



Факультет
компьютерных наук

Программная
инженерия

Москва
2026

Сервис для автоматического управления ценами на маркетплейсах

Студент: Зуев Аким Иванович

Руководитель: Смирнов Никита Сергеевич, преподаватель ДПИ ФКН НИУ ВШЭ



Сервис для автоматического управления ценами на маркетплейсах

Описание предметной области

RepricerX — веб-приложение для автоматизации ценообразования продавцов на маркетплейсах Wildberries и Ozon. Сервис подключается к магазинам по API, отслеживает рынок и обновляет цены по выбранной стратегии без вмешательства продавца.

Общие ожидания от решения:

- Автоматическая переоценка сотен SKU без участия пользователя
- Единый интерфейс для управления магазинами на разных маркетплейсов
- Прозрачный журнал изменений и ошибок интеграции
- Гибкие ограничения (min/max, шаг, маржа) под политику продавца
- Симуляция расчёта до отправки цен в маркетплейсы



~300 тыс. потенциальных
пользователей

средние и крупные селлеры WB и Ozon с
каталогом 20+ SKU



Архитектура

клиент-серверное приложение



2 интеграции

Wildberries и Ozon через единый
адаптер



4 стратегии

Ценообразования с гибкими
ограничениями

wildberries
x
OZON



Сервис для автоматического управления ценами на маркетплейсах

Актуальность работы

13,4 трлн ₽

объём e-commerce РФ в 2025 году (+19%), прогноз 2026 — свыше 15 трлн ₽

Источник: Data Insight, «Интернет-торговля в России 2025»

81%

всех онлайн-заказов в РФ приходится на маркетплейсы (+2 п.п. за год). Суммарный GMV Wildberries и Ozon в 2025 — 10,3 трлн ₽

Источник: Data Insight, «годовые отчёты Wildberries и Ozon 2025»

10+

репрайсеров

уже работают на рынке (MPSTATS, MP Manager, Indeera, MarketGuru и др.), тарифы — от 1 900 до 19 000 ₽/мес.

~140 тыс.

целевых селлеров

средний бизнес на WB и Ozon (доход 1-3 млн ₽/мес) вырос с 5% до 11% от 1,26 млн селлеров. Микробизнес уходит, рынок концентрируется на профессиональных продавцах

Источник: Data Insight, «исследование селлеров 2025»



Цели и задачи проекта

Цели:

Спроектировать и реализовать веб-приложение для автоматизированного управления ценами продавцов на маркетплейсах Wildberries и Ozon, обеспечивающее расчёт цен по заданным стратегиям и отправку обновлений через API маркетплейсов без участия пользователя.

Задачи:

- Спроектировать модульную архитектуру с единым интерфейсом маркетплейса, поддерживающую расширение на новые площадки без правок основного кода.
- Реализовать движок ценообразования с четырьмя стратегиями расчёта и системой ограничений, гарантирующей корректность итоговой цены.
- Реализовать интеграции с API Wildberries и Ozon, обеспечив устойчивость к сбоям через rate-limit, ретраи и фоновую обработку длительных операций.
- Обеспечить безопасность данных и возможность удобного развертывания.



Сервис для автоматического управления ценами на маркетплейсах

Сравнительный анализ аналогов

Параметр	RepricerX	MPSTATS	Indeera
Маркетплейсы	WB, Ozon	WB, Ozon, ЯМ	WB, Ozon, ЯМ
Стратегии ценообразования	4	2	по 30+ показателям
Ограничения (min/max, шаг, маржа, интервал)	все	частично (без max-шаг)	все
Симуляция перед отправкой	есть	нет	нет
Лимит SKU	без жёсткого ограничения	500 базово, до 3 000 индивидуально	под крупные каталоги
Журнал изменений + выгрузка CSV	180 дней + CSV	детализация по времени	финансовые отчёты
Интеграция с 1С / МойСклад	нет	нет	через API
Защита от автоприменения акций МП	нет	нет	да



