Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
(повна назва)

Кафедра Безпеки інформаційних технологій
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Децентралізована система ідентифікації
(тема)

Виконав:
студент 2 курсу, групи БІКСзм-19-1
Слиш О.В.
(прізвище, ініціали)
Спеціальності 125 Кібербезпека
(код і повна назва спеціальності)
Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма Безпека інформаційних і комунікаційних систем
(повна назва освітньої програми)
Керівник доцент Мельникова О.А.
(посада. прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Халімов Г.З.
(підпис) (прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
(повна назва)

Кафедра Безпеки інформаційних технологій
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 125 Кібербезпека
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна, або освітньо-наукова)

Освітня програма «Безпека інформаційних і комунікаційних систем»
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри: _____ Халімов Г.З.
(підпис)
«____»_____ 2020 р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові Слиш Олексій Вікторович
(прізвище, ім'я, по батькові)

1. Тема роботи Децентралізована система ідентифікації

затверджена наказом по університету від 23.10. 2020 р. № 166 Стз

2. Термін подання студентом роботи до екзаменаційної комісії 17 грудня 2020 р.

3. Вихідні дані до роботи функція захисту – ідентифікація та автентифікація, технологія Blockchain, тип системи - децентралізована.

4. Перелік питань, що потрібно опрацювати в роботі
Аналіз систем цифрової ідентифікації та автентифікації
Децентралізовані системи цифрової ідентифікації
Аналіз blockchain проектів з ідентифікації особистості.
Реалізація протоколу ідентифікації blockstack.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5. включається до завдання за рішенням випускової кафедри)
презентаційний матеріал у вигляді слайдів

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання	23.10.20	Виконано
2	Робота з джерелами за тематикою роботи	23.10.20-31.10.20	Виконано
3	Вивчення основних понять в сфері децентралізованих систем	01.11.20-13.11.20	Виконано
4	Аналіз систем цифрової ідентифікації та автентифікації	14.11.20-22.10.20	Виконано
5	Децентралізовані системи цифрової ідентифікації	23.11.20-27.11.20	Виконано
6	Аналіз blockchain проектів з ідентифікації особистості	28.11.20-04.12.20	Виконано
7	Реалізація протоколу ідентифікації blockstack	05.11.20-10.12.20	Виконано
8	Публікація тез конференцій за результатами досліджень	23.10.20-15.12.20	Виконано
9	Оформлення пояснювальної записки	11.12.20-15.12.20	Виконано

Дата видачі завдання _____ _____20__ р.

Студент _____
(підпис)

Керівник роботи _____ __доцент Мельникова О.А.___
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка включає в себе 81 сторінка, 20 рисунків, 36 джерел, 1 додаток.

ІДЕНТИФІКАЦІЯ, АВТЕНТИФІКАЦІЯ, БЛОКЧЕЙН, БІТКОІН, ДЕЦЕНТРАЛІЗОВАНА СИСТЕМА

Об’єктом дослідження є децентралізована система ідентифікації з використанням технології blockchain.

Предмет дослідження це процес управління децентралізованою ідентифікацією користувача в інформаційній мережі.

Метою роботи є забезпечення управління децентралізованою ідентифікацією користувача в інформаційній мережі на основі використання технології blockchain.

В роботі проведений аналіз сучасних систем цифрової ідентифікації та автентифікації, розглянуті децентралізовані системи цифрової ідентифікації, проведений аналіз blockchain проектів з ідентифікації особистості та реалізований протокол ідентифікації blockstack.

ABSTRACT

Explanatory note to the thesis contains 81 pages, 20 figures, 36 references, 1 appendix.

IDENTIFICATION, AUTHENTICATION, BLOCKCHAIN, BITCOIN, DECENTRALIZED SYSTEM

The object of research is a decentralized identification system using blockchain technology.

The subject of research is the process of managing decentralized user identification in the information network.

The purpose of the work is to provide management of decentralized user identification in the information network based on the use of blockchain technology.

The paper analyzes modern digital identification and authentication systems, considers decentralized digital identification systems, analyzes blockchain projects on identity identification and implements blockstack identification protocol.

ЗМІСТ

с.

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

IT – інформаційна технологія

API – Application Programming Interface, інтерфейс створення додатків

CA – Certification Authority, засвідчувальний центр

DID – Decentralized ID, система децентралізованої ідентифікації

DNS – Domain Name System, система доменних імен

GDPR – General Data Protection Regulation, захист персональних даних на території Європейського Союзу

ION – Identity Overlay Network

KYC – Know Your Customer, знай свого клієнта

PII – Personal Identifiable Information, особиста ідентифікаційна інформація

URL – Uniform Resource Locator, система уніфікованих адрес

ВСТУП

Одним із основних елементів інформаційної безпеки є питання ідентифікацій та автентифікації, тобто перевірки автентичності однієї зі сторін інформаційного обміну. Традиційні системи мають деякі недоліки, а саме можливість втрати паролів, крадіжки аккаунтів, витоку призначених для користувача даних. На чорному ринку набирає популярність продаж цифрових особистостей - аккаунтів людей. Їх вартість сильно змінюється, від одного долара при покупці сотень "особистостей" до декількох десятків доларів при виборі відповідного облікового запису, що потрібен для певних цілей [1].

Другою проблемою є ідентифікація особи, тобто наявності у людини законної особистості – набору персональних даних про людину як суб'єкта права, його права і обов'язки. Згідно некомерційної організації ID2020, яка проводить щорічні конференції в штаб-квартирі ООН, одна п'ята населення Землі живе без надійного способу ідентифікувати себе, таким чином випадаючи з правового поля і стаючи вразливими для залучення в кримінальну діяльність [2].

Для більшості людей на даний момент основний спосіб засвідчити свою особистість - це паспорта, що видаються державними органами. При цьому одна і таж людина може бути громадянином декількох країн і мати кілька документів, які ідентифікують особистість.

Розвиток інформаційних технологій вимагає застосування систем електронної ідентифікації. Багато постачальників послуг в мережі Інтернет вимагають реєстрації, зберігання призначених для користувача даних на своїх серверах. У той же час цифрові записи, які пов'язані з унікальним ідентифікатором, становлять великий ризик для конфіденційності та захисту даних.

