



# Зачем и где нужен API Gateway

Акманова Елизавета



# Акманова Елизавета

Старший аналитик UseTech

Спикер на IT конференциях

Автор статей, подкастов



# ПЛАН

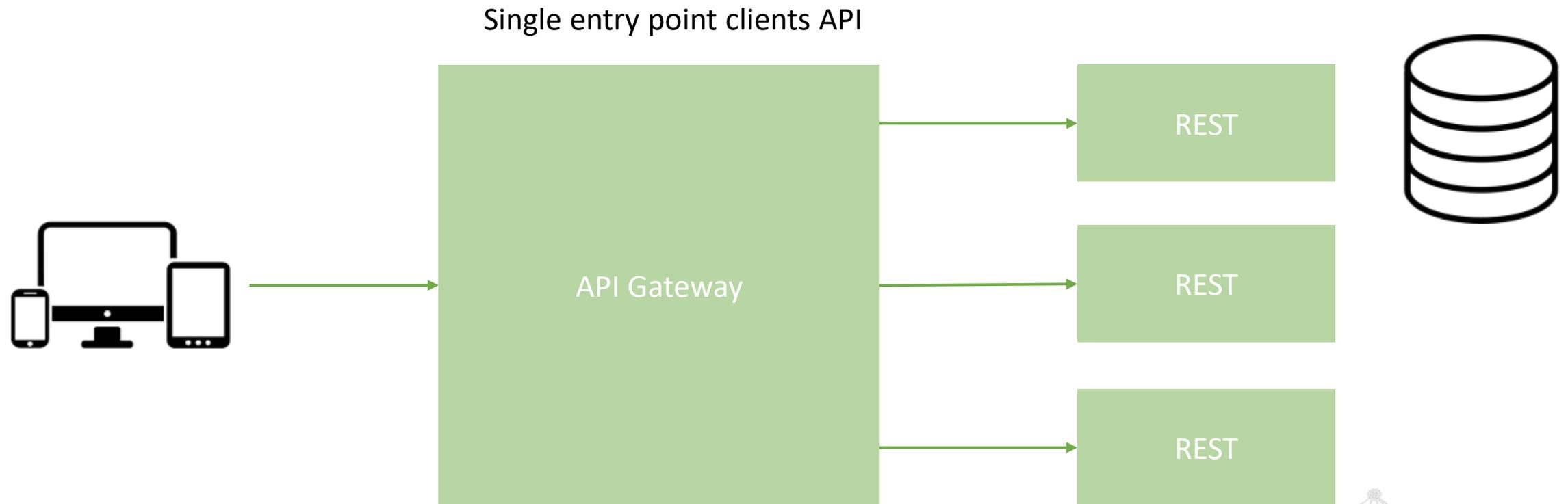
1 | Архитектура API Gateway

2 | Решаемые задачи Gateway

3 | Когда использовать?

4 | Примеры использования

# Архитектура API Gateway



# Если бы HTTP запросы летали самолетами

# Если бы HTTP запросы летали самолетами



# Если бы HTTP запросы летали самолетами



# Если бы HTTP запросы летали самолетами



# Если бы HTTP запросы летали самолетами



# Если бы HTTP запросы летали самолетами



# Унификация

Один интерфейс для всех сервисов

Унификация позволяет разработчикам работать по одному принципу, не разбираясь с разными интерфейсами

Все изменения и обновления применяются в одном месте, что упрощает поддержку системы

# Маршрутизация

Перенаправление запросов на целевые сервисы

Балансировка распределяет нагрузку между одинаковыми сервисами

# Валидация

Проверка наличия обязательных параметров

Проверка доступов предотвращает несанкционированные запросы

Валидация – первый рубеж защиты API

# Авторизация и Аутентификация

Единая точка аутентификации пользователей и/или сервисов

Проверка доступа перед выполнением операций

Сервисам не нужно думать о интеграции и решать эту задачу

# Трассировка запросов

Самое главное свойство: предотвращение зацикливания запросов

Возможность наблюдать за пройденными путями запроса для выполнения следующих функций:

- ограничение времени жизни запроса
- отслеживание связей микросервисов и рекурсивная логика

# Логирование

Все запросы пишутся в журнал

Данные журналы непрерывно пересылаются в некое хранилище для выполнения определенных функций:

