



# Проект AMANITA: функциональная и социальная суть





---



## Функциональная суть проекта

**AMANITA** — это модульная экосистема лояльности и распределённой торговли, в которой каждый участник может выступать как продавец, промоутер или покупатель. Система построена на основе **Telegram-ботов** (для продавцов), **Invite NFT** и **внутреннего токена AMANITA Coin** и реализована на открытых блокчейн-протоколе Polygon.

### Основные компоненты:

-  **Invite NFT:**  
Цифровой NFT-пропуск, создаваемый текущими участниками системы, с функцией отслеживания рефералов. Пригласившие получают вознаграждение при первой покупке приглашённого.
  -  **Telegram-бот:**  
Персональный торговый агент, выполняющий функции:
    - продажи цифровых и физических товаров/услуг с учетом получения информации для осуществления доставки с оплатой через смарт контракт
    - генерации инвайтов и система лояльности через AMANITA coin,
    - коммуникации с покупателями,
    - Система репутации и отзывов,
  -  **AMANITA Coin:**  
Внутренний невзаимозаменяемый токен системы. Он **не подлежит торговле или передаче**, используется исключительно:
    - как вознаграждение за привлечение новых покупателей,
    - для получения скидок, доступа к бонусам или приоритетных продуктов.
  -  **Смарт-контракт продаж:**  
Обрабатывает поступления оплаты от пользователей, проверяет инвайт и начисляет AMANITA Coin пригласившему участнику.
-

## Социальная суть проекта

Проект AMANITA предлагает **новую модель этичного p2p-маркетинга**, в котором:

- **Отсутствуют централизованные платформы, агрегаторы и посредники.**
  - **Каждый продавец сам управляет своей торговой точкой** через Telegram-бота.
  - **Анонимность участников защищена** благодаря децентрализованной структуре и отсутствию KYC.
  - **Приглашение новых участников вознаграждается напрямую**, в автоматическом режиме, без комиссий и прослойки.
  - **Покупатели участвуют в построении экосистемы**, становясь потенциальными нодами/продавцами в будущем.
- 

## Социальная миссия

Проект направлен на:

- **Стимулирование честной горизонтальной торговли** — без рекламы, алгоритмов и внешнего давления.
- **Поддержку микроэкономик и локальных сообществ**, в которых каждый может быть как производителем, так и распространителем ценности.
- **Развитие цифровой культуры распределённого доверия**, основанной на прозрачных, но не контролирующих технологиях.

AMANITA — это **органическая сеть торговли и лояльности**, не нуждающаяся в централизованном управлении, но способная масштабироваться за счёт интересов и активности самих участников.

---



## Техническое задание для MVP



Цель:

Развернуть MVP, в котором один Telegram-бот управляет продажами и инвайтами через Invite NFT. После покупки, токен AMANITA Coin начисляется автоматом через смарт-контракт.

---

## 1. Контракты на Polygon (Solidity)

### InviteNFT.sol

Контракт **InviteNFT** отвечает за создание уникальных инвайтов, которые представляют собой **цифровые токены доступа (ERC-721)**, используемые исключительно **внутри экосистемы AMANITA**. Эти инвайты позволяют отслеживать, кто пригласил нового участника, и активировать внутреннюю программу лояльности.

**Invite NFT создаётся и назначается новому пользователю в момент активации invite-кода.**

- Владелец NFT получает:
  - Доступ к функционалу Telegram-бота (экосистема).
  - Право создавать новые invite-коды (ограничено, например, 12).
- Invite NFT выступает:
  - Как цифровой маркер доверия.
  - Как идентификатор, подтверждающий право участия.
  - Как логическая связь между участниками: **inviter** → **invitee**.

Invite NFT:

- **Не может быть повторно активирован.**
- Может быть **использован как пропуск** для доступа в функционал.
- Должен отображаться в интерфейсе бота с подписью: "**Вы были приглашены пользователем X**".

Чтобы избежать правовой классификации таких токенов как регулируемых криптоактивов (в том числе по MiCA и будущим поправкам ЕС к NFT), в контракт закладываются следующие **ограничения и принципы архитектуры**.

## Процесс генерации и активации Invite NFT

- Invite NFT создаётся **\*\*в момент активации inviteCode новым пользователем\*\***.
- При переходе по invite-ссылке (или вводу кода в Telegram-боте), система:
  - Проверяет допустимость inviteCode (off-chain или on-chain).
  - Определяет `inviter` (через маппинг или базу).
  - Вызывает функцию `mintInvite(inviter, newUser)` от имени нового пользователя.
  - Новый пользователь становится `owner` NFT и получает доступ в экосистему.
  - Его Invite NFT — это одноразовый цифровой маркер доверия.
- **Invite NFT служит:**
  - Пропуском в экосистему;
  - Источником права генерировать собственные инвайты;
  - Связующим звеном между участниками (для системы лояльности).

