

РЕФЕРАТ

Робота складається з 39 сторінок та містить у собі 5 рисунків, 9 використаних джерел та 3 додатки.

БЛОКЧЕЙН, ETHEREUM, СЕРТИФІКАТ, ВАЛІДАЦІЯ, ТРАНЗАКЦІЯ, РОЗУМНИЙ КОНТРАКТ, ДЕЦЕНТРАЛІЗОВАНИЙ ДОДАТОК.

Об'єктом розробки є система зберігання та валідації виданих дипломів у цифровому вигляді, використовуючи блокчейн-технології та smart-контракти зокрема.

Метою роботи є дослідити існуючі підходи, запропонувати та розробити власну систему.

Методами дослідження є пошук в електронних ресурсах та консультування з спеціалістами в даній галузі.

Результатом роботи є розроблена функціональна система, в якій запропоновано новий підхід до зберігання та підтвердження власності даних.

На сьогодні існує декілька аналогів: Blockcerts від Массачусетського технологічного інституту, BCDiploma та інші.

Розроблений продукт може бути використаний у будь-якому освітньому закладі.

Даний підхід в роботі з сертифікатами є новітнім та надзвичайно надійним, і може назавжди вирішити проблему підробок будь-яких документів.

Можливий подальший горизонтальний розвиток - поширення запропонованого підходу на інші галузі.

ЗМІСТ

ВСТУП	4
РОЗДІЛ I. Технологія блокчейн	7
1.1 Загальний опис та порівняння	7
1.2 Визначення цифрової довіри	10
РОЗДІЛ II. Розгляд існуючих підходів	12
2.1 Blockcerts	12
2.1.1 Загальний опис	12
2.1.2 Відкрита платформа для репутації	12
2.1.3 Як це працює	14
2.4 Древа Меркла	14
2.4.1 Приклад роботи	15
2.4.2 Переваги	17
2.4.3 Використання	18
2.5 DApps - децентралізовані додатки	19
2.5.1 Переваги Dapps	21
2.5.2 Приклади	22
2.5.3 Класифікація Dapps	22
2.5.4 Як працюють Dapps	22
2.6 Smart - контракти	23
2.6.1 Що таке розумний контракт?	23
2.6.2. Розумні контракти складні	25
РОЗДІЛ III. Розробка підходу до вирішення проблеми	31
3.1 Опис	31
3.2 Реалізація	33
3.3 Інструкція з розгортання в тестувальних цілях	33
ВИСНОВКИ	35
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	37

ВСТУП

Сертифікати - це ознаки певного досягнення або членства, а деякі з них відіграють набагато важливішу роль в житті. Університетські ступені (певний тип сертифіката) можуть допомогти вам отримати потрібну роботу або завадити його отриманню, якщо у вас немає відповідного сертифіката. Наша поточна, переважно, аналогова система керування сертифікатами є повільною, складною та ненадійною. Є багато переваг для створення цифрових інфраструктур для сертифікатів, але ставки є високими, оскільки така система може зростати, щоб представляти нашу професійну репутацію. Ми повинні бути вдумливими щодо його дизайну та типу установ, яким ми доручаємо керувати нею.

Коли ми жили в невеликих громадах, люди знали, з ким вони можуть звернутися, коли їм потрібен фахівець (і кого уникати). Проте, коли ми почали рухатися далі, а наші суспільства і коло знайомих зростало, нам потрібно було запропонувати портативні способи передачі нашого досвіду новим знайомим. Деякі з цих оригінальних систем все ще існують. Наприклад, у Німеччині багато теслярів все ще проводять учніські заняття, які тривають не менше трьох років і одного дня. Вони носять невелику книгу, в якій вони збирають марки та довідники від майстрів плотників, з якими вони працюють на цьому шляху. Традиційний наряд столяра, книга штампів, яку вони

несуть, і, якщо все добре, сертифікат про прийняття в гільдію столяра є доказом того, що перед вами чоловік або жінка, яким ви можете довірити побудувати свій будинок.

В ідеалі ми повинні відповідати за власні повноваження, подібні до теслярів, які носять свої книги штампів та довідників. Але більшу частину часу треба покладатися на треті сторони, такі як університети або роботодавці, щоб вони зберігали, перевіряли та підтверджували наші облікові дані. Працівники, які шукають роботу, змушені вимагати офіційних протоколів зі своїх альма-матер (і зазвичай платять невелику плату), а роботодавці все ще повинні зателефонувати до університету, якщо вони хочуть бути впевненими, що стенограма не підроблена. Це повільний і складний процес, що є однією з причин, чому підробка звань є справжньою проблемою. Передавання сертифікатів та їх легкість перевірки є однією перевагою цифрових систем.

Кожного року в світі сотні мільйонів людей отримують дипломи, нагороди, сертифікати про закінчення курсів, та інші документи, що засвідчують їхні досягнення. Станом на сьогоднішній день практично всі навчальні заклади в світі здійснюють видачу таких документів шляхом їх друку та нанесення відповідних маркувань, що підтверджують валідність. Однак такий підхід є застарілим та ненадійним, з огляду на розвиток сучасних технологій. Як відомо, близько 30% дипломів бакалавра є підробкою, ще гірша ситуація серед кандидатів наук - цифра перетнула позначку в 50% [1].

Інноваційне вирішення проблеми було запропоноване в Массачусетському технологічному інституті - команда фахівців розробила електронну систему, основу на технології блокчейн, що здатна гарантувати неможливість підробки. Вона стала першою в своєму роді. Існує велика ймовірність, що вже в найближчому майбутньому такими сервісами будуть користуватися більшість освітніх установ. В даній роботі представлений детальний опис цієї розробки та її порівняння з іншими існуючими.

РОЗДІЛ І. Технологія блокчейн

"Практичні наслідки [... це ...] вперше - спосіб для одного інтернет-користувача передавати унікальний фрагмент цифрової власності іншому користувачеві Інтернету, такий, що передача гарантується безпечною та безпечною, всі знають, що передача відбулося, і ніхто не може кинути виклик легітимності передачі.

Наслідки цього прориву важко перебільшити ".

Марк Андреєссен

1.1 Загальний опис та порівняння

З крейсерської висоти блокчейн може не виглядати таким, що відрізняється від речей, які вам знайомі.

За допомогою блокчейну багато людей можуть створювати записи інформації, а спільнота користувачів може контролювати як змінюється та оновлюється запис інформації. Подібним чином, записи в Вікіпедії - це не продукт одного видавця. Ніхто не контролює інформацію.

Знижуючись до рівня землі, відмінності, які роблять унікальну технологію блокчейн, стали більш зрозумілими. Хоча обидва працюють на розподілених мережах, Wikipedia вбудована у всесвітню мережу (WWW), і використовує мережеву модель клієнт-сервер.

Користувач (клієнт) з дозволами, пов'язаними з його обліковим записом, може змінювати записи Вікіпедії, збережені на централізованому сервері.

Кожного разу, коли користувач отримує доступ до сторінки Вікіпедії, він отримає оновлену версію "основної копії" входу до Вікіпедії. Контроль за базою даних здійснюється за допомогою адміністраторів Вікіпедії, що дозволяє отримати доступ та дозволи для підтримки центрального органу влади.

