

# От PostgreSQL к Tarantool

**выбор в отсутствие выбора**



**Любовь Вайнштейн**

РТ Лабс | СМЭВ

ЗАМЕСТИТЕЛЬ ДИРЕКТОРА  
ДЕПАРТАМЕНТА РАЗРАБОТКИ

- Стабильность производства
- Сквозная интеграция решений
- Сотрудничество команд



**Евгений Демкин**

РТ Лабс | СМЭВ

ГЛАВНЫЙ АРХИТЕКТОР

- Производительность
- Отказоустойчивость
- Мониторинг и диагностика



**Дмитрий Кольцов**

Пикодата | Tarantool

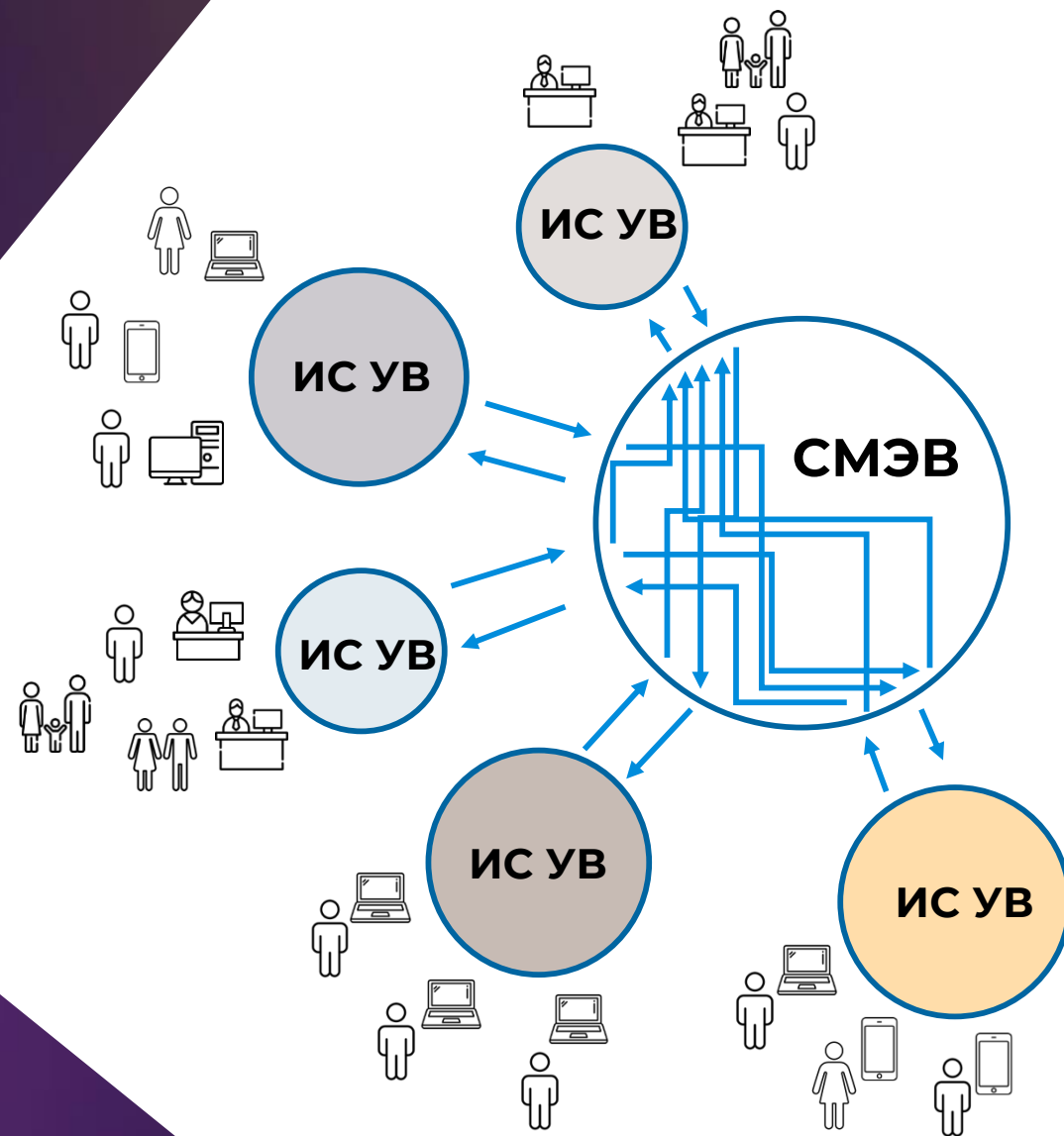
РУКОВОДИТЕЛЬ ГРУППЫ  
РАЗРАБОТКИ

- Производительность
- Доступность
- Масштабируемость

# СИСТЕМА МЕЖВЕДОМСТВЕННОГО ЭЛЕКТРОННОГО ВЗАИМОДЕЙСТВИЯ

Информационные системы (ИС) участников взаимодействия (УВ) обмениваются сведениями через СМЭВ для предоставления гражданам услуг и сервисов

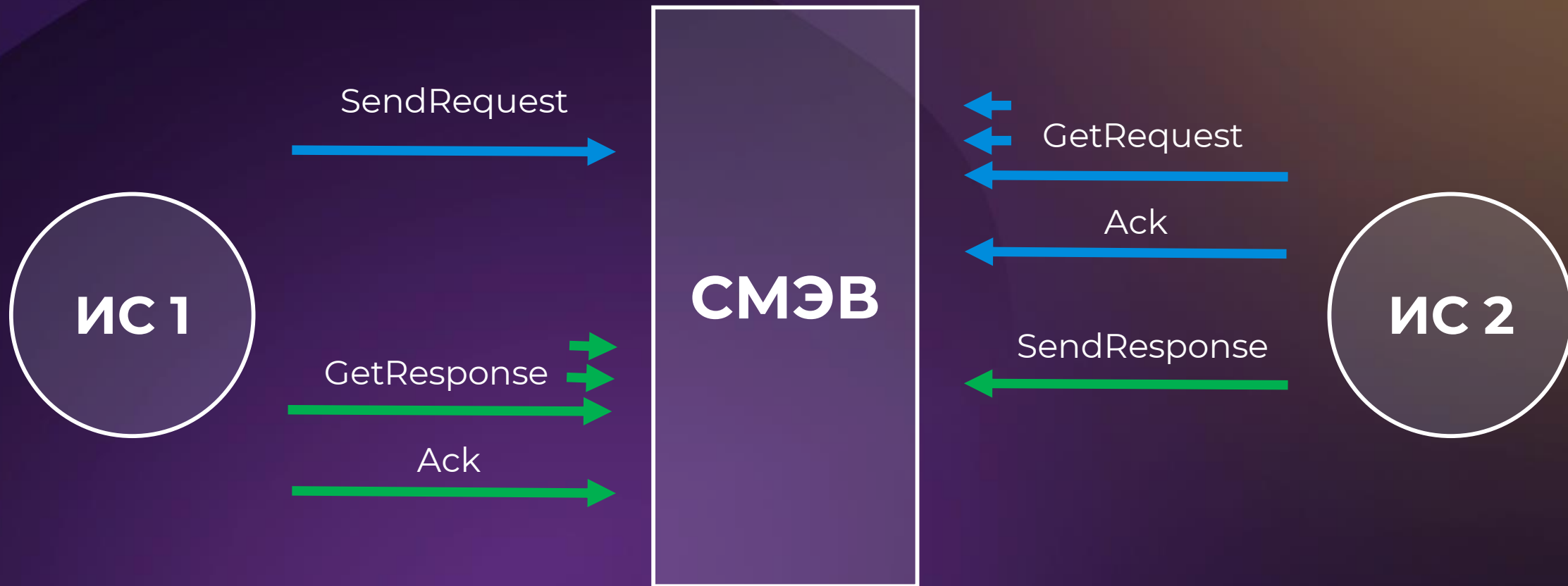
- ❖ Федеральные органы власти
- ❖ Региональные органы власти
- ❖ Многофункциональные центры (МФЦ)
- ❖ Кредитные организации
- ❖ Страховые компании
- ❖ Операторы сотовой связи