Сервис для автоматического управления ценами на маркетплейсах

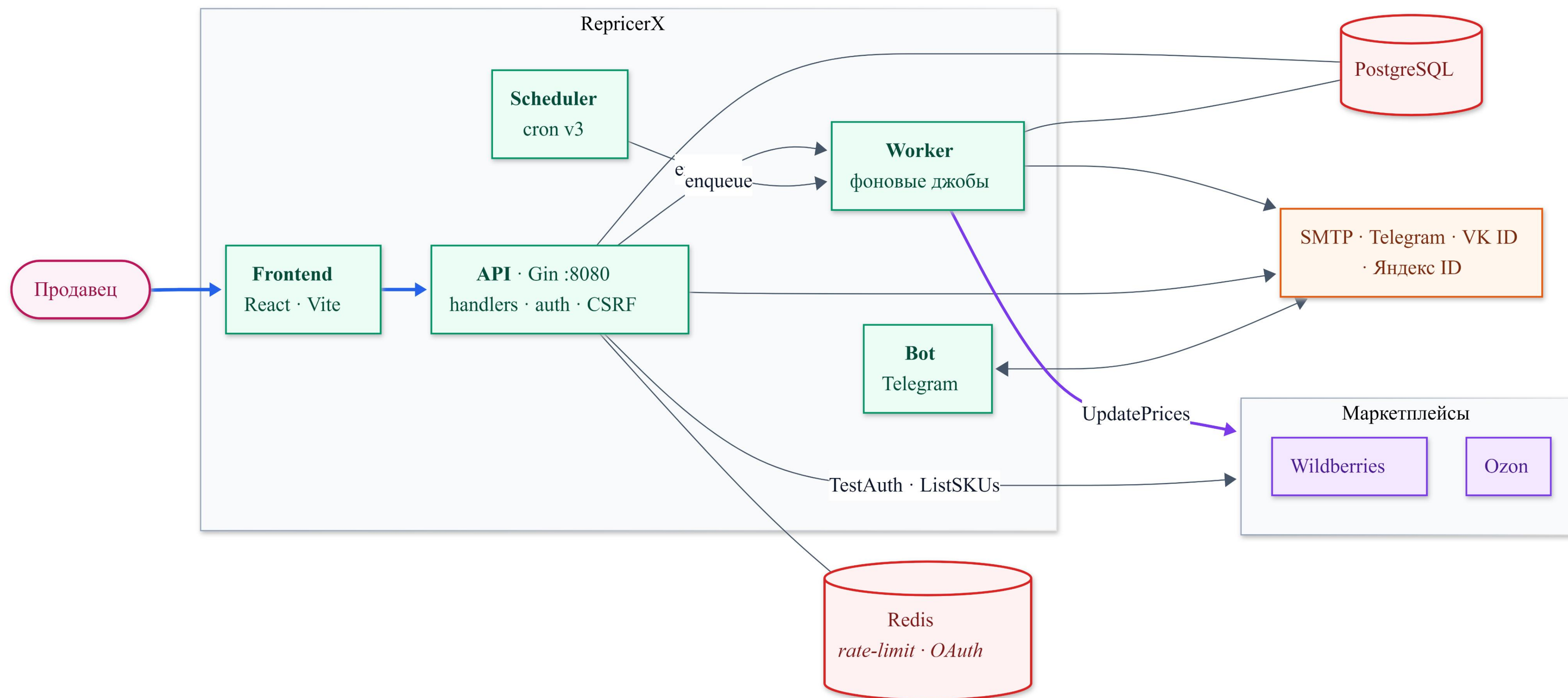
Функциональные требования

1. Регистрация и аутентификация пользователей по email/паролю и через сторонние сервисы (VK ID, Яндекс ID).
2. Подключение магазинов на маркетплейсах Wildberries и Ozon по API-ключу с автоматической проверкой подключения.
3. Импорт каталога товаров из подключённого магазина и ручное добавление SKU.
4. Настройка ценовых ограничений по каждому товару — минимальная и максимальная цена, себестоимость, минимальный интервал обновления.
5. Создание стратегий ценообразования четырёх типов — медиана конкурентов, минимум конкурентов + шаг, фиксированная маржа, фиксированная цена — с настраиваемыми параметрами и fallback-политикой.
6. Назначение стратегий на отдельные товары или группы по фильтру с управлением приоритетами.
7. Симуляция расчёта цен по выбранной стратегии без отправки в маркетплейс — отображение плана изменений до применения.