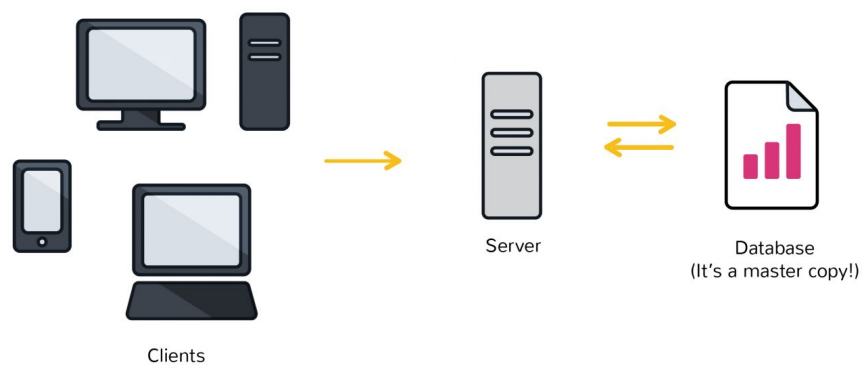


Рис. 1. Інфраструктура класичної централізованої системи

Цифрова інфраструктура Wikipedia подібна до високо захищених та централізованих баз даних, які сьогодні зберігають уряди, банки або страхові компанії. Контроль централізованих баз даних залежить від їх власників, включаючи управління оновленнями, доступ і захист від кібер-загроз.

Розподілена база даних, створена технологією блокчейн, має принципово інший цифровий хребет.

"Магічна копія" Вікіпедії редагується на сервері, і всі користувачі бачать нову версію. У випадку блокчейна кожен вузол мережі приходить до одного результату, кожен з яких оновлює запис

самостійно, причому найпопулярніший запис стає де-факто офіційною репутацією, замість того, щоб бути основною копією.

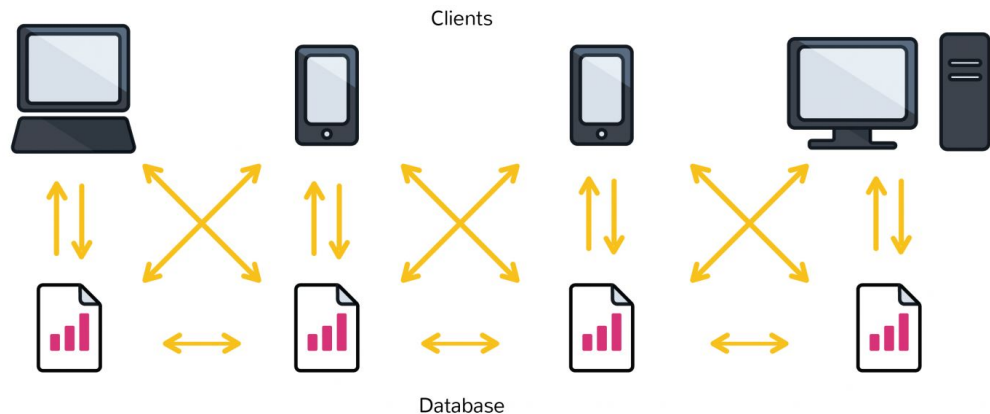


Рис. 2. Інфраструктура децентралізованої системи

Транзакції транслюються і кожен вузол створює власну оновлену версію подій. Саме ця відмінність робить технологію блокчейн настільки корисною - це інновація у веденні записів та розповсюдженні інформації, яка усуває необхідність третьої довіреної сторони для полегшення цифрових відносин.

Проте технологія блокчейн, не зважаючи на всі її переваги, не є цілком новою технологією. Скоріше, це поєднання перевірених технологій, що застосовуються новим способом. Це був особливий оркестр трьох технологій: Інтернет, криптографія приватного ключа та протокол, що керує стимулюванням, що зробило ідею творця криптовалюти Bitcoin Сатоши Накамото настільки корисною.

Результатом є система для цифрових взаємодій, яка не потребує довіреної третьої сторони. Робота з забезпечення цифрових взаємин

неявно забезпечується елегантною, простою, але надійною мережевою архітектурою самої технології блокчейн.

1.2 Визначення цифрової довіри

Довіра є судженням про ризик між різними сторонами, і в цифровому світі визначаючи довіру часто зводиться до підтвердження тотожності (автентифікації) та доказів дозволів (авторизації).

Простіше кажучи, ми хочемо знати відповідь на запитання: "Ви дійсно є тим, ким ви кажете, що ви є?" і "Чи маєте ви змогу робити те, що ви намагаєтеся робити?"

У випадку технології блокчейн криптографія приватного ключа забезпечує потужний інструмент власності, який виконує вимоги автентифікації. Власником приватного ключа є власність. Це також заощаджує людину від необхідності надсилати більше особистої інформації, ніж необхідно для обміну, залишаючи їх відкритими для хакерів.

Аутентифікації недостатньо. Авторизація - маючи достатньо грошей, транслуючи правильний тип транзакції тощо - потрібна розподілена мережа рівних рівнів як вихідна точка. Розподілена мережа зменшує ризик централізованої корупції і помилок.

Ця розподільна мережа також повинна бути призначена для ведення обліку та безпеки мережі транзакцій. Авторизація транзакцій є результатом роботи всієї мережі, що застосовує правила, на яких він був розроблений (протокол блокчейн).

Аутентифікація та авторизація, надані таким чином, дозволяють взаємодіяти в цифровому світі, не покладаючись на (вартісну) довіру. Сьогодні підприємці в галузях промисловості в усьому світі прокинулися від наслідків цього розвитку - можливі немислимі, нові та потужні цифрові відносини. Технологія Blockchain часто описується як хребет для рівня транзакції для Інтернету, основи Інтернету вартості.

Насправді, ідея про те, що криптографічні ключі та спільні облікові книги можуть стимулювати користувачів захищати та формалізувати цифрові взаємозв'язки, породжує цікаві фантазії. Кожен, від урядів до IT-компаній та банків, прагне побудувати цей транзакційний рівень.

Аутентифікація та авторизація, необхідні для здійснення цифрових транзакцій, встановлюються внаслідок конфігурації технології блокчейн.

Ідея може бути застосована для будь-якої потреби в надійній системі запису.

РОЗДІЛ II. Розгляд існуючих підходів

2.1 Blockcerts

2.1.1 Загальний опис

Blockcerts є відкритим стандартом для створення додатків, які видають та перевіряють офіційні записи на базі блоків. Вони можуть включати в себе сертифікати для свіс-записів, академічних мандатів, професійні ліцензії, розвиток робочої сили тощо [2].

Blockcerts складаються з бібліотек з відкритим кодом, інструментів і мобільних додатків, що забезпечують децентралізовану екологічну систему, орієнтовану на стандартизацію, що забезпечує надійну перевірку завдяки технології блокчейн.

Blockcerts використовує та заохочує консолідацію за відкритими стандартами. Blockcerts прагнуть до самостійної ідентичності всіх учасників та дозволяють одержувачам контролювати свої вимоги за допомогою простих у користуванні інструментів, таких як кошик сертифікатів (мобільний додаток). Blockcerts також зацікавлені в наявності векселів, без окремих пунктів невдачі [3].

2.1.2 Відкрита платформа для репутації

Стежка повноважень і досягнень, яку ми генеруємо протягом усього нашого життя, говорить про те, хто ми є, і може відкрити двері, які дозволять нам стати тим, кого ми хочемо бути. Деякі документи, такі як університетські ступені, важливіші за інші. Але наприкінці дня всі ці повноваження представляють досвід, який є частиною нашого життя.

