

Коммерческое предложение

White-Label решение для крипто-обменников (12 криптовалют ↔ Наличные)

1. О продукте

xChange Whitelabel — это готовое, полностью брендируемое white-label решение для организации обмена между криптовалютами и наличными рублями (RUB). Продукт представляет собой **Telegram Mini App + Telegram-бот** (фронтенд на React + TypeScript) и бэкенд-сервер на Go, которые поставляются в виде готовых Docker-образов и разворачиваются на инфраструктуре заказчика.

Поддерживаемые криптовалюты: USDT, BNB, BTC, DOGE, ETC, ETH, GRAM, LTC (Litecoin), SOL, TRX, USDC, XRP.

Решение ориентировано в первую очередь на **офлайн-криптообменники**, работающие через сеть физических офисов, а также на онлайн-сервисы, желающие быстро запустить собственный обменный сервис под своим брендом.

2. Ключевая функциональность

2.1. Обмен криптовалюты ↔ Наличные (RUB)

- **12 поддерживаемых криптовалют:** USDT, BNB, BTC, DOGE, ETC, ETH, GRAM, LTC (Litecoin), SOL, TRX, USDC, XRP
- **Два направления обмена для каждой валюты:**
 - Криптовалюта → Наличные RUB (продажа)
 - Наличные RUB → Криптовалюта (покупка)
- Автоматический расчёт суммы к получению по актуальному курсу для выбранной валюты
- **Фиксация курса выбранной крипто-валютной пары и итоговой суммы в рублях в момент создания заявки** — курс и сумма не изменятся после создания
- Поддержка точных денежных расчётов (целочисленная арифметика, исключение ошибок округления)

2.2. Telegram Mini App (Фронтенд)

- Полностью интегрирован в экосистему Telegram (Mini App)

- Авторизация через Telegram Init Data (HMAC-подпись, защита от replay-атак)
- Адаптивный интерфейс, оптимизированный для мобильных устройств
- Брендинг под партнёра (логотип, название, акцентный цвет, ссылка на поддержку)
- Онбординг с выбором офиса для новых пользователей
- **Создание заявки на обмен в выбранном пользователем офисе**
- **Виджет курсов всех валют** с автоматической каруселью на главном экране
- **Выбор криптовалюты** через удобные чипсы с горизонтальным скроллом
- **История заявок с просмотром их статусов**
- **Возможность отмены заявки пользователем**
- FAQ-раздел (настраиваемый, обсуждается индивидуально)
- Реферальная система (готова к интеграции, обсуждается отдельно)

2.3. Telegram Bot

- Команда `/start` с приветственным сообщением и кнопкой запуска Mini App
- **Команда `/rates`** — запрос курсов на покупку и на продажу для всех доступных криптовалют к рублю
- **Инлайн-запрос к боту** — запрос курса конкретной криптовалюты к рублю (покупка/продажа) в любом чате, без необходимости добавлять бота
- **Инлайн-калькулятор** — расчёты как с простыми числами, так и с валютами (например, `100usdt-2%`)
- **Система уведомлений в личные сообщения:** новая заявка, изменение статуса заявки
- Уведомления о создании заявки (сумма, курс, офис на карте)
- Уведомления об изменении статуса заявки (новая, в обработке, завершена, отменена)
- Отправка геолокации офиса в чат

2.4. Управление офисами (CRUD)

- Добавление, редактирование, удаление физических офисов
- Для каждого офиса: страна, город, адрес, координаты (GPS), часы работы
- **Индивидуальный мультипликатор курса для каждого офиса** (процентная наценка или скидка к базовому курсу, для маржи)
- **Координаты на карте** для отображения офиса на карте
- Привязка пользователя к конкретному офису

2.5. Управление заявками (транзакциями)

- Создание заявки пользователем через Mini App в выбранном офисе
- Жизненный цикл заявки: `new` → `pending` → `completed` / `cancelled`
- Обновление статуса менеджером/оператором (ролевая модель)

- **Возможность отмены заявки пользователем**
- Оптимистичная блокировка (optimistic locking) для предотвращения race condition
- Фильтрация и пагинация списка заявок (функционал фильтрации обсуждается индивидуально)

