



# Осторожно PGPOOL

Игорь Буренков

ОКТАБРЬ 2017

25+ YEARS OF ENGINEERING LEADERSHIP



# Требования

- › Производительность

  - XX млн. запросов в сутки

  - latency в пределах 100мс

- › Масштабирование и отказоустойчивость

  - на уровне сервиса

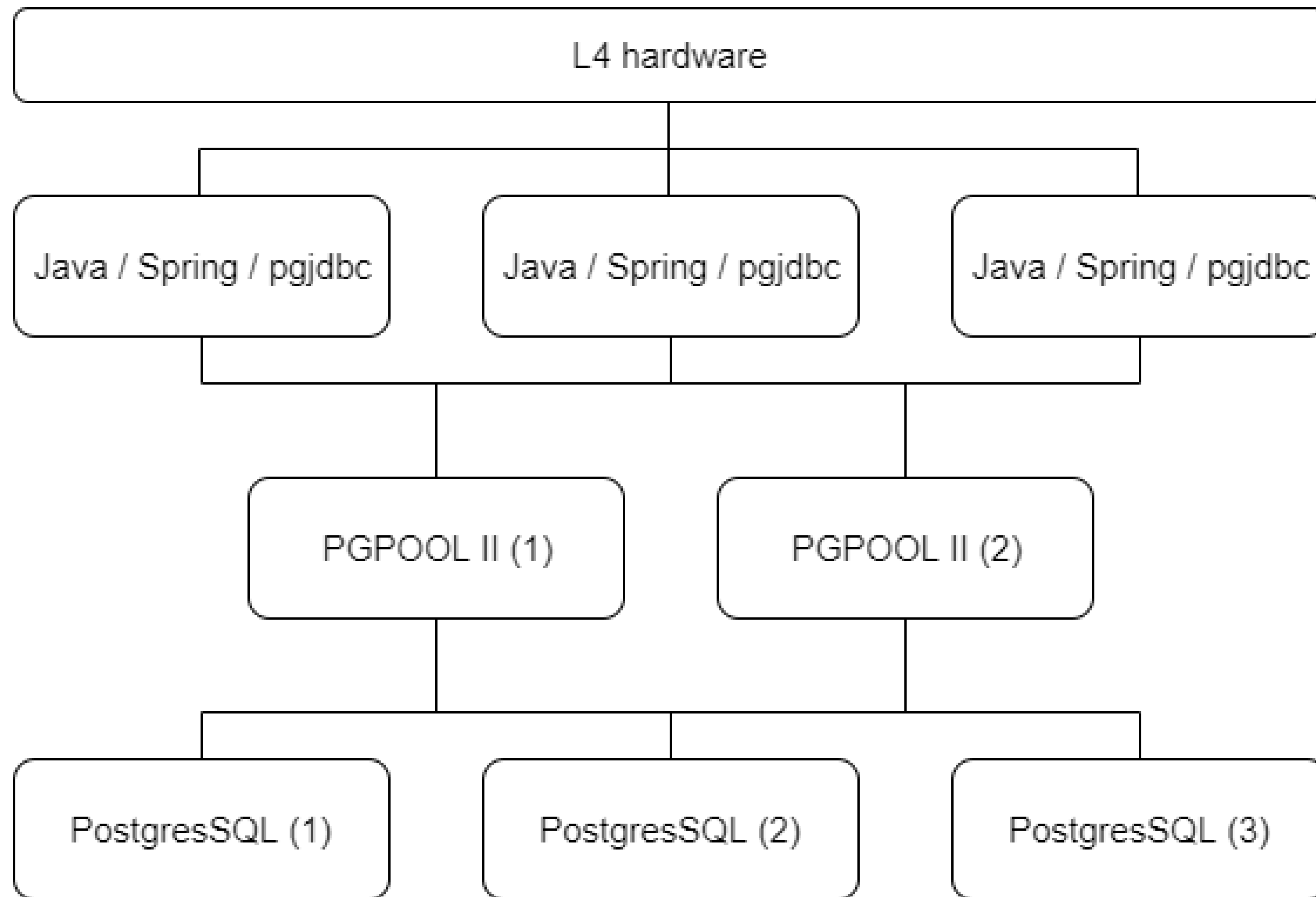
  - на уровне источника данных

- › PostgreSQL + PGPOOL

# PGPOOL-II это

- › Пул подключений
- › Ограничение количества подключений
- › Репликация
- › Балансировка