Для одержувачів існує багато переваг для більшого контролю над сертифікатами, які вони здобувають. Перебувати під контролем не означає, що можна легко обдурити. Подібно до книжкової довідки столяра, не повинно бути можливості просто виривати кілька сторінок, яких ніхто не помітить. Але контроль означає, що ви маєте можливість зберігати облікові дані, переносити їх разом із нами та поділитися ними з роботодавцем, якщо ми вирішили це зробити (не платити, не вимагати дозволу або співпраці емітента).

Щоб це сталося, потрібна відкрита платформа для цифрових сертифікатів і репутації. За допомогою блочного циклу та сильної криптографії тепер можна створити сертифікаційну інфраструктуру, яка контролює повну інформацію наших досягнень та досягнень. Це дозволить нам поділитися цифровим ступенем з роботодавцем, одночасно даючи роботодавцю повну довіру, що ступінь фактично була видана особі, яка її представляла.

Це цікаво, тому що це не тільки кращий спосіб вирішити, як працюють сертифікати сьогодні, але це також можливість думати про

те, які сертифікати можуть виглядати в майбутньому. Кілька років тому я співавтор білого паперу з цифровими значками (просто інше ім'я для сертифікатів), в якому ми виклали деякі з цих основних ідей. То, що в той час було відсутнім, була технічною інфраструктурою, яка дозволила нам надійно зберігати і керувати сертифікатами [4].

Блокчейн найбільш відомий своїм зв'язком з біткойном криптовалюти. Але, по суті, це просто розподілений обліковий запис для зберігання транзакцій. Що робить його особливим, так це витривалість, перевіреність часом, прозорість та децентралізованість. Ці характеристики є однаково корисними для управління фінансовими операціями, як для системи репутації. Насправді ви можете думати про репутацію як тип валюти для соціального капіталу, а не фінансового капіталу.

2.1.3 Як це працює

Видача сертифікату є відносно простою: ми створюємо цифровий файл, що містить деяку основну інформацію, таку як ім'я одержувача, ім'я емітента, дату випуску тощо. Потім ми підписуємо вміст сертифіката використовуючи приватний ключ, до якого має доступ лише Media Lab, і додати цей підпис до самого сертифіката. Далі ми створюємо хеш, який представляє собою короткий рядок, який може бути використаний для перевірки того, що ніхто не порушив вміст сертифіката. І, нарешті, ми знову використовуємо наш приватний ключ, щоб створити запис про блокування біткойнів, в якому зазначено, що ми видали певний сертифікат певній особі на

визначену дату. Наша система дає змогу перевірити, хто видавав сертифікат, кого та перевірити вміст самого сертифікату. [4]

2.4 Дерева Меркла

Дерево Меркла є основною частиною технології блокчейнів. Це дерево є структурою, яка дозволяє ефективно і безпечно перевіряти вміст у великій кількості даних. Ця структура допомагає перевірити послідовність та зміст даних. Дерева Меркла використовують як біткоїн, так і ефіріум.

Дерево Меркла підсумовує всі транзакції в блоці шляхом створення цифрового відбитка всього комплексу транзакцій, що дозволяє користувачеві перевіряти, чи транзакція включена в блок.

Дерева Меркла створюються повторно хешуючими парами вузлів, поки не залишиться лише одного хешу (цей хеш називається Root Hash або Merkle Root). Вони побудовані знизу вгору, від хешів окремих транзакцій (відомих як ідентифікатори транзакцій).

Кожен листовий вузол є хеш-транзакційними даними, і кожен не-листовий вузол є хешем попередніх хешів. Дерева Меркла є двійковими і тому вимагають рівної кількості листових вузлів. Якщо кількість операцій є непарною, останній хеш буде повторюватися один раз для створення рівної кількості листових вузлів [5].

2.4.1 Приклад роботи

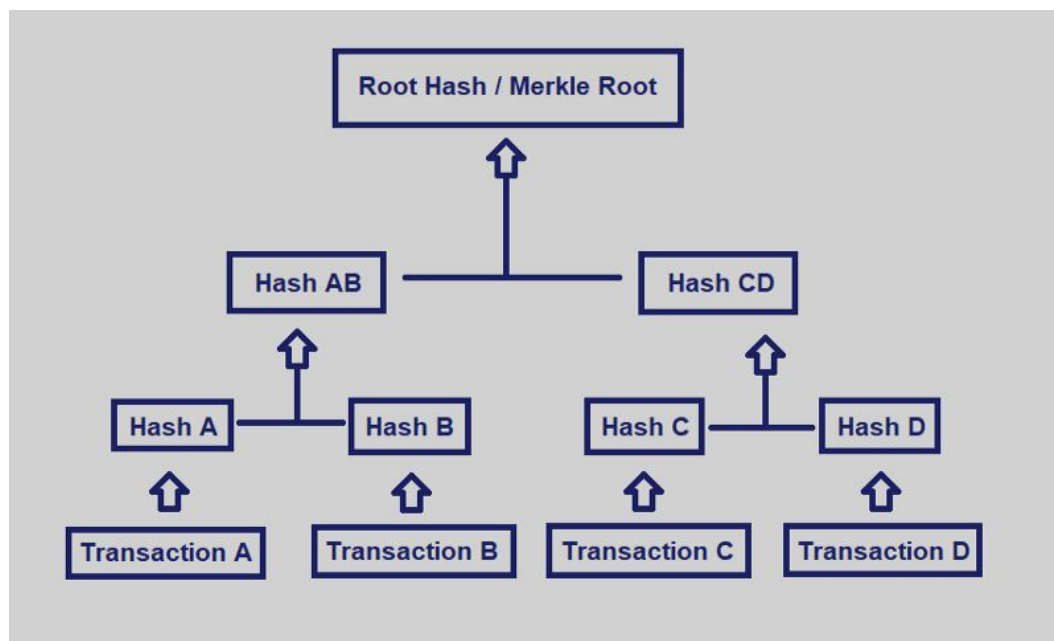


Рис. 3. Дерево Меркла транзакція A, B, C & D.

Давайте розглянемо приклад чотирьох транзакцій у блоці: A, B, C і D. Кожен з них є хешуванням, і хеш зберігається в кожному листовому вузлі, в результаті чого отримуються Hash A, B, C і D. Послідовні пари листових вузлів потім підсумовуються в батьківському вузлі шляхом хешування Hash A та Hash B, що приводить до Hash AB, і окремо хешування Hash C і Hash D, що приводить до Hash CD. Потім дві хеші (Hash AB та Hash CD) знову перемішуються, щоб створити Root Hash (Merkle Root).

Цей процес можна провести і на великих наборах даних: послідовні блоки можуть бути хешованими, доки у верхній частині

відсутній тільки один вузол. Хешінг зазвичай здійснюється за допомогою криптографічної хеш-функції SHA-2, хоча інші функції також можуть бути використані.

Merkle Root підсумовує всі дані у відповідних операціях і зберігається в заголовку блоку. Він зберігає цілісність даних. Якщо змінюється одна деталь у будь-якій транзакції або порядок транзакцій, то і Merkle Root. Використання дерева Merkle дозволяє швидко і просто перевірити, чи входить певна транзакція в набір чи ні.

