

PGConf.Сибирь 2023

PosgresPro

# Как мы делаем встроенную отказоустойчивость в Postgres



## **Рутман Михаил Валерьевич**

- Работаю ведущим разработчиком в Postgres Pro
- Более 20 лет в IT (Сигнатек, Eyeline, Navitel, Parallels, Plesk, Kublr)
- Преподаю в Новосибирском гос. университете

- Доступность
- Безотказность
- Ремонтпригодность
- ...

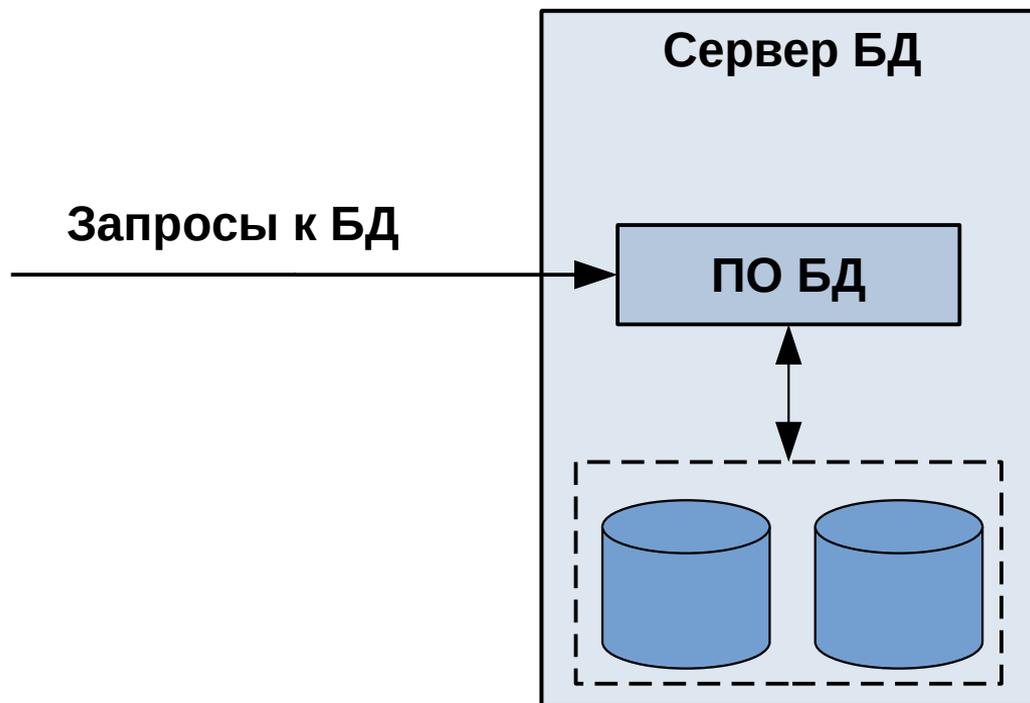
