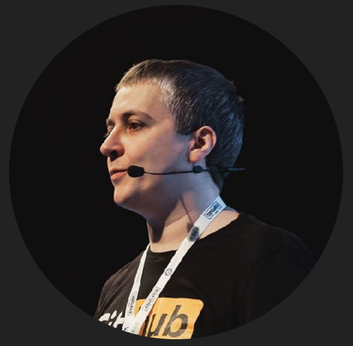


Рустам Курамшин
Архитектор системы

Александр Янчий
Архитектор данных

Рустам Гулямов
Архитектор интеграции

КОМАНДА «JAVA BOYS»

ЗАДАЧА «WEB-СЕРВЕР»

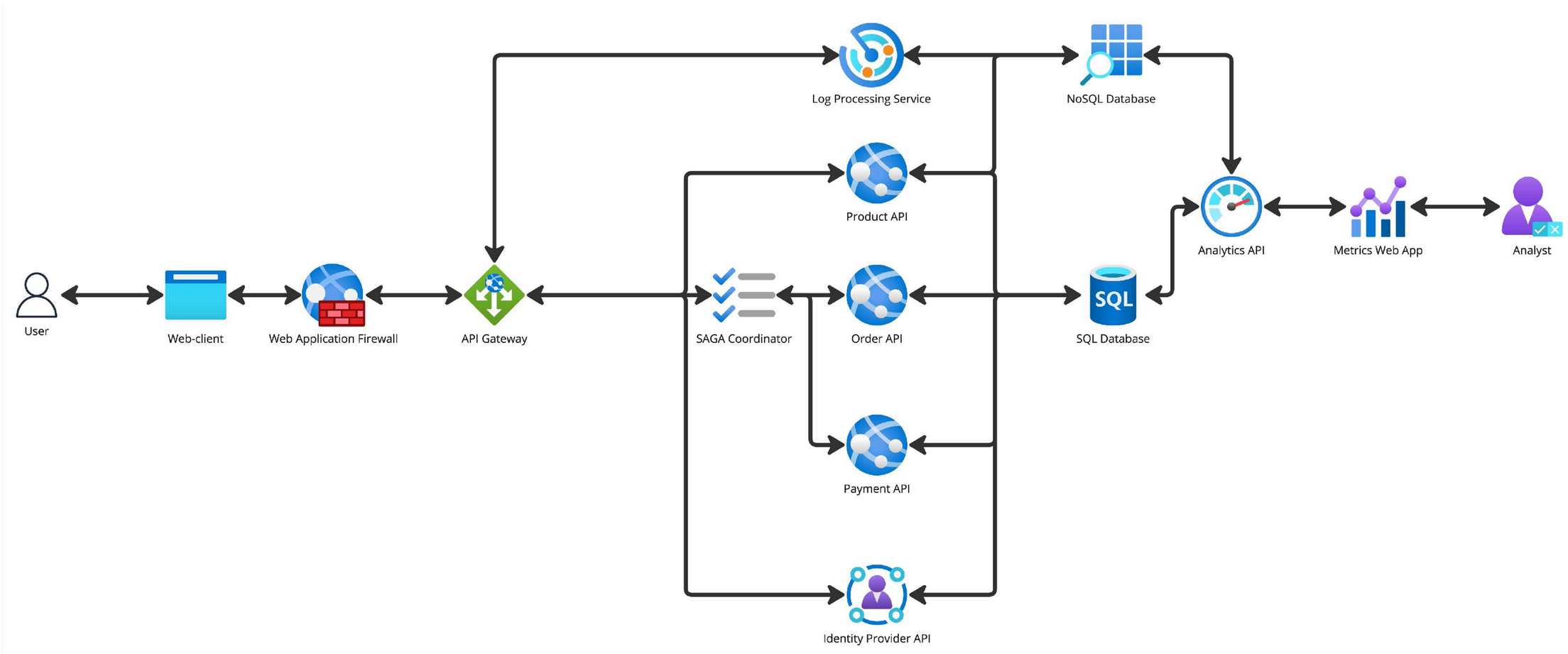
ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ (ФТ)