# Архитектура



- › Горизонтальное масштабирование
- › Дублирование
- › Балансировка
- › Сервис управления PGPOOL

# Тестирование

Что делали:

- › Jmeter 10threads
- › Отключение серверов
- › VM, docker pause
- › Сетевые проблемы
- › Миграция VM между хостами

Что мерили:

- › время обработки запроса
- › время на обработку в базе
- › время на обработку запроса в коде

# Fail

- › Время на запрос в пределах **200..600мс**
- › **Ошибки в приложении** при attach/detach PG в пуле
- › **“Зависания”** при attach/detach PG в пуле
- › Самопроизвольное **“Выключение”** PG из пула (failover)
- › **Рассинхронизация** состояний PGPOOL
- › Какой PGPOOL активен?

# Рассинхронизация и “Выключение” PG из пула

## Источник

- › Проблемы с подключением к PG
- › PGPOOL-II Failover

## Решение

- › Внешний сервис failback

# Ошибки и зависания при attach/detach PG

## Источник

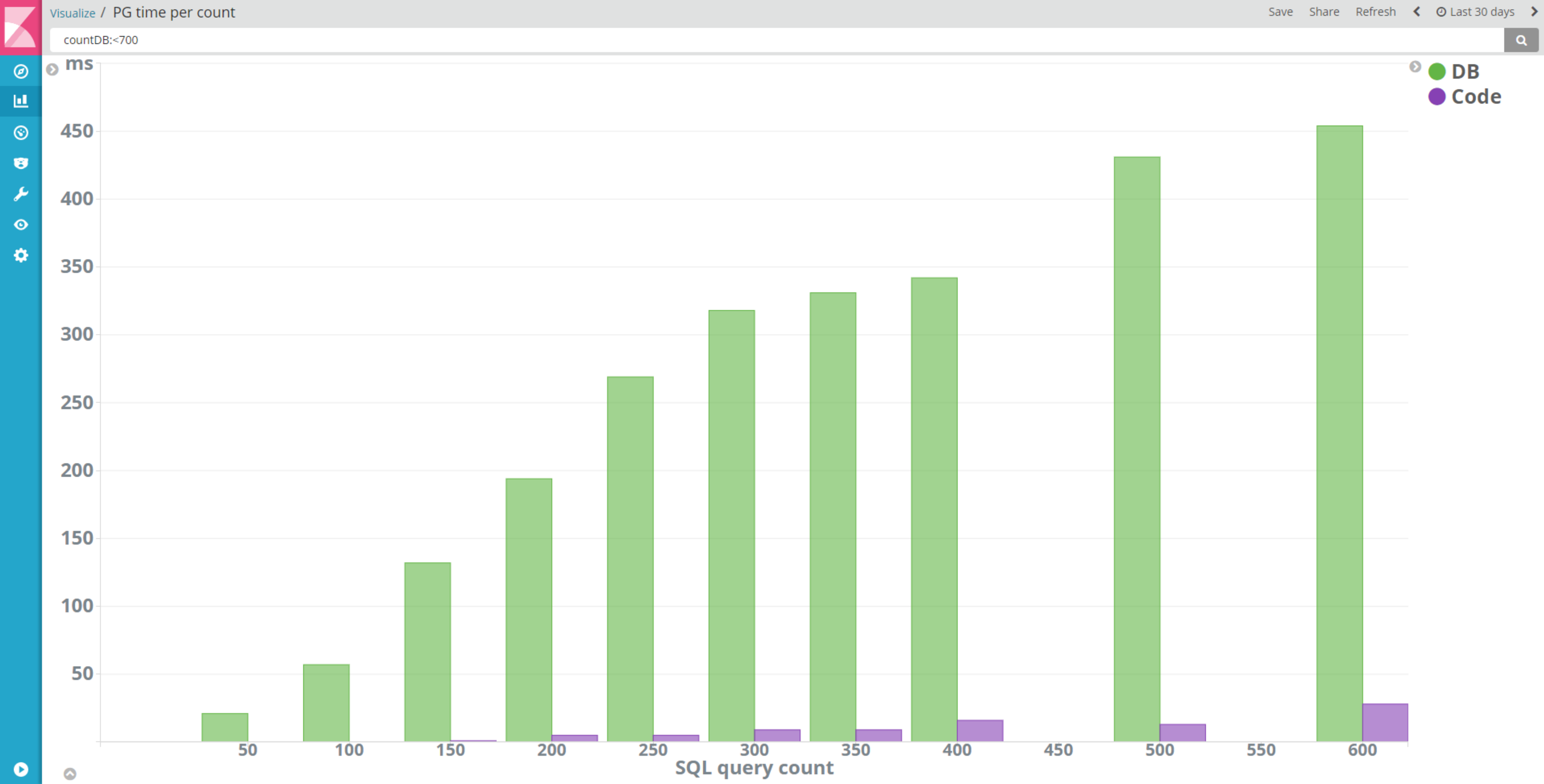
