

**Yandex Cloud**

# Как клиенты облака используют Yandex Database Serverless

**Антон Коваленко**

Технический руководитель проектов

# Содержание

01 Кратко про Yandex Database

02 Сценарии пользователей

03 Serverless БД — это сложно?

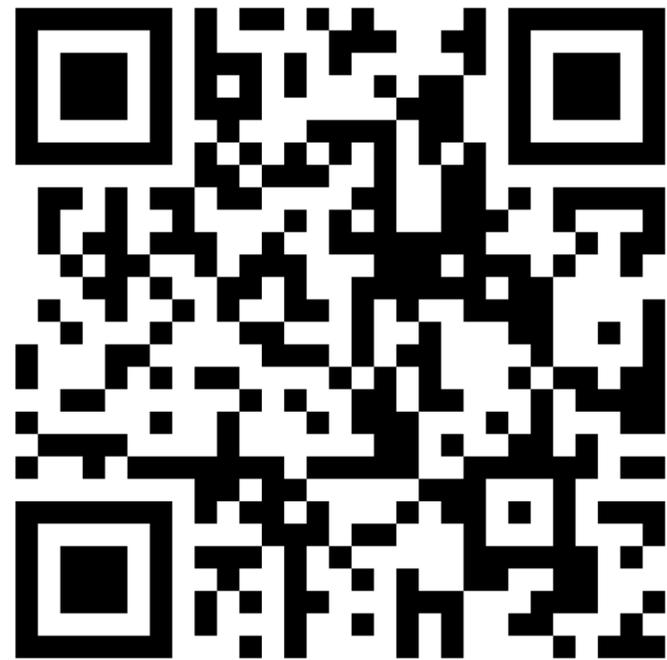
04 Совместимость с DynamoDB

05 Сколько стоит?

01

Yandex Database — что это?

# Отвечим на любой ваш вопрос про YDB в чате



[t.me/yandexdatabase\\_ru](https://t.me/yandexdatabase_ru)

# YDB — Cloud Native DistributedSQL Database



- Yandex Database — распределённая база данных
- › Разработана в Яндексе
- › Надежное хранение данных с автоматической репликацией
- › Горизонтальная масштабируемость на десятки тысяч серверов
- › Автоматическое восстановление после сбоев
- › Механизм распределенных ACID-транзакций со строгой консистентностью
- › Высокая пропускная способность при малом времени отклика
- › Диалект SQL (YQL)
- › Совместимость с AWS DynamoDB (в режиме Serverless)

# Yandex Database популярен внутри Яндекса

 auto.ru

Яндекс  Алиса



Яндекс  Турбо-страницы

Яндекс Репетитор

Яндекс Новости

Яндекс  Директ

Яндекс  Дзен

Яндекс  Погода

Яндекс  Толока

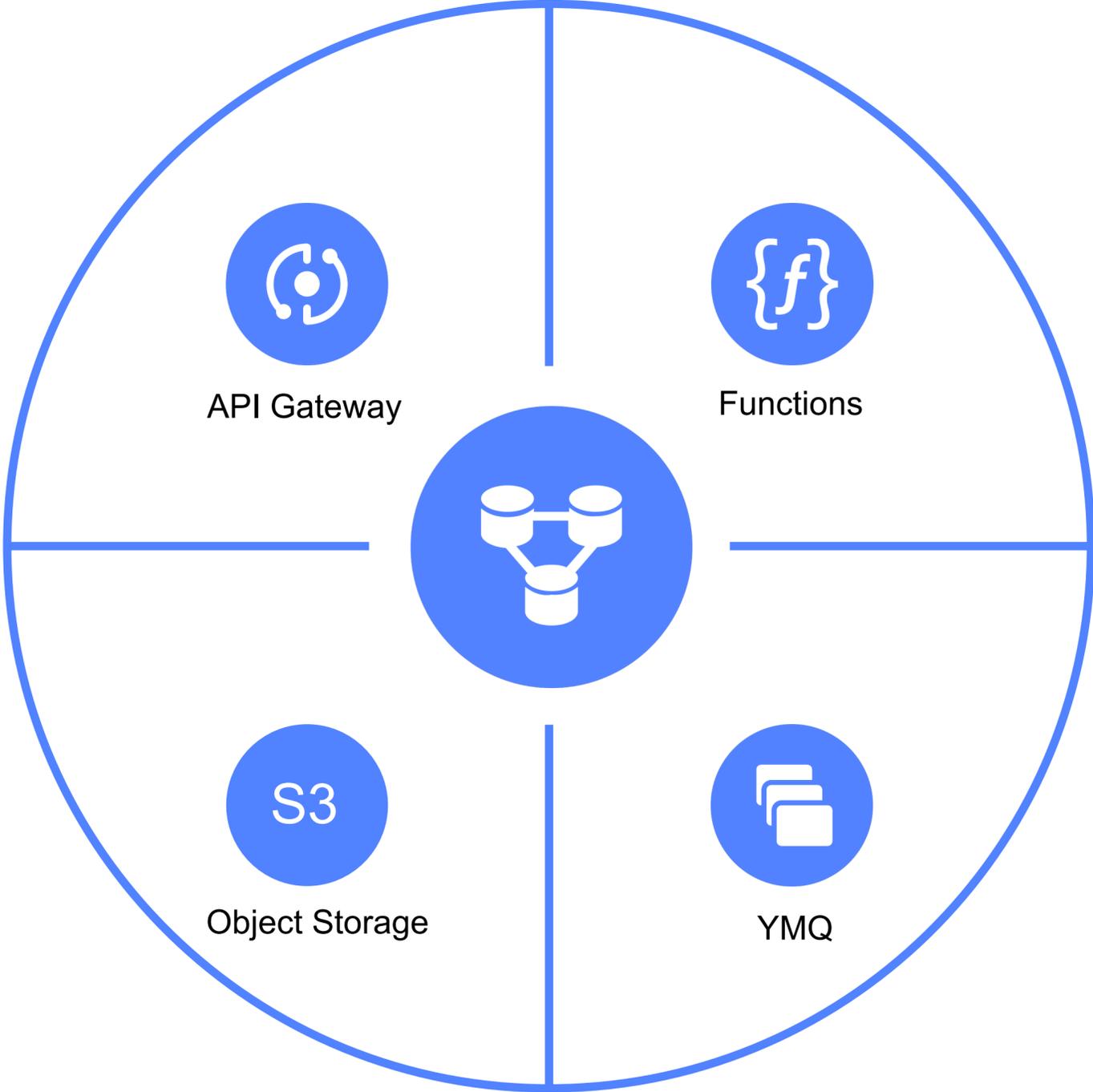
Яндекс  Такси

Яндекс  Услуги

**КиноПоиск**

Яндекс Паспорт

# Serverless-платформа — недостающая база



# YDB Serverless для пользователя



## Оплата за использование

- › Нет нагрузки — нет счетов за выделенные ресурсы

## Быстрое масштабирование

- › Не нужно думать о выделенных ресурсах

## Эффективность при малых нагрузках

- › Например, 1 запрос в минуту

## Совместимость с DynamoDB

- › Позволяет запускать приложение в разных облаках

02

# Истории пользователей

# Пользователь Serverless



## Стартап или эксперимент

- › Времени на раскачку нет
- › Небольшая кросс-функциональная команда
- › Нет выделенных админов