- Управление товарами (Product API): Добавление, обновление, удаление и просмотр товаров. Поиск товаров по различным критериям, включая категории, цену и наличие на складе.
- Управление заказами (Order API): Создание заказа с возможностью добавления нескольких товаров в корзину. Отображение статуса заказа (например, обработан, отправлен, доставлен). Отмена и изменение заказа до его финализации.
- Платежи (Payment API): Интеграция с внешними платежными системами для обработки онлайн-платежей. Обработка статусов платежей и обеспечение безопасности транзакций.
- Аутентификация и управление пользователями (Identity Provider API): Регистрация новых пользователей и аутентификация с выдачей токенов. Управление профилями пользователей, включая изменение пароля и контактной информации.
- Логирование и обработка логов (Log Processing Service): Автоматический сбор логов с API Gateway/реверс-прокси и их анализ. Отслеживание действий пользователей и посещенных веб-страниц.
- Аналитика (Analytics API и Metrics Web App): Сбор данных о посещаемости страниц, взаимодействиях с продуктами и покупках. Визуализация данных через дашборды для анализа эффективности маркетинговых кампаний и пользовательского поведения.

НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ (НФТ)

- Производительность: Система должна обрабатывать транзакции и запросы к API в реальном времени с минимальной задержкой. Способность обрабатывать пиковые нагрузки, особенно в периоды максимальной активности покупателей.
- Масштабируемость: Система должна поддерживать горизонтальное масштабирование для обслуживания увеличивающегося числа пользователей и данных. Автоматическое масштабирование компонентов системы при необходимости.
- Отказоустойчивость: Разработка механизмов для быстрого восстановления после сбоев. Резервное копирование данных и регулярное тестирование восстановления системы.
- Безопасность: Шифрование данных, передаваемых через интернет и хранящихся в базах данных. Реализация политик безопасности для защиты данных пользователей и системы.
- Удобство использования: Интуитивно понятный интерфейс для пользователей всех возрастных групп. Доступность системы через мобильные устройства и настольные компьютеры.
- Совместимость: Поддержка всех основных браузеров и мобильных платформ. Взаимодействие с различными внешними платежными системами и сервисами.

ИНТЕРНЕТ-МАГАЗИН И СИСТЕМА WEB-АНАЛИТИКИ



ОПИСАНИЕ КОМПОНЕНТОВ

- **Web-client** - веб-браузер или мобильное приложение. Позволяет работать с каталогом интернет-магазина (фильтрация и просмотр списка товаров, добавление в корзину, оформление/оплата заказов и доставка).
- **Web Application Firewall (WAF)** - межсетевой экран для блокировки сетевых атак и DDoS.
- **API Gateway** - API шлюз/реверс-прокси, точка входа в бэкенд-сервисы интернет-магазина.

ОПИСАНИЕ КОМПОНЕНТОВ

- **Product API** - микросервис доменной зоны «Товары». Отвечает за просмотр списков/фильтрацию товаров, просмотр товара, поиск товаров по характеристикам и поисковым запросам.
- **Order API** - микросервис доменной зоны «Заказы». Отвечает за корзину товаров, оформление заказа.
- **Payment API** - микросервис платежного шлюза. Отвечает за интеграцию с платежными системами и проведение платежей.

ОПИСАНИЕ КОМПОНЕНТОВ

- **Identity Provider API** - микросервис отвечающий за IAM (аутентификацию и контроль доступа). В том числе управляет доменной зоной «Пользователи».
- **SAGA Coordinator** - микросервис отвечающий за управления распределенными транзакциями между микросервисами. Например, при оформлении заказа и его оплате.
- **Log Processing Service** - сервис решающий задачи парсинга логов реверс-прокси (API-шлюза) для выделения данных о URL-адресах страниц, IP-пользователя, штампа времени и т.п.

