# Мастер-класс по инструментам для gRPC API

**Сергей Шайкин**

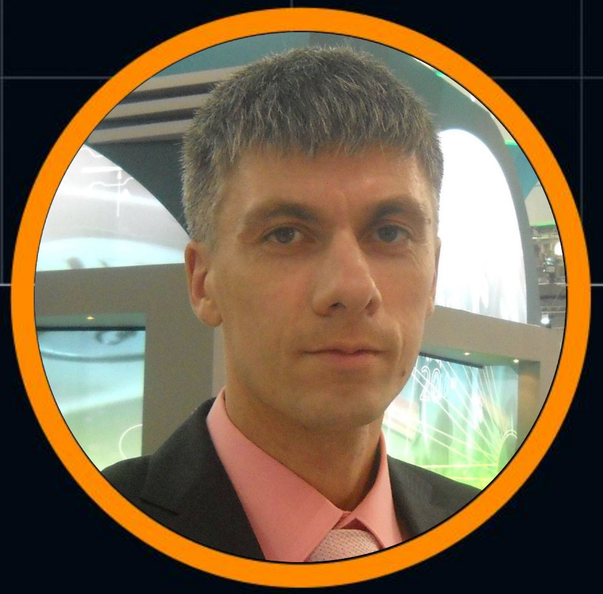
Electrolux AB

@ShaikinSergei

shaikins@yahoo.com

HEISENBUG

Electrolux Group



**Сергей Шайкин**

@shaikin

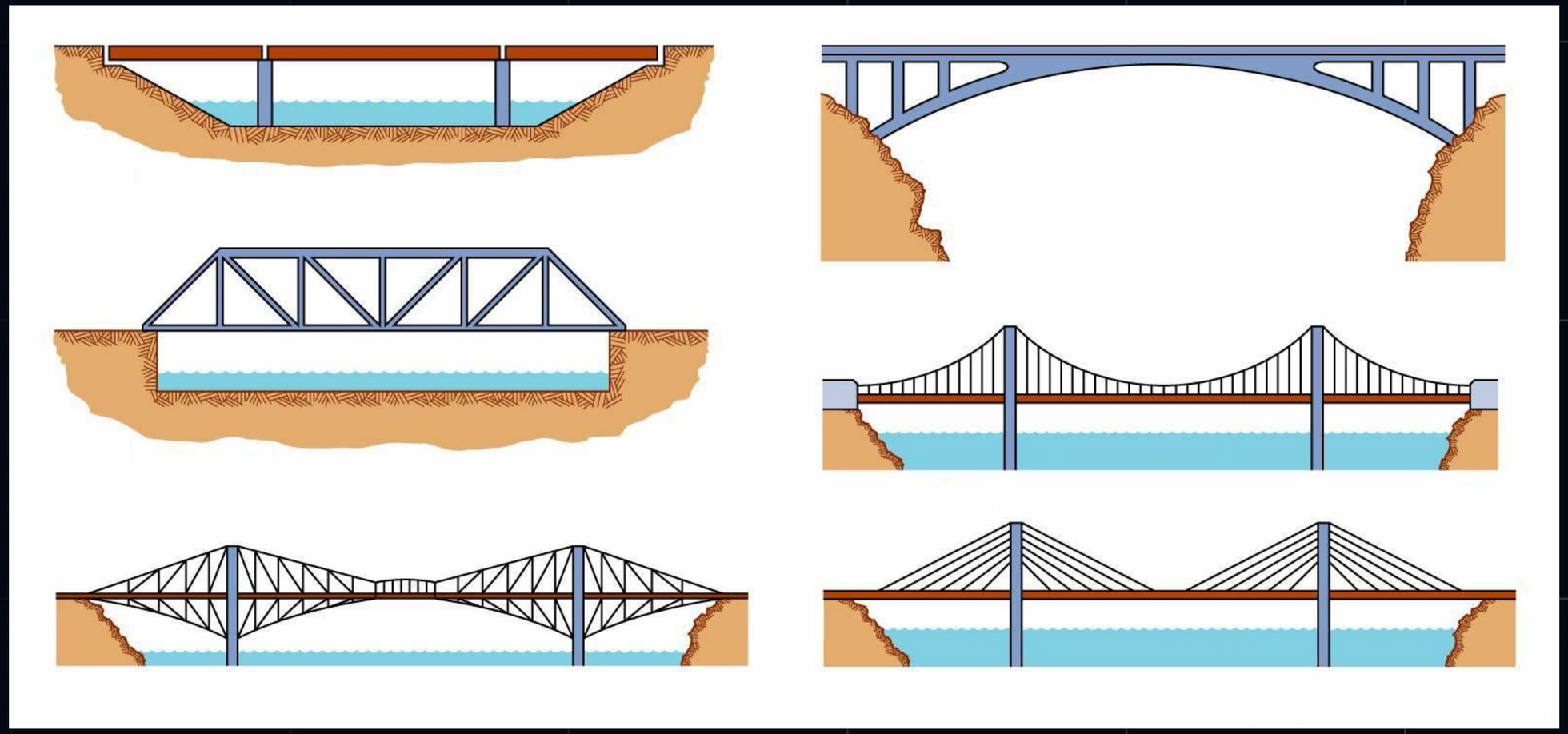
# Bio

- В тестировании более 8 лет.
- Веб, мобильное тестировании и тестировании бэкенда.
- В настоящее время тестирование The Internet of Things (IoT)

# Зачем?

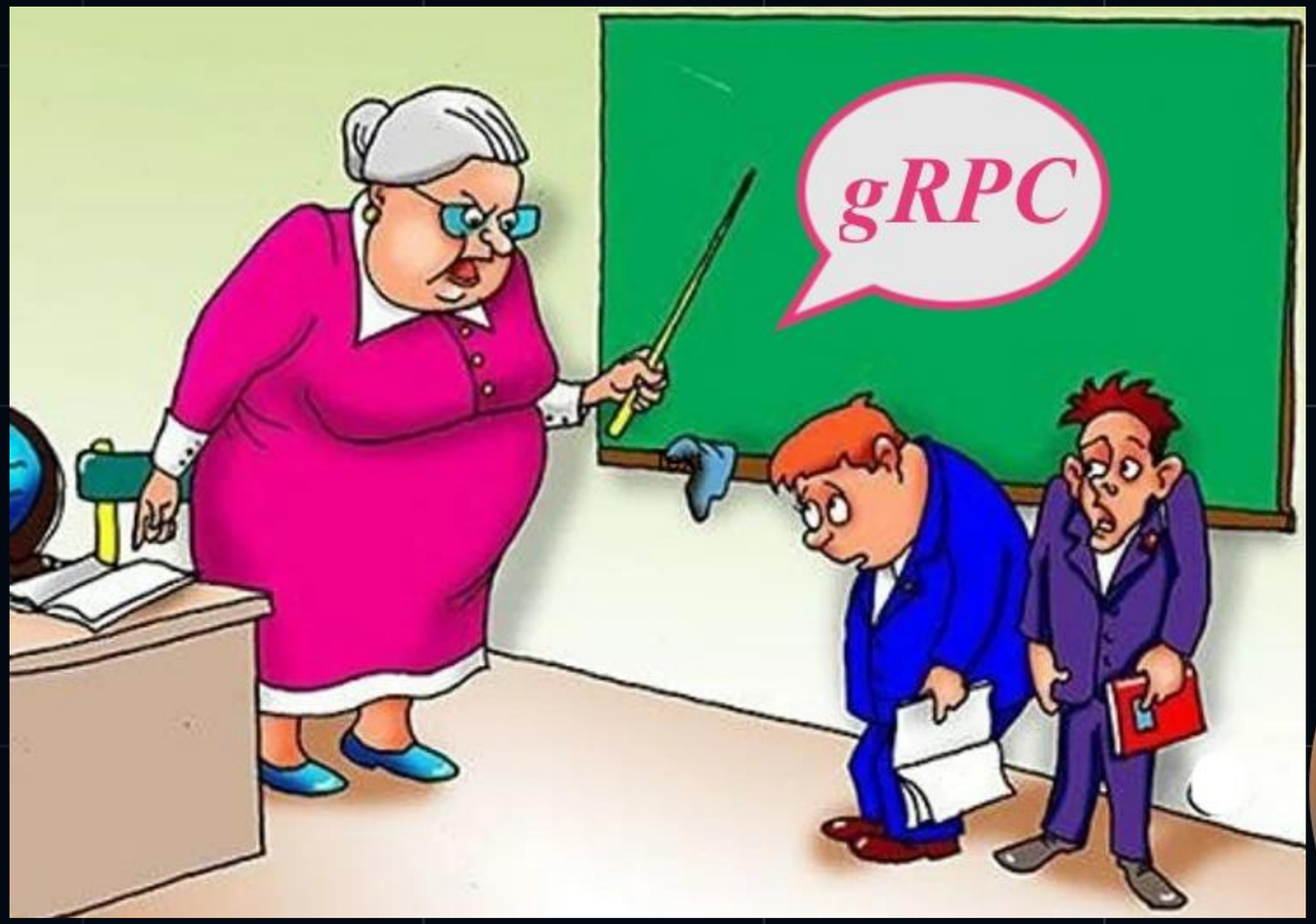
# Что сегодня будет?