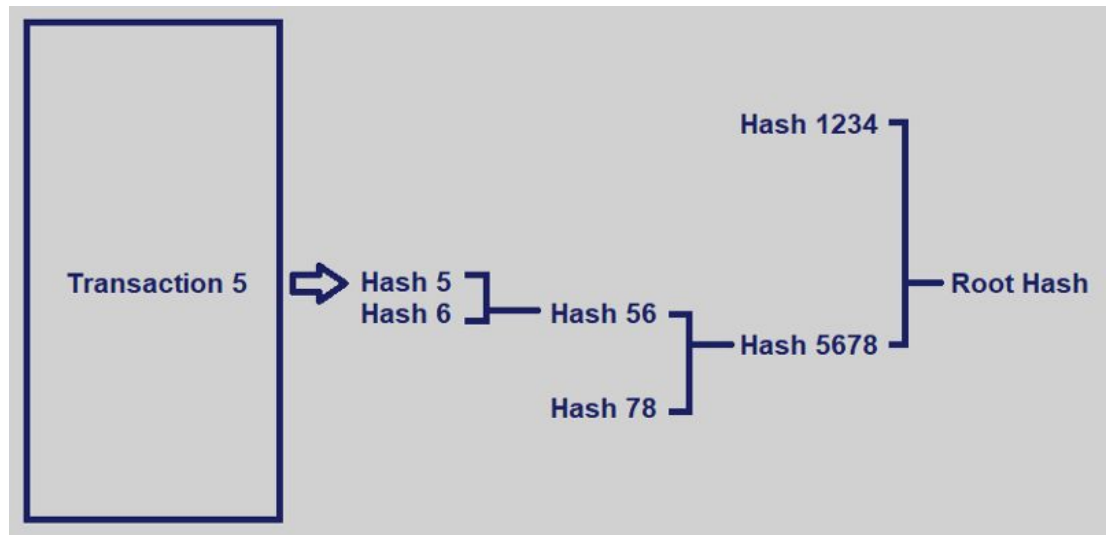


Рис. 4. Не обов'язково завантажувати всі дані, щоб підтвердити валідність транзакції 5

Дерево Меркла відрізняється від хеш-списку тим, що в дереві Меркла, одна гілка може бути завантажена в момент часу, а цілісність кожної гілки може бути негайно підтверджена, навіть якщо решта дерева ще доступна. Це вигідно, тому що файли можна розділити на

дуже малі блоки даних, наприклад, якщо тільки оригінальна версія пошкоджена, слід завантажувати лише невеликі блоки.

Використання дерева Меркла може значно зменшити кількість даних, які довірений орган повинен підтримувати для перевірки. Це відокремлює перевірку даних від самих даних. Дерево Меркла може розташовуватися локально або в розподіленій системі.

2.4.2 Переваги

Такі дерева мають три основні переваги:

1. Вони забезпечують засоби для підтвердження цілісності та обґрунтованості даних.
2. Вони вимагають мало пам'яті на диску, оскільки докази обчислюються легко і швидко.
3. Їхні докази та управління потребують лише невеликої кількості інформації, яка буде передана через мережі.

Можливість довести, що журнал є повним і послідовним є важливим для технології блокчейнів та концепції головної книги. Дерева Меркла допомагають перевірити, що пізніші версії журналу містять все, що стосується попередньої версії, і що всі дані записуються та відображаються в хронологічному порядку. Підтвердження того, що журнал є послідовним, потребує показу, що попередні записи не були додані або змінені, а також, що журнал ніколи не був розгалуженим.

2.4.3 Використання

Дерева Меркла корисні майнерам і користувачам на блокчейні. Майнер може поступово розраховувати хеш, оскільки він отримує транзакції від пірів. Користувач може перевіряти окремі частини блоків і може перевіряти окремі транзакції за допомогою хешів інших гілок дерева.

Спрощена перевірка платежів (SPV) - це спосіб перевірки того, чи входять певні транзакції в блок без завантаження всього блоку. Дерева Меркла широко використовуються вузлами SPV.

У вузлах SPV немає даних з усіх транзакцій у блоці. Вони завантажують лише заголовки блоків. Дерева Меркла вмикають вузли SPV на блок-схемі, щоб перевірити, чи майнери підтвердили транзакції в блоці без завантаження всіх транзакцій у блоці. Цей метод в даний час використовується деякими легкими клієнтами Bitcoin.

Ethereum використовує три різних дерева Меркла в кожному блоці:

1. Перший кореневий варіант - транзакції в блоці
2. Другий корінь - це стан
3. Третій корінь - для отримання транзакційних надходжень

Ethereum використовує спеціальний тип хеш-дерева під назвою Merkle Patricia Tree [5].

Дерева Меркла є потужними та незамінними інструментами для майнерів та користувачів на блокчейні. Вони надзвичайно надійні і є основою кількох однорангових мереж, таких як BitTorrent, Git, Bitcoin та Ethereum.

2.5 DApps - децентралізовані додатки

Більшість людей знайомі з "додатками" як з програмним забезпеченням. Програмні додатки - це програмне забезпечення, яке визначає певну мету. На відміну від найбільш розповсюджених централізованих моделей програмного забезпечення (модель "сервер-клієнт" є централізованою), dApps - це додатки, моделі яких є децентралізованими, а інші поширюються, як показано на малюнку нижче.

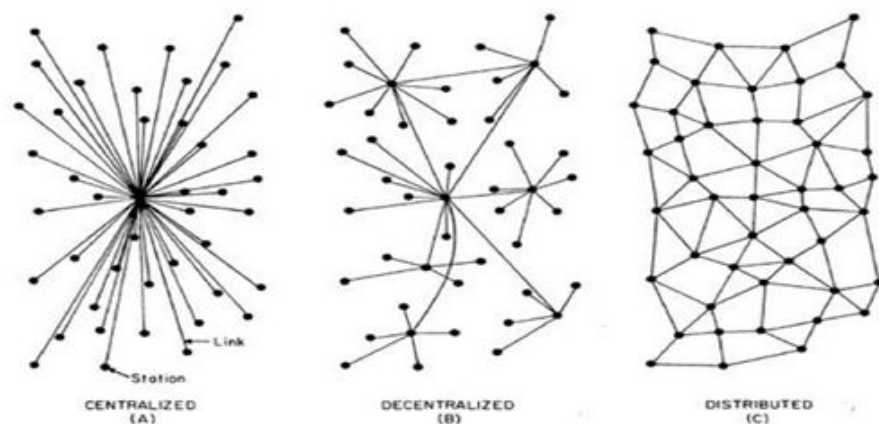


Рис. 5. Інфраструктура децентралізованих систем

Децентралізовані додатки мають відповідати наступним критеріям:

Вони повинні бути відкритими та автономними. Це означає, що будь-які зміни можуть виконуватися лише консенсусом, і немає єдиного органу, який володіє більшістю токенів у мережі.

Протоколи та дані зберігаються криптографічно в блок-схемі.

Криптографічні токени використовуються як для нагородження користувачів мережі, так і для доступу до додатків.

Токени створюються за допомогою алгоритму, який заохочує внесок членів системи в систему.

Ці чотири критерії можна було б спростити у трьох областях, згідно Віталію Бутеріну, засновнику Ethereum:

“Архітектурна - скільки комп'ютерів має система, і скільки вона їх може втримати, залишаючись працездатною.

Політична - скільки людей контролює комп'ютери, що компроментують систему.

Логічна - чи погляд на інтерфейс і структуру даних системи виявляють централізовану / єдину структуру, чи вона показує розподілену, яка може бути розбита на незалежні одиниці?” [6].

