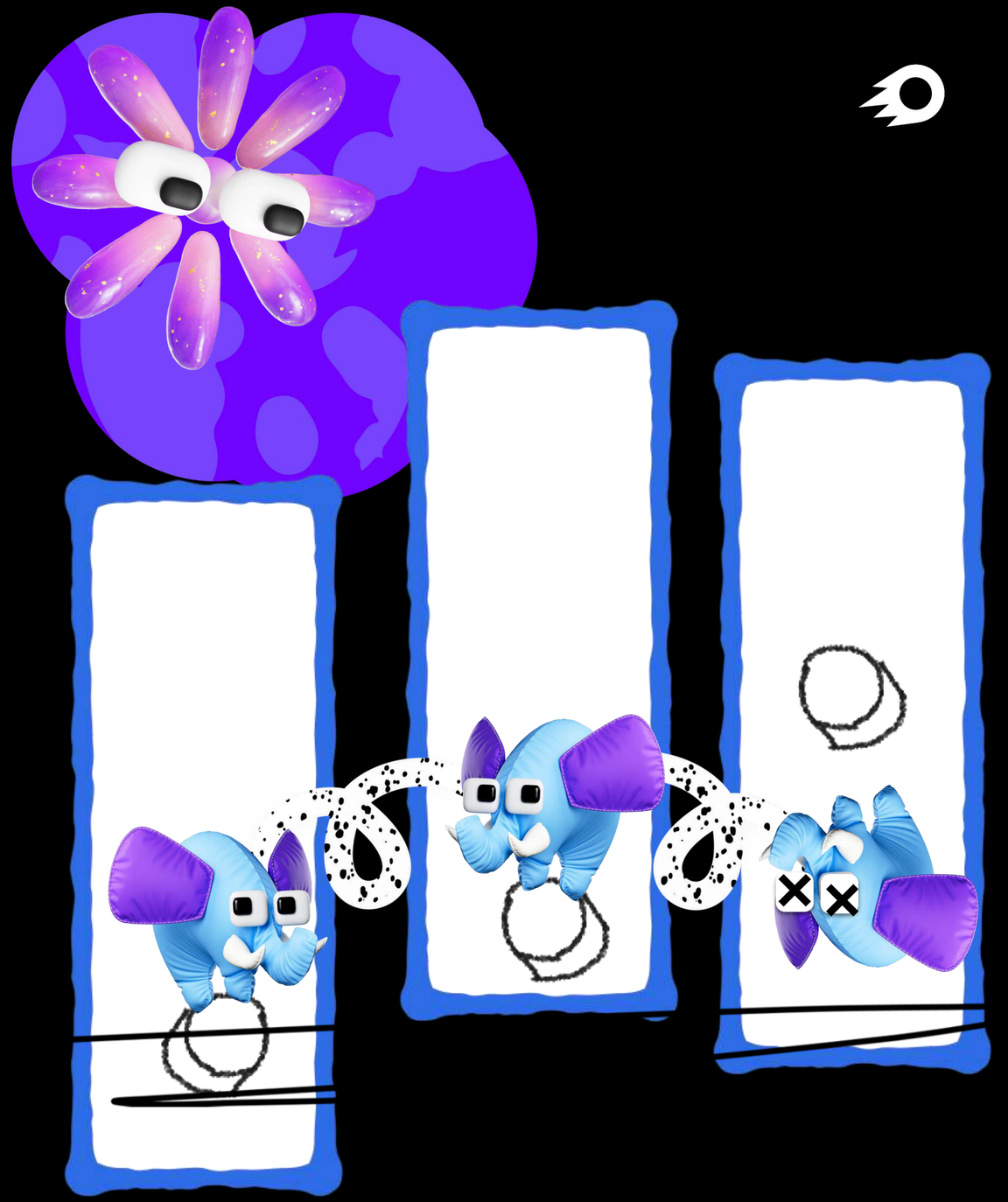
avito.tech Москва 2024

Паттерны управления базами данных в multi-cluster (multi-dc) kubernetes среде



Никита Жига инженер @ DBaaS

Полина Кудрявцева инженер @ SQL

Никита Жига

Backend-разработчик в DBaaS

- E5 инженер;
- Пилю DBaaS, занимаюсь интеграцией с PaaS;
- Работаю над DX с базами;
- Работал над DBaaS, когда он ещё был только микросервисом

Полина Кудрявцева

DBA PostgreSQL в Авито

- Инженер DBA
- Развиваем PostgreSQL и CockroachDB на платформе DBaaS

Что нас ждет

Мы обсудим:

- С какими вызовами столкнулись при проектировании платформы
- Общая схема платформы DBaaS
- Какие паттерны проектирования платформы выделили
- Как видят базы данных пользователи - Database Discovery

DBaaS в цифрах

>4 тыс.

баз зарегистрировано

Включая разные контуры: production, staging, perfomance

Более половины микросервисов используют базы данных
Все они выданы через платформу DBaaS

>20 тыс.

инстансов баз запущено

Включая шардированные инсталляции и без HA

Это **PostgreSQL**, **Redis**, **MongoDB**, ZooKeeper, ClickHouse, Kafka, Elasticsearch, **CockroachDB**, Tarantool.

>500 тыс.

успешных операций над базами

По API. Применение этих операций автоматическое

Типичная такая операция — пересмотр лимитов базы. Обслуживанием занимается команда из ~10 человек.

Вызовы

1 2 3 4

Требования к платформе

Домен отказа - датацентр

Базы данных должны переживать отказ любого из датацентров в любое время

Scaling

Должна существовать возможность горизонтального и вертикального масштабирования

Scheduling

Необходим автоматический шедулинг с учетом распределения ресурсов по DC и по k8s-кластерам

High Availability

Предоставление пользователям высокой доступности баз данных

Discovery

Предоставление пользователям актуальной информации, необходимой для работы с базой данных

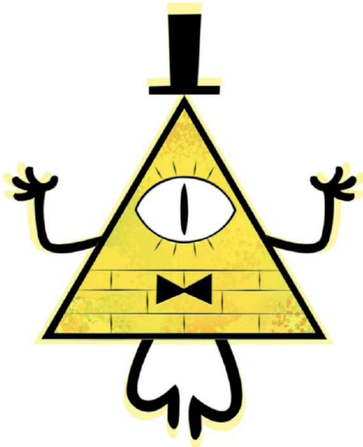
…и еще требования платформе

Сохранность данных

Регулярные бэкапы, проверка их валидности, возможность восстановления на определенный момент времени