2.6. Курсы валют

- Автоматическое получение курсов всех 12 криптовалют из внешнего API
- **Кросс-курсы:** для валют, торгуемых к USDT, курс к RUB рассчитывается через USDT/RUB
- **Гибкая настройка спреда:** отдельные мультипликаторы для покупки и продажи (в процентах)
- Кэширование курсов в Redis (настраиваемый TTL)
- Применение мультипликатора офиса к базовому курсу (расширение/сужение спреда)

2.7. Система ролей и администрирование

- **Ролевая модель:** администратор, менеджер(-ы), пользователи
- **Администратор** — полное управление всеми данными: пользователи, заявки, офисы
- **Менеджер** — управление только заявками конкретного офиса, который установлен у него в профиле
- **Пользователь** — создание заявок, просмотр истории и статусов, отмена заявок

2.8. Безопасность

- Аутентификация через Telegram Init Data (HMAC SHA-256)
- Защита от replay-атак (проверка времени жизни init data — 24 часа)
- Rate limiting (скользящее окно на Redis, Lua-скрипт для атомарности)
- Ограничение размера тела запроса (1 MB, защита от DoS)
- CORS с валидацией Origin
- Параметризованные SQL-запросы (GORM)
- Реализация с учётом всех современных стандартов безопасности (чистая архитектура)

2.9. Инфраструктура

- **Поставка в виде Docker-образов** — заказчик получает готовые контейнеры (API, Frontend), готовые к развёртыванию
- **Docker Compose** для оркестрации сервисов (PostgreSQL 17, Redis 8, API, Frontend)
- **Makefile** для управления жизненным циклом (up, down, build, logs, shell)
- **Multi-stage Dockerfile** для Go API (легковесный scratch-образ)
- **Nginx** для раздачи статики фронтенда
- **Healthcheck** для всех сервисов

- **Runtime-конфигурация фронтенда** через `config.js` (без пересборки образа)

3. Технический стек

3.1. Клиентская часть (Frontend)

Компонент	Технология
Фреймворк	ReactJS (TypeScript) v19.2.8
Навигация	React-router
Работа с формами	React-hook-form + @hookform/resolvers
Валидация	Zod (валидация форм и данных)
Управление состоянием	Zustand (контроль и управление состоянием форм и приложения)
HTTP-клиент	@asouei/safe-fetch
Telegram SDK	@tma.js/sdk-react (взаимодействие с клиентом Telegram)
Качество кода	Oxlint, Oxfmt, Oxlint-tsgolint (форматирование и качество кода)
Стилизация	TailwindCSS, CSS-Modules
Сборка	Vite, Rolldown

3.2. Серверная часть (Backend)

Компонент	Технология
Язык	Go v1.26.5
Роутинг	Go-Chi
ORM	GORM
Telegram Bot API	gotgbot/v2
Событийная модель	Signals (pub/sub)
Кеш и Rate-Limiter	Redis (кеширование, Rate-Limiter со скользящим окном + Lua)

3.3. Инфраструктура

Компонент	Технология
База данных	PostgreSQL 17

Компонент	Технология
Кеш	Redis 8
Интеграция	Telegram Bot API + Mini Apps
Контейнеризация	Docker, Docker Compose
Реестр	GitVerse (поддержка любого Docker registry)

4. Преимущества для офлайн-криптообменников

4.1. Мгновенный запуск под своим брендом

- Не нужно разрабатывать приложение с нуля — разворачиваете готовое решение за 1 день
- Полный white-label: ваш логотип, название, цветовая схема, домен
- Поддержка нескольких партнёров из единой кодовой базы

4.2. Привязка к физическим офисам

- Клиент выбирает ближайший офис при онбординге
- Для каждого офиса можно установить индивидуальную наценку/скидку к курсу
- После создания заявки клиент получает геолокацию офиса в Telegram (при наличии координат – долготы, широты)
- Часы работы офиса отображаются в уведомлении (обсуждается и настраивается отдельно)

4.3. Прозрачность для клиента

- Фиксация курса в момент создания заявки — курс не изменится
- Уведомления на каждом этапе: создана → в обработке → завершена
- Полная история операций в приложении

4.4. Гибкое управление курсами

- Настройка спреда покупки/продажи через переменные окружения
- Индивидуальный мультипликатор для каждого офиса
- Автоматическое обновление курсов всех 12 валют из внешнего API
- Кеширование для снижения нагрузки на внешние источники

4.5. Безопасность и надёжность

- Аутентификация через Telegram (не нужно хранить пароли)
- Защита от повторных запросов (replay attack prevention)
- Rate limiting на уровне API