## Функционал приглашений (Модель с “отложенным владельцем” и off-chain меткой)

Каждый пользователь получает **12 инвайтов**, реализуемых как NFT.

Передача происходит **через Telegram-бот**, и инвайт **содержит уникальный код**.

Цель — избежать необходимости устанавливать криптокошелёк до момента, когда пользователь станет активным.

**Генерация Invite NFT происходит в момент активации кода новым пользователем.**

- При получении inviteCode (например, `invite_3N4XQ8`) пользователь заходит в Telegram-бот.
- Бот проверяет валидность inviteCode (off-chain или через mapping).

- Если код допустим, бот вызывает `mintInvite(address inviter, address newUser)`:
  - Создаётся Invite NFT:
    - `owner = newUser`
    - `inviterOf[tokenId] = inviter`
    - `used = true` (или `active = true`)
  - Новый участник становится **владельцем** NFT и получает доступ к экосистеме.
- Участник получает свои 12 Invite NFT с возможностью их распространения (через бота).

Как это работает:

1. При генерации Invite NFT:
  - NFT создаётся на смарт-контракте с `owner = bot` или системный адрес.
  - Каждому инвайту присваивается уникальный `inviteCode`, и он **помечается как “не активированный”**.
2. Пользователь получает **изображение с QR-кодом** или **просто ссылку с этим `inviteCode`**.
  - Пример: `https://t.me/XXX_bot?start=invite_3N4XQ8`
3. Получатель (новый пользователь):
  - Заходит в бота по этой ссылке.
  - Бот проверяет invite-код.
  - Если код валиден и ещё не активирован:
    - Назначается `owner = msg.sender` (или Telegram ID).
    - NFT переводится или связывается с новым пользователем.
    - Выполняется `mintReward()` пригласившему.

## Плюсы:

- Не нужно криптокошелек на этапе приглашения.
- Поддерживает анонимность.
- Invite можно отправить как картинку, ссылку или QR.

## Внимание!!!

- Нужно аккуратно хранить `inviteCode` → `tokenId` → `inviter` off-chain или в контракте.



Что будет на Invite NFT (визуально/данные)

### 1. Изображение (визитка)

- Статичная или анимированная картинка.
- Внизу: `AMANITA Invite`
- Код: `Invite Code: 3N4XQ8`
- Или QR: `https://t.me/XXX_bot?start=invite_3N4XQ8`

### 2. Что зашито в `inviteCode`

- Уникальный ID NFT или slug (`tokenId` или `hash(issuer, #)`).
- Необязательно делать его криптографически сложным — 8 знаков достаточно.

Механизм передачи Invite NFT через Telegram без кошельков

#### 1. Определить формат `inviteCode`:

- UUID, short hash или ID.
- Сопоставление с `tokenId` в контракте.

#### 2. Разработать `mintInvite(address inviter)` в контракте `InviteNFT.sol`, который:

- Привязывает `tokenId` → `inviter`.
- Устанавливает `used = false`.

#### 3. Создать Telegram-бот-функционал:

- Команда `/generate_invites` → возвращает 12 кодов + картинки.
- Команда `/start invite_xxxxx` → активирует NFT и регистрирует нового пользователя.

#### 4. Контрактную часть:

- Функция `activateInvite(tokenId, newOwner)` вызывается бэкендом один раз.
- После чего `used = true`, и NFT больше не может быть повторно активирован.

## 5. Защита:

- Invite можно активировать **только один раз**.
- Если попытка повторного использования — отклоняется.

---

## Безопасность и UX

- Invite **не требует Web3-кошелька до активации**.
- NFT **живёт на контракте**, пока не будет "востребован".
- Код можно распространять как **текст, QR, картинку, ссылку**.
- Telegram-бот **всё делает за пользователя**.

## Особенности контракта

### 1. Уникальность и неделимость (Non-fungibility)

- Контракт реализует стандарт **ERC-721**.
- Каждый Invite NFT — **уникальный**, с неизменяемым `tokenId`, созданным системой.
- Никакие токены не являются взаимозаменяемыми, не агрегируются и не дробятся.

