Sellersflare: разбор вложения и первичный ресерч

Дата: 2026-05-23

Источник: `/Users/mikhailkozyrev/Downloads/Telegram Desktop/sellersflare.docx`

Статус: исследование перед разработкой

1. Что находится во вложении

Файл `sellersflare.docx` создан 2026-05-23, содержит 320 текстовых абзацев, без таблиц, изображений и embedded media.

Содержательно это не ТЗ на немедленную разработку, а набор из двух Codex-ready промтов и рамка проверки гипотез:

- этап 0: сначала проверять ценность через GPT, затем писать код;
- промт 1: простой risk engine для карточек WB / Ozon;
- промт 2: платформа уровня "защитный слой для селлеров маркетплейсов" с core engine, enforcement intelligence database, distribution layer, growth loops и freemium;
- блок тестирования: sanity check, adversarial test, usefulness test, product clarity test.

Главная мысль файла правильная: до разработки нужно проверить, умеет ли система отличать реальный риск от паники и помогает ли продавцу принять безопасное действие.

2. Перевод идеи в продуктовую рамку

Рабочее название: Sellersflare.

Не публичное позиционирование: "юридический бот", "гарантия отсутствия нарушений", "защита от исков".

Публичное позиционирование для MVP: "предварительная проверка карточки товара перед публикацией: риск-скоринг, объяснение триггеров и список документов/правок, которые стоит проверить".

Целевая аудитория первой версии:

- продавцы WB / Ozon с небольшим каталогом;
- агентства, которые заводят карточки для клиентов;
- селлеры в категориях с высоким риском: fashion, beauty, electronics, accessories, tools, toys, auto;
- продавцы параллельного импорта и ресейла, где важны документы происхождения товара.

Основной job-to-be-done:

> *Перед публикацией карточки понять, не выглядит ли она как риск по товарному знаку, фото, описанию или категории, и что исправить до модерации/жалобы.*

3. Выводы внешнего ресерча

1. Риск не выдуман. Wildberries описывает "Цифровой арбитраж" как досудебный сервис по вопросам интеллектуальной собственности между правообладателем и продавцом, а также упоминает последствия повторных жалоб для карточек/бренда/категории.

2. Ozon в договорных материалах для продавцов указывает, что при запросе правообладателя может запросить документы, подтверждающие право продажи/использования объектов ИС, и до предоставления документов блокировать карточку товара.

3. Обе площадки имеют API-поверхность, пригодную для загрузки/получения данных карточек. Для MVP достаточно ручного ввода или импорта JSON/CSV; интеграции с WB/Ozon API стоит делать после доказательства ценности.

4. Роспатент/ФИПС дают официальные источники для самостоятельного поиска товарных знаков. Это полезно как reference-data слой, но автоматическая правовая оценка сходства до степени смешения не должна подаваться как финальный юридический вывод.

5. Судебная практика и публичные новости по искам к продавцам маркетплейсов показывают, что enforcement может быть дорогим для селлера. Это усиливает ценность early-warning продукта, но одновременно повышает требования к дисклеймерам и качеству объяснений.

4. MVP, который стоит проверять первым

MVP должен быть не "платформа", а проверка одной карточки:

1. Пользователь вводит:

- marketplace: `wb` или `ozon`;
- название товара;
- описание;
- бренд;
- категорию;
- ссылки на изображения или заглушки;
- флаг: оригинальный товар / ресейл / собственный бренд / неизвестно.

2. Система возвращает:

- общий `risk_score` 0-100;
- `risk_level`: LOW / MEDIUM / HIGH;
- breakdown: trademark, image, content, category, documents;
- uncertainty: что неизвестно и почему результат нельзя считать юридическим заключением;
- seller actions: что проверить/исправить/подготовить.

3. На старте risk engine можно сделать rules-first:

- fuzzy match бренда и названия;
- словарь риск-категорий;
- признаки "реплики", "аналог", "совместимо с", "в стиле";
- наличие/отсутствие бренда и документов;
- похожесть описания на известные шаблоны только если есть безопасный корпус;
- image-risk пока как чек-лист, без обещания реального поиска копий.

4. LLM использовать как explanation layer, а не как единственный арбитр:

- классифицировать сигналы;
- формулировать понятное объяснение;

- предлагать безопасные next steps;
- явно писать uncertainty.

