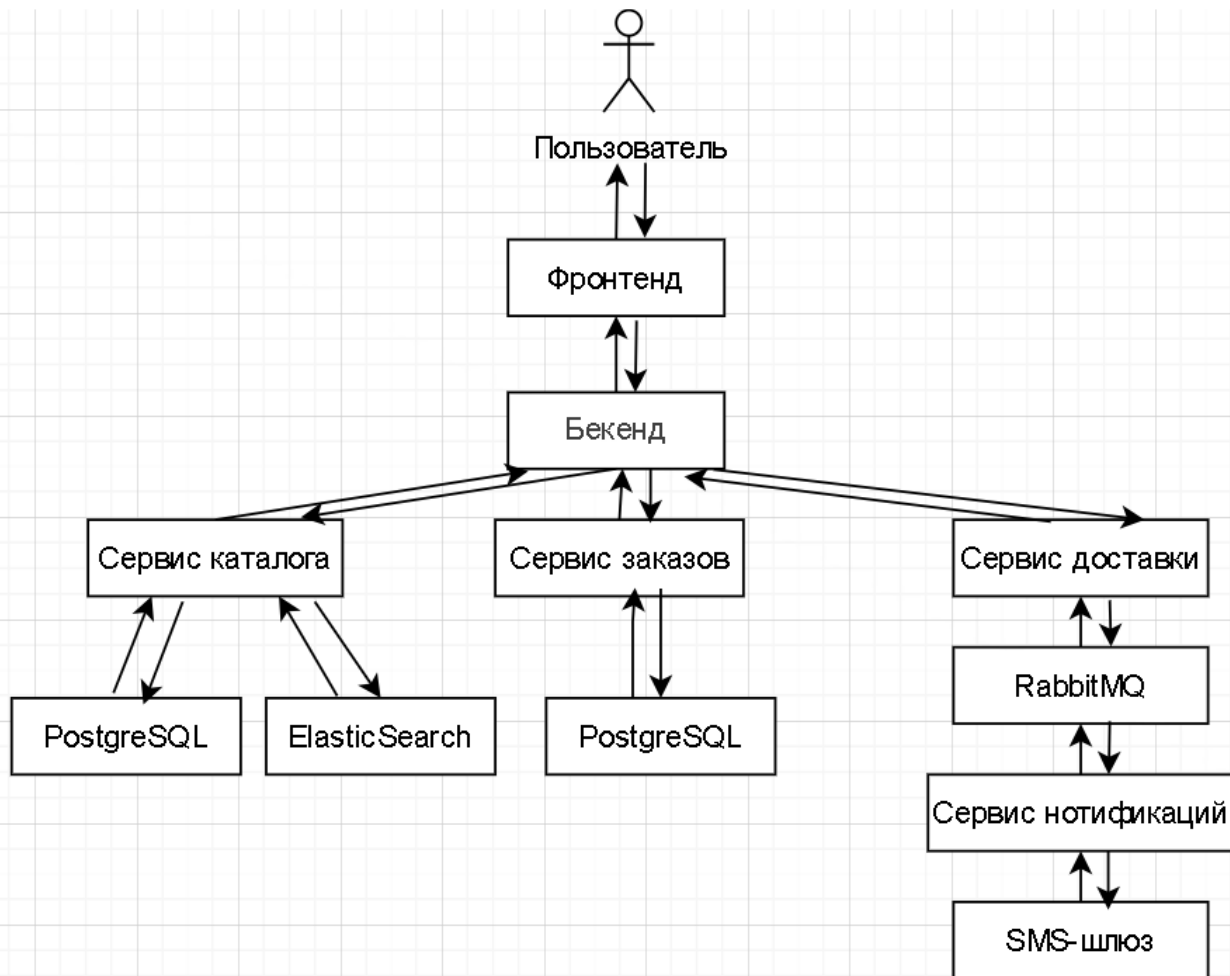


Приложение для осуществления онлайн-покупок



1. Пользователь — конечный пользователь, который взаимодействует с системой через веб-интерфейс или мобильное приложение.

2. Фронтенд-сервис представляет собой пользовательский интерфейс, через который пользователь взаимодействует с приложением

Основные функции

- Отображение товаров из каталога.
- Оформление заказов.
- Просмотр статуса доставки.

3. Бекенд-сервис — обрабатывает запросы от фронтенда и координирует работу специализированных сервисов.

4. Сервис каталога — специализированный микросервис для управления товарами.

Компоненты

- **PostgreSQL** — реляционная база данных. В ней хранятся данные, например, цены, категории.
- **ElasticSearch** — система, отвечающая за быстрый полнотекстовый поиск и фильтрацию данных.

5. Сервис заказов — отвечает за жизненный цикл заказа: от оформления до выполнения. Хранение данных осуществляется в базе данных PostgreSQL.

6. Сервис доставки — управляет логистикой: маршруты, трекинг, уведомления. Отправляет события в **RabbitMQ** (например, "заказ передан курьеру").

7. RabbitMQ (брокер сообщений) — это брокер сообщений, который позволяет приложениям обмениваться сообщениями, обеспечивая асинхронную и надёжную передачу информации

8. Сервис уведомлений

Отправляет уведомления пользователям через SMS.

- **Источник данных:** События из **RabbitMQ**.
- **Канал доставки:** **SMS-шлюз**.

9. SMS-шлюз — внешний сервис для отправки SMS. **Отвечает** за доставку текстовых уведомлений.

Взаимодействие компонентов

- Когда пользователь делает заказ через сайт или мобильное приложение (фронтенд), вся информация сначала попадает в основной сервер приложения (бекенд). Бекенд выступает как главный координатор — он проверяет данные пользователя, сохраняет заказ в базу данных PostgreSQL и передает информацию дальше другим сервисам.
- Сервис каталога в этот момент помогает уточнить детали заказанных товаров. Он использует базу данных PostgreSQL для хранения основной информации о товарах (цена, описание, наличие) и Elasticsearch для быстрого поиска по каталогу.
- После оформления заказа начинает работать сервис доставки. Как только сервис доставки определяет, какой курьер и когда привезет заказ, он отправляет сообщение в специальную систему обмена сообщениями RabbitMQ.
- RabbitMQ работает как промежуточное звено между разными частями системы. Когда сервис доставки кладет в него сообщение, сервис уведомлений сразу получает это сообщение и начинает готовить SMS для клиента. При этом сами сервисы не связаны напрямую, они даже не знают о существовании друг друга, общаясь только через RabbitMQ. Это делает систему более надежной: если сервис уведомлений временно не работает, сообщения просто будут ждать в очереди, а не потеряются.
- Сервис уведомлений, получив информацию о заказе, обращается к внешнему SMS-шлюзу, который непосредственно отправляет сообщение клиенту.

Каждый сервис отвечает только за свою часть работы: каталог — за информацию о товарах, заказы — за обработку покупок, доставка — за логистику, а уведомления — за общение с клиентом. Такое разделение функций делает систему гибкой и надежной.

