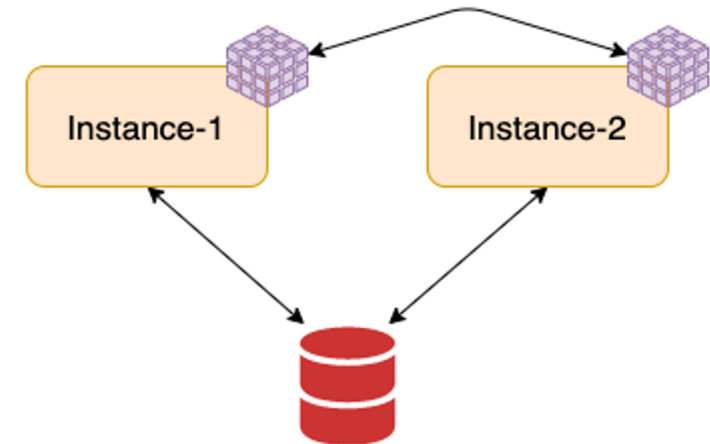


Embedded (Apache Ignite, Hazelcast)

Sharding, Replicas

Same JVM

~~Network latency~~

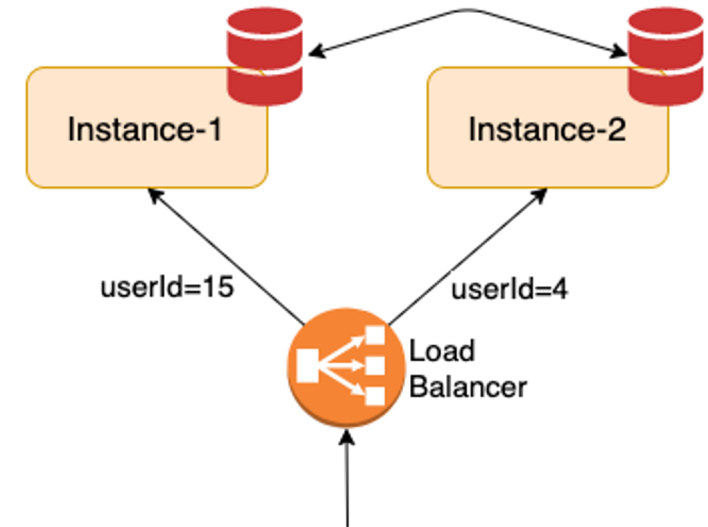


Stateful Microservices



Embedded Apache Ignite

Smart Load Balancer
Affinity collocation



Demo



Add Redis for Spring Cache

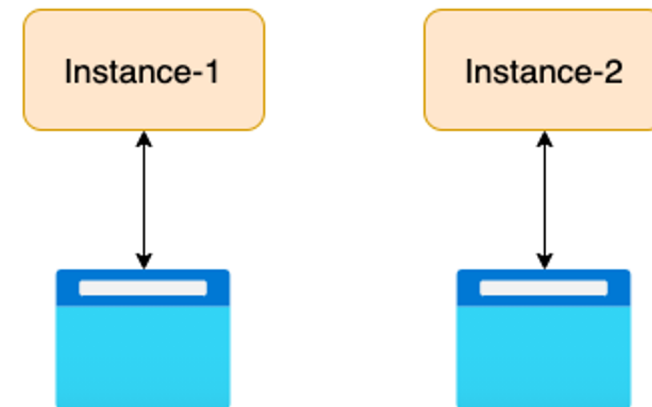
Потенциальные проблемы



- ✓ Schedulers
- ✓ Sessions
- ✓ Cache
- WebSockets



WebSockets

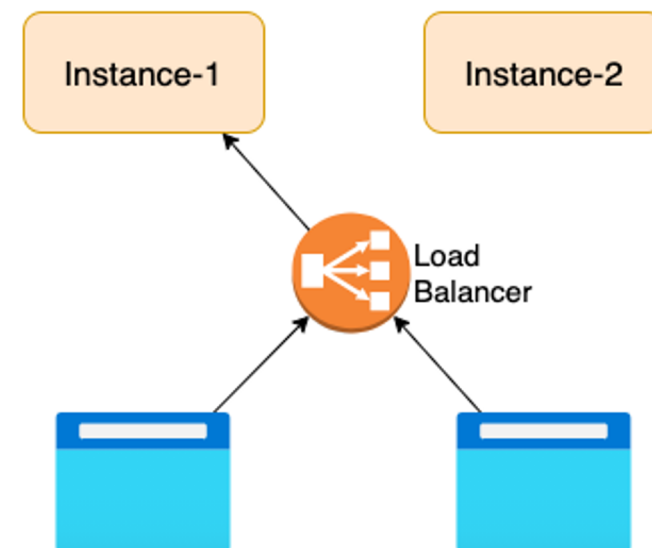


WebSockets



Load balancer

WebSocket на каждый чат
Перегрузка инстанса



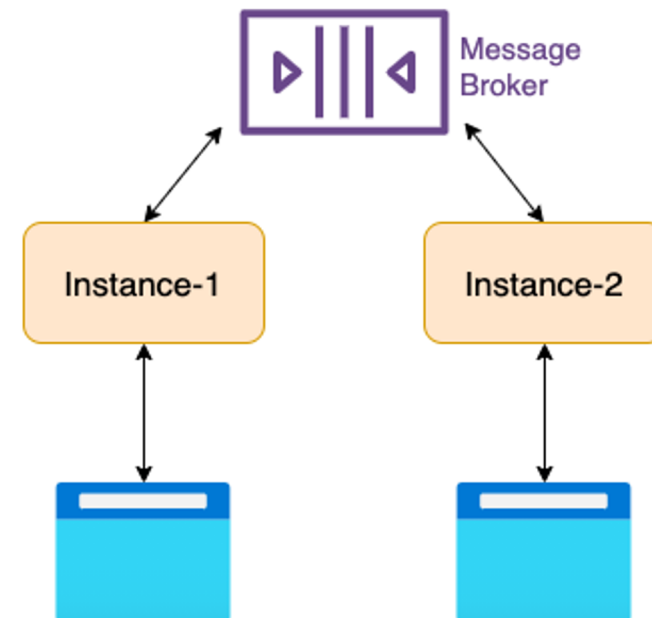
