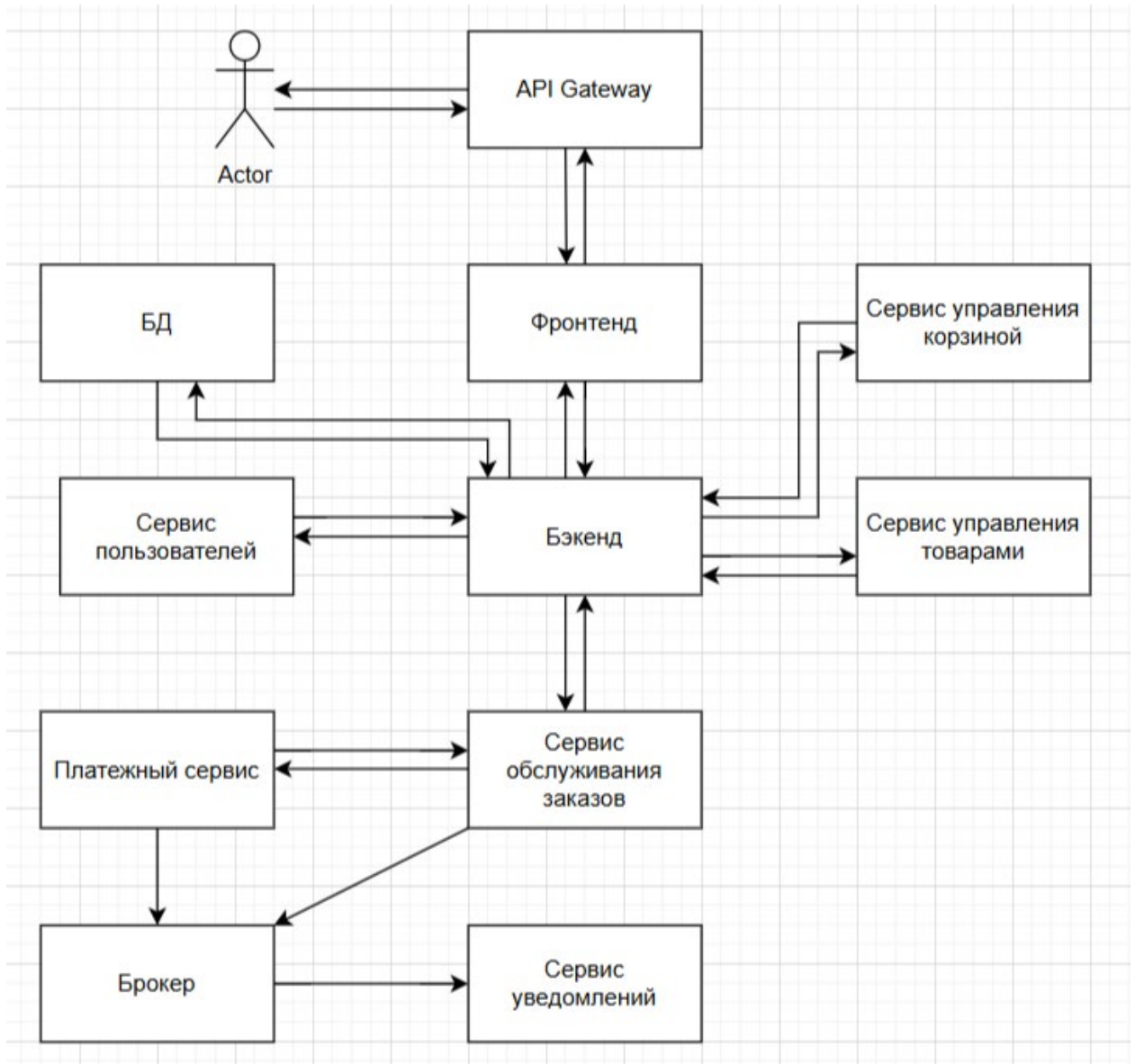


Микросервисное приложение для онлайн-магазина



Это приложение, позволяющее пользователям просматривать товары, добавлять их в корзину, оформлять заказы и осуществлять оплату.

Фронтенд

Интерактивное пользовательское приложение — фронтенд — является точкой входа для пользователей, которое отправляет запросы к API Gateway. Это может быть:

Веб-приложение (React, Angular, Vue.js)

Мобильное приложение (iOS/Android)

Примеры действий пользователя:

- Просмотр каталога товаров
- Добавление товаров в корзину
- Оформление заказа
- Отслеживание статуса доставки

API Gateway

Все запросы идут через API Gateway — маршрутизатора и балансировщика нагрузки. Он обеспечивает маршрутизацию запросов к нужным микросервисам.

Микросервисы:

- **Сервис пользователей** — управляет данными профиля пользователей, обеспечивает регистрацию, вход и управление токенами. Для хранения данных используется PostgreSQL, что позволяет эффективно управлять транзакциями и обеспечивать безопасность данных.
- **Сервис управления товарами** — управляет каталогом товаров. Хранит информацию о товарах для гибкости схемы и быстрого поиска. Обеспечивает API для получения списка товаров, поиска, фильтрации и получения детальной информации о товаре.
- **Сервис управления корзиной** — управляет корзинами пользователей - обеспечивает мгновенный доступ и обновление корзин. Пользователь добавляет или удаляет товары, а сервис подсчитывает итоговую сумму.
- **Сервис обслуживания заказов** — система для оформления, обработки и отслеживания заказов. Создание заказов: преобразует выбранные пользователем товары из корзины в оформленный заказ, а также фиксирует все детали (состав, стоимость, данные получателя). Интеграция с платежами: автоматически передает данные в *Платежный сервис* для проведения транзакции, обрабатывает ответы об успешной/неуспешной оплате, обновляя статус заказа.

Отслеживание заказов: предоставляет пользователю историю статусов (например: «Заказ оплачен» → «Заказ собран» → «Заказ передан в

доставку» → «Заказ доставлен»), после отправляет уведомления при изменении состояния заказа.

- **Платежный сервис** — обрабатывает платежи через интеграцию с внешними платежными шлюзами (Stripe, PayPal). После успешной/неуспешной оплаты сервис обновляет статус заказа и инициирует уведомление пользователя.
- **Сервис уведомлений** — отвечает за отправку уведомлений. Использует брокер сообщений для асинхронной отправки email/push-уведомлений/SMS о статусе заказа, успешной/неуспешной оплате и других событиях.

Брокер сообщений

Использование Kafka или RabbitMQ обеспечивает асинхронную связь между сервисами. После оплаты *Платежный сервис* передает сообщение брокеру о завершении платежа, включая ID заказа, сумму оплаты, статус платежа, дополнительные метаданные. *Сервис уведомлений* его обрабатывает, формирует персонализированное уведомление для пользователя и отправляет его через выбранные каналы (email, push-уведомление, SMS). *Сервис Обслуживания заказов* тоже передает сообщение брокеру о статусе заказа (например: «Заказ оплачен» → «Заказ собран» → «Заказ передан в доставку» → «Заказ доставлен»), после отправляет уведомления при изменении состояния заказа.

Базы данных

Каждая служба использует наиболее подходящую БД. Пользовательские данные и заказы — PostgreSQL. Такой подход повышает эффективность и масштабируемость системы.