8. Автоматический пересчёт и отправка обновлённых цен в маркетплейсы по расписанию или ручному запуску с учётом лимитов API и автоматическими ретраями при ошибках.
9. Ведение журнала всех изменений цен с фильтрами по магазину, товару, статусу и периоду; экспорт результатов в формате CSV.
10. Ведение журнала интеграционных ошибок и отображение статуса подключения каждого магазина.
11. Уведомления пользователя о результатах массовых операций и ошибках интеграции.



Сервис для автоматического управления ценами на маркетплейсах

Архитектура приложения





Сервис для автоматического управления ценами на маркетплейсах

Средства разработки и их обоснование

Backend

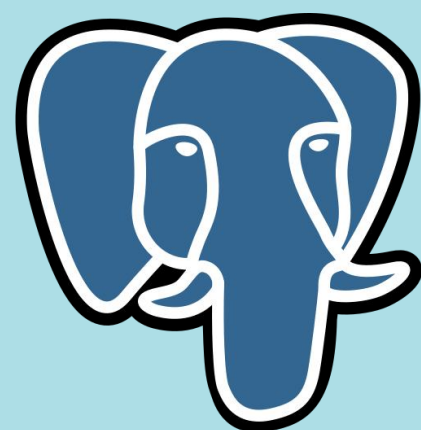


- компилируемый
- статическая типизация
- нативные goroutines

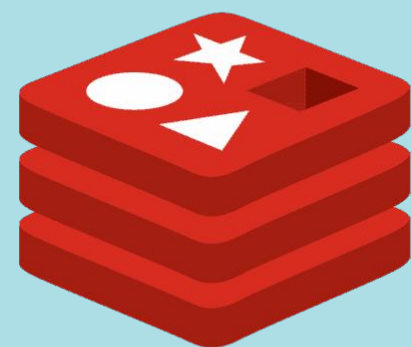
Инфра



Хранилища

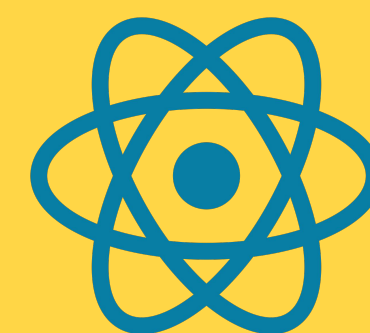


- чек-ограничения для журналов аудита
- ACID + FK
- JSONB для гибких стратегий
- GIN/BRIN-индексов



- In-memory KV с TTL
- rate-limit
- хранилище OAuth state + PKCE-verifier
- дешёвый health-check