- Архитектурные стили API
-



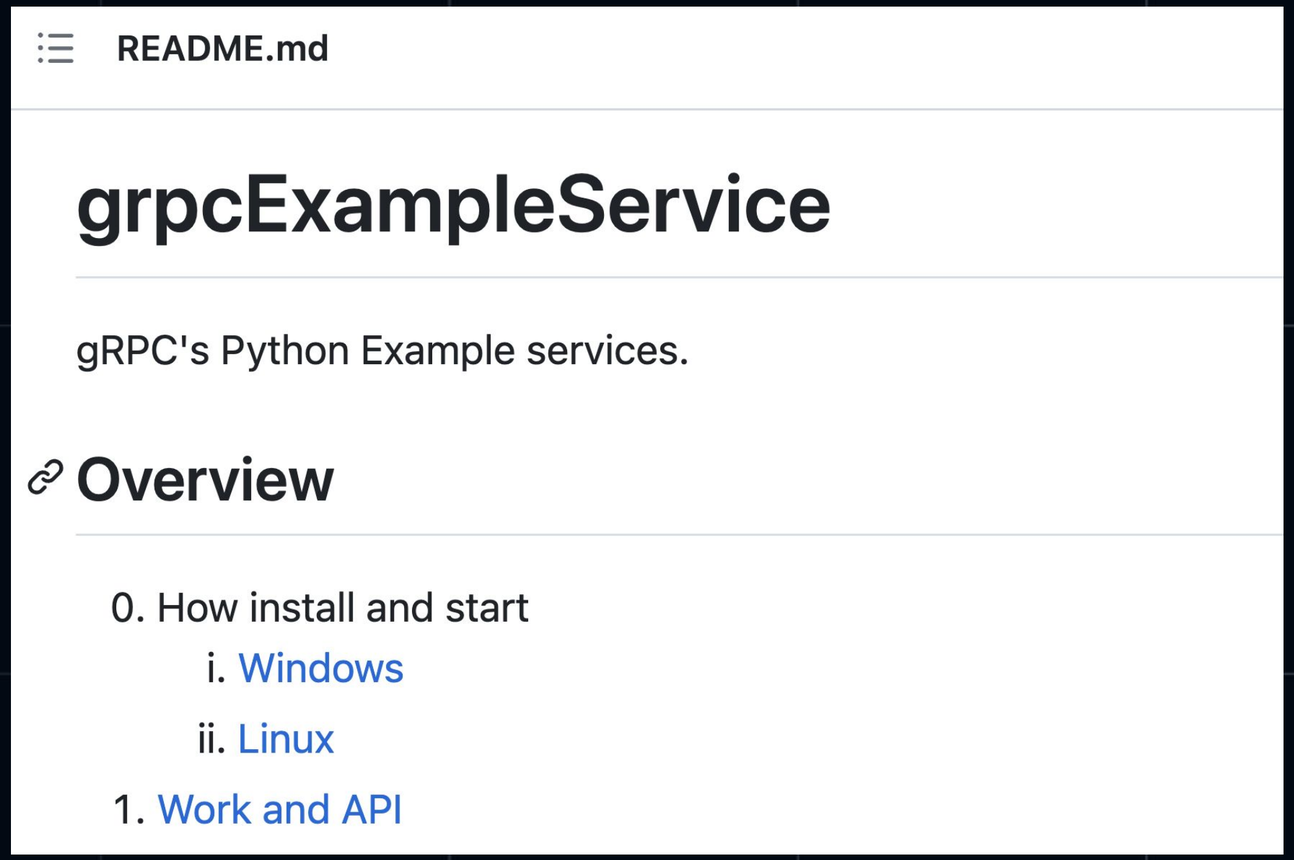
# Что сегодня будет?

