ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Образовательная программа «Прикладная математика и информатика»
Специализация «Распределенные системы»

План курсовой работы

на тему «Реализация NFT маркетплейса на базе Discord API»

Выполнил:

студент группы БПМИ19Х XXXXXXXXXXXXXXXXXX _____
Подпись

_____
Дата

Принял:

руководитель проекта XXXXXXXXXXXXXXXXXX

Дата проверки _____2022 _____ _____
Оценка Подпись

Москва 2022

СОДЕРЖАНИЕ

Аннотация

Первый глобальный старт блокчейна был в 2009 году с криптовалютой, которая называется Bitcoin - распределенная децентрализованная платежная система, разработанная Сатоси Накамото[1]. С тех самых пор многие приложения начали формировать так называемый web3. Если изначальная идея блокчейна была возможностью передачи электронной валюты по средствам peer-to-peer без каких-либо посредников, например, как банки, то на сегодняшний день блокчейн - это распределенный реестр, который представляет как аспекты вычислений, так и хранения данных, которые согласуются на основе консенсуса. Представленные возможности реализуются с помощью, так называемых, smart-контрактов - некоторые куски кода, которые выполняются непосредственно в блокчейне.

Вместе с концептами блокчейна и smart-контрактов нам пришло понятие NFT(non-fungible token). NFT - это уникальный криптографический токен, который не может быть замещен другим таким же, данный токен представляет некоторые цифровой объект - файл, текст или некоторый медиа объект. На идее NFT существует множество площадок, которые позволяют обмениваться данными токенами. Цель нашего проекта - реализовать свою такую площадку, ознакомиться с основными теоретическими понятиями связанных с этим и описать их в данном плане.

Постановка задачи

Для того, чтобы заняться платформой требовалось выбрать блокчейн, посредствам которого все это будет реализовано, мы выбрали NEAR protocol[2]. NEAR protocol - это децентрализованная платформа, которая обеспечивает идеальную среду для разработки DApps.

Определение. *DApps — это приложения, которые включают логику работы с функциями блокчейна [1].*

NEAR protocol работает по схеме Proof-of-Stake(Pos), от других блокчейнов, его отличает большая пропускная способность, скорость, улучшенная масштабируемость а также, что для нас стало решающим фактором - это его дружественность к разработчикам(developer friendly) и предоставляет огромное количество источников для изучения их инструментария.

В DApps самой значимой частью кода являются Smart-контракты. Копии Smart-контрактов разворачивается с помощью специальной транзакции на всех узлах-участниках(валидаторов) и исполняются в сети блокчейна.

Определение. *Smart-контракт — это неизменяемый исполняемый код, представляющий логику DApp, работающий непосредственно в блокчейне [1]. Часто сокращают до слова контракт. В некоторых протоколах называют по-другому, например в Solana[3] - это программы[4].*

[1)] Сатоси Накамото на самом деле псевдоним человека или группы людей. https://en.wikipedia.org/wiki/Satoshi_Nakamoto

[2)] https://near.org/

[3)] https://solana.com/

[4)] https://spl.solana.com/

В разных блокчейнах - разный язык программирования для написания Smart-контрактов. Near protocol предоставляет некоторый функционал для написания Smart-контрактов на языках Rust и AssemblyScript [2](near-sdk-rs[5] и near-sdk-as[6] соответственно). Авторы не рекомендуют использовать AssemblyScript, отдавая свое предпочтение больше Rust для написания контрактов.

Каждый smart-контракт в Near(написанный на Rust/Assembly Script) переводится в WebAssembly(Wasm), который непосредственно исполняет виртуальная машина на участвующем узле(валидаторе) блокчейна. У smart-контракта, есть два вида функций: которые меняют состояние блокчейна - «change operations» и так называемые «view operations», которые не меняют состояние машины, из названия данных операций можно понять, что первый вид операций, что-то сохраняет в блокчейн, а другая получает некоторую информацию с блокчейна, то есть readonly операция. Каждая операция имеет некоторую стоимость, которая измеряется в «Gas» [1, 2]. Также есть «payable» операции, которые запрашивают некоторую сумму токена, но это больше не как вид функций, а дополнение к ним.