5. Что не делать в первой версии

- Не делать вывод "нарушение есть".
- Не обещать защиту от блокировок, претензий или исков.
- Не парсить чужие карточки и изображения без проверки правил источника.
- Не строить browser extension до проверки standalone-flow.
- Не делать paid freemium до первых 10-20 ручных проверок.
- Не хранить документы поставщиков без отдельной модели безопасности.
- Не копировать тексты/дизайн внешних legal-tech, marketplace или brand-protection сервисов.

6. Предлагаемый validation-план до кода

День 1: ручная валидация

- Собрать 20 карточек-кейсов:
- 5 safe;
- 5 спорных;
- 5 очевидно рискованных;
- 5 с нехваткой данных.
- Прогнать через LLM по схеме из вложения.
- Отдельно отметить false positive и false negative.

День 2: скелет правил

- Описать JSON-схему входа и выхода.
- Сделать таблицу scoring rules без кода.
- Проверить, объяснимы ли баллы для продавца.

День 3: пользовательская ценность

- Показать 5-10 продавцам/агентствам демо-отчеты.
- Проверить, понятны ли рекомендации.
- Спросить, за что они заплатили бы: разовая проверка, пакет, API, агентский кабинет.

День 4-5: только если есть сигнал

- Сделать CLI или минимальный FastAPI endpoint для локального теста.
- Без интеграций с реальными WB/Ozon аккаунтами.
- Без хранения секретов и пользовательских документов.

7. Архитектура после доказательства ценности

Минимальные компоненты:

- `listing_intake`: нормализация карточки товара;
- `risk_rules`: детерминированный scoring;
- `reference_data`: риск-категории, признаки товарных знаков, источники;
- `llm_explainer`: объяснение, uncertainty, рекомендации;
- `report_store`: история проверок;
- `review_queue`: ручная проверка спорных кейсов;
- `distribution`: Telegram/MAX alerts и weekly digest только после core-value.

Возможные интеграции позже:

- WB Content API для карточек и справочников;
- Ozon Seller API для карточек и описаний;
- ФИПС/Роспатент как справочный слой по товарным знакам;
- судебные и новостные источники только после правовой и ToS-проверки.

8. Открытые вопросы

- Для какой юрисдикции первая версия: РФ-only или СНГ/международные продавцы?
- Нужно ли проверять только до публикации или также мониторить уже опубликованные карточки?
- Кто покупатель: селлер, агентство, юрист, marketplace-интегратор?
- Какие данные пользователь готов вводить вручную?
- Есть ли доступ к реальным обезличенным кейсам претензий?
- Нужно ли включать risk по маркировке, сертификации, параллельному импорту и документам происхождения товара?
- Где граница между software intelligence и юридической услугой?

9. Источники для следующего этапа

- Wildberries, "Цифровой арбитраж": https://seller.wildberries.ru/instructions/uz/uz/material/digital-arbitration
- Wildberries API, работа с товарами: https://dev.wildberries.ru/en/docs/openapi/work-with-products
- Ozon, договор для продавцов / Intellectual Property Rights: https://docs.ozon.com/global/en/contracts-for-sellers/dogovor/
- Ozon, загрузка товаров через API: https://docs.ozon.com/global/api/via-api/
- Роспатент, самостоятельный поиск товарных знаков: https://rospatent.gov.ru/ru/faq/gde-samostoyatelno-provesti-poisk-tovarnyh-znakov-v-internete
- Пример публичного enforcement-кейса по Ozon seller / Makita: https://liapunov.com/en/news/300-millionov-za-kontrafakt-makita-corporation-vyigrala-rekordnyj-isk-protiv-prodavca-na-ozon/

10. Решение по текущему проекту

Sellersflare пока не добавляется в `src/clean_bot/subprojects.py`, потому что:

- домен не совпадает с коммерческой недвижимостью;
- нет подтверждения, что это часть Telegram/MAX clean-bot MVP;

- преждевременное добавление в API `GET /api/subprojects` смешает рабочий сервисный контур с отдельной startup-гипотезой.

Следующее корректное действие: провести ручную валидацию на 20 карточках и только после этого решать, делать ли отдельное приложение, отдельный бот или интеграцию в общий Nova-контур.