

# Архитектурный План для AI-Управляемой Платформы Анализа ICO

## Часть 1: Механизм Сбора Данных — Архитектура для Масштабируемого Сcrapинга ICO

Эта часть отчета закладывает архитектурный фундамент для всей системы. Надежный и масштабируемый конвейер сбора данных имеет первостепенное значение; если данные не могут быть получены надежно, продвинутые аналитические агенты будут лишены информации, необходимой для их функционирования. Основным принципом здесь является **декаплинг (разделение)**: отделение процесса получения данных (скрапинг) от их потребления (анализ) для обеспечения отказоустойчивости, масштабируемости и удобства сопровождения.

### Раздел 1.1: Разделенная, Асинхронная Архитектура с Celery и RabbitMQ

В этом разделе будет подробно описана рекомендуемая архитектура для системы, предназначеннной для одновременного мониторинга более 20 веб-сайтов. Ключевым моментом является переход от монолитной, синхронной модели к асинхронной, распределенной модели с очередью задач.

- **Основная Рекомендация:** Будет внедрена система распределенной очереди задач с использованием **Celery** в качестве оркестратора задач и **RabbitMQ** в качестве брокера сообщений.<sup>1</sup> Эта архитектура является отраслевым стандартом для масштабируемых, асинхронных рабочих нагрузок.
  - Веб-приложение или главный сервис-планировщик ("производитель") не будет выполнять скрапинг самостоятельно. Вместо этого он будет

- создавать "задачу" (например, "выполнить скрапинг [icodrops.com/upcoming](http://icodrops.com/upcoming)") и отправлять эту задачу в очередь RabbitMQ.<sup>1</sup>
- Пул независимых "воркеров Celery" ("потребителей") будет прослушивать очередь, забирать задачи и выполнять фактическую логику скрапинга.<sup>2</sup> Это освобождает основное приложение от блокировок и позволяет ему обрабатывать множество одновременных запросов или событий по расписанию.<sup>1</sup>
  - **Стратегия Масштабирования для Задач, Ограниченных Вводом-Выводом (I/O-Bound):** Веб-скрапинг — это классическая задача, ограниченная вводом-выводом (процесс большую часть времени ожидает ответа от сети), а не вычислениями (CPU-bound). Это имеет критические последствия для того, как мы масштабируем наших воркеров.
    - Воркеры Celery будут настроены на использование пула выполнения на основе потоков (например, gevent) вместо стандартного пула на основе процессов (prefork).<sup>4</sup> Это позволяет одному процессу-воркеру (работающему на одном ядре CPU) обрабатывать сотни или даже тысячи одновременных задач скрапинга, переключаясь между ними во время ожидания сетевого ввода-вывода. Это значительно более эффективно с точки зрения использования ресурсов, чем создание отдельного процесса для каждой задачи.<sup>4</sup>
    - Для горизонтального масштабирования будет запущено несколько экземпляров воркеров Celery, возможно, на нескольких серверах или в Docker-контейнерах.<sup>4</sup> Можно использовать встроенное автомасштабирование Celery (`--autoscale`) или, для более продвинутой настройки, Kubernetes с KEDA (Kubernetes-based Event-Driven Autoscaling) для автоматического масштабирования количества подов с воркерами в зависимости от длины очереди RabbitMQ.<sup>1</sup> Это гарантирует, что у нас будет ровно столько ресурсов, сколько необходимо для обработки нагрузки, и позволит сокращать их для экономии средств, когда очередь пуста.
  - **Архитектурная Диаграмма:** Будет предоставлена высокоуровневая диаграмма, иллюстрирующая поток: Планировщик → RabbitMQ (Очередь Задач) → Воркеры Celery (Скрaperы) → Хранилище Сырых Данных.

Выбор архитектуры Celery/RabbitMQ — это не просто техническая деталь реализации; это фундаментальное решение, которое обеспечивает "разделение ответственности" (separation of concerns), что напрямую соответствует ключевым принципам построения надежных, нерупких AI-систем, обсуждавшимся в предоставленных вами документах.<sup>6</sup> Монолитный скрапер, который

одновременно извлекает и обрабатывает данные, нарушает этот принцип. Он становится сложным, трудным для отладки и единой точкой отказа. Модель Celery/RabbitMQ принудительно внедряет это разделение.<sup>1</sup> "Планировщик" знает только, как создавать задачи. "Воркер" знает только, как выполнить конкретную задачу скрапинга для заданного URL. "Аналитический агент" (обсуждается позже) знает только, как читать данные из чистой базы данных. Такое архитектурное разделение отражает логическое разделение агентов в фреймворке LangGraph. Это делает систему более отказоустойчивой (сбой скрапера не выводит из строя аналитический конвейер), более масштабируемой (можно добавлять больше воркеров-скраперов, не затрагивая аналитических агентов) и более простой в обслуживании (сломанный парсер для одного сайта — это изолированная проблема в одной функции воркера).

## Раздел 1.2: Озеро и Хранилище Данных — Гибридная Стратегия Хранения

Этот раздел рассматривает критически важный вопрос о том, где и как хранить данные на разных этапах конвейера. Одной базы данных недостаточно для удовлетворения различных потребностей в сырых, неструктурированных данных и чистых, структурированных данных.

- **Хранилище Сырых Данных ("Озеро Данных"):** MongoDB. Для первоначальной выгрузки "сырых" данных со скраперов MongoDB является предпочтительным выбором.<sup>8</sup>
  - **Обоснование:** Данные, полученные в результате скрапинга, по своей природе неструктурированы, и их схема часто меняется по мере обновления верстки веб-сайтов. Гибкая, документо-ориентированная модель MongoDB (BSON, бинарный JSON) идеально подходит для этой цели.<sup>8</sup> Сcrapер может просто выгрузить весь сырой HTML или JSON-объект с извлеченным текстом, не беспокоясь о жесткой, предопределенной схеме.<sup>10</sup> Это делает сами скраперы проще и более устойчивыми к незначительным изменениям на сайтах.
  - Это "озеро данных" служит постоянной, аудируемой записью того, что и когда было собрано, что неоценимо для отладки и повторной обработки данных в случае изменения логики парсинга.
- **Хранилище Структурированных Данных ("Хранилище Данных"):** PostgreSQL с JSONB. После того как сырые данные будут разобраны и очищены на отдельном этапе обработки, полученные структурированные

данные должны храниться в базе данных **PostgreSQL**.<sup>11</sup>

- **Обоснование:** PostgreSQL предлагает лучшее из двух миров: надежность и транзакционную целостность реляционной базы данных (соответствие ACID) и превосходную поддержку полустроктурированных данных через тип данных JSONB.<sup>11</sup> Тип JSONB хранит JSON в оптимизированном бинарном формате, что позволяет эффективно индексировать и запрашивать вложенные поля.<sup>11</sup>
  - Это "хранилище данных" будет содержать чистые, валидированные данные, которые будут потреблять наши аналитические агенты. Использование PostgreSQL позволяет нам обеспечивать целостность данных с помощью четко определенных схем (используя модели Pydantic для генерации структур таблиц) для ключевых полей (например, project\_name, sale\_date), в то же время используя JSONB для более сложных, вложенных данных, таких как графики вестинга токенов или информация о членах команды.
- **Таблица 1: Сравнение MongoDB и PostgreSQL JSONB для хранения данных скрапинга**

