Міністерство освіти і науки України
Національний університет «Києво-Могилянська Академія»
Кафедра мережних технологій

КУРСОВА РОБОТА

за спеціальністю «Інженерія програмного забезпечення» 121

на тему:

Розробка смарт-контракту в мережі Ethereum для продажу елементів NFT-колекцій

Науковий керівник:
доцент Франчук О. В.

Виконав:
студент 1-го курсу Скрипнік А. О.

Київ – 2022

Міністерство освіти і науки України

Національний університет «Києво-Могилянська Академія»

Кафедра мережних технологій

ЗАТВЕРДЖУЮ

Зав. кафедри мережних технологій,

професор, д.ф.-м.н.

_____Г. І. Малашонок

“____”_____ 2022 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу

студенту Скрипніку Андрію 1-го курсу факультету інформатики

ТЕМА: Розробка смарт-контракту в мережі Ethereum для продажу елементів NFT-колекцій

Зміст ТЧ до курсової роботи:

Анотація

Вступ

1. Огляд технології блокчейн
2. Смарт-контракти в мережі Ethereum
3. NFT та імплементація смарт-контракту для продажу NFT-колекцій

Висновки

Список джерел

Додатки (за необхідністю)

Дата видачі “___” _____ 2022 р.

Керівник _____ Завдання отримано _____

Календарний план виконання курсової роботи

Тема: Розробка смарт-контракту в мережі Ethereum для продажу елементів NFT-колекцій

№ п/п	Назва етапу курсового проекту (роботи)	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу	жовтень 2021 р.	
2.	Огляд документації та літератури за темою роботи	жовтень - листопад 2021 р.	
3.	Оволодіння навичками розробки смарт-контрактів на мові програмування Solidity	листопад - січень 2022 р.	
4.	Реалізація практичної частини роботи	січень - березень 2022 р.	
6.	Написання теоретичної частини курсової роботи	лютий - березень 2022 р.	
7.	Надання роботи керівнику для перевірки, демонстрація практики	березень 2022 р.	
8.	Корегування роботи за результатами перевірки керівником	квітень 2022 р.	
9.	Остаточне оформлення теоретичної частини та слайдів доповіді	травень 2022 р.	
10.	Захист курсової роботи	червень 2022 р.	

Студент Скрипнік А.О.

Керівник Франчук О.В.

“_____” _____ _____ р.

Зміст

Анотація

Роботу присвячено теоретичному огляду технології блокчейн та деяких її витоків, таких як смарт-контракти, NFT, а також, практичній імплементації смарт-контракту для продажу елементів NFT-колекцій.

Практичним результатом роботи є розроблений в тестовому блокчейні Ethereum смарт-контракт за стандартом ERC-721, написаний з використанням мови програмування Solidity.

Ключові слова: блокчейн, Bitcoin, Ethereum, Proof-of-Work, смарт-контракт, Solidity, ERC-721, Remix IDE, NFT, DeFi, ICO, DApp, DEx, DSN, Metamask, Rinkeby, OpenSea, Open Zeppelin.

Вступ

9-го листопада 2021 року Bitcoin досягнув максимальної ціни за свою історію – приблизно 69000$ за монету. Криптовалюти вже давно стали частиною нашого світу. Ідея децентралізованої фінансової системи, що зародилася одразу після кризи 2008, знайшла свою імплементацію у вигляді різноманітних проектів на базі технології блокчейн. І хоча Bitcoin не досягнув першочергової мети створення кардинально нової, незалежної від регуляторів фінансової системи, його успіх породив тисячі інших криптовалютних проектів, які рухають світ до поступової цифровізації економіки і всього, що нас оточує.

Вслід за Bitcoin з'явився Ethereum, який розширив сферу використання технології блокчейн та реалізував ідею смарт-контракту – комп'ютерного алгоритму, що призначений для формування, управління та передачі інформації про володіння чим-небудь. На базі цього виросла ціла екосистема фінансових сервісів, які називаються DeFi (Decentralized Finance). Головною цінністю DeFi є те, що учасники взаємодіють між собою напряму без посередників – банків, кредитних організацій. Транзакції можуть проводитись швидше та дешевше.

На хвилі розвитку Ethereum з'явилися технологія NFT (non-fungible token), яка у 2021 стали надзвичайно популярним видом цифрового мистецтва, що перегорнула індустрію. Професіональні митці, розробники та просто авантюристи створювали картини, музику, ігрові елементи у вигляді NFT, щоб бути першопроходьцями в цій вітці цифрової революції.

Я обрав тематику дослідження протоколу смарт-контрактів, адже ця технологія має широкі перспективи бути інтегрованою в наше повсякденне життя. До того ж, ідеологія децентралізованої фінансової системи близька

до мого світогляду та в близькому майбутньому може кардинально змінити наш звичний життєвий устрій.

Моєю метою є огляд технології блокчейну, а також її характеристик, що стали основою для створення смарт-контрактів. Крім цього, метою є дослідження протоколу смарт-контрактів, його реалізація в мережі Ethereum, та прикладів використання у вигляді NFT.

Для того, щоб досягнути поставленої мети, було опрацьовано десятки матеріалів, статей та відео-лекцій, що вилилось у створені власного смарт-контракту для продажу елементів NFT-колекцій, написаному на мові програмування Solidity, в тестовій мережі Ethereum – Rinkeby.

Робота складається з трьох розділів.

У першому розділі розглядається технологія блокчейн в цілому, принципи її роботи, сфери її застосування та зроблений висновок, чому вона набула популярності та широкого використання.

Другий розділ присвячено огляду смарт-контрактів та особливостей платформи Ethereum, на базі якої в основному створюються смарт-контракти.

У третьому розділі описано феномен цифрового мистецтва у вигляді NFT, розглянута історія цієї технології та можливості взаємодії звичайної людини з NFT-проектами. Крім цього, розглянуто особливості мови програмування Solidity, стандарт токенів ERC-721, реалізовано та протестовано основні функції написаного смарт-контракту.

1 Огляд технології блокчейн

1.1 Погляд в історію

Технологія була вперше описана у 1991 році американськими науковцями-дослідниками Стюартом Хабером та Скоттом Шторнеттою. Вони хотіли створити обчислювально-практичне рішення для відмітки цифрових документів таким чином, щоб вони не могли бути перевидані заднім числом чи підроблені. Вони розробили систему, що використовувала концепцію криптографічно безпечних ланцюгів блоків, в яких зберігались цифрові документи.

Основою запропонованої системи у 1992 були закладені дерева Меркла – структура даних, що містить в собі інформацію про якийсь більший обсяг даних. Кожен блок даних містить інформацію про попередній блок. Таким чином найновіший блок містить історію про увесь ланцюг. На жаль, тоді ця революційна технологія не знайшла свого користувача, й фактично була не задіяна.