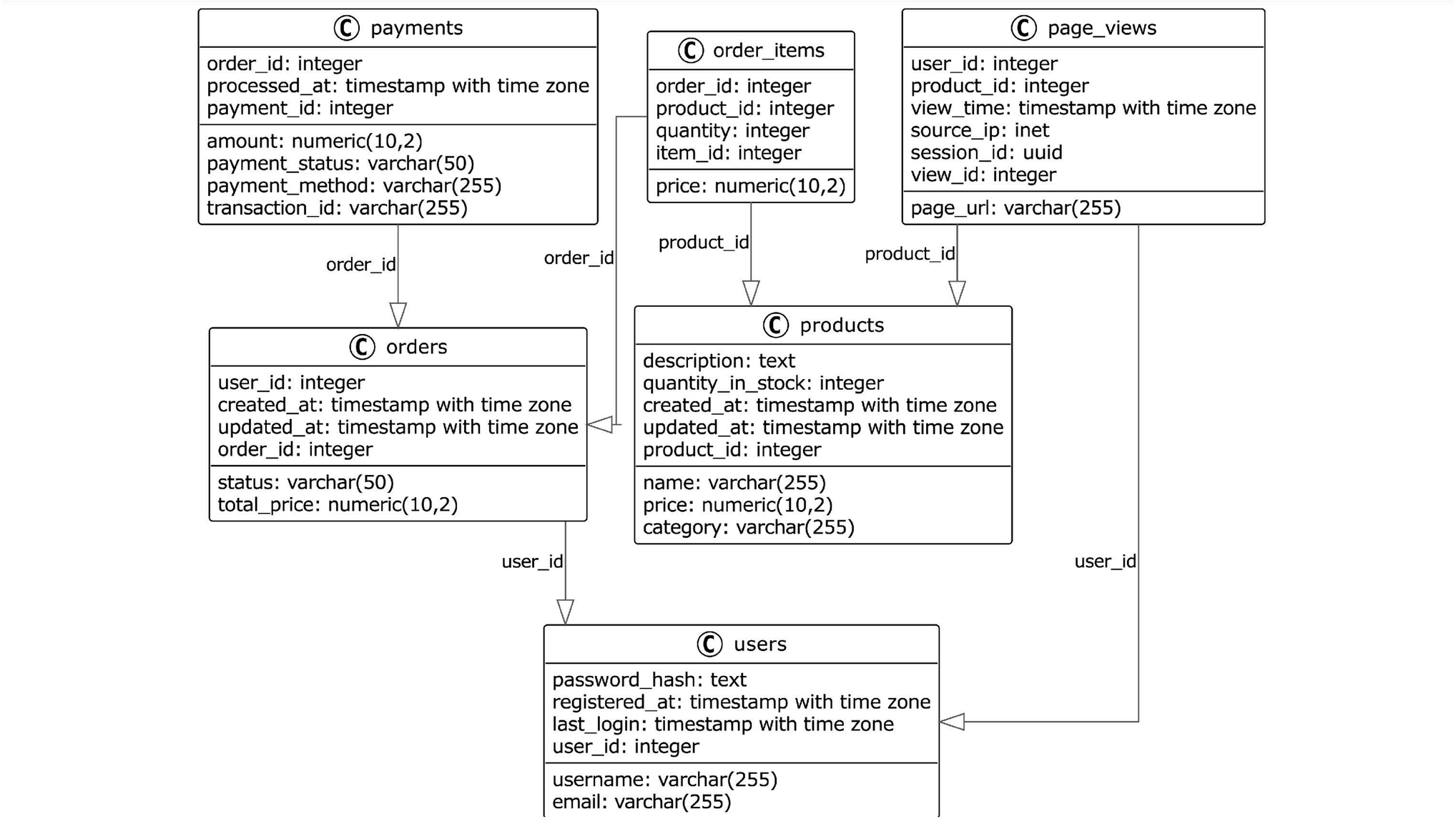
ОПИСАНИЕ КОМПОНЕНТОВ

- **SQL Database** - сервер управления реляционными базами данных. Хранит данные бизнес-доменов системы в табличном представлении.
- **NoSQL Databas**e - сервер NoSQL СУБД работающий с документами (json). В него реплицируются данные о товарах. Используется для поиска товаров. Также в него сохраняются данные выделенные сервисом Log Processing Service, что позволяет строить запросы к данными временных рядов лог-записей.
- **Analytics API** - микросервис веб-аналитики. Получает данные о просмотренных страницах, товарах на этих страницах.