Доставка секретов

Автоматическая доставка, синхронизация, ротация секретов клиентов баз данных



TLS, mTLS

Доступ к базе данных по зашифрованном каналам

Ограничение ресурсов

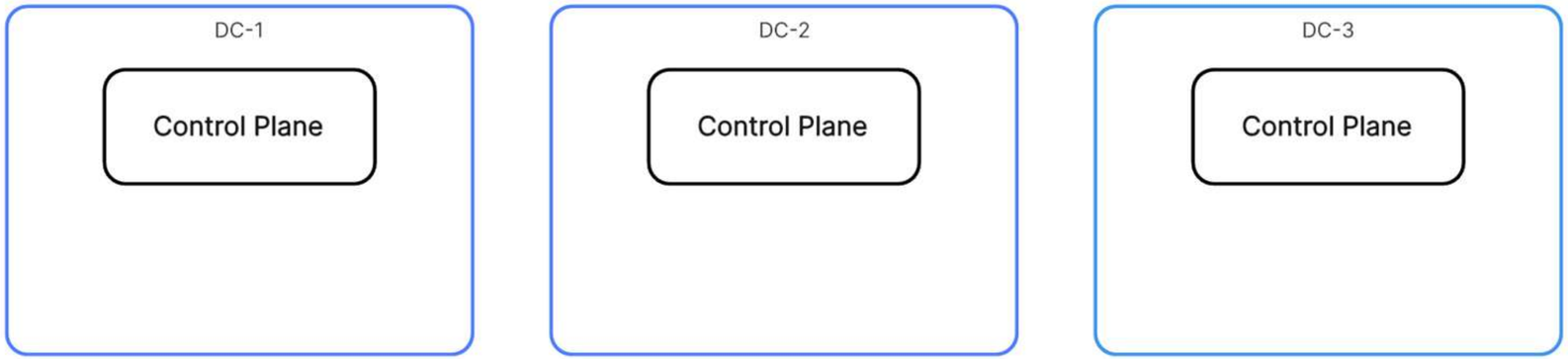
Отсутствие влияния шумных соседей

Общая архитектура

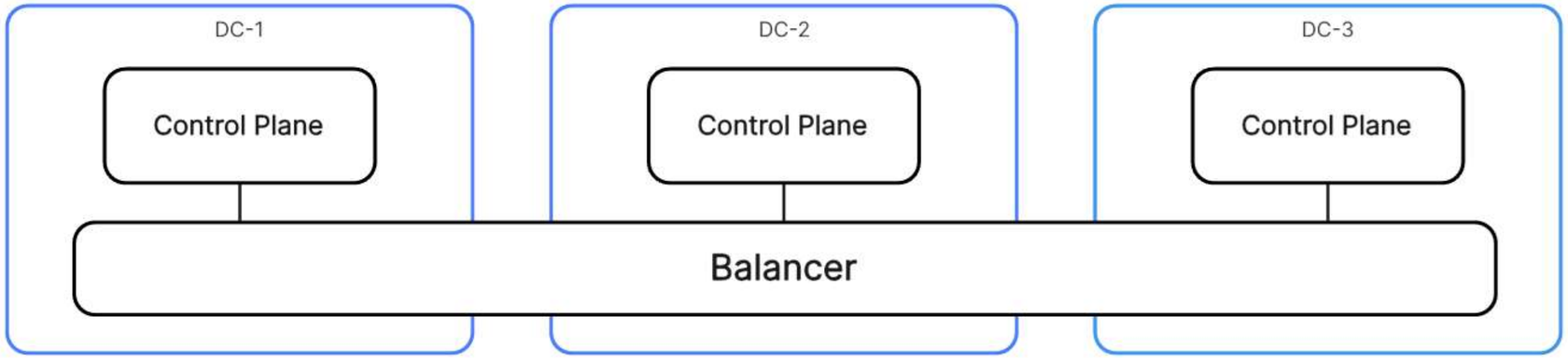
1 2 3 4

Общая архитектура

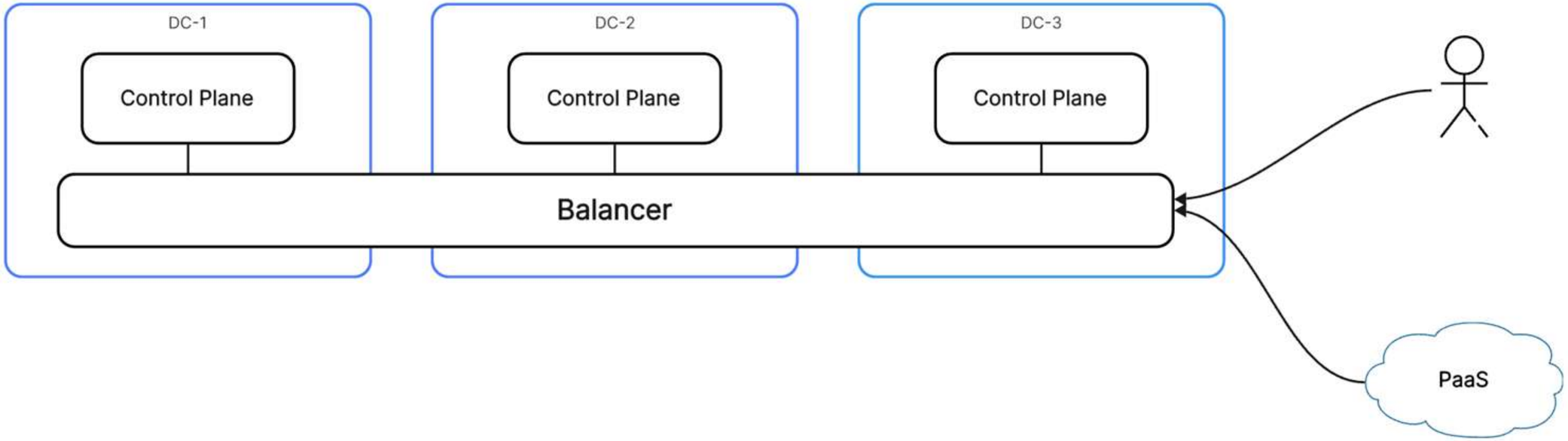
Общая архитектура



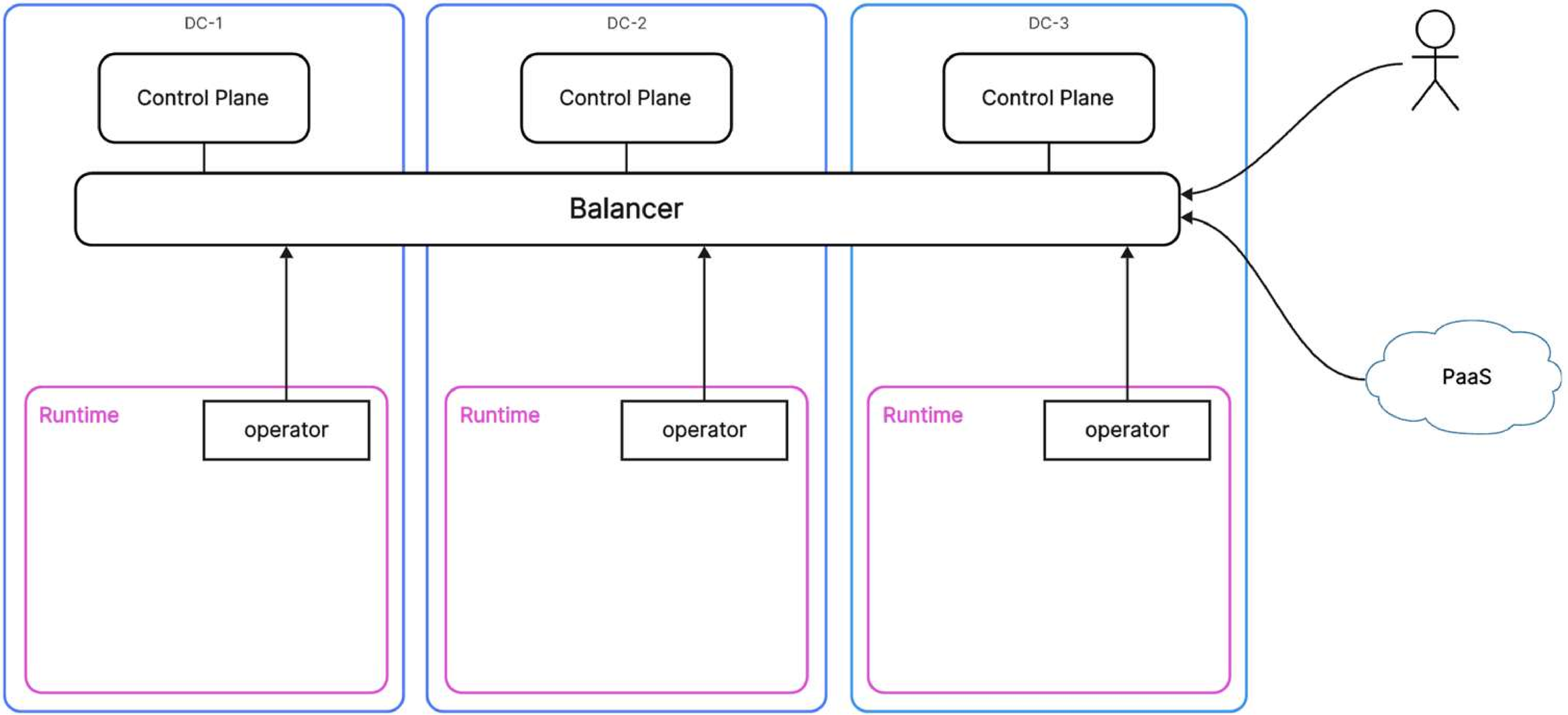
Общая архитектура



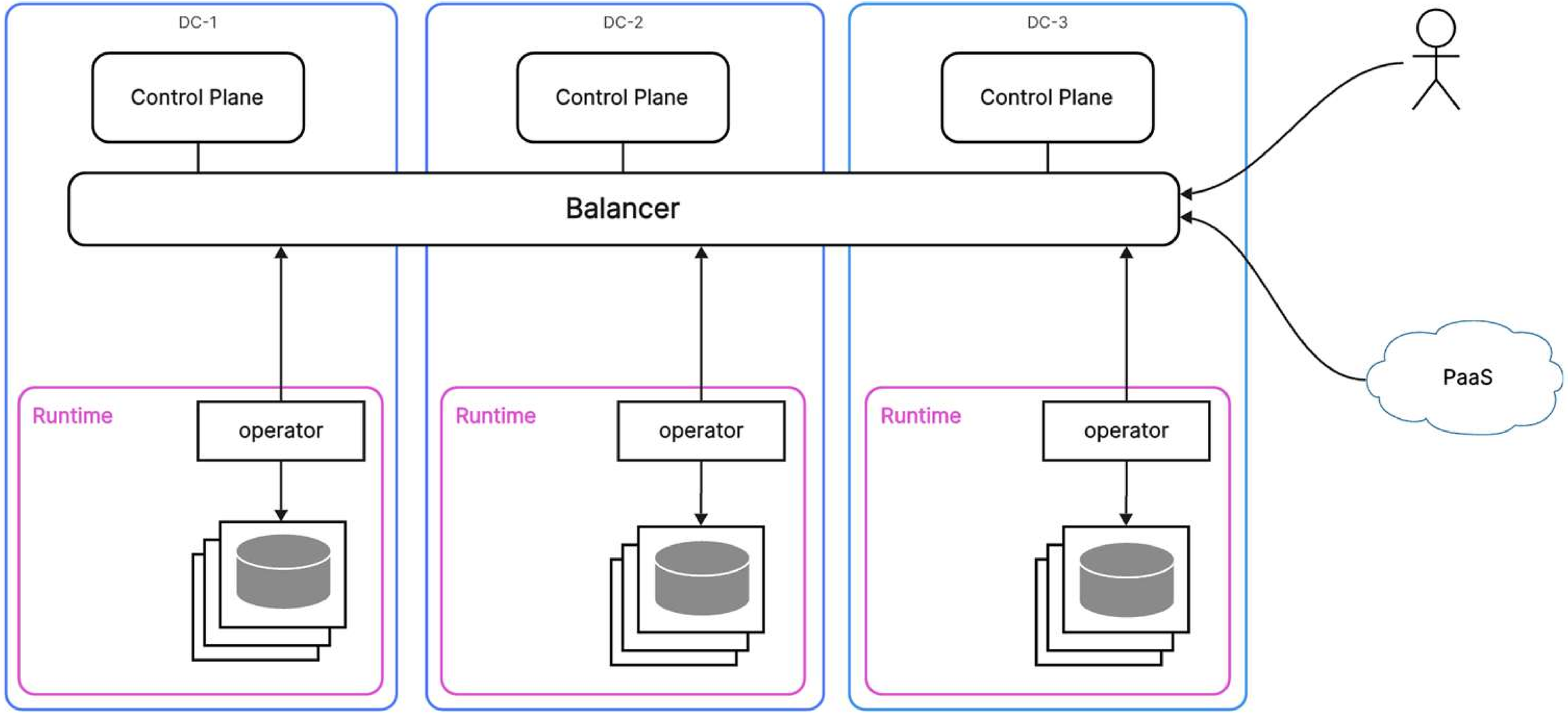
Общая архитектура