- для аналитики/статистики
- для разбора проблем
- для мониторинга

# Стандартизация

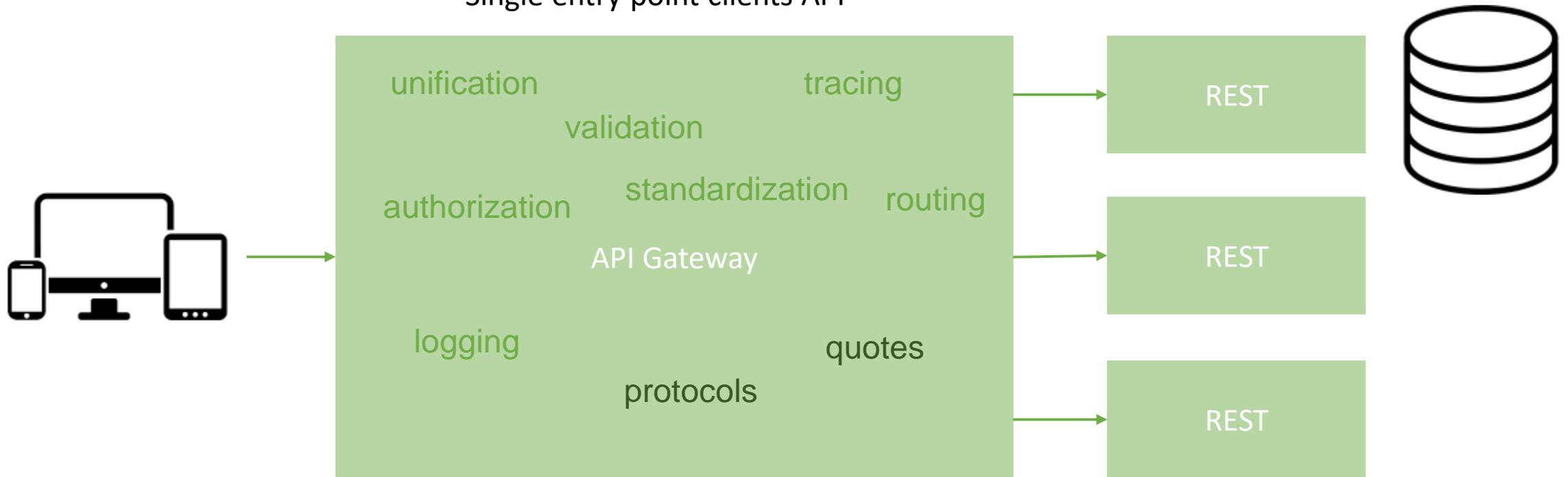
Удаление лишних данных – API Gateway фильтрует ненужную информацию

Обогащение запроса – добавление недостающих данных для работы сервисов

Единый стандарт – единый формат запросов для всех сервисов

# Архитектура API Gateway (AGW)

Single entry point clients API



# Протоколирование

Внешний мир, как правило, требует что-то на входе REST, WebSockets, JSON RPC...

Внутренний мир может требовать gRPC...

# Квотирование запросов

Ограничение количества запросов к сервисам на основе разных политик/правил

Две возможных реализации квотирования:

- прекращение обслуживания
- замедление запросов

Позволяет избежать падения сервисов из-за чрезмерной нагрузки

# Промежуточный вывод

## Примечание:

- Не все перечисленные функции обязательны
- Их может быть 2, 1, или все...

## Преимущества:

- Меньше разработки за счет переиспользования функций
- Лучший контроль инфраструктуры за счет единой точки входа

## Недостатки:

- Сложное решение
- Сложно интегрировать в существующую систему

# ПЛАН

1 | Архитектура API Gateway

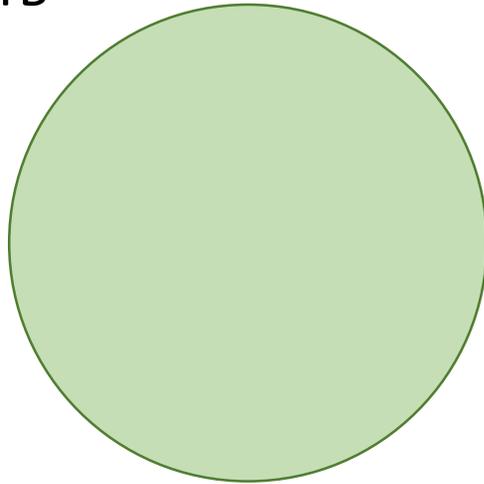
2 | Решаемые задачи Gateway

3 | Когда использовать?

4 | Примеры использования

# Когда использовать?

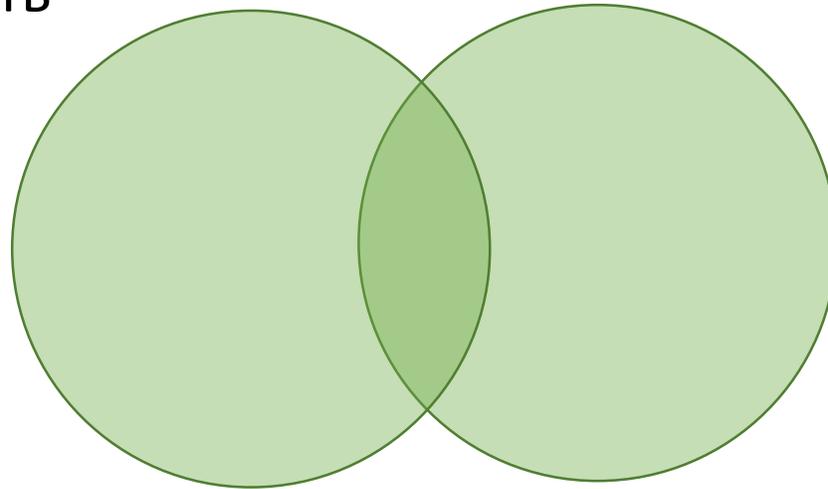
Проект очень большой и  
функции нужно унифицировать