PGConf.Сибирь 2023

Отказоустойчивость

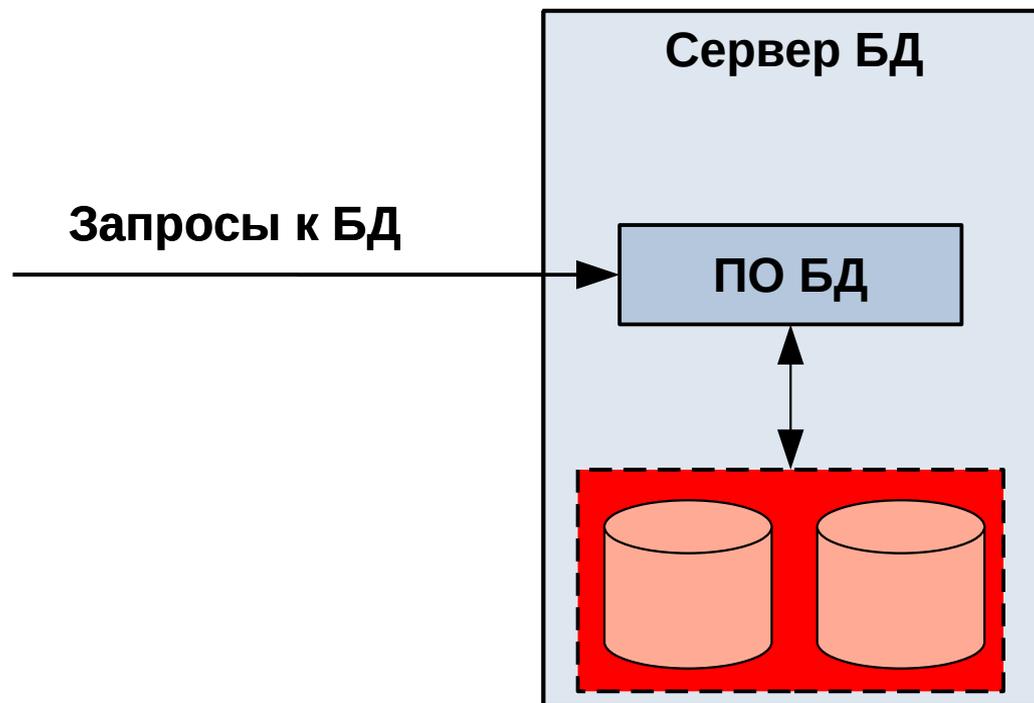
PostgresPro

# Способность продолжить работу при наличии отказов

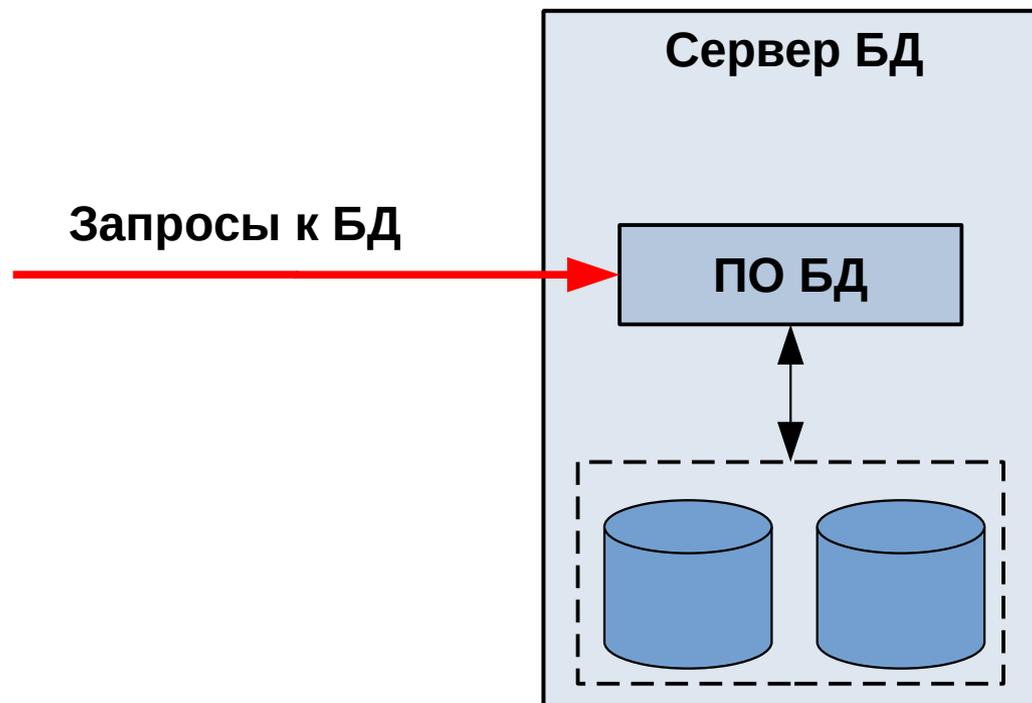
# Отказоустойчивость Баз Данных



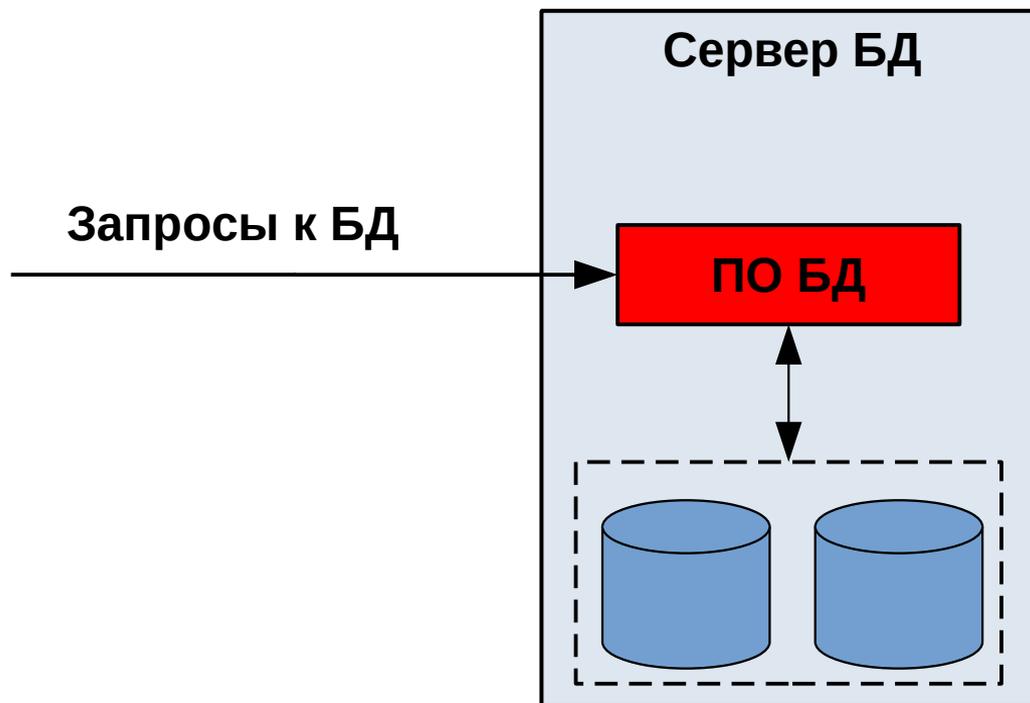
# Отказоустойчивость Баз Данных



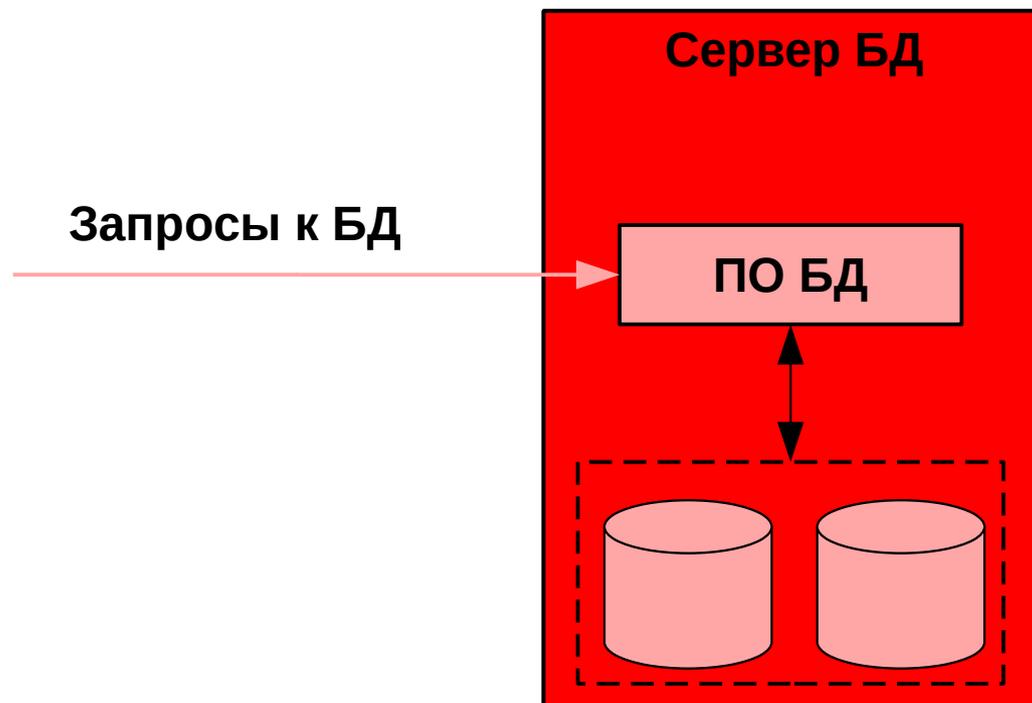
# Отказоустойчивость Баз Данных



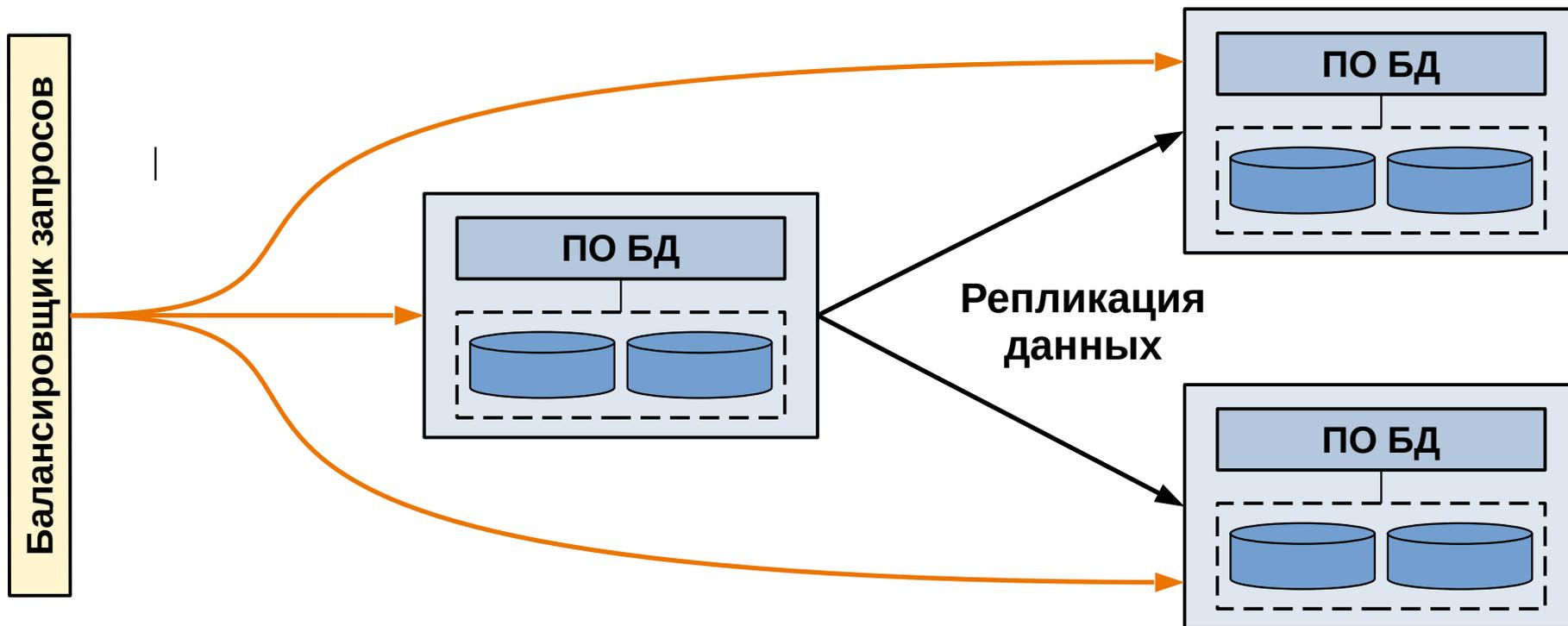