---

### 2. Ограниченная функциональность

- Invite NFT **не является предметом коллекционирования или обмена**.
- Не содержит вложенной ценности или обещания доходности.
- Используется только как **механизм доступа к системе и метка приглашения**.

---

### 3. Ограниченный вторичный оборот

- По умолчанию, метод `transferFrom()` не отключён, но:
  - В реализации Telegram-бота отсутствует функциональность передачи.
  - Необходимо ограничить передачу **только один раз** (например, после использования NFT становится "неактивным").
- Опционально: реализовать **burn-on-use** или `active/inactive` флаг.

```
mapping(uint256 => bool) public isActive;

function markUsed(uint256 tokenId) external onlySaleContract {
    require(isActive[tokenId], "Already used");
    isActive[tokenId] = false;
}
```

---

#### ✓ 4. Отслеживание происхождения (реферальная логика)

Контракт хранит **адрес пригласившего пользователя**:

```
mapping(uint256 => address) public inviterOf;
```

- Используется исключительно смарт-контрактом продаж для распределения наград.
- Эти данные не используются для идентификации или мета-анализа вне системы.

---

#### ✓ 5. Ограниченное применение

- Invite NFT **не предоставляет автоматический доступ к финансовым сервисам**.
- Он:
  - Выполняет стандартную функцию NFT-Gate чтобы допускать пользователя к функционалу боту.



- Не даёт право на AMANITA Coin напрямую.
  - Не является активом.
  - Лишь идентифицирует логическую связь между пользователями (кто кого пригласил). Позволяет держать экосистему чистой от скаммеров и посторонних так как за возможность участвовать за тебя должен кто-то поручиться. А количество инвайтов умышленно ограничено.
- 

#### ✓ 6. Без внешнего маркетинга или торговых предложений

- В пользовательских соглашениях и публичных материалах **не допускается описание Invite NFT как инвестиции или объекта коллекционирования.**
  - Он **не должен представляться как цифровое имущество с ценностью.**
- 

#### ✓ 7. Автоматическое или ограниченное создание

- Invite NFT выпускается:
    - Только **из интерфейса Telegram-бота.**
    - В ответ на запрос пользователя, **при условии, что он уже активный участник.**
  - Таким образом исключается массовая генерация и спекулятивное накопление NFT.
- 

## Заключение

Invite NFT в проекте AMANITA:

- Не является ни финансовым активом, ни цифровым объектом коллекционной ценности.
- Используется как **одноразовый маркер доверия** между пользователями.
- Не предоставляет доступ к рынкам или функциям хранения стоимости.
- Ограничен в применении, передаче и маркетинге.

👉 Это делает его юридически безопасным с точки зрения действующего регулирования MiCA и потенциальных нормативов ЕС в сфере NFT.

## AmanitaSale.sol

### 1. Цель контракта AmanitaSale

Контракт `AmanitaSale.sol` выполняет **центральную роль** в системе AMANITA:

- Обработывает покупки пользователей. Продажи идут через стейблкоин на Ethereum.
- Проверяет валидность Invite NFT — можно осуществлять покупку только если получен Invite NFT что дает членство в экосистеме Amanita. Это очень важная проверка.
- Если это **первая покупка** с этого адреса, то начисляет **AMANITA Coin** тому, от которого был получен этот Invite NFT (тот кто пригласил в систему).
- Эмиттирует события для Telegram-бота (frontend) фактически используя все best practices из e-commerce процессов, которые обычно происходят на сайтах и маркетплейсах.

**Для обеспечения масштабируемости и поддержки множества независимых продавцов, каждая торговая единица (бот или Telegram-продавец) имеет свой payout-адрес.**

Адрес выплаты задаётся администратором через функцию:

```
function setSellerPayoutAddress(uint256 sellerId, address payout)
external onlyAdmin;
```

- 
- При продаже:
  - Telegram-бот передаёт sellerId.

Контракт определяет адрес получения выплаты:

```
address recipient = sellerPayoutAddress[sellerId];

require(recipient != address(0), "Seller not registered");

transferFunds(buyer, recipient, amount);
```

- 
- Это позволяет:
  - Разделять доходы продавцов.

- Поддерживать приватность и масштабируемость.
- Автоматизировать управление экосистемой с множеством точек входа.

## Система идентификации продавцов и адресов выплат

- Каждый Telegram-продавец имеет уникальный `sellerId`, определяющий:

- Его payout-адрес;
- Привязку заказов;
- Отображение в интерфейсе.