- Архитектурные стили API
- Знакомство с gRPC
-



# Что сегодня будет?

- Архитектурные стили API
- Знакомство с gRPC
- Демо проект

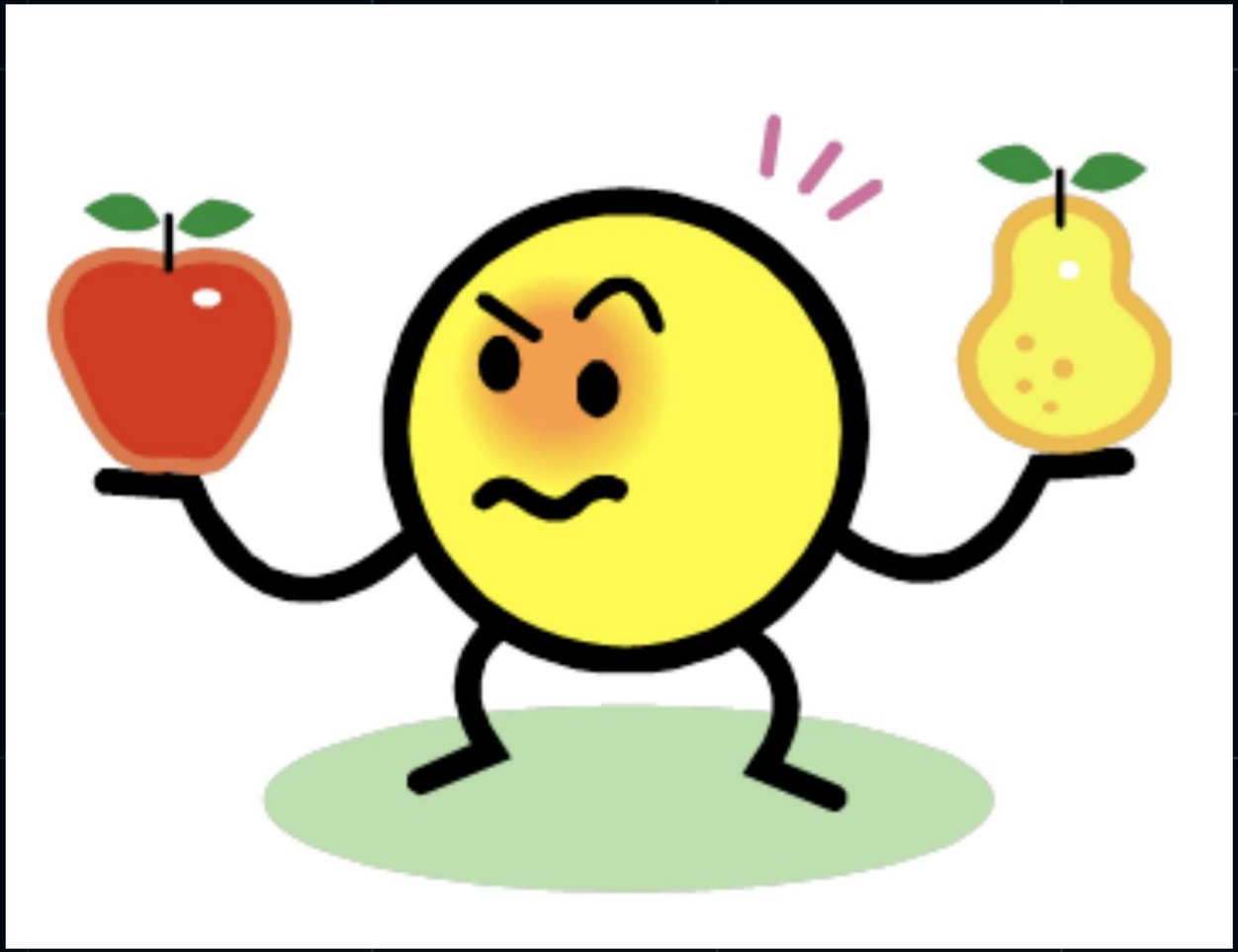


# Что сегодня будет?

- Архитектурные стили API
- Знакомство с gRPC
- Демо проект
- Инструменты

# Что сегодня будет?

- Архитектурные стили API
- Знакомство с gRPC
- Демо проект
- Инструменты
- Сравнение



# Что сегодня будет?

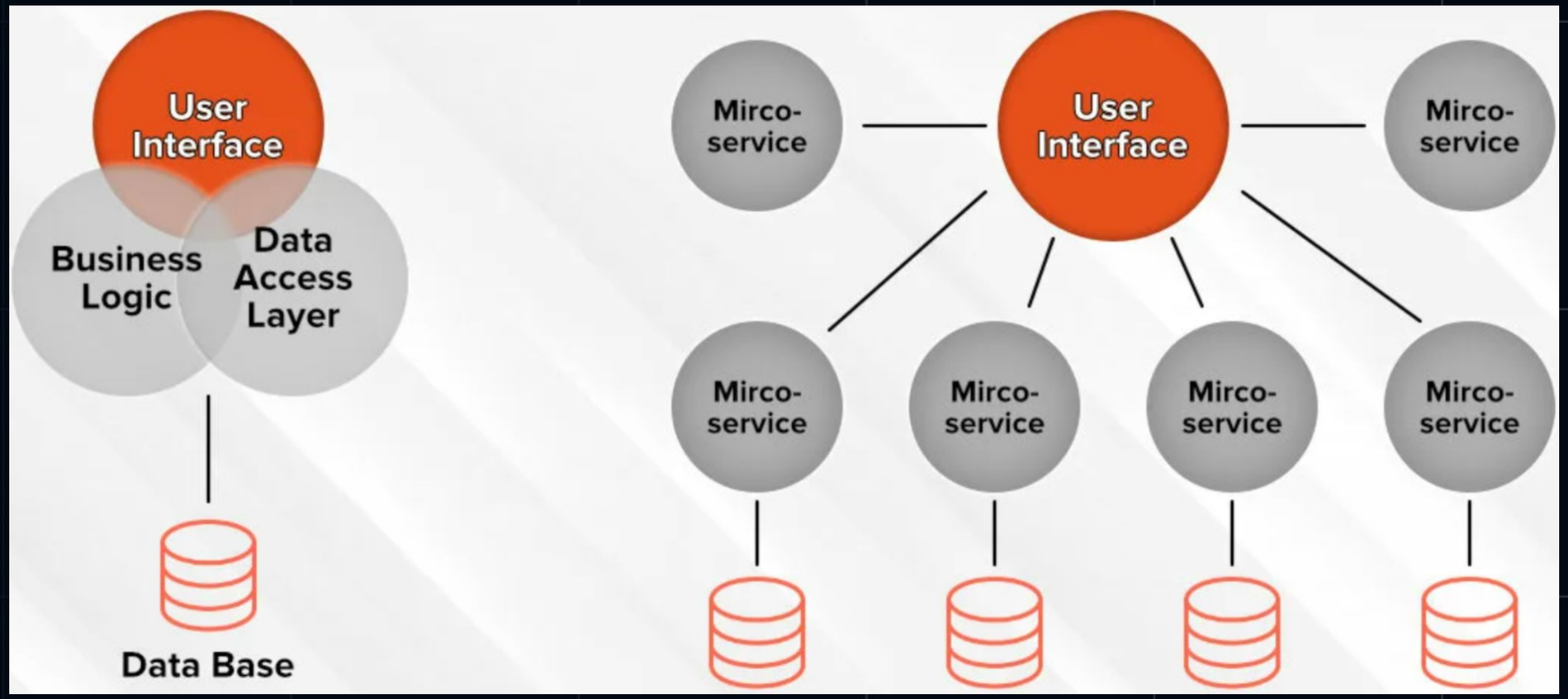
- Архитектурные стили API
- Знакомство с gRPC
- Демо проект
- Инструменты
- Сравнение
- Результаты

# Что сегодня будет?

- Архитектурные стили API
- Знакомство с gRPC
- Демо проект
- Инструменты
- Сравнение
- Результаты
- Обсуждение

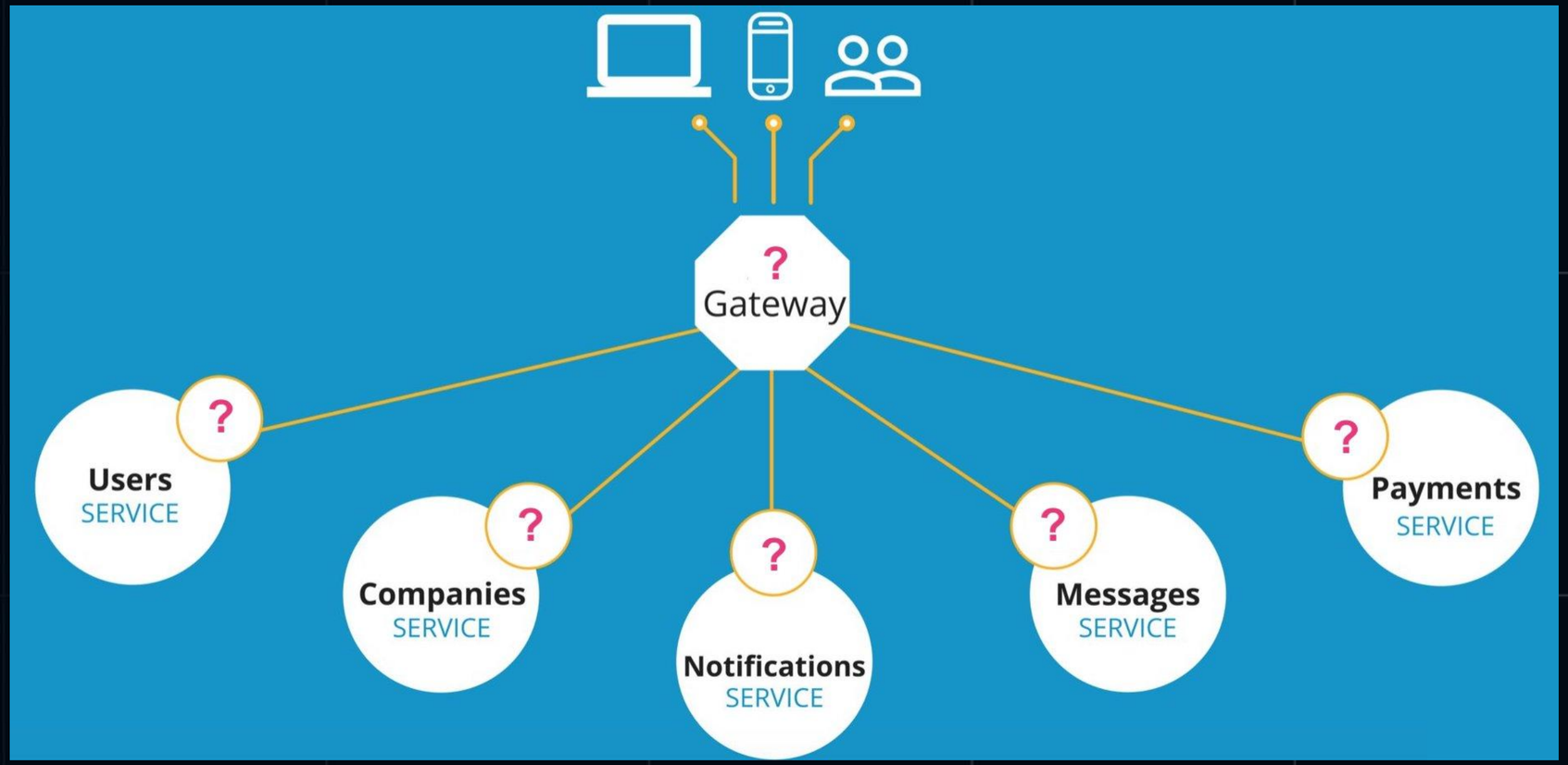
# ПОГНАЛИ?!