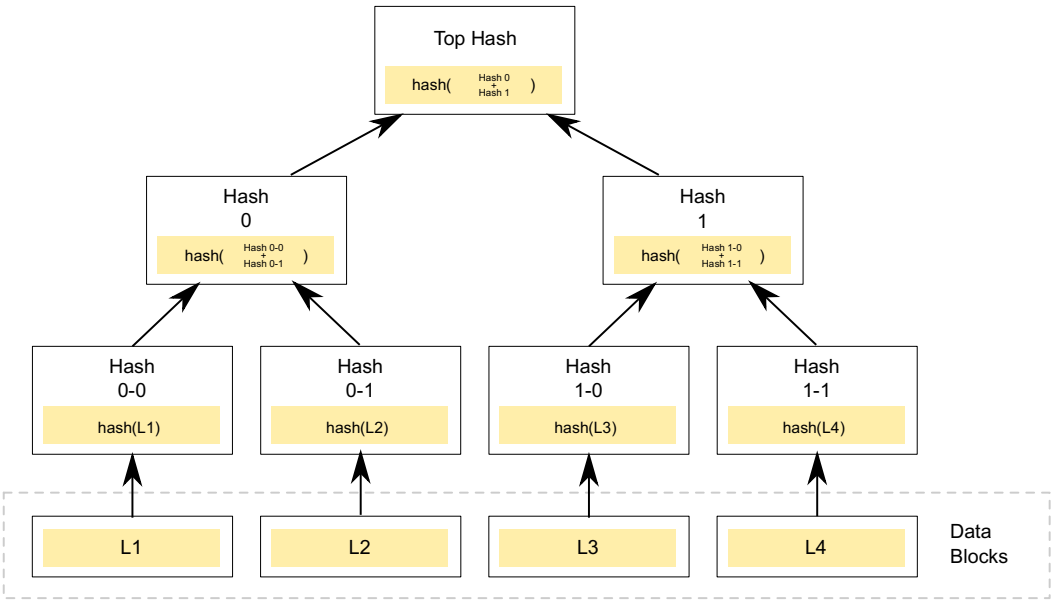


Рисунок 1.1.1 – дерево Меркла
(https://en.bitcoinwiki.org/wiki/Merkle_tree#/media/File:Hash_Tree.svg)

У 2004 році криптографічний активіст Гел Фінні, який у майбутньому стане учасником першої Bitcoin-транзакції, презентував систему RPoW, Reusable Proof of Work, – прототип цифрової монети. Хоча Гел не задумував робити щось більше, ніж звичайний прототип, його ідея зробила вагомий внесок у розвиток крипто-індустрії в майбутньому. Система працювала, отримуючи незамінний токен, заснованому на proof-of-work (принцип захисту мережевих систем від зловживань, наприклад, DoS-атак) та підписаному за допомогою RSA-алгоритму, який міг бути переданий від користувача до користувача. Інформація про приналежність токенів зберігалась на спеціальному сервері, який був запущений на IBM 4578 безпечному криптографічному ко-процесорі.

У кінці 2008 року людина або група людей під псевдонімом «Сатоші Накамото» представив ідею розподілених блокчейнів в електронному листі, що він надіслав вузькому колу людей, що займалися криптографією. Втіленням ідеї була децентралізована p2p система електронних платежів під назвою Bitcoin. Якщо коротко описати принцип роботи, то транзакції в мережі Bitcoin записуються в блоки блокчейну, який обслуговується децентралізованими вузлами мережі. Так звані майнери змагаються за те, щоб створити наступний блок в ланцюгу, вирішуючи певну задачу, за що отримують винагороду – певну кількість монет Bitcoin.

3 січня 2009 року перший Bitcoin блок був створений самим Сатоші Накамото, за що він отримав 50 монет у винагороду.

Перша в історії Bitcoin-транзакція відбулася 12 січня 2009 року, коли Сатоші переказав 10 монет на гаманець вищезгаданого Гела Фінні.

У 2013 році програміст та один із засновників журналу “Bitcoin” Віталік Бутерін (громадянин Канади) заявив, що Bitcoin-у потрібна скриптова мова для створення децентралізованих застосунків. Не отримавши підтримки у спільноті, він перейшов до розробки нової

розподіленої обчислювальної платформи на базі блокчейн Ethereum, який породив таке явище, як смарт-контракти.

В чому ж криється важливість технології блокчейн? Розглянемо приклад тривіальної фінансової операції, як купівля нерухомості. Право власності переходить до покупця після передачі коштів. При цьому, ні покупцю, ні продавцю не можна повністю довіряти: покупець може стверджувати, що він вже відправив кошти, хоча не робив цього, а продавець може стверджувати, що він не отримав кошти, хоча це не так. Щоб уникнути такого роду проблем, має існувати третя сторона, яка контролюватиме та підтверджуватиме транзакції. Однак, присутність цієї сторони тільки ускладнює угоду. Більше того, створюється єдина точка відмови. Якщо є порушення в центральній базі даних, то постраждати можуть обидві сторони.

Блокчейн вирішує ці проблеми шляхом створення децентралізованої та захищеної від ручного несанкціонованого доступу системи для реєстрації транзакцій. У нашому прикладі блокчейн слугує єдиним реєстром для покупця та продавця. Якщо транзакція підтверджена обома сторонами, вона відображається в реєстрі. Якщо хтось захоче внести некоректні зміни в реєстр, це має бути затверджено більше, ніж половиною учасників мережі, що нереально в умовах зрілої та широковикористованої мережі, як наприклад, Bitcoin.

1.2 Принцип роботи блокчейн на прикладі Bitcoin

Розподілений реєстр – загальна база даних мережі, в котрій зберігаються копії транзакцій. Існують строгі правила відносно того, хто і як може вносити зміни в базу. Наприклад, не можна видаляти записи після їх реєстрації.

Кожен новий блок є елементом ланцюга – блокчейну. Блок вміщає ряд записів про виконані операції в мережі, які є новими з точки зору попередніх блоків у ланцюгу. Коли блок додається в ланцюг, він в собі містить інформацію у тому числі і про попередній стан ланцюга. Зміни структури блоку неможливі. Ланцюг блоків і є розподіленим реєстром – базою даних, куди записуються всі операції.

Як користувач може бути впевненим, що система не підроблена та містить в собі тільки реальні транзакції? Розглянемо структуру блоку:

- Магічне число – завжди містить значення 0xD9B4BEF9 – ідентифікує, що формат файлу відповідає структурі, що використовується в мережі Bitcoin.
- Розмір блоку – кожен блок лімітований у розмірі 1 МБ.
- Заголовок блоку – складна структура, що містить в собі: номер версії протоколу, хеш заголовка попереднього блоку, хеш усіх транзакцій в поточному блоці, час створення, складність блоку, та “nonce” – 32-бітне число, що має змінити майнер для того, щоб правильно вирішити задачу для поточного блоку.
- Лічильник транзакцій – кількість транзакцій в блоці.
- Список усі транзакцій – зазвичай в блоці достатньо транзакцій, щоб досягнути до ліміту розміру блоку в 1 МБ.