# ЗАМЕНА ЗАГРАНПАСПОРТА



# ПРИНЦИП ОБМЕНА



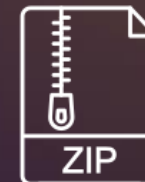
# КАКИМИ ДАННЫМИ ОБМЕНИВАЮТСЯ УЧАСТНИКИ ВЗАИМОДЕЙСТВИЯ



**Структурированные  
сведения**



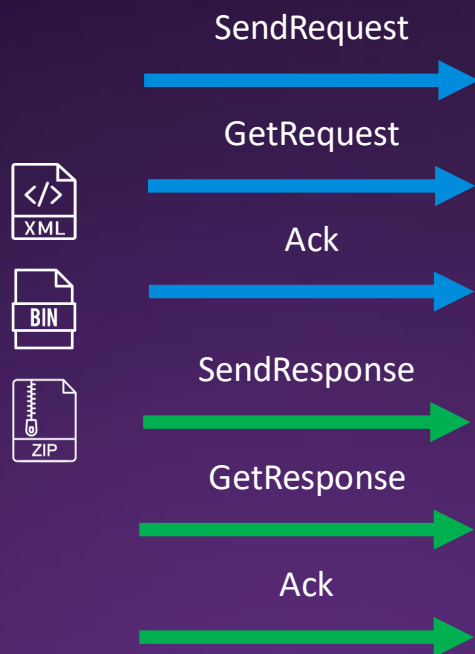
**MTOM вложения  
до 20 МБ**



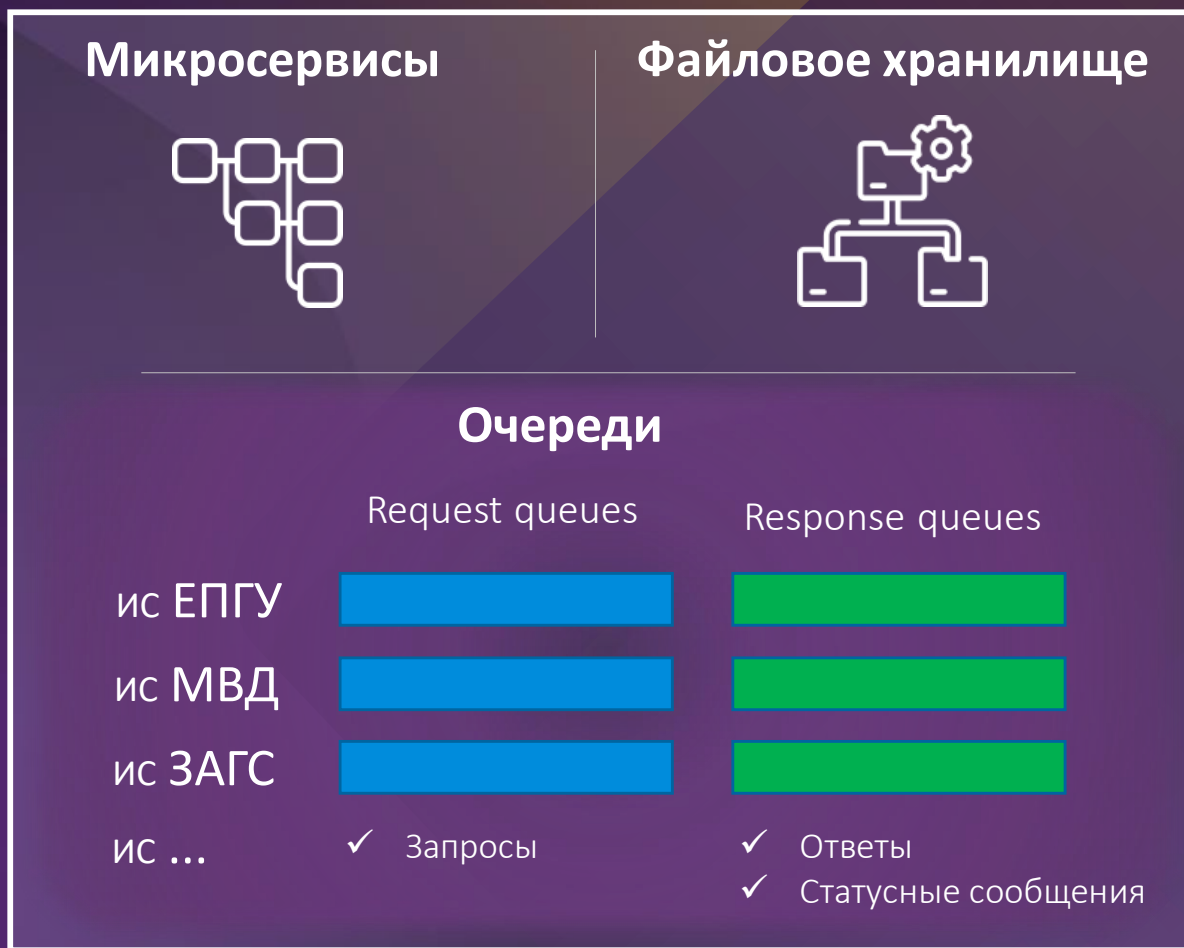
**FTP вложения  
до 1 ГБ**

# ВНУТРИ СМЭВ

1. Идентификация
2. Форматно-логический контроль
3. Управление файлами
4. Маршрутизация
5. Завершающая обработка



## СМЭВ



## НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

- ❖ Скорость обмена не менее 200 000 RPS
- ❖ Объем трафика не менее 500 МВ/сек
- ❖ Высокая отказоустойчивость
- ❖ Доступность не менее 99.99%
- ❖ Масштабируемость

## ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

- ✓ Гибкое администрирование настроек обменов
- ✓ Гибкая фильтрация сообщений
- ✓ Гарантированная доставка сообщений
- ✓ Регулирование трафика «на лету»

## НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

- ❖ Скорость обмена не менее 200 000 RPS
- ❖ Объем трафика не менее 500 МВ/сек
- ❖ Высокая отказоустойчивость
- ❖ Доступность не менее 99.99%
- ❖ Масштабируемость

## ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

- ✓ Гибкое администрирование настроек обменов
- ✓ Гибкая фильтрация сообщений
- ✓ Гарантированная доставка сообщений
- ✓ Регулирование трафика «на лету»