Крім того, в даний час все більше виникає нових систем

децентралізованої ідентифікації користувачів, основаних на технології blockchain. Тому виникає завдання проведення аналізу існуючих рішень ідентифікації користувачів на базі технології blockchain, покликаної усунути залежність від однієї компанії і децентралізувати механізм видачі законних особистостей в мережі Інтернет.

Об’єктом дослідження є децентралізована система ідентифікації з використанням технології blockchain.

Предмет дослідження це процес управління децентралізованою ідентифікацією користувача в інформаційній мережі.

Метою роботи є забезпечення управління децентралізованою ідентифікацією користувача в інформаційній мережі на основі використання технології blockchain.

Для досягнення мети в роботі вирішуються наступні задачі:

1. Аналіз систем цифрової ідентифікації та автентифікації.
2. Аналіз принципів побудови децентралізованих систем цифрової ідентифікації.
3. Аналіз blockchain проектів з ідентифікації особистості.
4. Реалізація протоколу ідентифікації blockstack.

1 АНАЛІЗ СИСТЕМ ЦИФРОВОЇ ІДЕНТИФІКАЦІЇ ТА АВТЕНТИФІКАЦІЇ

1.1 Аналіз проблем систем цифрової ідентифікації та автентифікації

На даний час, незважаючи на розвиток систем біометричної та багатофакторної автентифікації, найбільш поширеним способом автентифікації користувача залишається пароль. Але цей підхід, ідеальний для персональних комп'ютерів, став давати збої в багатокористувацьких системах та абсолютно не задовольняє умовам безпеки сучасних систем, де недовірених може бути і пристрій, і мережеве з'єднання, і навіть сервер. Сьогодні відомі багаточисельні інциденти, пов'язані зі зломом популярних сайтів і «зливом» звідти бази даних користувачів, включаючи хеші паролів [3].

Розвиток багатьох інформаційних сервісів, основаних на використанні мережі Інтернет, потребує від користувачів проходження незалежної перевірки ідентифікаційних даних при вході до кожної окремої системи (або мережі організації). Недоліком цього підходу є те, що користувачу кожен раз необхідно проходити однаковий процес реєстрації, ідентифікації і автентифікації, навіть якщо надається ідентичний набір даних.

Таким чином це призводить до додаткової ресурсозатратності цих процесів, розрізненості ідентифікаційних даних у різних постачальників послуг, відсутності єдиного уніфікованого формату при збереженні даних та складність пошуку зберігача окремих даних і неможливість підтвердження їх цілісності та автентичності.

Для кожного з користувачів інформаційних систем можна скласти набір з певних атрибутів (даних), який однозначно визначає його особистість, причому це можуть бути як фізичні характеристики (відбитки пальців, малюнок сітківки ока), так і їх цифрові аналоги (адреса електронної пошти, відкритий ключ

цифрового підпису). Всі ці дані є персональними даними або personal identifiable information (PII) [4].

Коли користувач хоче вперше скористатися послугою сервісу (наприклад, відкрити рахунок в банку, завести електронну пошту, отримати біометричний паспорт і т.п.), система запропонує йому надати набір ідентифікаційних даних про себе і на підставі них створить в своїй локальній базі обліковий запис.

При цьому бази даних окремих інформаційних систем ніяк не пов'язані і не синхронізовані одна з одною. Це призводить до того, що конкретний ідентифікатор може використовуватися тільки в системі, в якій він був виданий. Тому в даний час однієї фізичної особи (physical identity) відповідає безліч digital identities (рис. 1.1).

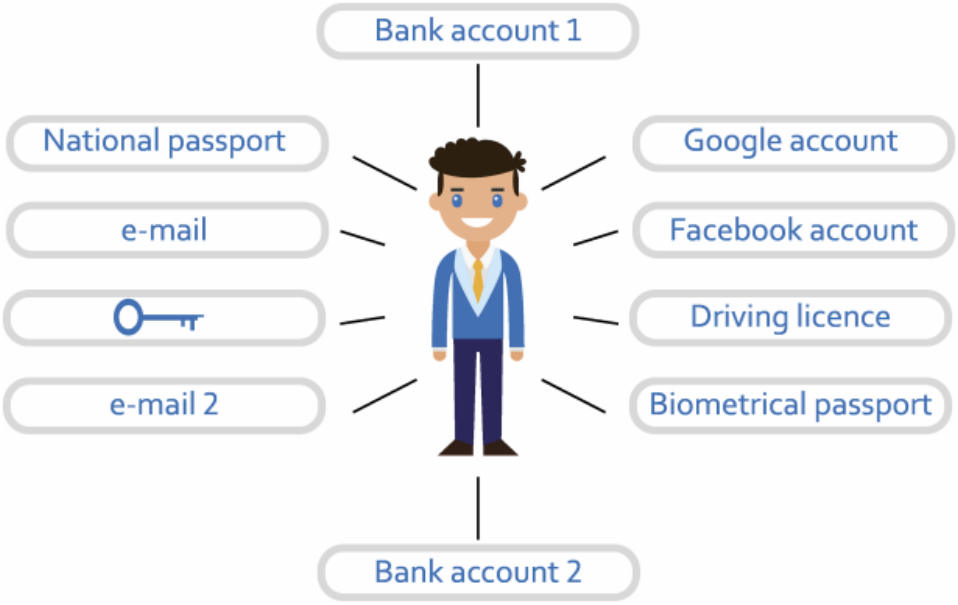

Рисунок 1.1 - Ідентифікатори користувача в різних облікових системах

Таким чином існуючі системи ідентифікації мають основні недоліки:

– нераціональне використання ресурсів;
– недотримання політики обробки персональних даних;
– ризик розкрадання персональних даних користувача.

Рішенням даних проблем може бути можливість використання раніше зібраних ідентифікаційних даних повторно в інших додатках.

Сучасні глобальні сервіси, такі як Google, Facebook, Twitter та інші, дозволяють користувачам входити в інші сервіси через протоколи OAuth, OpenID і OpenID Connect [5-12]. Проведемо аналіз даних систем.

1.2 Принципи функціонування протоколу OAuth

Протокол OAuth [5, 8-12] надає рішення, яке дозволяє стороннім додаткам отримати обмежений контрольований доступ до ресурсів (наприклад, до персональних даних користувача).