Замечание. *Gas: сборы на исполнение транзакции не рассчитываются в токенах NEAR, вместо это она рассчитываются через Gas. Преимущество в том, что данные единицы - детерминированы, то есть одна и та же транзакция будет всегда будет стоить одинаковое количество Gas. Стоимость Gas пересчитывается в зависимости от загруженности сети в блокчейне [2].*

Для того, чтобы уметь работать с NFT, нужно написать соответствующий Smart-контракт, он базируется на описанном стандарте в спецификации Near Protocol [2, 3].

Чтобы реализовать данный проект были поставлены следующие задачи:

— Ознакомиться и понять NEAR Protocol, в частности выучить язык Rust для написания smart-контрактов. Реализовать некоторые несложные примеры smart-контрактов. Выучить nodejs/typescript для того, чтобы реализовать взаимодействия пользователя со smart-контрактами.

— Разобраться с Discord API, в частности с библиотеками discord.js/discord.ts;

— Написать код smart-контракта на языке Rust, который будет описывать логику NFT;

— Написать код Discord бота, который будет предоставлять удобный интерфейс взаимодействия пользователю;

— На основе признаков обучить и внедрить модель, которая будет рекомендовать пользователю купить NFT. Данной задачей по большей мере будет заниматься другой участник группы;

— По возможности обучить gan, чтобы пользователь мог создавать NFT. Аналогично, по большей мере должен реализовать иной человек с команды;

— Проанализировать и представить полученные результаты.

Определение. *GAN(Generative adversarial network) - алгоритм машинного обучения без учителя, которая позволяет генерировать фотографии. Позднее были изучение и иные генеративные модели, которые умеют генерировать не только фотографии, но и например музыку.*

[5] https://github.com/near/near-sdk-rs
[6] https://github.com/near/near-sdk-as

Актуальность и значимость

Актуальность блокчейна на сегодняшний день довольно легко понять, ведь хоть кто-нибудь, кто даже не является программистом или человеком, которые не проводит много времени в Интернете, можно сослаться на опрос проведенный РИА[7] в 2018 году и на тот момент 44% россиян слышали, что-то о криптовалюте. Или стоит хотя-бы вспомнить недавнюю новость о том, что ЦБ РФ хотел ввести некоторые ограничения на использование криптовалютой[8]. Все это наталкивает на мысль, что в наше время блокчейн обсуждается среди всех людей, независимо от их профессий и предпочтений.

В книге «Blockchain in Action» [1] приводится множество примеров применение технологий блокчейна. Стоит процитировать пример применения технологий блокчейна про актуальную на сегодняшний день проблему про COVID-19: «Although blockchain is well suited to solving many problems in this type of situation, I feel that it is ideally suited to performing a crucial task in mitigating the spread of this virulent disease: that of contact tracing. According to the U.S. Centers for Disease Control (CDC), contact tracing identifies cases by testing and tracing the source and pathway to the affected patient. This task of contact tracing is similar to tracking a fraction of a Bitcoin cryptocurrency to its origin. This trace for a cryptocurrency is recorded automatically on the DLT of the blockchain. Thus, blockchain infrastructure and DLT, along with the smart contract code collectively, could provide an innovative solution for contact tracing in an epidemic.».

Определение. *Distributed ledger technology(DLT, ledger) - список записей в блокчейне, содержащий произведенные транзакции. [1, 4]*

Для того, чтобы удостовериться, что на сегодняшний все еще продолжают пользоваться блокчейнами, достаточно взглянуть на количество транзакций проводимы в данный момент. Можно заметить на Рисунок 0.1, что в NEAR protocol эта цифра, в среднем, достигает около 800 тысячи в день.