## Нет истории

- › Невозможно прогнозировать нагрузку на год

## Имеет опыт работы с другими облаками

- › Совместимый API ускорит разработку

# Serverless Database — ЧТО ХОЧЕТ ПОЛЬЗОВАТЕЛЬ



- | **Канарейку за копейку, чтобы пела и не ела**
  - › Привычную базу данных
  - › Чтобы всегда работала
  - › Не заниматься настройкой
- | **Платить как можно меньше**
  - › С оплатой за использование
  - › Free Tier

# Cloud Advisor — безопасность и оптимизация расходов



[cloudadvisor.ru](https://cloudadvisor.ru)

- | **CloudAdvisor выполняет**
  - › Аудит безопасности облачной инфраструктуры
  - › Оптимизацию расходов

# Cloud Advisor — вариант с выделенными ресурсами



## Запустим приложение в VM

- › Нужна группа виртуальных машин
- › Нужен балансер
- › Нужно обновлять версии ПО на VM

## Запустим управляемую БД

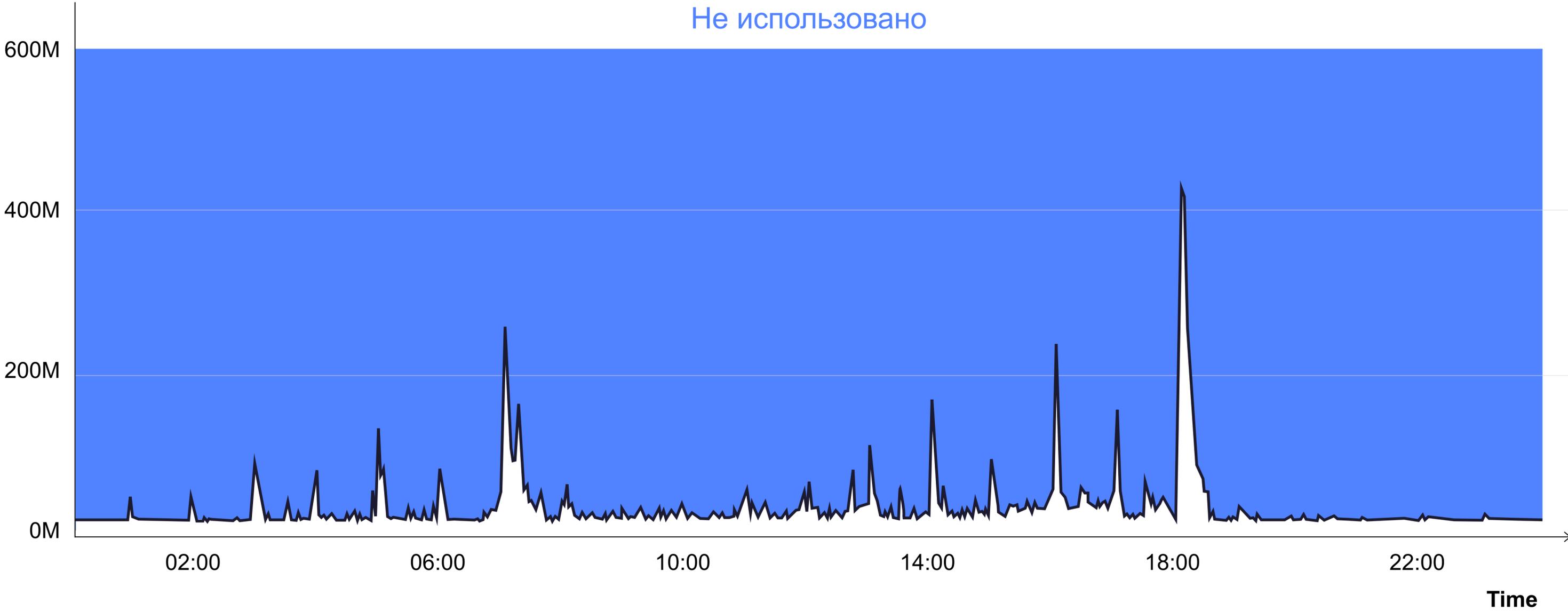
- › Нужно планировать ресурсы

# Cloud Advisor — сложности

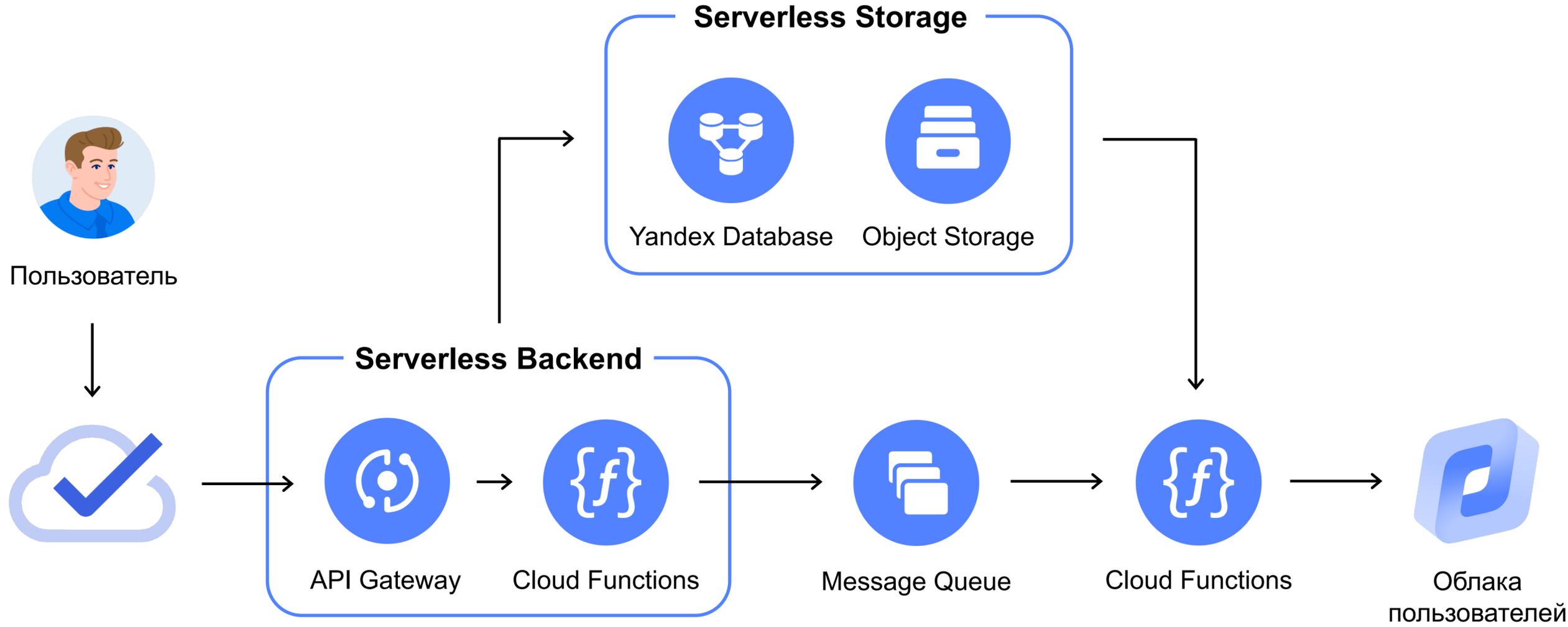


- | **На старте проект сложно предсказать нагрузку**
  - › Нет исторической нагрузки
- | **Нужен запас от планируемого пика**
  - › И всё равно можно упереться в ресурсы
- | **Давайте попробуем `serverless`**
  - › Чтобы упростить жизнь и оценить нагрузку