В протоколі OAuth власником ресурсу фактично є сам користувач. Він може надати дозвіл на отримання доступу до нього іншим сторонам (клієнтам).

Клієнтом може бути додаток/сервіс, який може формувати запити від імені власника ресурсу з метою отримання даних, які належать власнику ресурсу.

Хранитель ресурсу (resource server) - сервер, на якому зберігається захищений ресурс (наприклад, сервер Google, який зберігає дані користувача).

Сервер авторизації (authorization server) - сервер, який випускає токени доступу для клієнта після його успішної автентифікації (підтвердження прав на доступ до ресурсу).

Основна ідея протоколу OAuth полягає в тому, щоб замість облікових даних користувача (його логіна та пароля) для отримання доступу до захищеного ресурсу використовувати токен спеціального призначення.

Токени доступу видаються сервером авторизації зі схвалення власника ресурсу. Сервіс використовує токен для отримання доступу до захищених ресурсів, розміщених на стороні зберігача ресурсів (рис. 1.2).

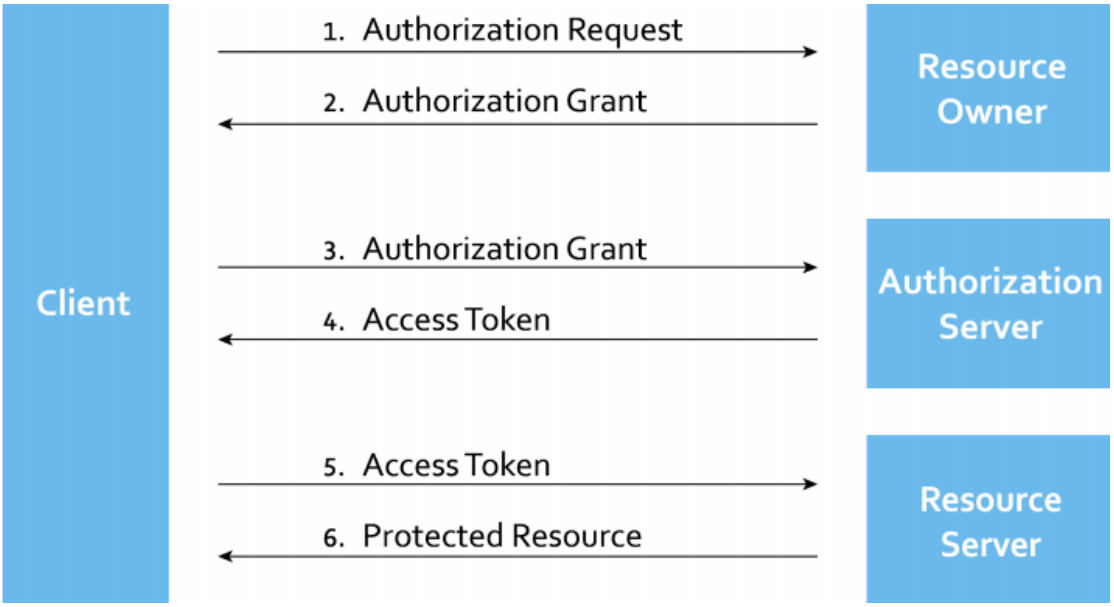

Рисунок 1.2 - Функціонування протоколу OAuth

Згідно з протоколом OAuth 2.0 передбачається наступний порядок взаємодії:

1. Сервіс формує авторизаційний запит і направляє його власнику ресурсу. Запит на авторизацію може бути відправлений користувачеві безпосередньо (рис. 1.2) або через сервер авторизації.

2. Власник ресурсу передає сервісу Authorization Grant, який містить підтвердження того, що користувач дійсно має права для надання доступу до ресурсу.

3. Сервіс запитує токен доступу на сервері авторизації. Для підтвердження того, що сервіс діє від імені власника ресурсу, в запит включається Authorization Grant, отриманий на попередньому кроці. В даному випадку Authorization Grant виступає підтвердженням успішної автентифікації користувача на стороні сервера авторизації.

4. Сервер авторизації перевіряє дозвіл сервісу, перевіряє Authorization Grant і, якщо він дійсний, видає токен доступу.

5. Сервіс запитує захищений ресурс на стороні зберігача ресурсу після автентифікації і використовує токен доступу.

6. Хранитель ресурсів перевіряє токен доступу, і, якщо він дійсний, обробляє запит. На цьому етапі хранитель ресурсу точно знає, якому додатку видається токен доступу і точно упевнений, що це відбувається за згодою власника даних.

Таким чином, користувачеві досить один раз ввести набір персональних даних на одному сервісі, після чого можливо буде видавати маркери доступу до різних частин такого набору різних клієнтам.

1.3 Протоколи OpenID і OpenID Connect

OpenID пропонує концепцію створення єдиної облікової системи для різних Інтернет-ресурсів, яка позбавила б користувачів необхідності постійно проходити процедуру ідентифікації і, відповідно, накопичувати нові пари логін-пароль.

Ідея полягає в тому, щоб зберігати єдиний обліковий запис на одному сервісі (у одного з OpenID провайдерів) і користуватися ним для реєстрації на інших сервісах, які підтримують OpenID протокол.

Протокол передбачає участь трьох сторін:

- користувач, який хоче використовувати свій OpenID ідентифікатор;
- Інтернет-сервіс, доступ до якого хоче отримати користувач;
- OpenID провайдер, який раніше провів процедуру ідентифікації користувача.

Протоколи OpenID першого і другого покоління [6, 7, 12] були націлені виключно на автентифікацію користувача на стороні одного з OpenID провайдерів, результат якої передавався безпосередньо стороні, що перевіряє, - сервісу, з яким хотів взаємодіяти користувач (рис. 1.3).