- › Настройки PGPOOL

## Решение

- › Зависания при вводе/выводе
- › Ошибки при вводе
- › Ошибки при выводе



# Время обработки запроса



# Время обработки запроса

## Источник

- › Напрямую к базе в разы быстрее!!!
- › Настройки PG и пула ?

## Решение

- › Уменьшить количество запросов
- › Работать напрямую с PG

# PGPOOL II Итог

- › Медленно при большом количестве SQL запросов
- › Ошибки в приложении при выводе PG из пула
  - Повтор запроса из приложения
- › Самопроизвольное “Выключение” PG из пула (failover)
  - Внешний сервис
- › Рассинхронизация состояний PGPOOL-II
  - Внешний сервис

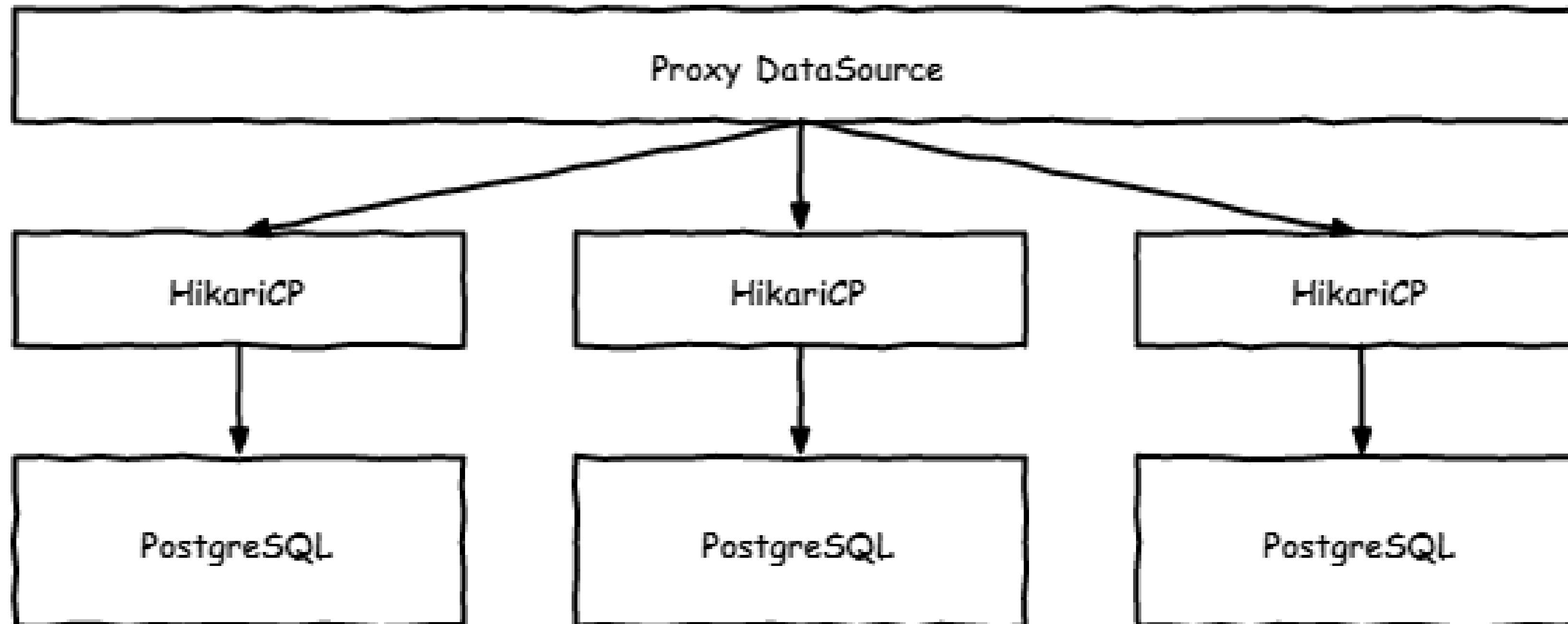
# Альтернатива

- › HAProxy
- › PgBouncer
- › HA-JDBC

# Proxy DataSource

- › Обращение напрямую к базе
- › Пул коннектов HikariCP
- › Полный контроль failover/failback
- › Управление пулом PG
- › Удаление и добавление PG "на ходу"