# Cloud Advisor — сложности



# Cloud Advisor — что получилось



# Cloud Advisor — какие боли были устранены



**Не обязательно планировать мощности заранее**

› Выбранные сервисы быстро масштабируются

**Оплата за потребление**

› Расходы пропорциональны нагрузке

**Не требует администрирования**

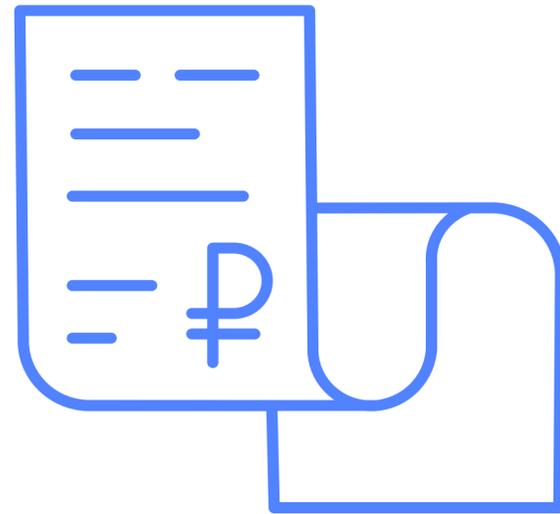
› Быстрее запустим новые возможности

› Больше времени на развитие продукта

**Совместимость с AWS DynamoDB**

› Позволяет запускать сервис в другом Облаке

# Анализ цен на маркетплейсах



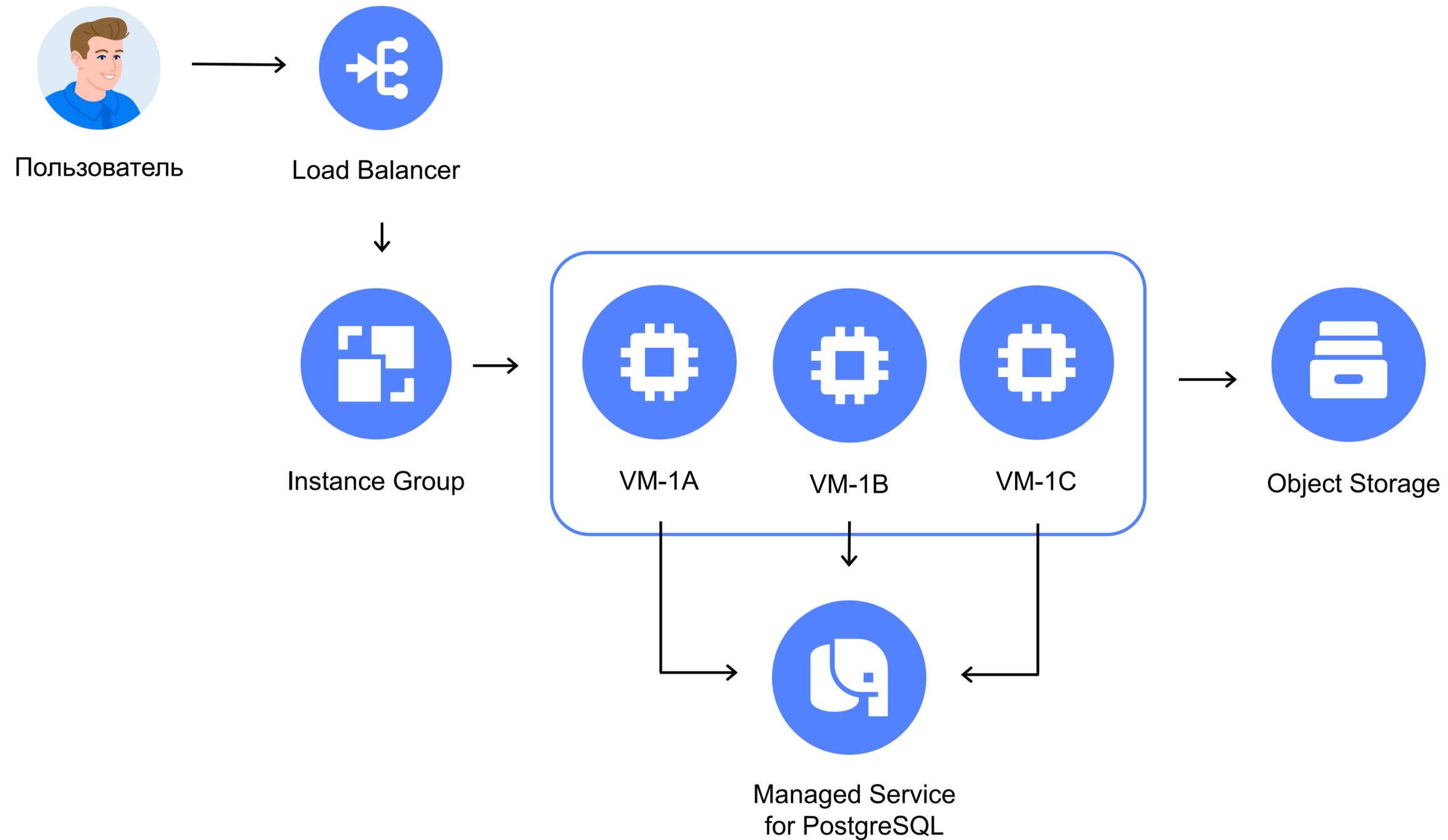