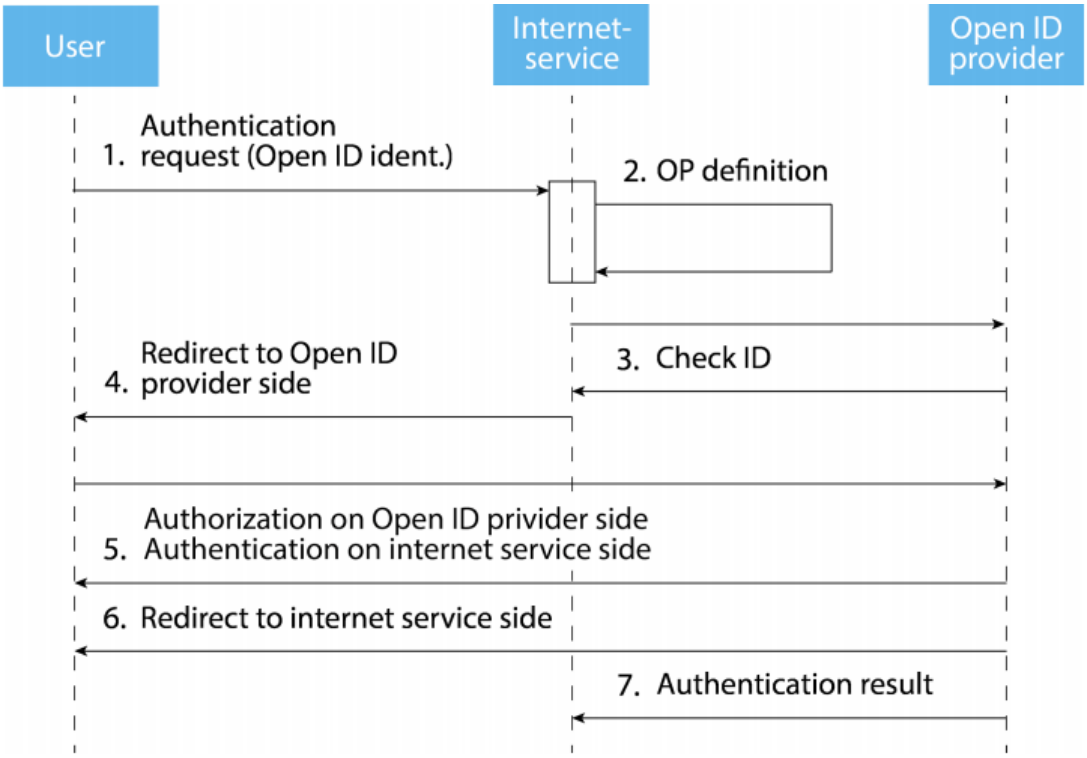

Рисунок 1.3 - Функціонування протоколу OpenID

Процедура встановлення каналу зв'язку між інтернет-сервісом і OpenID провайдером (на кроці 3) здійснюється за внутрішнім алгоритмом check_Id, який являє собою протокол двохпрохідної автентифікації і дозволяє взаємодіючим сторонам переконається в достовірності одна одної. Опціонально між Інтернет-сервісом і провайдером можливе встановлення загального секрету по протоколу Діффі-Хеллмана (в цьому випадку, за допомогою MAC-коду, Інтернет-сервіс може надійно автентифікувати провайдера без додаткових запитів щодо отримання інформації про його сертифікаті).

В результаті роботи протоколу: Інтернет-сервіс засвідчується в тому, що користувач є тим, за кого себе видає, тому що це підтвердив один з довірених провайдерів ідентифікації, проте, ніяка інша інформація про користувача не розкривається. Варто відзначити, що ні OpenID 1.0, ні OpenID 2.0 були сумісні з протоколом OAuth.

Протокол третього покоління, який отримав назву OpenID Connect [7], являє собою надбудову над протоколом авторизації OAuth 2.0. OpenID Connect дозволяє Інтернет-сервісам перевірити особу користувача на основі автентифікації, виконаної авторизаційним сервером (рис. 1.4).

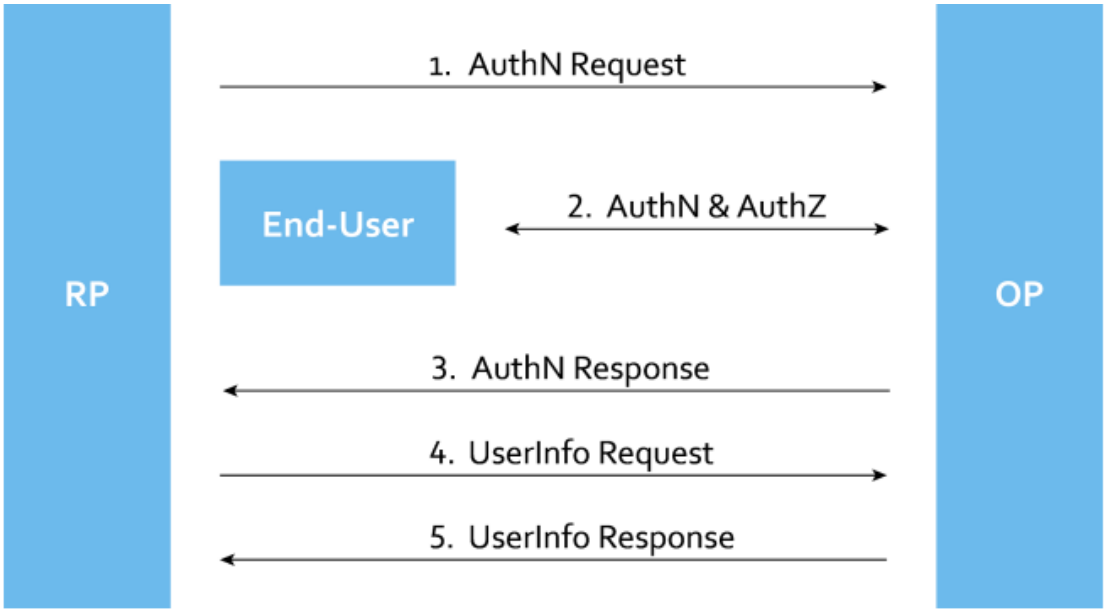

Рисунок 1.4 - Функціонування протоколу OpenID Connect

В рамках протоколу OpenID Connect використовуються ролі:

- кінцевий користувач (End-User) - об'єкт, який отримує доступ до ресурсу клієнта;

- сторона, що здійснює перевірку, (клієнт, RP) - сервіс, який згідно з протоколом OAuth 2.0 вимагає автентифікації кінцевого користувача і доступ до частини інформації про нього від OpenID провайдера.