- Payout-адрес задаётся через:

```
function setSellerPayoutAddress(uint256 sellerId, address payoutAddress) external
onlyAdmin;
```

### В момент покупки:

Telegram-бот указывает sellerId при вызове buy(...).

Контракт использует sellerPayoutAddress[sellerId] для перевода средств.

### sellerId может быть:

Хешом от Telegram ID;

UUID;

Сгенерированной меткой на backend.

### Необходимо обеспечить:

Uniqueness sellerId;

Маппинг TelegramBot ↔ sellerId;

Безопасное администрирование sellerId при масштабировании.



Связь inviteCode ↔ tokenId ↔ inviter

**\*\*Off-chain / On-chain связка и архитектура валидации invite-кодов\*\***

- inviteCode создаётся заранее и рассылается off-chain.

- При активации кода:
    - inviteCode проверяется ботом (по базе данных);
    - Определяется связка ``inviteCode` → inviter``;
    - Контракт mint'ит NFT с привязкой к ``inviter`` и ``msg.sender``.
  - Данные inviteCode ↔ inviter хранятся:
    - В PostgreSQL (или Redis) на backend;
    - Продублированы в IPFS/Arweave для восстановления;
    - Защищены checksum или подписью (``HMAC(inviteCode + секрет)``).
  - Резервное восстановление:
    - Каждая активация логируется с timestamp, inviteCode, Telegram ID, wallet.
    - Регулярные бэкапы хранилища данных.
- 

## 2. Ключевые функции и логика

Вот основная функциональная структура, исходя из описания:

### ♦ `buy(uint256 inviteTokenId)`

- **Параметр:** `inviteTokenId` — ID NFT, представленного как invite.
  - **Логика:**
    - Проверить валидность Invite NFT.
    - Проверить если это первая покупка. Если так то вызвать `distributeReward`.
    - Перевести сумму покупки на адрес продавца. Пока это фиксированный адрес. Надо заранее продумать как в последствии можно менять адрес продавца с целью анонимизации и просто безопасности.
    - Выпустить событие про совершение сделки.
- 

### ♦ `distributeReward(address inviter, uint256 amount)`

- Вызывается только внутри `buy()` или доверенными функциями.
  - Выпускает AMANITA Coin на адрес пригласившего (от кого пришел Invite NFT который для текущего адреса служит доступом в экосистему).
  - Может включать логики caps, уровней или зависимости от суммы покупки (опционально).
- 

#### ♦ `setTokenAddress(...)`, `setNFTAddress(...)`

- Административные методы: задают адреса токенов и NFT-контракта, адрес продавца.
  - Нужно иметь возможность админу контракта назначать новых продавцов и как-то мапить их на названия их ботов, либо какие-то другие идентификаторы.
  - Только для `owner` или `onlyAdmin`.
- 

### 3. Взаимодействие с другими контрактами




- **InviteNFT.sol**
    - Проверка действительности инвайта: `isActive(tokenId)`
    - Получение пригласившего: `inviterOf(tokenId)`
    - Деактивация инвайта: `markUsed(tokenId)`
  - **AmanitaToken.sol**
    - Вызов `mint(address to, uint amount)` для начисления токенов. Нужно иметь возможность всегда минтить так как использованные токены будут сжигаться и этот токен не для трейдинга, то есть мы не боимся инфляции.
  - **ERC-20 токен (например, USDT):**
    - Получение платежа через `transferFrom(buyer, address(this), price)`
-

## 4. Безопасность и контроль

### Важные аспекты:

- **Ограничение одного использования инвайта:**
    - `inviteTokenId` → должен быть `"used = false"` перед покупкой.
    - После покупки → `used = true`.
  - **Проверка разрешения на `transferFrom`:**
    - Убедиться, что покупатель одобрил смарт-контракту списание USDT.
  - **Реентерабельность:**
    - Все внешние вызовы (например, `mint`) должны вызываться после обновления состояний.
  - **Проверка данных:**
    - Покупатель не может использовать чужой `invite` более одного раза.
- 

## 5. Как этот контракт поддерживает цели AMANITA

Цель проекта	Как реализуется в контракте
 Инвайт → один клиент	<code>markUsed(tokenId)</code> запрещает повторное использование
 Вознаграждение пригласившего	<code>mint(inviter, amount)</code> через <code>distributeReward()</code>
 Интеграция с Telegram-ботом	<code>emit</code> события позволяют боту обновлять интерфейс

💰 Оплата в стейблкойне

`transferFrom(buyer, this, amount)` — чистый UX

⚖️ Логика реферальной честности

Валидация NFT, однократность активации

---

## 📖 6. Модель соответствия MiCA / юридическим рискам

- Этот контракт:
  - Не работает как маркетплейс.
  - Не удерживает средства на долго.
  - Не выпускает торгуемый токен напрямую пользователям.
  - Использует **внутренний utility-токен** без вторичного рынка.
- Следовательно, он **не попадает под MiCA сам по себе, при условии**, что:
  - AMANITA Coin не торгуется.
  - Invite NFT не рассматриваются как инвестиционный актив.