WebSockets



External Message Broker

Делегирование обработки

Горизонтальная масштабируемость



WebSockets

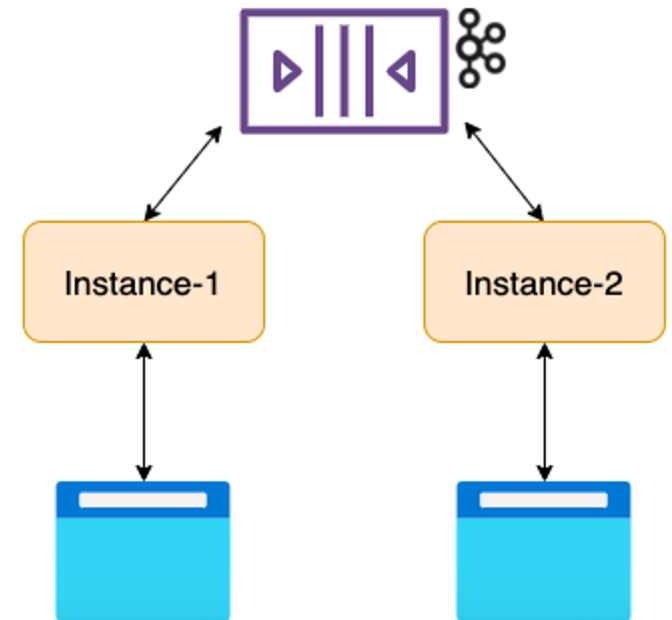


Apache Kafka

Persistent

~~Dynamic topics~~

~~Out of box~~



WebSockets

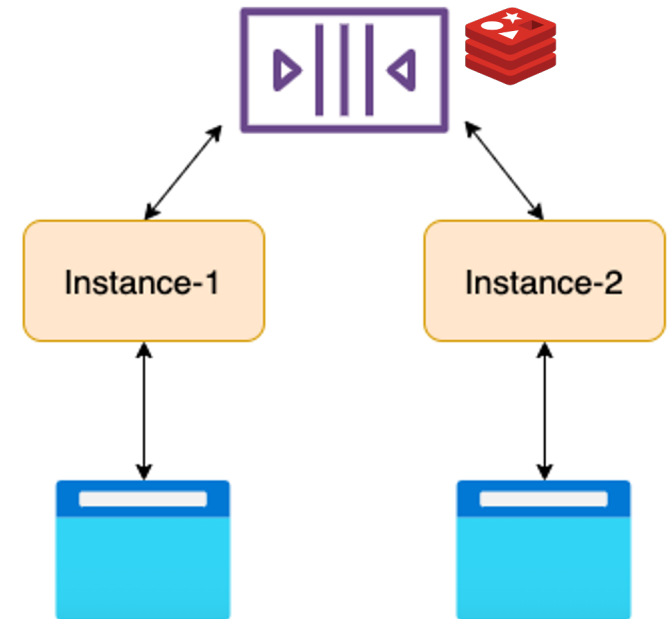


Redis PUB/SUB

Dynamic topics

Pull-model

Out of box



WebSockets

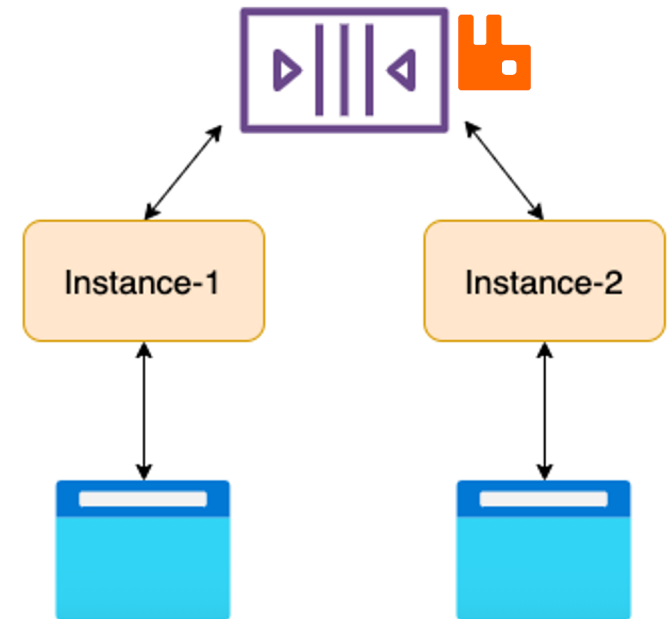


RabbitMQ / ActiveMQ

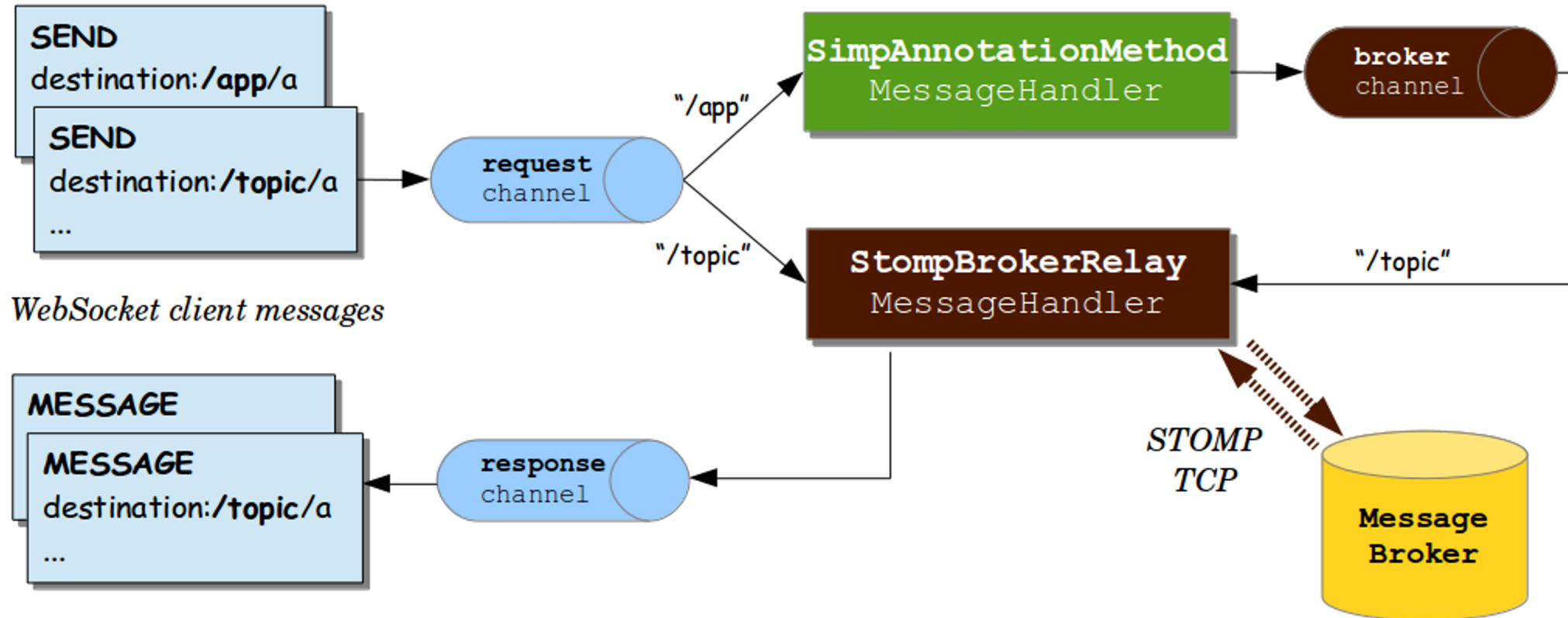
Dynamic topics

Push-model

Out of box



WebSockets



<https://docs.spring.io/spring-framework/docs/current/reference/html/web.html#websocket>