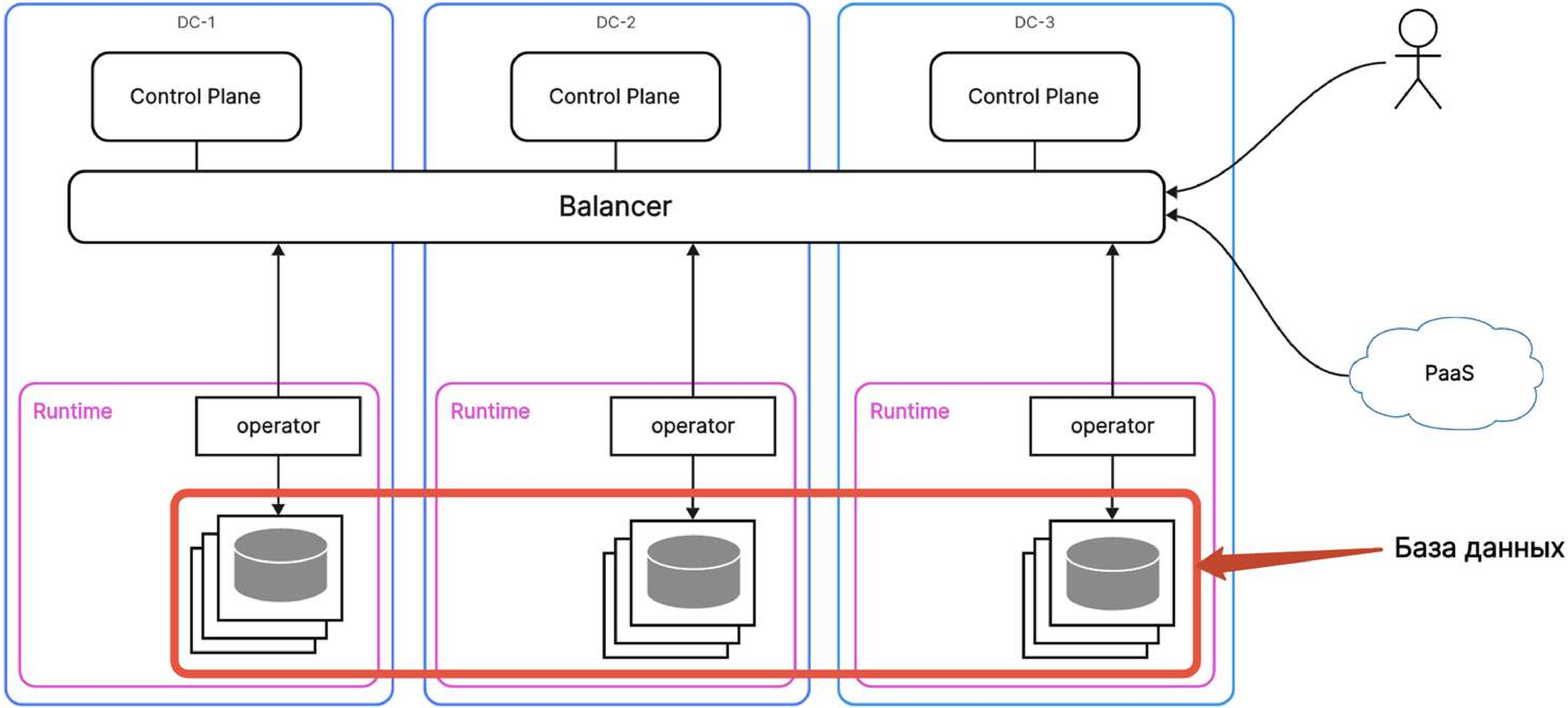
Общая архитектура



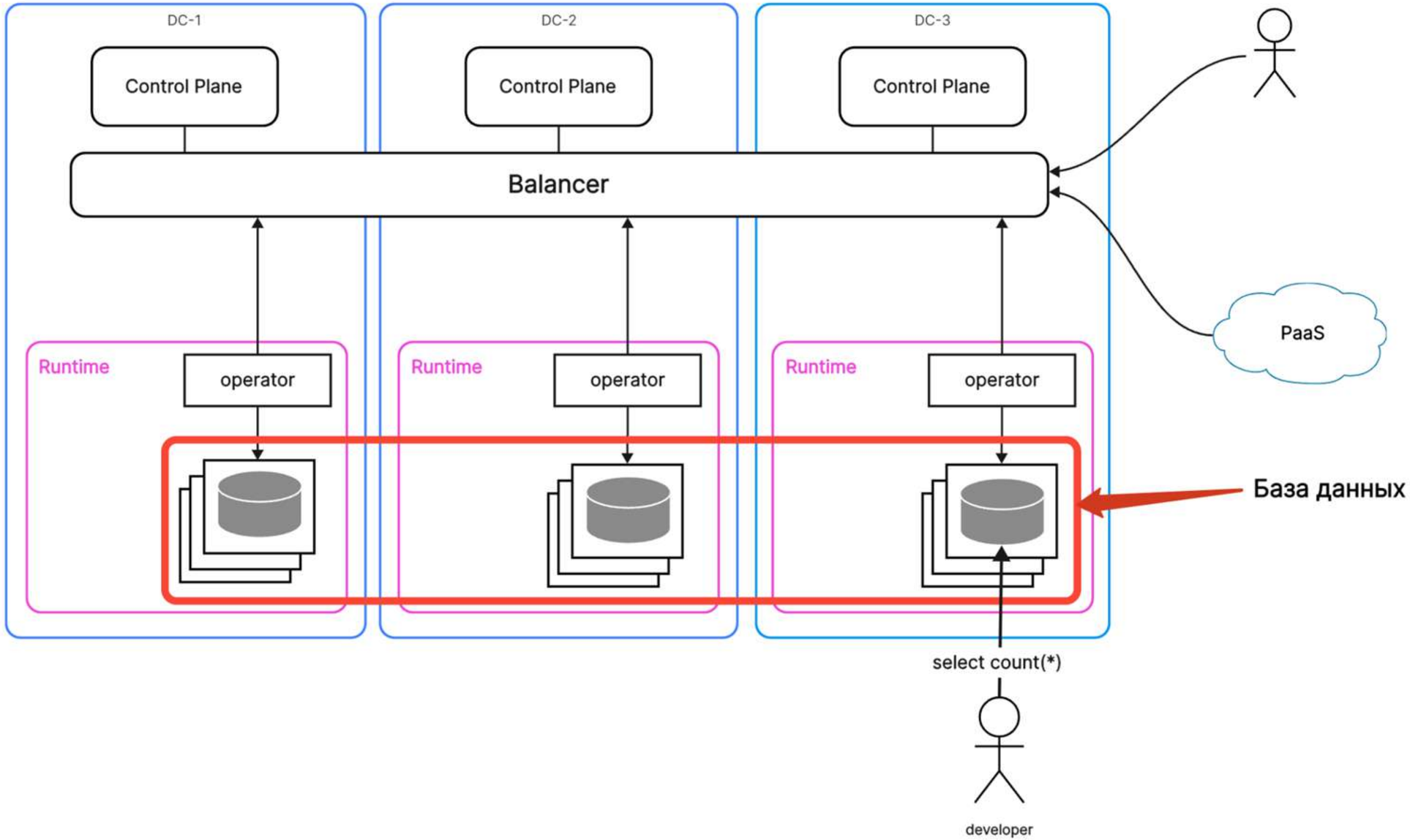
Общая архитектура



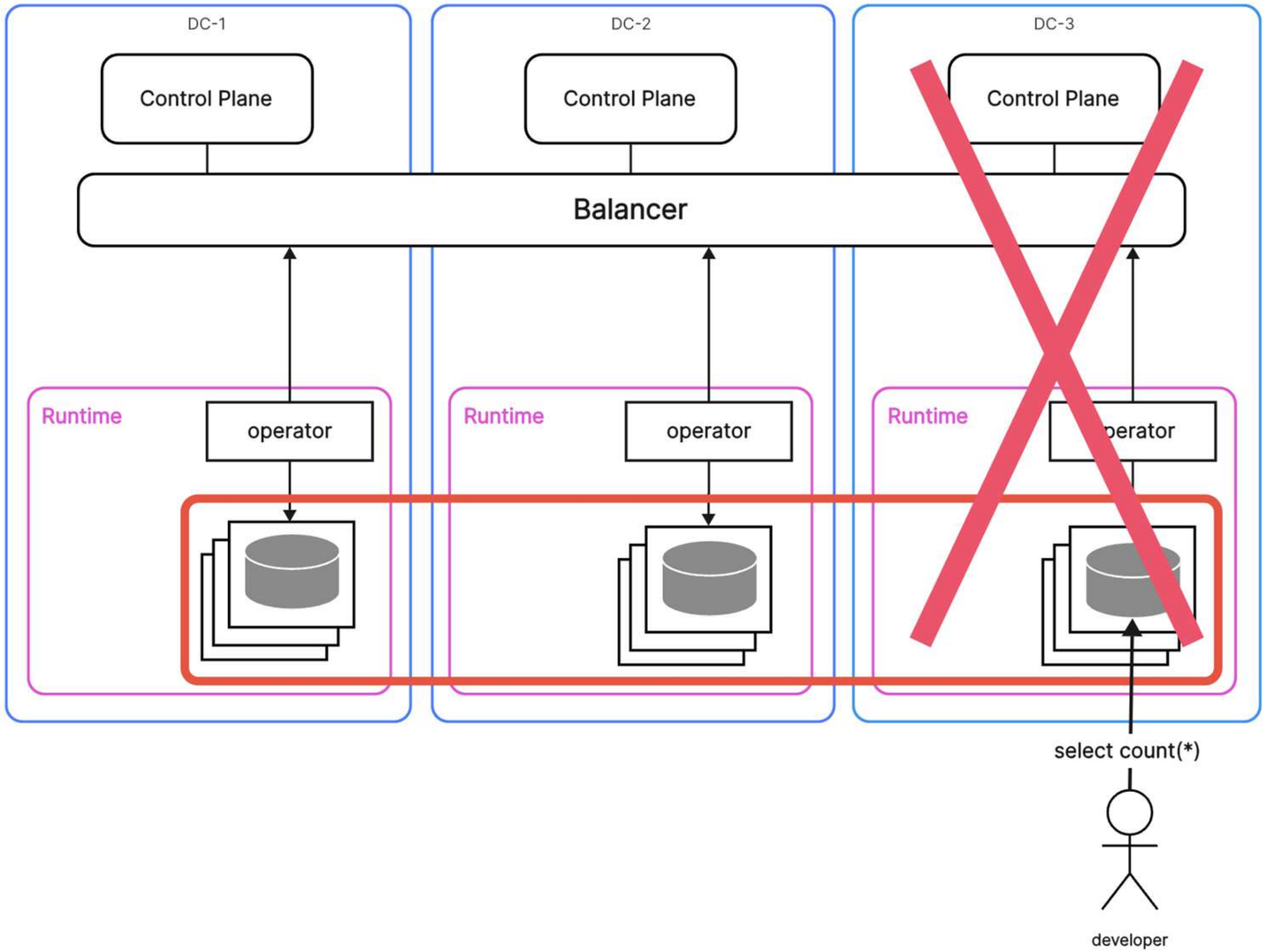
Общая архитектура



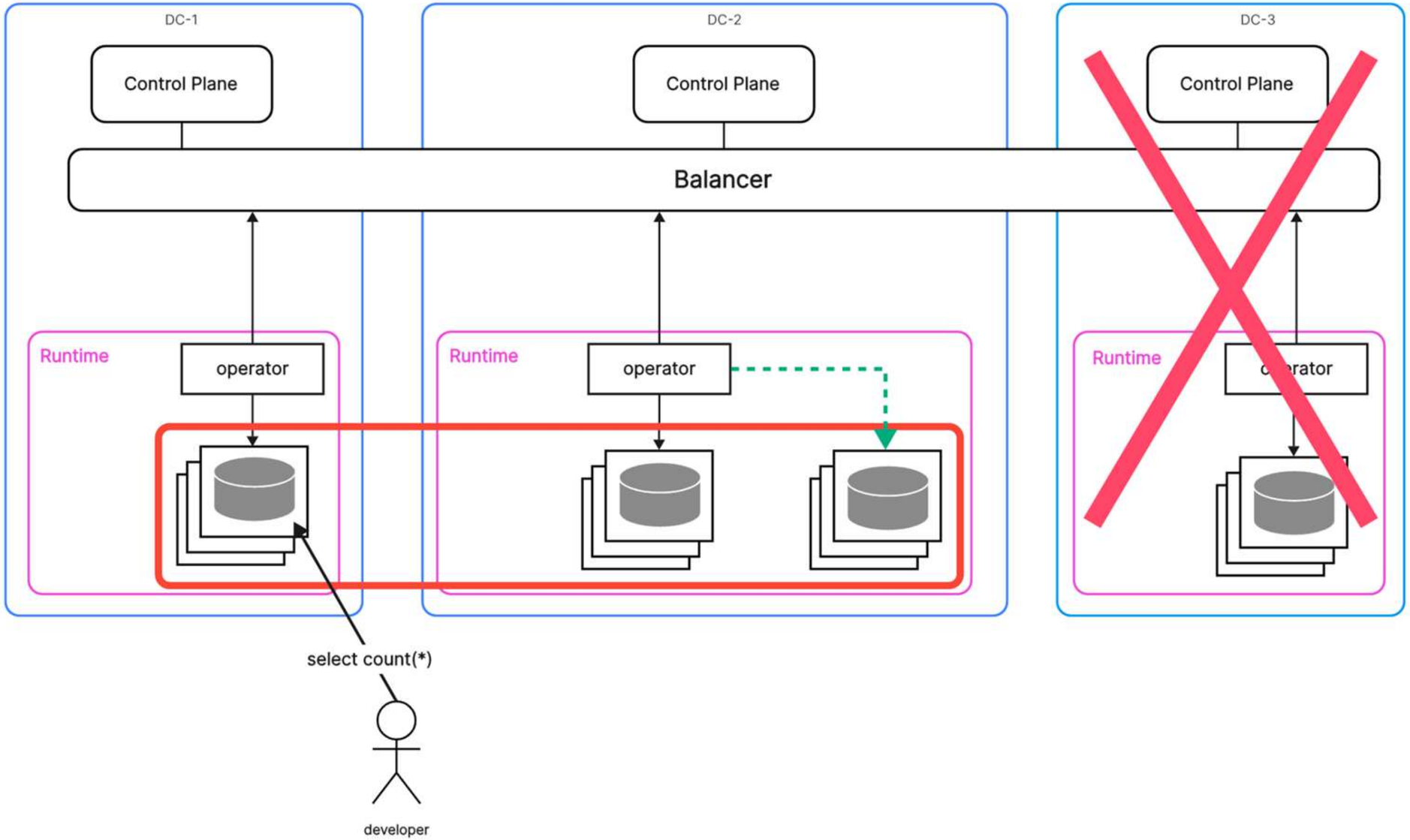
Общая архитектура



Общая архитектура



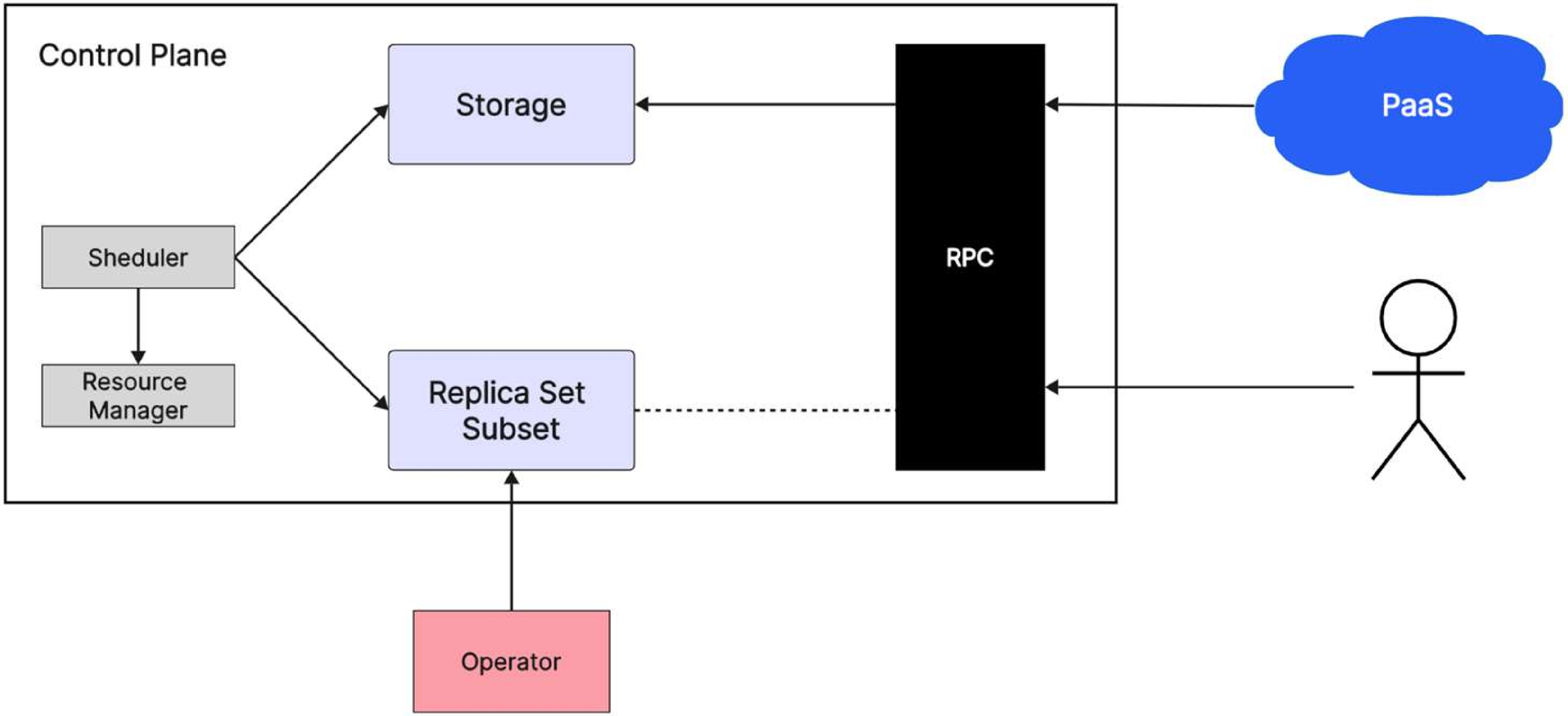
Общая архитектура



Выработанные паттерны

1 2 3 4

Паттерн метаоператора



Немного про данные

Storage

- Храним информацию о базах в виде данных
- Управление данными закрыто за CRUD API
- Редактирование данных возможно прямо через консольную утилиту

```yaml
name: that-one-database
software: postgresql # redis, mongodb или cockroachdb
replica_sets:
- name: rs001
  replicas: 3
  config:
    # конфигурация реплики
requirements:
  platform: k8s
  alerts:
    # что делать с алертами
  backup:
    # что делать с бекапами
  locality:
    # доступы к репликам
  resources:
    # сколько база потребляет
  scheduling:
    # настройки по шедулингу
environment: prod
info:
  # мета-информация
config:
  # всякая конфигурация базы
access:
  # доступы
```