---

## 🔧 Потенциальные улучшения / дополнения

- Добавить возможность установки **лимитов** по количеству наград в день/период.
- Ввести **роли или уровни** (например, Silver/Gold пригласитель).
- Подключить модуль **скидок** за оплату AMANITA вместо стейблкойна (в будущем).

## 📄 `AmanitaToken.sol`

Контракт токена **AMANITA Coin** построен с учётом ограничений **регламента MiCA (Markets in Crypto-Assets Regulation)**, действующего в Европейском союзе. Основная

цель архитектуры токена — обеспечить, чтобы он **не подпадал под классификацию “crypto-asset”**, определяемую MiCA.

Для этого в контракт заложены следующие **функциональные ограничения и принципы**:

---

### ✓ 1. Невзаимозаменяемость в правовом смысле

- Хотя AMANITA реализован на стандарте ERC-20, он **не предназначен для обращения на вторичном рынке**.
  - Он используется исключительно **внутри экосистемы Amanita**, и не имеет ценности вне неё.
- 

### ✓ 2. Ограниченная обращаемость (non-transferable token)

- Метод `transfer()` заблокирован. Пользователи не могут передавать токен друг другу.
- Методы `approve()` и `transferFrom()` также отключены — это предотвращает использование токена в сторонних dApps, DEX'ax и рынках.

```
function _beforeTokenTransfer(address from, address to, uint256
amount) internal override {

    if (from != address(0) && to != address(0)) {

        revert("AMANITA token is non-transferable");

    }

}
```

---

### ✓ 3. Отсутствие рыночной функции

- Токен **не продаётся и не покупается** — он начисляется **только как внутренняя награда** за действия внутри системы (приглашение новых



пользователей, покупки и т.п.).

- **Обратный обмен на стейблкоины, другие токены или фиат невозможен.**
- 

#### ✓ 4. Замкнутая экосистема

- AMANITA Coin используется только внутри Telegram-бота и подключенных смарт-контрактов:
    - Для получения скидок.
    - Для доступа к закрытому контенту или функциональности.
    - Для персонализированных бонусов.
- 

#### ✓ 5. Централизованное управление mint/burn

- Токены **начисляются (mint)** исключительно смарт-контрактом продаж ([AmanitaSale.sol](#)) при выполнении чётко определённых условий.
  - Токены **сжигаются (burn)** только внутри экосистемы, при активации бонусов или доступа.
- 

#### ✓ 6. Прямое указание на неинвестиционную природу

- В технической документации и пользовательском соглашении системы закреплено, что AMANITA Coin:
    - Не является средством накопления.
    - Не предполагает прироста стоимости.
    - Не является инвестиционным инструментом.
- 



Таким образом, AMANITA Coin:

- Не предназначен для свободного оборота.
- Не выполняет функции инвестиционного актива.
- Используется только как **лояльный балл** в замкнутой системе.

👉 Эти условия позволяют **юридически классифицировать токен как вне поля действия MiCA**, до тех пор, пока не будет расширена его функция до рыночного обращения или листинга.



## Order NFT — Заказ как ончейн-объект

AmanitaSale.sol

- ↳ при успешной оплате
- ↳ mint OrderNFT(tokenId) → buyer
- ↳ привязка tokenId к seller

**Order NFT** — это уникальный цифровой маркер, фиксирующий факт завершённой покупки в экосистеме AMANITA. Он создаётся автоматически при успешной оплате через смарт-контракт продажи ([AmanitaSale.sol](#)) и выступает в роли **ончейн-квитанции**, доступной как для пользователя, так и для системы.

### ♦ Назначение:

- Подтверждение факта покупки.
- Привязка пользователя к конкретному заказу.
- Возможность построения истории покупок.
- Основа для создания отзывов (каждый отзыв может ссылаться на конкретный заказ).
- Будущая поддержка споров, возвратов, доставки и других пост-продажных действий.

### ♦ Поведение:

- NFT автоматически создаётся после успешной транзакции.
- Закрепляется за покупателем.
- Содержит ссылку на продавца, дату, ID, и опциональные метаданные о заказе.