ОПИСАНИЕ КОМПОНЕНТОВ

- **Metrics Web App** - веб-приложение позволяющее строить дашборды с графиками на основе данных из Analytics API. Позволяет просматривать статистику посещения веб-страниц интернет-магазина.

АРХИТЕКТУРА ДАННЫХ



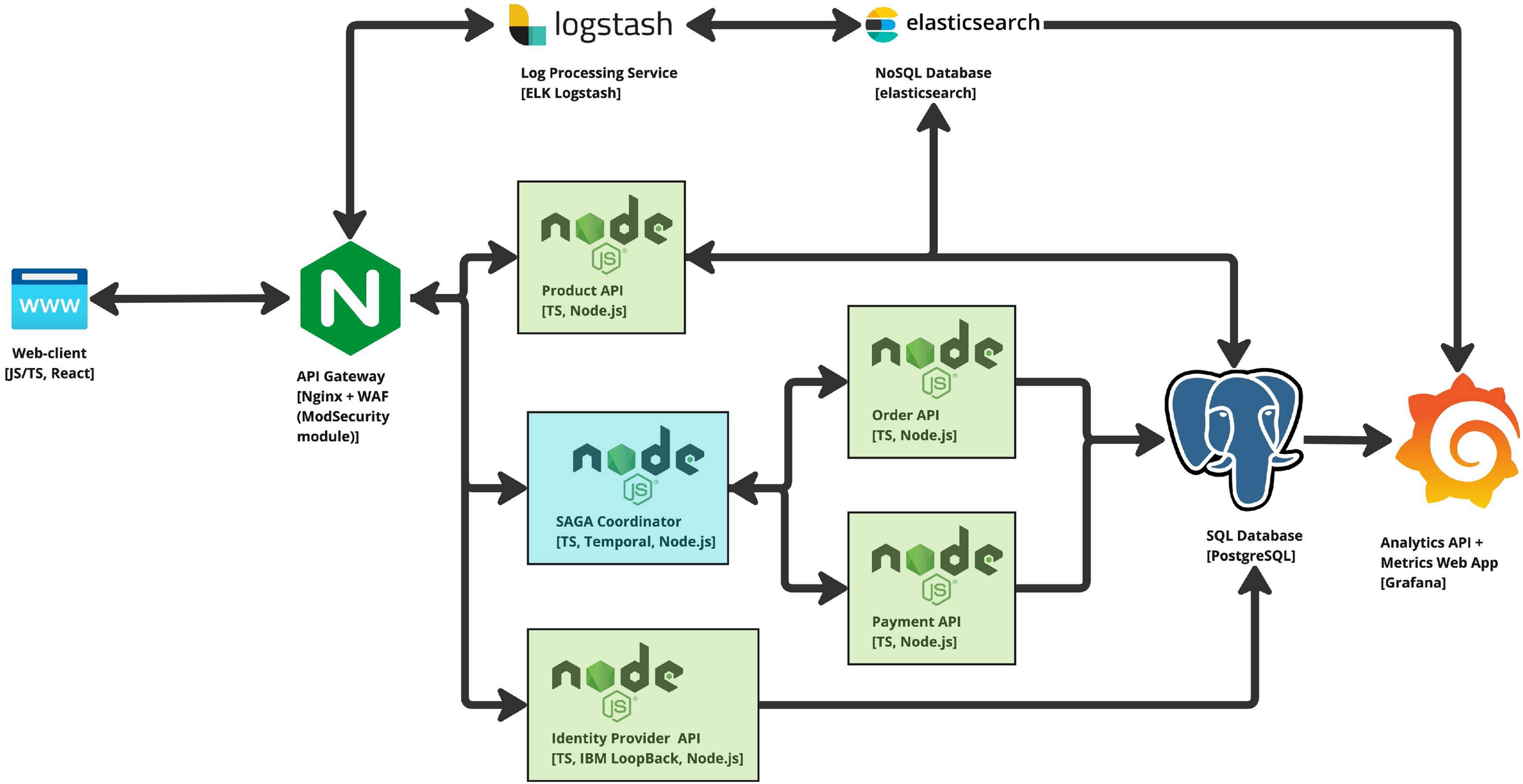
ОПИСАНИЕ СУЩНОСТЕЙ

- Products - Хранение информации о товарах в каталоге (описание, цена, количество на складе и т.д.).
- Orders - Управление заказами клиентов (статус заказа, общая стоимость и т.д.).
- Order Items - Хранение информации о каждом товаре в заказе (количество, цена за единицу и т.д.).
- Payments - Обработка платежей по заказам (сумма платежа, статус платежа, метод оплаты и т.д.)

ОПИСАНИЕ СУЩНОСТЕЙ

- Users - Управление информацией о пользователях (имя пользователя, хэш пароля, электронная почта и т.д.).
- Page Views - Аналитика по просмотрам страниц (время просмотра, URL страницы, IP-адрес пользователя и т.д.)

АРХИТЕКТУРА СИСТЕМЫ



PRODUCT API

- GET /products - Получение списка всех товаров.
- POST /products - Добавление нового товара в каталог.
- GET /products/{id} - Получение детальной информации о товаре по его ID.
- PUT /products/{id} - Обновление информации о товаре.
- DELETE /products/{id} - Удаление товара из каталога.

ORDER API

- POST /orders - Создание нового заказа.
- GET /orders/{id} - Получение информации о конкретном заказе.
- PUT /orders/{id} - Обновление информации о заказе (например, изменение статуса).
- DELETE /orders/{id} - Отмена заказа.

PAYMENT API

- POST /payments - Инициация платежа для заказа.
- GET /payments/{id} - Получение информации о статусе платежа.
- PUT /payments/{id} - Обновление статуса платежа (например, подтверждение платежа).