Frontend

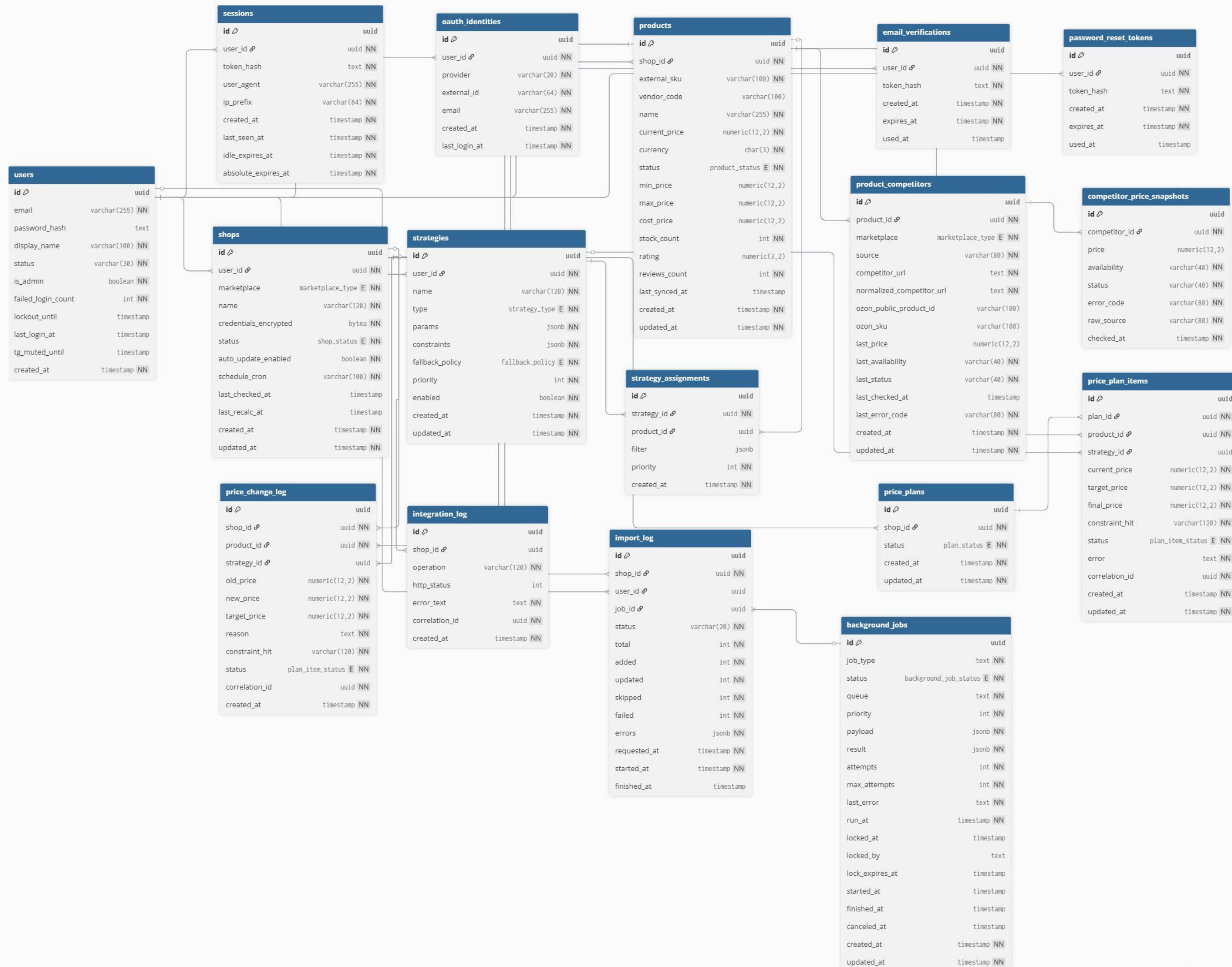


- статическая типизация в UI
- общий язык DTO с бэком
- крупнейшая экосистема,
- асинхронная загрузка данных
- мгновенный HMR
- оптимизированный prod-бандл



Сервис для автоматического управления ценами на маркетплейсах

Структура базы данных и ERD-диаграмма





Сервис для автоматического управления ценами на маркетплейсах

Описание разработанных сервисов и алгоритмов



Сервис для автоматического управления ценами на маркетплейсах

Pricing — движок ценообразования

Является сердцем системы автоматического репрайсинга. Сервис отвечает за управление экземплярами вспомогательных компонентов — `calculator` (формулы стратегий), `competitor` (рыночные данные) и `dispatcher` (отправка в маркетплейс), — которые и выполняют всю работу над ценами.

Исходя из текущего состояния назначенных на товары стратегий (`strategy_assignments`) и собранных конкурентных данных (`product_competitors`), он вычисляет целевую цену по одной из четырёх формул — медиана конкурентов –

$X\%$, $\min(\text{конкурент}) + \text{шаг}$, $\text{cost} \times (1 + Y\%)$ или фиксированная цена, — затем применяет бизнес-ограничения (`min_price`, `max_price`, `max_change_pct`, `min_interval_minutes`) и сохраняет результат в `price_plans` /

`price_plan_items` с уникальным `correlation_id`, гарантирующим идемпотентность последующей отправки.