## Пользователи сервиса платят за вызов

- › Цена зависит от количества SKU

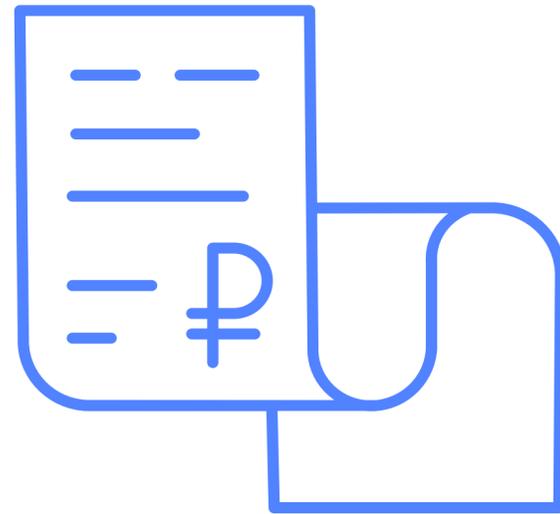
## Отчёты о ценах конкурентов

- › Подписка на сигналы
- › Подписка на сводные отчёты

# Анализ цен — как можно сделать



# Анализ цен — проблемы

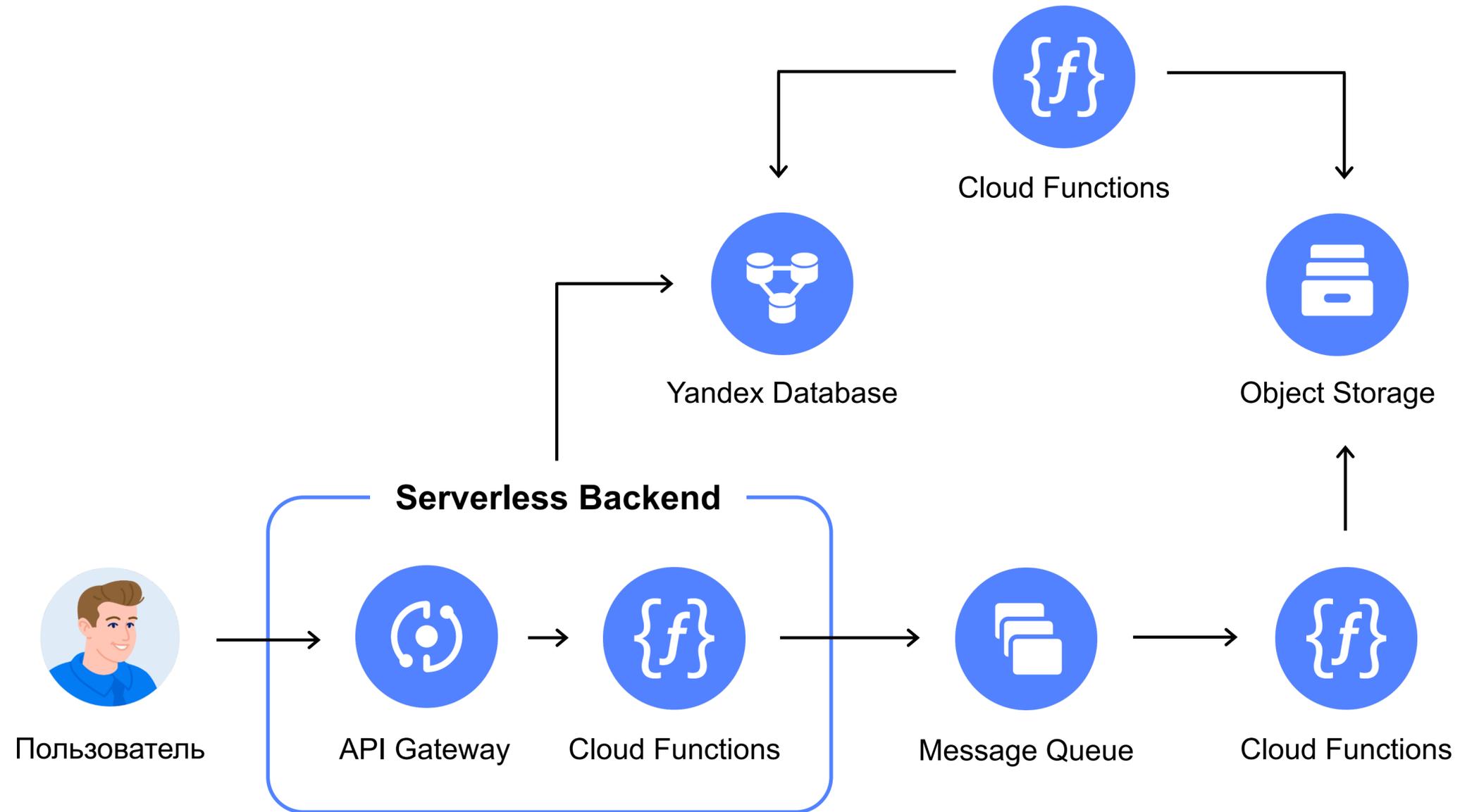


Один разработчик

Платим всегда

В случае успеха — нужно масштабироваться

# Анализ цен — serverless way



# Командор — мессенджер доверенных коммуникаций

Командор®

[komandor.app](https://komandor.app)