Характеристика	MongoDB	PostgreSQL (JSONB)	Рекомендация в нашем конвейере
<b>Гибкость схемы</b>	Очень высокая. Схема не навязывается, идеально для изменяющихся данных. <sup>8</sup>	Умеренная. Основная таблица имеет схему, но поле JSONB гибкое. <sup>11</sup>	<b>MongoDB</b> для "озера данных" (сырые данные).
<b>Производительность записи</b>	Высокая для неструктурированных данных, так как нет проверки схемы. <sup>12</sup>	Высокая, но JSONB выполняет валидацию и бинарную сериализацию. <sup>11</sup>	<b>MongoDB</b> для быстрой выгрузки сырых данных скраперами.
<b>Гибкость запросов</b>	Мощный язык запросов MQL, оптимизированный для документов. <sup>8</sup>	Стандартный SQL с мощными операторами и функциями для JSONB. <sup>10</sup>	<b>PostgreSQL</b> для сложного анализа и объединения структурированных данных.

<b>Индексация</b>	Индексы по любым полям, включая вложенные в документы.	GIN-индексы для JSONB обеспечивают высокую производительность запросов к вложенным данным. <sup>10</sup>	Оба варианта мощные, но PostgreSQL лучше для реляционных связей.
<b>Целостность данных (ACID)</b>	Поддерживает ACID-транзакции на уровне нескольких документов. <sup>8</sup>	Полная ACID-совместимость на уровне всей базы данных, что является золотым стандартом. <sup>12</sup>	<b>PostgreSQL</b> для "хранилища данных" (чистые данные), где целостность критична.
<b>Основной сценарий использования</b>	<b>Озеро данных:</b> Хранение сырого HTML и первоначально извлеченных, неструктурированных данных.	<b>Хранилище данных:</b> Хранение очищенных, структурированных и валидированных данных для потребления AI-агентами.	Гибридный подход: использовать каждый инструмент для своей сильной стороны.

### Раздел 1.3: Обеспечение Бесперебойной Работы и Надежности — Автоматизированный Мониторинг Состояния Парсеров

Скрaper полезен только тогда, когда он работает. В этом разделе будет разработана система для автоматического обнаружения "поломки" парсера для конкретного сайта из-за изменений в верстке и оповещения команды разработчиков.

- **Многоуровневая Стратегия Мониторинга:** Полагаться на одну проверку — хрупкое решение. Будет реализован трехуровневый подход.
  1. **Мониторинг на Уровне Задач (ScrapeOps):** Будет интегрирован сервис, такой как **ScrapeOps**.<sup>13</sup> SDK ScrapeOps можно добавить в наши Python-скраперы всего несколькими строками кода. Он автоматически отслеживает процент успешных задач, HTTP-коды состояния, время выполнения и ошибки. Его можно настроить на отправку оповещений (через Slack, email), когда метрики задачи значительно отклоняются от

своего исторического скользящего среднего (например, если процент успеха для [icodrops.com](https://icodrops.com) падает на 50%).<sup>13</sup>

2. **Проверка Качества Данных ("Канареечные" Проверки):** Для каждого целевого сайта будет определена "канареечная" точка данных — критически важная информация, которая всегда должна присутствовать. Для сайта ICO это может быть "Дата Продажи Токенов" или сумма "Всего Собрано". После скрапинга этап валидации проверит, было ли это поле успешно извлечено и находится ли оно в правильном формате. Если "канареечная" проверка не проходит для нескольких последовательных запусков, генерируется оповещение. Это позволяет обнаруживать "тихие сбои", когда скрапер выполняется без ошибок, но не извлекает полезных данных.
3. **Обнаружение Аномалий на Основе Контента:** Будет храниться хэш или структурная подпись собранного HTML. Если подпись страницы кардинально меняется между запусками (что указывает на серьезный редизайн верстки), система пометит ее для ручной проверки. Это обеспечивает раннее предупреждение еще до того, как сам парсер сломается. Сервисы, такие как **Browse AI**, упоминают адаптацию к изменениям сайта, что указывает на важность этой концепции.<sup>14</sup>

Проблема "сломанных парсеров" — это не просто технический сбой, это кризис целостности данных. "Тихий сбой" может отравить последующий набор данных нулевыми или неверными значениями, что приведет к тому, что AI-агенты примут ошибочные решения. Следовательно, система мониторинга должна быть сосредоточена на *качестве данных*, а не только на *времени безотказной работы скрапера*. Аналитические агенты (*LangGraph*) доверяют данным, которые они получают из хранилища PostgreSQL. Скрапер может "успешно" выполниться (вернуть код состояния 200), но не извлечь никаких данных из-за изменения CSS-класса. Если эти нулевые данные будут записаны в базу данных, *Analysis Agent* может интерпретировать это как "У проекта нет цели по сбору средств", что приведет к неверной отрицательной оценке. Поэтому мониторинг должен происходить на уровне валидации данных, а не только на уровне сетевых запросов. Подход с "канареечной" проверкой напрямую решает эту проблему, связывая определение "успеха" с наличием критически важных, валидированных точек данных, тем самым защищая целостность всего аналитического конвейера.

## Анализа

Этот раздел определяет "рукопожатие" между новым модулем скрапинга и существующим оркестратором LangGraph, обеспечивая чистую, разделенную архитектуру.

- **Золотое Правило: Никакой Прямой Связи.** Discovery Agent (или любой другой агент LangGraph) **не должен** напрямую вызывать скраперы или читать данные из "озера данных" (MongoDB).<sup>16</sup> Это создает жесткую связь, которая является хрупкой и трудной в обслуживании.
- **Событийно-Ориентированная, Базо-Центральная Интеграция:** Будет использоваться разделенная, событийно-ориентированная архитектура.<sup>18</sup>
  1. **Модуль Скрапинга:** Воркеры Celery собирают данные, выполняют первоначальную очистку/парсинг и записывают чистые, структурированные данные по ICO в хранилище данных **PostgreSQL**. Это их единственная обязанность.
  2. **Триггер:** При успешном добавлении новой записи (или обновлении) в таблицу ico\_projects в PostgreSQL, триггер базы данных или отдельный сервис уведомлений (например, прослушивающий поток WAL PostgreSQL) опубликует сообщение о событии (например, {"event": "new\_ico\_found", "project\_id": 123}) в выделенную очередь **RabbitMQ** для аналитического конвейера.
  3. **Аналитический Модуль: Оркестратор LangGraph** будет иметь специального слушателя ("агент-триггер"), который потребляет сообщения из этой очереди. Когда он получает событие new\_ico\_found, он использует project\_id для извлечения полных, структурированных данных из PostgreSQL и инициирует рабочий процесс анализа, передавая данные первому соответствующему агенту (например, Analysis Agent).
- **Преимущества такого Разделения:**
  - **Отказоустойчивость:** Аналитическая система может продолжать функционировать, даже если система скрапинга не работает, и наоборот.<sup>19</sup>
  - **Независимая Масштабируемость:** Модули скрапинга и анализа могут масштабироваться независимо в зависимости от их конкретных нагрузок.<sup>20</sup>
  - **Целостность Данных:** Агентам LangGraph гарантируется, что они всегда будут работать только с чистыми, структуризованными и валидированными данными из хранилища PostgreSQL, а не с сырьем,

потенциально грязным контентом скрапинга.<sup>17</sup> Это напрямую реализует философию "структурированного вывода" из предоставленных вами документов.<sup>6</sup>

## Часть 2: Инструментарий Скрапера — Продвинутые Техники Извлечения Данных

Эта часть переходит от высокоуровневой архитектуры к практическим тактикам, необходимым для извлечения данных из современного, враждебного веба.