Для асинхронного исполнения плана используется встроенная очередь PostgreSQL — таблица `background_jobs` с `pg advisory locks`: сервис ставит задачу типа `price_dispatch`, а `worker`-процесс забирает её через `ClaimNext` и

через интерфейс `integration.Marketplace` отправляет финальные цены в Wildberries или Ozon.



Сервис для автоматического управления ценами на маркетплейсах

Dispatcher — отправка цен в маркетплейс

Является исполнителем уже рассчитанного плана цен. Сервис отвечает за управление экземплярами адаптеров маркетплейсов (`integration.Marketplace` для Wildberries и Ozon), которые и выполняют всю работу по

обновлению цен через REST API продавцов.

Исходя из текущего состояния плана `price_plans` (`status = pending`), он расшифровывает `credentials` магазина (AES-256-GCM), нарезает позиции `price_plan_items` на чанки (по умолчанию 100 SKU), пропускает каждый чанк через `per-shop token-bucket` (5 rps по умолчанию) и отправляет в маркетплейс. По итогу финализирует статус каждой позиции (`applied` / `skipped` / `failed`), пишет успешные изменения в `price_change_log` (хранение 180 дней) и ошибки интеграции в `integration_log` (30 дней) — всё со сквозным `correlation_id` для трассировки.

При сбоях применяется политика ретраев из T3 4.1.1.7.4: ошибка `ErrUnauthorized` → терминальный `failed` без повторов; временные ошибки (сеть, `5xx`, `ErrRateLimited`) → `exponential backoff` 30 с → 60 с → 120 с через ту же очередь `background_jobs`. После 3 попыток оставшиеся `pending`-позиции принудительно маркируются `failed` (`MarkExhausted`), а `notifier` эмитит событие `dispatch_completed` пользователю



Сервис для автоматического управления ценами на маркетплейсах

Auth — аутентификация и сессии

Является охранником входа в систему. Сервис отвечает за управление учётными записями, паролями и сессиями пользователей, привлекая для криптографии и доставки писем вспомогательные пакеты — `pkg/password`

(`argon2id`), `pkg/token` (генерация и SHA-256-хеширование), `pkg/mailer` (SMTP), а также адаптеры `integration/oauth/{vkid,yandex}` для входа через VK ID и Яндекс ID.

Исходя из текущего запроса (`Register`, `Login`, `Logout`, `VerifyEmail`, `ResetPassword`, `OAuthCallback`), он хеширует пароль через `Argon2id` (64 МБ памяти, 3 итерации) и записывает запись в `users`; при логине — сравнивает хеш в постоянное время (`subtle.ConstantTimeCompare`), а для несуществующих email прогоняет `fake-hash`, чтобы не утечь информацию по таймингу. Успешная аутентификация создаёт строку в `sessions` с двойным TTL по OWASP — `idle_expires_at` (24 ч неактивности) и `absolute_expires_at` (7 суток жёсткий лимит), — где в БД лежит только `sha256(token)`, а `plaintext` уходит клиенту в `HttpOnly-cookie rx_session`.

Для OAuth-входа реализован серверный `Authorization Code Flow` с PKCE S256: `state` и `code_verifier` хранятся в `Redis` с TTL 10 минут (`pkg/oauthstate`); при конфликте email с существующим паролем аккаунтом сервис возвращает одноразовый `link_token` и редиректит на `/link-oauth` для подтверждения паролем. Защита от перебора — экспоненциальный лок-аут (`failed_login_count + lockout_until`) и `rate-limit` на уровне HTTP-middleware через `Redis` (`pkg/redislimit`).



Сервис для автоматического управления ценами на маркетплейсах

Shop — магазины и подключение к маркетплейсам

Является точкой подключения продавца к маркетплейсу. Сервис отвечает за CRUD магазинов (shops) и за безопасное хранение API-ключей Wildberries и Ozon, используя для этого вспомогательные пакеты — pkg/crypto

(шифрование AES-256-GCM) и фабрики адаптеров integration/wildberries и integration/ozon, которые и выполняют всю работу по проверке подлинности ключей через API маркетплейса.