# Отказоустойчивость Баз Данных



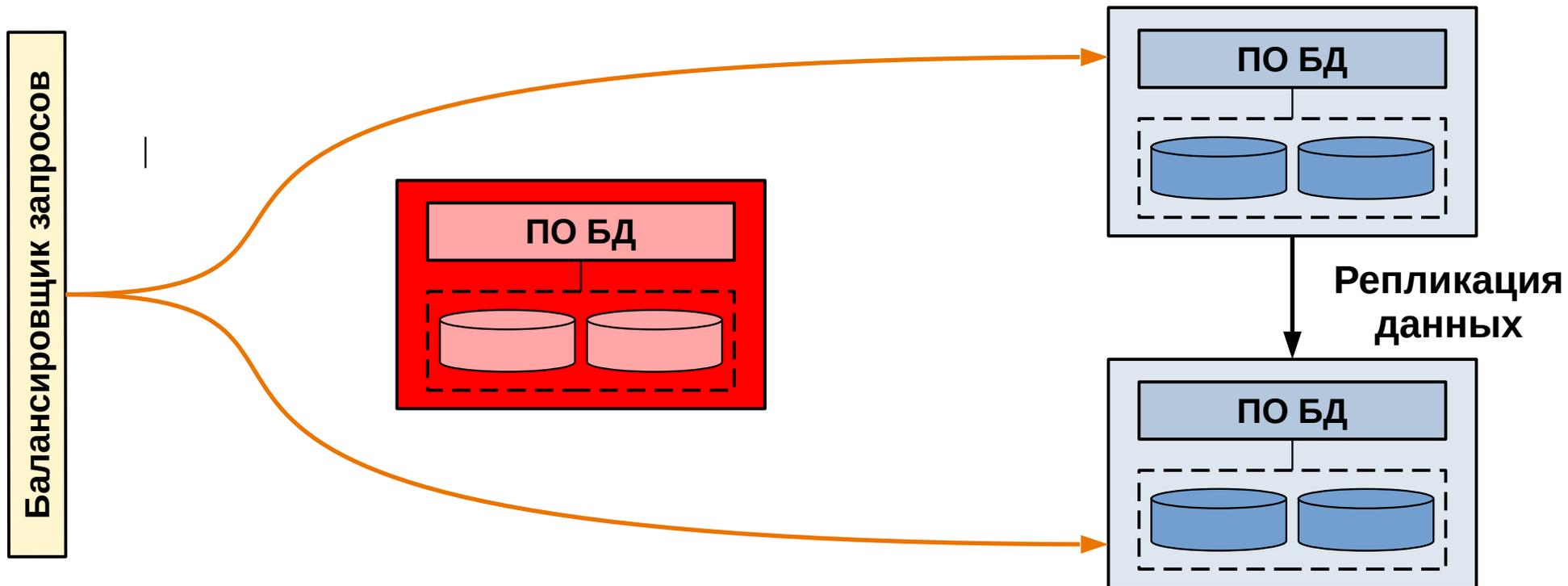
# Отказоустойчивость Баз Данных



# Отказоустойчивость Баз Данных



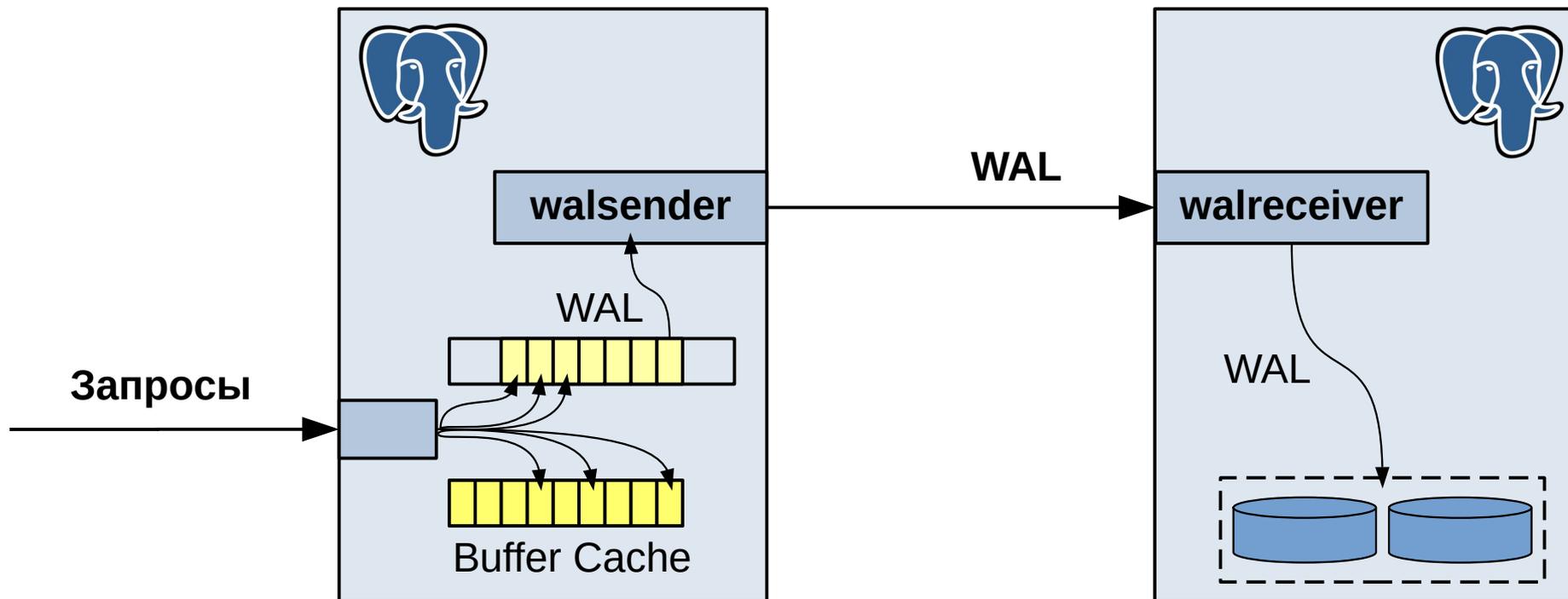
# Отказоустойчивость Баз Данных



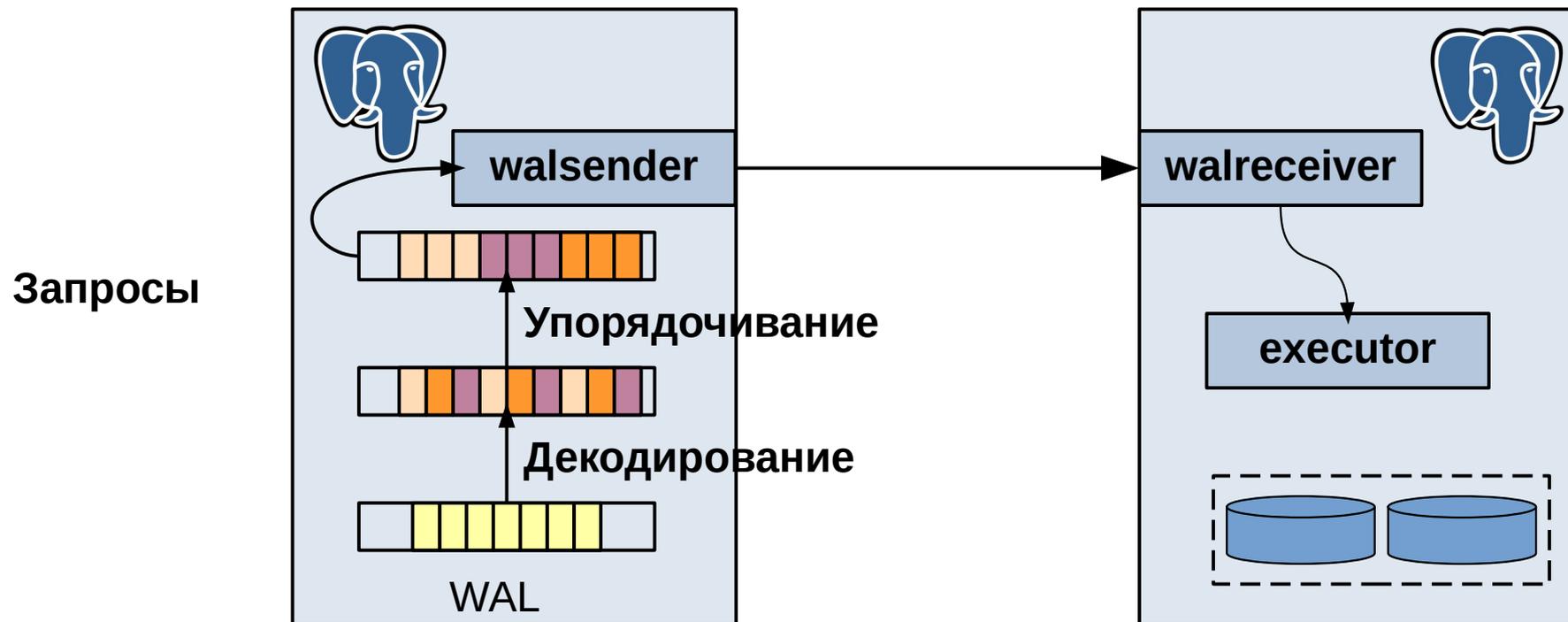
# Механизмы отказоустойчивости Postgres

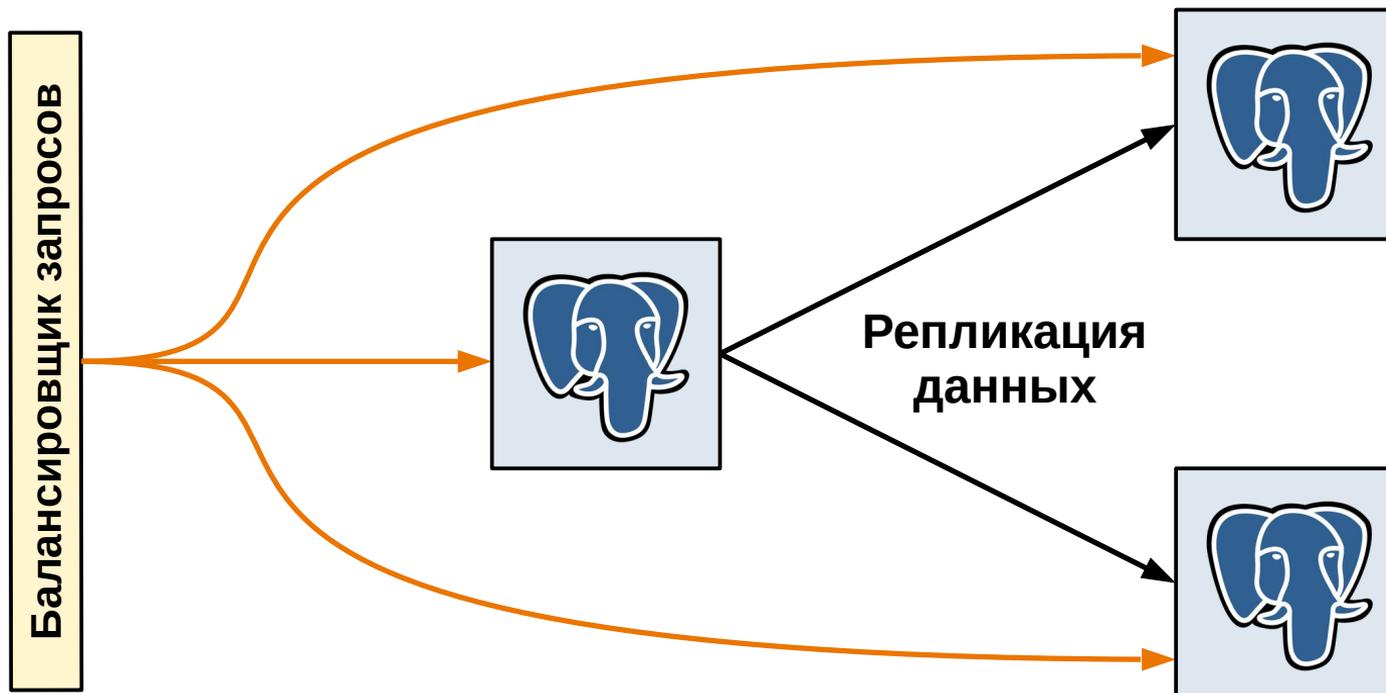
- Физическая репликация
- Логическая репликация