### Раздел 2.1: Укрощение Современного Веба — Выбор Headless-Браузера

Многие лаунчпады ICO являются современными одностраничными приложениями (SPA), которые загружают данные динамически с помощью JavaScript. Простой скрапер на основе requests потерпит неудачу. Нам нужен headless-браузер.

- **Рекомендация:** **Playwright** является однозначно лучшим выбором для этого проекта по сравнению с Selenium и Puppeteer.<sup>21</sup>
  - **Производительность:** Playwright значительно быстрее Selenium и часто сопоставим или быстрее Puppeteer, особенно в сложных сценариях, благодаря своей современной архитектуре, которая напрямую взаимодействует с браузером.<sup>21</sup>
  - **Функциональность:** Он обладает лучшими в своем классе функциями, такими как автоматические ожидания (автоматическое ожидание готовности элементов перед взаимодействием с ними), перехват сетевых запросов и поддержка нескольких браузеров (Chromium, Firefox, WebKit) "из коробки".<sup>22</sup> Это крайне важно для проверки, ведет ли себя сайт по-разному в разных браузерах.
  - **Поддержка Языков:** В отличие от Puppeteer, который официально поддерживает только JavaScript, Playwright имеет первоклассные, официально поддерживаемые API для Python, Java и C#.<sup>21</sup> Это огромное преимущество для проекта на Python.
  - **Опыт Разработки:** Инструменты отладки Playwright, такие как Playwright Inspector и Trace Viewer, значительно превосходят инструменты

конкурентов, что кардинально ускоряет разработку и устранение неполадок.<sup>22</sup>

## Раздел 2.2: Плащ-Невидимка — Обход IP-Блокировок и Фильтрации по Отпечаткам

Веб-сайты используют два основных метода для обнаружения скраперов: ограничение скорости на основе IP и анализ отпечатков браузера (fingerprinting). Мы должны обойти оба.

- **Стратегия Прокси: Многоуровневые Ротируемые Резидентные Прокси.**
  - **Почему прокси необходимы:** Для скрапинга 20+ сайтов в больших масштабах использование одного IP-адреса приведет к немедленным блокировкам.<sup>24</sup> Прокси-серверы являются обязательным требованием.<sup>25</sup>
  - **Резидентные vs. Дата-центровые:** Для высокозащищенных целей, таких как лаунчпады ICO, **резидентные прокси** являются единственным жизнеспособным вариантом. Это IP-адреса от реальных интернет-провайдеров (ISP), что заставляет их выглядеть как легитимные домашние пользователи и делает их гораздо сложнее для обнаружения.<sup>26</sup> Дата-центровые прокси, хотя и дешевле и быстрее, происходят из известных IP-блоков и легко помечаются сложными системами защиты.<sup>28</sup>
  - **Ротация:** Будет использоваться **сервис ротируемых прокси**, который предоставляет новый IP-адрес для каждого запроса или каждого нескольких запросов. Такое распределение запросов по тысячам IP-адресов делает невозможным создание профиля блокировки на основе объема запросов из одного источника.<sup>29</sup>
  - **Интеллектуальная Ротация:** Простая случайная ротация неоптимальна. Будет реализована стратегия взвешенной случайной ротации, которая штрафует неработающие или недавно использованные прокси/подсети и поощряет здоровые, быстрые, максимизируя процент успеха.<sup>31</sup>
- **Борьба с Фильтрацией по Отпечаткам: Stealth-Плагины.**
  - Headless-браузеры оставляют характерные следы в своей среде JavaScript (например, флаг navigator.webdriver). Системы защиты от ботов, такие как Cloudflare, обнаруживают эти сигнатуры.<sup>32</sup>
  - Будет использоваться stealth-плагин для Playwright (например, playwright-extra-plugin-stealth или аналогичные библиотеки).<sup>32</sup> Эти плагины автоматически "патчат" среду браузера во время выполнения,

чтобы удалить распространенные признаки автоматизации, делая скрапер гораздо более похожим на реального пользователя.<sup>33</sup>

## Раздел 2.3: Победа над Стражами — Программное Решение CAPTCHA

Самые продвинутые системы защиты от ботов, такие как Cloudflare Turnstile и hCaptcha, все равно будут представлять собой вызовы. Их необходимо решать программно.

- **Решение: Сторонние Сервисы Решения.** Создание собственного решателя CAPTCHA нецелесообразно. Будет выполнена интеграция со специализированным, AI-управляемым сервисом решения CAPTCHA.
  - **Ведущие Сервисы:** Capsolver является одним из лучших вариантов, известным своей скоростью, высоким процентом успеха и поддержкой широкого спектра CAPTCHA, включая Cloudflare Turnstile и hCaptcha, с прямыми API-интеграциями.<sup>35</sup> Другими сильными конкурентами являются 2Captcha и Anti-Captcha.<sup>38</sup>
  - **Процесс Интеграции:**
    1. Скрипт Playwright обнаруживает наличие CAPTCHA на странице.
    2. Он захватывает необходимую информацию (например, site-key, URL страницы или скриншот задачи).
    3. Он отправляет эту информацию в API сервиса-решателя (например, API Capsolver).
    4. Сервис решает CAPTCHA (используя комбинацию AI и людей-решателей) и возвращает токен-решение.<sup>34</sup>
    5. Скрипт Playwright вставляет этот токен на страницу и отправляет форму, обходя защиту.
- **Таблица 2: Сравнение сервисов для решения CAPTCHA**

Сервис	Ключевая технология	Поддерживающие CAPTCHA	Среднее время решения	Цена за 1000 решений (пример)	Интеграция с Python
Capsolver	AI/ML <sup>36</sup>	reCAPTCHA v2/v3,	1-9 секунд <sup>36</sup>	hCaptcha: ~\$0.8,	Да, через API <sup>35</sup>

		hCaptcha, FunCaptcha, Cloudflare Turnstile <sup>35</sup>		Turnstile: ~\$1.2 <sup>36</sup>	
<b>2Captcha</b>	Люди-реша тели <sup>38</sup>	reCAPTCHA v2/v3, hCaptcha, turnstile, и др. <sup>36</sup>	7-20 секунд <sup>38</sup>	reCAPTCHA: ~\$2.89 <sup>36</sup>	Да, через API и библиотеки <sup>40</sup>
<b>Anti-Captch a</b>	Люди-реша тели <sup>36</sup>	reCAPTCHA, hCaptcha, и др.	13-20 секунд <sup>38</sup>	reCAPTCHA: ~\$2.00	Да, через API
<b>DeathByCa ptcha</b>	Гибрид (OCR + Люди) <sup>36</sup>	reCAPTCHA, hCaptcha, Audio	~9-30 секунд <sup>38</sup>	Normal: ~\$0.99, reCAPTCHA: ~\$2.89 <sup>36</sup>	Да, через API

## Раздел 2.4: AI-Управляемое Извлечение — Уход от Хрупких Селекторов

Традиционные скраперы полагаются на CSS-селекторы или XPath, которые являются хрупкими и ломаются каждый раз, когда меняется фронтенд-код сайта. Можно создать более устойчивую систему с использованием больших языковых моделей (LLM).