Запис нових блоків у ланцюг виконується майнерами, тому їхня роль в перевірці цілісності інформації є ключовою. Вони змагаються за рішення певної обчислювальної задачі (алгоритм Proof-of-Work). Майнер має взяти хеш попереднього блоку, “nonce” попереднього блоку та суми хешів транзакцій за попередні 10 хвилин, та засновуючись на цьому, обчислити новий хеш, який би відповідав певним параметрам системи.

Якщо майнер знайшов “nonce”, він його транслює іншим учасникам мережі, щоб вони могли перевірити цей розв’язок. Якщо більшість майнерів, принаймні 51%, досягають консенсусу по цьому розв’язку,

майнер має право додати новий блок в ланцюг та отримати відповідну винагороду. Новий блок в мережі Bitcoin створюється приблизно кожні 10 хвилин.

Алгоритм Proof-of-Work представляє собою значний дизбаланс між часом, витраченим на пошук рішення, та часом перевірки цього рішення. Тож процес перевірки правильності хешу не є ресурсоємким. Знайдений хеш є певною «печаткою», що підтверджує достовірність усього попереднього ланцюга.

Якщо зловмисник захоче змінити дані одного з блоків, хеш зміниться, і учасники мережі не приймуть цю зміну. На зорі розвитку Bitcoin, коли учасників мережі було досить мало, існувала ймовірність, що хтось внесе невалідні зміни в ланцюг, адже тоді в теорії можна було відносно легко здобути 51% всіх обчислювальних машин в мережі. Проте цього не сталося, а зараз, коли десятки тисяч машин по всьому світі здобувають Bitcoin, це виглядає неможливим.

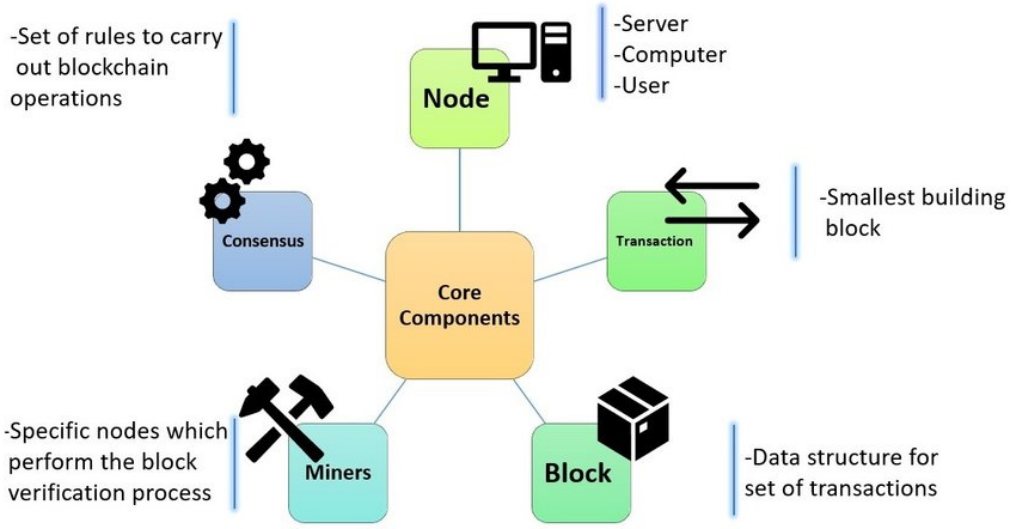


Рисунок 1.2.1 – компоненти блокчейну

(https://www.researchgate.net/figure/Core-components-of-the-blockchain-system_fig3_339046744)

1.3 Сфери використання технології блокчейн

Інновації, запропоновані блокчейном, знаходять своє застосування у найрізноманітніших галузях.

Передбачувано, що фінансовий сектор наразі інтегрує блокчейн у свої бізнес-процеси, наприклад, для управління онлайн-платіжками, ринковою торгівлею. Інвестиційна холдингова компанія “Singapore Exchange Limited” використовує цю технологію для проведення більш ефективного міжбанківського розрахунку. Вони розв’язали проблему мануальної звірки тисяч фінансових транзакцій.

Роздрібний гігант Amazon запатентував систему розподіленого реєстру відслідковування переміщення товарів між постачальниками та покупцями, що допомагатиме перевіряти справжність товарів, що продаються на платформі.

Енергетичні компанії створюють однорангові платформи для торгівлі енергоносіями між приватними особами. Власники будинків із сонячними батареями можуть використовувати цю систему для продажів надлишків електроенергії своїм сусідам. Спеціальні «розумні» лічильники створюють транзакції, а блокчейн їх реєструє.

Компанії із області мультимедіа та розваг також опановують блокчейн, наприклад, для управління даними про авторські права. “Sony Music Entertainment Japan” таким чином підвищує ефективність технічних засобів захисту авторських прав. Це відіграє ключову роль при визначенні справедливої винагороди для творців контенту.

Блокчейн дозволив створити прорив у сфері мистецтва – NFT. Витвори цифрового мистецтва продаються на аукціонах за мільйони доларів. Ця нова сфера дозволила художникам, музикантам та інфлуенсерам отримувати додаткові прибутки за свої роботи.

І це лише мала частина галузей, де може бути використаний блокчейн.

2 Смарт-контракти в мережі Ethereum

2.1 Огляд технології смарт-контракту

Смарт-контракт – цифровий аналог звичайних договорів, спеціальна програма, що виконує певні дії при виконанні сторонами угоди певних умов. Базовий приклад: відправити гроші продавцю при поставці товару. Смарт-контракт дозволяє безпечно обмінюватись криптовалютою, грошима, NFT, цінними паперами між учасниками угоду напряму, без посередників.

В сучасних реаліях, коли сторони можуть укладати угоди, знаходячись у різних точках планети, використовуючи електронну пошту чи месенджери. Плинність світу, технологічний бум, всесвітня пандемія, цифровізація всіх сфер життя також стали вдалим підґрунтям для закріплення технології смарт-контрактів.

Концепція представив криптограф Нік Сабо ще у далекому 1994 році, коли він дійшов до висновку, що за допомогою цифрового децентралізованого реєстру можна укладати цифровий контракт та автоматично запускати певний код.