Как оптимизировать  
работу с очередями



01

Почти классическая  
задача



02

Сила и слабость  
PostgreSQL



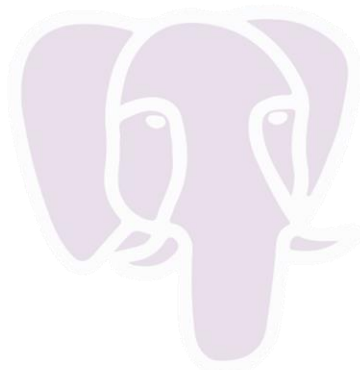
03

Идея реализации  
бизнес логики



04

Немного  
статистики



# PostgreSQL

выбор в отсутствии выбора

# Tarantool

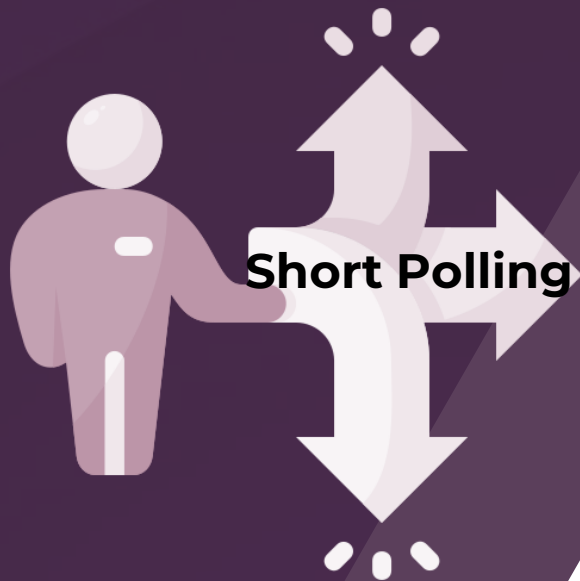


# КЛАССИЧЕСКАЯ ЗАДАЧА

асинхронный обмен и надежная доставка



# ТРЕБОВАНИЯ К ВЫБОРУ ИЗ ОЧЕРЕДЕЙ



Клиент сам решает как вычитывать сообщения



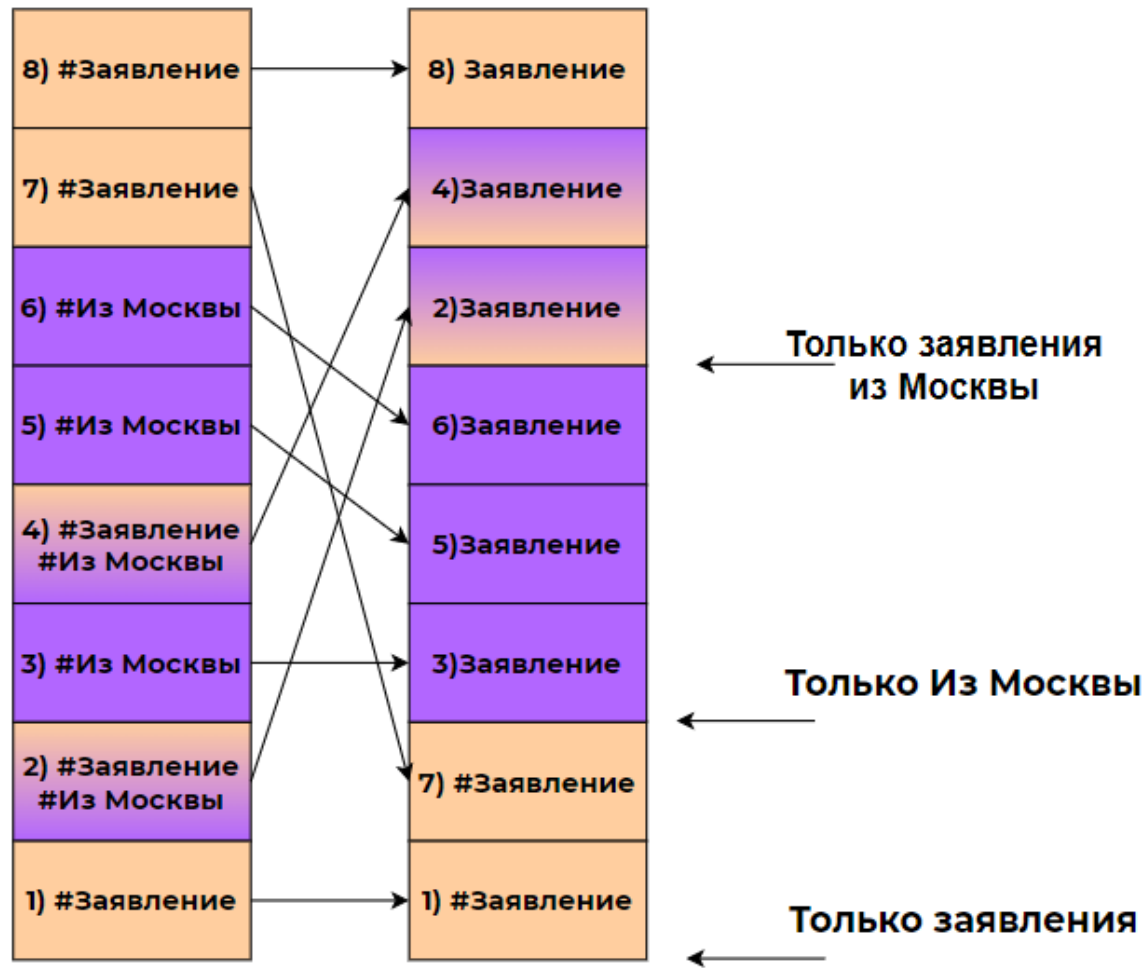
Существуют клиенты, которые интенсивно пишут и читают из одной очереди с низкой кардинальностью