- **Концепция:** Вместо написания `page.locator(':token-price-value').inner_text()`, можно передать сырой HTML релевантного раздела страницы в LLM (например, Gemini) с конкретным промптом.
- **Реализация с LangExtract:** Библиотека **LangExtract** от Google разработана специально для этой цели.<sup>41</sup>
  - **Процесс:**
    1. Скрaper Playwright изолирует основной div с контентом на странице проекта ICO.
    2. Этот фрагмент HTML передается в функцию, использующую LangExtract или аналогичную стратегию промптинга.
    3. Промпт инструктирует LLM действовать как экстрактор данных и вернуть JSON-объект с определенными полями: `{"token_price": float, "sale_start_date": "YYYY-MM-DD", "total_supply": int,...}`. Можно

использовать наши модели Pydantic для генерации желаемой JSON-схемы для промпта.

- **Преимущества:** Этот подход устойчив к изменениям в CSS-классах, ID и структуре HTML. Пока человекочитаемый текст присутствует на странице, LLM, скорее всего, сможет его найти. Это кардинально снижает нагрузку на обслуживание скраперов.

Использование LLM для извлечения — это не просто повышение отказоустойчивости; это абстрагирование *намерения* извлечения данных. CSS-селектор, такой как `div.info-panel > span.price`, является *процедурной* инструкцией. Он говорит скраперу, как найти данные. Промпт для LLM, такой как "Найди цену токена из этого HTML", является *декларативной* инструкцией. Он говорит модели, какие данные найти. Декларативные инструкции по своей природе более устойчивы к изменениям в базовой реализации (HTML сайта). Это фундаментальный принцип хорошего проектирования программного обеспечения. Применяя этот подход, мы не просто чиним сломанные скраперы; мы фундаментально меняем нашу философию извлечения данных, чтобы она больше соответствовала принципам современной разработки AI, делая всю систему более интеллектуальной и адаптируемой. Это также соответствует вашему предпочтению к структурированным выводам, определенным в Pydantic<sup>6</sup>, поскольку LLM можно попросить вернуть данные, которые напрямую соответствуют этим моделям.

## Часть 3: Мозг Аналитика — Фреймворк для Оценки ICO

В этой части подробно описывается "мозг" операции. Как только у нас есть данные, как мы их анализируем для получения действенной информации? Будет построен фреймворк, вдохновленный профессиональными венчурными фондами и криpto-нативными исследовательскими фирмами, такими как Delphi Digital и Paradigm.

### Раздел 3.1: Анатомия ICO — Ключевые Метрики и Токеномика

Этот раздел определяет обязательные количественные и качественные точки данных, которые наша система должна собирать и анализировать для каждого ICO.

- **Количественные "Стандартные" Метрики:** Это абсолютный минимум для любого анализа токеномики.<sup>42</sup>
  - **Метрики Оценки:**
    - **Начальная Рыночная Капитализация (Initial Market Cap, IMC):** (Начальное циркулирующее предложение при TGE) \* (Цена ICO). Эта метрика представляет оценку проекта в первый день.
    - **Полностью Разводненная Оценка (Fully Diluted Valuation, FDV):** (Общее предложение токенов) \* (Цена ICO). Эта метрика представляет долгосрочную, теоретическую оценку проекта, если бы все токены были в обращении.<sup>45</sup>
    - **Соотношение IMC/FDV:** Это, возможно, самая важная отдельная метрика. Низкое соотношение (например, < 20%) является **серьезным красным флагом**, указывающим на massive будущую инфляцию токенов и давление продавцов.<sup>47</sup>
  - **Метрики Предложения и Распределения:**
    - **Общее и Максимальное Предложение:** Предложение фиксированное или инфляционное?<sup>42</sup>
    - **Распределение Токенов (%):** Разбивка аллокаций для Команды, Советников, Инвесторов (Seed, Private), Публичной Продажи, Казначейства/Экосистемы.<sup>48</sup> Большая аллокация для команды/инсайдеров является красным флагом.<sup>50</sup>
    - **Графики Вестинга и Клиффа:** Для токенов команды и инвесторов, какой период клиффа (например, 12 месяцев) и общая продолжительность вестинга (например, 36 месяцев)? Более короткие графики являются красным флагом.<sup>52</sup> Мы будем искать графики, которые соответствуют долгосрочным обязательствам, — принцип, часто подчеркиваемый ведущими венчурными фондами, такими как a16z.<sup>53</sup>
  - **Качественные Метрики (будут оцифрованы в Разделе 3.2):**
    - **Проект:** Качество whitepaper, ясность видения, реальный сценарий использования и технологические инновации.<sup>50</sup> Действительно ли проекту нужен токен?<sup>51</sup>
    - **Команда:** Опыт, служебной список, прозрачность (анонимны ли они?), и качество советников.<sup>44</sup>

- **Инвесторы (Бэкеры):** Качество вовлеченных венчурных фондов является мощным сигналом. Поддержка от фондов высшего уровня, таких как **Paradigm** или **a16z**, является огромным зеленым флагом.<sup>56</sup>
- **Сообщество и Хайп:** Вовлеченность в социальных сетях (Twitter, Telegram), количество подписчиков и сентимент.<sup>49</sup>

## Раздел 3.2: От Качественного к Количественному — Взвешенная Модель Оценки

Этот раздел представляет формальную методологию для преобразования качественных оценок из раздела 3.1 в числовой балл, что позволяет объективно сравнивать проекты. Именно так профессиональные аналитики избегают чисто субъективных решений.