Тоді цю ідею не вдалось реалізувати, але у 2008 році вона отримала нове дихання завдяки імплементації технології блокчейн в криптовалюті Bitcoin. Концепція блокчейну дозволяла зробити смарт-контракти максимально інформативними та захищеними від шахрайства. Наприклад, можна було б подивитися історію даних про те, хто був власником певної нерухомості, а також бути впевненим, що ці дані правдиві, адже вони зберігаються децентралізовано та змінюються за згодою більшості учасників мережі.

Реалізувати смарт-контракти в Bitcoin не вийшло, адже система не передбачала цієї можливості. Але у 2013 році була створена універсальна децентралізована блокчейн-платформа Ethereum із можливістю

програмувати смарт-контракти. Можливо, саме ця особливість дозволила Ethereum стати другою основною криптовалютою після Bitcoin.

По своїй натурі технологія смарт-контрактів дуже гнучкий інструмент. Розробники можуть створювати програми для різних сценаріїв. Декілька прикладів:

- ICO (initial coin offering) – процедура отримання певної кількості токенів за інвестиції в реальному грошовому еквіваленті. Часто використовується на певних етапах розвитку крипто-проектів, коли вони залучають інвестиції для майбутньої реалізації свого алгоритму/протоколу/ідеї.
- DApp – децентралізована ігри чи застосунки. Часто застосовується в сферах аукціонів, букмекерських послуг, азартних та відео іграх.
- DEx – децентралізовані обмінники. У цьому випадку можна позбутись третьої особи, наприклад, крипто-біржі, під час покупки чи продажу криптовалюти.
- DSN – децентралізована соціальна мережа. Ця ідея ще не є реалізованою, проте вона полягає в тому, що користувач може сам повністю володіти своєю сторінкою в мережі та напряму отримувати кошти за розміщену на ній рекламу.

Смарт-контракти можуть стати альтернативою класичним паперовим контрактам, однак не зараз. Є певний ряд перепон, які мають бути здолані перед тим, як ця технологія, дійсно, увійде в широкий ужиток.

Перш за все, немає законодавчої бази для використання смарт-контрактів. Влада різних країн не сприяє популяризації цієї технології через те, що це кардинально новий інструмент, який докорінно змінює юриспруденцію.

По-друге, далеко не всі люди можуть читати програмний код і розуміти, що відбувається. Натомість, всі можуть прочитати, що написано в класичному договорі. Головним драйвером ідеї смарт-контрактів є

позбавлення третьої особи. Але враховуючи складність написаного коду, скоріш за все прийдеться залучати консультанта, що розуміється на коді, щоб перевірити, чи є умови смарт-контракту дійсно такими, як того очікують сторони.

По-третє, розробка та розгортка смарт-контракту теж потребує спеціаліста зі знаннями. Більше того, після моменту, коли смарт-контракт розгорнули в мережі Ethereum, його вже не можна змінити. Тобто він є іммутабельним. В той час як звичайні паперові договори можуть зазнавати змін.

По-четверте, смарт-контракт працює в блокчейні й здійснює певні транзакції в межах блокчейну. Тому він не має безпосереднього впливу на «реальний світ» за межами блокчейну, тому мають бути розроблені додаткові механізми для регуляції зв’язків між реальними та віртуальними активами.

Як бачимо, технологія вже набула широкого використання в цифровому світі, проте є ще досить сирою для того, щоб казати про її перспективи замінити паперові угоди в найближчі роки. Головною перепоною є, безперечно, державні органи, які завжди з насторогою ставляться до проривних технологій, що можуть змусити докорінно змінити життєвий устрій.

2.2 Принцип роботи смарт-контрактів в мережі Ethereum

Ethereum запропонував мережу, яка дозволяє створювати та розгортати децентралізовані застосунки – DApp. Наприклад, децентралізовані фінанси DeFi – одна із найпопулярніших галузей, що зараз розробляється в мережі. Вона являє собою традиційні фінансові послуги без посередників. Це однорангова система, в якій взаємодія відбувається безпосередньо між користувачами. До прикладу, можливість брати позику

напряму без банку, якому треба платити комісію. Замість цього відсотки по позиці йтимуть напряму до кредитора.

Крім вищезазначеної глобальної цілі Ethereum, варто зазначити, що нові блоки в його блокчейні додаються в середньому кожні 12 секунд, в той час Як у Bitcoin – кожні 10 хвилин.

Стан в Ethereum зберігають так звані акаунти. Акаунт має 20-ти байтову адресу. Зміна стану відбувається за допомогою передачі так званих повідомлень.

Акаунт зберігає наступні поля:

- nonce – лічильник, який забезпечує обробку транзакції лише один раз і не більше.
- storageRoot – масив даних.
- balance – кількість монет, якими володіє даний акунт.
- codeHash – хеш, який посилається на код смарт-контракту в EVM (Ethereum virtual machine), якщо це акаунт типу “контракт”.

Є два типи акаунтів. “Контракт” – смарт-контракт, розгорнутий в мережі, контролюється кодом. “Зовнішня власність” – акаунт, що є власністю будь-кого, хто має приватний ключ.

Обидва типи аканту мають можливість: отримувати, зберігати та відправляти ETH та інші токени; взаємодіяти з розгорнутими смарт-контрактами.

Натомість, є ряд відмінностей між двома типами акаунтів: акаунт типу “зовнішня власність” є безкоштовним при створенні, може ініціювати транзакції. Акаунт типу “контракт” створюється за певну комісію, що називається “газ”, адже він займає певний шмат пам’яті в мережі. Також акаунт цього типу може ініціювати транзакції лише у відповідь на іншу вхідну транзакцію. Навіть більше, транзакції, ініційовані акаунтом типу “контракт” можуть створювати нові смарт-контракти, а не лише відправляти токени.

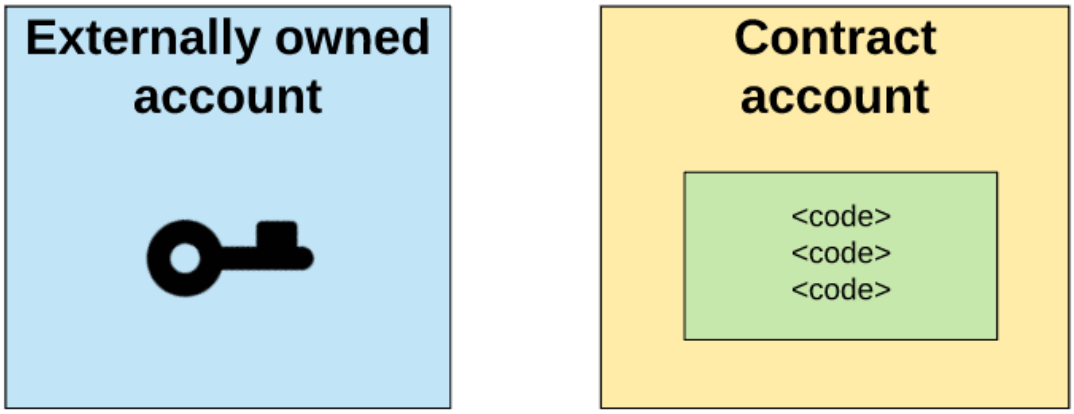


Рисунок 2.2.1 – види акаунтів в Ethereum

(https://habr.com/en/post/407583/)