# Когда использовать?

Проект очень большой и функции нужно унифицировать

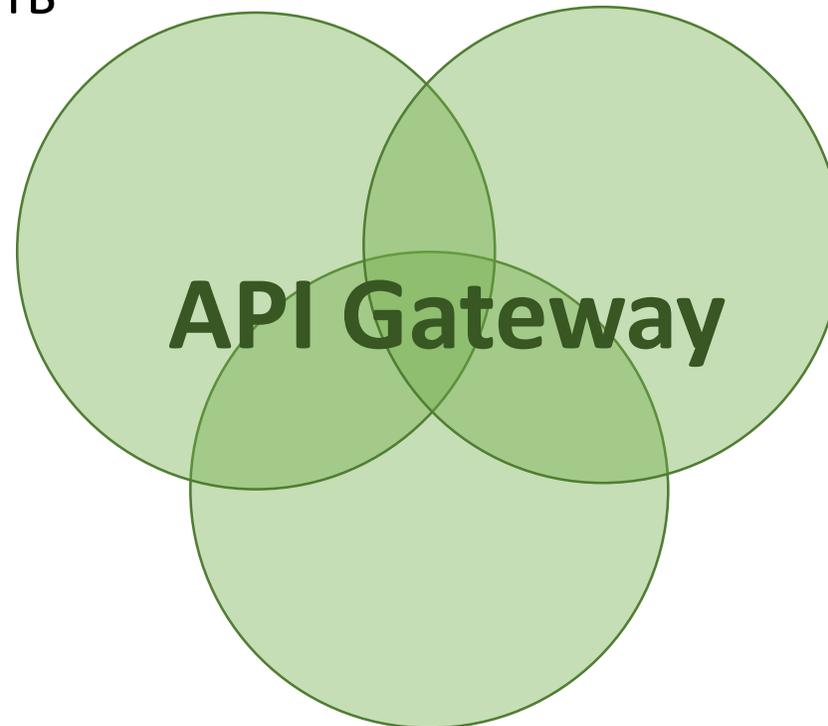
Необходимо управлять большим количеством микросервисов



# Когда использовать?

Проект очень большой и функции нужно унифицировать

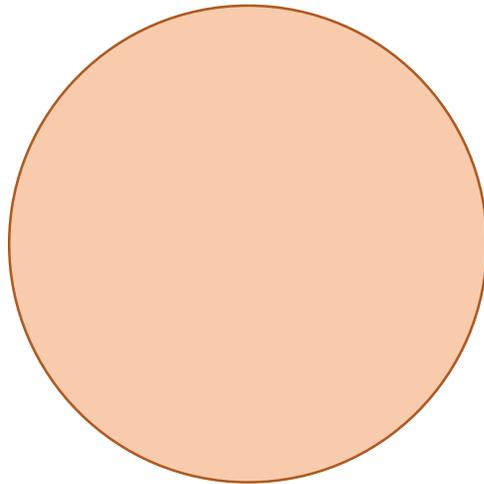
Необходимо управлять большим количеством микросервисов



Внедрение и обслуживание API Gateway дешевле, чем реализация проекта без него

# Когда не использовать?

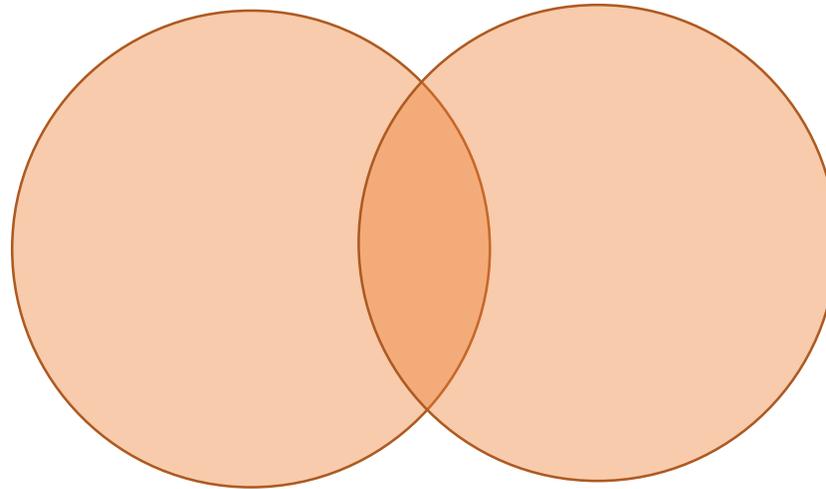
Проект небольшой и прост в реализации



# Когда не использовать?

Проект небольшой и прост в реализации

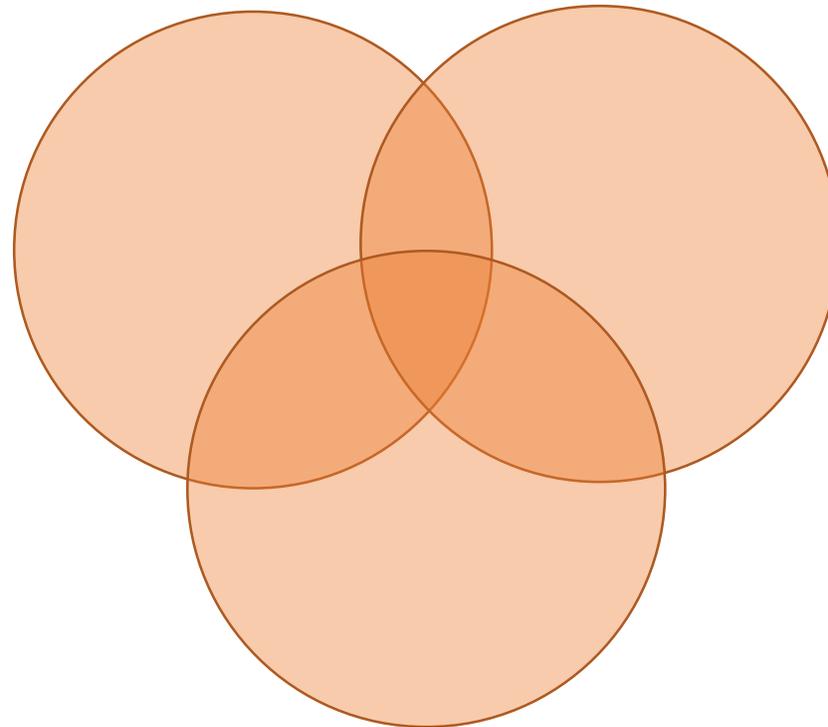
Небольшое количество микросервисов



# Когда не использовать?

Проект небольшой и прост в реализации

Небольшое количество микросервисов



Ограниченный бюджет и ресурсы

# Когда не использовать?

Проект небольшой и прост в реализации

Небольшое количество микросервисов



Ограниченный бюджет и ресурсы

# Добавим к выводу

Использование API Gateway целесообразно не всегда и зависит от множества факторов