- **Матрица Оценки:** Будет создана комплексная матрица оценки, где каждой метрике присваивается вес в зависимости от ее важности. Конечным результатом является единый "ICO Score" от 1 до 100.
- **Оцифровка Качественных Показателей:**
  - **Сила Команды (Балл 1-10):** Баллы начисляются на основе предыдущих экзитов основателей, опыта работы в FAANG или крупных криpto-проектах, а также наличия PhD в релевантных областях. Публичная, неанонимная команда — это минимальное требование.<sup>50</sup>
  - **Качество Инвесторов (Балл 1-10):** Создаются уровни для венчурных фондов. Уровень 1 (Paradigm, a16z и т.д.) = 10 баллов. Уровень 2 (другие известные криpto-фонды) = 7 баллов. Неназванные/ангельские инвесторы = 3 балла. Отсутствие заметных бэкеров = 1 балл. Это является прокси для "доступа к конкурентным сделкам".<sup>60</sup>
  - **Технология/Иновации (Балл 1-10):** Агент на базе LLM будет анализировать whitepaper на предмет технической глубины, сравнивая его с базой данных существующих решений. Балл присваивается на основе новизны и осуществимости. Проект без демо, без беты и без кода на GitHub получает очень низкий балл.<sup>51</sup>
  - **Токеномика (Балл 1-10):** Формульный балл, основанный на количественных метриках из раздела 3.1. Высокое соотношение IMC/FDV, длительные графики вестинга и низкая аллокация для инсайдеров приведут к высокому баллу.
- **Красные и Зеленые Флаги (Бинарные Модификаторы):**

- **Красные Флаги:** Это критические проблемы, которые могут автоматически дисквалифицировать проект или сильно оштрафовать его балл.<sup>50</sup>
  - Анонимная команда.
  - Нереалистичные обещания гарантированной прибыли.
  - Отсутствие четкого сценария использования или whitepaper.
  - Соотношение IMC/FDV ниже 10%.
  - Вестинг для команды/инвесторов менее 12 месяцев.
- **Зеленые Флаги:** Это сильные положительные сигналы, которые могут дать бонус к баллу.
  - Поддержка от венчурного фонда Уровня-1 (Paradigm, a16z).
  - У команды есть предыдущие успешные экзиты.
  - У проекта есть работающий продукт/бета до ICO.
  - Смарт-контракты прошли аудит у авторитетной фирмы.
- Таблица 3: Матрица оценки ICO

Категория	Метрика	Источник(и) данных	Метод оцифровки	Вес (%)
<b>Токеномика</b>	Соотношение IMC/FDV	Сcrapинг, API	Прямой расчет. Оценка по шкале ( >50% = 10, <10% = 1)	25%
	Вестинг команды/инвесторов	Сrapинг (Whitepaper)	Оценка по шкале (Клифф >12 мес, Вестинг >36 мес = 10)	15%
	Распределение токенов	Сrapинг (Whitepaper)	Оценка по шкале (% инсайдеров <20% = 10)	10%
<b>Команда</b>	Опыт и послужной список	Сrapинг (Сайт, LinkedIn)	Оценка 1-10 на основе опыта (экзиты, FAANG, PhD)	15%
	Прозрачность	Сrapинг	Бинарная (Публичная = 1,	5%

	(Анонимность)	(Сайт)	Анонимная = 0, Красный флаг)	
<b>Технология</b>	Качество Whitepaper	Сcrapинг, LLM-анализ	Оценка 1-10 (ясность, техническая глубина, новизна)	10%
	Наличие продукта/GitHub	Сrapинг (Сайт, GitHub)	Оценка 1-10 (наличие беты, активность на GitHub)	5%
<b>Инвесторы</b>	Качество VC-бэкеров	Сrapинг (Сайт, анонсы)	Оценка 1-10 на основе уровня VC (Tier-1, Tier-2, и т.д.)	10%
<b>Рынок</b>	Социальный хайп	API (Twitter), Сrapинг	Оценка 1-10 (рост подписчиков, сентимент)	5%

### Раздел 3.3: Предсказание Будущего — Модель Машинного Обучения для Прогнозирования ROI

В то время как модель оценки определяет качество, модель машинного обучения (ML) может попытаться спрогнозировать количественный результат. Этот раздел описывает этот процесс.

- **Шаг 1: Создание Исторического Набора Данных.** Это самый важный и трудоемкий шаг.
  - **Источники Данных:** Необходимо агрегировать данные по сотням прошлых ICO. Ключевыми источниками будут:
    - **На основе API:** CoinGecko <sup>61</sup> и CryptoRank <sup>64</sup> предоставляют исторические данные о ценах, рыночной капитализации и некоторую информацию об ICO через свои API.

- **На основе скрапинга:** Сайты, такие как **ICODrops**<sup>66</sup>, содержат исторические детали ICO (сумма сбора, даты и т.д.), которых может не быть в API. Необходимо будет собрать и разобрать эти исторические данные.
  - **Ручной/Гибридный:** Некоторые качественные данные (например, состав команды на момент ICO) могут потребовать ручного восстановления или целевого скрапинга.
- **Шаг 2: Инженеринг Признаков (Feature Engineering).** Признаками для модели будут результаты нашей Матрицы оценки ICO из Раздела 3.2. Каждое историческое ICO будет строкой в нашем наборе данных, со столбцами для каждой оцифрованной метрики (Балл Команды, Балл Токеномики и т.д.).
- **Шаг 3: Определение Целевой Переменной.** Целевой переменной ( $y$ ) будет **Возврат на Инвестиции (ROI)** через фиксированный период, например, 6 месяцев.  $ROI = (\text{Цена}_{\text{б}} \text{ _месяцев _после_листинга} - \text{ЦенаICO}) / \text{ЦенаICO}$ . Эти исторические данные о ценах будут собраны из API CoinGecko/CryptoRank.
- **Шаг 4: Выбор и Обучение Модели.**
  - **Рекомендуемая Модель: XGBoost.** Для структурированных, табличных данных, как у нас, модели градиентного бустинга являются передовыми. **XGBoost** — отличный выбор благодаря своей производительности, масштабируемости и способности обрабатывать сложные, нелинейные зависимости в финансовых данных.<sup>68</sup> Существует множество примеров использования XGBoost для финансового прогнозирования.<sup>71</sup>
  - **Процесс Обучения:** Будет использоваться техника кросс-валидации, учитывающая временные ряды, такая как walk-forward optimization, для обучения и валидации модели, чтобы гарантировать отсутствие утечки будущей информации в обучающий набор.<sup>68</sup> Модель будет обучена предсказывать 6-месячный ROI на основе признаков, существовавших до ICO. Существующие проекты на GitHub показывают осуществимость таких моделей.<sup>72</sup>

Основная ценность ML-модели заключается не только в ее сыром прогнозе, но и в ее способности определять, какие признаки являются наиболее предсказательными для успеха. Модель XGBoost и другие древовидные модели могут выводить "важность признаков" (например, с помощью графиков SHAP, как показано в<sup>76</sup>), которые ранжируют входные переменные по их предсказательной силе. Это позволяет нам отвечать на более глубокие, стратегические вопросы: "Исторически, было ли Качество Инвесторов важнее, чем соотношение IMC/FDV?" или "Действительно ли активность на GitHub коррелирует с положительной доходностью?". Эта обратная связь бесценна. Она позволяет нам

уточнять веса в нашей основной Матрице оценки (Раздел 3.2) на основе эмпирических, основанных на данных доказательств, делая весь наш аналитический фреймворк умнее и эффективнее со временем. Таким образом, ML-модель становится инструментом для мета-анализа нашего собственного фреймворка.