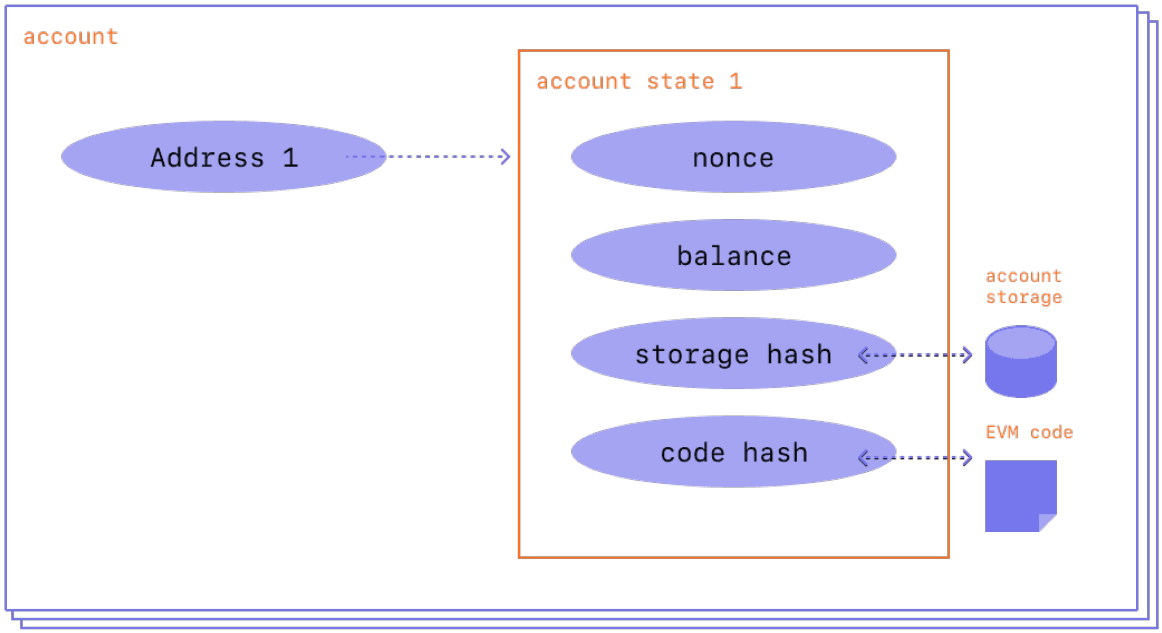


Рисунок 2.2.2 – поля акаунтів в Ethereum

(https://ethereum.org/en/developers/docs/accounts/)

Повідомлення в Ethereum – аналог транзакцій в Bitcoin. Але вони мають певні ключові відмінності. Перш за все, повідомлення можуть бути створені обома типами акаунтів, в той час як в Bitcoin транзакція може бути створена тільки користувачем (“зовнішня власність”). По-друге, адресата повідомлення може повернути відповідь, якщо це акаунт типу “контракт”.

Кожен вузол мережі може розповсюдити запит на створення повідомлення, яке має змінити стан EVM. Коли цей запит з'являється, майнер має виконати транзакцію та розповсюдити зміну стану до інших вузлів мережі. Транзакції виконуються за певну плату – "газ".

Повідомлення має наступну інформацію:

- Recipient – 20-байтова адреса. Якщо ця адреса належить акаунту типу "зовнішня власність", то транзакція перераховуватиме певну кількість токенів. Якщо адреса належить акаунту типу "контракт", то транзакція запустить виконання коду контракту.
- Nonce – порядковий номер транзакції, відправлених з даної адреси. Є запобіжником від виконання однієї й тієї ж транзакції двічі.
- Signature – ідентифікатор відправника. Підпис створюється, коли відправник підписує транзакцію приватним ключем. Це підтверджує, що повідомлення відправив дійсно цей користувач системи.
- Value – кількість ETH для переказу від відправника до отримувача. Вводить в WEI. 1 ETH = 10^{18} WEI.
- Data – опціональне поле для довільних даних.
  Max priority fee per gas – максимальна плата майнеру за виконання транзакції.
- Max fee per gas – максимальна плата за виконання транзакції. Включає в собі плату майнеру та плату мережі за ініціювання транзакції.

Коли відправник ініціює транзакцію, підписує її своїм приватним ключем, вона потрапляє в мережу, та має бути спочатку провалідована вузлами мережі перед фактичним виконанням:

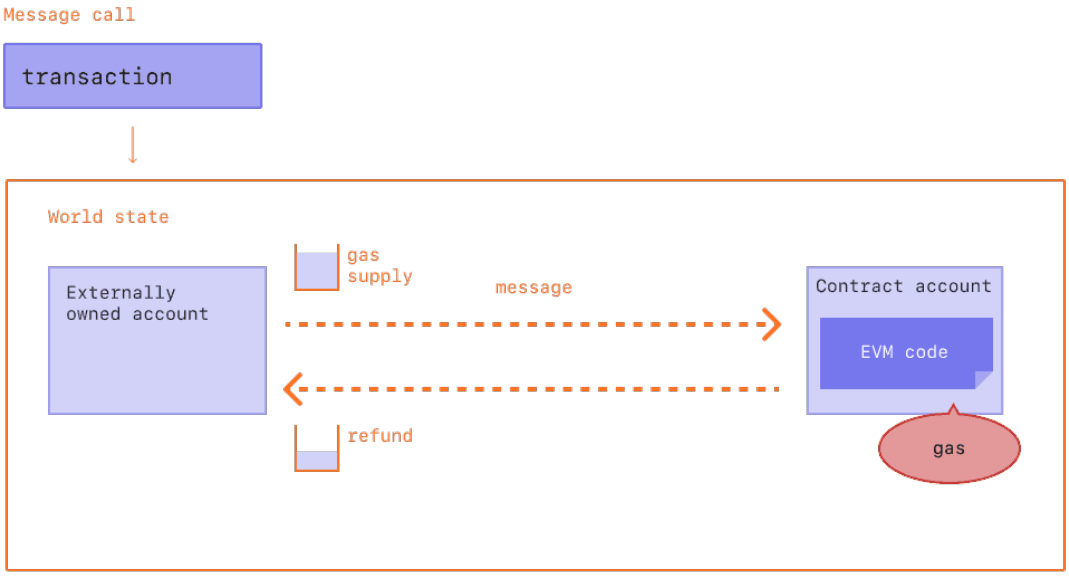
- Перевіряється коректність Nonce (номер транзакції збігається із номером акаунту) та Signature.