# Вступление

- Микросервисы vs Монолит
- 

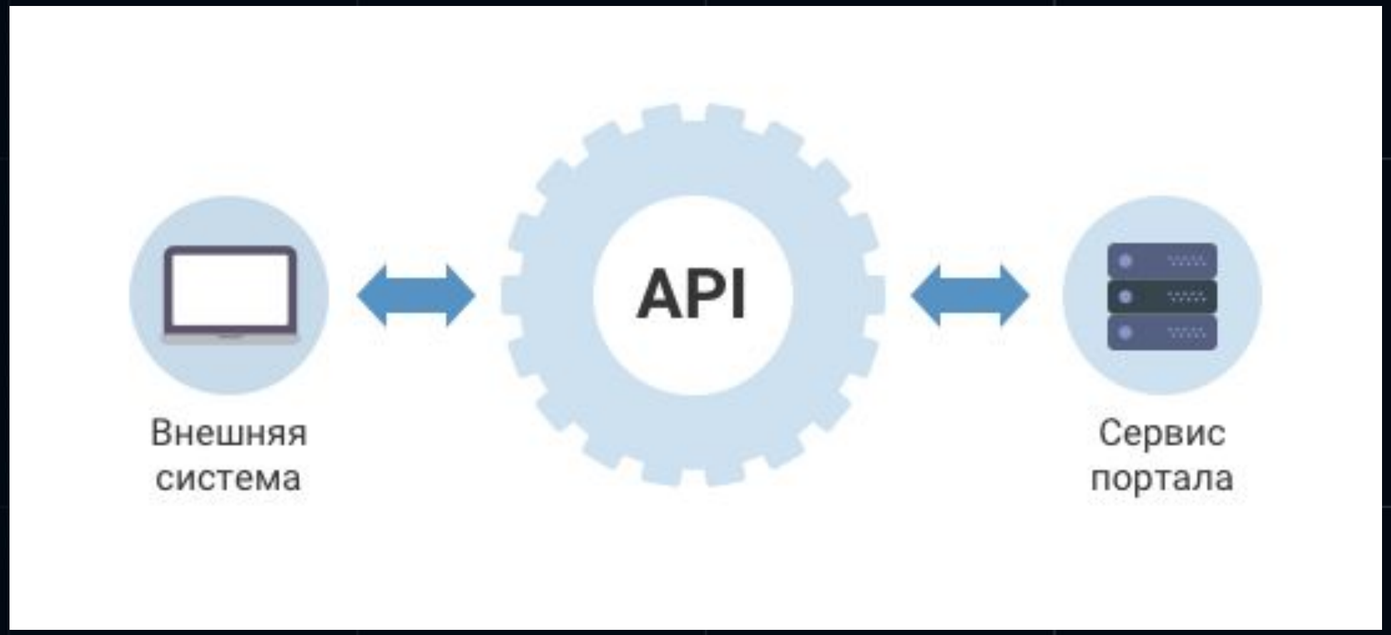
# Вступление

- Микросервисы vs Монолит
- Взаимодействие сервисов между собой
-



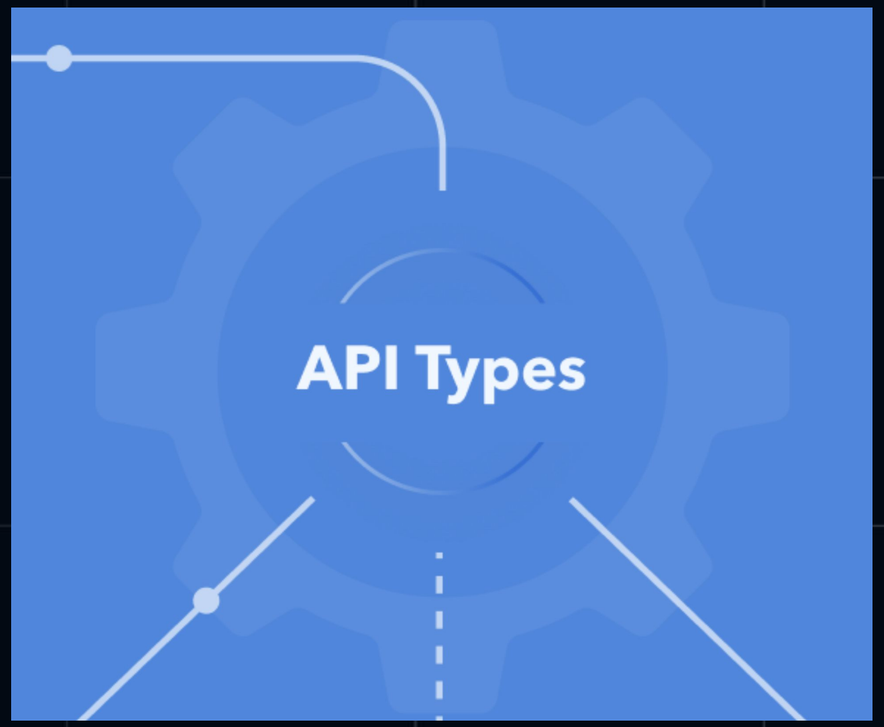
# Вступление

- Микросервисы vs Монолит
- Взаимодействие сервисов между собой
- API (Application Programming Interface)
-



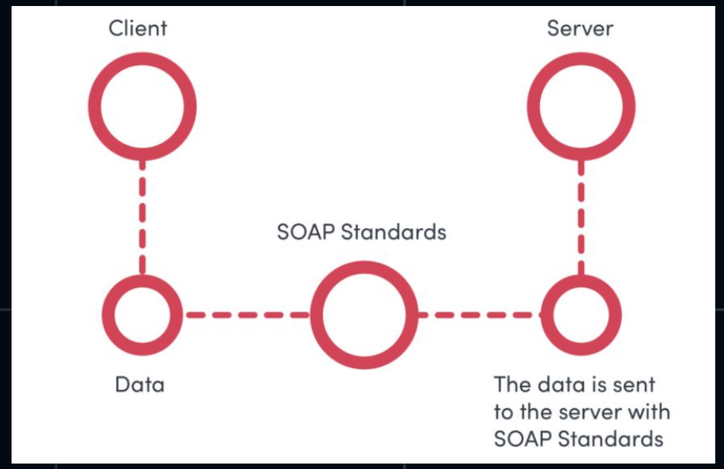
# Вступление

- Микросервисы vs Монолит
- Взаимодействие сервисов между собой
- API (Application Programming Interface)
- Типы:
  - 



# Вступление

- Микросервисы vs Монолит
- Взаимодействие сервисов между собой
- API (Application Programming Interface)
- Типы:
  - SOAP (Simple Object Access Protocol),
  -