# ПЛАН

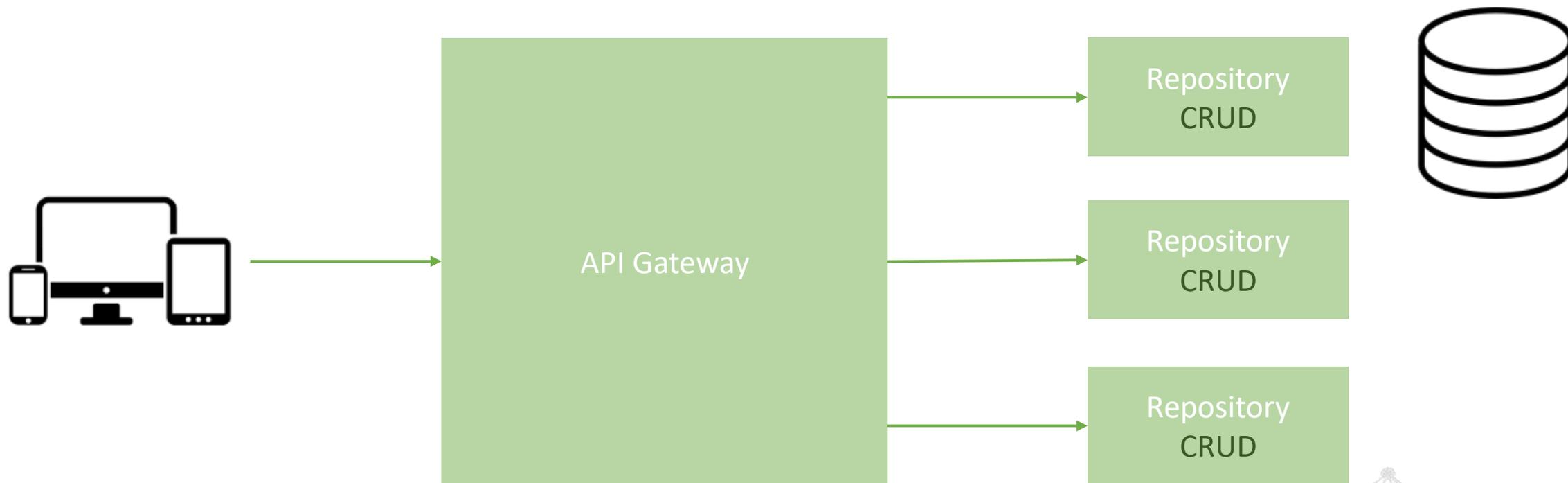
1 | Архитектура API Gateway

2 | Решаемые задачи Gateway

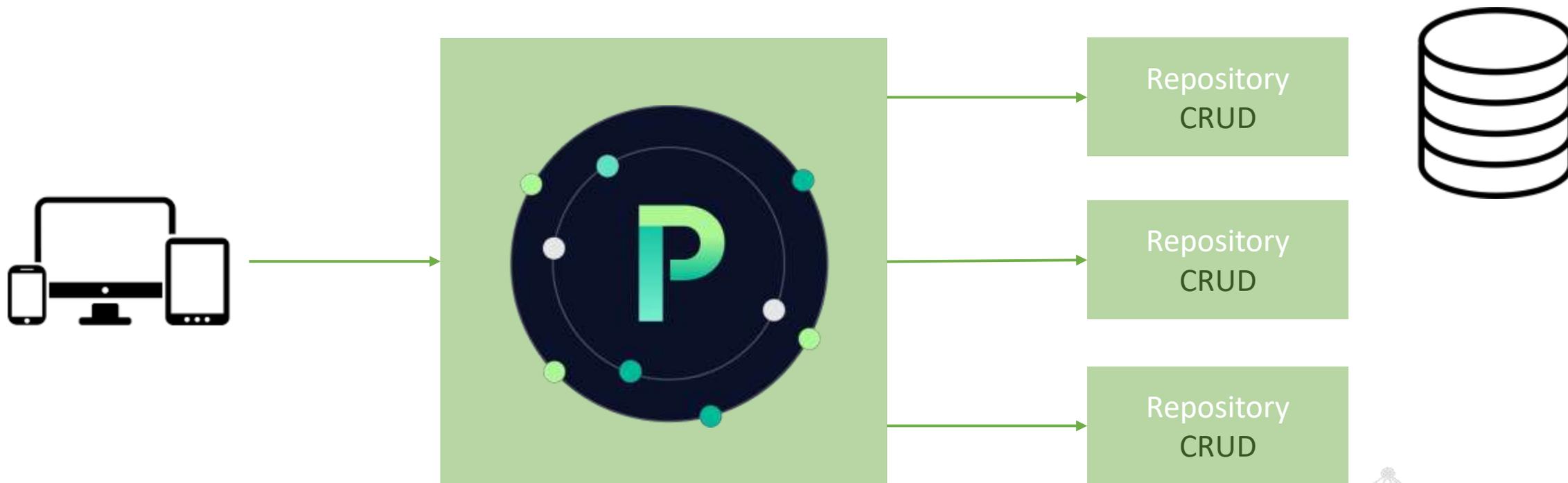
3 | Когда использовать?