с какой скоростью и в какие моменты

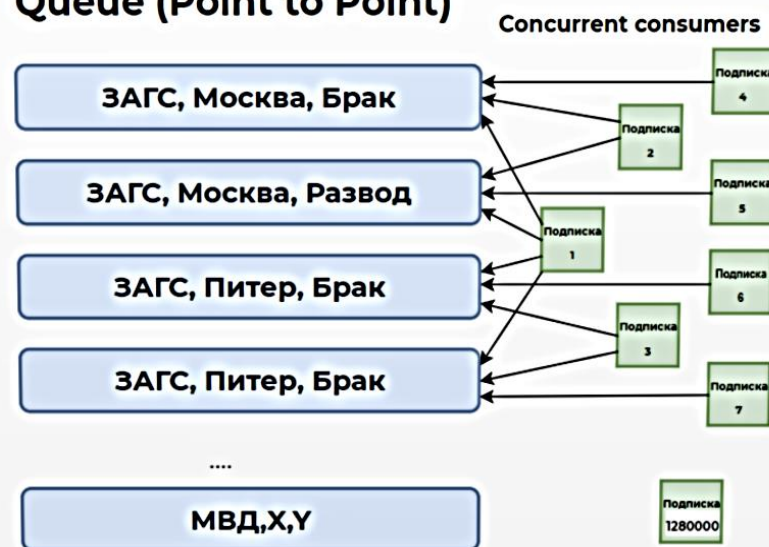


с какой очередностью

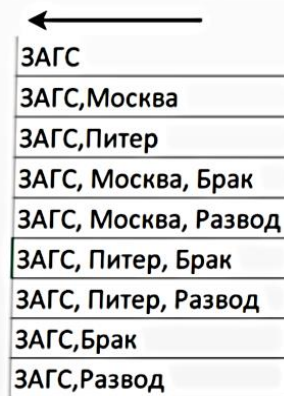


# НАИВНАЯ РЕАЛИЗАЦИЯ ЧЕРЕЗ КЛАССИЧЕСКИЕ ОЧЕРЕДИ

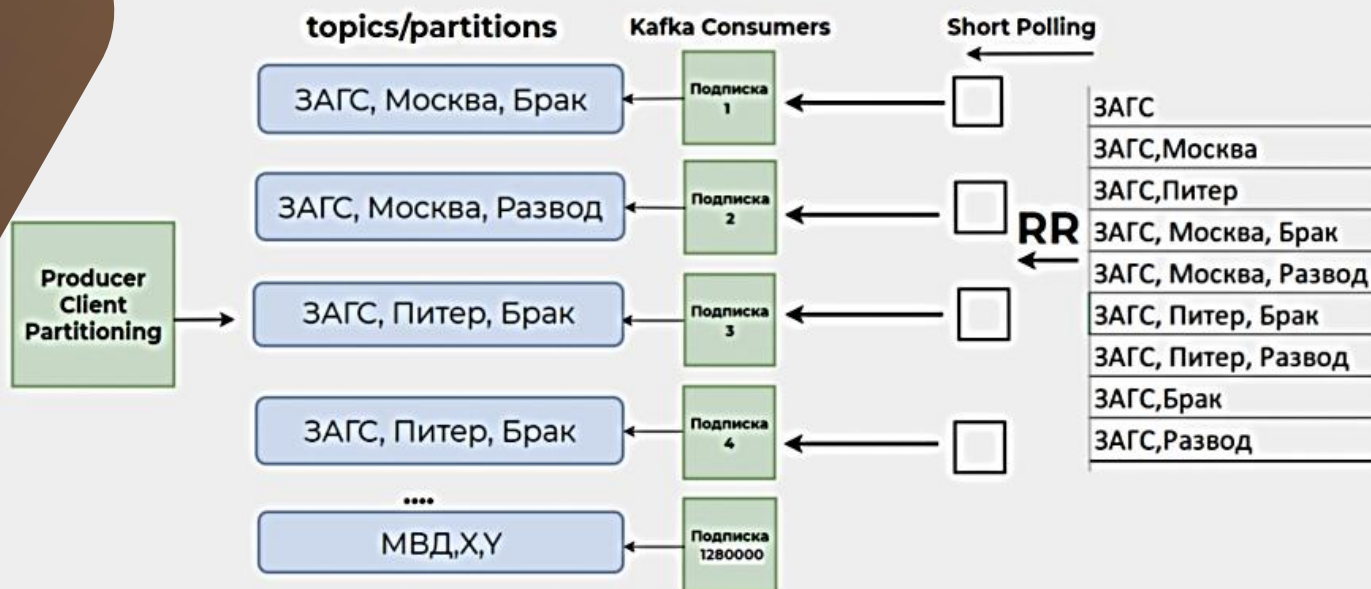
## Queue (Point to Point)



## Short Polling



# ЧЕРЕЗ КАФКА ВСЕ ЕЩЕ ОЧЕНЬ МНОГО ПРОБЛЕМ



# KAFKA STREAMS

Не получилось разогнать



1) Две системы: Журнал + БД

2) Если нам нужна только БД, то зачем Журнал?



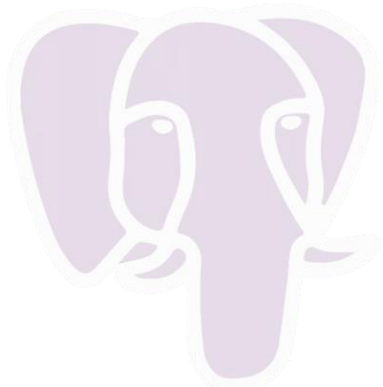
Задачу не удалось решить классическими брокерами/очередями



# PostgreSQL для очередей

Почему бы и нет?

Почему  
выбрали  
PostgreSQL?



База данных подходит для хранения состояния и работы с фильтрами



PostgreSQL - производительная БД



SQL/ACID – Стандарт рынка, все знают, все любят

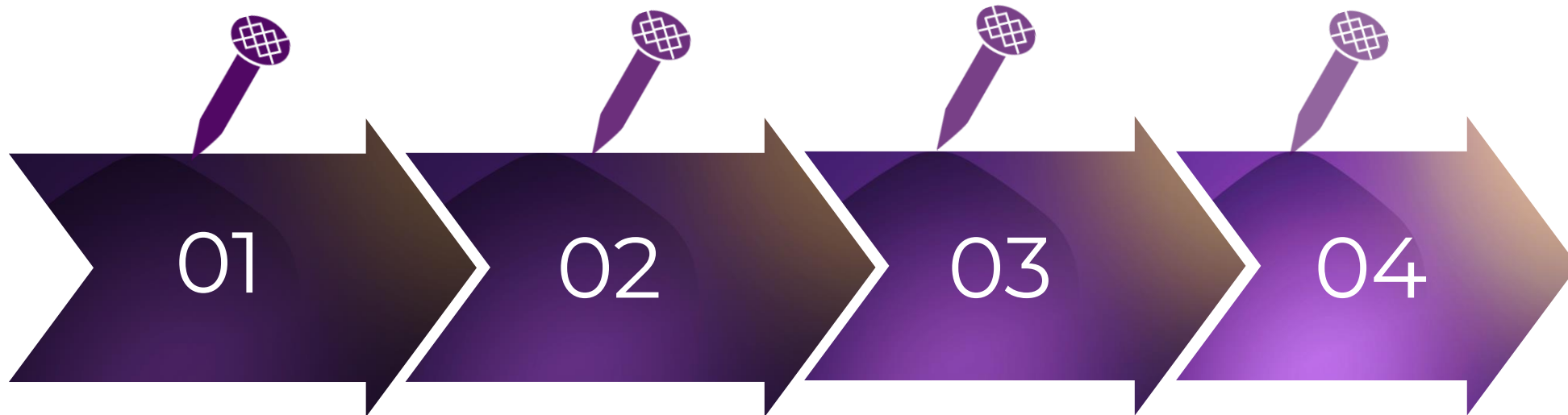


Keep Simple!