## Раздел 3.4: Адаптация Агентной Системы — Модели Pydantic и Логика Принятия Решений

Этот заключительный раздел предоставляет конкретные детали реализации для интеграции нового фреймворка анализа ICO в существующую систему LangGraph. Он напрямую отвечает на последний вопрос пользователя.

- **Обновленные Модели Pydantic:** Существующие модели AnalysisReport и RiskAssessmentReport будут расширены для включения новых, специфичных для ICO полей. Это обеспечивает соблюдение принципа "железобетонных контрактов данных".<sup>6</sup>

Python

```
# В models/ico_models.py
from pydantic import BaseModel, Field, HttpUrl
from typing import Optional, List, Dict
from enum import Enum

class VestingSchedule(BaseModel):
    beneficiary: str = Field(description="Например, 'Team', 'Seed Investors'")
    percentage_total_supply: float
    cliff_months: int
    vesting_duration_months: int

class TokenomicsReport(BaseModel):
    initial_market_cap_usd: float
    fully_diluted_valuation_usd: float
    imc_to_fdv_ratio: float = Field(description="Initial Market Cap / FDV. Ключевой
индикатор будущей инфляции.")
    total_supply: int
    token_utility: List[str]
    vesting_schedules: List
```

```
class QualitativeScores(BaseModel):
    team_score: int = Field(ge=0, le=10, description="Оценка 1-10 за опыт и
прозрачность команды.")
    investor_quality_score: int = Field(ge=0, le=10, description="Оценка 1-10 на основе
уровня VC-бэкеров.")
    technology_score: int = Field(ge=0, le=10, description="Оценка 1-10 за
инновационность и качество whitepaper.")
    tokenomics_score: int = Field(ge=0, le=10, description="Формульная оценка на
основе количественной токеномики.")

# Обновленный AnalysisReport
class AnalysisReport(BaseModel):
    #... существующие поля...
    project_name: str
    ico_sale_date: str
    whitepaper_url: Optional[HttpUrl]
    tokenomics: TokenomicsReport
    qualitative_assessment: QualitativeScores # Заменяет одно текстовое поле
'analysis_summary'

class RecommendationEnum(str, Enum):
    INVEST = "INVEST"
    MONITOR = "MONITOR"
    REJECT = "REJECT"

class InvestmentThesis(BaseModel):
    recommendation: RecommendationEnum
    confidence_score: float = Field(ge=0.0, le=1.0)
    investment_summary: str = Field(description="Резюме инвестиционного кейса в 3
предложениях.")
    strengths: List[str] = Field(description="Список сильных сторон, основанный на
зеленых флагах.")
    weaknesses: List[str] = Field(description="Список слабых сторон, основанный на
красных флагах и низких оценках.")
    actionable_steps: List[str] = Field(description="Конкретные шаги для участия в
сейле или мониторинга.")

# Обновленный RiskAssessmentReport
class RiskAssessmentReport(BaseModel):
    #... существующие поля...
```

```

overall_ico_score: int = Field(ge=0, le=100, description="Взвешенная оценка из
Матрицы оценки ICO.")
predicted_6m_roi: Optional[float] = Field(description="Прогноз ROI от модели
XGBoost.")
key_red_flags: List[str]
key_green_flags: List[str]
# Выходной результат теперь является частью отчета
investment_thesis: InvestmentThesis

```

- **Измененная Логика Decision Agent:** Decision Agent больше не будет выдавать простой сигнал. Он будет потреблять RiskAssessmentReport и производить структурированную рекомендацию с четкими, действенными шагами.
  - **Вход:** Агент получает объект RiskAssessmentReport, который уже содержит overall\_ico\_score, predicted\_6m\_roi, key\_red\_flags и key\_green\_flags.
  - **Логика:** Новый промпт агента будет инструктировать его действовать как Инвестиционный комитет венчурного фонда. Он проанализирует входные данные и заполнит структуру InvestmentThesis.
  - **Выход:** Агент сгенерирует и вернет заполненный объект InvestmentThesis, который затем будет добавлен в RiskAssessmentReport. Пример логики для actionable\_steps:
    - Для рекомендации "INVEST": ["1. Подготовить кошелек [X] с ETH для публичной продажи.", "2. Установить напоминание в календаре на [Дата и время продажи].", "3. Пройти предварительную регистрацию на лаунчпаде [Z], если требуется."]
    - Для рекомендации "MONITOR": ["1. Добавить проект в список наблюдения.", "2. Установить оповещение на новости, связанные с новым финансированием от VC Уровня-1."]

Эта трансформация поднимает систему с уровня простого "поисковика гемов" до сложной, автоматизированной платформы для инвестиционного анализа и исполнения, напрямую выполняя запрос на создание "продвинутой системы".

## Works cited

1. Scaling Celery workers with RabbitMQ on Kubernetes - LearnKube, accessed on August 11, 2025, <https://learnkube.com/scaling-celery-rabbitmq-kubernetes>
2. Web Scraping With Celery & RabbitMQ - How to Run Thousands of Scrappers Without Losing Your Mind | ScrapeOps, accessed on August 11, 2025,

<https://scrapeops.io/web-scraping-playbook/celery-rabbitmq-scraper-scheduling/>

3. How to build docker cluster with celery and RabbitMQ in 10 minutes - Medium, accessed on August 11, 2025,  
<https://medium.com/@tonywangcn/how-to-build-docker-cluster-with-celery-and-rabbitmq-in-10-minutes-13fc74d21730>
4. Scale up Messaging Queue with Python Celery (Processes vs Threads) — Part 1, accessed on August 11, 2025,  
<https://python.plainenglish.io/scale-up-messaging-queue-with-python-celery-processes-vs-threads-402533be269e>
5. How to build a scalable crawler to crawl million pages with a single machine in just 2 hours | by TonyWang | Medium, accessed on August 11, 2025,  
<https://medium.com/@tonywangcn/how-to-build-a-scalable-crawler-to-crawl-million-pages-with-a-single-machine-in-just-2-hours-ab3e238d1c22>
6. RATIONAL APPROACH TO BUILDING AI AGENTS.docx
7. Scrapy - Use RabbitMQ only or Celery + RabbitMQ for scraping multiple websites?, accessed on August 11, 2025,  
<https://stackoverflow.com/questions/31834738/scrapy-use-rabbitmq-only-or-celery-rabbitmq-for-scraping-multiple-websites>
8. Comparing MongoDB vs PostgreSQL, accessed on August 11, 2025,  
<https://www.mongodb.com/resources/compare/mongodb-postgresql>
9. JSON performance – PostgreSQL vs MongoDB | Webinar | Umair Shahid (Stormatics), accessed on August 11, 2025,  
[https://www.youtube.com/watch?v=bvO3Khm7\\_n4](https://www.youtube.com/watch?v=bvO3Khm7_n4)
10. Postgres JSONB meets MongoDB... - Jim Blackhurst - Medium, accessed on August 11, 2025,  
<https://jimb-cc.medium.com/postgres-jsonb-meets-mongodb-part-1-the-basics-43d9fc95c40b>
11. PostgreSQL JSON field type vs MongoDB - Stack Overflow, accessed on August 11, 2025,  
<https://stackoverflow.com/questions/43881210/postgresql-json-field-type-vs-mongodb>
12. Can PostgreSQL with its JSONB column type replace MongoDB? | by Yuriy Ivon - Medium, accessed on August 11, 2025,  
<https://medium.com/@yureexus/can-postgresql-with-its-jsonb-column-type-replace-mongodb-30dc7feffaf3>
13. Powerful Job Monitoring & Scheduling for Web Scraping - ScrapeOps, accessed on August 11, 2025, <https://scrapeops.io/monitoring-scheduling/>
14. Scrape data for health care companies - Browse AI, accessed on August 11, 2025, <https://www/browse.ai/use-cases/healthcare>
15. Best Automated Web Scraping Tools & Software in 2025 (Organized by Category) - Bitcot, accessed on August 11, 2025,  
<https://www.bitcot.com/best-automate-web-scraping-tools/>
16. Decoupling: Taking your API integrations to the Next Level - GigaSpaces, accessed on August 11, 2025,