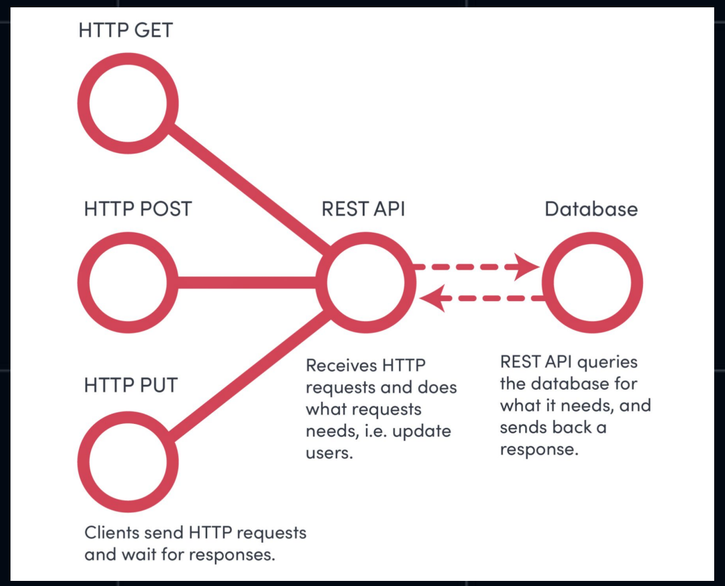


# Вступление

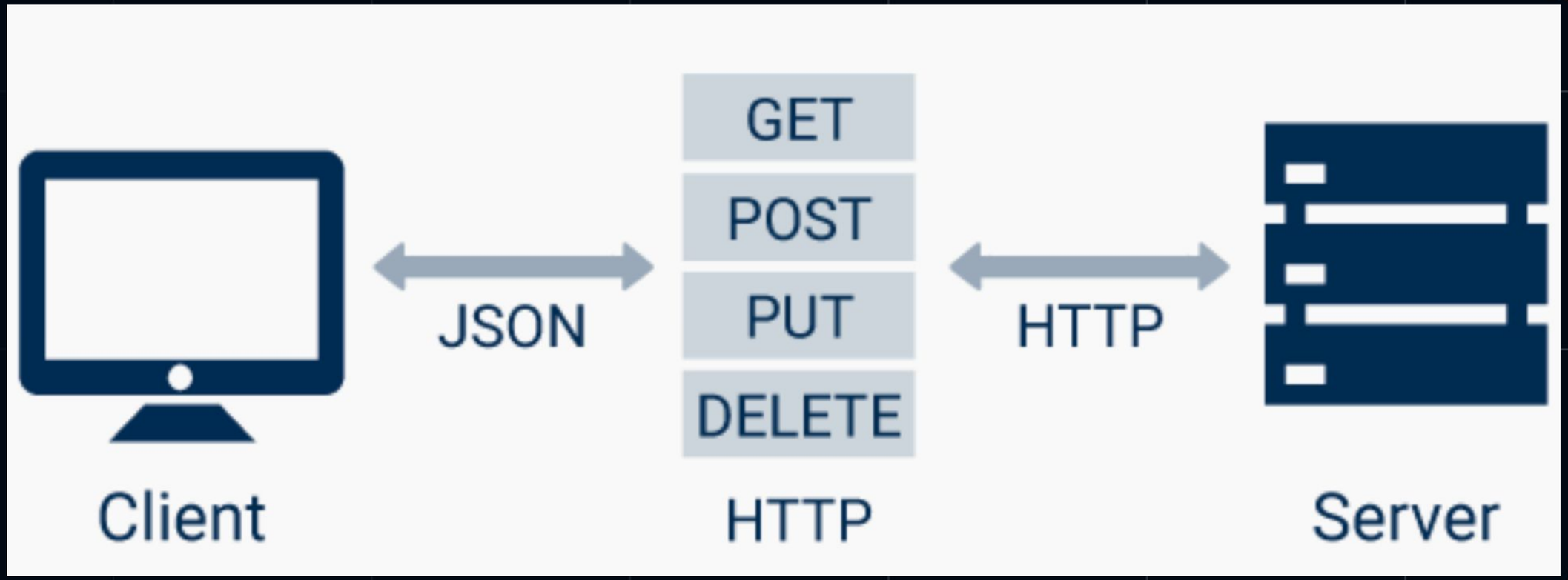
```xml
<?xml version="1.0"?>
<SOAP:Envelope
    xmlns:xsi="http://www.w3.org/1999/XMLSchema/instance"
    xmlns:xsd="http://www.w3.org/1999/XMLSchema/instance"
    xmlns:SOAP="urn:schemas-xmlsoap-org:soap.v1">
  <SOAP:Body>
      <calculateArea>
        <origin>
           <x xsd:type="float">10</x>
           <y xsd:type="float">20</y>
        </origin>
        <corner>
           <x xsd:type="float">100</x>
           <y xsd:type="float">200</y>
        </corner>
       </calculateArea>
   </SOAP:Body>
</SOAP:Envelope>
```

# Вступление

- Микросервисы vs Монолит
- Взаимодействие сервисов между собой
- API (Application Programming Interface)
- Типы:
  - SOAP (Simple Object Access Protocol),
  - REST (REpresentational State Transfer),
  -



# Вступление



# Вступление

Endpoint

```
https://apiurl.com/review/new
```

HTTP Method

```
POST
```

HTTP Headers

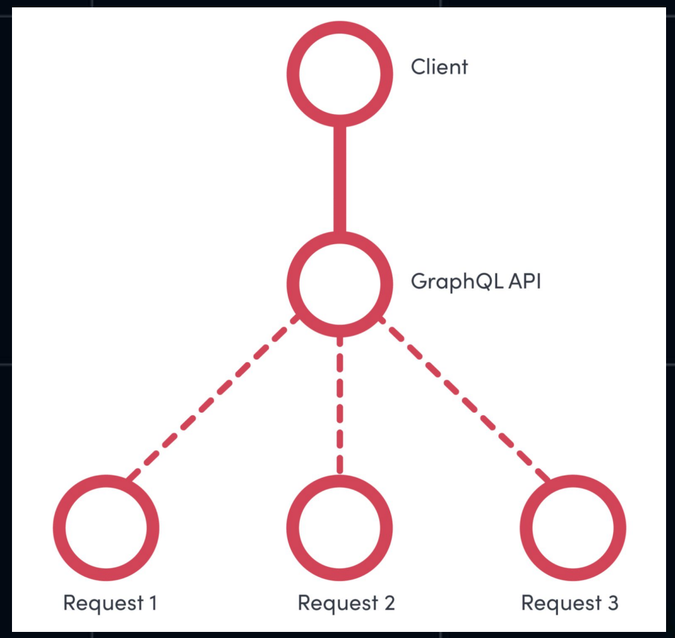
```
content-type: application/json
accept: application/json
                Basic abase64string
```

Body

```
{
  “review” : {
    “title” : “Great article!”,
    “description” : “So easy to follow.”,
    “rating” : 5
  }
}
```

# Вступление

- Микросервисы vs Монолит
- Взаимодействие сервисов между собой
- API (Application Programming Interface)
- Типы:
  - SOAP (Simple Object Access Protocol),
  - REST (REpresentational State Transfer),
  - GraphQL (Graph Query Language)