# СХЕМАТИЧНАЯ РЕАЛИЗАЦИЯ РАБОТЫ



# Гвозди в “крышку” PostgreSQL



Большое количество индексов



PostgreSQL периодически промахивается в нужные индексы

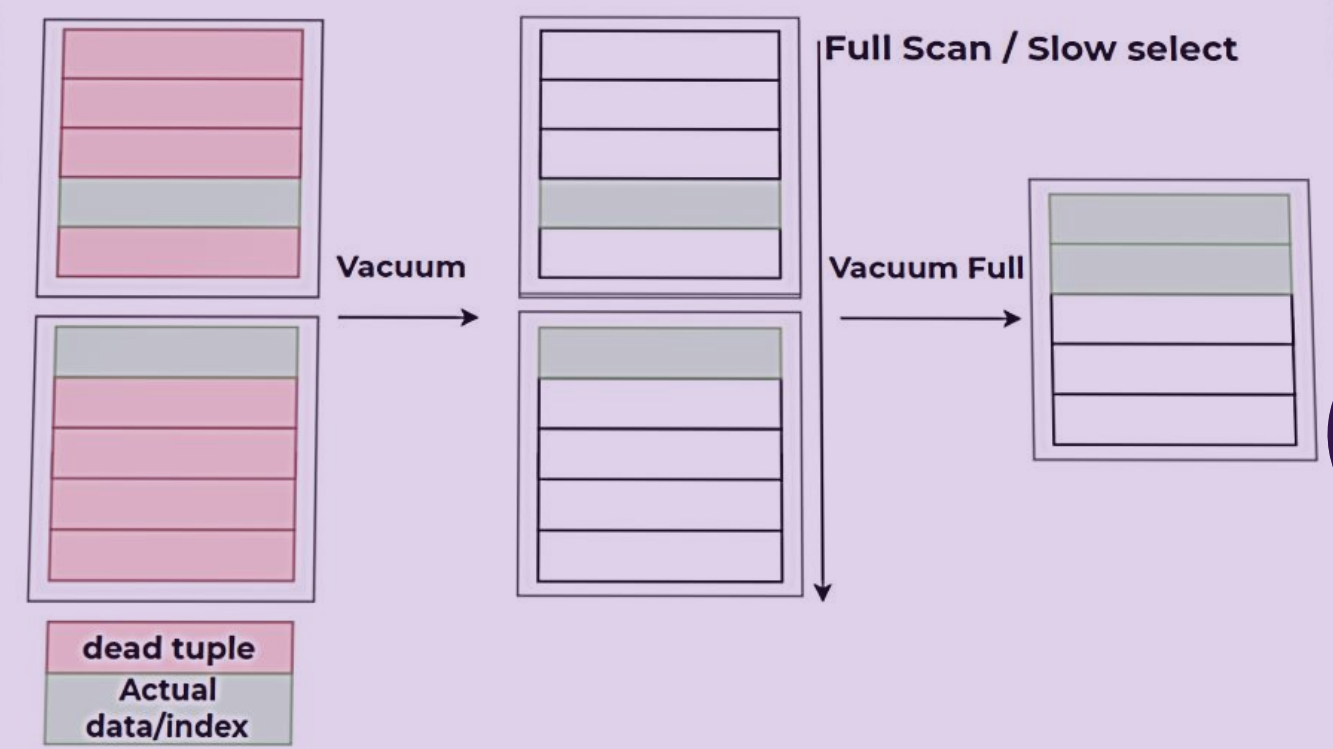


Vacuum



Медленная синхронная репликация в режиме Active-Active

# ВЛИЯНИЕ VACUUM



# Что хотим от очереди?

1

Очередь должна быть быстрая



2

Быстрое решение задачи синхронной репликации



3

Горизонтальное масштабирование пропускной способности



4

Быстрый поиск по фильтрам



# ПРОБУЕМ ТАРАНТУЛ

Это  
База  
данных

Базовые  
функции

Данные  
надежно  
хранятся  
на диске

Можно  
делать  
запросы



# Tarantool – это База Данных

Стандартные  
операции

put



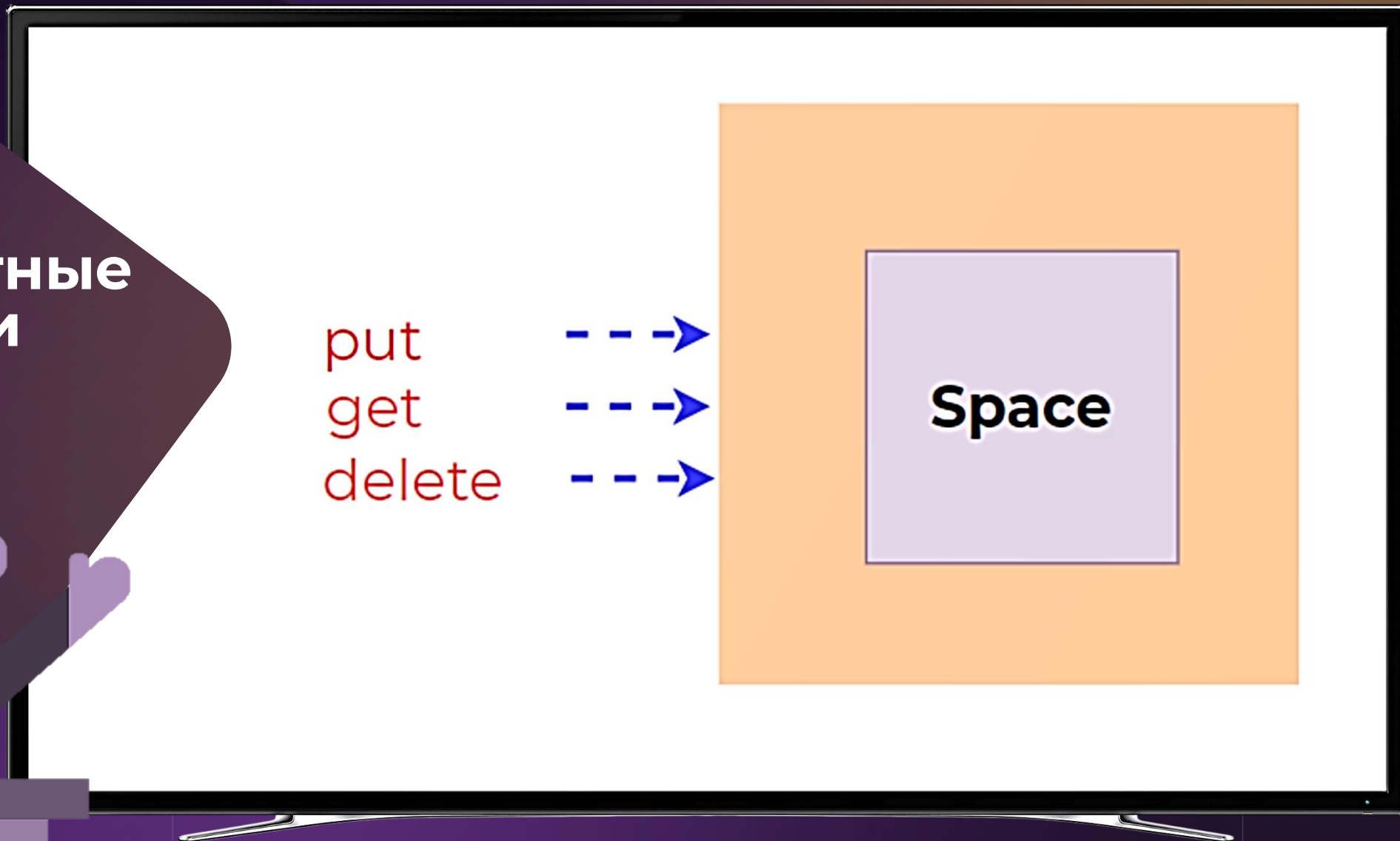
get



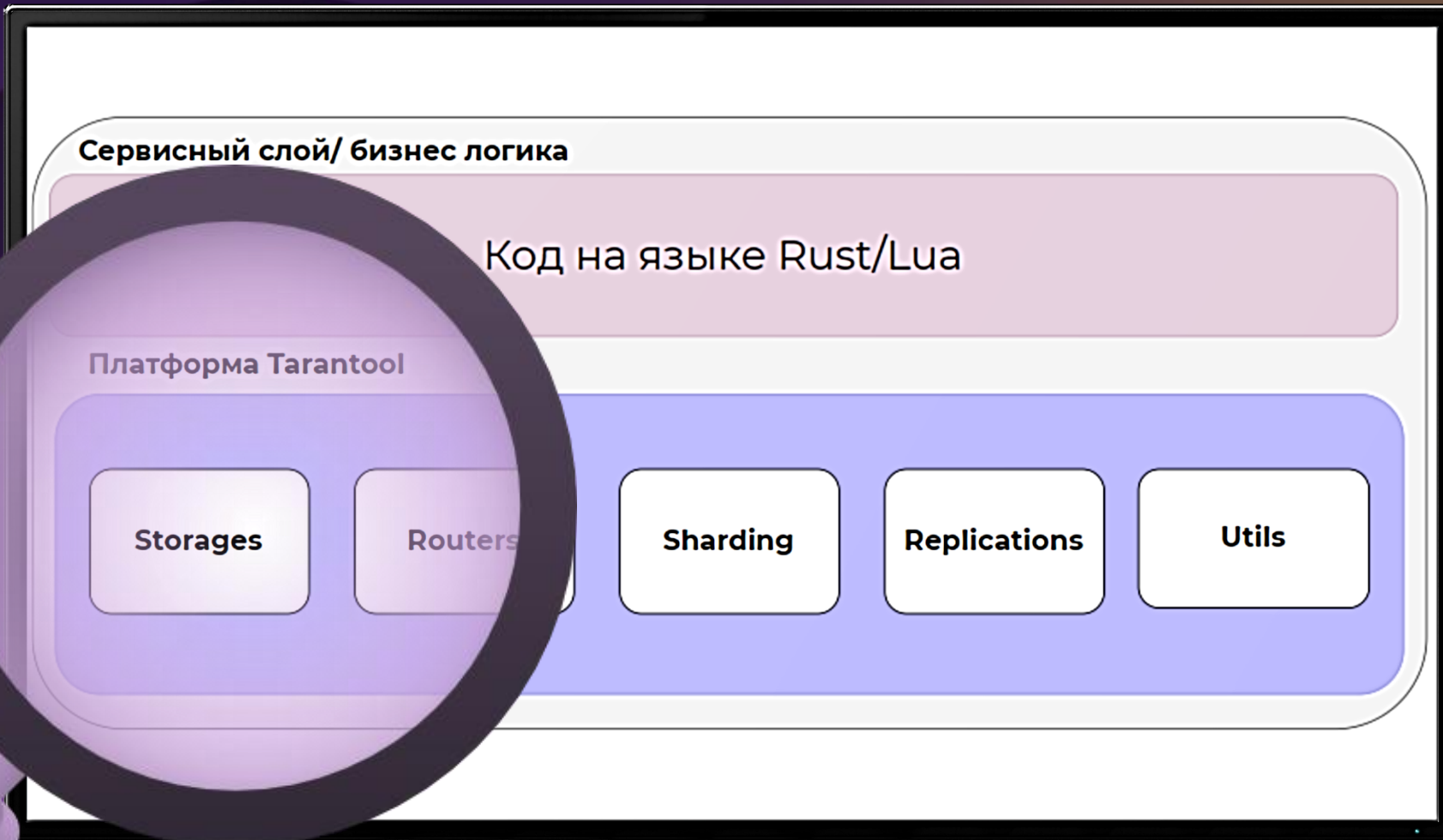
delete



Space



# Tarantool – это сервер приложений



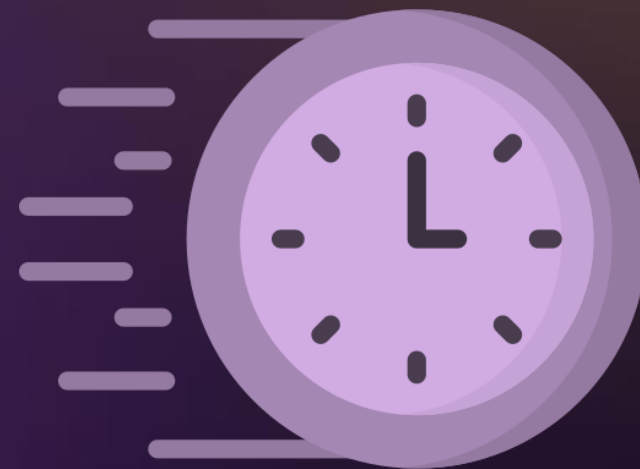