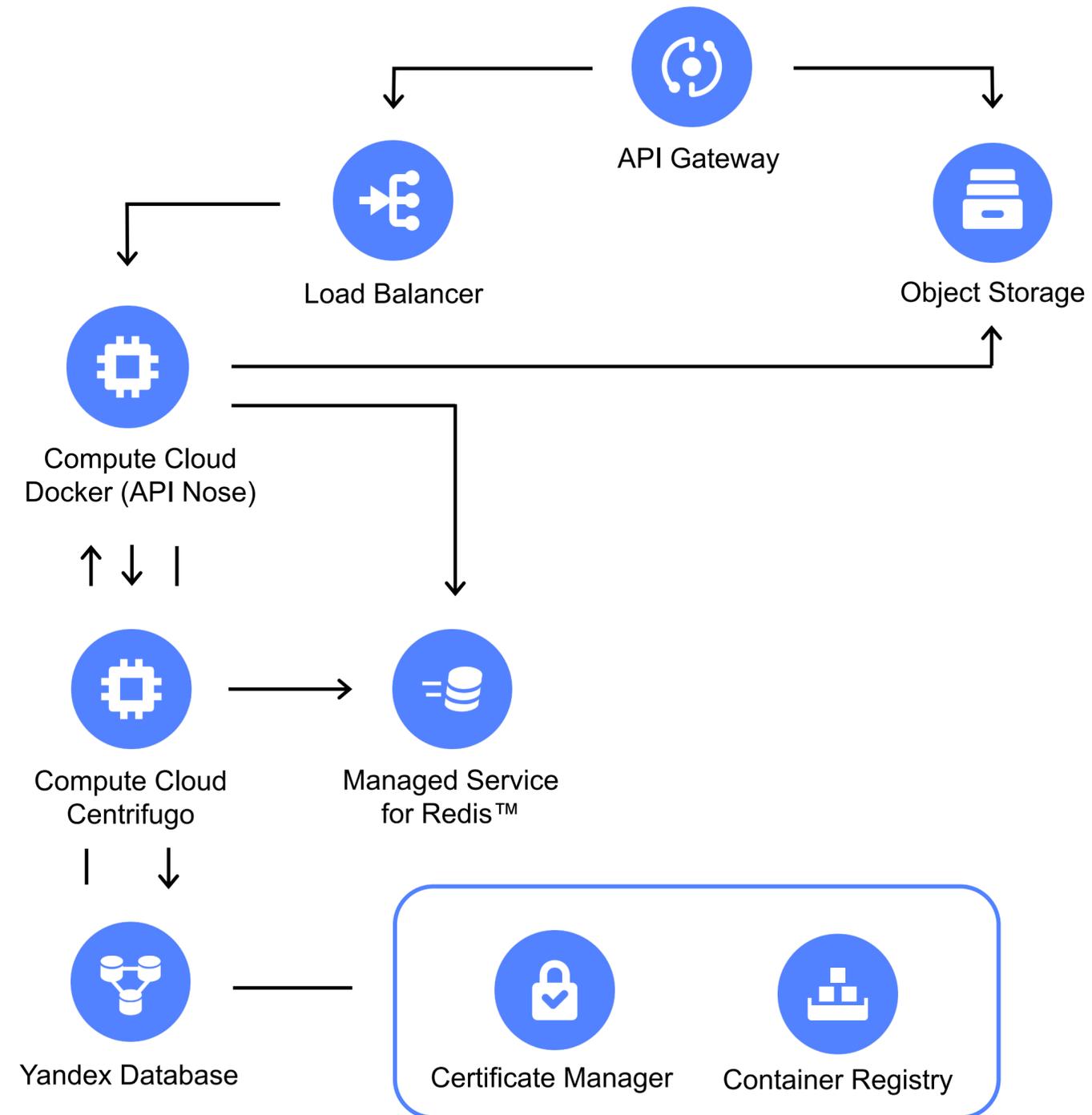
## Обмен конфиденциальными сообщениями

- › Шифрование на устройстве пользователя
- › Соответствует российским стандартам
- › Надёжность хранения — ключевое преимущество

## Электронный документооборот в мессенджере

- › Можно обмениваться сообщениями и документами, подписанными ЭП (КЭП, НКЭП)

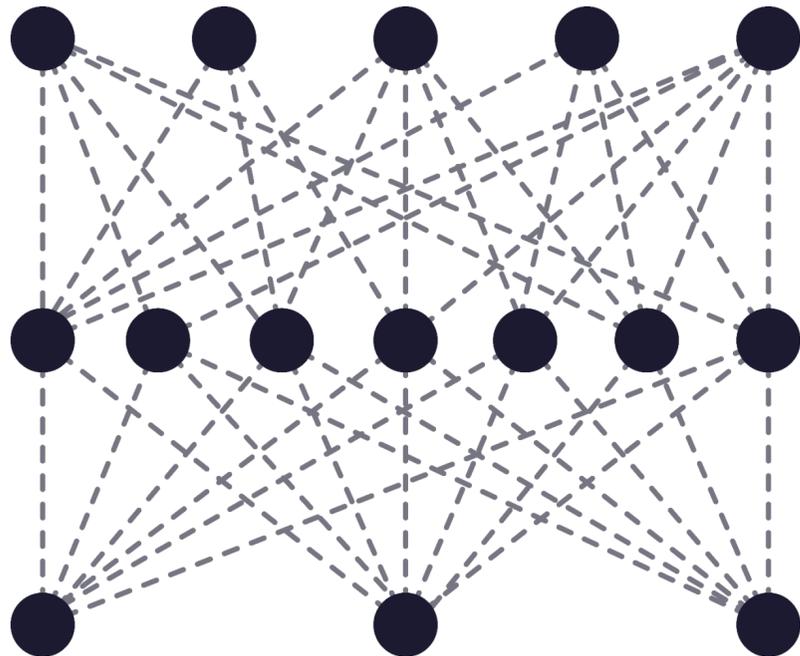
# На старте выбрали Yandex Database



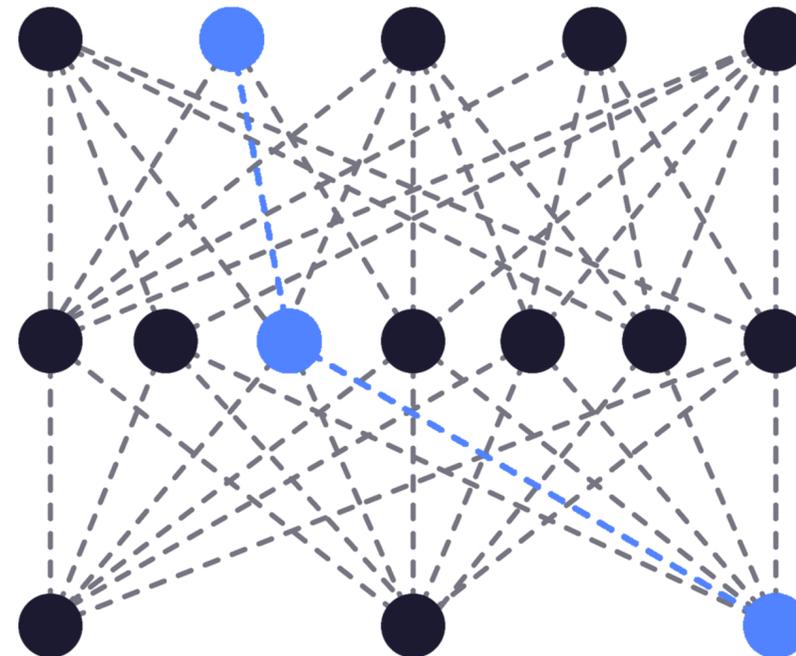
# Хранение и анализ трейсов

Что случилось с моим запросом?