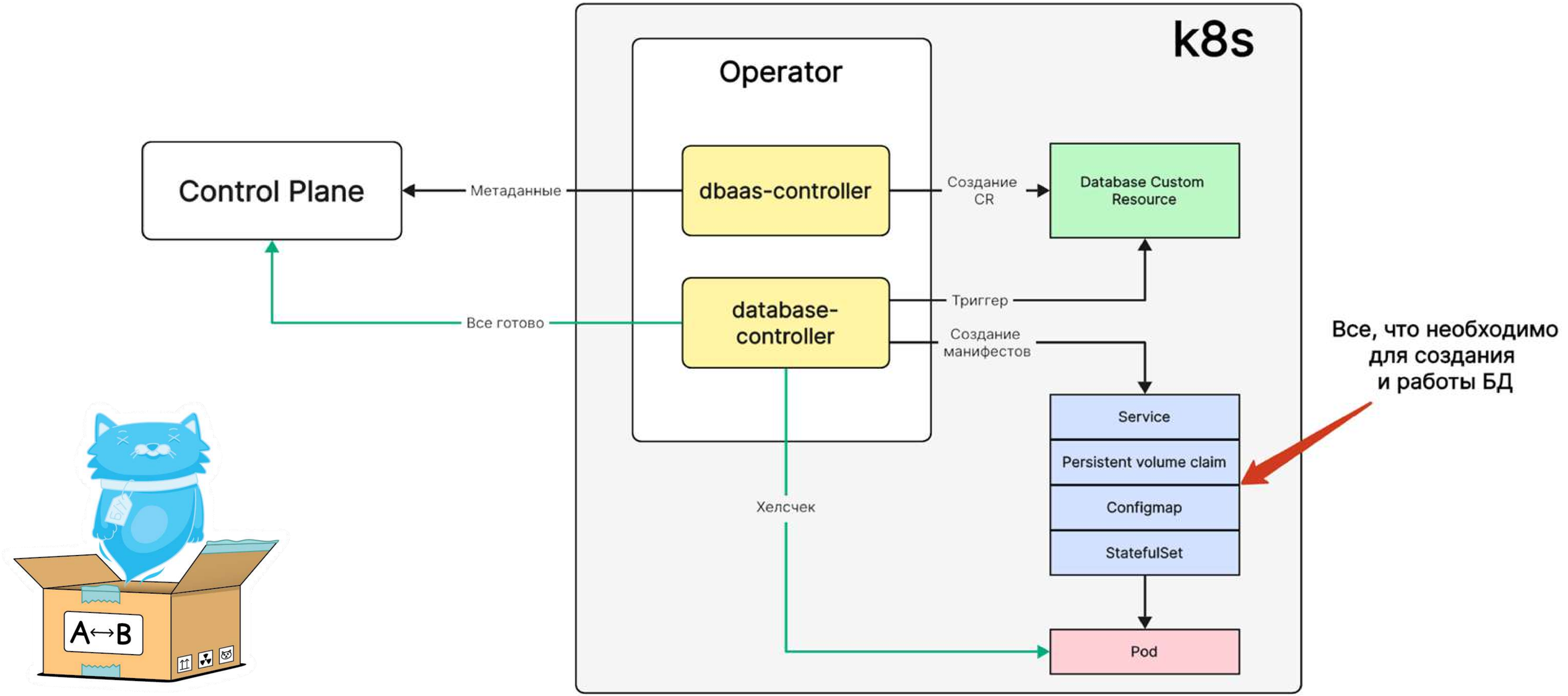
Немного про данные

Replica Set Subset

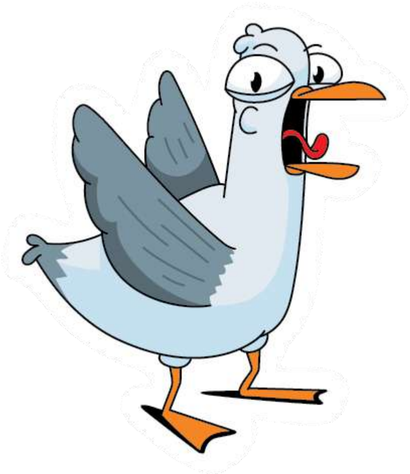
- Создаётся на основе метаданных Storage
- Шедулится в соответствии с заданным доменом отказа
- Содержит runtime информацию и текущий статус
- Механизм rolling update реализован последовательно по replicaset subset-ам - полной недоступности базы не будет

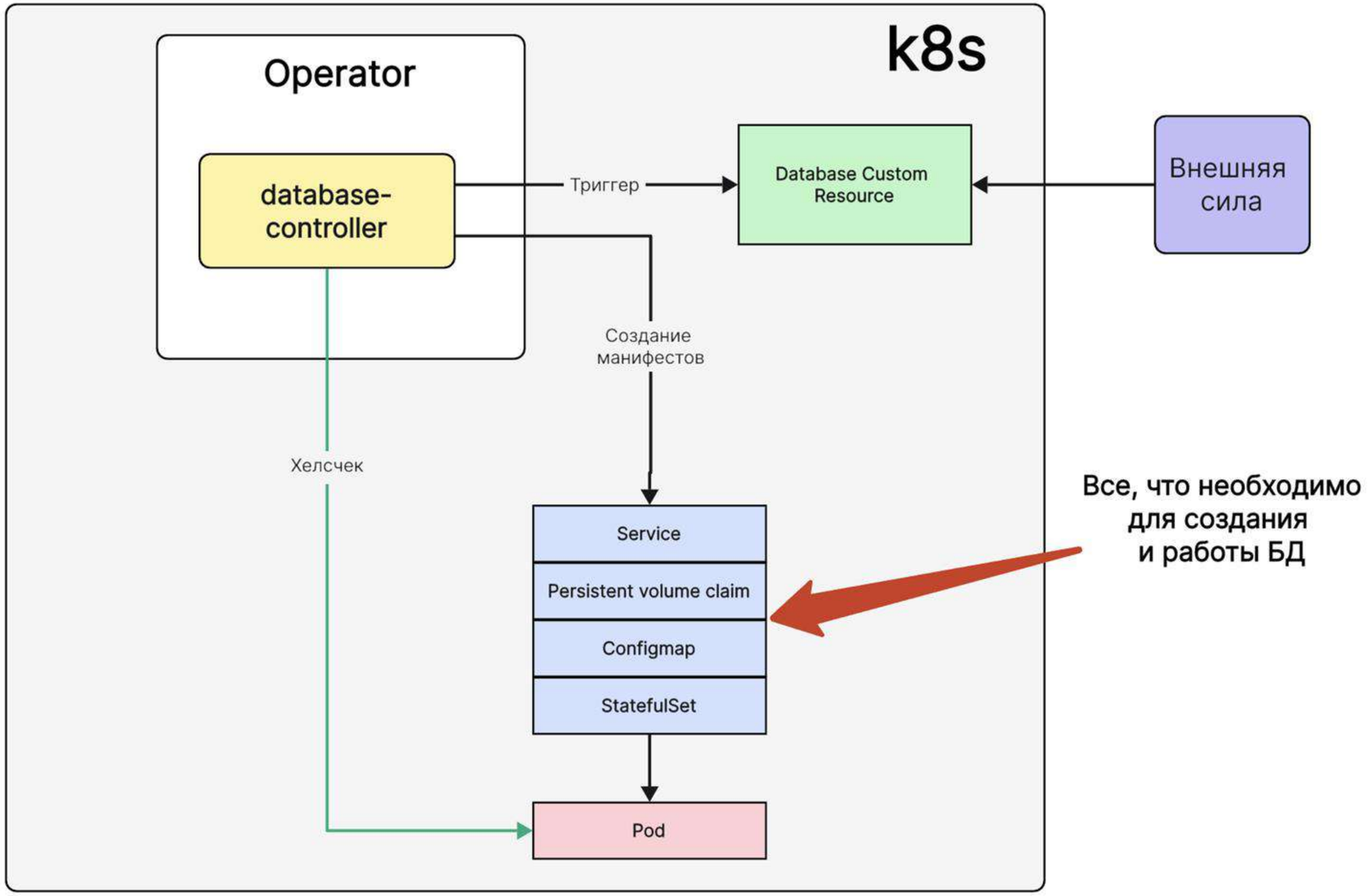
```
storage: # ...
k8s_replica_set_subsets:
- storage_name: that-one-database
  replica_set_name: rs001
  kubernetes_cluster: dc-1
  replicas: 1
  update_status: completed
  runtime_info: # порты, endpoint-ы

- storage_name: that-one-database
  replica_set_name: rs001
  kubernetes_cluster: dc-2
  replicas: 1
  update_status: completed
  runtime_info: # ...

- storage_name: that-one-database
  replica_set_name: rs001
  kubernetes_cluster: dc-3
  replicas: 1
  update_status: completed
  runtime_info: # ...
```

Паттерн оператора (prod)