- Перевіряється, чи достатньо “газу” на рахунку відправника, щоб оплатити виконання транзакції. Інакше повертається відповідна помилка.
- Ether передається отримувачу. Якщо акаунту отримувача не існує, він створюється. Якщо акаунт отримувача має тип “контракт”, виконується відповідний код контракту.

Код смарт-контракту виконується в Ethereum Virtual Machine (EVM). У машини є доступ до трьох структур даних: stack виконання, memory – динамічний масив, long-term contract storage – холодне сховище ключ-значення (доступне навіть після виконання контракту).

Смарт-контракти можуть бути написані з використанням високорівневих мов програмування Solidity чи Vyper.

До прикладу, Solidity – об’єктно-орієнтована мова, що має наступні характеристики: статична типізація, наслідування, модульність, типи, визначені користувачем. Solidity підтримує можливість написання бібліотек для перевикористання коду.



2.2.1 – транзакція в Ethereum

(https://ethereum.org/en/developers/docs/transactions/)

3. Мова Solidity та імплементація смарт-контракту для продажу елементів NFT-колекцій

3.1 Огляд NFT як виду цифрового мистецтва

Ринок NFT у 2021 році досяг позначки 17 мільярдів доларів, зробивши скачок на 21000% порівняно з 2020 роком. Найдорожчий у світі NFT “The Merge” був проданий за 91.8 мільйонів доларів в грудні 2021 року. Із появою NFT цифрові художники отримали можливість представляти свої витвори широкому загалу, заробляти на них великі кошти.

Технологія NFT (non-fungible token) була створена й розвинена на основі смарт-контрактів в мережі Ethereum у 2017 році.

Невзаємозамінні токени – повністю унікальні, вони надають володіння цифровим об’єктом, який існує в єдиному екземплярі. Це може бути будь-що: художня картина, музичний витвір, ігрові предмети, інші віртуальні активи.

При покупці NFT користувач не купує фактично сам цифровий актив, але купує запис в блокчейні про те, що він ним володіє. Сам актив нікуди не рухається, він зберігається в так званому IPFS – Inter-Planetary File System – одноранговій розподіленій файловій системі.

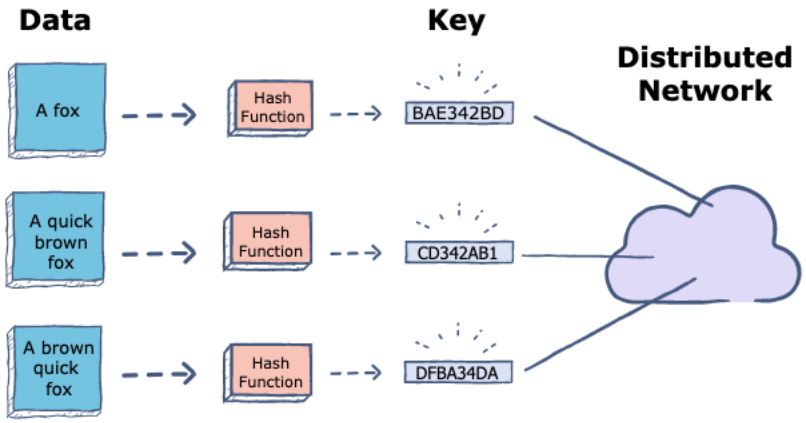


Рисунок 3.1.1 – розподілена хеш-таблиця (https://www.educative.io/edpresso/what-is-a-distributed-hash-table)

IPFS дозволяє користувачам отримувати та роздавати контент в схожій манері із BitTorrent. Кожен користувач може зробити запит на контент з будь-якого вузла мережі, який роздає даний контент, використовуючи розподілену хеш-таблицю.

По суті цифровий файл може подивитися чи навіть завантажити хто-завгодно, але запис в блокчейні підтверджує, що їм володіє саме той, хто здійснив транзакцію. Вимальовується аналогія з картинами: вона може належати музею, державі, приватній особі, але ви зможете на неї подивитися на виставці.

NFT вирішує проблему визначення, чи являється та чи інша цифрова робота оригінальною, і саме це зробило прорив в індустрії. Раніше цифрові активи не могли мати такої характеристики як «оригінальність».

Тим не менше NFT в Ethereum мають ряд проблем. Перш за все, пропускна спроможність мережі, яка на даний момент дорівнює близько 30 в секунду, та алгоритм майнингу Proof-of-Work, який є дуже ресурсоємким. Все це впливає на ціну транзакції. Наприклад, цін звичайного переказу певної кількості монет ETH на інший гаманець стандартно може дорівнювати близько 20 доларам. Цей недолік Ethereum дає поле для розвитку іншим платформам, як Telos чи Rose, які є більш ефективними. Більше того, Ethereum 2.0 перебуває в процесі розробки. Серед покращень буде перехід на алгоритм Proof-of-Stake, що має здешевити ціну транзакції та збільшити пропускну здатність мережі.

Найголовнішою проблемою є певне нерозуміння технології серед звичайних людей, що чують про неї вперше. Платити кошти, достатньо великі, за картинку, яку можна завантажити будь-де в мережі інтернет виглядає досить дивним з першого погляду. Тому наразі ринок скоріш за все не перебуває на піку, бо все-таки NFT все ще не стали загальновживаним явищем, а є нішовим інструментом для митців, зірок, артистів, благодійних організацій, що збирають пожертви, та інвесторів

3.2 Мова Solidity та стандарт ERC-721 для смарт-контрактів

Смарт-контракти в Ethereum розробляються на одній з мов, спеціально спроектованих для трансляції в байт-код віртуальної машини Ethereum – EVM. Мова Solidity є найпопулярнішою, вона схожа на C та JavaScript. Також серед варіантів є Vyper і Serpent (схожі на Python), Mutan (заснований на Go) та LLL.

Всі умови смарт-контракту мають мати програмний опис та ясну логіку виконання. Засновуючись на логіці, закладеній в код, смарт-контракт виконує ту чи іншу дію при досягненні певних умов або при виклику його функцій.

Контракт виконується в EVM, яка повністю ізольована – це значить, що код, який виконується в EVM, не має доступу до мережі та файловій системі, проте може звертатись до інших контрактів.

Кожен вузол блокчейну Ethereum не тільки зберігає дані та код, але і зберігає EVM для виконання коду контрактів. Це є ключовою відмінністю Ethereum від Bitcoin, який не має віртуальної машини.

Solidity – є статично-типізованою мовою, схожою на JavaScript. Його розробка була запропонована Гевіном Вудом у 2014 році. Розробка велась командою програмістів в рамках проекту Ethereum. Офіційно представлений в серпні 2015 року.