Demo



Enable StompBrokerRelay

WebSockets. @SubscribeMapping



1. @GetMapping

Переписывание клиентов

Релизный цикл

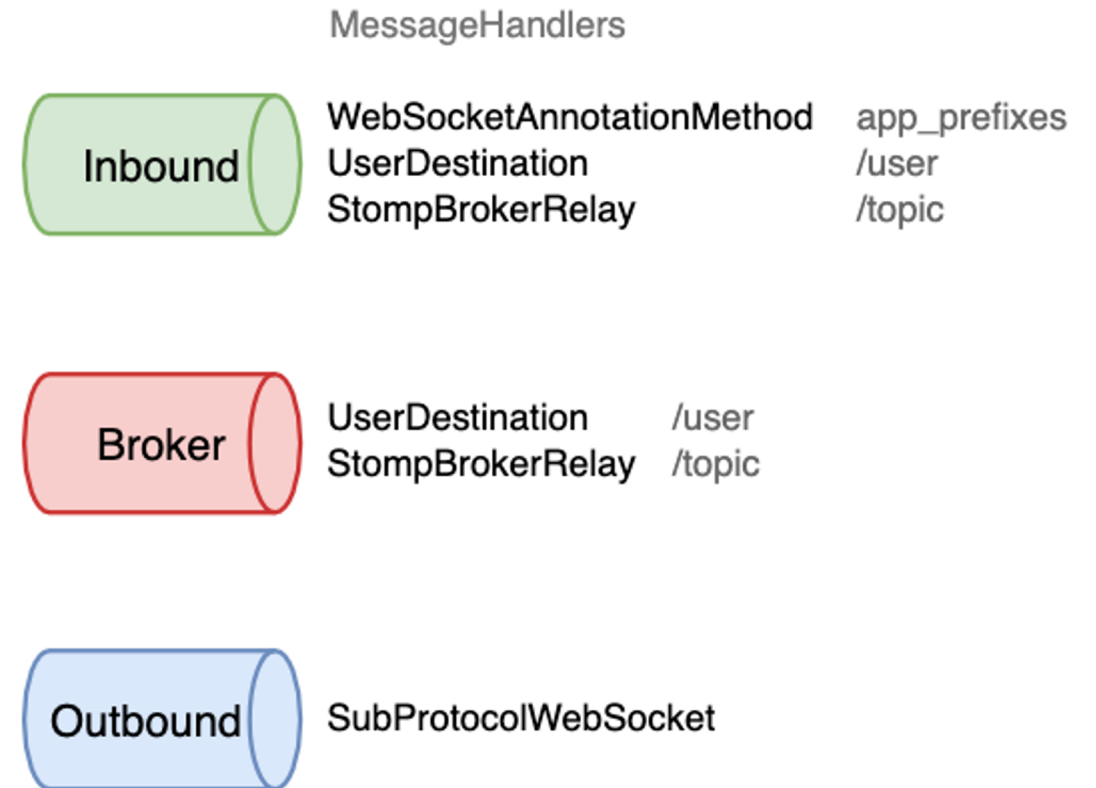
1. BeanPostProcessor

Искусственное замедление

Дублирование всем подписчикам

1. Встраивание хэндлера для /topic, /user