# Физическая репликация

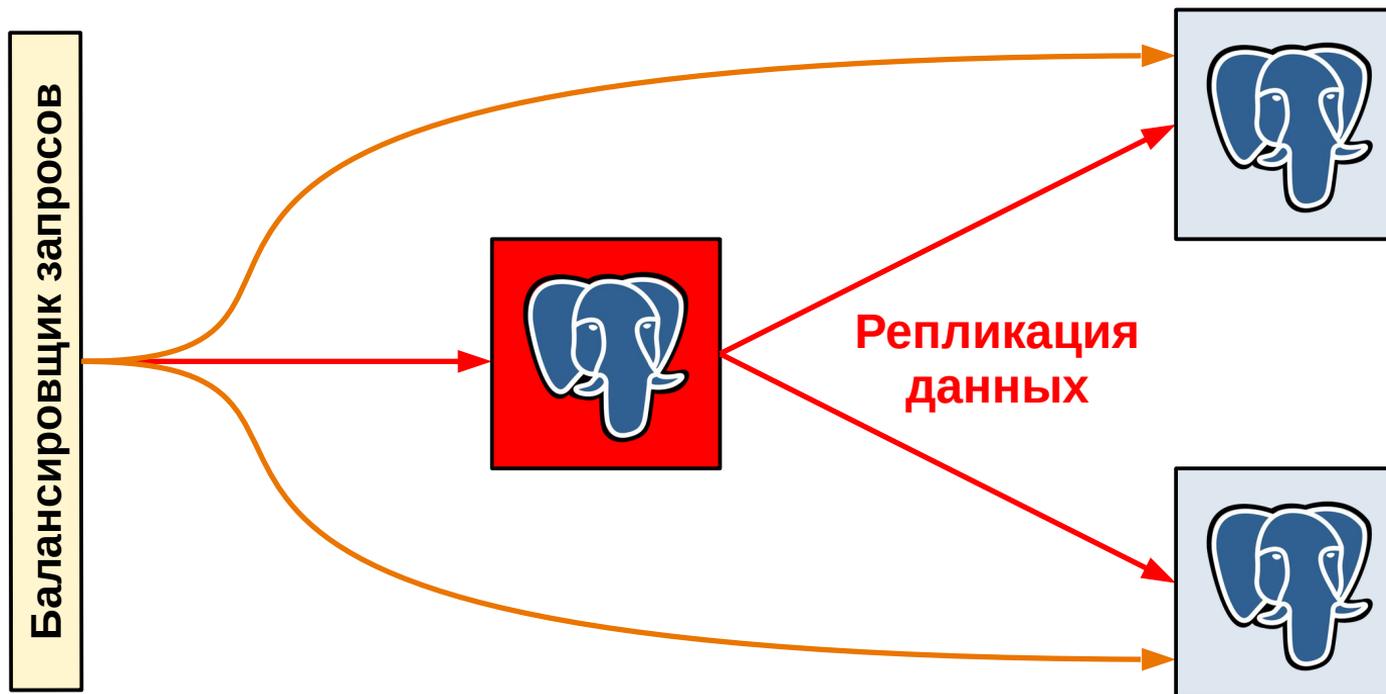


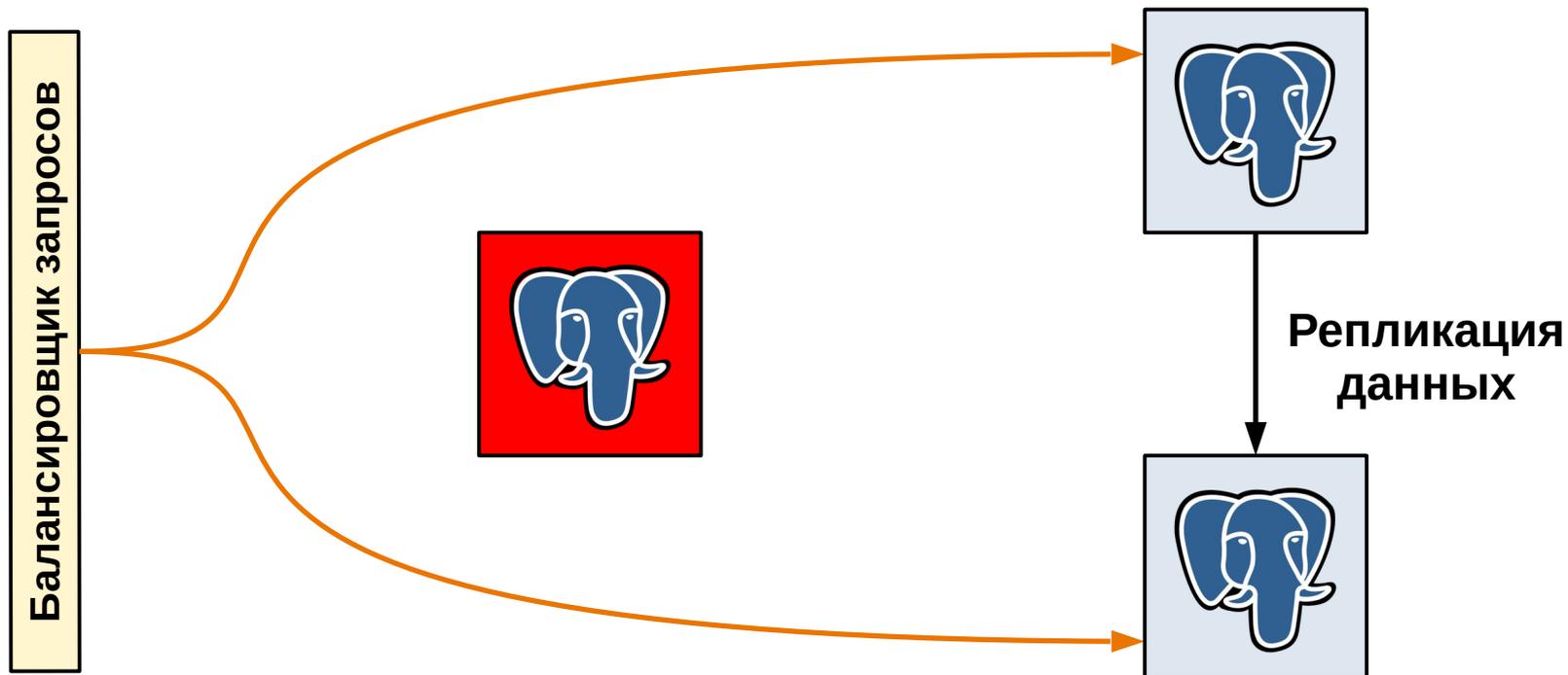
# Логическая репликация



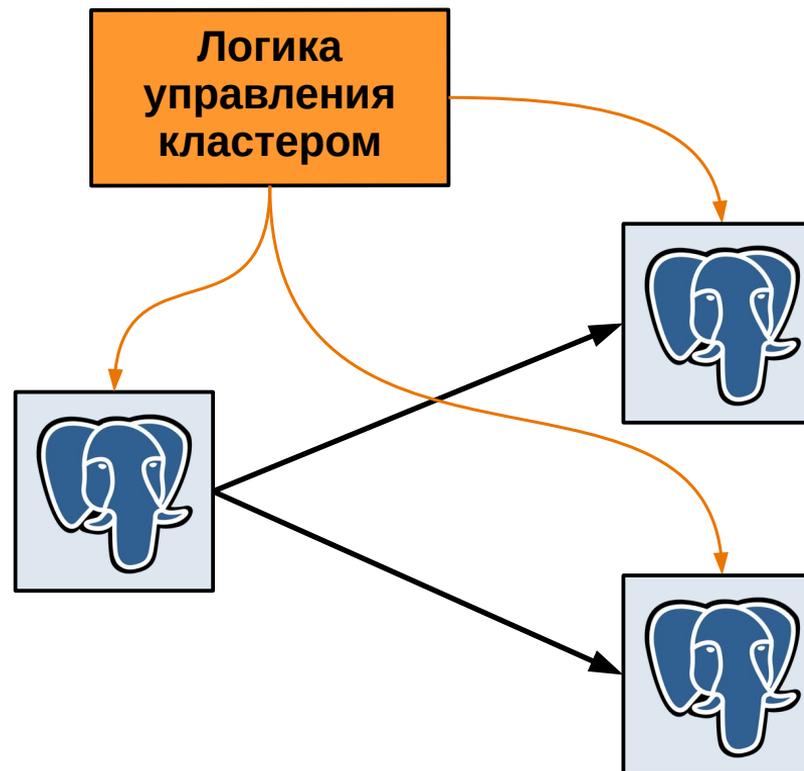


# Нет переключения на резервный Сервер

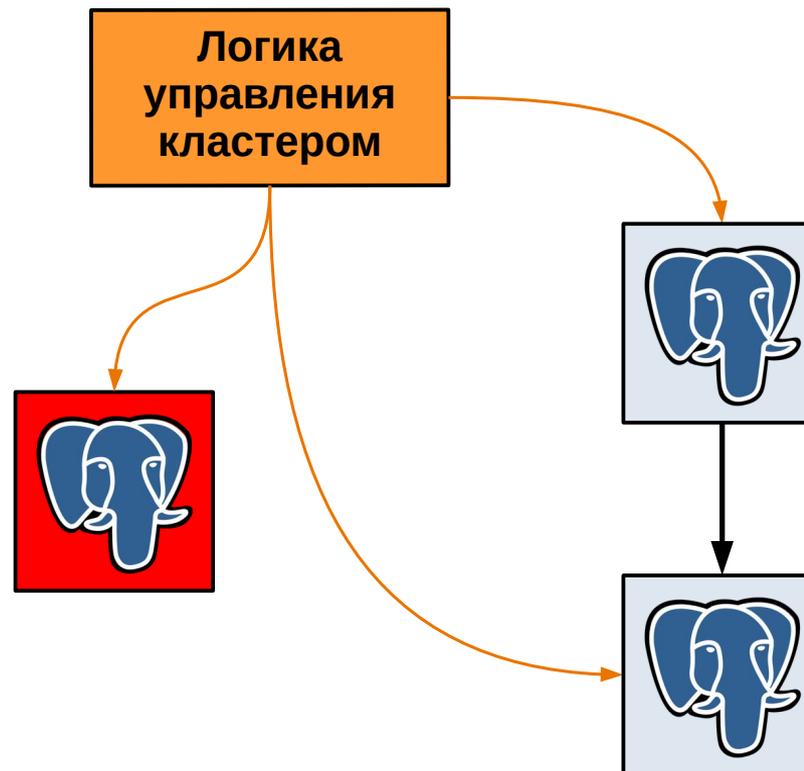




- Stolon от SorintLab
- Patroni от Zalando
- Corosync&Pacemaker от ClusterLabs



- Stolon от SorintLab
- Patroni от Zalando
- Corosync&Pacemaker от ClusterLabs



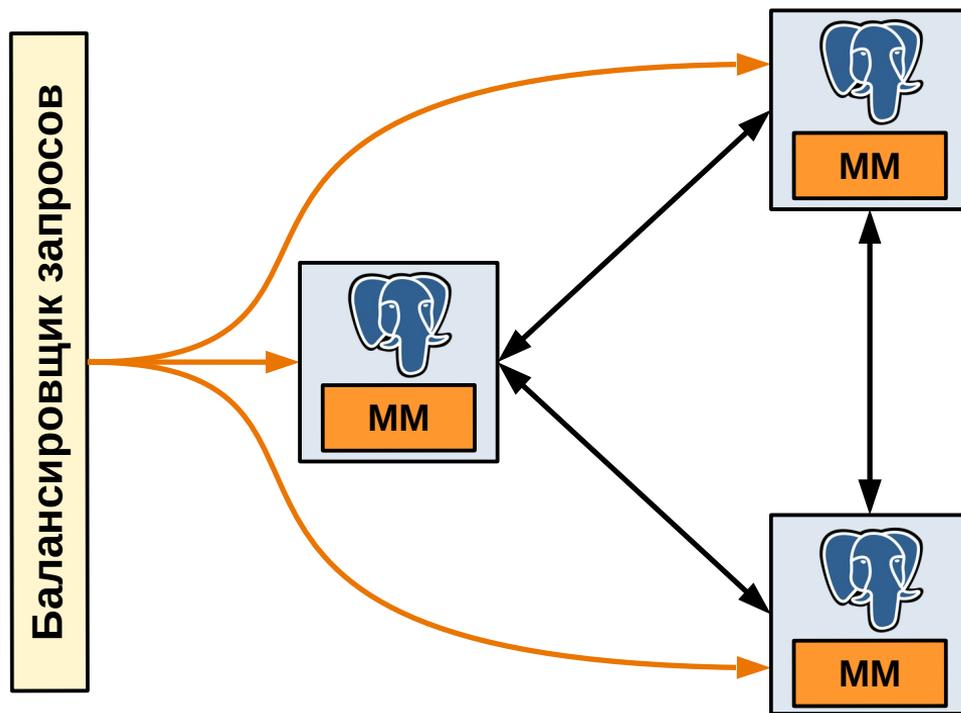
## Что нам не нравится

- Дополнительные администрируемые единицы
- Сложность настройки
- Стороннее ПО
- Ограниченные возможности мониторинга и управления

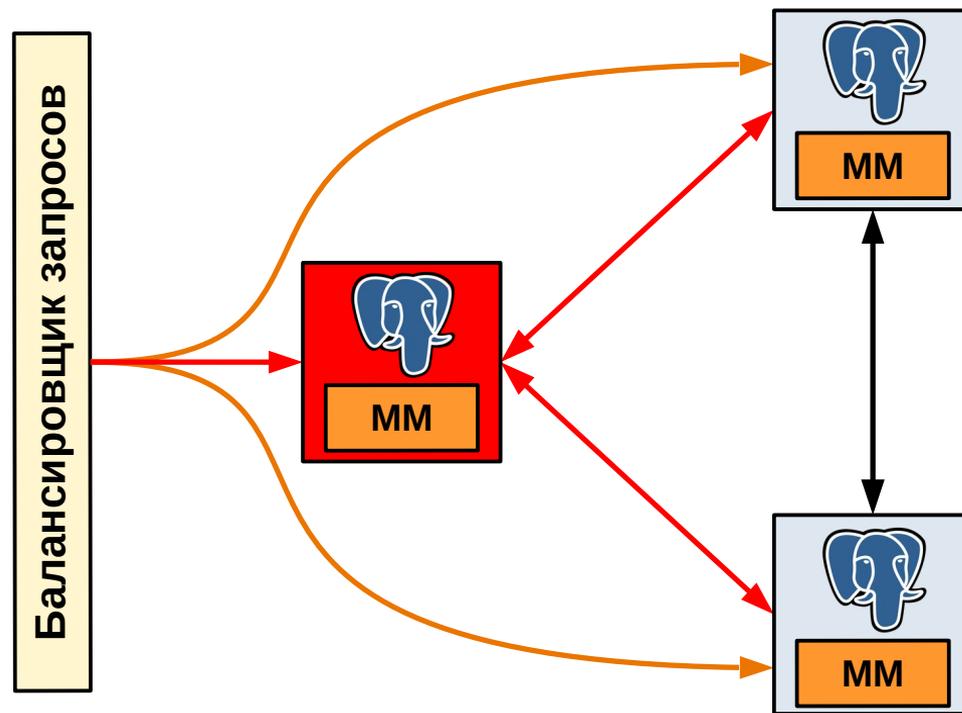
- Мультимастер
- ВіНА



- Симметричный кластер
- Высокая доступность
- Atomic Commit Protocol
- Логическая репликация
- Репликация DDL
- Параллельный Executor