4 | Примеры использования

# Пример использования



# Пример использования



# Пример использования



# Пример использования

API Gateway с открытым исходным кодом

Широкий набор плагинов



# Пример использования

Платформа для обработки потоков данных в реальном времени

Обеспечивает отказоустойчивое хранение данных и высокую пропускную способность



# Пример использования

Open-source решение для управления идентификацией и доступом



# Пример использования



**Kong**



**kafka**



**KEYCLOAK**

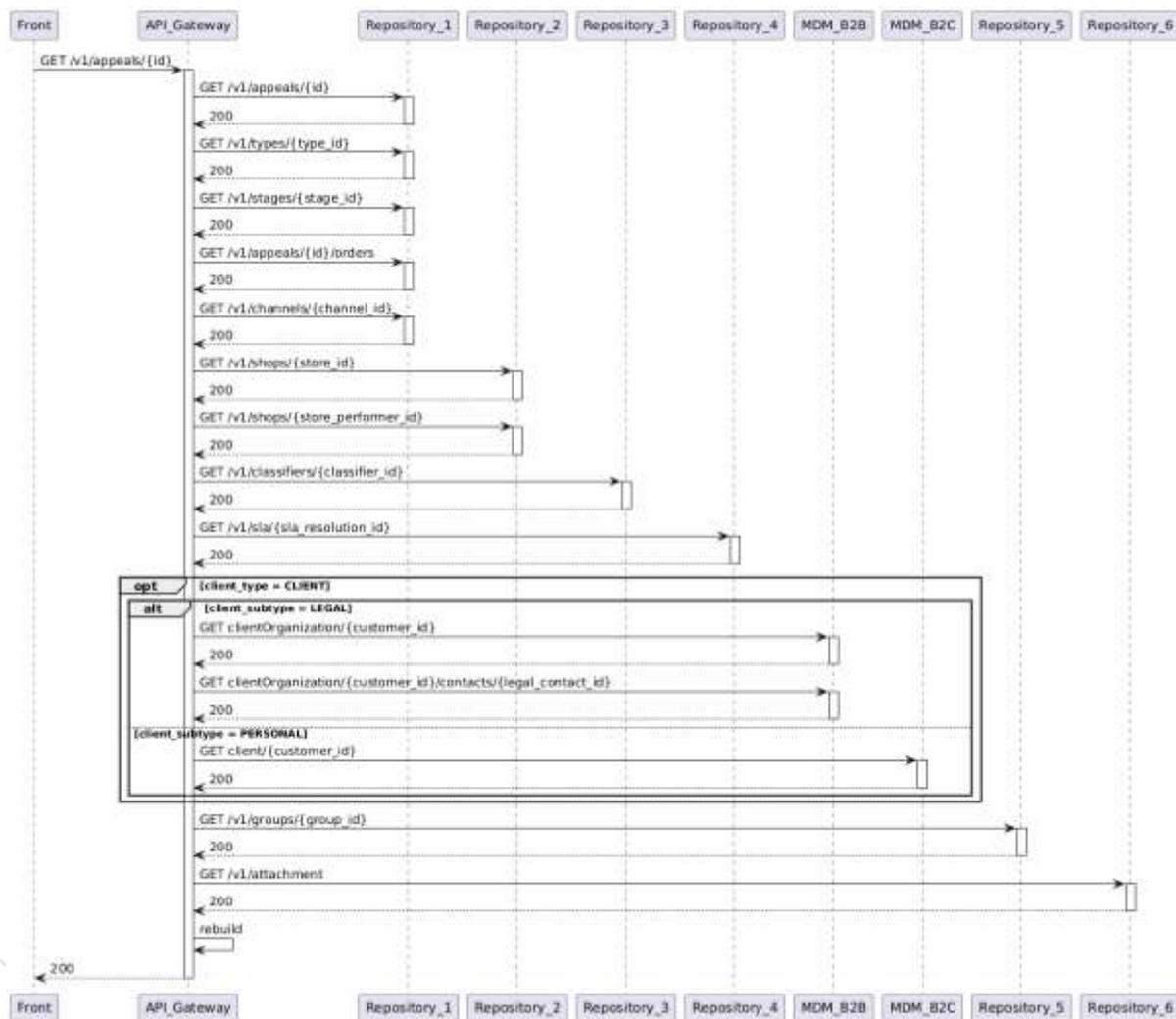
# Пример использования

10-15% — прокси-методы

Прокси-методы — это простейшие методы, которые просто принимают запрос от клиента и перенаправляют его на другой сервис