- OpenID провайдер (OP) - сервер авторизації, здатний автентифікувати кінцевого користувача і надати клієнтській стороні інформацію про подію автентифікації і про кінцевого користувача.

В узагальненому вигляді взаємодія в рамках протоколу OpenID Connect складається з наступних кроків:

1. Сервіс направляє запит автентифікації до OpenID провайдера.

2. На кроці 2 користувач проходить процедуру автентифікації на стороні OpenID провайдера, після якої користувач повідомляє, яким саме сервісу необхідно надати доступ до персональної інформації.

3. В результаті видаються два токена (ID Token і Access Token), які включаються в AuthResponse. ID Token несе в собі інформацію про Authentication event - набір інформації про те, який користувач ініціював видачу токенів і для якої клієнтської сторони на сервері. Access Token є токеном, як в протоколі OAuth.

4. Використовуючи отриманий Access Token, сторона, що перевіряє, направляє запит на отримання інформації про користувача, яка зберігається на стороні OpenID провайдера.

5. OpenID провайдер перевіряє токен доступу, і, якщо він дійсний, обробляє запит.

Недоліками розглянутих протоколів є:

– відсутність прямого контролю користувачем своїх персональних даних;

– проблема довіри централізованим провайдерам;

– вразливості, пов'язані з автентифікацією за логіном/паролем;

– ризики використання сесій;

– відсутність синхронізації ідентифікаційних подій.

Розглянуті протоколи припускають зберігання критичної інформації (в т.ч. PII) користувача на одному ресурсі. Відповідно, забезпечення конфіденційності повністю залежить від політики безпеки даного ресурсу. Таким чином, в даному випадку користувач все ще не може повністю контролювати свої персональні дані.

Друге обмеження пов'язане з необхідністю довіри централізованим провайдерам ідентифікації. Сервіс отримує інформацію від конкретного

провайдера і вважає її вірною. Тобто, по суті, відсутня об'єктивність в процесі вирішення питання про довірі окремої identity.

Вразливості протоколу OAuth також пов'язані з сесійною природою протоколу. Якщо зловмисникові вдасться перехопити Authorization Grant під час такої сесії, то у нього з'явиться можливість реалізації атаки «Man in the middle».

Також користувач може мати кілька OpenID ідентифікаторів, які не будуть пов'язані один з одним. Таким чином, питання існування безлічі цифрових identity до кінця не вирішене.

Але ці протоколи не забезпечують необхідної надійності даних криптографічними методами і використовують механізми сеансу для отримання доступу до даних користувача.

Вирішення цих проблем можливо за рахунок використання децентралізованих систем цифрової ідентифікації.

2 ДЕЦЕНТРАЛІЗОВАНІ СИСТЕМИ ЦИФРОВОЇ ІДЕНТИФІКАЦІЇ

2.1 Принципи побудови глобальної системи ідентифікації

Піонером глобальної системи ідентифікації була Microsoft з її ідеєю Microsoft Passport, який став іменуватися першим кроком на шляху до справжньої віртуальної ідентичності - Identity 1.0. Споживачі могли завести собі зручне уніфіковане посвідчення особи від відомої корпорації, а комерційні компанії - безліч зареєстрованих користувачів. Однак, у Microsoft Passport були проблеми зі зручністю і збереженням налаштувань користувача, так і постійні перейменування цього сервісу - в .NET (2001 рік), потім в Windows Live ID (2006 рік) і нарешті в Microsoft account (2012 рік) - не надавали впевненості в його надійності і послідовності. Крім того, за гіпотетичну зручність доводилося платити небезпекою втрати, разом з єдиним паролем, відразу доступу до всіх використовуваних ресурсів.

Identity 2.0, заснована на використанні профілів соціальних мереж як посвідчень особи на багатьох інших сайтах. Популярність перших вдало поєдналася тут з розвитком мережевої ідентичності на тлі проблем з логінами і паролями версії 1.0. З іншого боку, зручність завжди обертається загрозою безпеки або приватності: разом з ідентичністю користувача в тій чи іншій соціальній мережі веб-сайт або корпорація отримують картину інтересів і зв'язків користувача.

Основні принципи Identity 3.0 були сформульовані в 2014 році за трьома групами факторів: ризик, приватність і функціональність. У принципах першої групи обґрунтовується, що ідентифікація повинна працювати як онлайн, так і автономно, видаватися авторитетним джерелом по кожному з атрибутів такої ідентичності і використовуватися з повним визнанням усіх можливих ризиків. Приватність підкреслює відсутність централізованої системи видачі

ідентифікації та мінімізацію публічності шляхом використання тільки необхідних відомостей в кожному окремому випадку залежно від контексту, а також накладає заборону на поширення біометричної інформації. Нарешті, в плані функціональності відзначається, що процес мережевої ідентифікації повинен бути по можливості якомога більше непомітним для кінцевого користувача, протікаючи в фоновому режимі і передбачаючи взаємозамінність цифровий репрезентації людей, пристроїв і організацій.

Проведений аналіз показав, що основним принципами глобальної системи ідентифікації є [12, 13]:

– прямий контроль користувачем власних ідентифікаційних даних;

– Identity користувача містить цифровий «digest» персональних даних з метою забезпечення перевірки їх цілісності;

– зв'язок ідентифікаційних даних з ключами, які належать користувачу;

– розподілене зберігання ідентифікаційних даних провайдерами, які ці дані підтвердили;

– синхронізація ідентифікаційних подій між незалежними провайдерами.

Основна роль глобальної цифрової системи ідентифікації - зв'язок глобального ідентифікатора користувача з його відкритим ключем і персональними даними, які належать цьому користувачу, за допомогою перевірки цього зв'язку декількома identity providers.

Основний принцип ідентифікації - вона починається з людини, яка знаходиться в центрі кожного цифрового взаємодії.

В основі даної моделі лежать принципи цифрової ідентифікації. Вони сфокусовані на правах володіння та користування даними, на конфіденційності, згоді, прозорості, безпеці та відкритості. Разом вони складають фундаментальне право кожного: «Я володію власної електронної ідентичністю і контролюю управління своїми персональними даними» [14].