# gRPC

•

# gRPC

- Простое определение сервисов
-

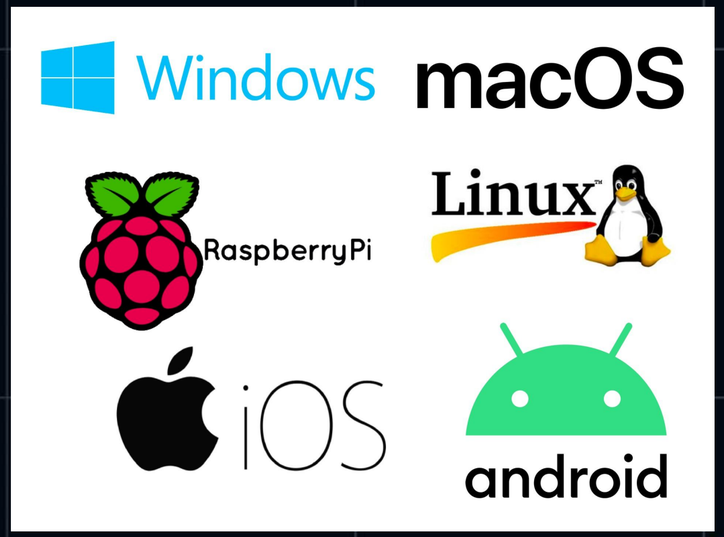
```
// The greeter service definition.
service Greeter {
  // Sends a greeting
  rpc SayHello (HelloRequest) returns (HelloReply) {}
}

// The request message containing the user's name.
message HelloRequest {
  string name = 1;
}

// The response message containing the greetings
message HelloReply {
  string message = 1;
}
```

# gRPC

- Простое определение сервисов
- Быстрое начало и дальнейшее расширение
- 10 - десять
  100 - сто
  1 000 - тысяча
  1 000 000 - миллион
  1 000 000 000 - миллиард
  1 000 000 000 000 - триллион
  1 000 000 000 000 000 - квадрильон
  1 000 000 000 000 000 000 - квинтильон
  1 000 000 000 000 000 000 000 - секстильон
  1 000 000 000 000 000 000 000 000 - септильон
  1 000 000 000 000 000 000 000 000 000 - октальон
  1 000 000 000 000 000 000 000 000 000 000 - нональон
  1 000 000 000 000 000 000 000 000 000 000 000 - декальон
  1 000 000 000 000 000 000 000 000 000 000 000 000 - эндекальон
  1 000 000 000 000 000 000 000 000 000 000 000 000 000 - додекальон

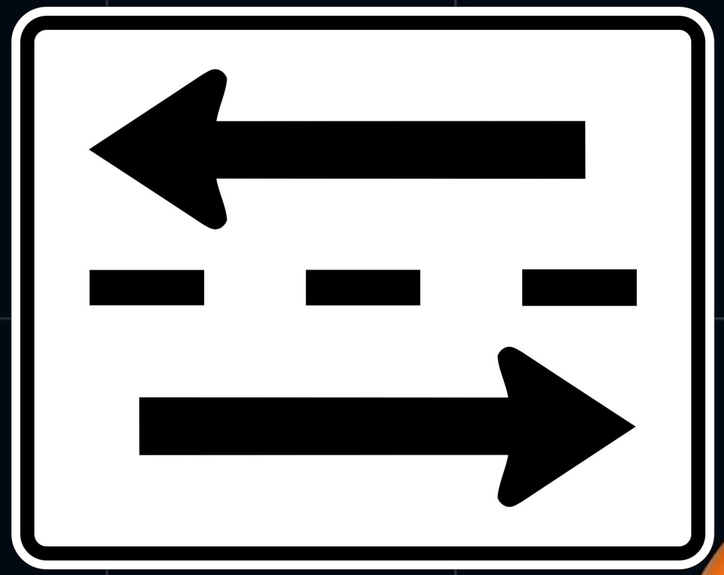
# gRPC

- Простое определение сервисов
- Быстрое начало и дальнейшее расширение
- Взаимодействие с различными языками программирования и платформами
- 

# gRPC

- Простое определение сервисов
- Быстрое начало и дальнейшее расширение
- Взаимодействие с различными языками программирования и платформами
- Реверсивная потоковая передача данных
-



# gRPC

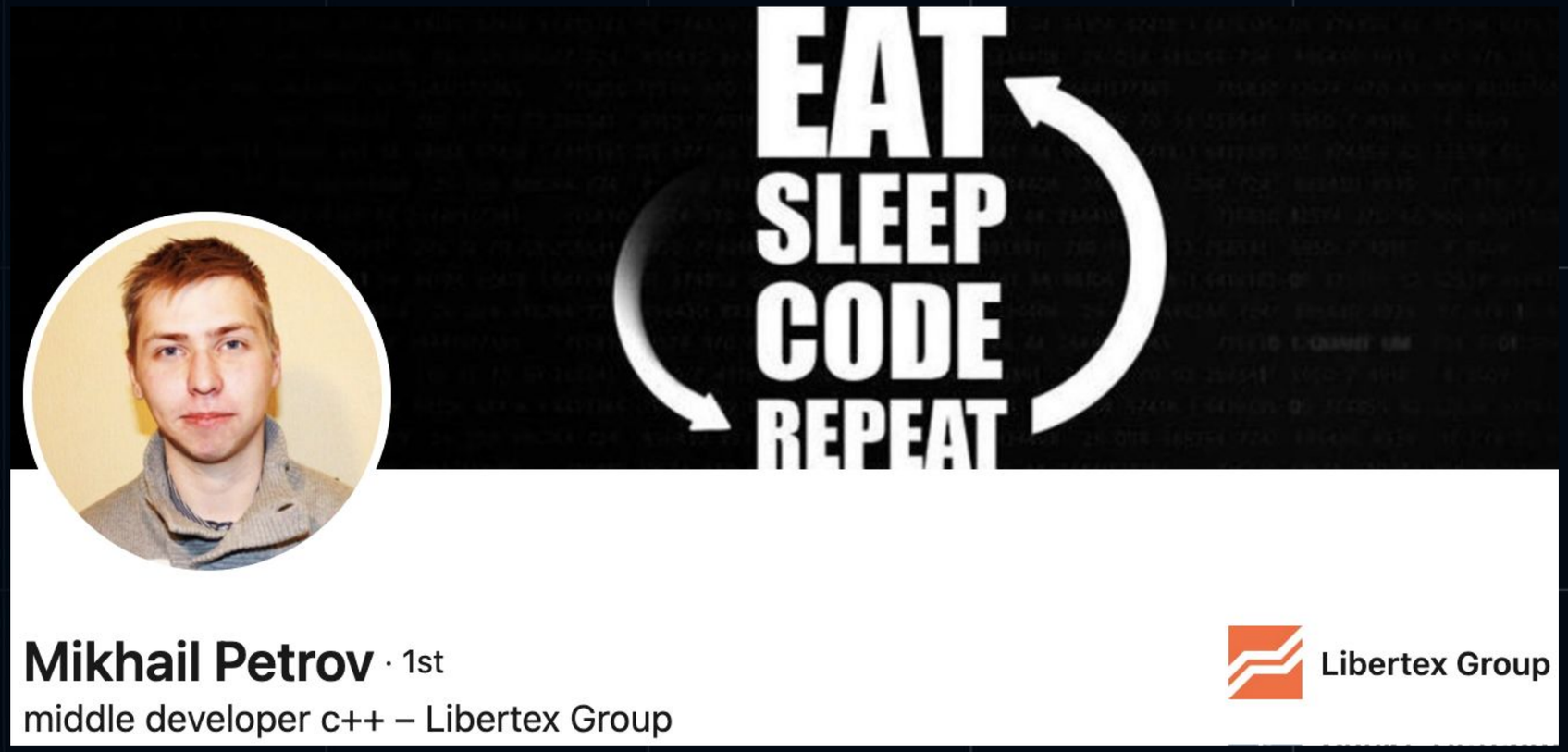
- Простое определение сервисов
- Быстрое начало и дальнейшее расширение
- Взаимодействие с различными языками программирования и платформами
- Реверсивная потоковая передача данных
- Встроенная проверка подлинности

# Демо проект gRPC

- 41 lines (37 sloc) | 1.49 KB

```python
# Autor Mikhail Petrov
# 02.10.2022

"""gRPC's Python Example services."""
import grpc
from grpc_reflection.v1alpha import reflection
import concurrent.futures as futures
import GrpcExampleService_pb2
import GrpcExampleService_pb2_grpc
import DBHelper

""" A python class that implements .proto file's methods """
class GrpcExampleService(GrpcExampleService_pb2_grpc.GrpcExampleService):
  dbhelper = 0
```