Без распределённой трассировки



С распределённой трассировкой



- Service-to-Service Connection
- Individual Request Path

# Писать всё



- › По любой `trace_id` получить трассу (ни одна ошибка не уйдёт обиженной)
- › Для этого — писать трассы на все запросы (убрать `sampling` — пролив части трасс)
- › ~300K `spans/s`, исходя из `prod`-нагрузки

# Требования к базе



- › Региональная
- › Отказоустойчивая
- › Горизонтально масштабируемая
- › С высокой пропускной способностью
- › С малым временем отклика

# Jaeger tracing



## Особенности

- › Open Source
- › Golang
- › Плагины

## Компоненты

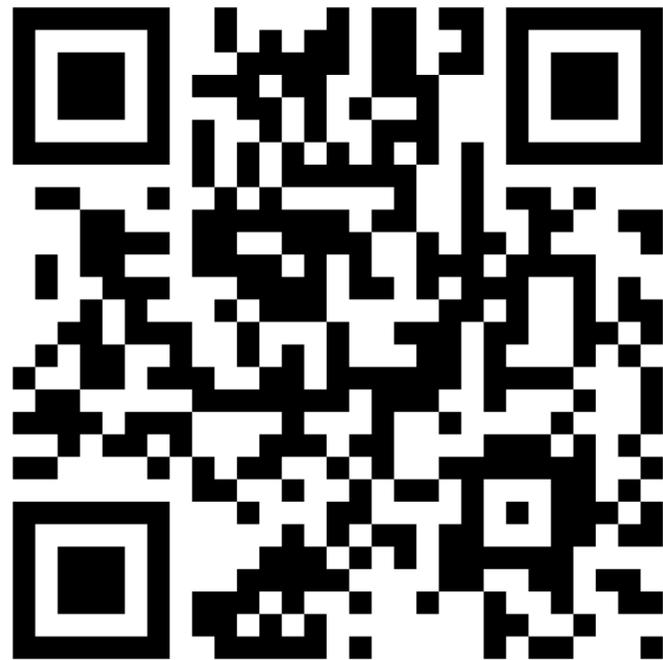
- › Agent (sidecar)
- › Collector
- › Query (UI)
- › DB-основные: Cassandra, ElasticSearch

# Плагин для Yandex Database



- › Spans
- › Index table by service
- › Index table by key-value tags
- › Index table by request duration

# Yandex Database Serverless + Jaeger



[clck.ru/Usgku](https://clck.ru/Usgku)

- › Тестировали с помощью **tracegen**
- › **~0,6 request units/span** при **~8K span/s**
- › Эффективно использовать при нагрузке не более **668 span/s**
- › Дальше dedicated-кластер из 1 машины получается дешевле при пересчёте на **₽/span**

Рецепты в репозитории плагина

03

Почему serverless БД – сложно?

# Как обеспечить характеристики



- | **Оплата за использование**
  - › Нет счетов за выделенные ресурсы
- | **Быстрое масштабирование**
  - › Не нужно думать о выделенных ресурсах
- | **Эффективность при малых нагрузках**
  - › Например, 1 запрос в минуту
- | **Быстрый старт функции**
  - › Соединение с БД бывает долгим

# Оплата за использование



## **ДунатоDB — из чего складывается стоимость?**

- › Количество прочитанных/записанных байт
- › Количество прочитанных/записанных строк

## **Почему байт/строк недостаточно?**

- › Через YQL можно выполнить затратный по CPU запрос без доступа к данным

## **YDB Serverless — из чего складывается стоимость?**

- › Прочитанные/записанные байты/строк
- › Потраченный ресурс CPU

# Оплата за использование – решение



## Доработали Yandex Database

- › Рассчитываем стоимость запроса внутри системы
- › Возвращаем стоимость вместе с результатами запроса

# Эволюция Yandex Database



## Один проект — один кластер

- › Обслуживать 7-10 кластеров — команда страдает
- › Проектов, которым нужен свой кластер — единицы

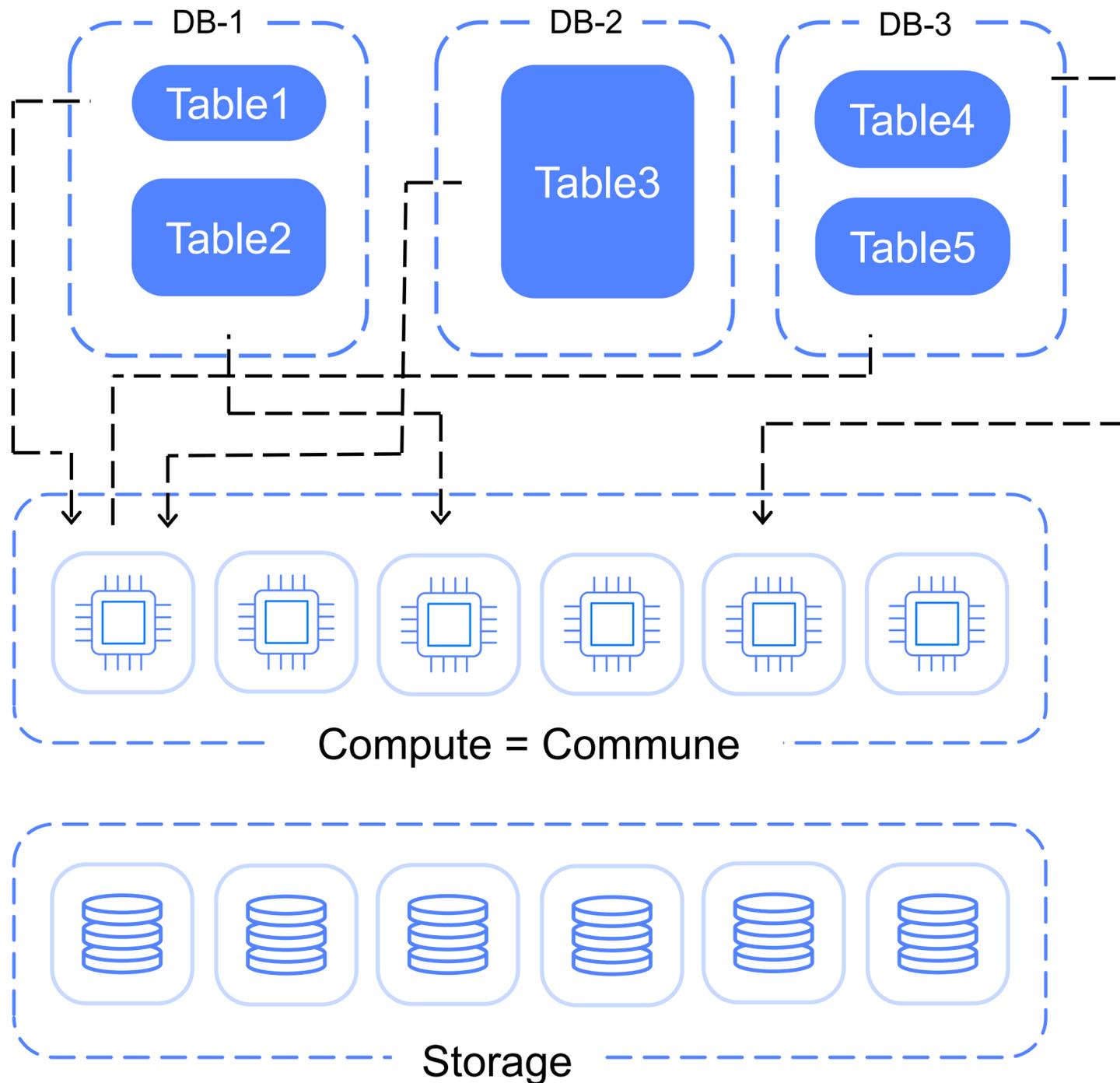
## Мультитенатные кластеры

- › Для каждой базы — отдельная группа VM
- › Слой хранения — распределённый, доступен по сети