1. Інклюзивність. Кожен має право на цифрову ідентифікацію особистості.

2. Право власності. Персональні дані належать тільки користувачу.

3. Простота використання. Підтвердження особистості онлайн повинно бути простим і інтуїтивним для користувача.

4. Конфіденційність. Користувач має право зберігати конфіденційність своїх персональних даних.

5. Згода. Персональні дані можуть бути використані і передані третім сторонам тільки за згодою користувача або відповідно до закону.

6. Прозорість. Користувач має право знати, як використовуються і поширюються його особисті дані.

7. Безпека і цілісність. Ідентифікаційні дані та транзакції, в яких задіяно електронна ідентичність, повинні оброблятися у відповідності з найвищими стандартами безпеки.

8. Права володіння та користування даними. Користувач має право на доступ, виправлення і видалення особистих даних, а також на звернення до суду в разі правопорушень.

9. Законне використання. Ідентифікаційні дані користувача можуть бути використані тільки в законних та прозорих цілях, що не дискримінують його.

10. Свобода вибору. Користувач повинен мати свободу при виборі провайдера цифрової ідентифікації особистості, а також право відмовитися від його послуг.

Проведений аналіз дозволив визначити наступні характеристики цифрової ідентифікації [14]:

- комбінація актуальних, високоякісних цифрових даних, які ідентифікують людину;

- динамічна, багатофункціональна, дозволяє багаторазове використання;

- метод перевірки інформації для забезпечення права доступу до послуг, виконання завдань або отримання бонусів;
- широка мережа сховищ даних (банки, мобільні оператори, державні установи), яка при необхідності забезпечує підтвердження особи.

Найбільш безпечний підхід для автентифікації (і авторизації) полягає в тому, що для кожної облікової системи, з якою взаємодіє користувач, право проведення дозволених для ідентифікатора операцій (і доступу до дозволених даних) перевіряється за допомогою володіння відкритим ключем, до якого прив'язаний ідентифікатор. Володіння відкритим ключем перевіряється за допомогою обчислення (і подальшої перевірки) цифрового підпису.

Облікова система, з якою в даний момент хоче взаємодіяти користувач, отримуючи запит від нього, може звернутися до глобальної цифрової системі ідентифікації і отримати актуальний відкритий ключ, відповідний конкретному ідентифікатору, після чого перевірити підпис запиту (рис. 2.1).

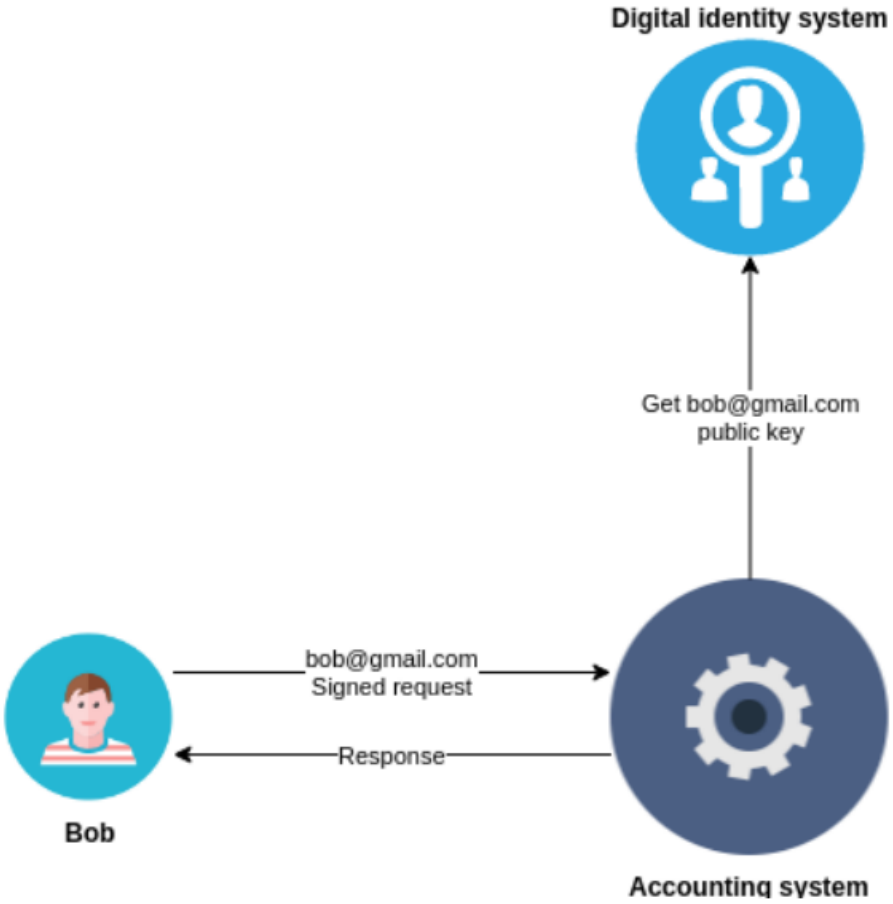

Рисунок 2.1 – Перевірка зв'язку ідентифікаційних даних з відкритим ключем при отриманні запиту

Зберігання критичної інформації у відкритому вигляді вимагає дорогих методів захисту, проте можливо зберігати не саму інформацію, а її цифровий відбиток (геш-значення), за яким буде легко визначити, що користувач дійсно володіє доступом до своїх персональних даних.

Користувач повинен мати можливість зберігати різні набори ідентифікаційних даних в різних облікових системах і при необхідності надавати запитувану третьою стороною інформацію без необхідності створення нового набору.

Якщо користувачу необхідно надати системі данні, що зберігаються в різних облікових системах, то на сьогоднішній момент, навіть з використанням протоколів OpenID та OAuth, не існує зручного для користувача способу дистанційно надати необхідний набір даних.

В концепції глобальної системи ідентифікації користувачу, по суті, необхідно буде сформувати і видати системі набір дозволів із зазначенням, в яких з існуючих облікових систем зберігаються частини необхідного набору даних (рис. 2.2).

Такий набір дозволів може підписуватися особистим ключем користувача, так що кожен з identity providers зможе надійно перевірити, що запит виконується з дозволу власника даних (користувача).