- Симметричный кластер
- Высокая доступность
- Atomic Commit Protocol
- Логическая репликация
- Репликация DDL
- Параллельный Executor



**Производительность уступает решениям  
на основе физической репликации**

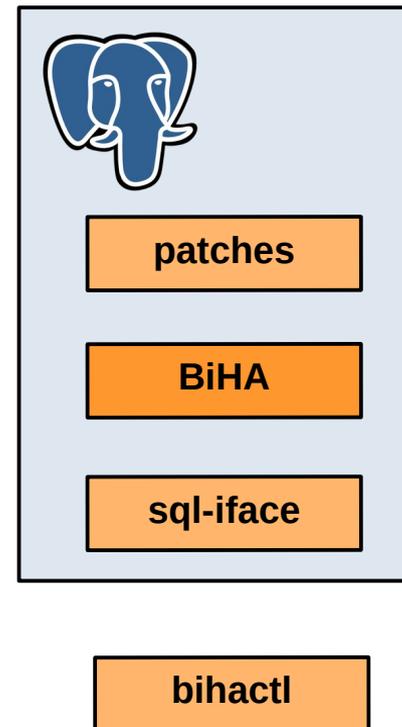
PGConf.Сибирь 2023

Встроенная отказоустойчивость

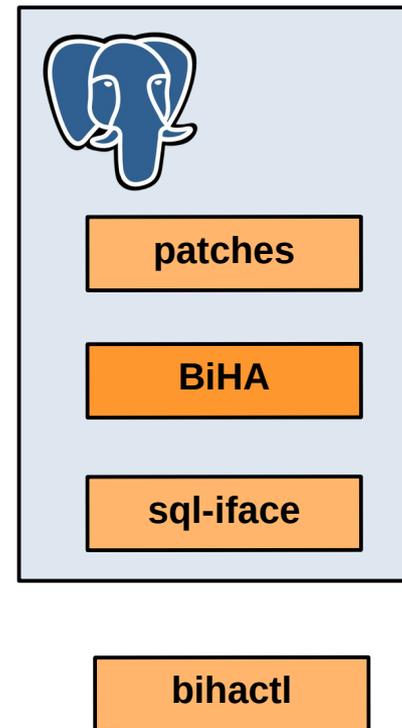
PostgresPro

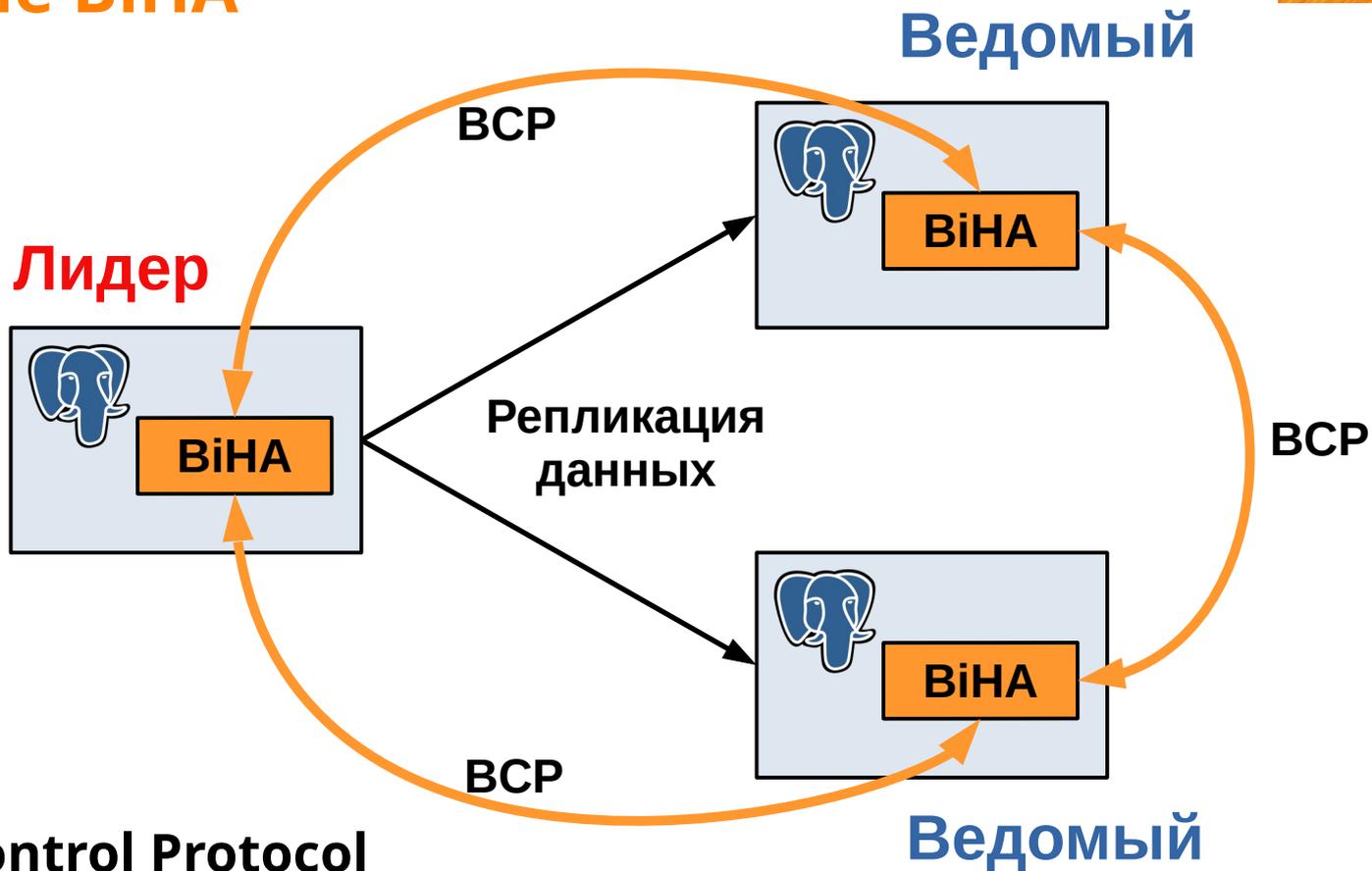
**BiHA — Built-in High Availability**

- Расширение Postgres
- Набор патчей ядра Postgres
- SQL интерфейс
- Утилита bihactl



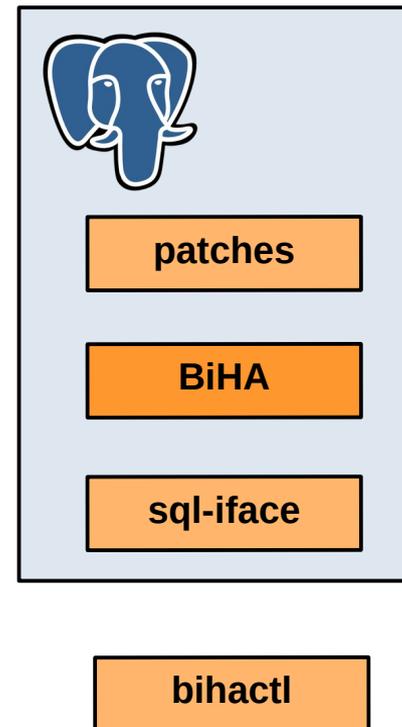
- Контроллирует состояние Узла
- Взаимодействует с Узлами Кластера
- Конфигурирует Кластер
- Обеспечивает работоспособность и доступность Кластера





VCP — ViNA Control Protocol

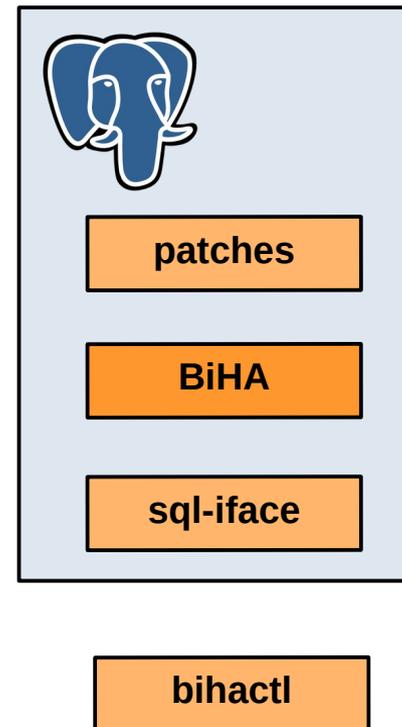
- Настройка режимов работы Postgres
- Ранняя реакция на возникающие проблемы
- Дополнительный мониторинг и сбор недостающей статистики