Використання синтаксису, схожого на ECMAScript було умисним для того, щоб залучити якомога більше програмістів та дещо понизити поріг входу. Solidity має наступні характеристики:

- Об’єктно-орієнтованість
- Підтримка наслідування, в тому числі множинного
- Підтримка бінарного інтерфейсу контракту (ABI) – надає можливість звертатись до функцій смарт-контракту, розгорнутого в блокчейні, використовуючи зрозумілий інтерфейс.

Solidity має розгорнуту документація та багато бібліотек, які значно спрощують написання коду. Серед найвідоміших: Open Zeppelin, Truffle. Бібліотеки допомагають створити свій токен, на основі готових шаблонів, що відповідають стандартам – ERC20 для взаємозамінних токенів та ERC721 для невзаємозамінних токенів (NFT).

ERC-721 токени мають наступні функції, які мають також ERC20:

- Name – використовується іншими контрактами та застосунками.
- Symbol – дозволяє DApps доступатися по короткому імені токена.
- Total supply – загальна кількість випущених токенів.
- Balance of – кількість ETH на рахунку заданого гаманця

Крім цього, ERC-721 визначають ряд функцій, що є індикатором власності токена:

- Owner of – вертає адресу власника токена.
- Transfer – передає власність токена іншому користувачу.
- Approve – надає дозвіл передавати токен третій особі від імені власника токена.
- Take ownership – коли юзеру дозволено розпоряджатись токеном (див. попередню функцію), він може передати собі цей токен.
- Token of owner by index – вертає токен, яким володіє користувач за індексом в масиві токенів цього контракту, якими володіє цей користувач.
- Token metadata – посилання на певні атрибути цього токену, наприклад, на його IPFS хеш.

Крім ряду функцій, стандарт ERC721 визначає два подій (повідомлень), які запускаються, коли контракт їх викликає. Зовнішні програми, які слухають блокчейн події можуть виконати певну логіку, коли подія запущена.

Перша подія – Transfer – запускається, коли токен змінює власника. Ця подія має деталі щодо того, хто відправив токен, хто отримав токен, та який токен був відправлений.

Друга подія – Approval – запускається, коли користувач дозволяє іншому користувачу взяти токен у володіння. Ця подія має деталі щодо того, який акаунт володіє токеном, якому акаунту дозволено володіти токеном в майбутньому, та який це токен (за його ідентифікатором).

Solidity надає ряд модифікаторів доступу за замовченням: public, private, protected, internal, external. Але є випадки, коли певні функції не мають бути використані ким-завгодно, але лише власником контракту. Наприклад, у випадку продажів NFT-колекцій можна створити функцію, яка запускатиме продажі в певний момент часу. Очевидно, доступ до неї має мати тільки власник колекції. Це відбувається за допомогою контракту Ownable із бібліотеки Open Zeppelin. Необхідно наслідувати контракт від Ownable, для того щоб з’явився модифікатор onlyOwner, за допомогою якого можна задовольнити дану потребу.

3.3 Імплементація смарт-контракту для продажу NFT-колекцій

NFT-колекція – це група NFT токенів, власність над якими породжується з одного смарт-контракту. Зазвичай митці розробляють NFT не по одній штуці, а одразу декілька тисяч, наприклад 10000. Кожен токен схожий на інших концептуально. Наприклад, візьмемо мавп із колекції “Bored Apes Yacht Club”. Ці мавпи відрізняються за наступними характеристиками: колір фону, одяг, очі, шкіра, рот, сережка та шляпа.

Рисунок 3.3.1 – токен 3947 колекції "Bored Ape Yacht Club" (https://opensea.io/assets/ethereum/0xbc4ca0eda7647a8ab7c2061c2e118a18a936f13d/3947)

Певні характеристики поширені серед багатьох токенів колекції, деякі є дуже рідкісними і присутні лише в одиницях. Чим більш рідкісні характеристики у конкретного NFT, тим більш дорогим він є зазвичай. На момент 10-го червня 2022 року на NFT-біржі OpenSea (https://opensea.io/) найдешевший NFT із колекції "Bored Ape Yacht Club" коштує 69.42 ETH (124,460$), найдорожчий – 18880 ETH (33,838,420$).

Звичайно, художники не малюють усі 10000 картин вручну, вони малюють лише окремі елементи цих картин (вуха, одяг, сережки і т.д.). Потім, ці елементи поєднуються в одну картину спеціальним скриптом, який має бути написаний програмістом.

У ході виконання курсової роботи було розроблено власну NFT-колекцію із 25 елементів та розгорнуто смарт-контракт в тестовій мережі Ethrereum Rinkeby.

```
1323    // Standard mint function
1324    function mintToken(uint256 _amount) public payable {
1325        uint256 supply = totalSupply();
1326        require( saleActive,                    "Sale isn't active" );
1327        require( _amount > 0 && _amount < 11,   "Can only mint between 1 and 10 tokens at once" );
1328        require( supply + _amount <= MAX_SUPPLY, "Can't mint more than max supply" );
1329        require( msg.value == price * _amount,  "Wrong amount of ETH sent" );
1330        for(uint256 i; i < _amount; i++){
1331            _safeMint( msg.sender, supply + i );
1332        }
1333    }
```

Рисунок 3.3.2 – функція mintToken

Основною функцією є “mintToken”. Вона працює за таким алгоритмом: на вхід приймається кількість токенів, що хоче придбати покупець; перевіряється чи включений продаж на даний момент; перевіряється, що кількість токенів, що хоче придбати покупець знаходить у визначеному діапазоні; перевіряється, що при придбанні цих токенів не буде перевищена загальна кількість; перевіряється, чи правильну кількість ETH пропонує покупець за дану кількість токенів. Якщо всі умови успішні, то для кожного токену викликається функція “_safeMint()” із батьківського контракту ERC721 бібліотеки Open Zeppelin.

```
1358    // Start and stop sale
1359    function setSaleActive(bool val) public onlyOwner {
1360        saleActive = val;
1361    }
1362
1363    // Set new baseURI
1364    function setBaseURI(string memory baseURI) public onlyOwner {
1365        baseTokenURI = baseURI;
1366    }
```

Рисунок 3.3.3 – функції setSaleActive та setBaseURI

Функція “setSaleActive” встановлює булевий прапорець, який визначає, чи доступні зараз продажі. Це важливо, адже зазвичай продажі колекцій починаються у певний визначений час, щоб підігріти аудиторію. Тому спершу поле “saleActive” має значення “false”, а коли настає момент продажу власник контракту викликає функцію “setSaleActive”, щоб встановити прапорцю значення “true”.