Blockchain - розподілена база даних, яку вперше почали використовувати для криптовалют, таких як Bitcoin. Blockchain - база даних, в криптовалютах вона зберігає послідовність всіх грошових транзакцій.

Копія цієї бази даних знаходиться одночасно на всіх вузлах системи (хоча для продуктивності деякі вузли можуть мати урізані права і не зберігати базу в повному вигляді). Додавання до блокчейну нової транзакції можливо тільки на основі консенсусу, тобто спільної верифікації декількома вузлами системи [13, 15].

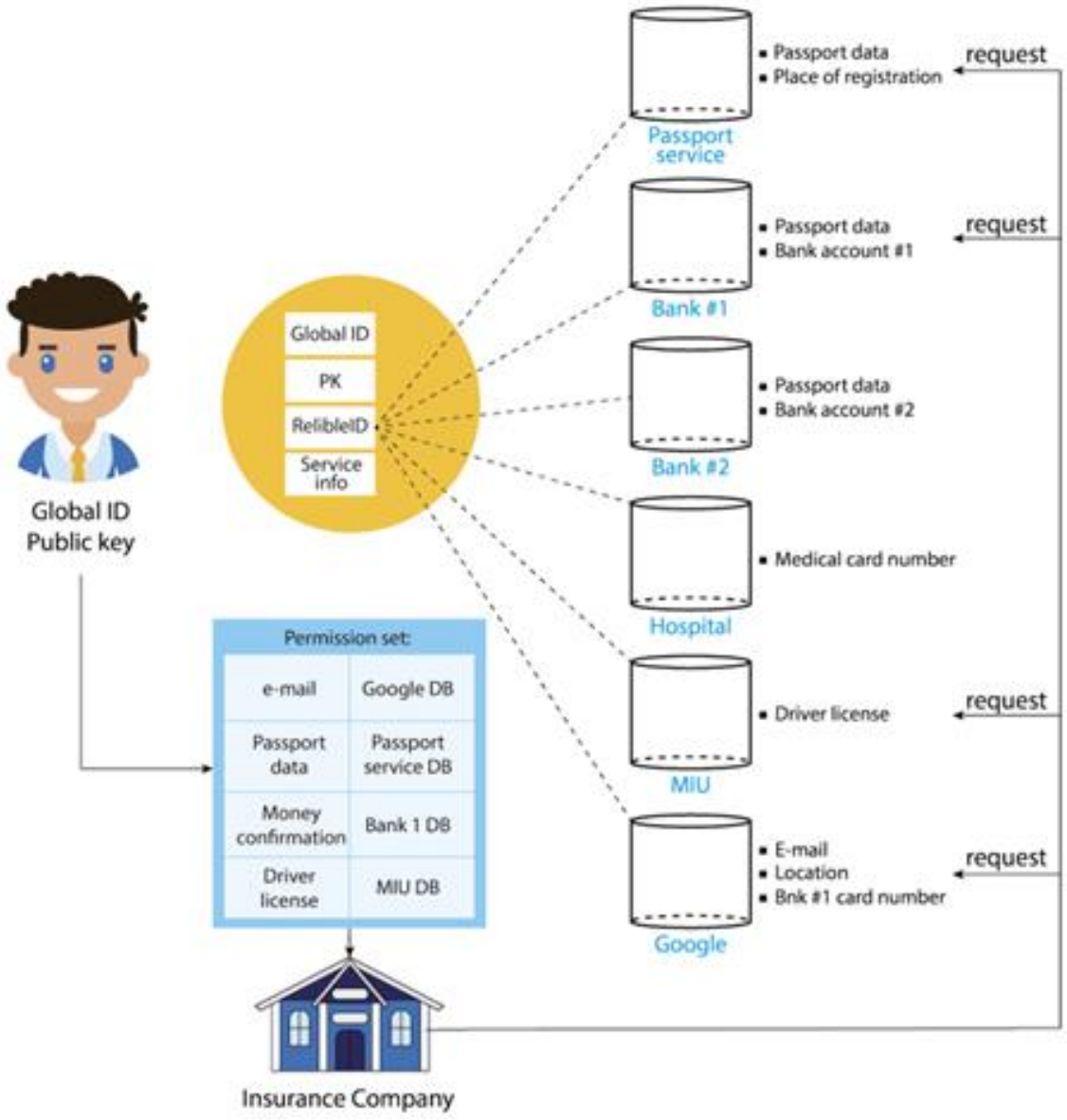

Рисунок 2.2 - Отримання інформації від різних провайдерів ідентифікації

Згодом технологія blockchain знайшла застосування у багатьох сферах, де була потрібна надійність, захищеність і відсутність єдиної сторони, якої потрібно довіряти. Останнім часом Blockchain почав використовуватися і в області ідентифікації: Microsoft заявили про розробку розподіленої системи ідентифікації спільно з двома компаніями, що спеціалізуються на blockchain - Blockstack Labs і ConsenSys [16]. Рішення цих двох компаній, Blockstack і uPort, є досить перспективними розробками, проте також не позбавлені недоліків.

2.2 Глобальна система ідентифікації з використанням технології blockchain

Технологія Blockchain здатна перетворити усталені бізнес-процеси і радикально змінити роботу з регуляторами.

В середині жовтня 2020 року компанія PwC (PricewaterhouseCoopers) представила аналіз, згідно з яким до 2030 року блокчейн-технології забезпечать зростання світової економіки на $ 1,7 трлн [17]. PwC зазначила п'ять ключових областей застосування блокчейна і оцінила їх потенціал для створення цінності, використовуючи економічний аналіз і галузеві дослідження. У ці п'ять ключових областей входять відстеження потоків грошових коштів; платежі і фінансові послуги; управління ідентифікацією; контракти і врегулювання спірних ситуацій; взаємодія з клієнтами. На думку аналітиків PwC, найбільш вигідно застосування блокчейна в таких галузях, як державне управління, освіта та охорона здоров'я [17].

Організація мережі між провайдерами ідентифікації з використанням технології blockchain дозволяє створити розподілену базу даних з мітками часу для кожної проведеної транзакції. Транзакцією в такій системі може бути подія проходження користувачем процедури ідентифікації у одного з провайдерів. Це дозволяє легко відслідковувати транзакції, вони є незворотними, а також запобігає шахрайству, зловживанням і будь-яким іншім видам маніпуляцій [13].