- Целочисленная арифметика для денежных расчётов (никаких ошибок float)
- Оптимистичная блокировка транзакций

4.6. Простота развёртывания

- Поставка в виде готовых Docker-образов — достаточно запустить контейнеры на вашем сервере
- Docker Compose — всё поднимается одной командой
- Разделение build-time и runtime конфигурации
- Healthcheck для мониторинга состояния сервисов

5. Почему стоит приобрести этот продукт

Приобретение готового white-label решения вместо разработки с нуля позволяет **сэкономить месяцы работы и десятки тысяч долларов**, а также **снять риски**, связанные с наймом, управлением и удержанием команды разработчиков.

5.1. Сравнение стоимости разработки с нуля

Ниже приведена оценка разработки аналогичного продукта (Telegram Mini App + Telegram-бот + бэкенд на Go + инфраструктура) силами наёмных специалистов. Оценка учитывает полный цикл: проектирование, разработку, тестирование, интеграцию с Telegram, развёртывание и документацию.

Сценарий разработки	Минимальный уровень специалистов	Кол-во специалистов	Ориентировочный срок	Ориентировочная стоимость
Один разработчик	Senior (full-stack)	1	4–6 месяцев	20 000 – 35 000 USDT
Несколько разработчиков	1× Senior (backend) + 1× Senior (frontend)	2	2.5–4 месяца	25 000 – 45 000 USDT
Команда (полный стек)	Senior backend + Senior frontend + QA + DevOps	3–4	2–3 месяца	35 000 – 60 000 USDT

Важно: приведённые суммы — это только **стоимость разработки**. В неё не входят расходы на управление командой, коммуникацию, доработки по ходу проекта, а также риски срыва сроков и ухода ключевых специалистов.

5.2. Почему нужны именно Senior и Middle+ специалисты

Продукт использует современный и требовательный стек, который **не под силу Junior-разработчикам**:

- **Бэкенд на Go** — чистая архитектура, событийная модель (pub/sub), конкурентность, оптимистичная блокировка, атомарные операции в Redis через Lua-скрипты. Требуется **Senior Go-разработчик**.
- **Фронтенд на React 19 + TypeScript** — интеграция с Telegram Mini Apps SDK, управление состоянием (Zustand), валидация (Zod), сборка на Vite/Rolldown. Требуется **Senior/Middle+ frontend-разработчик**.
- **Безопасность** — HMAC-подпись Telegram Init Data, защита от replay-атак, rate limiting, корректная работа с денежными суммами (целочисленная арифметика). Ошибки здесь критичны и стоят дорого.
- **Интеграция с Telegram Bot API** — инлайн-запросы, инлайн-калькулятор, уведомления, работа с геолокацией. Требуется глубокого понимания платформы.

Минимальный уровень специалистов для разработки с нуля — Middle+ и Senior. Найм Junior-разработчиков для такого проекта приведёт к затягиванию сроков, накоплению технического долга и росту стоимости поддержки.

5.3. Экономия при покупке готового решения

Показатель	Разработка с нуля (команда)	Покупка готового решения
Срок до запуска	2–6 месяцев	1–5 рабочих дней
Стоимость	20 000 – 60 000 USDT	от 2 000 USDT
Риск срыва сроков	Высокий	Отсутствует
Риск ухода ключевых специалистов	Высокий	Отсутствует
Необходимость найма команды	Да	Нет
Технический долг	Возможен	Минимальный
Поддержка и обновления	За свой счёт	Включены в лицензию

Итог: готовое решение позволяет запустить сервис **в десятки раз быстрее и в 10–30 раз дешевле**, чем разработка с нуля, при этом вы получаете проверенный, безопасный и полностью брендируемый продукт.

6. Варианты внедрения

Важно: любой вариант внедрения поддерживает **неограниченное количество офисов** с возможностью указать **индивидуальный мультипликатор курса** для каждого офиса. Ограничений по числу офисов нет ни в одном из тарифов.