<https://www.gigaspaces.com/blog/decoupling-taking-your-api-integrations-to-the-next-level>

17. The Decoupling Principle For Future-Proof Data Architectures - Awadelrahman M. A. Ahmed, accessed on August 11, 2025,  
<https://awadrahman.medium.com/the-decoupling-principle-for-future-proof-data-architectures-9c8ace859905>
18. Data Decoupling Layer: Manage Your Data | Mia-Platform Blog, accessed on August 11, 2025, <https://mia-platform.eu/blog/data-decoupling-layer/>
19. Event-Driven Architecture - AWS, accessed on August 11, 2025,  
<https://aws.amazon.com/event-driven-architecture/>
20. Decoupled Architecture & Microservices | by Saurabh Gupta - Medium, accessed on August 11, 2025,  
<https://medium.com/@saurabh.engg.it/decoupled-architecture-microservices-29f7b201bd87>
21. Puppeteer vs Selenium vs Playwright: Best Web Scraping Tool? - PromptCloud, accessed on August 11, 2025,  
<https://www.promptcloud.com/blog/puppeteer-vs-selenium-vs-playwright-for-web-scraping/>
22. Dynamic Web Scraping tools Comparison: Selenium vs Puppeteer vs Playwright, accessed on August 11, 2025,  
<https://www.vocso.com/blog/dynamic-web-scraping-tools-comparison-selenium-vs-puppeteer-vs-playwright/>
23. Playwright vs Puppeteer: Best Choice for Web Scraping? - BrowserCat, accessed on August 11, 2025,  
<https://www.browsercat.com/post/playwright-vs-puppeteer-web-scraping-comparison>
24. Advanced Web Scraping Tactics - Pluralsight, accessed on August 11, 2025,  
<https://www.pluralsight.com/resources/blog/guides/advanced-web-scraping-tactics-python-playbook>
25. Advanced Web Scraping With Python Tactics in 2025 - Oxylabs, accessed on August 11, 2025, <https://oxylabs.io/blog/advanced-web-scraping-python>
26. Datacenter vs. Residential Proxies: Comparison Guide - Oxylabs, accessed on August 11, 2025,  
<https://oxylabs.io/blog/the-difference-between-data-center-and-residential-proxies>
27. Datacenter Vs Residential Proxies Demystified - GeeLark | The 1st Antidetect phone, accessed on August 11, 2025,  
<https://www.geelark.com/blog/datacenter-vs-residential-proxies-demystified/>
28. Datacenter vs. Residential Proxies: 5 Key Differences - anyIP, accessed on August 11, 2025, <https://anyip.io/blog/datacenter-proxies-vs-residential-proxies>
29. How to Scrape Websites with Python and Rotating Proxies - hide.me, accessed on August 11, 2025,  
<https://hide.me/en/blog/how-to-scrape-websites-with-python-and-rotating-proxies/>
30. Using Python to Rotate Proxies and Avoid Detection | by swiftproxy - Medium,

- accessed on August 11, 2025,  
<https://medium.com/@swiftproxy/using-python-to-rotate-proxies-and-avoid-detection-9689ac3845b5>
31. How to Rotate Proxies in Web Scraping - Scrapfly, accessed on August 11, 2025,  
<https://scrapfly.io/blog/posts/how-to-rotate-proxies-in-web-scraping>
  32. How to Bypass Cloudflare with Playwright in 2025 - ZenRows, accessed on August 11, 2025, <https://www.zenrows.com/blog/playwright-cloudflare-bypass>
  33. How to ByPass Cloudflare Challenges using Selenium: Tips and Tricks - BrowserStack, accessed on August 11, 2025,  
<https://www.browserstack.com/guide/selenium-cloudflare>
  34. Bypassing CAPTCHA with Playwright - ScrapingAnt, accessed on August 11, 2025, <https://scrapingant.com/blog/bypass-captcha-playwright>
  35. playwright-recaptcha - PyPI, accessed on August 11, 2025,  
<https://pypi.org/project/playwright-recaptcha/>
  36. Best CAPTCHA Solver: Top Picks That Work in 2025 - Multilogin, accessed on August 11, 2025, <https://multilogin.com/blog/best-captcha-solver-in-2025/>
  37. The Best 6 CAPTCHA Solver Tools for Automation - CapSolver, accessed on August 11, 2025, <https://www.capsolver.com/blog/All/top-6-solvers>
  38. Comparison of CAPTCHA-Solving Services: A Peek Under the Hood and a Look at the Numbers - Habr, accessed on August 11, 2025,  
<https://habr.com/en/articles/931968/>
  39. 7 Best Captcha Solvers | PrivateProxy.me, accessed on August 11, 2025,  
<https://privateproxy.me/blog/7-best-captcha-solvers/>
  40. How to Bypass CAPTCHA with Playwright - Automatically - Webshare, accessed on August 11, 2025,  
<https://www.webshare.io/academy-article/playwright-bypass-captcha>
  41. Introducing LangExtract: A Gemini powered information extraction library, accessed on August 11, 2025,  
<https://developers.googleblog.com/en/introducing-langextract-a-gemini-powered-information-extraction-library/>
  42. ICO valuation - What the blockchain industry should know - Eqvista, accessed on August 11, 2025,  
<https://eqvista.com/company-valuation/valuation-crypto-assets/ico-valuation/>
  43. Designing Effective Token Economics for an ICO - TokenMinds, accessed on August 11, 2025,  
<https://tokenminds.co/blog/blockchain-projects/designing-effective-token-economics-for-ico>
  44. ICO ROI: How to calculate the return on investment for your ICO and demonstrate its profitability and sustainability - FasterCapital, accessed on August 11, 2025,  
<https://fastercapital.com/content/ICO-ROI--How-to-calculate-the-return-on-investment-for-your-ICO-and-demonstrate-its-profitability-and-sustainability.html>
  45. What is fully diluted valuation in crypto? - CoinTracker, accessed on August 11, 2025, <https://www.cointracker.io/blog/what-is-fully-diluted-valuation>
  46. What Does Fully Diluted Market Cap (FDV) Mean In Cryptocurrency? | Mudrex