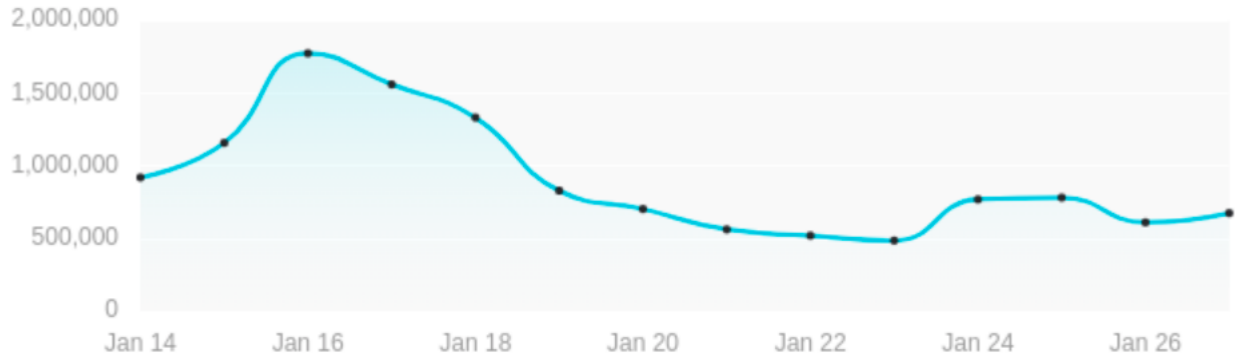

Рисунок 0.1. Количество транзакций обработанных в NEAR за последние 14 дней. Дата: 28.01.2022. https://explorer.near.org

[7] https://ria.ru/20180807/1526054613.html
[8] https://www.rbc.ru/finances/20/01/2022/61e9231a9a79477514c2b9ce

Самая важная концепция NFT - это то, что он дает право доказать, что данный конкретный файл, интеллектуальная собственность, часть данных принадлежит конкретному одному человеку. Если в современном мире права на картину, музыку либо любой другой медиа объект хранятся на каком-нибудь сервере и подкрепляются бумажной подписью на каком-нибудь документе и не без помощи юристов, то NFT позволяет упростить весь этот процесс ненужной бумажной волокиты. Немногие готовы перенести права картины на NFT, но на сегодняшний день есть и примеры, например, блокчейн-компания «Injective Protocol» купила картину «Morons» художника Бэнкси и сожгла ее в прямом эфире, чтобы в последствии перевести данную картину в NFT[9].

Наш маркетплейс строиться на довольно новом блокчейне и будет использоваться непременно как бот, чего я еще не видел. Наверное, такого нет в качестве бота, из-за того, что многие думают, что невозможно предоставить хороший интерфейс для этого, но это совсем неверное представление, ведь Discord API предоставляет невероятный функционал, чтобы пользователю было удобно. Также я не еще не видел такого маркетплейса, где есть хоть какая-нибудь технология связанная с машинным обучением: рекомендация цены за NFT, рекомендации самих NFT, GAN NFT картинок.

Существующие работы и решения

На сегодняшний день NFT маркетплейсов огромное количество, на блокчейне Ethereum, как примеры: opensea[10], rarible[11], на Solana: solanart[12]. На Near protocol, на данный момент самыми популярными являются: Paras[13], Mintbase[14].

Из двух представленных маркетплейсов, наиболее популярным является Paras. Он представляет базовый функционал маркетплейса: аутентификация в кошелек; покупка, продажа, просмотр NFT. У Paras также есть свой токен - Paras. В Paras представлены следующие медиа файлы: картины, фотографии, пиксель-арты.

Хоть и Mintbase менее популярен, чем Paras, но представляет гораздо больше видов медиа объектов: 3D модели, gif-изображения, аудио объекты. Но сама подача(оформление сайта) выглядит гораздо хуже, чем в Paras.

Объединяет данные маркетплейсы то, что у них единый стандарт реализации NFT, предоставленный Near protocol [3]. Уже как 3 года есть стандарт ERC-721[15], который был утвержден сообществом Ethereum. Стандарт в Near немного расширяет возможности, из-за уникальных возможностей реализованных в Near protocol [3]. Наша команда планирует придерживаться данного стандарта и реализовывать его, возможно потребуется некоторое его расширение, но об этом пока неизвестно. Важным вопросом является хранение в блокчейне самого NFT.

[9] https://rg.ru/2021/03/04/zachem-sozhgli-kartinu-benksi-za-95-tysiach-dollarov.html
[10] https://opensea.io/
[11] https://rarible.com/
[12] https://solanart.io/
[13] https://paras.id/
[14] https://www.mintbase.io/
[15] https://ru.bitcoinwiki.org/wiki/ERC-721

В данном стандарте количество, балансы ограничены 128-битным беззнаковым целочисленным типом. Вся сериализация, десериализация происходит с помощью JSON. Сам контракт отслеживает изменения в хранилище.

NFT хранит следующую информацию о себе: сам идентификатор токена, ID владельца и метаданные этого токена. Метаданные делятся на типа: первый класс, который хранит версию, название токена, ссылка на информацию, которая представляется в виде JSON, ссылка на sha256 хеш. Другой класс хранит: название, описание, время создания, время истечение представленное в UNIX epoch.