## Коммуны YDB

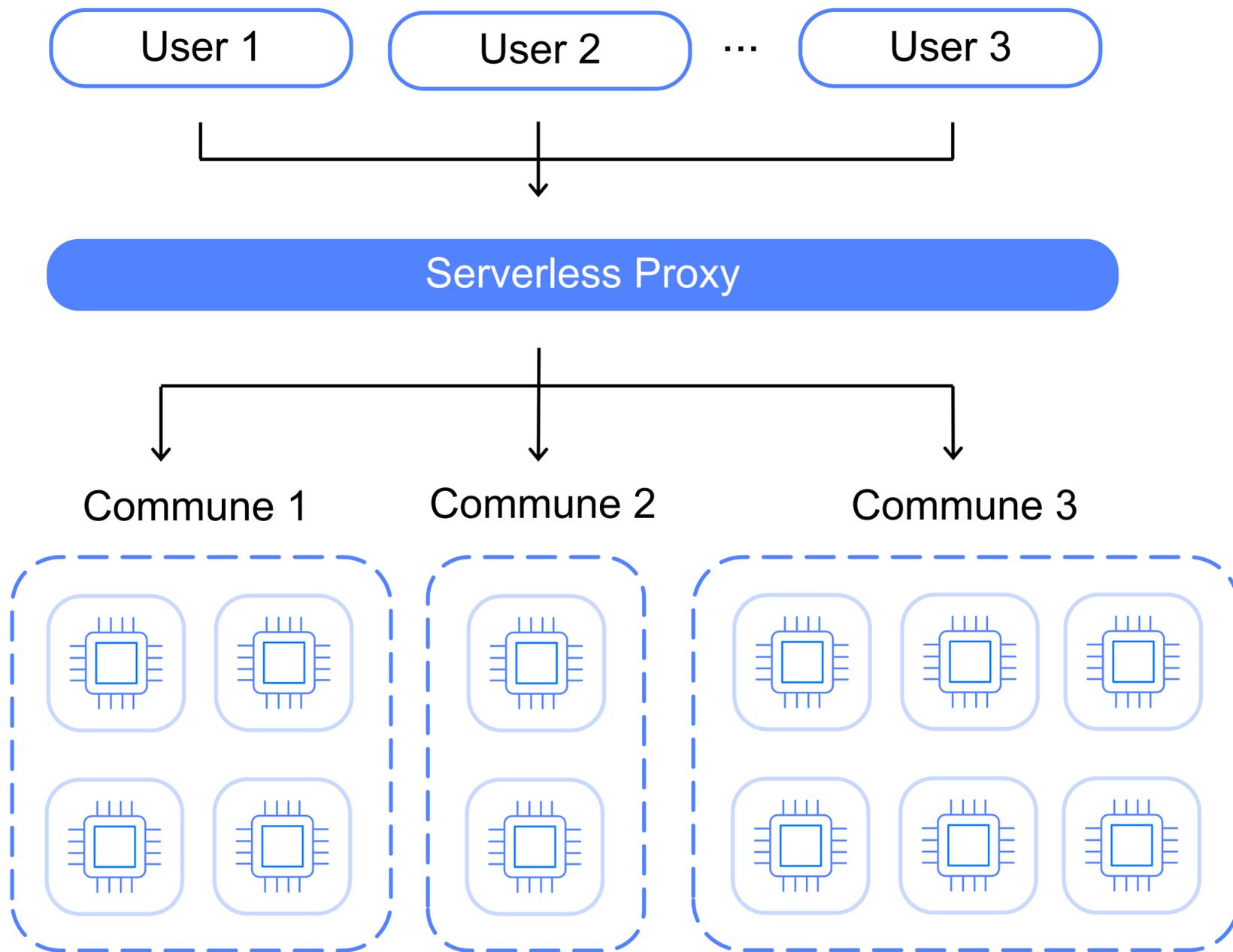
- › Новый объект схемы — YDB Serverless
- › Выделенные ресурсы разделены между базами

# Коммуна для размещения Serverless БД



- › Коммуна в YDB — набор Compute и Storage ресурсов, где «живут» Serverless БД
- › Запрос к любой из Serverless может быть обработан на любой Compute ноде
- › Storage у баз данных — общий
- › Механизмы автоматического партиционирования таблиц YDB разделяют таблицы на примерно одинаковые партии незаметно для пользователя

# Serverless Proxy



Является точкой входа:  
[ydb.serverless.yandexcloud.net](https://ydb.serverless.yandexcloud.net)

- › Выполняет DB Discovery (определяет коммуну, в котором живет база)
- › Содержит пул соединений к БД
- › Реализует metering

04

# Совместимость с DynamoDB

# Что такое AWS DynamoDB



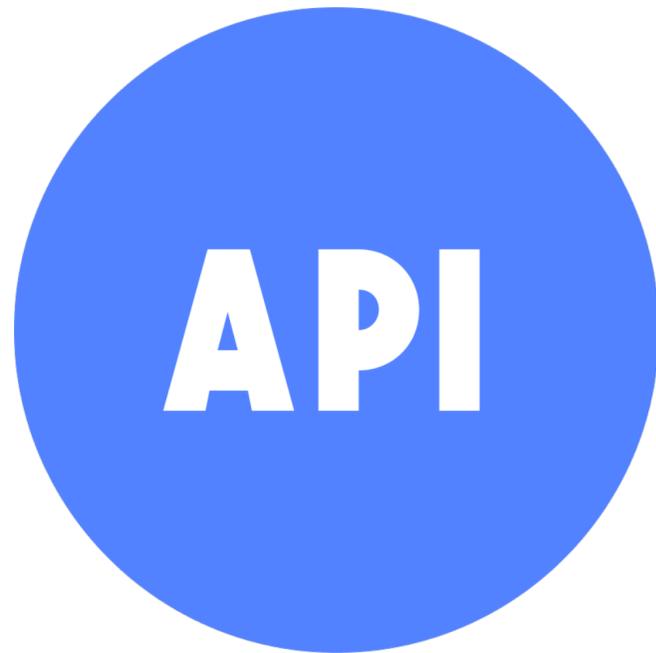
## Популярная cloud-native база данных

- › Основной сценарии — KV-доступ к документам
- › Доступна в облаке AWS с оплатой за использование
- › №1 в рейтинге [db-engines.com](https://db-engines.com) среди cloud native баз-данных