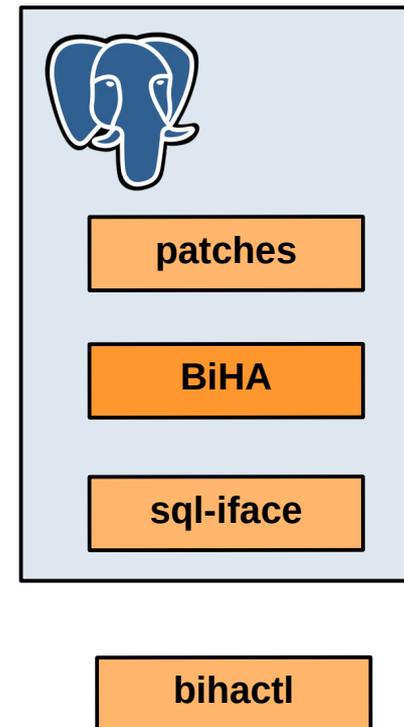
# PGConf.Сибирь 2023

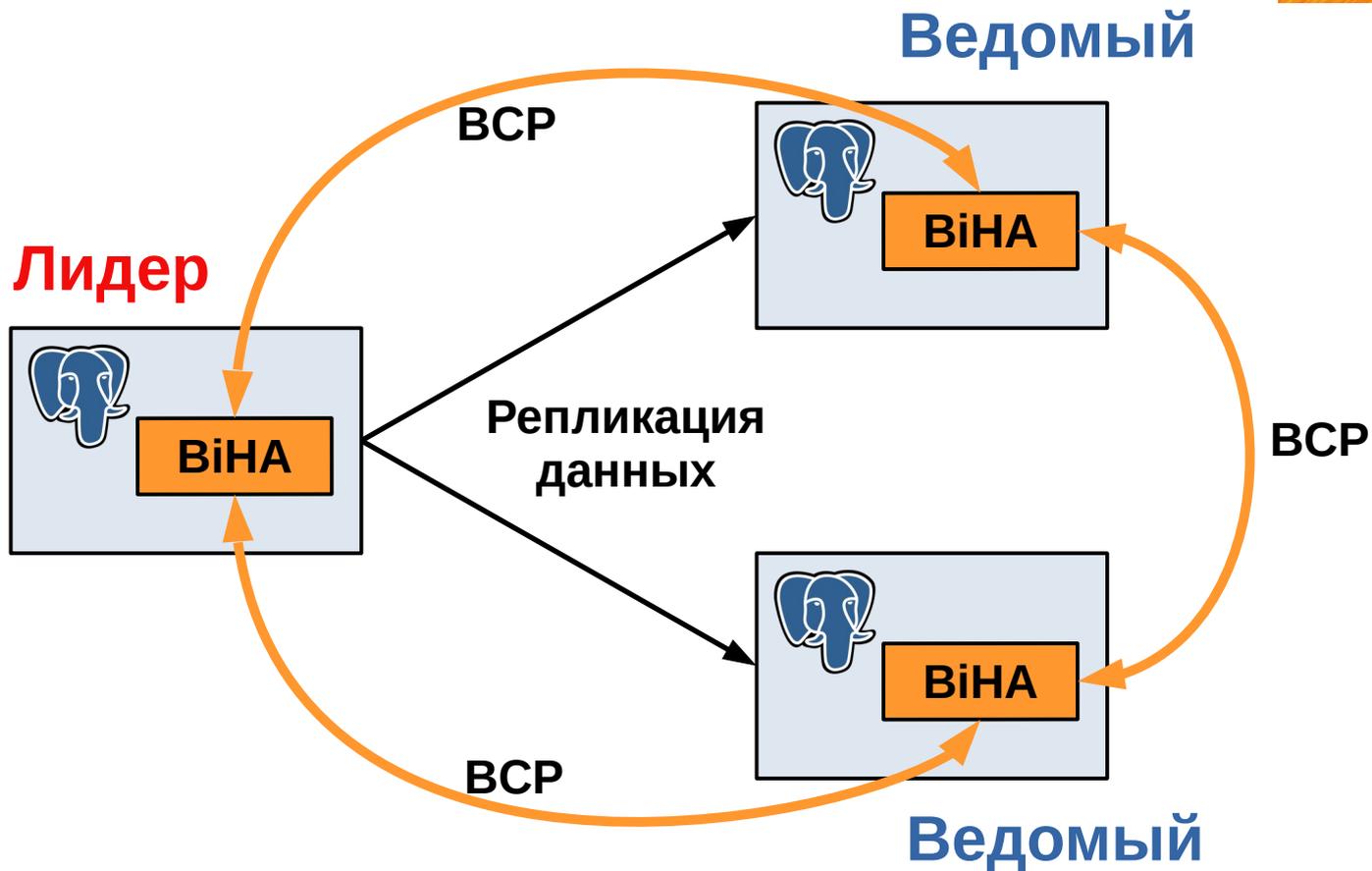
## SQL-интерфейс

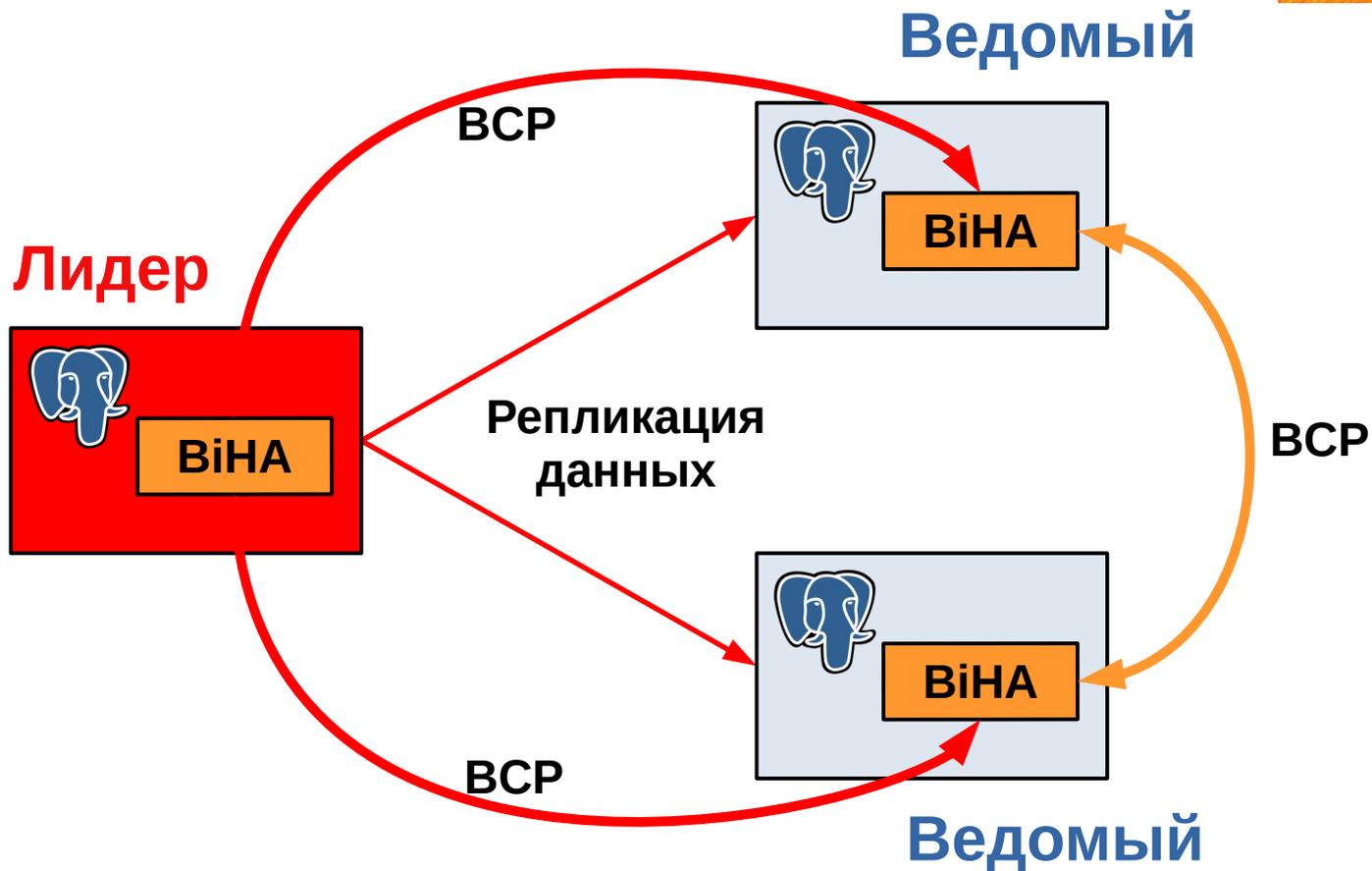
- Управление кластером в процессе работы
- Запросы состояния Узла Кластера
- Запросы состояния Кластера с точки зрения Узла Кластера



- Создание кластера
- Добавление новых Узлов к Кластеру
- Конвертация Postgres-Primary и Postgres-Primary+Standby в BiHA Кластер