В Smart-контракте NFT должны быть реализованы следующие функции:

— nft_transfer - «change operation> функция передачи NFT от одного аккаунта другому.;

— nft_transfer_call - такая же «change operation» функция, что и предыдущая, за исключением того, что вызывается nft_on_transfer при удачной передачи NFT и nft_resolve_transfer при неудачной попытке;

— nft_token - «view operation» функция, которая по ID токена возвращает всю информацию о нем.

Предлагаемые подходы и методы

Для проведения транзакций(исполнения Smart-контрактов) NEAR protocol предоставляет near-cli[16] и near-api-js[17] [2]. Для наших целий требуется near-api-js, а следовательно вытекает потребность использовать язык javascript(nodejs, typescript) для написания «frontend» части бота.

Для реализации самого бота, как следует из названия проекта, используется Discord API. Но конечно, будет использоваться обертка над HTTP запросами. Существует множество всевозможных модулей для разнообразных языков. Для наших целей нам понадобится discord.js(javascript/nodejs)[18]. Также было принято решение, взять обвязку над этой библиотекой - discord.ts(typescript)[19]

Аккаунт

Аккаунты в NEAR protocol устроены так, что они имеют человеко-читаемый ID в отличие от большинства других блокчейнов, где обычно используется некоторый hash. Длина логина от 2 до 64 символов и содержит в конце суффикс обозначающий сеть блокчейна: main, test, beta. Аккаунт может создавать подаккаунты, которые по своему функционалу ничем не отличаются от обычного аккаунта. Но они нам понадобятся так как, на один аккаунт можно развернуть только один smart-контракт, а связанно это с тем, что при предложении подписания smart-аккаунта мы будем указывать название аккаунта, что сразу обозначает, который smart-контракт мы хотим инициировать для подписания. И самое главное, что нам нужно так это dev аккаунты. С помо-

[16] https://github.com/near/near-cli
[17] https://github.com/near/near-api-js
[18] https://discord.js.org/
[19] https://discord-ts.js.org/

щью этих тестовых аккаунтов мы и будем тестировать наши smart-контракты. Они создаются с помощью ранее упомянутого near-cli [2]. Для того, чтобы мы могли предлагать на подпись smart-контракты, нам нужно, чтобы этот пользователь авторизовался в NEAR wallet или иными словами подписать некоторую транзакцию на образование Access Key.

Access Key

Каждый аккаунт имеет создавать множество public/private ключей, которые в Near называются Access Key. Существует два типа Access Key: FullAccess и FunctionCall. Из названия первого можно предположить, что он дает полный доступ и он нам не годится, так как пользователи попросту не будут доверять нашему приложению. FunctionalCall ключ уникален и дает разрешения только на подписание функций контрактов, которые не являются «payable». Он имеет несколько атрибутов: количество Near, которые разрешается тратить на Gas, название контракта, чьи функции разрешается вызывать и названия этих функций, которые будут вызываться. Рисунок 0.2

Хранение в блокчейне

Важным вопросом, который не затрагивался при описании стандарта является хранение медиа объекта.В описанном стандарте хранится url на медиа объект, но никак не сам этот медиа объект, связанно это с тем, что хранение объекта невероятно дорогое. Авторы стандарта предлагают несколько подходов, один из них: пользователь изначально просто дает url и мы храним его, но в данном решении есть огромный минус - это то, что мы накладываем обязательство на пользователя об поддержании данного url валидным. Следующий подход заключается в том, что мы - сервис будем хранить данные в централизованном хранилище и тогда нашим опознавательным знаком будет какой-нибудь ключ, который будет соответствовать адресу NFT. И в этом подходе есть тоже минус в виде того, что непонятно какого размера должно быть это хранилище, какие ограничения вводить на размер медиа файла. И последнее предложение, так это хранить в специальном децентрализованном контенто-адресованном хранилище. Одним из примеров таких хранилищ является filecoin[20]. Минусом данного решения, наверное, является то, что мы начинаем использовать еще один блокчейн, который требует времени для понимания его спецификации. Наша команда приняла решение двигаться от самого простого подхода, то есть использование отданного пользователем url, к более сложному - использование filecoin.