# Демо проект gRPC

- Проект был подготовлен моим коллегой, талантливым разработчиком Михаилом Петровым
- 


LinkedIn

# Демо проект gRPC

- Проект был подготовлен моим коллегой, талантливым разработчиком Михаилом Петровым
- Усовершенствование проекта провел другой мой коллега и не менее талантливый разработчик Александр Смыслов
- 


LinkedIn

# Демо проект gRPC

- Проект был подготовлен моим коллегой, талантливым разработчиком Михаилом Петровым
- Усовершенствование проекта провел мой другой коллега и не менее талантливый разработчик Александр Смыслов
- https://github.com/MikhailPO/grpcExampleService

SCAN ME

# Демо проект - подготовка

- `/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"`
- `brew install python`
- `brew unlink python && brew link python`
- `pip install grpcio-tools`
- `pip install grpcio-reflection`

# Демо проект - запуск

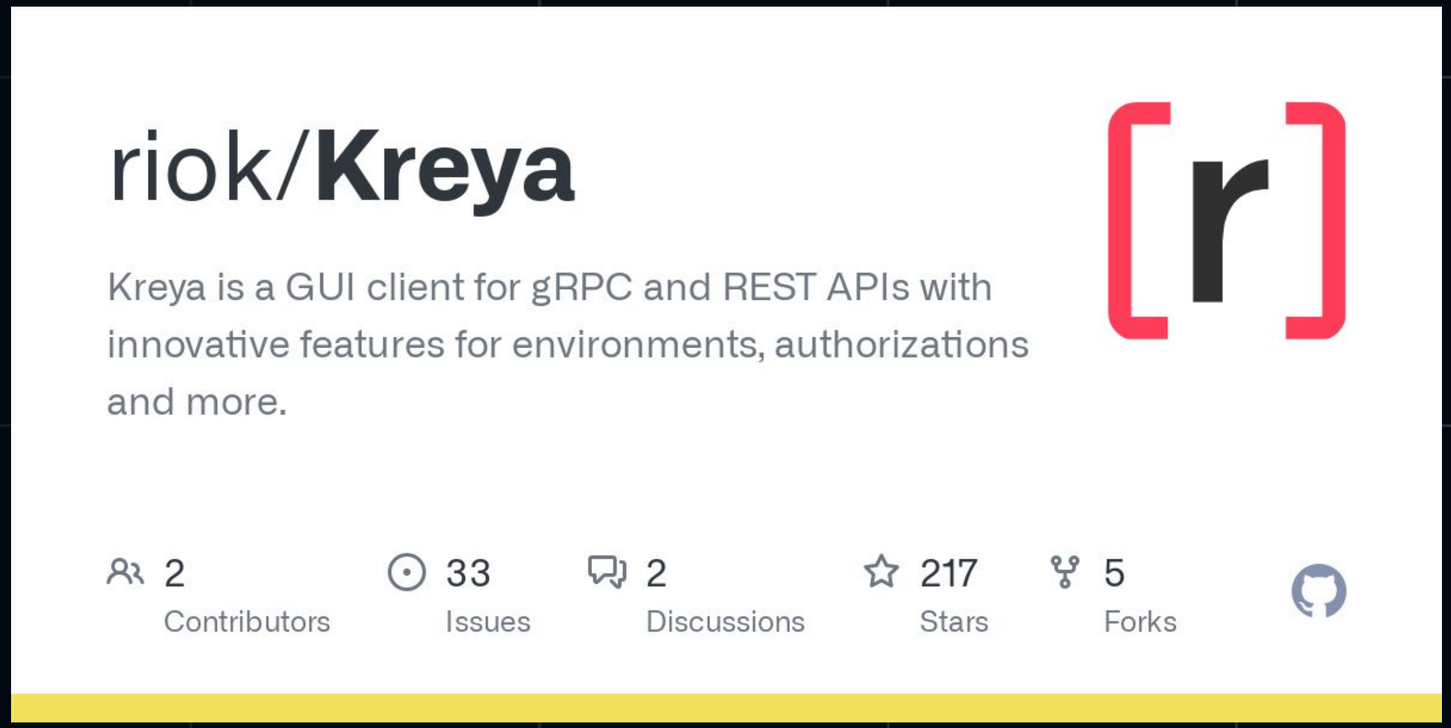
- `git clone https://github.com/MikhailPO/grpcExampleService`
- `cd grpcExampleService`
- `find . -name "*.sh" -exec chmod +x {} \;`
- `./generate_file.sh`
- `./install.sh`
- `./start.sh`

# Сравнение

| Инструмент | gRPC | Server Reflection | Удобство | Команда | Импорт |
|---|---|---|---|---|---|
| Kreya | | | | | |
| BloomRPC | | | | | |
| Insomnia | | | | | |
| Postman | | | | | |
| Hoppscotch | | | | | |

# Kreya

- Компания [riok]
- Молодой
- gRPC и REST



# BloomRPC

- Создатель Fabrizio Fenoglio
- Впервые выпущен в декабре 2018
- Больше не поддерживается

# Insomnia

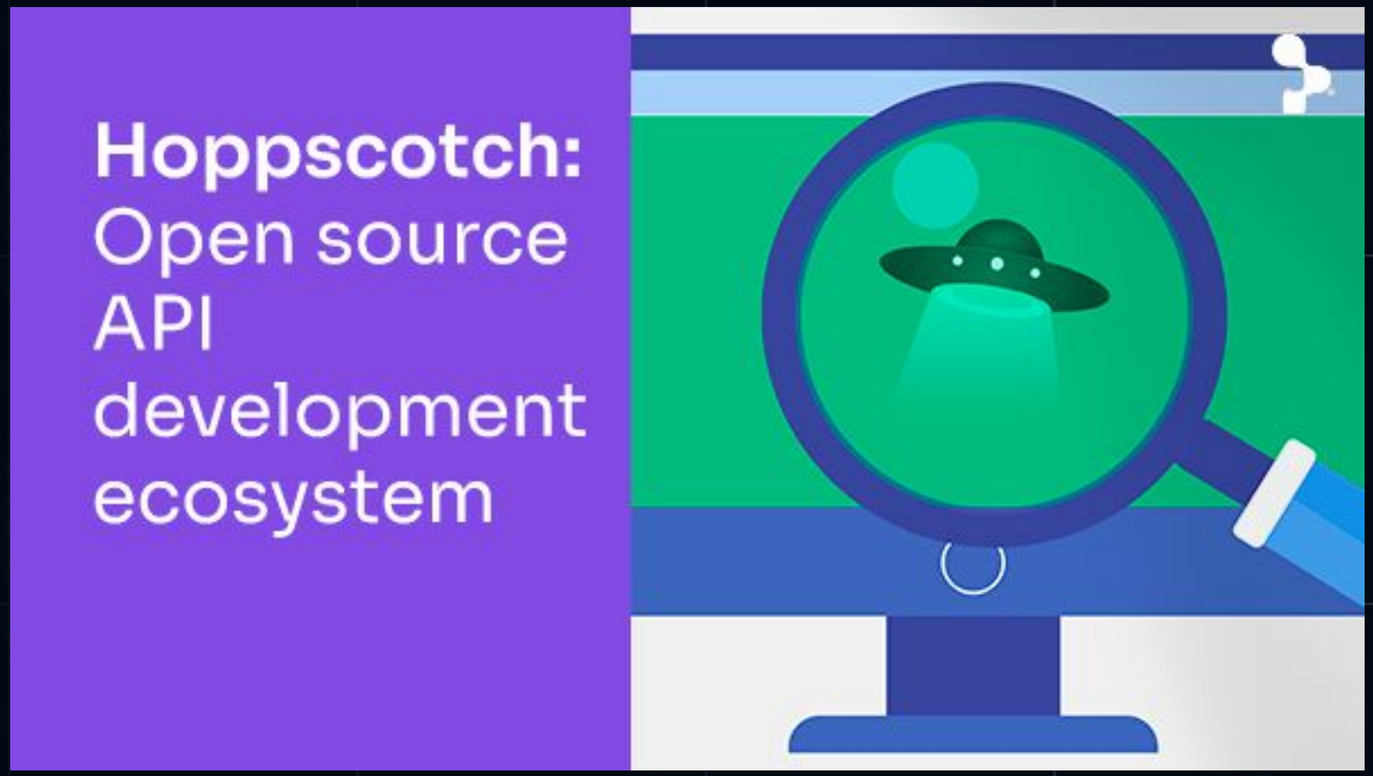
- Выпускается компанией Kong
- Есть платная версия