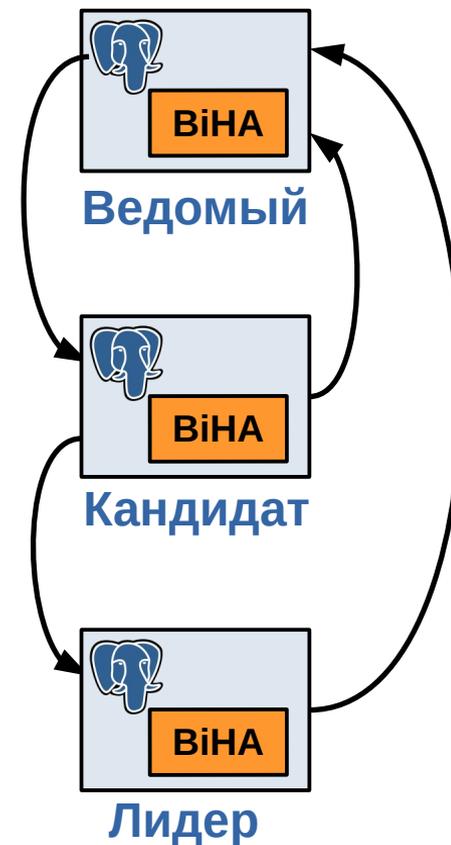


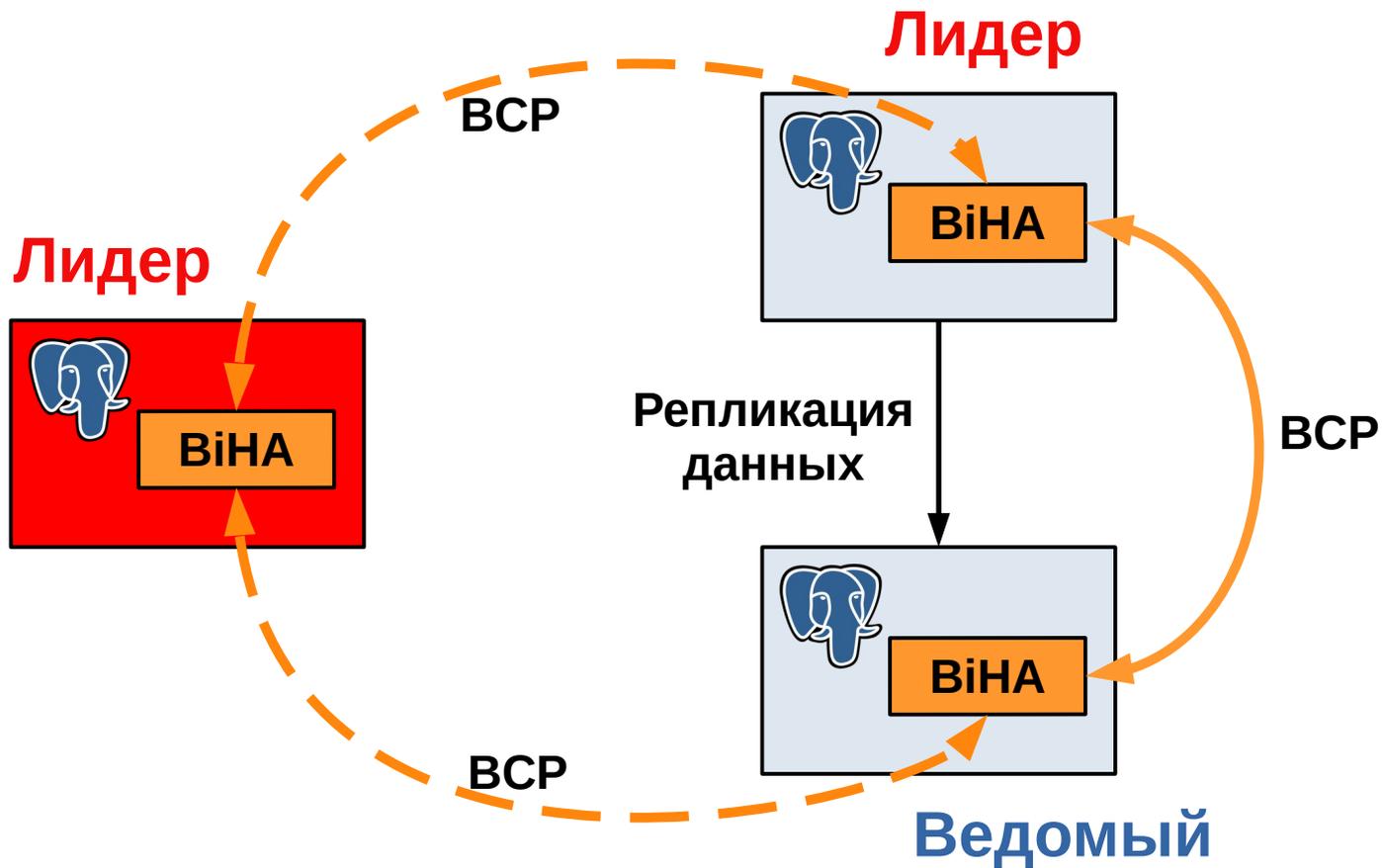


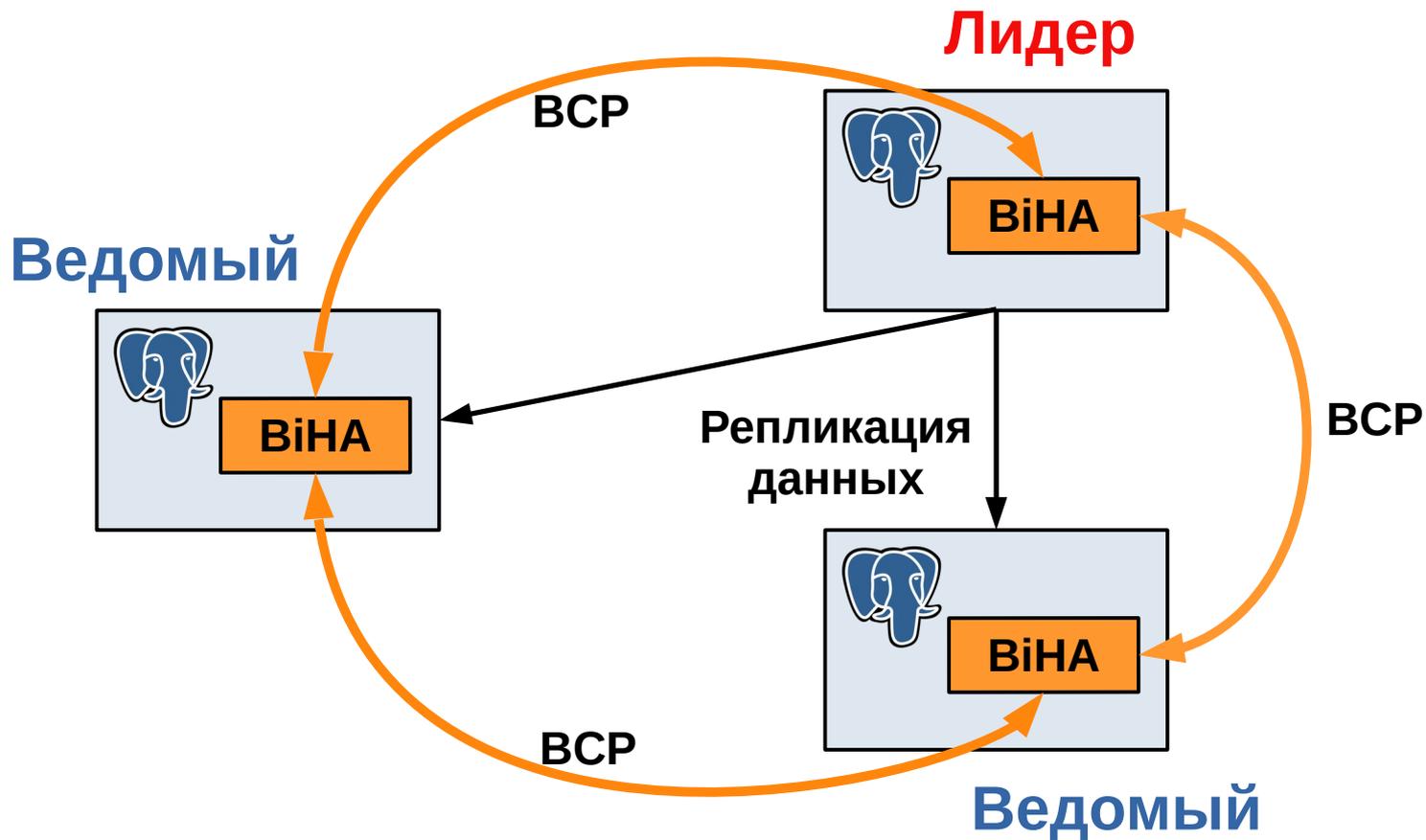


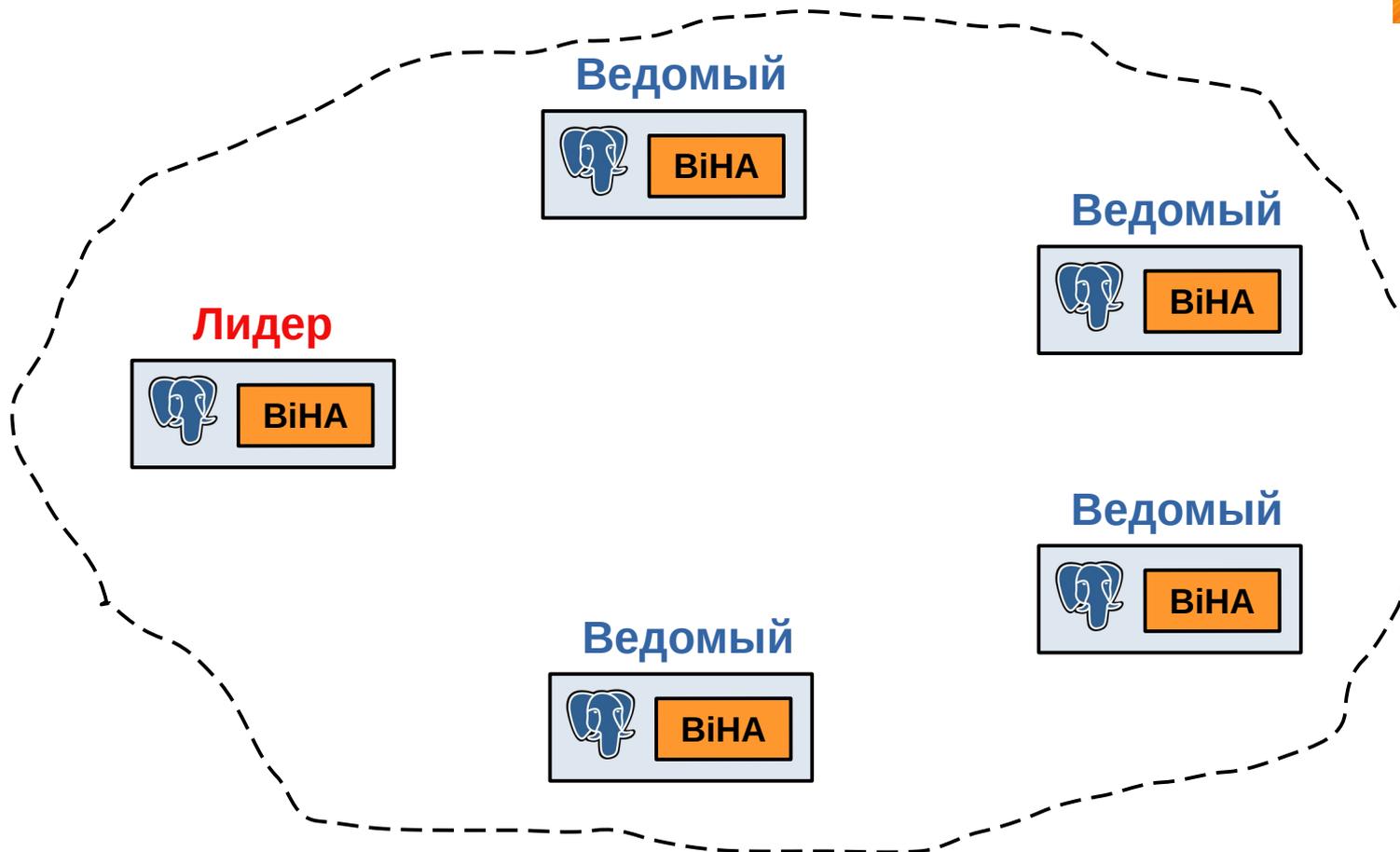
## Выборы нового Лидера

- Алгоритм на основе RAFT
- Принятие решения о выборах
- Предварительное определение наиболее подходящего Узла
- Выдвижение Кандидата
- Голосование за Кандидата
- Переключение Кандидата в Лидеры или новые выборы