Машинное обучение в маркетплейсе

За данную часть я отвечаю в меньшей степени. Но в рамках данного проекта нашей командой принято решение реализовать две идеи: рекомендательную систему или GAN. Рекомендательная система также делится на две части - это предсказание цены за конкретную NFT

[20] https://filecoin.io/

и более сложная идея - это рекомендовать сам NFT-токен. Идея с GAN выглядит гораздо легче, сравнивая с рекомендательной системой, например, создание аниме персонажа, пиксель арта, которое в последствии пользователь сможет использовать в своих коллекциях или для продажи.

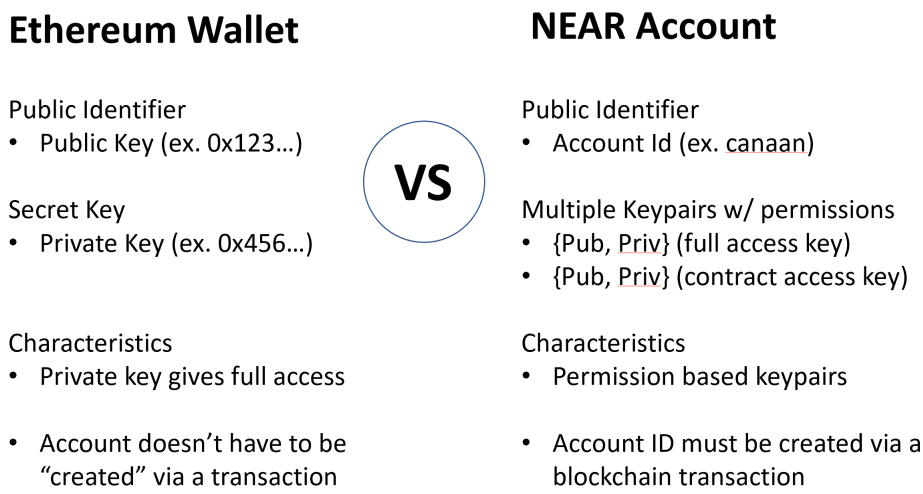

Рисунок 0.2. Сравнение спецификации аккаунтов и public/private ключей в блокчейнах Ethereum и Near. Источник: medium.com/@clinder

Ожидаемые результаты

Ожидаемым результатом является: NFT маркетплейс в Discord посредствам взаимодействия с ботом, которые имеет следующий функционал:

— Реализация меню как и по средствам команд в чате, так и интерактивно с помощью функционала представленным Discord API(slash commands, buttons, select menus);

— Взаимодействия с ботом:

  – Авторизация в NEAR wallet;

  – Информация о кошельке: текущее количество NEAR, список NFT;

  – Просмотри списка продаваемых NFT. Агрегация, фильтрация данного списка. Покупка продаваемой NFT;

  – Просмотр списка самых дорогих NFT проданных на площадке;

  – Продажа имеющихся NFT: моментальная продажа;

  – Реализация системы(модели), которая предлагает цену NFT;

  – GAN, который создает NFT;

  – Система конкурсов: возможность разыграть NFT в области канала Discord;

— Вся логика smart-контрактов NFT должна удовлетворять стандарту;

— Максимально большое покрытие тестами всего кода;

План работ

№	Дата	Содержание этапа работы
1	10.01.2022	Изучение NEAR Protocol и языка Rust для написания smart-контрактов, практика в написании некоторых простых примеров smart-контрактов, изучение NodeJS TypeScript для «frontend»
2	20.01.2022	Изучение стандарта NFT в Near Protocol
3	01.02.2022	Написание скелета бота на базе Discord API, используя NodeJS/Typescript.
4	10.02.2022	Добавление модуля авторизации через NEAR кошелек.
5	25.03.2022	Реализация NFT smart-контракта вместе с тестами.
6	05.04.2022	Добавление функций покупки/продажи NFT в Discord боте.
7	15.04.2022	Сбор признаков для рекомендательной системы
8	30.04.2022	Обучение и внедрение моделей рекомендательной системы или GAN в Discord бота.
9	-	Предъявление итоговой версии командного проекта.

Список источников

[1] Bina Ramamurthy. *Blockchain in Action*. S.l: Manning Publications, 2020. ISBN: 9781617296338.

[2] NEAR Protocol. *NEAR DOCS*. URL: https://docs.near.org/.

[3] NEAR Protocol. *NEAR Protocol Specification*. https://nomicon.io/ and https://nomicon.io/README.html.

[4] Solana Foundation. *SOLANA DOCUMENTATION*. URL: https://docs.solana.com/.