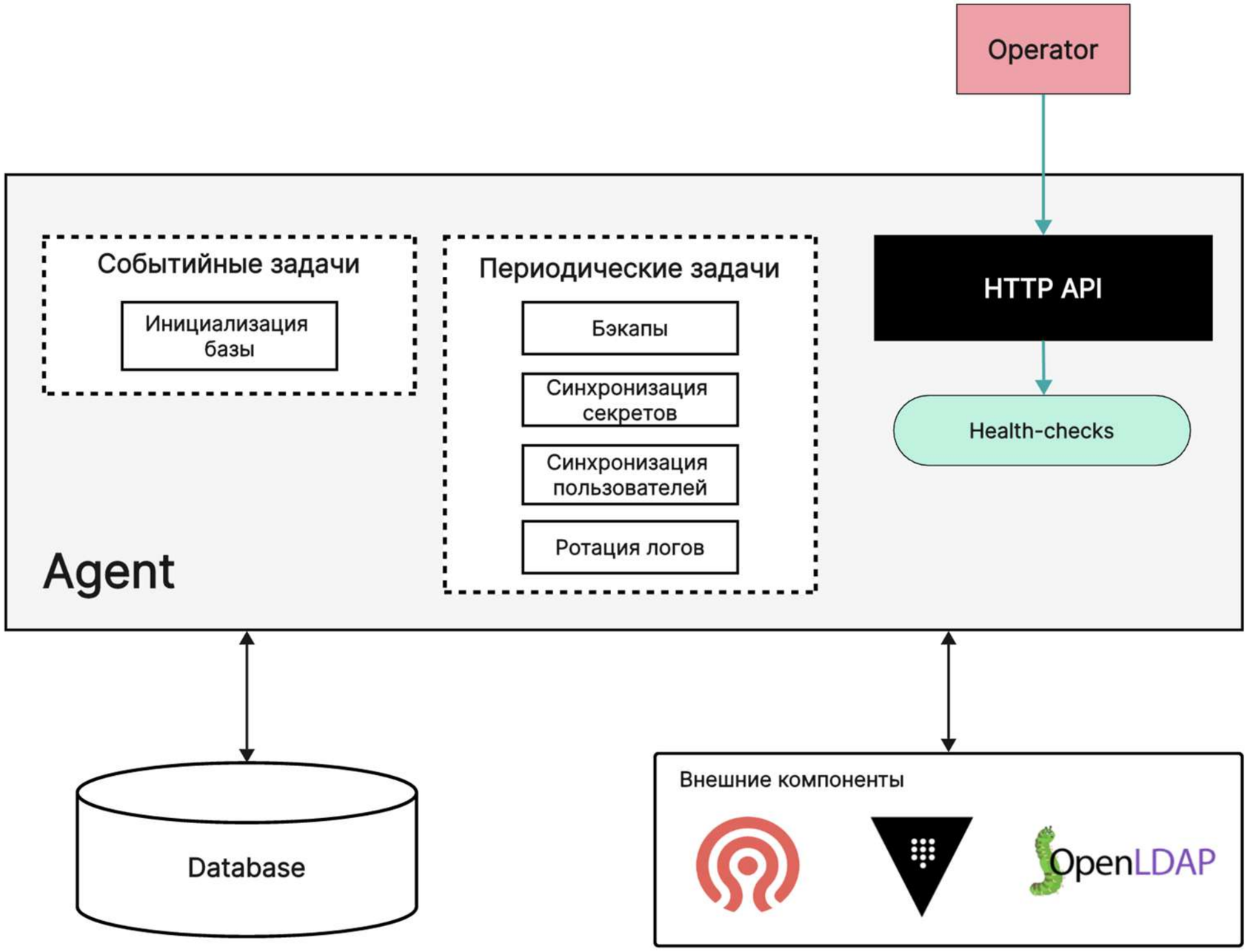


Паттерн оператора (local)





Паттерн агентов



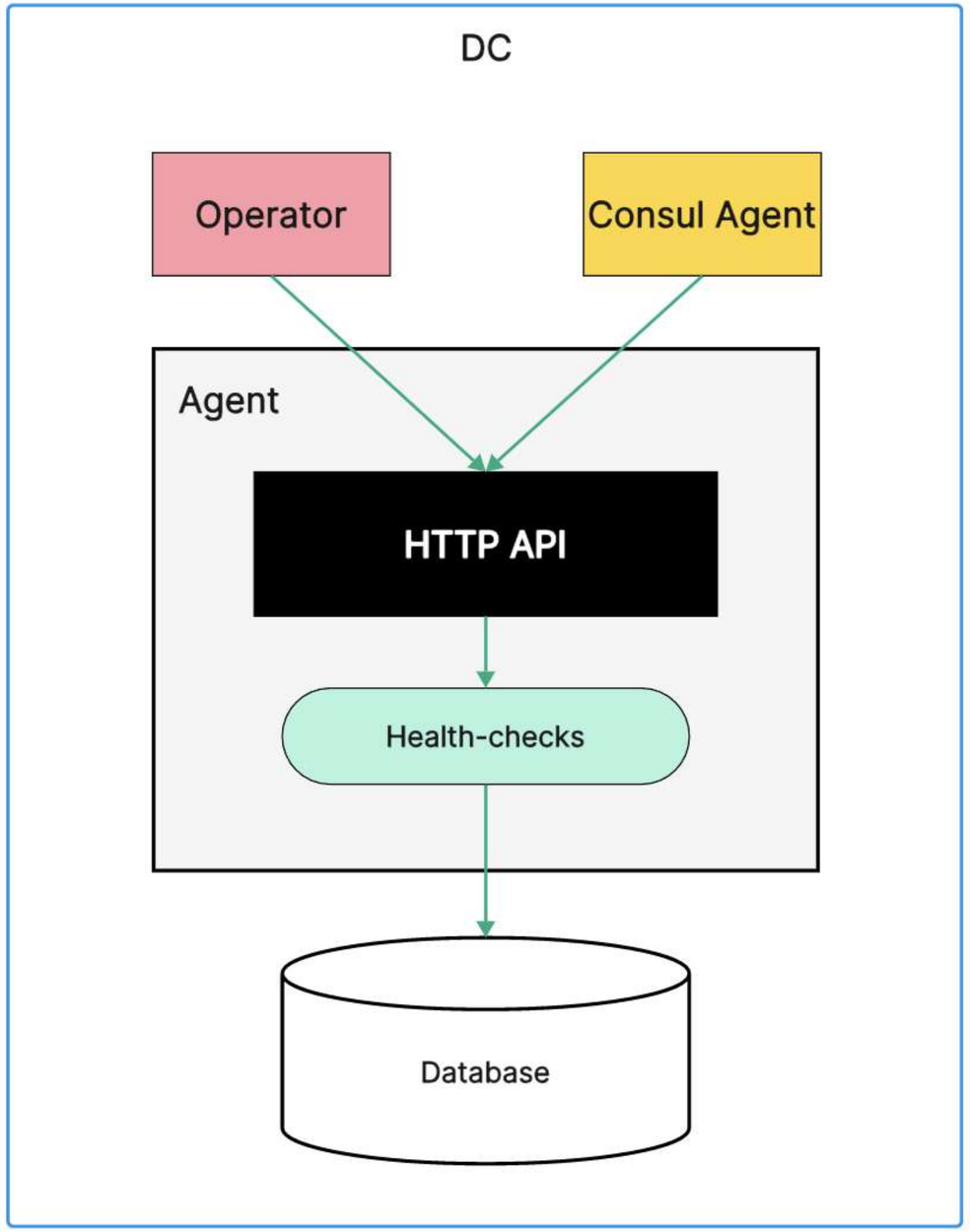
Как работают health-check-и

Мы отказались от k8s liveness/readiness probes. Почему?

- Неопределенное поведение в случае падения мастера
- Неопределенное поведение в случае создания новой реплики
- Перезапуск пода в любой непонятной ситуации
- Может принять восстановление базы за нерабочую базу
- Backoff limit приводит к неконтролируемому поведению рестарта инстанса
- Liveness/Readiness probe привязаны к контейнеру — изменение ведет за собой перезапуск БД

Благодаря реализации health-check-ов на стороне агента:

- Пробы не привязаны к поду и основаны на особенностях софта
- Можем крутить без рисков перезапуска базы
- Не привязаны к ограничениям K8S