## Специальный DynamoDB API

- › Доступны SDK от AWS для многих языков

# Document API



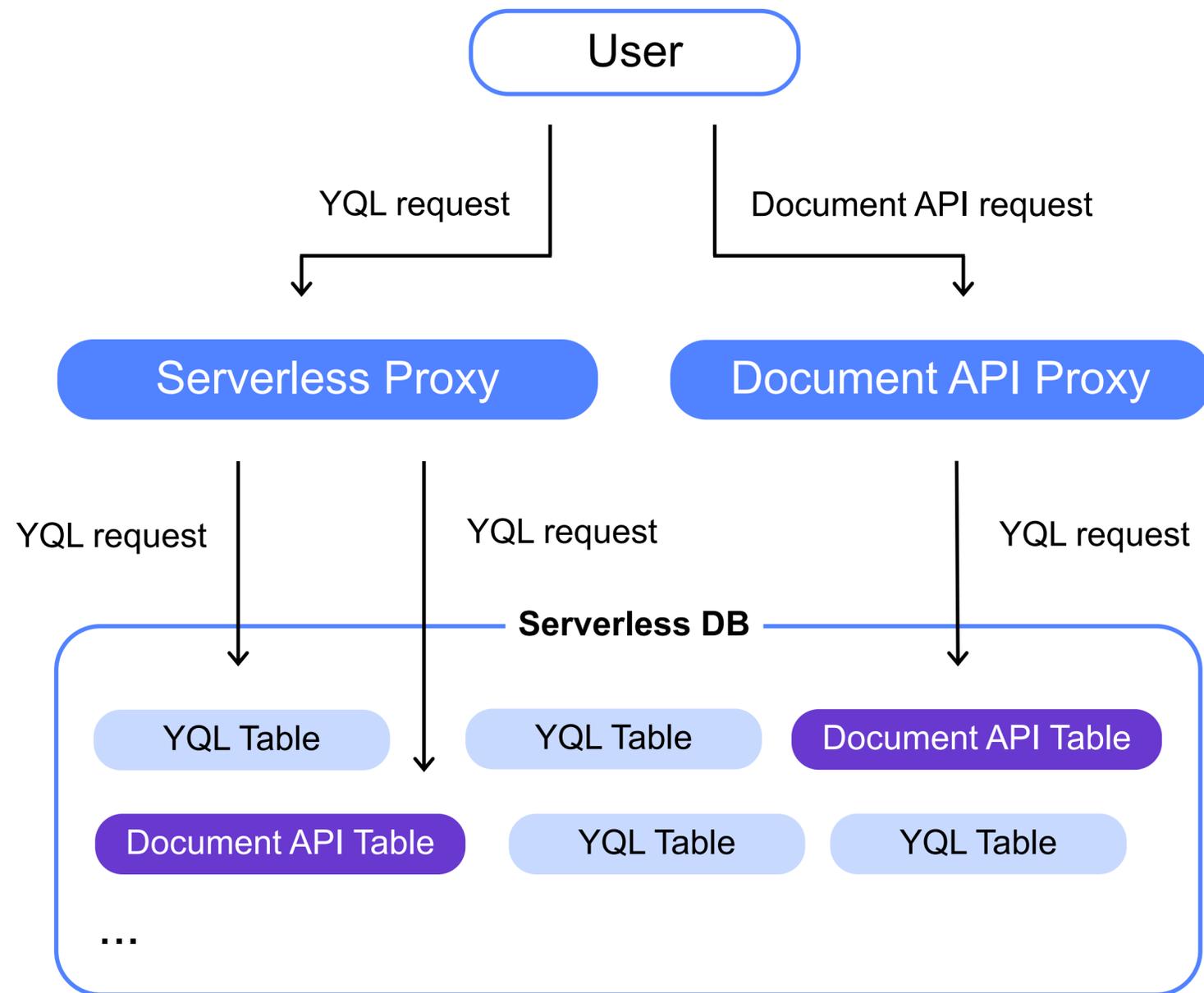
## Совместимость с Amazon DynamoDB

- › Основной сценарий — хранение KV-документов
- › Drop-in replacement для большинства приложений

## Реализован поверх YQL

- › Таблицы, созданные через YQL и Document API — суть одно и то же

# Document API Proxy



## Транслирует вызовы Document API в YQL

- › В остальном повторяет поведение Serverless Proxy (использует общий код)
- › Интероперабельность с Document API таблицами можно работать из YQL (через Serverless Proxy)

05

СКОЛЬКО СТОИТ YDB Serverless

# YDB Serverless: Document API



Вид запроса	Размер данных	Единиц запроса (RU)
Операция чтения строки	4 КБ	1
Операция чтения строки в рамках транзакции	4 КБ	2
Операция записи строки	1 КБ	2
Операция записи строки в рамках транзакции	1 КБ	4

**1 RU — это эталонная стоимость выполнения чтения строки таблицы объемом до 4КБ по первичному ключу**

# YDB Serverless: YQL API



YQL-запросы сильно более выразительны

$$\begin{aligned} \text{CostRows} = & \text{Max}(\text{ReadRows}, \text{ReadBytes}/4\text{KB}) \\ & + \text{Max}(\text{WriteRows}, \text{WriteBytes}/1\text{KB}) * 2 \\ & + \text{DeleteRows} \end{aligned}$$
$$\text{Cost} = \text{Max}(\text{CostRows}, \text{CPUConsumed}/\text{CPUReadRow})$$

# YDB Serverless: цены



Услуга	Цена с НДС
Операции с данными по требованию	13,36 ₽ за 1 млн RU
Хранение данных	13,41 ₽ за 1 ГБ в месяц
Выполнение резервной копии по требованию	0,34 ₽ за 1 ГБ
Восстановление данных из резервной копии	6,4 ₽ за 1 ГБ

# Мы подготовили информацию:

Рады вашим идеям:

<https://cloud.yandex.ru/features>



Приходите в наш телеграм-чат:

[t.me/yandexdatabase\\_ru](https://t.me/yandexdatabase_ru)



Twitter

[twitter.com/yandexdatabase](https://twitter.com/yandexdatabase)



Примеры кода на GitHub:

[github.com/yandex-cloud/examples/tree/master/serverless](https://github.com/yandex-cloud/examples/tree/master/serverless)

Документация:

[cloud.yandex.ru/docs/ydb/](https://cloud.yandex.ru/docs/ydb/)

Хаб про Serverless:

[habr.com/ru/hub/serverless/](https://habr.com/ru/hub/serverless/)

Вопросы?



**Yandex Cloud**

**Спасибо!**

**Антон Коваленко**

Технический руководитель проектов