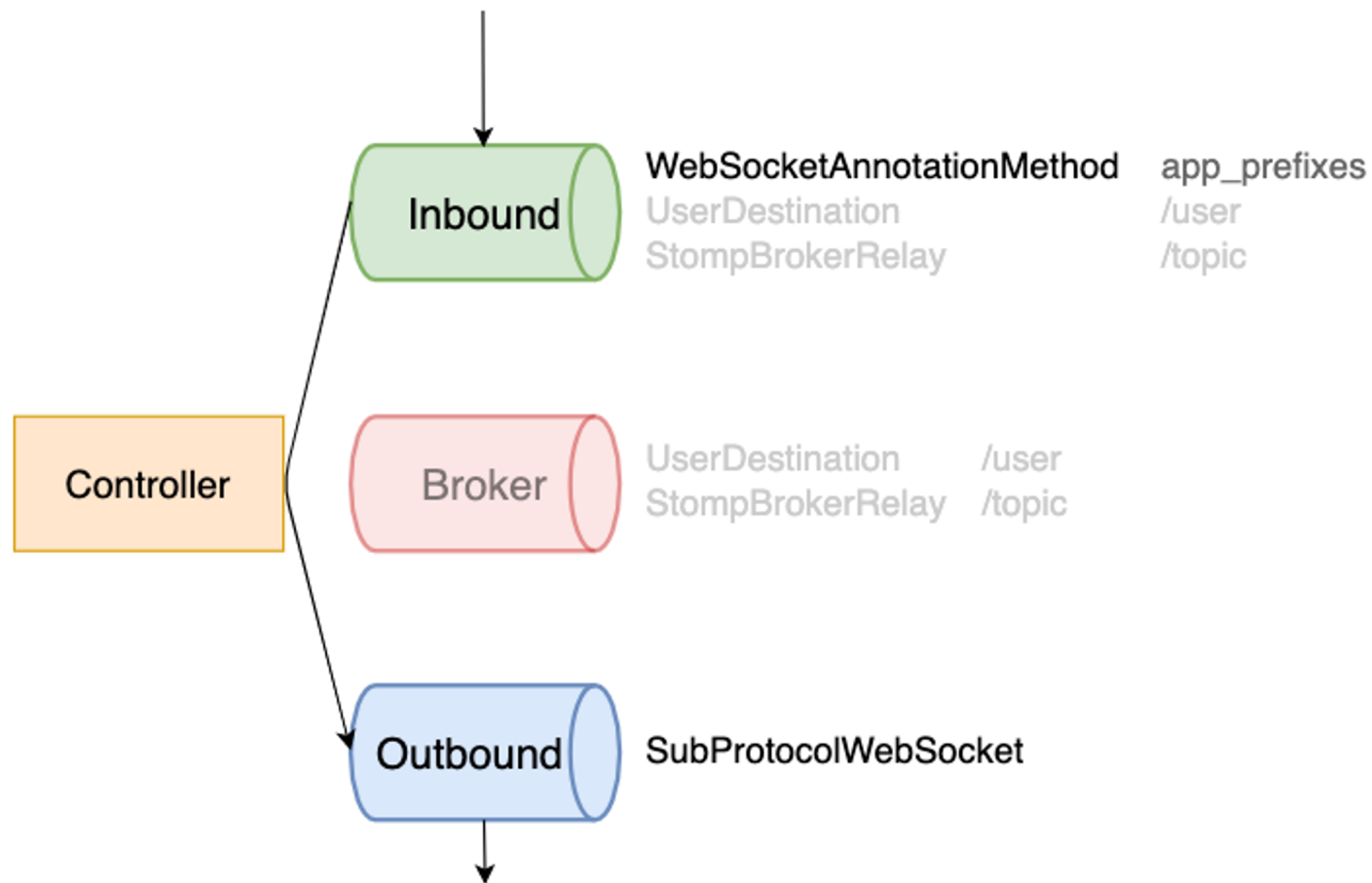
WebSockets. Channels



WebSockets. Channels



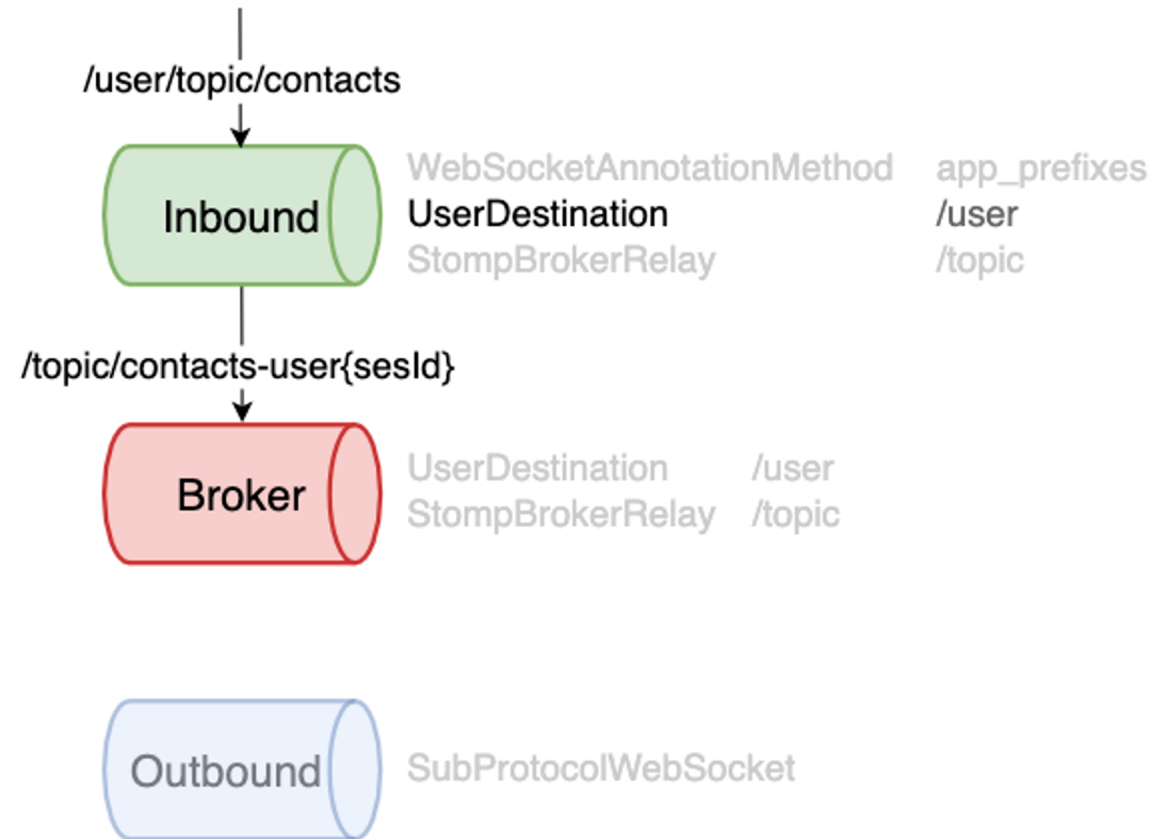
Обработка контроллером



WebSockets. Channels



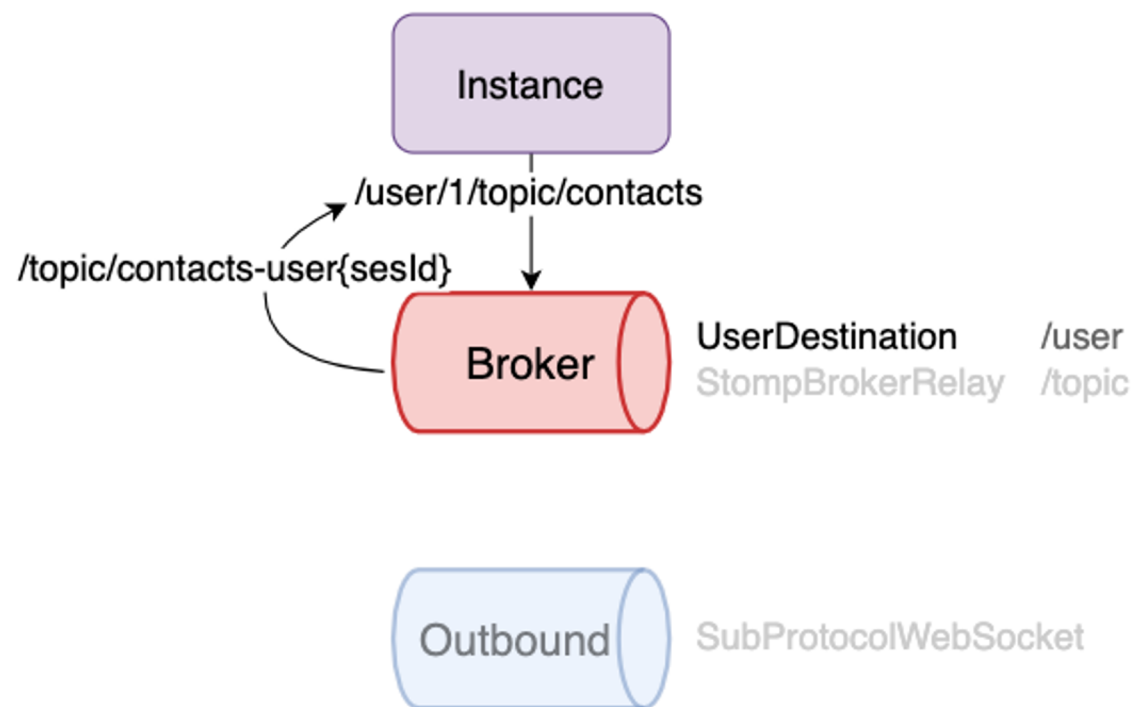
User resolve



WebSockets. Channels



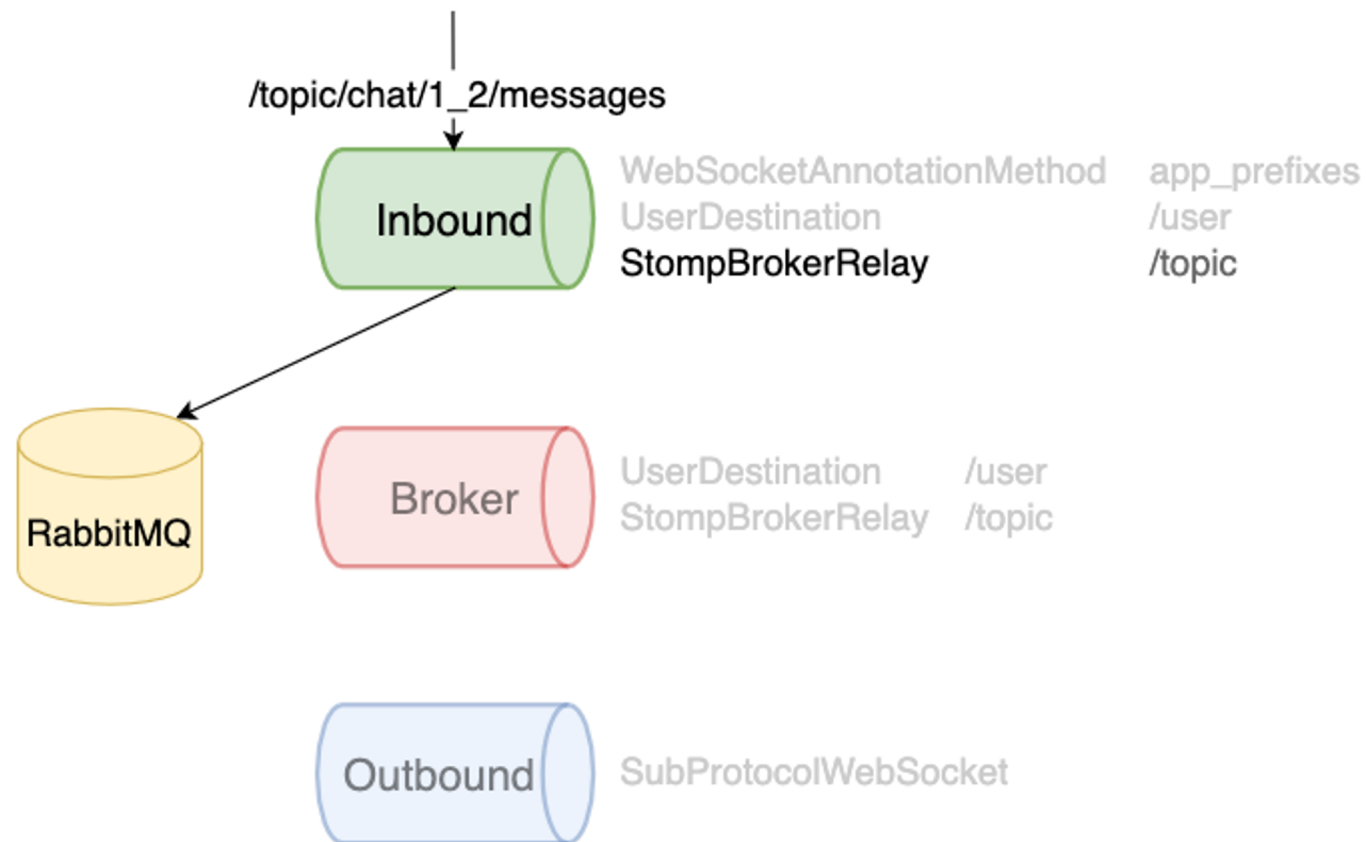