- Learn, accessed on August 11, 2025,  
<https://mudrex.com/learn/what-does-fully-diluted-market-cap-fdv-mean-in-cryptocurrency/>
47. Tokenomics - Why Should We Care About This - Empirica, accessed on August 11, 2025, <https://empirica.io/blog/tokenomics-why-should-we-care-about-this/>
48. The Ultimate Checklist Before Listing Your Token on Major Exchanges | by Blockchain App Factory | Predict | Jul, 2025 | Medium, accessed on August 11, 2025,  
<https://medium.com/predict/the-ultimate-checklist-before-listing-your-token-on-major-exchanges-c1fa121313f9>
49. 10 Key Metrics to Track for ICO Reports - TokenMinds, accessed on August 11, 2025, <https://tokenminds.co/blog/knowledge-base/ico-metrics>
50. 5 Crypto Red Flags: What to Look Out For Before You Invest - TokenMinds, accessed on August 11, 2025,  
<https://tokenminds.co/blog/crypto-learning/red-flags-to-spot-before-investing-in-a-crypto-project>
51. ICO Red Flags Every Investor Must Know | by Thomas L. McLaughlin | Blockstake | Medium, accessed on August 11, 2025,  
<https://medium.com/blockstake/ico-red-flags-every-investor-must-know-2c5af33d13a8>
52. Vesting : the Good Practices. - Medium, accessed on August 11, 2025,  
<https://medium.com/@Nomiks/vesting-the-good-practices-6a189b408131>
53. Your guide to tokens: How to design, launch, structure rights, and more - a16z crypto, accessed on August 11, 2025,  
<https://a16zcrypto.com/posts/article/guide-to-tokens/>
54. ICO 101: A beginner's guide to raising capital using cryptocurrencies - Cointelegraph, accessed on August 11, 2025,  
<https://cointelegraph.com/learn/articles/initial-coin-offering>
55. 7 Crypto Red Flags to Watch Out for in ICOs and Token Sales - Obiex Blog, accessed on August 11, 2025,  
<https://blog.obiex.finance/7-crypto-red-flags-to-watch-out-for-in-icos-and-token-sales/>
56. A Framework for Navigating Down Markets | Andreessen Horowitz, accessed on August 11, 2025, <https://a16z.com/a-framework-for-navigating-down-markets/>
57. Paradigm, accessed on August 11, 2025, <https://www.paradigm.xyz/>
58. Paradigm Shifts - Colossus, accessed on August 11, 2025,  
<https://joincolossus.com/article/paradigm-shifts-matt-huang/>
59. From Tweets to token sales: Assessing ICO success through social media sentiments - InK@SMU.edu.sg, accessed on August 11, 2025,  
[https://ink.library.smu.edu.sg/context/sis\\_research/article/10170/viewcontent/2\\_From\\_Tweets\\_to\\_Token\\_Sales\\_Assessing\\_ICO\\_Success\\_through\\_Social\\_Media\\_Se ntiments.pdf](https://ink.library.smu.edu.sg/context/sis_research/article/10170/viewcontent/2_From_Tweets_to_Token_Sales_Assessing_ICO_Success_through_Social_Media_Se ntiments.pdf)
60. A Framework for Evaluating Crypto VC Funds | by BrainGenius - Medium, accessed on August 11, 2025,  
<https://medium.com/@BrainGenius/a-framework-for-evaluating-crypto-vc-funds>

[-f0842c45787a](#)

61. Most Comprehensive Cryptocurrency Price & Market Data API | CoinGecko API, accessed on August 11, 2025, <https://www.coingecko.com/en/api>
62. 2. Get Historical Data - CoinGecko API, accessed on August 11, 2025, <https://docs.coingecko.com/docs/2-get-historical-data>
63. Coin Historical Data by ID - CoinGecko API, accessed on August 11, 2025, <https://docs.coingecko.com/reference/coins-id-history>
64. Cryptorank API, accessed on August 11, 2025, <https://api.cryptorank.io/documentation/>
65. Cryptorank API documentation, accessed on August 11, 2025, <https://api.cryptorank.io/docs>
66. ICO Drops Web Scraper: Automate ICO Data Extraction | Automatio.ai, accessed on August 11, 2025, <https://automatio.ai/templates/en/ico-drops-web-scraper>
67. ICO Drops - Calendar of active and upcoming ICO & IEO. Complete list with Token Sales, accessed on August 11, 2025, <https://icodrops.com/>
68. Implement Walk-Forward Optimization with XGBoost for Stock Price Prediction in Python, accessed on August 11, 2025, <https://blog.quantinsti.com/walk-forward-optimization-python-xgboost-stock-prediction/>
69. Use XGBoost for Time-Series Forecasting - Analytics Vidhya, accessed on August 11, 2025, <https://www.analyticsvidhya.com/blog/2024/01/xgboost-for-time-series-forecasting/>
70. Forecasting time series with XGBoost - cienciadedatos.net, accessed on August 11, 2025, <https://cienciadedatos.net/documentos/py56-forecasting-time-series-with-xgboost>
71. IBM/xgboost-financial-predictions: Use Machine Learning to Predict Bank Client's CD Purchase with XGBoost and Scikit Learn in Watson Studio - GitHub, accessed on August 11, 2025, <https://github.com/IBM/xgboost-financial-predictions>
72. KaleabTessera/ICOomen\_ML\_Model: IcoOmen, a machine learning model which will predict the value of an ICO token after 6 months. This uses historic data that has been aggregated from various public websites and APIs (Application programming interfaces), as well as data that has been manually collected and calculated. - GitHub, accessed on August 11, 2025, [https://github.com/KaleabTessera/ICOomen\\_ML\\_Model](https://github.com/KaleabTessera/ICOomen_ML_Model)
73. alimohammadiamirhossein/CryptoPredictions: CryptoCurrency prediction using machine learning and deep learning - GitHub, accessed on August 11, 2025, <https://github.com/alimohammadiamirhossein/CryptoPredictions>
74. Machine-learning forecasting of successful ICOs | Request PDF - ResearchGate, accessed on August 11, 2025, [https://www.researchgate.net/publication/361235310\\_Machine-learning\\_forecasting\\_of\\_successful\\_ICOs](https://www.researchgate.net/publication/361235310_Machine-learning_forecasting_of_successful_ICOs)
75. Identifying Likely-Reputable Blockchain Projects on Ethereum - arXiv, accessed on August 11, 2025, <https://arxiv.org/html/2503.15542v1>

76. Initial Coin Offerings Success Prediction Using Social Media and Large Language Models, accessed on August 11, 2025,  
[https://www.researchgate.net/publication/387879238\\_Initial\\_Coin\\_Offerings\\_Success\\_Prediction\\_Using\\_Social\\_Media\\_and\\_Large\\_Language\\_Models](https://www.researchgate.net/publication/387879238_Initial_Coin_Offerings_Success_Prediction_Using_Social_Media_and_Large_Language_Models)