2.5.1 Переваги Dapps

Згідно з опитуванням компанії "Акана"[6], серед 250 фахівців з безпеки інформаційних технологій, відносний брак безпеки у традиційних додатках породив широкий спектр проблем. На додаток до DDoS, ін'єкції SQL і XML-бомби, інші проблеми, процитовані респондентів в опитуванні включені XML-брандмауер і безпеки на рівні повідомлень (43%), міжсайтовий скриптинг (38%), а також

XML-атаки (37%). Інші серйозні проблеми викликали атаки методом “грубої сили”, фішинг та персоналізовані атаки.

Ключовою перевагою децентралізованих програм є розподіл основних компонентів, рух, який покращує відмовостійкість і робить практично неможливим та дорогим атаку такої мережі. Основна частина Dapps-blockchain - забезпечує швидке, надійне та безпечне застосування для покращення взаємодії з клієнтами. Завдяки збільшеному обсягу та швидкості передачі даних з нових джерел, таких як IoT та соціальних мереж, blockchain та Dapps забезпечують швидкий, ефективний та доступний спосіб обробки великих даних. Крім того, децентралізація створює перешкоду для змови, який часто дозволяє корпораціям та урядам використовувати інші.

2.5.2 Приклади

Прикладами Dapps є:

- Проект, який дозволяє користувачам ділитися відео місць з усього світу.
- KYC-ланцюжок, який допомагає підтримувати "приватний гаманець", який можна використовувати для автентифікації персони в юридичних, фінансових або комерційних установах.

2.5.3 Класифікація Dapps

Одним з способів класифікувати Dapps є те, чи є у них власний blockchain, чи вони покладаються на інший блокчейн dapps. Виходячи з цього критерію, існує три типи розподілених додатків:

Тип I Dapps. Це Dapps, які мають власний блокчейн. Приклади включають bitcoin, litecoin та інші криптовалюти.

Тип II Dapps. Ці Dapps базують своє застосування на блоці Dapps Type I. Вони є протоколами і повинні мати функціональні значки. Прикладом є протокол Omni.

Тип III Dapps. Вони використовують протокол II типу Dapps. Вони мають протоколи та жетони як необхідність їх функціонування. Прикладом може бути мережа SAFE, яка використовує протокол Omni при видачі "safeecoins", які можуть бути використані для отримання розподіленого сховища файлів [6].

2.5.4 Як працюють Dapps

Існує, по суті, два шляхи, за допомогою яких експлуатуються Dapps: доказ роботи та доказ ставки. З доказом роботи консенсус та рішення стосовно будь-яких змін, які необхідно внести на Dapps, досягаються на основі кількості робіт, які кожна окрема зацікавлена сторона виконує для роботи операцій Dapps. Цей параметр зазвичай використовується Bitcoin для виконання своїх повсякденних операцій. Доказ роботи широко відомий як майнинг.

З підтвердженням ставкою, рішення про зміни в Dapps залежить від того, якою часткою конкретна зацікавлена сторона володіє відносно всіх інших.

2.6 Smart - контракти

Так само, як слова "блокчейн", "AI" та "хмара", "розумний контракт" - це одна з тих фраз, які отримують багато уваги.

Зрештою, що може бути краще, ніж можливість довіряти, що станеться замість використання судової системи? Обіцянки розумних контрактів включають:

- Забезпечення виконання договорів автоматично, без необхідності довіри та неупереджено
- Вилучення третьої сторони в будівництві контрактів та виконанні контрактів
- Вилучення адвокатів

2.6.1 Що таке розумний контракт?

Розумний контракт - це угода між двома або більше сторонами, яка зв'язує їх з чимось у майбутньому. Аліса може заплатити Бобу певні гроші за користування будинком Боба (орендна плата). Чарлі може погодитися відшкодувати будь-який збиток автомобілю Деніз у майбутньому в обмін на щомісячний платіж (автомобільне страхування).

Що відрізняє «розумний» контракт, так це те, що умови оцінюються та виконуються комп'ютерним кодом, а це робить його

довірливим. Тож, якщо Аліса погодиться платити Бобу 500 доларів за диван для доставки через 3 місяці, який-небудь код може визначити, чи відповідають ці умови (чи Аліса заплатила Бобу? Чи вже пройшло 3 місяці?) І виконати завдання (доставити кушетку з депозиту), не надаючи жодній із сторін можливість відмовитися.

Ключовою особливістю розумного контракту є те, що він має довірче виконання. Тобто вам не потрібно покладатися на третю сторону для виконання різних умов. Замість того, щоб покладатися на іншу сторону, сподіваючись, що вона дотримається свого слова або навіть гірше, покладаючись на адвокатів та правову систему, щоб виправити ситуацію, якщо щось піде не так, розумний контракт виконує те, що має відбуватися своєчасно та об'єктивно.

Використання слова "розумний" означає, що ці контракти мають певний вроджений інтелект. Вони цього не роблять. Розумна частина контракту полягає у відсутності необхідності співпраці іншої сторони для виконання угоди. Замість того, щоб виганяти людей, які не платять, "розумний" контракт заблокує неплатоспроможних орендарів з їхньої квартири. Виконання узгоджених наслідків є тим, що робить розумні контракти потужними, а не у контрактах вбудованого інтелекту [7].

По-справжньому розумний контракт враховував би всі пом'якшувальні обставини, дивився б на суть угоди та виносив справедливі рішення навіть у найбільш тендітних обставинах. Інакше кажучи, справді розумний контракт діяв би як справді хороший суддя. Але "розумний контракт" у цьому контексті не є розумним взагалі.

Насправді, це просто правила, які дотримуються і не можуть враховувати будь-які додаткові міркування.

Іншими словами, укладання контракту з довірою означає, що ми дійсно не можемо мати місця для двозначності, що створює наступну проблему.

2.6.2. Розумні контракти складні

Через велику кількість централізованого маркетингу від Ethereum існує помилкове переконання, що Smart Contracts існують лише в Ethereum. Це не правда. Біткоїн з самого початку 2009 року мав досить широку розумну контрактну мову Script. Фактично, розумні контракти існували до біткоїна ще в 1995 році. Різниця між розумною контрактною мовою Bitcoin та Ethereum полягає в тому, що Ethereum - Тюрінг повна. Тобто Solidity (розумна контрактна мова ETH) дозволяє більш складні контракти за рахунок того, що їх складніше аналізувати.

Є кілька істотних наслідків складності. Незважаючи на те, що складні договори можуть передбачати складніші ситуації, комплексний договір також дуже важко забезпечити. Навіть у звичайних контрактах, чим складніший договір, тим важче його застосовувати, оскільки ускладнення додають додаткову невизначеність та можливість інтерпретації. Завдяки розумним контрактам безпека означає обробку всіх можливих способів виконання контракту та забезпечення того, що контракт робить те, що намічають автори.

Виконання в контексті повноти за Тюрингом є надзвичайно неоднозначним і, відповідно, складним для аналізу. Захист інтелектуального контракту Тюринга стає еквівалентом доведення того, що в комп'ютерній програмі немає помилок. Ми знаємо, що це дуже важко, оскільки майже кожна комп'ютерна програма існує з помилками [8].

Враховуючи те, що написання звичайних контрактів вимагає багаторічного навчання та надзвичайно важкого іспиту, щоб вміти писати грамотно. Розумні контракти вимагають, принаймні, такого рівня компетенції і в даний час, багато які з них написані новачками, які не розуміють, наскільки безпечно те, що вони роблять. Це стає зрозуміло з різних контрактів, в яких були виявлені недоліки.

Рішення Bitcoin для цієї проблеми полягає в тому, щоб просто не мати повноти за Тюрингом. Це полегшує аналіз контрактів, оскільки можливі стани програми легше перелічити та вивчити.