Исходя из текущего запроса (Create, Update, Delete, TestConnection, List), он валидирует структуру credentials под выбранный маркетплейс (для WB — {api_key}, для Ozon — {client_id, api_key}), шифрует их через crypto.Encrypt (ключ из APP_SECRET_KEY) и кладёт в shops.credentials_encrypted как BYTEA — plaintext в БД никогда не попадает. При TestConnection сервис строит клиент через нужную MarketplaceFactory, дёргает TestAuth (для WB — /discounts-prices-api/ping, для Ozon — /v1/category/tree) и атомарно обновляет статус магазина — active при успехе или error при провале — параллельно записывая результат в integration_log со сквозным correlation_id.

Ошибки маркетплейса мапятся в типизированные сервисные: integration.ErrUnauthorized → ErrAuthFailed (HTTP 401 пользователю), ErrRateLimited → 429, прочие 5xx → ErrMarketplaceUnavailable (HTTP 502). При удалении магазина срабатывает защита из ТЗ: если за последние 30 дней в price_change_log есть записи — Delete отклоняется, чтобы не разорвать аудит-цепочку, и операция требует явного «архивирования» через status = disabled.



Сервис для автоматического управления ценами на маркетплейсах

Демонстрация



Результаты работы

1. Проведён анализ предметной области и сравнительный анализ существующих репрайсеров для маркетплейсов (Indeera, MPSTATS).
2. Спроектирована модульная архитектура с единым интерфейсом маркетплейса, позволяющая подключать новые площадки без изменения бизнес-логики.
3. Разработано полноценное клиент-серверное веб-приложение: серверная часть на Go (процессы api и worker, PostgreSQL, Redis) и клиентская часть на React + TypeScript.
4. Реализованы интеграционные адаптеры с API Wildberries и Ozon: проверка подключения, импорт каталога SKU, отправка обновлений цен с rate-limit и автоматическими ретраями.
5. Реализован движок ценообразования с четырьмя стратегиями расчёта и системой ограничений (min/max цена, максимальный шаг изменения, фиксированная маржа, минимальный интервал между обновлениями).
6. Реализованы система безопасности (AES-256-GCM шифрование API-ключей, серверные сессии, CSRF-защита, OAuth через VK ID и Яндекс ID), журнал изменений цен и журнал интеграционных ошибок с экспортом в CSV.
7. Приложение демонстрирует полный сценарий автоматизации: от подключения магазина и импорта каталога до автоматической отправки рассчитанных по стратегии цен в маркетплейсы.



Сервис для автоматического управления ценами на маркетплейсах

Направления дальнейшей работы

- Расширение списка поддерживаемых маркетплейсов — подключение Яндекс Маркета и СберМегаМаркета через тот же интерфейс-адаптер, без изменений бизнес-логики.
- Интеграция с учётными системами 1С и МойСклад для двустороннего обмена данными о товарах, остатках и ценах (есть у Huckster).
- Добавление защиты от автоматического применения акций маркетплейсов с настройкой пороговых условий маржи и стоп-цены (есть у Indeera и Huckster).
- Расширение библиотеки стратегий ценообразования — следование за позицией в выдаче (BuyBox), управление по оборачиваемости остатков, расписание по часам и дням недели.
- Реализация кросс-маркетплейсного ценообразования — единая цена товара на разных площадках с учётом комиссий, СПП и доставки.
- Развитие модуля аналитики и отчётности — встроенный калькулятор юнит-экономики, динамика прибыли по SKU, ABC-анализ ассортимента, интерактивные дашборды.
- Поддержка управления несколькими кабинетами одного маркетплейса в едином интерфейсе с фильтрацией по магазинам.
- Разработка мобильного приложения (iOS, Android) для оперативного управления ценами и просмотра журнала.



Сервис для автоматического управления ценами на маркетплейсах

Спасибо за внимание!