Базовый

- Неограниченное количество офисов
- Индивидуальные мультипликаторы курса для каждого офиса
- Полный брендинг (логотип, название, цветовая схема, поддержка)

Расширенный

- Всё из Базового
- Настройка (или помощь в настройке) инфраструктуры
- Приоритетная поддержка

Enterprise

- Всё из Расширенного
- Помощь в интеграции с вашей CRM/ERP
- Кастомные доработки под бизнес-процессы
- Выделенный менеджер внедрения
- SLA 99.9%

7. Модель лицензирования

Параметр	Базовый	Расширенный	Enterprise
Стоимость лицензии	2 000 USDT	3 500 USDT	Индивидуально
Техническая поддержка	3 месяца	6 месяцев	12 месяцев
Обновления	3 месяца	6 месяцев	12 месяцев
Интеграция	—	—	✓
Кастомные доработки	—	—	✓

Важно: оплата принимается **строго в USDT** (сеть TRC-20).

Дополнительно:

- Техническое сопровождение после окончания поддержки — 350 USDT/мес

8. Исходный код и архитектура

В стандартную поставку (любой вариант внедрения) входят только **скомпилированные Docker-образы**. Исходный код в стоимость лицензии не включён.

Почему стоит рассмотреть передачу исходников?

Кодовая база написана с соблюдением всех современных стандартов разработки, что делает её максимально простой для понимания, поддержки и доработки силами ваших разработчиков:

- **Чистая архитектура (Clean Architecture)** — строгое разделение на слои: domain (бизнес-сущности), usecase (бизнес-логика), repository (доступ к данным), handler (транспорт). Каждый слой имеет чёткие интерфейсы и не зависит от нижележащих реализаций.
- **Типобезопасность** — Go на бэкенде и TypeScript на фронтенде исключают целые классы ошибок на этапе компиляции.
- **Целочисленная арифметика для денег** — все денежные расчёты используют `int64` с точностью до копейки/цента, никаких `float64` и связанных с ними ошибок округления.
- **Промышленный стандарт инструментов** — Go (chi router, GORM), React 19 + TypeScript + Vite, Docker multi-stage сборка, Redis с Lua-скриптами для атомарных операций.
- **Интерфейсы и внедрение зависимостей** — все компоненты связаны через интерфейсы, что позволяет легко подменять реализации (например, заменить rate provider или способ хранения).
- **Event-driven архитектура** — встроенная система событий (pub/sub) для уведомлений и интеграций.
- **Optimistic locking** — защита от потерянных обновлений при конкурентном доступе.
- **Полностью документированный код** — все экспортируемые типы, функции и пакеты содержат doc-комментарии на английском языке.

Любой опытный разработчик, знакомый со стеком (Go / React / TypeScript), с лёгкостью разберётся в коде, сможет его поддерживать, модернизировать и интегрировать с внешними системами без нашего участия.

Условия передачи исходного кода:

Передача исходного кода **обсуждается индивидуально** и оплачивается отдельно от стоимости лицензии. Стоимость зависит от объёма прав (ознакомление, модификация) и оговаривается в лицензионном договоре.

9. Инфраструктура заказчика

Расходы на инфраструктуру НЕ входят в стоимость франшизы и оплачиваются заказчиком самостоятельно.

Для работы решения потребуется:

Ресурс	Минимальная конфигурация	Примерная стоимость (мес.)
VPS / VDS	2 vCPU, 4 GB RAM, 40 GB SSD, Linux (Ubuntu 22.04+)	~10–20 USDT
Домен	1 домен для Mini App и api (например, <code>app.example.com</code> , <code>api.example.com</code>)	~5–10 USDT/год

Ресурс	Минимальная конфигурация	Примерная стоимость (мес.)
Telegram Bot	Бесплатно (регистрация через @BotFather)	0 USDT

Заказчик также может использовать собственную существующую инфраструктуру (серверы, базы данных) при условии её совместимости.

10. Процесс внедрения

1. **Демонстрация** — показ работающего решения, ответы на вопросы
2. **Заключение договора** — согласование условий и объёма работ
3. **Брендинг** — настройка логотипа, названия, цветовой схемы
4. **Передача Docker-образов** — заказчик получает готовые контейнеры API и Frontend
5. **Развёртывание** — установка на инфраструктуре заказчика (возможно с нашей удалённой помощью)
6. **Тестирование** — проверка всех сценариев обмена
7. **Запуск** — публикация бота и Mini App
8. **Передача документации** — инструкции по эксплуатации

Ориентировочный срок внедрения: от 1 до 5 рабочих дней.

11. Контакты

По всем вопросам обращайтесь:

- **Telegram:** [@frntbck](https://t.me/frntbck)
- **Email:** criticalbang@yandex.ru

Дата составления: август 2026 г.