Рішення Ethereum - це покласти тягар на розробників розумного контракту. Вони повинні переконатися, чи контракт дійсно робить те, що він має робити згідно їх очікувань.

Розумні контракти не є дійсно контрактами (принаймні на Ethereum).

Залишаючи відповідальність за забезпечення контрактів розробникам, що звучить добре в теорії, на практиці це мало серйозні централізовані наслідки.

Ethereum запускається з ідеєю, що "код - це закон". Тобто, контракт на Ethereum є найвищим авторитетом, і ніхто не може зняти

договір. Ідея полягала в тому, щоб дати чітко зрозуміти розробникам розумних контрактів, що вони самі по собі. Якщо ви ввімкнули свій власний розумний контракт, то в певному сенсі ви цього заслуговуєте. Це сталося під час розбиття, коли відбулася подія DAO.

DAO - "Децентралізована автономна організація", і фонд Ethereum був створений як спосіб показати, що може зробити платформа. Користувачі можуть класти гроші на депозит до DAO та отримувати прибуток на основі інвестицій, зроблених DAO. Самі рішення будуть спільними та децентралізовані. DAO підняв \$ 150 млн. в Ethereum, коли він торгувався близько \$20. Це все звучало добре в теорії, але виникла проблема. Код був не забезпечений дуже добре, і в результаті хтось з'ясував спосіб витягувати гроші з DAO [9].

Багато хто називав особу, що забирає DAO грошей "хакером". У тому сенсі, що "хакер" знайшов спосіб взяти гроші з контракту таким чином, що не передбачалося творцями, це правда. Але в ширшому сенсі, це був не хакер взагалі, тільки той, хто скористався примарами в розумному контракті на свою користь. Це не зовсім інше, ніж творчий правознавець, який визначає податкову лазівку, щоб заощадити гроші своїх клієнтів.

Те, що сталося далі, полягає в тому, що Ethereum вирішив, що цей код більше не є законом і повернув всі гроші, які надійшли в DAO. Іншими словами, контрактники та інвестори зробили щось дурне, і розробники Ethereum вирішили їх викупити.

Випадки цього інциденту добре документовані. Ethereum Classic народився, зберігаючи DAO, як написано, і зберегти принцип "code is

law". Крім того, розробники почали відмовлятися від використання властивості повноти за Тюрінгом Ethereum, так як доведено, що це важко забезпечити. Стандарти ERC20 та ERC721 є найбільш часто використовуваними розумними шаблонами контрактів в Ethereum, і важливо зазначити, що обидва типи контрактів можуть бути написані без будь-якої повноти за Тюрінгом.

Смарт-контракти працюють лише з цифровими інструментами носіїв

Навіть без повноти за Тюрінгом, розумні контракти звучать дуже добре. Врешті-решт, хто любить брати участь у суді, щоб придбати щось, що належним чином належить їм довірче? Чи не є використання розумного контракту набагато легшим, ніж звичайного?

Наприклад, чи не буде користі в галузі нерухомості від розумних контрактів? Аліса може довести, що вона володіє будинком. Боб може відправити гроші на будинок і отримати житло в обмін. Немає питань щодо володіння, надійне та швидке виконання машиною, відсутність потреби в судах, бюрократії або страхуваннях. Звучить чудово, чи не так?

Однак, тут є дві проблеми. По-перше, розумне виконання контракту централізованою партією насправді не є надійним. Ви все ще повинні довіряти централізованій партії виконувати його. Надійність - це ключова особливість, тому централізоване виконання не має сенсу. Щоб зробити розумні контракти дуже довірливими, потрібна платформа, яка насправді децентралізована.

Це веде нас до другої проблеми. У децентралізованому контексті розумні контракти працюють лише у тому випадку, якщо існує певний зв'язок між цифровою версією та фізичною версією. Тобто, коли цифрова версія будинку змінює власність, фізична версія повинна також змінити власність. Необхідно, щоб цифровий світ "знав" про фізичний світ. Це відомо, як "проблема Оракула".

Коли Аліса передає будинок Бобу, розумний контракт повинен знати, що вона фактично передала будинок Бобу. Існує кілька способів зробити це, але всі вони мають однакову важливу проблему. Існує потреба в довіреній третій стороні для перевірки подій у фізичному світі.

Наприклад, будинок може бути представлений як незмінний токен на Ethereum. Аліса могла передати будинок Бобу як атомний обмін для деякої кількості Ethereum. Але існує проблема. Боб повинен довіряти, що токен фактично представляє будинок. Там повинен бути якийсь Оракул, який гарантує, що передача домашнього токenu йому фактично означає, що цей будинок є його законним.

Крім того, навіть якщо державний орган стверджує, що токен фактично являє собою будинок, що ж тоді станеться, якщо вкрати токен? Хіба будинок тепер належить злодію? Що робити, якщо токен втрачено? Чи будинок більше не доступний для продажу? Чи може бути повторно виданий токен будинку? Якщо так, то хто повинен це робити?

Існує складна проблема прив'язки цифрового активу до його фізичного еквіваленту, незалежно від того, фрукт це, автомобіль чи

будинки, принаймні в децентралізованому контексті. Фізичні активи регулюються юрисдикцією, в якій ви перебуваєте, і це означає, що вони певним чином довіряють чомусь, крім розумного контракту, який ви створили. Це означає, що володіння у розумному контракті не обов'язково означає володіння у реальному світі та страждає від однієї проблеми довіри, як звичайні контракти. Розумний контракт, який довіряє третій стороні, вже не позбавляє необхідності в довірі.

Навіть цифрові засоби, такі як електронні книги, медичні записи чи фільми, страждають від такої ж проблеми. "Права" на ці цифрові активи, зрештою, вирішуються деякими іншими органами влади, і Оракулу потрібно довіряти.

І в цьому світлі, Оракули просто слабші версії суддів. Замість того, щоб отримати машинне виконання та спростити виконання, то, що ви дійсно отримуєте, - це складність необхідності кодувати всі можливі результати з суб'єктивністю та ризиком людського судження. Інакше кажучи, роблячи контракт "розумним", ви докорінно ускладнили написання, доки треба довіряти комусь.

Єдине, що може працювати без Оракула - це інструменти цифрового носія. По суті, обидві сторони торгівлі повинні бути не просто цифровими, а носіями інструментів. Тобто право власності на токен не може мати залежностей за межами розумної платформи контракту. Лише коли розумний контракт має цифрові документи-носії, розумний контракт дійсно може бути довіреним.

РОЗДІЛ III. Розробка підходу до вирішення проблеми

Як вже згадувалося раніше, практично всі навчальні заклади здійснюють видачу документів у друкованому вигляді. Єдиний спосіб підтвердити такий папірець - надіслати офіційний запит до відповідної установи, що зазвичай не є безкоштовним та відповіді варто очікувати не один робочий день. Саме цим і користуються шахраї. В мережі за два кліки можна знайти безліч варіантів отримання документу в найкоротші терміни, що ніяк не буде відрізнятися від оригіналу.

Даний підхід дозволяє вирішити цю проблему, причому відповідь є практично миттєвою та не вимагає жодних зборів.

Пропонується зберігати інформацію про видані дипломи у вигляді захешованих блоків в ланцюзі будь-якої криптовалюти.