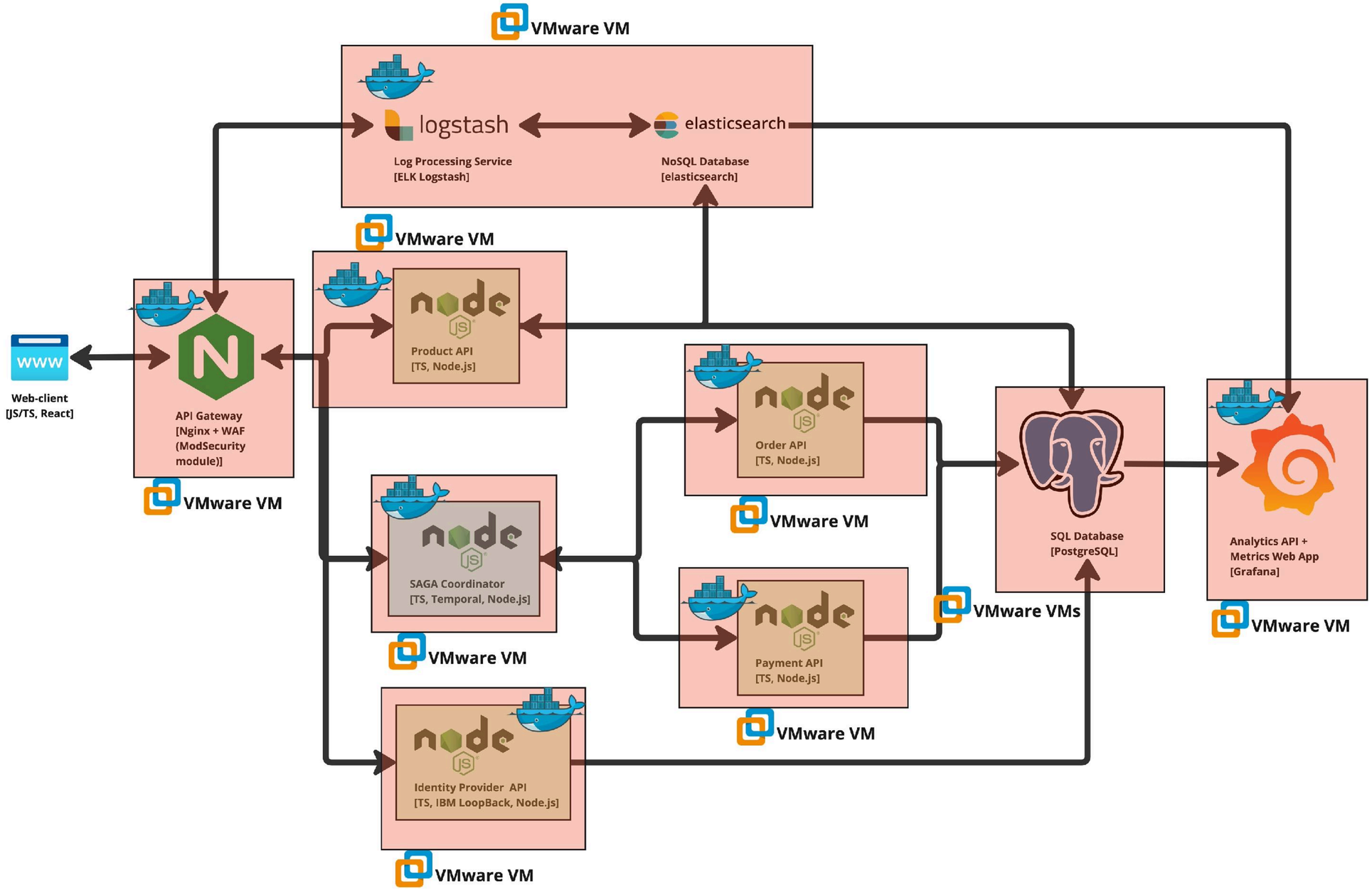
IDENTITY PROVIDER API

- POST /users/register - Регистрация нового пользователя.
- POST /users/login - Аутентификация пользователя и выдача токена.
- GET /users/{id} - Получение информации о пользователе.
- PUT /users/{id} - Обновление профиля пользователя.

ANALYTICS API И METRICS WEB APP

- GET /analytics/report - Генерация отчетов на основе собранных данных.
- GET /metrics/dashboard - Получение дашборда с графиками и аналитикой.
- POST /metrics/custom-report - Создание пользовательских отчетов по заданным параметрам.

АРХИТЕКТУРА РАЗВЕРТЫВАНИЯ



ИНФРАСТРУКТУРА

- Web-client - браузер или мобильное приложение.
- Web-server - реверс-прокси на основе Nginx и WAF-модуля ModSecurity. Прокидывает HTTPS-трафик клиента к бэкэнд сервисам.
- Виртуальные хосты VMware с Docker - в них в Docker-контейнерах разворачиваются API-сервисы доменных зон (товары, заказы, оплата). Координатор SAGA реализованный с помощью библиотеки Temporal. Identity Provider реализует аутентификацию за счет JS-фреймворка LoopBack. Пайплайн парсинг access-логов Nginx через Logstash и сохранения их данных, и индексирования в Elasticsearch.
- Кластер PostgreSQL развернутый на виртуальных машинах или Bare Metal.
- Веб-приложение Grafana развернутое на отдельном виртуальном хосте в контейнере.


JAVA BOYS
JAVA
1 0 1 1 1 0 1 1 1 0 1