- Не предназначен для передачи или торговли.

## ★ Review NFT — Отзыв как цифровая запись доверия

ReviewNFT.sol — Токен про отзыв (комментарий и оценки по разным критериям)

- ↳ mint при наличии OrderNFT
- ↳ ссылкой на tokenId покупки

**Review NFT** — это цифровой отзыв, создаваемый пользователем после совершения покупки. Он представляет собой уникальный ончейн-объект, содержащий оценку и краткий комментарий, привязанный к конкретному продавцу и заказу.

### ♦ Назначение:

- Формирование репутации продавцов на основе отзывов, поступающих от реальных покупателей.
- Создание прозрачной, неизменяемой и доступной для всех истории взаимодействий.
- Возможность агрегировать и анализировать данные для отображения рейтингов, повышения доверия и мотивации качества.

### ♦ Поведение:

- Создаётся через Telegram-бот, после выполнения команды `/leave_review`.
- Доступен только покупателю, у которого уже есть хотя бы один Order NFT, связанный с данным продавцом.
- Хранит ссылку на Order NFT (т.е. отзыв "привязан" к конкретной покупке).
- Содержит оценку (1–5) и текст комментария, хранимый в IPFS или другом децентрализованном хранилище.
- Является частью публичной истории продавца, которую можно анализировать, но не редактировать.



## Система заказов и репутаций: Как это работает вместе

1. **Покупка** → создаётся **Order NFT**.
2. Пользователь завершает сделку и получает доступ к команде `/leave_review`.

3. В интерфейсе Telegram-бота он оставляет **оценку и комментарий**.
  4. Создаётся **Review NFT**, ссылающийся на **Order NFT** и закреплённый за продавцом.
  5. Все отзывы отображаются в интерфейсе продавца, а система использует их для формирования общего рейтинга.
- 

## Преимущества такой архитектуры

- Полная **прозрачность и проверяемость отзывов**.
  - Исключена накрутка и фальшивые оценки — отзыв возможен **только при наличии реальной покупки**.
  - Простая масштабируемость (любое приложение может считать отзывы on-chain).
  - Гибкость: в будущем можно использовать **Order NFT** как основу для логистики, arbitration или даже пост-продажных миссий.
- 

## Telegram bot

### Функции

- **/invite** → генерирует Invite NFT (запрашивает адрес).
  - **/buy** → дает инструкции по оплате (с привязкой к **tokenId**).
  - Мониторинг событий контракта (через Infura/Alchemy).
  - Отображение состояния: "Ты заработал AMANITA", "У тебя X приглашённых".
- 

## Механизм взаимодействия Python ↔ смарт-контракт

### Используем:

- [Web3.py](#) — основная библиотека для Python ↔ EVM-сетей (Polygon).
- **Infura / Alchemy** — для получения событий в реальном времени (WebSocket/Webhook).
- **PolygonScan API** (если нужно сверять вручную).

### ✓ Примеры:

```
from web3 import Web3

# Подключение к Polygon через Infura
w3 =
Web3(Web3.HTTPProvider("https://polygon-mainnet.infura.io/v3/YOUR_INFURA_KEY"))

# Загрузка ABI и адрес контракта
contract = w3.eth.contract(address="0xYourSaleContract",
abi=sale_abi)

# Вызов функций:
tx = contract.functions.buy(inviteTokenId).build_transaction({...})
```

- Можно подписаться на события (`eventFilter`) или использовать `web3.middleware` + `async loop`.

---

## Архитектура (как масштабируемая нода)

flowchart TD

```
User[Пользователь в Telegram]
Bot[Бот продавца] --> NFT[Mint Invite NFT]
Bot --> Buy[Инструкция по оплате]
Buy --> Contract[Смарт-контракт продажи]
Contract --> Reward[Награда AMANITA inviter'y]
Contract --> Notify[Событие → Бот]
Bot --> User[Отображение в боте]
```

## ✦ Чтобы масштабировать:

- Бот имеет параметризованную конфигурацию:
  - `SELLER_ID`, `SELLER_WALLET`, `NFT_PREFIX`
- Все логики (mint, sale) происходят **на одном контракте**, с данными, кто кого пригласил.
- При клонировании бота → меняется только привязка продавца.