3.1 Опис

Зважаючи на приведений вище аналіз, для вирішення поставленої проблеми було обрано розробити розумний контракт на базі блокчейну Ethereum. Цей контракт дозволяє зберігати унікальні ідентифікатори документів в блокчейні Ethereum, щоб довести їх існування. Ключовими перевагами є анонімність, конфіденційність та отримання децентралізованого доказу, який не може бути стертий або змінений будь-яким (третіми сторонами або урядами).

Безпека є ключовим фактором в даній ситуації. Унікальний Ethereum гаманець установи можна перевірити, просто відвідавши веб-сайт установи.

При видачі сертифікатів організації сервіс зберігає всю надану інформацію з унікальною адресою гаманця Ethereum студента на постійне зберігання та надсилає його до розумного контракту в блокчейні. Оскільки блокчейн є децентралізованим, його не можна підробити, змінити чи видалити.

Для всіх, хто знає хеш студентського сертифікату або Ethereum гаманця, валідація здійснюється дуже простим чином. Інформація про сертифікат, що має посилання на Etherscan (для перевірки статусу транзакції), отримується лише одним натисканням.

Сам запис в блокчейні створюється закладом, що видає цей сертифікат. Установа має свій унікальний гаманець Ethereum, адресу якого має повідомити всім іншим. Щоб зрозуміти чи справжній це документ, потрібно лише звірити чи дійсно видавцем є офіційна адреса. Також можна отримати додаткову інформацію про виданий сертифікат, у вигляді трьох рядків: адреси отримувача, його імені та назви самого документу.

3.2 Реалізація

Використовується мова програмування Solidity, яка була створена виключно для розробки розумних контрактів, та спеціалізоване середовище програмування Remix IDE. Контракт складається з 4 методів:

- `concat` імплементує поки відсутню конкатенацію двох рядків.
- `issueDocument` видає документ, тобто проводить транзакцію, яка записує всі дані (адресу видавця, адресу отримувача, номер блоку, власника та назву документа до блокчейну).
- `validateDocument` верифікує, що даний документ дійсно належить цьому адресату.
- `getDocument` повертає інформацію про документ.

3.3 Інструкція з розгортання в тестувальних цілях

Локальне розгортання програми відбувається наступним чином:

Вимоги:

npm - менеджер додатків JS

node - платформа для виконання мережевих застосунків

python3 - інтерпретатор мови програмування Python

ganache - симулятор блокчейну Ethereum

web3.js - JavaScript фреймворк для роботи з смарт - контрактами

solc - компілятор мови Solidity

Відкрити термінал і використати наступні команди:

node

```
> Web3 = require('web3')
> web3 = new Web3(new
Web3.providers.HttpProvider("http://localhost:8545"
));
> code = fs.readFileSync('dd.sol').toString()
> solc = require('solc')
> compiledCode = solc.compile(code)
> abiDefinition =
JSON.parse(compiledCode.contracts[':dd'].interface)
> ddContract = web3.eth.contract(abiDefinition)
> byteCode = compiledCode.contracts[':dd'].bytecode
> deployedContract = ddContract.new({data:
byteCode, from: web3.eth.accounts[0], gas:
4700000})
> cInstance =
ddContract.at(deployedContract.address)
```

ВИСНОВКИ

Блокчейн як технологія має серйозний потенціал, здатний суттєво вплинути на найрізноманітніші процеси та технології. У своїй основі блокчейн - це система для усунення необхідності довіряти транзакції. Хоча це може звучати як просте твердження, сьогодні багато хто з найбільших установ світу працюють як довірені треті сторони, наприклад, SWIFT та клірингова компанія Trust Depositary. Існує дуже багато корпоративних можливостей для компаній, які можуть створювати прикладні технології блокчейн, орієнтовані на конкретні операції, наприклад, іпотечну галузь.

Ще однією важливою перевагою blockchain є його розподіленість між кількома мережами, що робить його надзвичайно стійким до падінь у випадку авторитарного уряду або незаконної ділової практики. Наприклад, нерухоме майно, придбане за документацією із "розумного контракту" на блокчейні, не може бути вилучено чи приховане будь-яким органом влади, що робить його захищеним від зловживань.

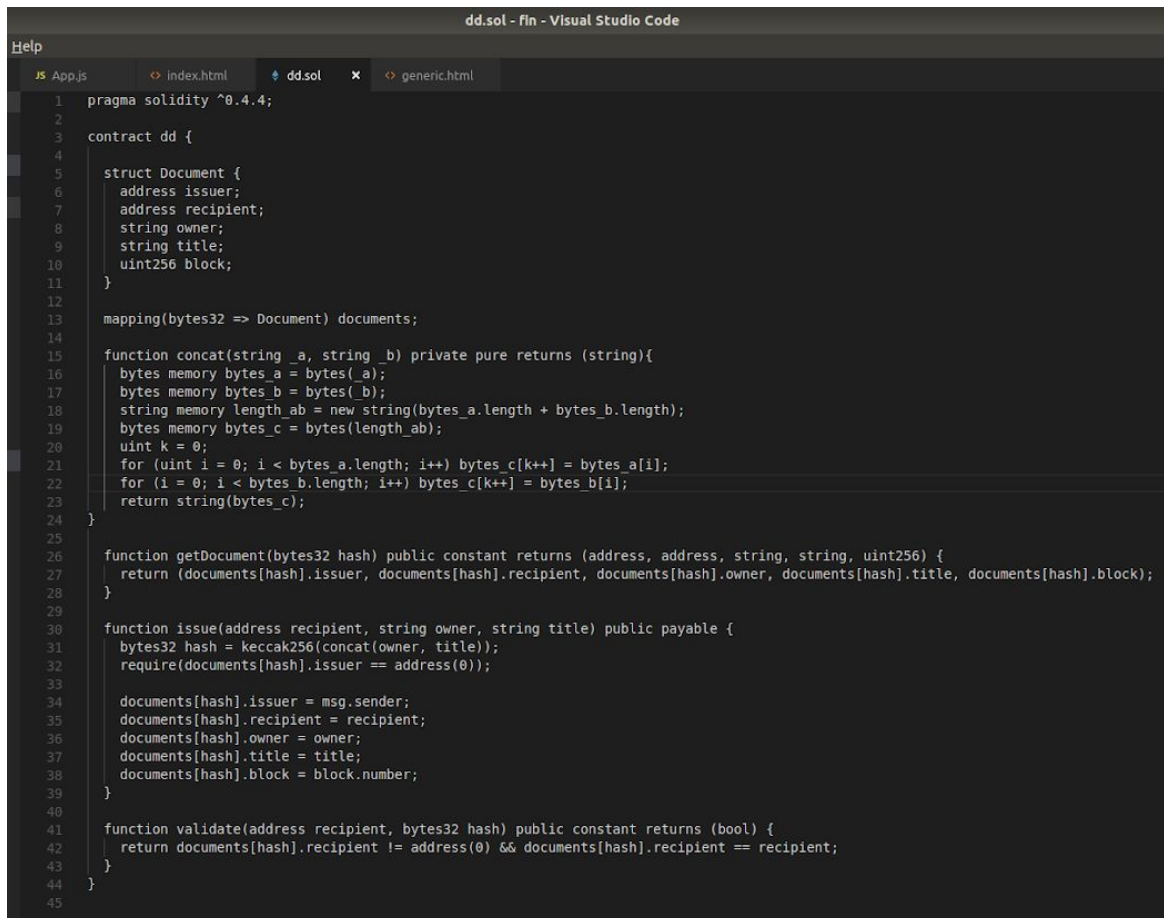
Нарешті, блокчейн - це чудовий інструмент для зберігання великої кількості важливих документів у таких галузях, як охорона здоров'я, логістика, авторське право тощо. Блокчейн усуває необхідність посередника при легалізації контрактів. Розумні контрактні платформи все ще удосконалюються, коли мова йде про зручність використання та, як очікується, будуть широко використовуватися в найближчі 5 років.