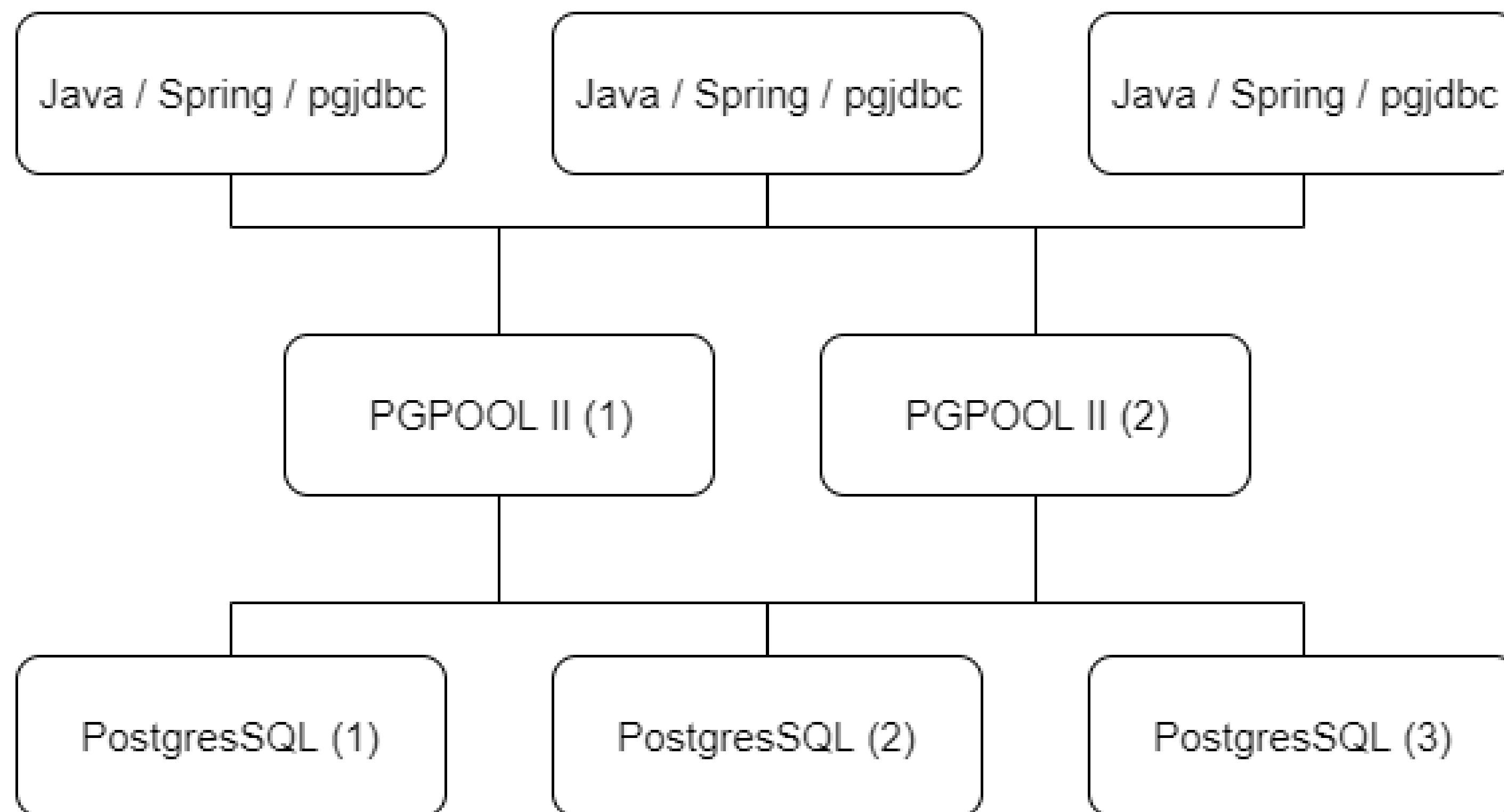
# DataSource pool



# DataSource Failover/Failback

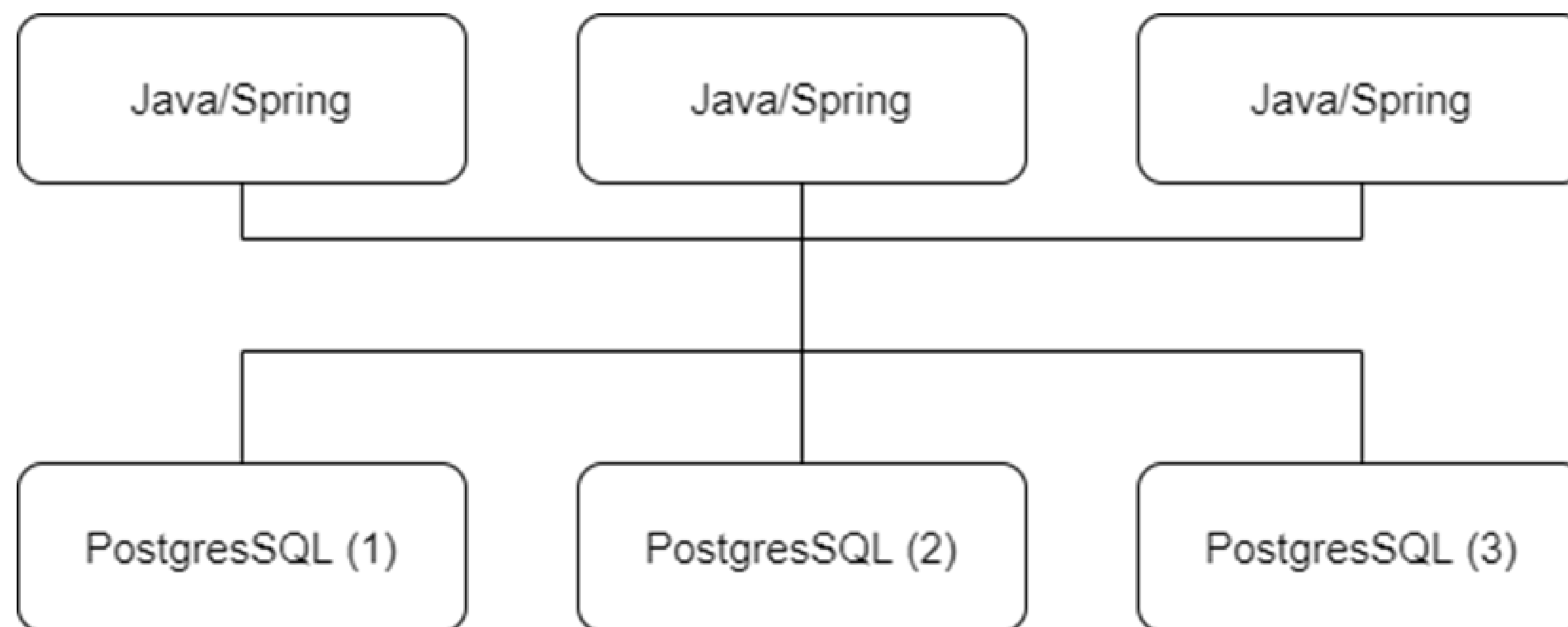
- › Random балансировка
- › Ловим ошибки `java.sql.*Statement`, `java.sql.Connection`
- › Реагируем только на `SQLException` с кодами кодами `08*` `NZ*` или `IOException`
- › База отключается сразу, только при возникновении ошибки
- › Отключенные базы постоянно мониторятся на доступность

# Архитектура (было)





# Архитектура (стало)



# ИТОГ

- › Прямое обращение к PG без “прокладки”
- › Полный контроль над Failover/Failback
- › Экономия “железа”
- › Поддерживает любую базу если есть jdbc драйвер



Спасибо за внимание!

Задавайте вопросы.

Игорь Буренков

[igor.burenkov@auriga.com](mailto:igor.burenkov@auriga.com)



25+ YEARS OF ENGINEERING LEADERSHIP

**АУРИГА**

**ОФИС В БОСТОНЕ**

400 TradeCenter Ste 5900  
Woburn, MA 01801, USA

Тел: +1 (866) 645-1119  
Факс: + 1 (603) 386-6097

**ОФИС В ЕВРОПЕ**

Ukmergės g. 322, LT- 12106,  
Vilnius, Lithuania

Тел: +370 (5) 203-1571  
Факс: +370 (5) 203-1573

**ОФИС В РОССИИ**

117587, Россия, г. Москва,  
Варшавское ш., д. 125, стр. 16А

Тел: +7 (495) 713-9900  
Факс: +7 (495) 939-0300

[sales@auriga.com](mailto:sales@auriga.com)  
[www.auriga.ru](http://www.auriga.ru)