User resolve



WebSockets. Channels



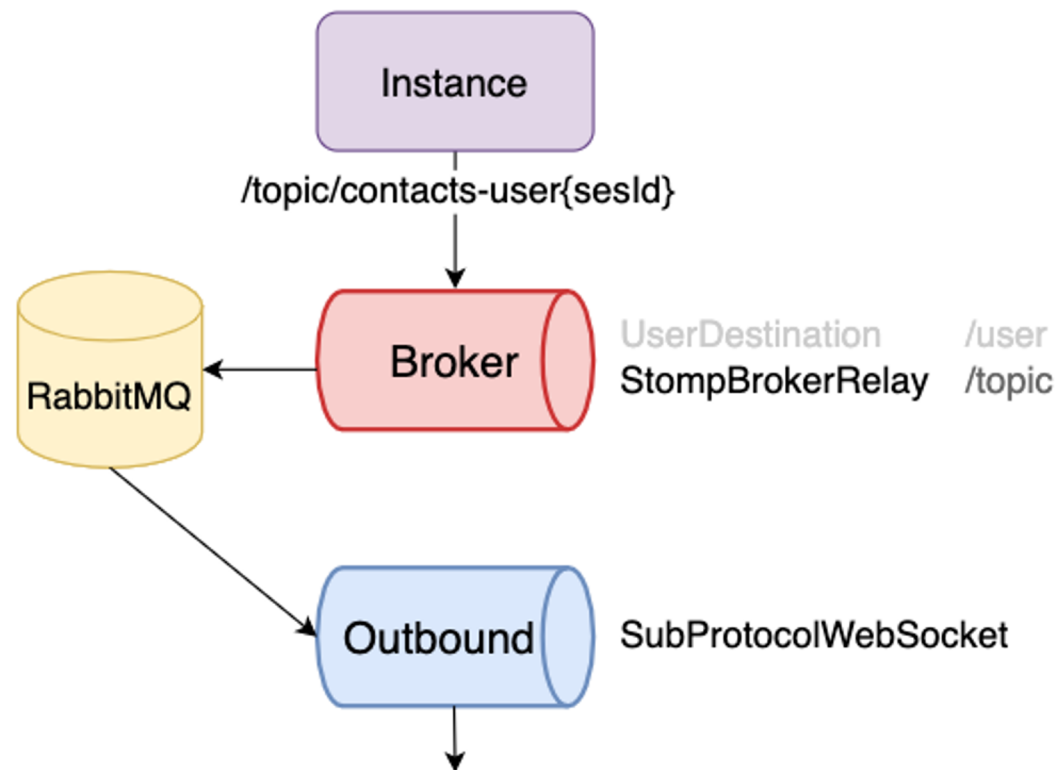
Отправка в брокер



WebSockets. Channels



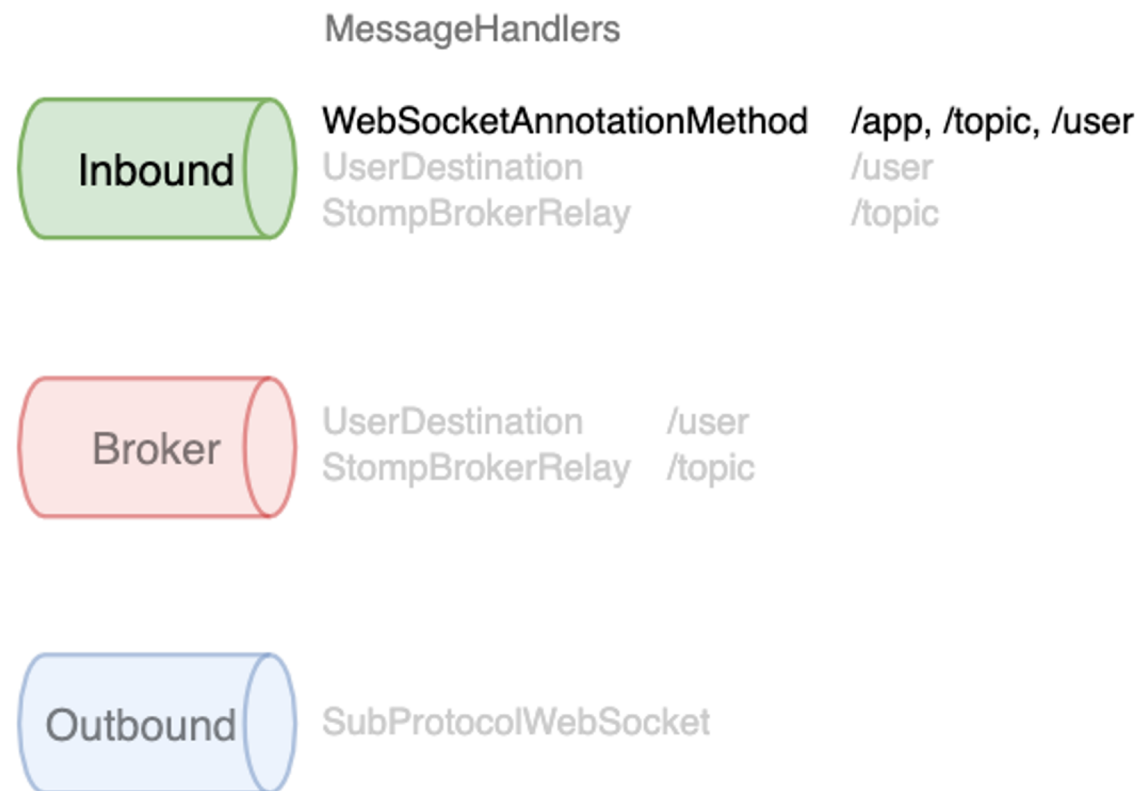
Отправка в брокер



WebSockets. Channels



App prefixes

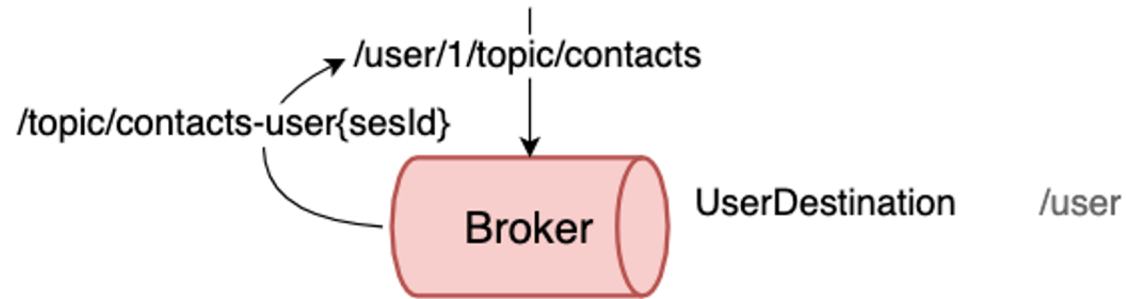


Demo