85-90% — методы с внутренней логикой

# Пример использования



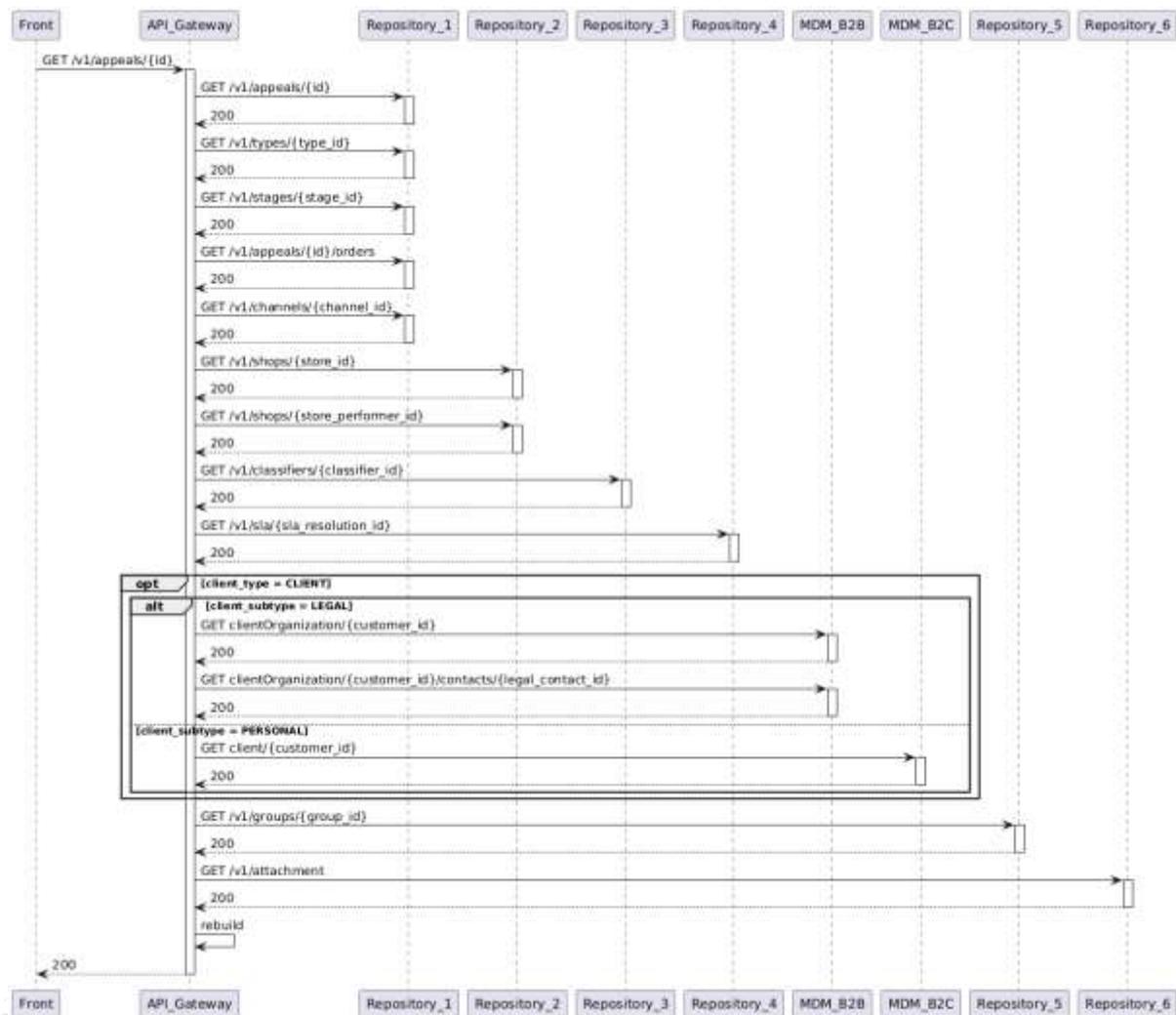
# Пример использования

8 репозиториев (не включая Front и AGW)