Функція “setBaseURI” теж є надзвичайно важливою, адже вона вказує на посилання, де лежать файли NFT-колекції. Зазвичай це посилання на розподілену мережу IPFS. Але по факту можна завантажити файл і на

певний централізований сервер, наприклад, AWS S3. Звісно, у аудиторії буде значно більше довіри до проекту, який користується децентралізованим IPFS.

```
// Price of each token
uint256 public price = 0.05 ether;

// Maximum limit of tokens that can ever exist
uint256 constant MAX_SUPPLY = 25;

// The base link that leads to the image / video of the token
string public baseTokenURI;

// Team addresses for withdrawals
address public a1;
address public a2;
```

Рисунок 3.3.4 – деякі поля смарт-контракту

Смарт-контракт зберігає стан. Наприклад, при створенні задається змінна “price”, яка визначає суму ETH за один токен. Задається константа “MAX_SUPPLY” – загальна кількість токенів в даній колекції. Оголошується змінна “baseTokenURI”, яку потім модифікує функція “setBaseURI”. Також оголошуються адреси, на які можна вивести токени ETH з адреси смарт-контракту.

```
1373    // Set team addresses
1374    function setAddresses(address[] memory _a) public onlyOwner {
1375        a1 = _a[0];
1376        a2 = _a[1];
1377    }
1378
1379    // Withdraw funds from contract for the team
1380    function withdrawTeam(uint256 amount) public payable onlyOwner {
1381        uint256 percent = amount / 100;
1382        require(payable(a1).send(percent * 50)); // 50% for Owner1
1383        require(payable(a2).send(percent * 50)); // 50% for Owner2
1384    }
1385 }
```

Рисунок 3.3.5 – функції setAddresses та withdrawTeam

Коли покупці викликають функцію “mintToken”, та вона успішно виконується, після завершення транзакції певна кількість ETH переводиться на адресу смарт-контракту. Для того щоб творці NFT-колекції мали змогу вивести ці кошти на власні гаманці, створюється функція “withdrawTeam”. Вона містить код, який відправляє монети ETH з адреси контракту на задані адрес в певному відсотковому співвідношенні. Попередньо має бути викликана функція “setAddresses”, щоб встановити

адреси творців колекції. Дуже важливо не помилитись при виклику “setAddresses”, щоб монети ETH не були втрачені.

Створення смарт-контракту – теж транзакція в блокчейні. Тому необхідно мати акаунт в Ethereum (в даному випадку в тестовій мережі Rinkeby). Контракт було написано, скомпільовано та розгорнути за допомогою середовища розробки Remix IDE. Варто зазначити, що тільки цей акаунт може викликати функції контракту, відмічені модифікатором “onlyOwner”.

Всі транзакції, пов’язані зі смарт-контрактом, можна переглянути на ресурсі https://rinkeby.etherscan.io/. Однією з родзинок блокчейну є прозорість. Всі можуть переглянути баланс ETH на рахунку смарт-контракту, адресу творця контракту, навіть номер транзакції, після якої було створено контракт.

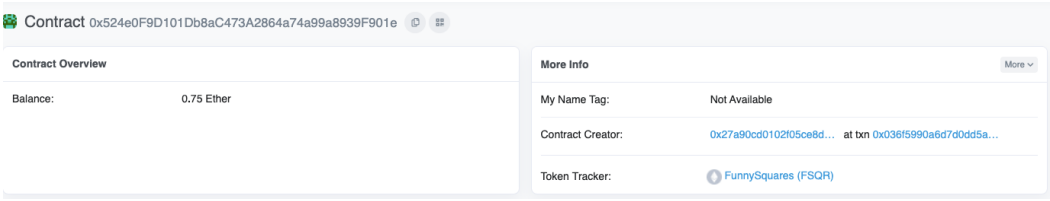


Рисунок 3.3.6 – інформація про смарт-контракт

Смарт-контракт можна знайти в etherscan за цією адресою: *0x524e0f9d101db8ac473a2864a74a99a8939f901e*.

Елементи NFT-колекції, породжені даним смарт-контрактом можна знайти на найпопулярнішій NFT-біржі за наступним посиланням: *https://testnets.opensea.io/collection/funnysquares*.

Висновки

В ході виконання курсової роботи була оглянута технологія блокчейн, історія її становлення, сфери її використання та проаналізовані чинники, що сприяли її популяризації. Крім цього, була досліджена технологія смарт-контрактів в мережі Ethereum, її перспективи.

Децентралізований реєстр даних, який є захищеним від несанкціонованого доступу в систему, є надійним джерелом довіри для людей, що хочуть зберігати свою анонімність та вільно розпоряджатись коштами без необхідності нагляду централізованих установ, як уряд чи банк. Ця рушійна ідея зробила блокчейн та технології, базовані на ньому, популярними та, безперечно, довготривалими.

P2P платіжна система Bitcoin свого часу здійснила революцію, проте платежі та фінтех – далеко не єдина сфера використання блокчейнів. Енергетичні, мультимедійні, роздрібні та інші компанії також інсталюють цю технологію в свої бізнес-процеси.

Мережа Ethereum пішла далі, ніж Bitcoin, створивши Ethereum Virtual Machine, яка дозволила розробляти DApp – децентралізовані застосунки. Базою для цього став протокол смарт-контрактів – цифрового аналогу звичайних договорів, суть якого полягає у виконанні прописаного коду при досягненні зазначених умов. І хоча наразі смарт-контракти здебільшого використовуються у цифровому світі через відсутність законодавчої бази, є ймовірність розповсюдження цієї технології на стандартні людські операції, як наприклад, купівля нерухомості.

Ще одним цікавим технологічним витоком, що з’явився завдяки Ethereum, стало цифрове мистецтво у вигляді NFT. В якості практичного застосування набутих знань в роботі реалізовано смарт-контракт для продажів елементів NFT-колекцій.

Список використаних джерел

1. Andreas M. Antonopoulos and Gavin Wood Ph. D (2021). *Mastering Ethereum*
2. Nathaniel popper (2015). *Digital Gold*
3. Jitendra Chittoda (2019). *Mastering Blockchain Programming with Solidity*
4. *https://ethereum.org/en/developers/docs/smart-contracts/*
5. *https://ethereum.org/en/developers/docs/transactions/*
6. *https://medium.com/crypto-currently/the-anatomy-of-erc721-e9db77abfc24*
7. *https://eips.ethereum.org/EIPS/eip-721*
8. *https://docs.openzeppelin.com/contracts/2.x/api/token/erc721*
9. *https://www.investopedia.com/terms/b/blockchain.asp*
10. *https://aws.amazon.com/ru/what-is/blockchain/*
11. *https://academy.binance.com/uk/articles/what-is-a-blockchain-consensus-algorithm*
12. *https://docs.opensea.io/docs/1-structuring-your-smart-contract*
13. *https://docs.soliditylang.org/en/v0.8.14/*