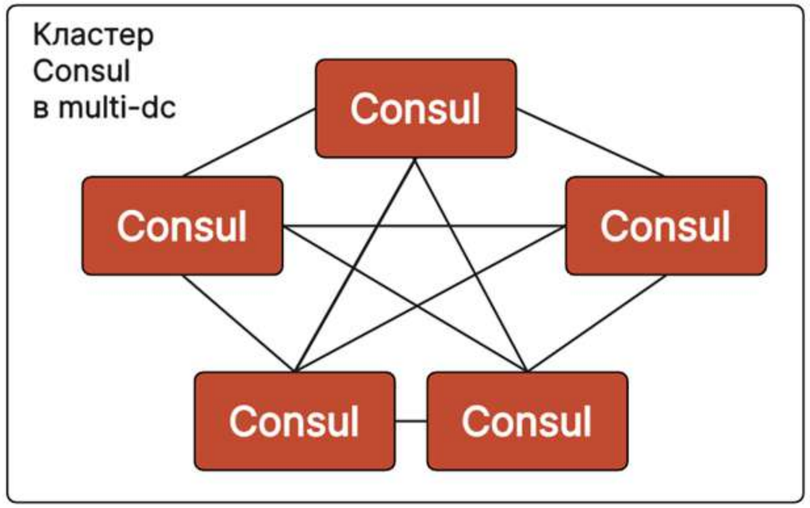


Database discovery

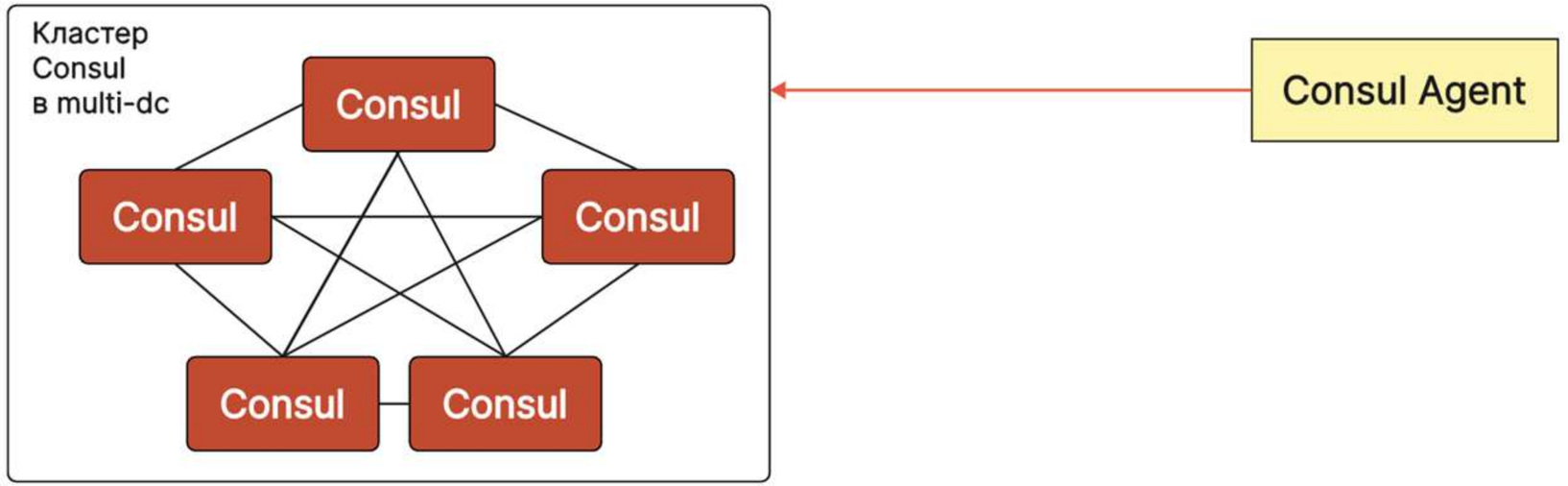
и локализация трафика

1 2 3 4

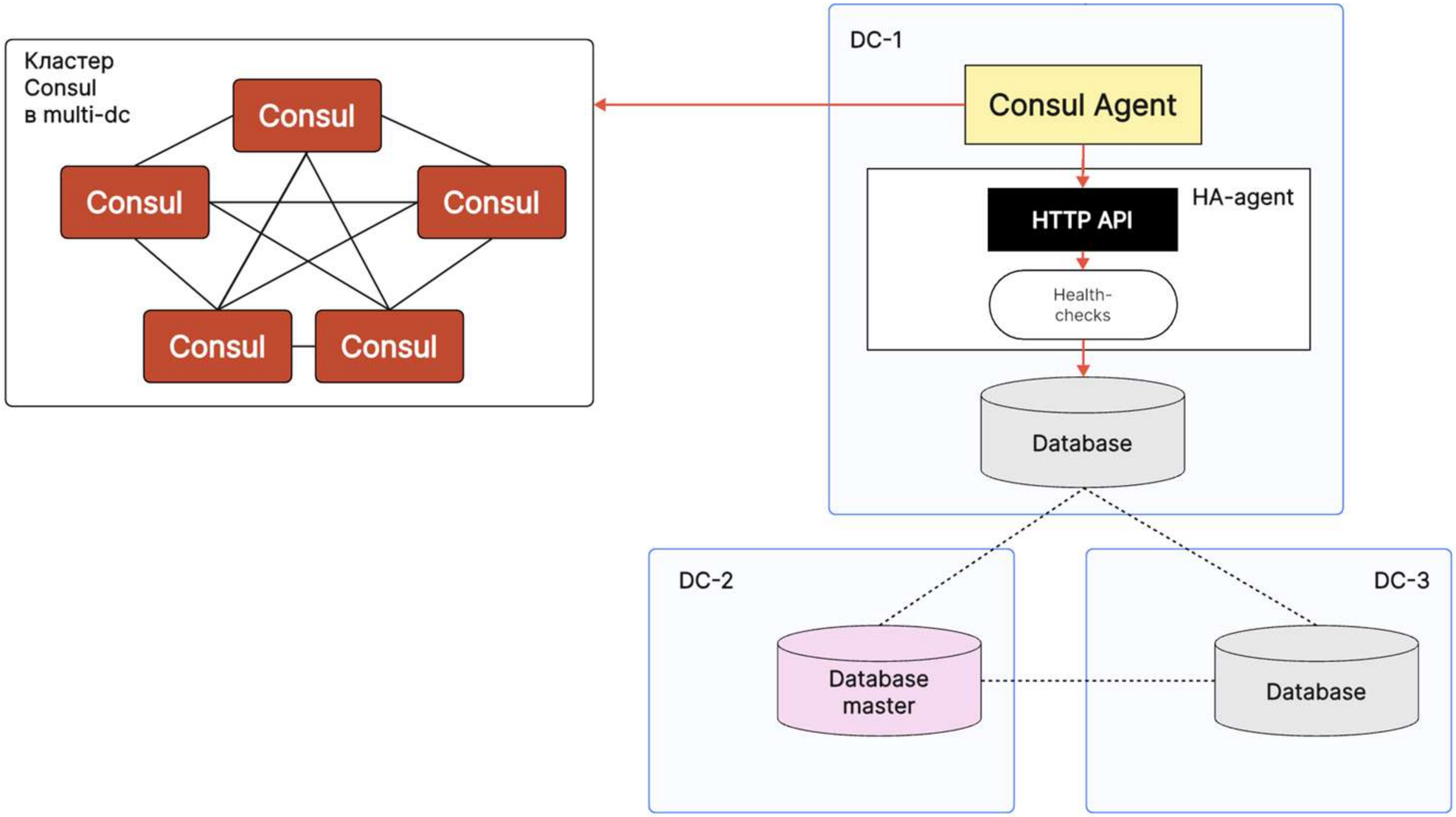
Database discovery



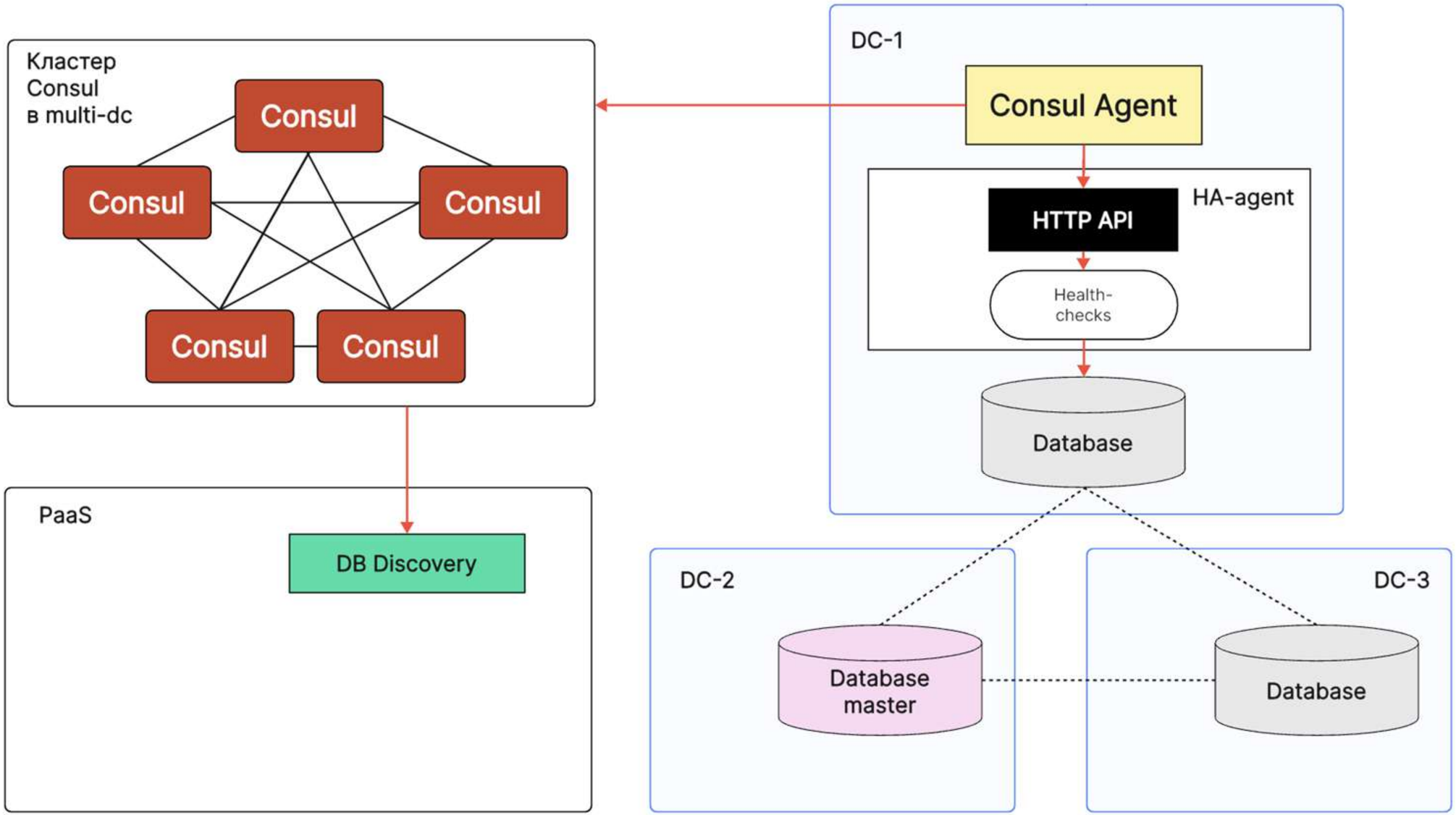
Database discovery



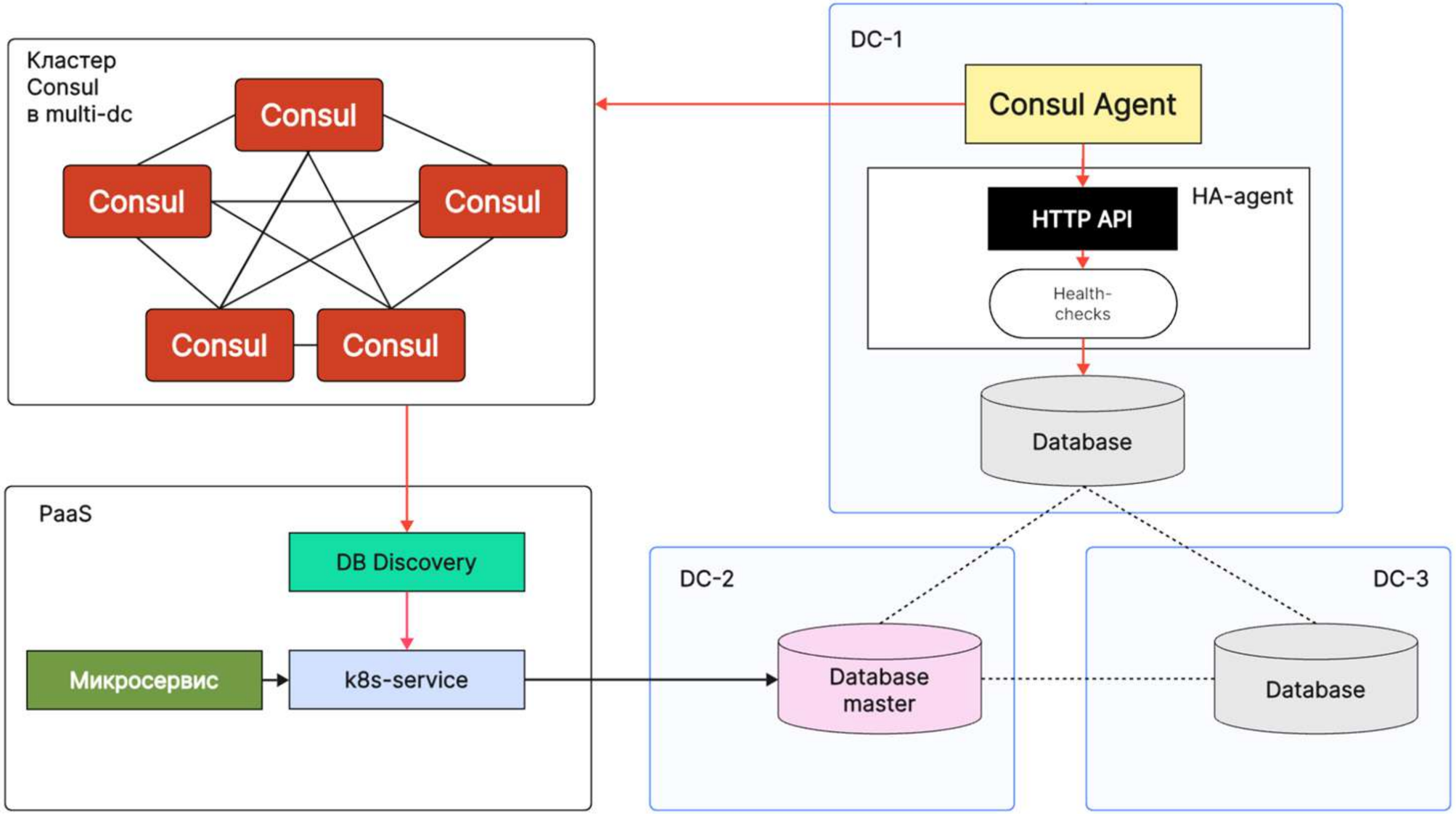
Database discovery



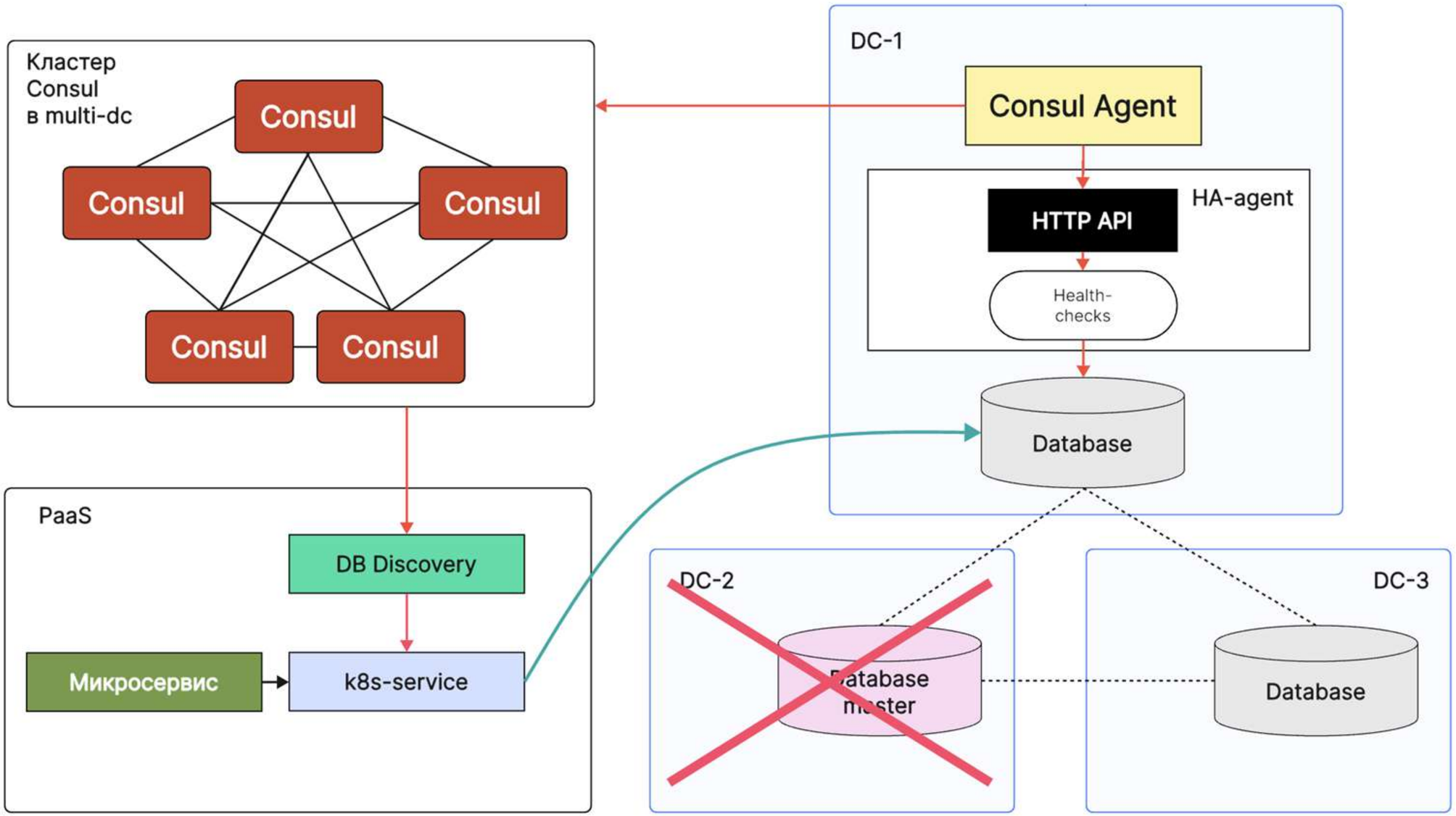
Database discovery



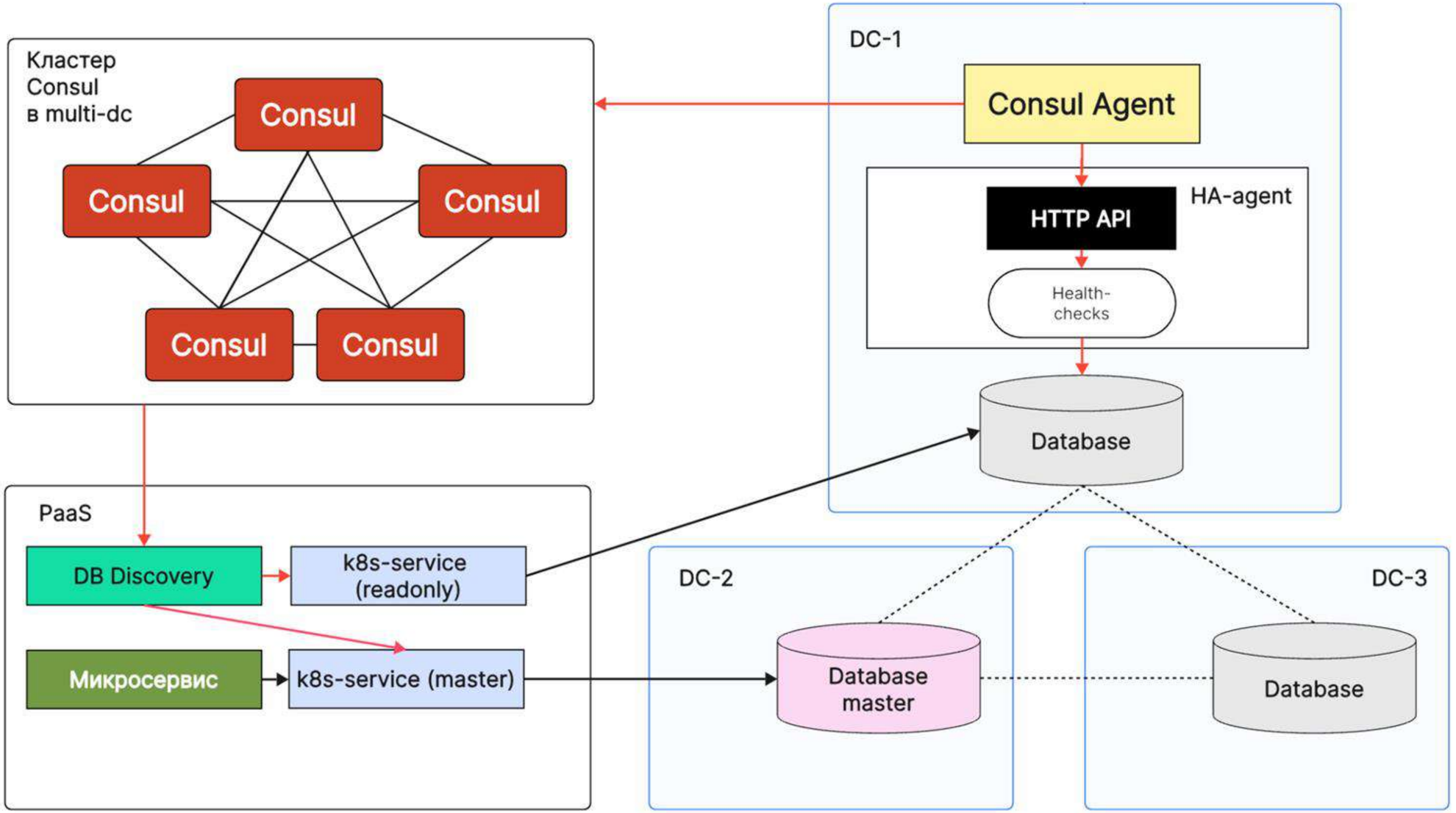
Database discovery



Database discovery: падение DC



Database discovery: доступ к репликам



Итоги

- Мы рассмотрели основные требования к платформе DBaaS и паттерны, которые помогают эти требования удовлетворить
- На основе паттерна метаоператора проектируется Control Plane
- На основе паттерна оператора проктируется компонент для приведения БД к желаемому состоянию на основе информации из Control Plane
- Паттерн HA-агента используется для реализации механики выполнения периодических и событийных задач, а также для хелсчеков
- Пользователи всегда обладают актуальной информацией о текущих эндпойнтах базы данных

Дублирование — это круто: это надёжно, это просто

Жига Никита

Engineer @ DBaaS Avito

Дублируйте ваши данные

Дублируйте ваши датацентры

Дублируйте вашу инфраструктуру

Откатывай красиво

Дебажся быстро

Программируй смело

Проектируй ловко

@nikita0873

Полина Кудрявцева

Engineer @ SQL Avito