## Роли и защита Telegram-бота

- Бот **\*\*не вызывает mint напрямую\*\***, а инициирует транзакции от имени пользователя или custodial-адреса.
- Только бот имеет право вызывать чувствительные методы:
  - `activateInvite(tokenId, user)``
  - `mintInvite(inviter, user)``
- Все вызовы проверяются через:
  - SignedMessage;
  - JWT-подпись от бэкенда;
  - Custodial wallet.
- **Внедрены меры:**
  - Rate-limiting;
  - Защита от повторного использования inviteCode;
  - Логирование всех действий (в отдельной базе).
- Структура вызовов Web3:  
  
`w3.eth.contract(address, abi).functions.activateInvite(...).transact({from: custodial_wallet})`

## FAQ

### ✦ Инвайты и вход в экосистему

#### ♦ Как я получаю доступ в экосистему AMANITA?

Вы должны получить **inviteCode** от действующего участника. Это может быть:

- текстовый код (`invite_3N4XQ8`),
- ссылка (`https://t.me/amanita\_bot?start=invite\_3N4XQ8`),

- QR-код.

#### ♦ Что происходит при активации invite-кода?

Когда вы переходите по ссылке и запускаете Telegram-бот:

1. Бот проверяет действительность кода.
2. Если он не использован, вызывается смарт-контракт `mintInvite()`.
3. Вы становитесь владельцем **Invite NFT**, дающим доступ в экосистему.

#### ♦ Могу ли я приглашать других?

Да. После активации вашего invite:

- Вы получаете право на генерацию **до 12 собственных invite-кодов**.
- Ваш Invite NFT сохраняется в кошельке, и вы можете делиться кодами с другими.

#### ♦ Что будет, если мой invite-код уже использован?

Вы получите уведомление об ошибке. Вам нужно запросить новый код у другого участника.

---

### Безопасность, идентичность и приватность

#### ♦ Мне нужен Web3-кошелёк для участия?

Нет. Все действия — через Telegram. Кошелёк создаётся на этапе активации инвайта автоматически (внутренне или custodial-адресом).

#### ♦ Есть ли KYC или проверка личности?

Нет. В системе не используется KYC. Ваша идентичность — это ваш Invite NFT и Telegram ID. Это обеспечивает анонимность и децентрализацию.

#### ♦ Что делать, если я потерял доступ к Telegram?

Пока нет поддержки переноса Invite NFT между аккаунтами. Для продвинутых пользователей будет реализован механизм восстановления через seed или подпись.

---

### Покупки и выплаты

#### ♦ Как совершаются покупки?

Вы выбираете товар/услугу в Telegram-боте и следуете инструкции. Оплата происходит через USDT, транзакция проводится смарт-контрактом `AmanitaSale`.

#### ♦ Кто получает деньги за покупку?

Каждый продавец имеет свой уникальный `sellerId`, привязанный к payout-адресу. Контракт переводит средства на этот адрес автоматически.

#### ♦ Где я могу увидеть историю моих покупок?

После каждой покупки вы получаете **Order NFT**, который служит ончейн-квитанцией. Историю можно просмотреть в интерфейсе Telegram-бота или через обозреватель блокчейна.

---

### 🌟 Отзывы и репутация

#### ♦ Как я могу оставить отзыв?

После завершения покупки:

1. Используйте команду `/leave_review`.
2. Вы оставляете оценку (1–5) и комментарий.
3. Генерируется **Review NFT**, связанный с вашим заказом.

#### ♦ Почему нельзя подделать отзывы?

Отзывы возможны **только при наличии Order NFT**, т.е. после реальной покупки. Это исключает накрутку.

#### ♦ Где отображаются отзывы?

В интерфейсе Telegram-бота продавца, а также публично через on-chain статистику.

---

### 🔗 Технические моменты и сбои

#### ♦ Где хранятся invite-коды и связи между ними?

- `inviteCode` → `inviter` → `tokenId` сохраняются:
  - в базе данных (например, PostgreSQL),
  - дублируются в IPFS для восстановления.

#### ♦ Что произойдёт, если Telegram-бот недоступен?

- Функциональность временно приостановится.
- On-chain данные (NFT, история, покупки) сохраняются.



- После восстановления вы сможете продолжить с текущего состояния.

- ♦ **Что если бот не показал, что я получил AMANITA Coin?**

Проверь транзакцию на PolygonScan. Бот может не обработать событие моментально — это UI-проблема, не влияющая на токеномику.

---

## Для продвинутых пользователей

- ♦ **Можно ли использовать собственный кошелёк (Metamask)?**

В будущем будет добавлена возможность подключить или привязать внешний кошелёк. Сейчас это ограничено внутренней системой Telegram.