# Postman

- Один из первых
- Один из самых популярных
- Один из самых навороченных

# Hoppscotch

- Молодой инструмент
- Очень похож на Postman



# Сравнение

| Инструмент | gRPC | Server Reflection | Удобство | Команда | Импорт |
|---|---|---|---|---|---|
| Kreya | + | + | + | - | - |
| BloomRPC | + | - | - | - | + |
| Insomnia | + | - | + | - | + |
| Postman | + | +/- | + | + | + |
| Hoppscotch | - | - | + | + | - |

# Kreya

- Плюсы
  - 

# Kreya

- Плюсы
  - Не нагруженный
  -

# Kreya

- Плюсы
  - Не нагруженный
  - Автоматически подтягивается тело запроса
  -

# Kreya

- Плюсы
  - Не нагруженный
  - Автоматически подтягивается тело запроса
  - Есть вкладки

# Kreya

- Минусы
  - 

# Kreya

- Минусы
  - Ошибки
  -

# Kreya

- Минусы
  - Ошибки
  - Нет работы в команде

# Postman

- Плюсы
  - 

# Postman

- Плюсы
  - Все протоколы
  -

# Postman

- Плюсы
  - Все протоколы
  - Есть вкладки
  -

# Postman

- Плюсы
  - Все протоколы
  - Есть вкладки
  - Работа в команде
  -

# Postman

- Плюсы
  - Все протоколы
  - Есть вкладки
  - Работа в команде
  - Импорт

# Postman

- Минусы
  - 

# Postman

- Минусы
  - Ограниченная работа в команде
  -

# Postman

- Минусы
  - Ограниченная работа в команде
  - gRPC регистрация

# Hoppscotch

- Плюсы
  - 

# Hoppscotch

- Плюсы
  - Неограниченная работа в команде
  -

# Hoppscotch

- Плюсы
  - Неограниченная работа в команде
  - Легко перейти с Postman
  -

# Hoppscotch

- Плюсы
  - Неограниченная работа в команде
  - Легко перейти с Postman
  - Не перегруженный

# Hoppscotch

- Минусы
  - 

# Hoppscotch

- Минусы
  - Не работает с gRPC
  -

# Hoppscotch

- Минусы
  - Не работает с gRPC
  - Нет вкладок
  -

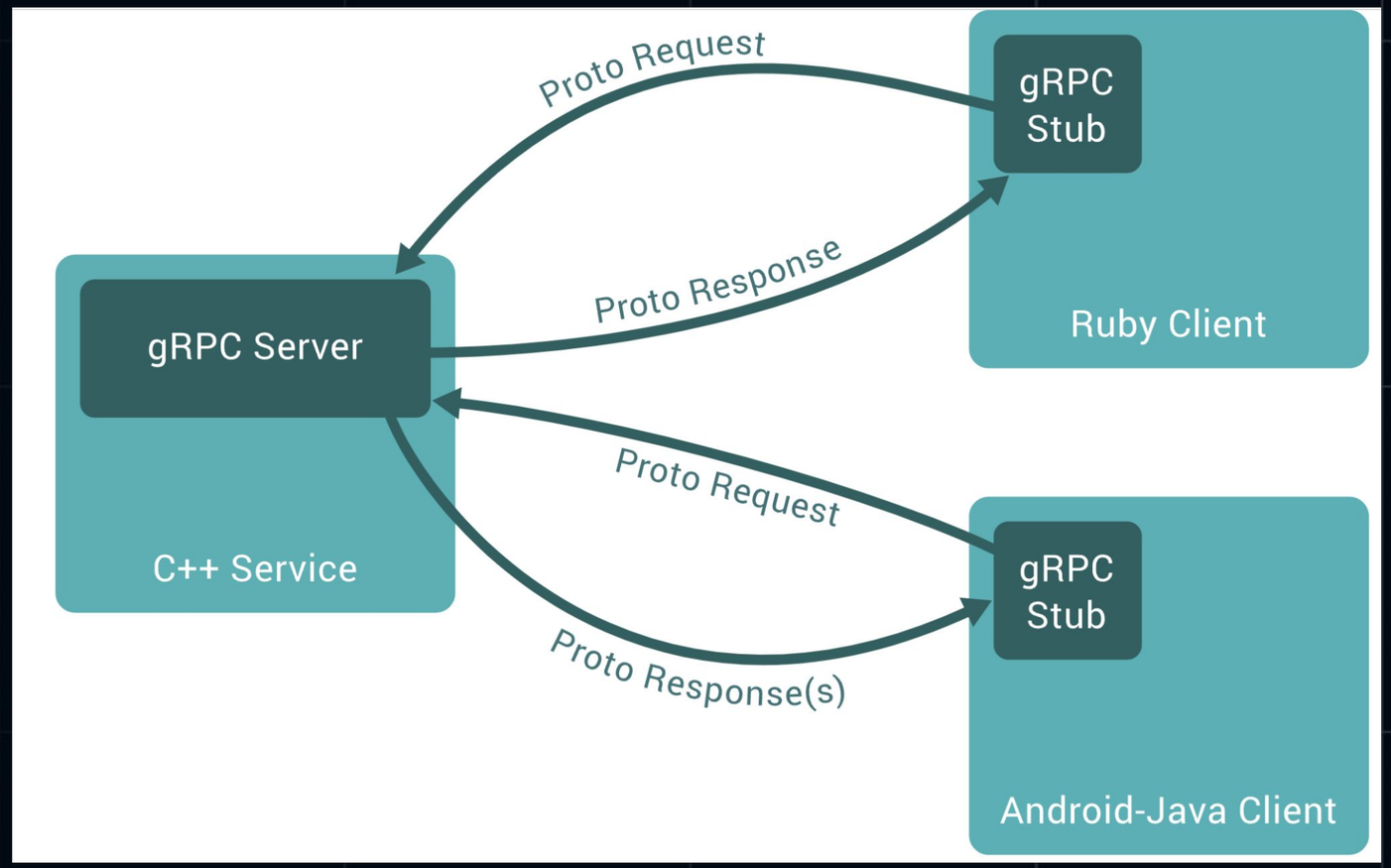
# Hoppscotch

- Минусы
  - Не работает с gRPC
  - Нет вкладок
  - Импорт коллекций


Выводы

# Выводы

- gRPC интересный и перспективный



# Выводы

- gRPC интересный и перспективный
- как тестировать

```
Terminal  Shell  Edit  View  Window  Help

user@Users-iMac-2 ~ % grpc-client-cli --address localhost:50051
? Choose a service: GrpcExampleService
? Choose a method: GetClients
Message json (type ? to see defaults): {}
{
  "response_clients": {
    "clients": [
      {
        "id": 1,
        "login": "Tom",
        "email": "postman@gmail.com",
        "city": "Postman"
      }
    ]
  }
}
Message json (type ? to see defaults):
```

# Выводы

- gRPC интересный и перспективный
- как тестировать
- инструменты тестера

# Выводы

- gRPC интересный и перспективный
- как тестировать
- инструменты тестера
- какой лучший

# Q&A сессия

- Прошу задавать свои вопросы
  - Здесь в чате
  - В LinkedIn
  - Напрямую в Telegram @shaikin

SCAN ME

# Благодарю за внимание