В даній роботі було здійснено спробу показати переваги технології блокчейн та розумних контрактів при вирішенні задачі підробки сертифікатів. Було розроблено і впроваджено в тестовому режимі сервіс з видачі та верифікації дипломів на блокчейні Ethereum. Він може застосовуватися будь - яким закладом, що здійснює видачу сертифікатів, необхідне лише змінити адресу контракту на свою.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. <https://www.coindesk.com/information/what-is-blockchain-technology/> (електронний ресурс)
2. <https://medium.com/mit-media-lab/certificates-reputation-and-the-blockchain-aee03622426f> (електронний ресурс)
3. <https://medium.com/learning-machine-blog/blockchain-credentials-b4cf5d02bbb7> (електронний ресурс)
4. <https://medium.com/mit-media-lab/what-we-learned-from-designing-an-academic-certificates-system-on-the-blockchain-34ba5874f196> (електронний ресурс)
5. <https://hackernoon.com/merkle-tree-introduction-4c44250e2da7> (електронний ресурс)
6. <https://www.coindesk.com/information/what-is-a-decentralized-application-dapp/> (електронний ресурс)
7. <https://medium.com/@Jernfrost/what-is-a-smart-contract-and-why-do-we-need-them-7d92f2131f03> (електронний ресурс)
8. <https://hackernoon.com/ethereum-smart-contracts-in-python-a-comprehensive-ish-guide-771b03990988> (електронний ресурс)
9. <https://www.coindesk.com/information/what-is-a-dao-ethereum/> (електронний ресурс)

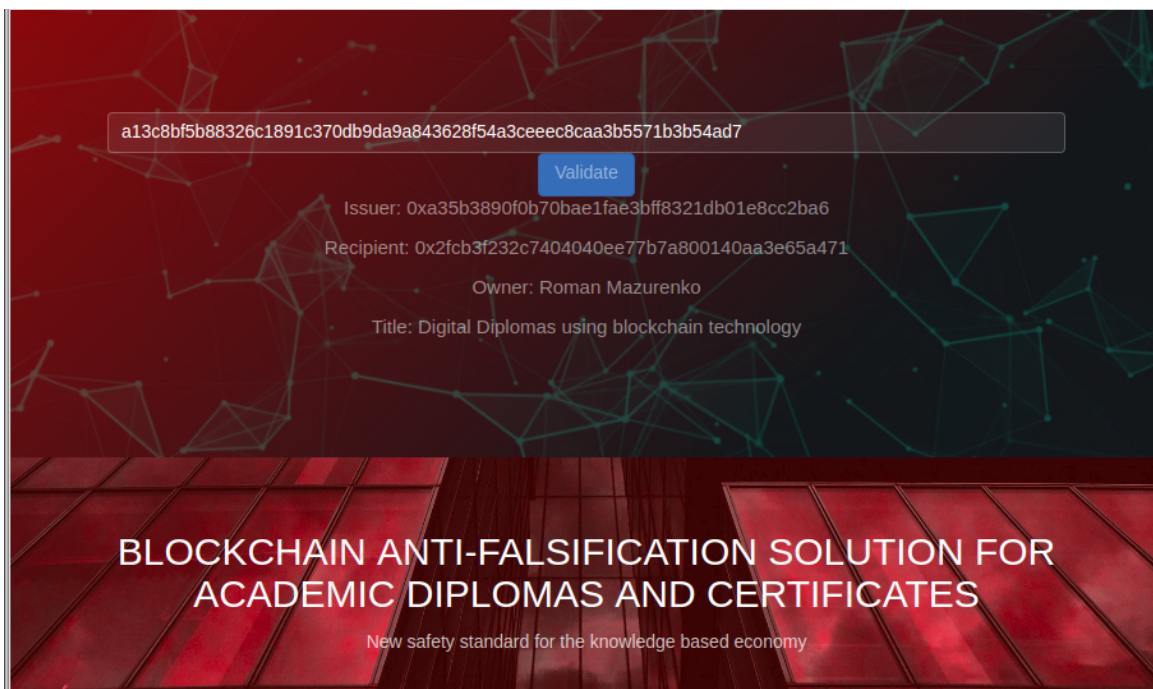
ДОДАТКИ



The image shows a screenshot of a Visual Studio Code editor window titled 'dd.sol - fin - Visual Studio Code'. The editor displays a Solidity smart contract named 'dd'. The code is as follows:

```
1 pragma solidity ^0.4.4;
2
3 contract dd {
4
5     struct Document {
6         address issuer;
7         address recipient;
8         string owner;
9         string title;
10        uint256 block;
11    }
12
13    mapping(bytes32 => Document) documents;
14
15    function concat(string _a, string _b) private pure returns (string){
16        bytes memory bytes_a = bytes(_a);
17        bytes memory bytes_b = bytes(_b);
18        string memory length_ab = new string(bytes_a.length + bytes_b.length);
19        bytes memory bytes_c = bytes(length_ab);
20        uint k = 0;
21        for (uint i = 0; i < bytes_a.length; i++) bytes_c[k++] = bytes_a[i];
22        for (i = 0; i < bytes_b.length; i++) bytes_c[k++] = bytes_b[i];
23        return string(bytes_c);
24    }
25
26    function getDocument(bytes32 hash) public constant returns (address, address, string, string, uint256) {
27        return (documents[hash].issuer, documents[hash].recipient, documents[hash].owner, documents[hash].title, documents[hash].block);
28    }
29
30    function issue(address recipient, string owner, string title) public payable {
31        bytes32 hash = keccak256(concat(owner, title));
32        require(documents[hash].issuer == address(0));
33
34        documents[hash].issuer = msg.sender;
35        documents[hash].recipient = recipient;
36        documents[hash].owner = owner;
37        documents[hash].title = title;
38        documents[hash].block = block.number;
39    }
40
41    function validate(address recipient, bytes32 hash) public constant returns (bool) {
42        return documents[hash].recipient != address(0) && documents[hash].recipient == recipient;
43    }
44 }
45
```

Додаток 1. Смарт контракт



Додаток 2. Вигляд веб-сторінки

```
roman@roman-X556UQ: ~/fin
File Edit View Search Terminal Help
> intance.issue("0x0e0122d3c606d4a8a6566ae7d501292e43240ba8", "Digital Diploma", {from: web3.eth.accounts[0]})
'0x1099f241617d1a8d20cff3b4d50dda4dc4133d6205350f8f8925b51b4c849c8a'
> intance.validate.call(0x0e0122d3c606d4a8a6566ae7d501292e43240ba8, "Digital Diploma")
false
> intance.validate.call(0x0e0122d3c606d4a8a6566ae7d501292e43240ba8, "Digital Diploma")
false
> intance.validate.call("0x0e0122d3c606d4a8a6566ae7d501292e43240ba8", "Digital Diploma")
true
> intance.validate.call("0x0e0122d3c606d4a8a6566ae7d501292e43240ba8", "Digital Diplom")
false
> deployedContract.address
'0xf370d364a57993915b27603a41b235b1cbc6b7e7'
>
```

Додаток 3. Консоль адміністратора