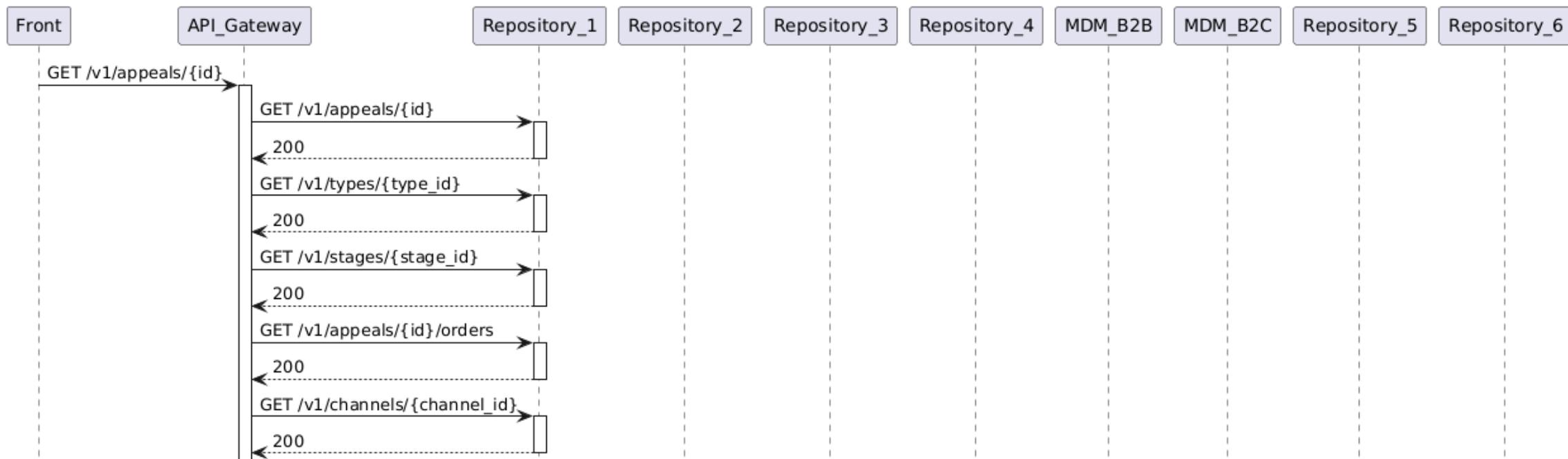
Оркестрация 14 запросов

opt и alt блоки схемы

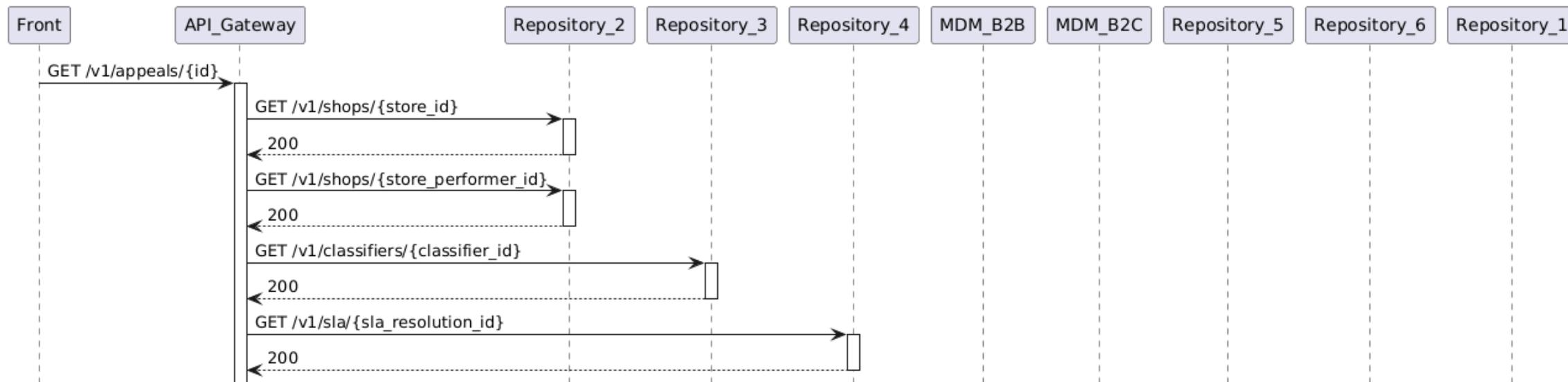
Формирование ответа



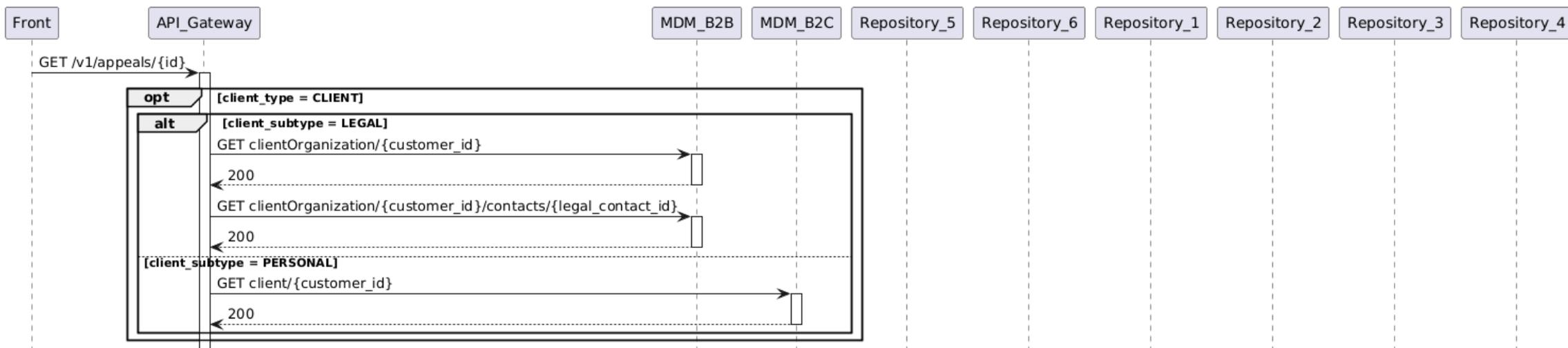
# Пример использования



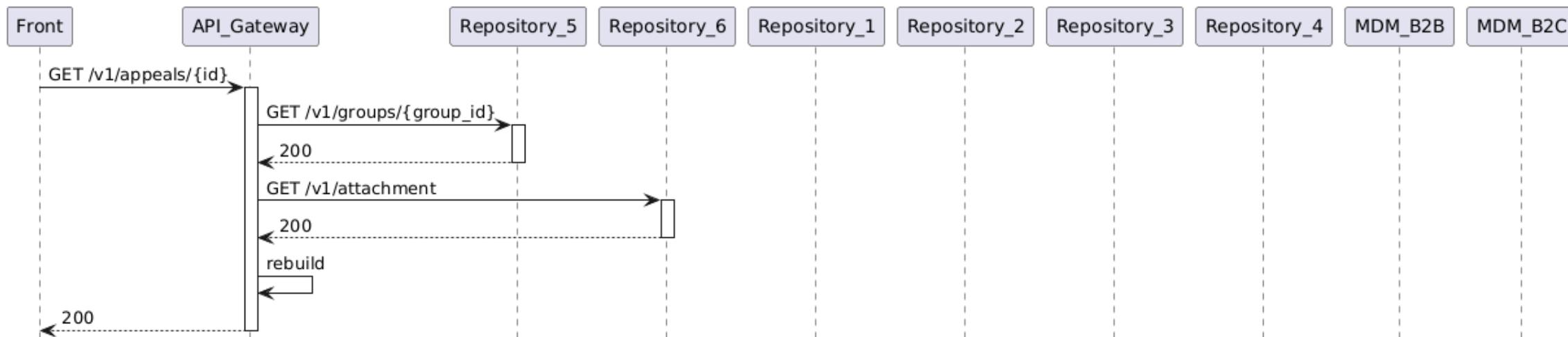
# Пример использования



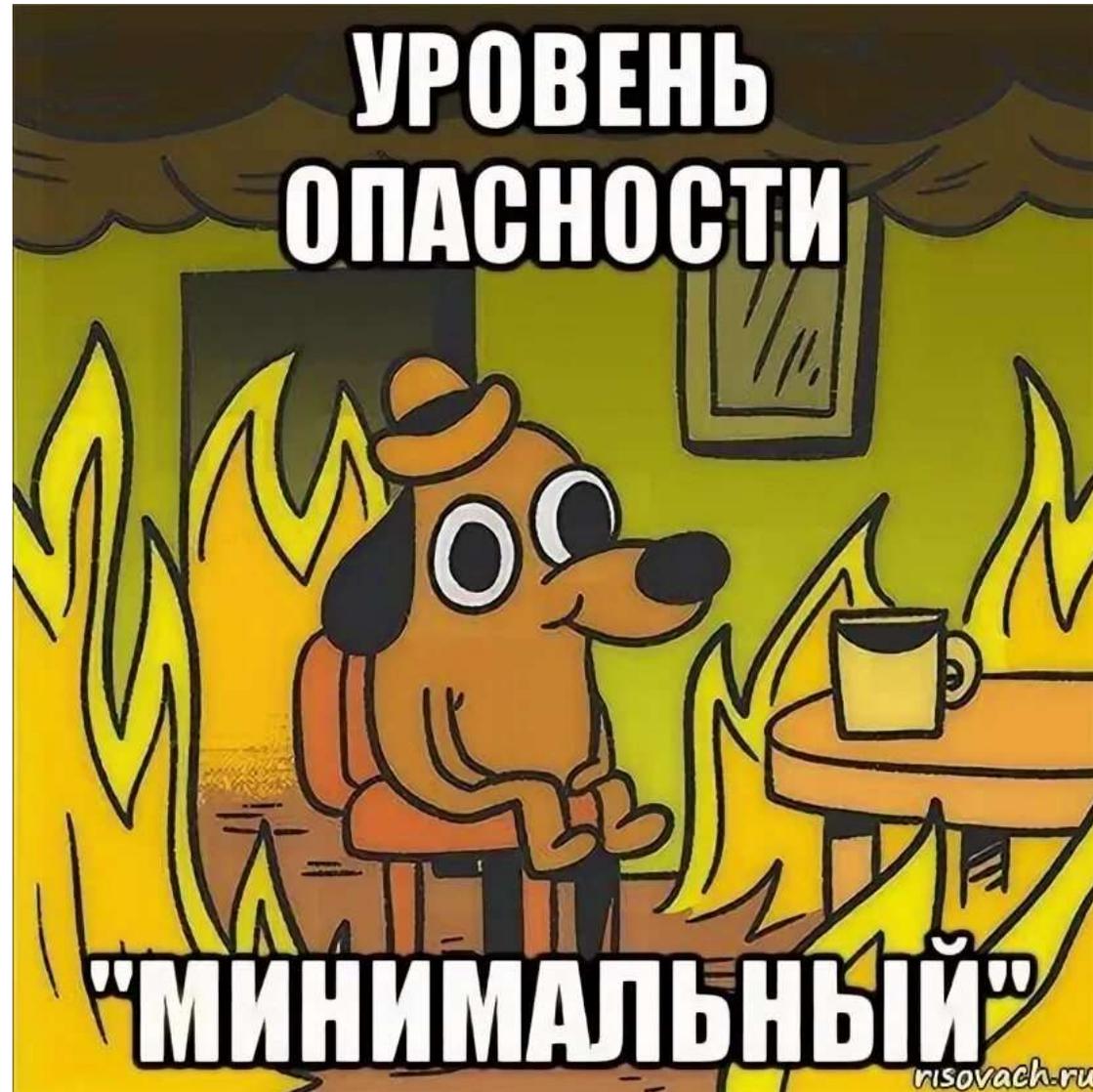
# Пример использования



# Пример использования



~ 900 ms



# Проблема

Риск стать узким местом

# Решение

# Проблема

Риск стать узким местом

->

# Решение

Группированные API Gateway

# Проблема

Риск стать узким местом

Снижение производительности

->

# Решение

Группированные API Gateway

# Проблема

Риск стать узким местом

->

# Решение

Группированные API Gateway

Снижение производительности

->

Горизонтальное масштабирование

# Проблема

Риск стать узким местом

->

Снижение производительности

->

Повышенная сложность системы

# Решение

Группированные API Gateway

Горизонтальное масштабирование

# Проблема

Риск стать узким местом

->

Снижение производительности

->

Повышенная сложность системы

->

# Решение

Группированные API Gateway

Горизонтальное масштабирование

Автоматизация развертывания

# Проблема

Риск стать узким местом

->

Снижение производительности

->

Повышенная сложность системы

->

Привязка к поставщику

# Решение

Группированные API Gateway

Горизонтальное масштабирование

Автоматизация развертывания

# Проблема

Риск стать узким местом

->

Снижение производительности

->

Повышенная сложность системы

->

Привязка к поставщику

->

# Решение

Группированные API Gateway

Горизонтальное масштабирование

Автоматизация развертывания

? ? ?

# Выводы

# Выводы

API Gateway определяется набором выполняемых функций

# Выводы

API Gateway определяется набором выполняемых функций

API Gateway не является универсальным решением и требует тщательной оценки

# Выводы

API Gateway определяется набором выполняемых функций

API Gateway не является универсальным решением и требует тщательной оценки

Пример реализации API Gateway в реальном проекте

# Спасибо за внимание!



## Акманова Елизавета



Контакты



Обратная связь