

VERO AI - Техническое Задание

Общее описание

VERO AI — Telegram-бот для агрегации крипто-новостей с AI-анализом.

Концепция: Собираем новости из 9+ источников → AI выбирает 1 самую важную новость в час → постим в каналы.

Telegram каналы:

- EN: @vero_crypto_news
- RU: @vero_crypto_news_ru

Telegram бот: @VeroAssetBot

Архитектура

Tech Stack

- **Backend:** NestJS + TypeScript
- **Database:** PostgreSQL + Prisma ORM
- **Telegram:** Telegraf library
- **AI:** OpenAI-compatible API (любой провайдер)




Структура проекта

```
src/
├── ai/                # AI сервис - анализ новостей, определение приоритета
├── telegram/          # Telegram бот - обработка команд, отправка сообщений
├── rss/               # RSS парсер - сбор новостей из источников
├── cron/              # Планировщик - автоматическое сканирование и постинг
├── database/          # Prisma сервис
├── common/            # Переводы интерфейса
└── main.ts           # Точка входа
```

RSS Источники (9 штук)

1. Cointelegraph (EN)
2. CoinDesk (EN)
3. Decrypt (EN)
4. BeInCrypto (EN)
5. The Block (EN)
6. Bitcoin Magazine (EN)
7. CryptoSlate (EN)
8. NewsBTC (EN)

Система приоритетов


Приоритет	Эмодзи	Критерии
RED		BTC/ETH $\pm 10\%$, хаки $> \$50M$, SEC/законы, банкротства бирж
YELLOW		Альты $\pm 15\%$, партнёрства, листинги, киты $> \$50M$
GREEN		Всё остальное

Расписание


- **Сканирование RSS:** каждые 15 минут
- **Постинг в каналы:** каждый час в :00
- **Логика:** из буфера выбирается новость с наивысшим приоритетом (RED > YELLOW > GREEN)

Формат постов (ТРЕБУЕТСЯ ДОРАБОТКА)


Текущий формат (нужно исправить):

 **Заголовок 5-7 слов**








4-6 предложений с контекстом и фактами.

 **VERO AI:**

****  Простым языком: **** 1-2 предложения.

****  Может привести к: **** 1-2 предложения о последствиях.

Требования к формату:

1.  Термины объяснять В СКОБКАХ внутри текста, например: “ETF (фонд на бирже)”
2.  НЕ “К чему приведет”, а “Может привести к”
3.  Эмодзи  и  в секции VERO AI
4.  Убрать отдельный блок “Термины:”
5.  Каждый пост ОБЯЗАТЕЛЬНО с картинкой



Логика картинок

1. Берём картинку из RSS item
2. Если нет — ищем ту же новость в других RSS источниках
3. Если нет — используем fallback (крипто-тематика)

Проблема: Иногда картинки не отправляются. Нужно:

- Проверять валидность URL перед отправкой
- Надёжные fallback URLs
- Логировать ошибки



Мультиязычность

Поддерживаемые языки: EN, RU

Важно:

- В EN канал — ТОЛЬКО на английском
- В RU канал — ТОЛЬКО на русском
- AI должен генерировать отдельный анализ для каждого языка



API Endpoints

Endpoint	Метод	Описание
/	GET	Health check
/admin/scan-now	POST	Ручной запуск сканирования
/admin/broadcast-now	POST	Ручной постинг в каналы
/webhook/telegram	POST	Webhook для Telegram
/api-docs	GET	Swagger документация



Переменные окружения

```
TELEGRAM_BOT_TOKEN=токен_от_botfather
DATABASE_URL=postgresql://user:pass@host:5432/db
ABACUSAI_API_KEY=ключ_для_llm_api
APP_ORIGIN=https://your-domain.com/
```

✓ Что работает

1. ✓ RSS парсинг 9 источников
 2. ✓ Фильтр свежести (только новости < 1 часа)
 3. ✓ Определение приоритета через AI
 4. ✓ Генерация анализа на EN и RU
 5. ✓ Постинг в каналы с картинками
 6. ✓ Автоматический cron (15 мин сканирование, 1 час постинг)
 7. ✓ Telegram бот с командами /start, /language, /filter
 8. ✓ База данных (пользователи, отправленные новости)
-

✗ Что нужно доделать/исправить

1. Картинки (КРИТИЧНО)

- Картинки иногда не отправляются
- Нужна надёжная логика с fallback
- Тестировать URL перед отправкой

2. Формат постов

- Термины в скобках, не отдельным блоком
- “Может привести к” вместо “К чему приведет”
- Проверить что AI не путает языки

3. AI промпт

- Упростить промпт
 - Убрать лишнюю валидацию (тратит кредиты)
 - Авто-замена markdown на HTML
-

🚀 Деплой на Render.com

1. Создать репозиторий на GitHub
2. Загрузить код
3. Render.com → New Web Service → Connect GitHub
4. `render.yaml` автоматически настроит всё
5. Добавить environment variables
6. После деплоя установить webhook:

```
curl "https://api.telegram.org/bot<TOKEN>/setWebhook?url=https://your-app.onrender.com/webhook/telegram"
```

Контакты каналов

- EN канал: @vero_crypto_news (добавить бота админом)
- RU канал: @vero_crypto_news_ru (добавить бота админом)

Бот должен иметь права администратора для постинга в каналы.

AI API

Используется OpenAI-совместимый API. Можно использовать:

- OpenAI напрямую
- Anthropic через прокси
- Любой другой провайдер с OpenAI-совместимым endpoint

Endpoint в коде: `https://routellm.abacus.ai/v1/chat/completions`

Можно заменить на любой другой в файле `src/ai/ai.service.ts`