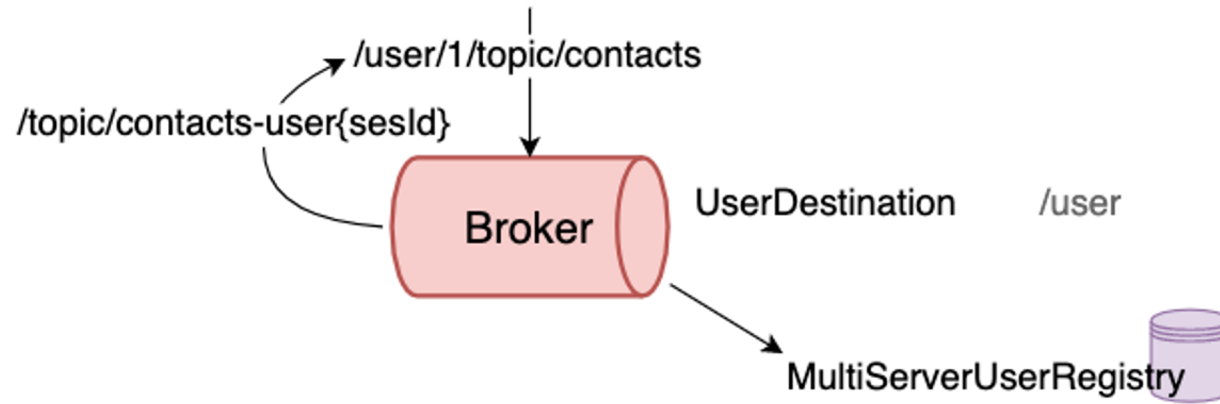


Fix WebSockets

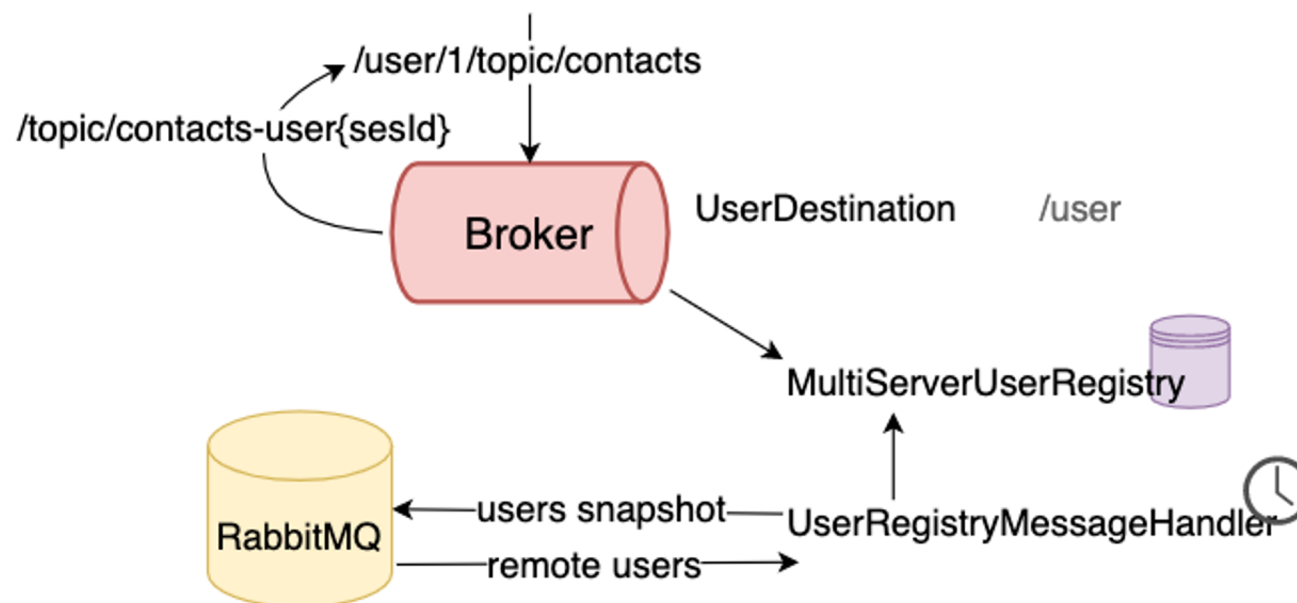
WebSockets. Users broadcast



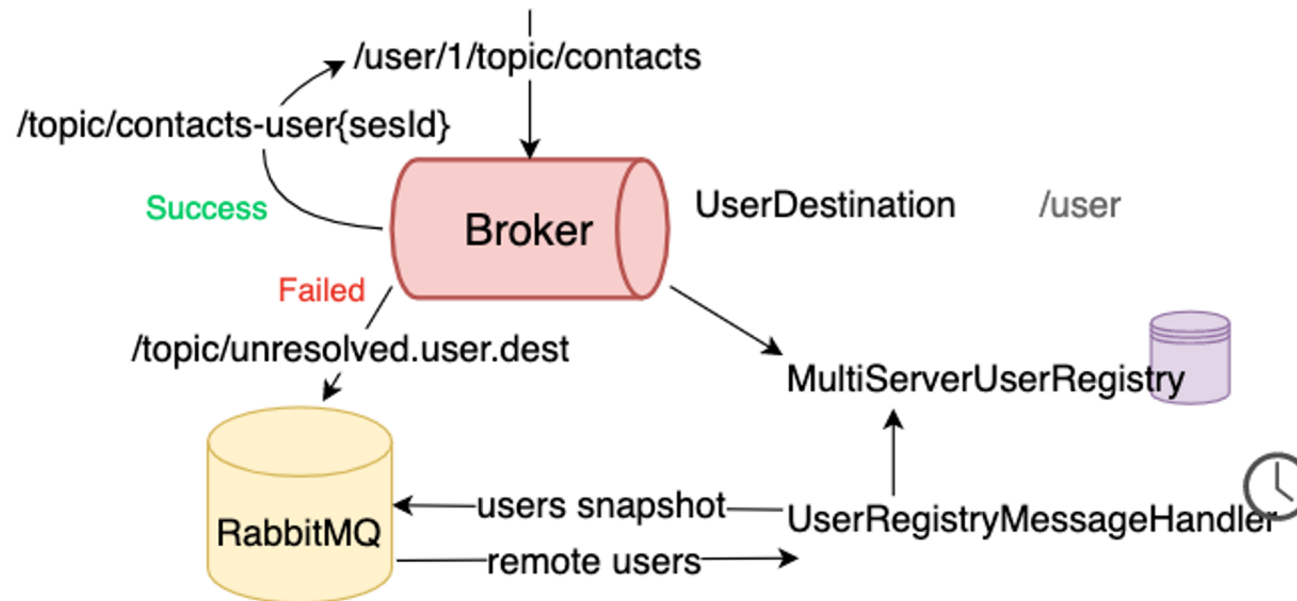
WebSockets. Users broadcast



WebSockets. Users broadcast



WebSockets. Users broadcast



Demo



Add users broadcast

Потенциальные проблемы



- ✓ Schedulers
- ✓ Sessions
- ✓ Cache
- ✓ WebSockets



Summary



1. Schedulers

- Distributes scheduler
- Атомарные изменения в БД
- ShedLock

1. Sessions

- Tomcat clustering
- Sticky sessions
- Persistent

1. Cache

- Nodes synchronization
- Distributed cache
- Embedded mode

1. WebSockets

- External broker
- @SubscribeMapping
- User broadcast



Ваши вопросы



Александр Коженков

Team Lead в Cloud/Tooling команде компании
GridGain

<https://github.com/wirtsleg/scaling-spring-boot-microservices>

 wirtsleg

 wirtzleg

 wirtzleg@gmail.com

Спасибо за внимание!