# ПРЕИМУЩЕСТВА

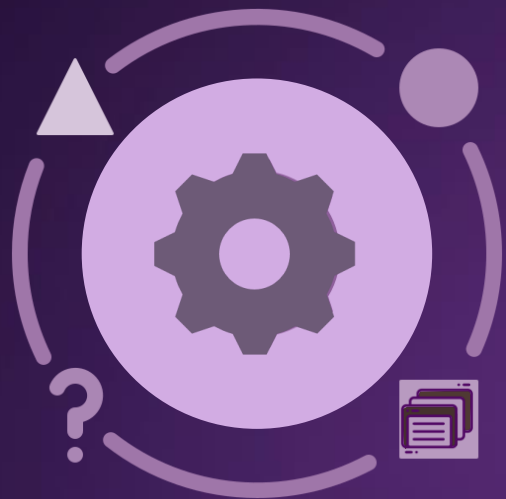
Не требуется  
Вакуум



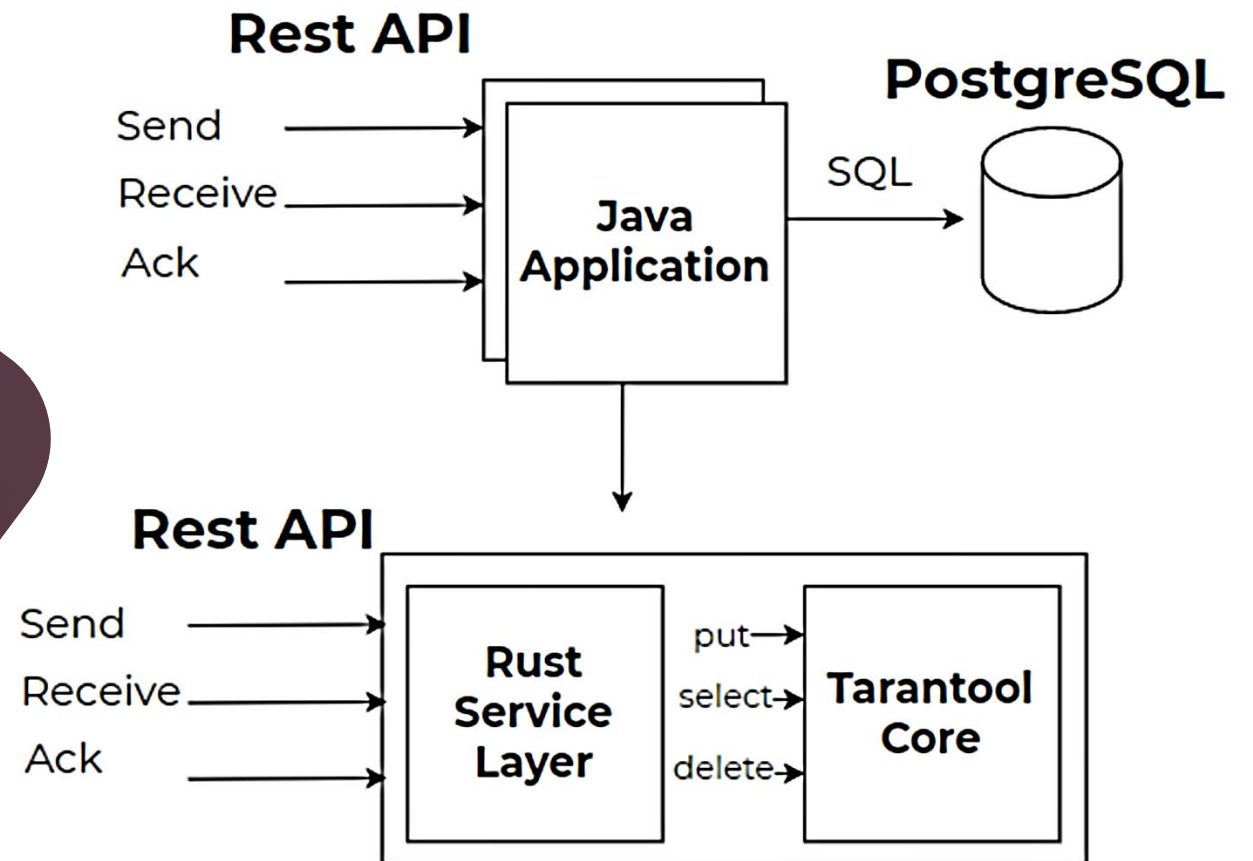
Задизайнен  
Как In Memory



Нет  
Планировщика  
запросов



# Пишем очередь на Tarantool



# Обязательное условие

Имя	Женя
Город	Москва
Место работы	RtLabs
messageId	jpg123

**tarantool.put**

`hashcode(message_id)`

**Storage**

Space

Space

**Storage**

Space

Space

Положили в заданный storage

шаг 1

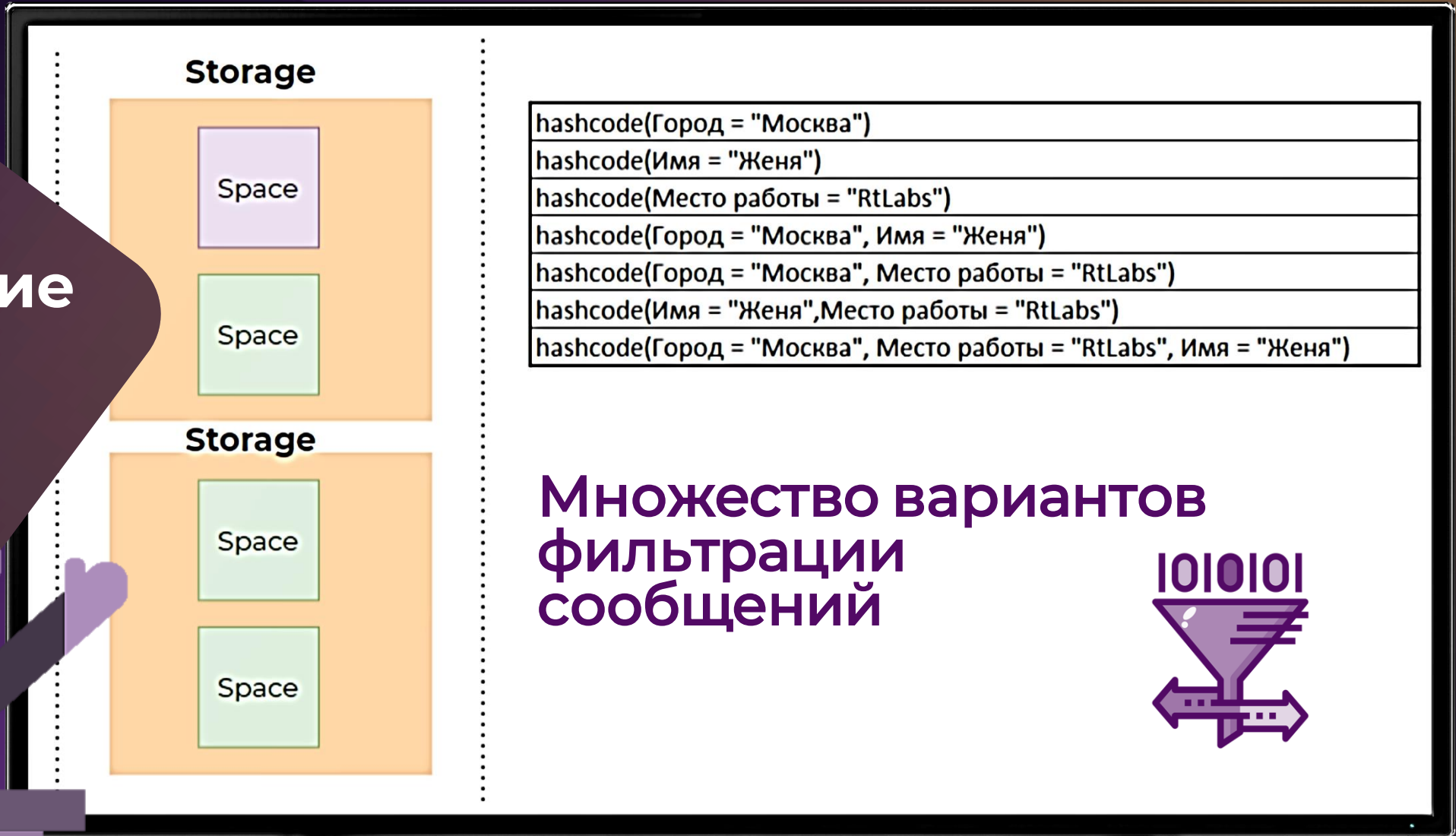
Распределенность  
и шардирование



# В каком Storage искать?

шаг 2

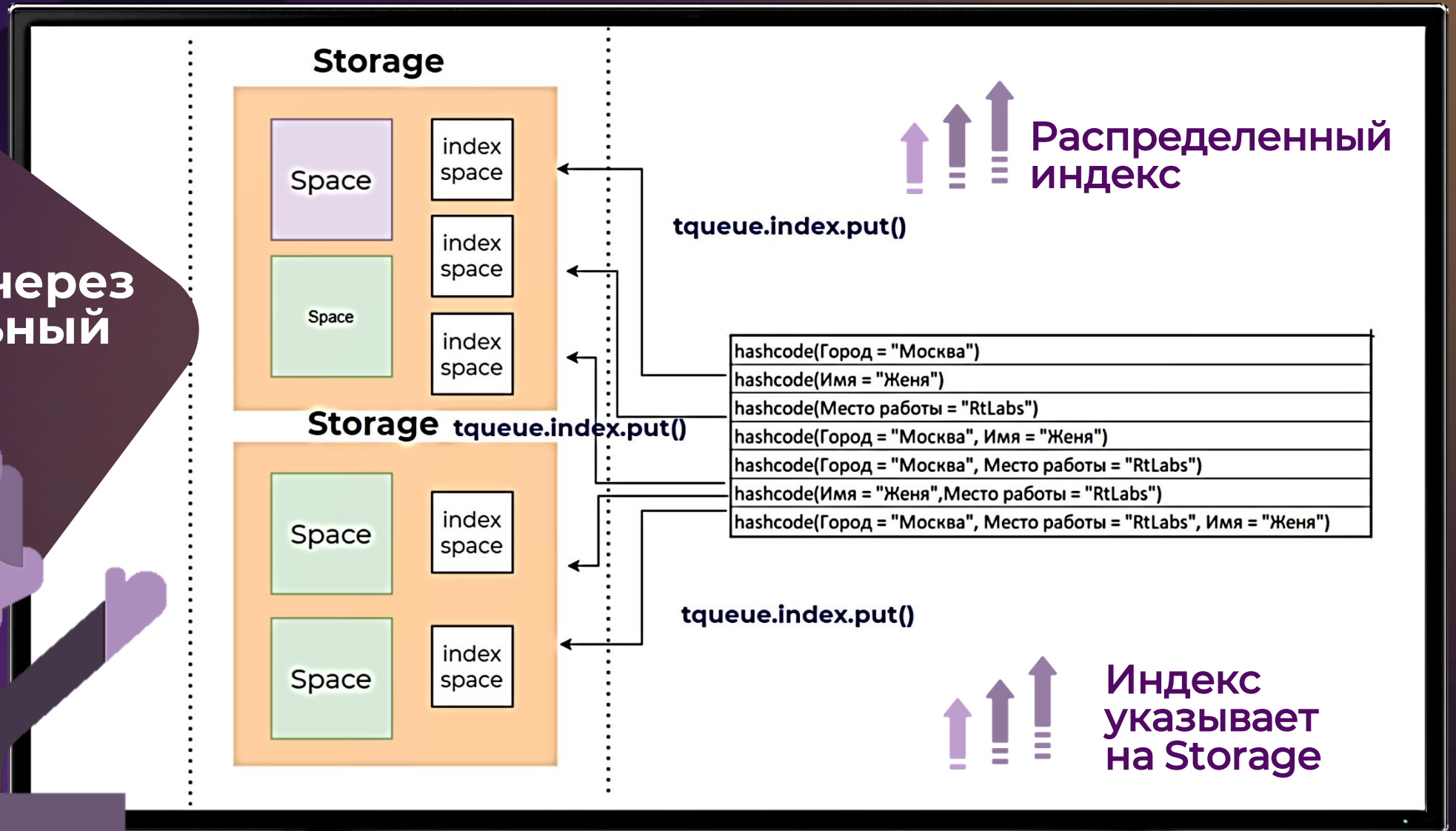
Вычитывание  
из очереди



# Вычитывание из очереди

шаг 3

Реализация через  
дополнительный  
индекс



# Вычитывание из очереди

шаг 4

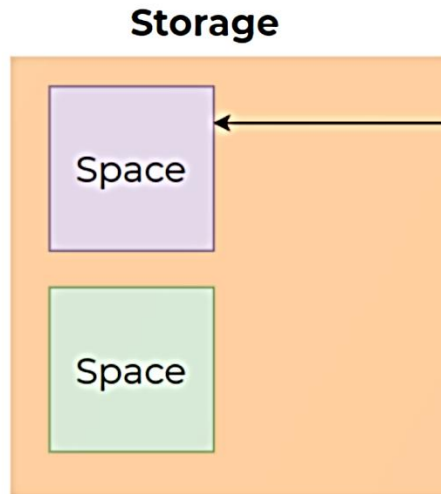
Поиск по индексу



# Удаление данных

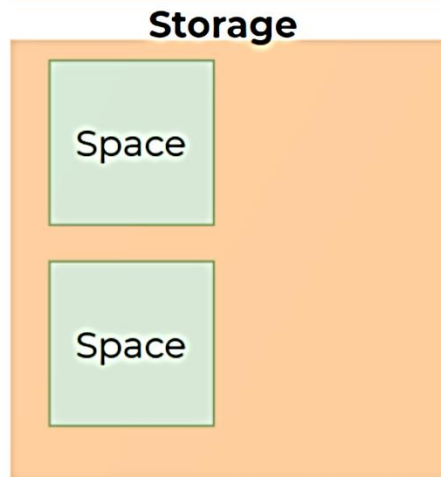
шаг 5

Подтверждение  
вычитки  
сообщения



1) Hashcode(messageId)