- ♦ **Могу ли я видеть свои NFT вне Telegram?**

Да. Все NFT on-chain — вы можете просматривать их через PolygonScan, OpenSea (если они публичные), или через интерфейс бот-проводника.

- ♦ **Как работают роли в контракте?**

- Только Telegram-бот может вызывать функции mint и активации.
- Админ может устанавливать payout-адреса для продавцов.

## Фокусировка на особых деталях

### Invite NFT

**? Новые 12 Invite NFT создаются при активации invite-кода, а не заранее ботом?**

✓ NFT создаётся в момент активации inviteCode через функцию `mintInvite(inviter, newUser)`.

---

**? Как и где определяется inviter, если NFT ещё не существует на момент рассылки inviteCode?**

✓ Связка `inviteCode` → `inviter` хранится в PostgreSQL и дублируется в IPFS с HMAC-подписью. Это обеспечивает достоверность даже при восстановлении данных после сбоя.

---

## ? Механизм предотвращения повторного использования кода

✓ Используется флаг `isActive[tokenId]`, и функция `markUsed(tokenId)` делает код "неактивным" после активации. Повторное использование отклоняется.

---

## Адреса продавцов

### ? Как система распознаёт, какой продавец обрабатывает покупку?

✓ Продавец идентифицируется через `sellerId`, который передаётся ботом в функцию `buy(...)` смарт-контракта.

---

### ? Как идентифицируются продавцы: через `sellerId`, Telegram ID, или другой способ?

✓ `sellerId` может быть:

- Хешем Telegram ID;
- UUID;
- Backend-сгенерированной меткой.

Рекомендуется обеспечить уникальность и консистентный маппинг через централизованный админ.

■ Требуется решение от тебя:

- Как именно генерировать `sellerId`?
  - `hash(telegramId + secret)`?
  - UUIDv4?
  - Кастомная метка?

**Рекомендую:** UUIDv4 + база `sellerMeta`, чтобы сохранить независимость от Telegram ID (на случай смены логина/аккаунта).

---

## ? mapping `sellerPayoutAddress[sellerId]`?

✓ Маппинг задаётся функцией `setSellerPayoutAddress(uint256 sellerId, address payout)` и используется при каждой покупке.

---

## Telegram-бот

### ? Пояснено ли, что бот не минтит инвайты, а инициирует процесс от лица нового пользователя?

✓ Да. Вся логика транзакций описана через вызовы Web3 с custodial-адреса или от имени пользователя. Бот лишь инициирует и подписывает вызовы.

---

### ? Есть ли описание, как бот проверяет `invite`-код перед вызовом `mint`?

✓ Да. `inviteCode` проверяется ботом по БД, валидируется по HMAC, затем вызывается `mintInvite()` при прохождении проверки.

---

### ? Зафиксированы ли роли и безопасность бота в архитектуре?

✓ Да. Подписи, JWT, ограничения по доступу, логирование, `rate-limiting` и разграничение доступа описаны подробно.

---

## Off-chain / on-chain связь

### ? Описана ли система логирования активаций `inviteCode` → `wallet`?

✓ Да. При каждой активации логируются:

- `inviteCode`
- `inviter`
- `wallet`
- Telegram ID

- timestamp

Все данные хранятся в PostgreSQL, дублируются в IPFS.

---

**? Есть ли указание, где именно хранится эта информация (БД, IPFS и т.п.)?**

**✓ Да:**

- Primary — PostgreSQL или Redis.
  - Backup/восстановление — IPFS/Arweave.
  - Подписи — HMAC checksum для каждого inviteCode.
- 

**? Указано ли, как решается проблема восстановления связи при сбое?**

**✓ Да.** Описаны регулярные бэкапы, логгирование, возможность верификации кода по подписи HMAC — даже если БД будет утеряна, данные можно восстановить из IPFS с верификацией.

---



## **Безопасность и MiCA**

**? Отражено ли, что ни один токен не торгуется и не несёт инвестиционной функции?**

**✓ Да, явно:**

- AMANITA Coin не продаётся, не передаётся и не торгуется;
  - Invite NFT — не объект коллекционирования, не актив с ценностью.
- 

**? Прописаны ли меры против использования токенов вне Telegram-экосистемы?**

✓ Да:

- AMANITA Coin: `transfer`, `approve`, `transferFrom` отключены.
- Invite NFT: не имеет передачи через Telegram-интерфейс, функция передачи отключается после активации.
- Пользовательское соглашение запрещает воспринимать токены как инвестицию.