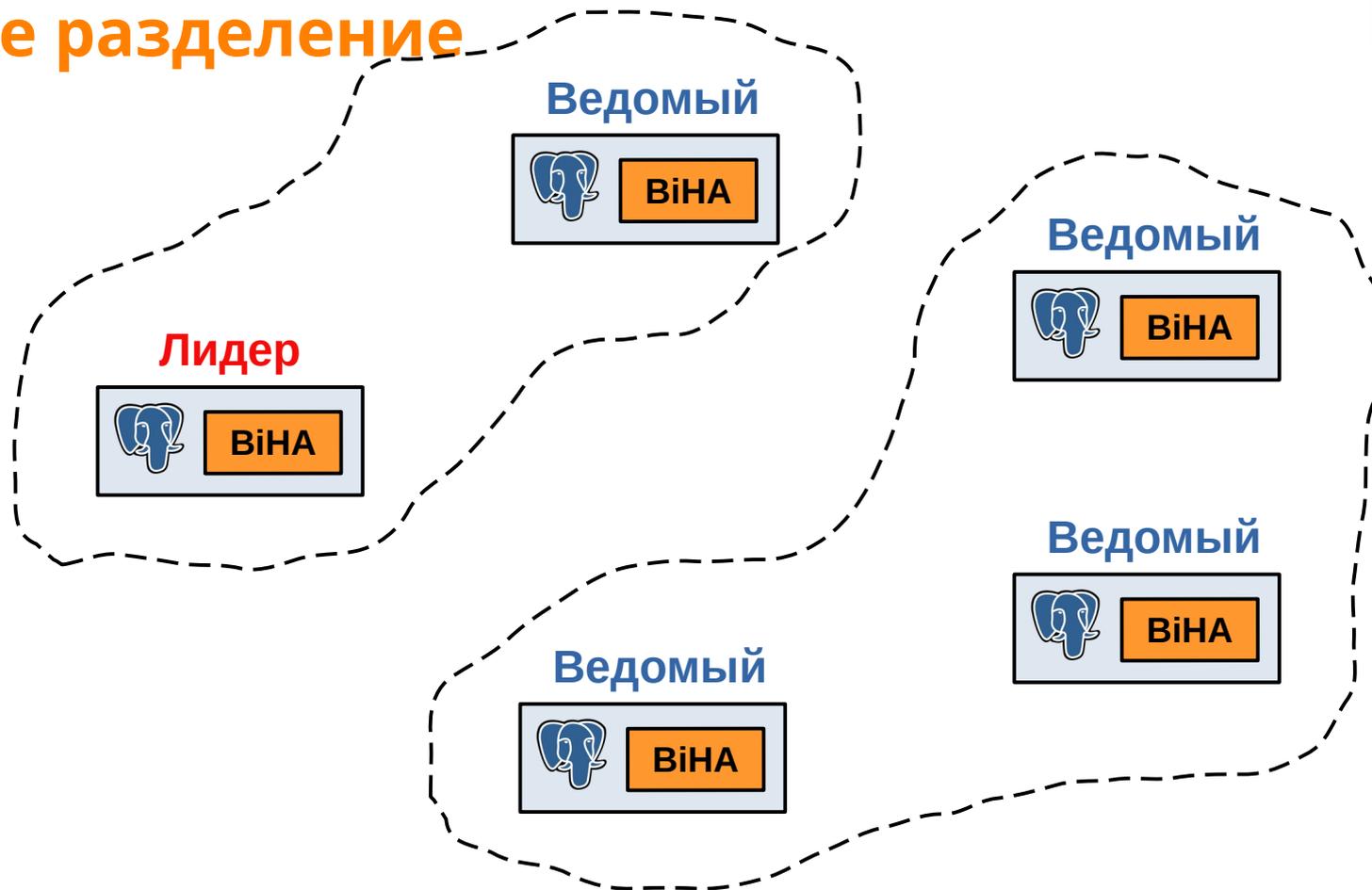




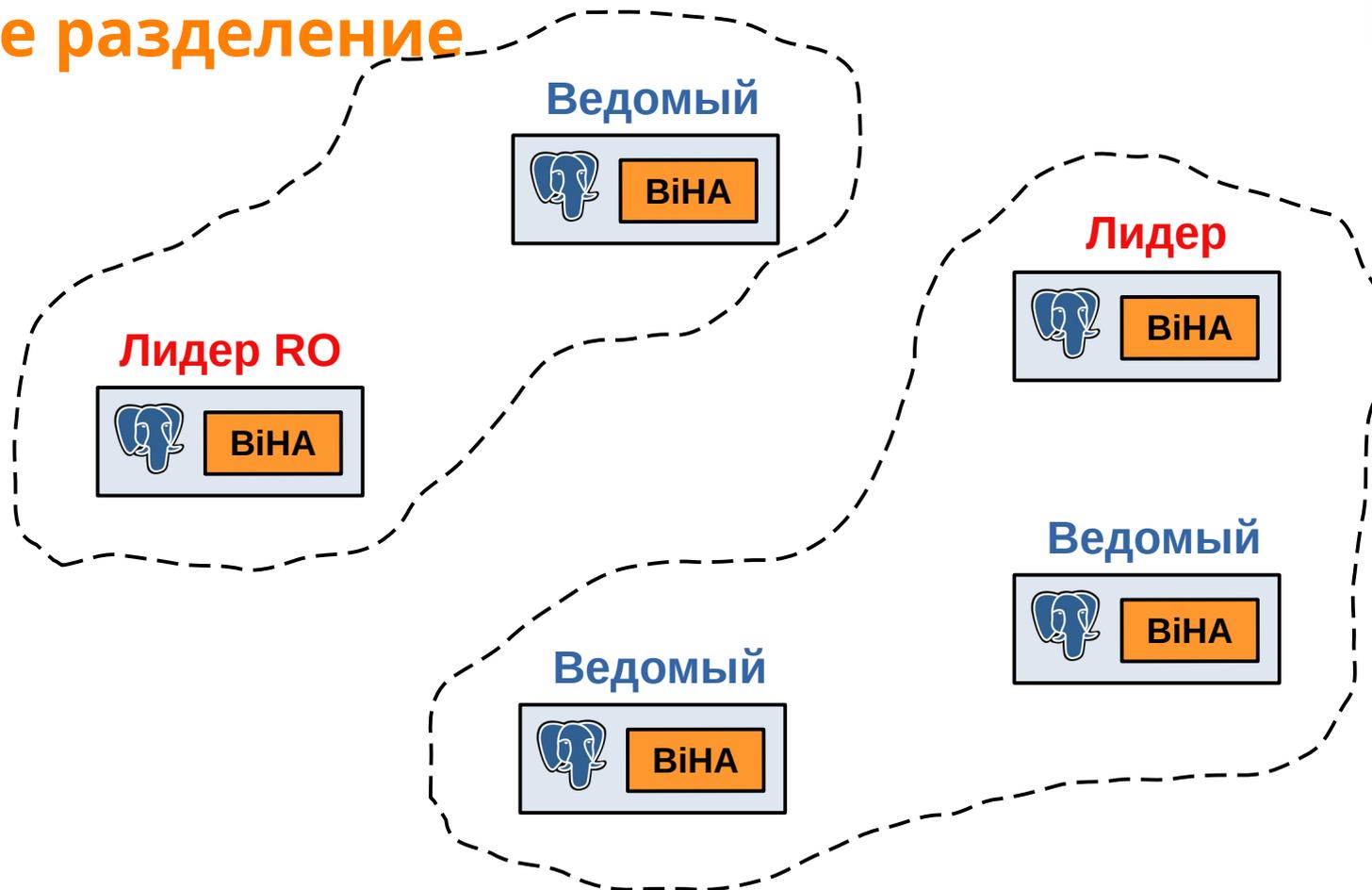


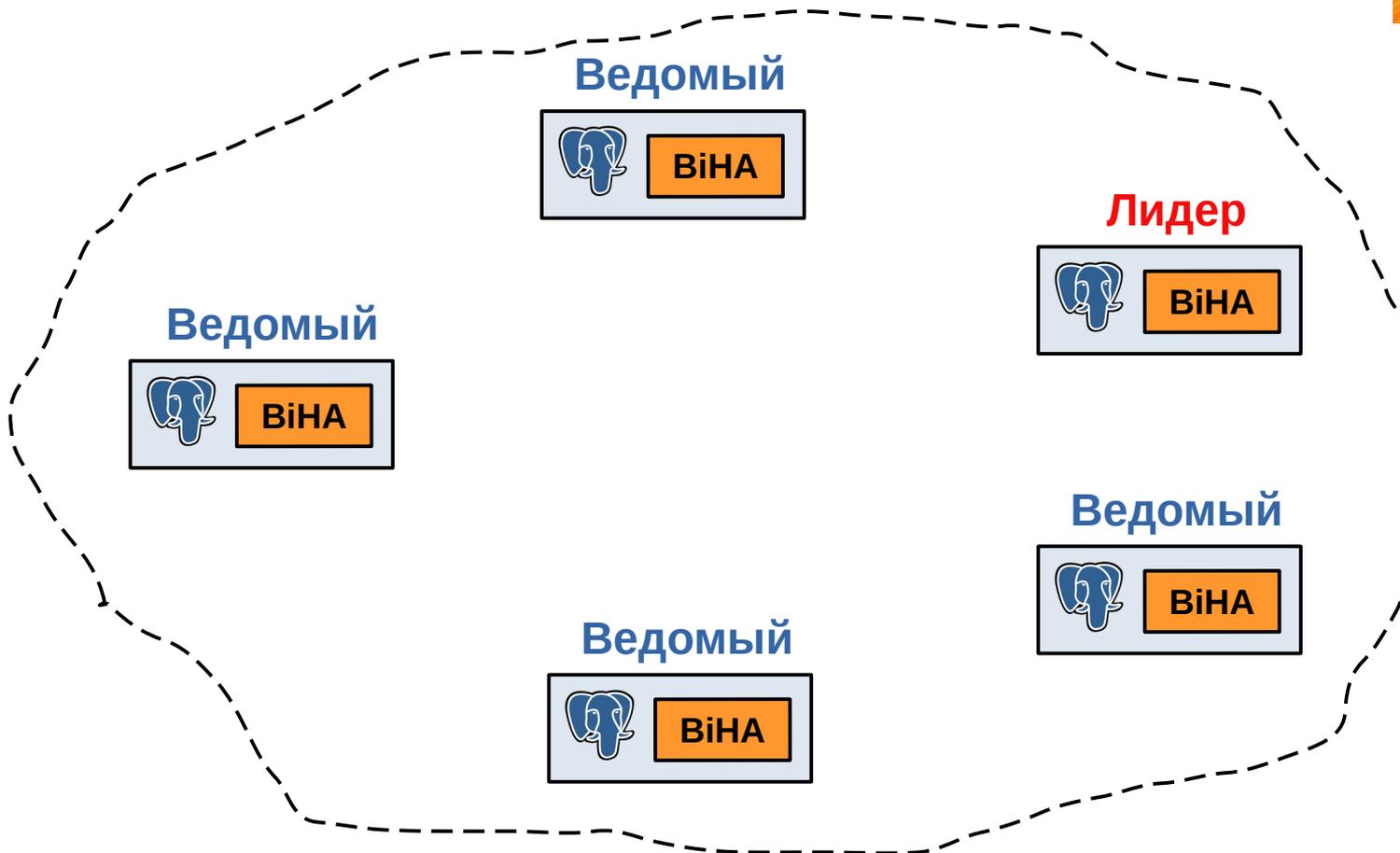


## Сетевое разделение



## Сетевое разделение





## Сценарии развертывания

- Создание кластера «с нуля»
- Создание кластера на основе Standalone Primary
- Создание кластера на основе Primary + Standby

## Простота развертывания

### Инициализация Лидера

- `bihactl init --nquorum=<num>`
- `pg_ctl start`

### Инициализация Ведомого

- `bihactl add --use_leader=<conn_str>`
- `pg_ctl start`

- Нет дополнительных администрируемых единиц
- Нет стороннего ПО
- Zero configuration
- Доступны внутреннее состояние и функции Postgres
- Физическая репликация

PGConf.Сибирь 2023

PosgresPro

**Спасибо!**