2) `tarantool.delete`



Сразу определен Storage



Данные удаляются по ключу



Реализовали  
коробочное  
решение –  
очередь под  
требования

Запись и чтение за  
константное время  
( не зависит от кол-  
ва реплик)



## Итого получили

Шардирование и  
равномерное  
распределение  
данных

Решение оказалось в 7  
раз быстрее  
предыдущего



# Бенчмарки

## СМЭВ v1 на PostgreSQL

Тип нагрузки	CPU, ядра	Latency 50, ms	Latency 90, ms	Latency 99, ms
1000000 RPS в N очередей	4760	30	109	320

## СМЭВ v2 на Tarantool

Тип нагрузки	CPU, ядра	Latency 50, ms	Latency 90, ms	Latency 99, ms
1000000 RPS в N очередей	770	20	44	130





## Мы уже эксплуатируем наше решение

В СМЭВ нужные данные бережно и быстро распределяются и вычитаются из Tarantool **Storages**, чтобы доставить их ведомствам



## Решение на Tarantool – если снижение CPU приоритетнее объемов RAM

Tarantool получает выигрыш в сокращении количества используемого CPU за счет увеличения memory footprint



## Решение на Tarantool позволит получить СУБД+Application Server «в одном флаконе»

Если в вашем скоростном решении нужно накрутить логику, но вы хотите оставить БД, то Tarantool позволяет надстраивать логику прямо на БД. На выходе - просто rest-ы.

# Выводы



РТЛабс

# Мы подготовили Демо для «пробы пера»



Это небольшое приложение на Tarantool, которое дает минимизированное CRUD API по HTTP



# Если остались вопросы

**От PostgreSQL  
к Tarantool**  
Выбор в отсутствие выбора

Вайнштейн Любовь

РТЛабс | СМЭВ

Email: [lubov.vaynshteyn@rtlabs.ru](mailto:lubov.vaynshteyn@rtlabs.ru)

Демкин Евгений

РТЛабс | СМЭВ

Email: [evgeniy.demkin@rtlabs.ru](mailto:evgeniy.demkin@rtlabs.ru)

Кольцов Дмитрий

Пикодата | Tarantool

Email: [dkoltsov@picodata.io](mailto:dkoltsov@picodata.io)