Після проведення процедури ідентифікації користувача, провайдер додає відповідний запис в ланцюжок блоків.

Якщо користувачеві необхідно отримати доступ до служб інший системи/сервісу, користувач звертається до нього, використовуючи свій власний ідентифікатор.

Система, в свою чергу, запрошує дані про те, хто з інших провайдерів вже ідентифікував користувача. Власник кожної системи самостійно визначає, наскільки довіряти подіям ідентифікації (а відповідно, і результатам цих подій),

випущеним іншою системою. Якщо рівень довіри максимальний, то система надає користувачеві послугу на підставі його ідентифікатора. Якщо довіра до системи (або набору систем) нижче, то клієнт може запросити дані користувача і переконатися, що вони відповідають тим, які зберігаються в розподіленому реєстрі.

Варто зазначити, що доступ до персональних даних здійснюється тільки після схвалення користувача. Якщо наданих даних достатньо, система дає користувачу доступ; якщо недостатньо (наприклад, необхідні додаткові дані), то система індивідуально проводить ідентифікацію користувача з відповідним записом у реєстрі.

Процедура ідентифікації в такій системі представлена на рис. 2.3.

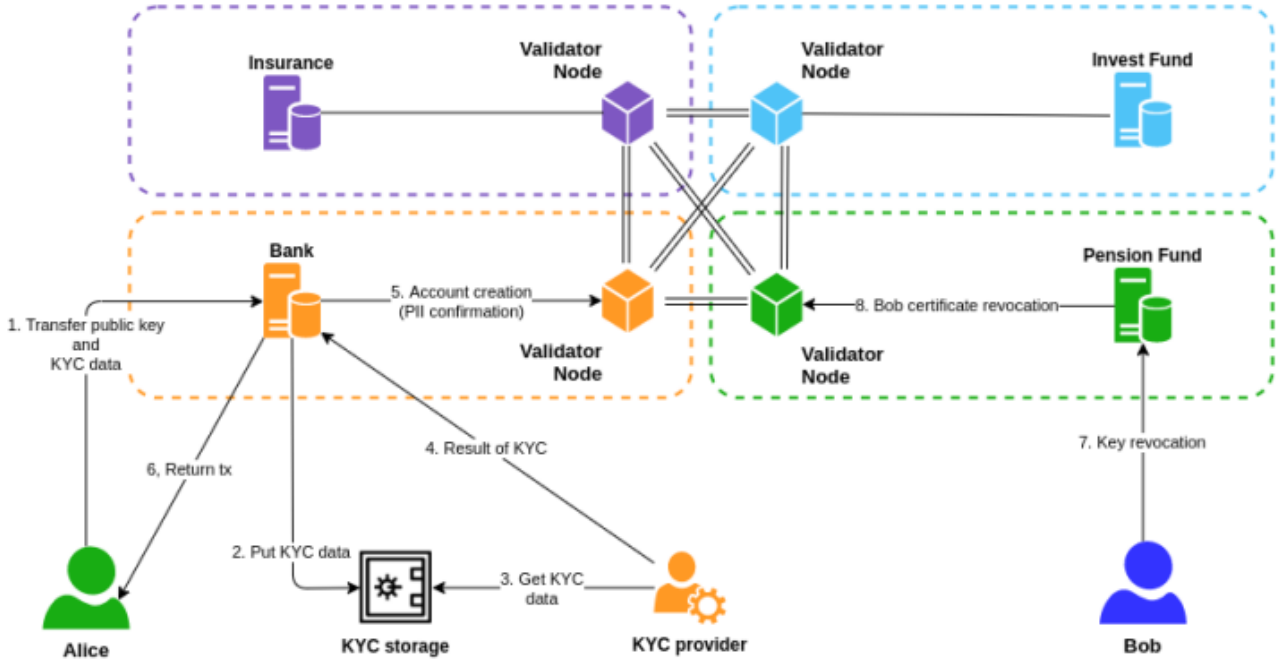

Рисунок 2.3 - Проходження користувачем ідентифікації з додаванням відповідної події в систему

1. Аліса відправляє свій власний відкритий ключ і дані, необхідні для проходження процедури ідентифікації KYC (Know Your Customer), банку, який є провайдером ідентифікаційних послуг.

2. Банк поміщає отримані персональні дані користувача в надійне сховище KYC, з дотриманням вимог GDPR (General Data Protection Regulation, захист персональних даних на території Європейського Союзу). Дані KYC повинні бути зашифровані банком, і тільки він може безпосередньо отримати доступ до даних сховища. Якщо дані KYC хоче отримати стороння організація, їй необхідно зв'язатися з банком, який, перш ніж надати ці дані, запросить схвалення користувача.

3. Постачальник KYC отримує доступ до даних користувача і проводить процедуру KYC (таким постачальником може бути як сам банк, так і сторона, якій банк довіряє проведення даної процедури).

4. Після цього провайдер KYC передає в банк результати процедури KYC. Якщо процедура не вдалася, банк інформує користувача про це, користувач додає відсутні дані і повторює крок 1.

5. Якщо процедура KYC була успішно завершена, банк створює транзакцію, яка містить відкритий ключ користувача (підписаний ключем банку) і значення геш-функції від усіх призначених для користувача даних (як доказ зберігання цих даних). Також він додає в транзакцію інформацію про те, що він ці дані підтвердив. Вузли-валідатори обробляють транзакцію і додають її в ланцюжок блоків.

6. Банк повертає Алісі підписану транзакцію. Аліса може перевірити, що відповідний запис був доданий в ланцюжок блоків і що дані в транзакції відповідають відправленим нею даними.

7. Якщо Боб хоче змінити свій відкритий ключ, він інформує про це свій орган ідентифікації.

8. Орган ідентифікації створює транзакцію, яка містить інформацію про анулювання раніше діючого сертифіката, і додає її в ланцюжок блоків.

Валідатори обробляють транзакцію і додають її в ланцюжок блоків, в результаті чого раніше встановлений відкритий ключ стає недійсним для всіх учасників системи.

Розглянемо, процес організації використання ідентифікатора для доступу до сервісів інших організацій (рис. 2.4).

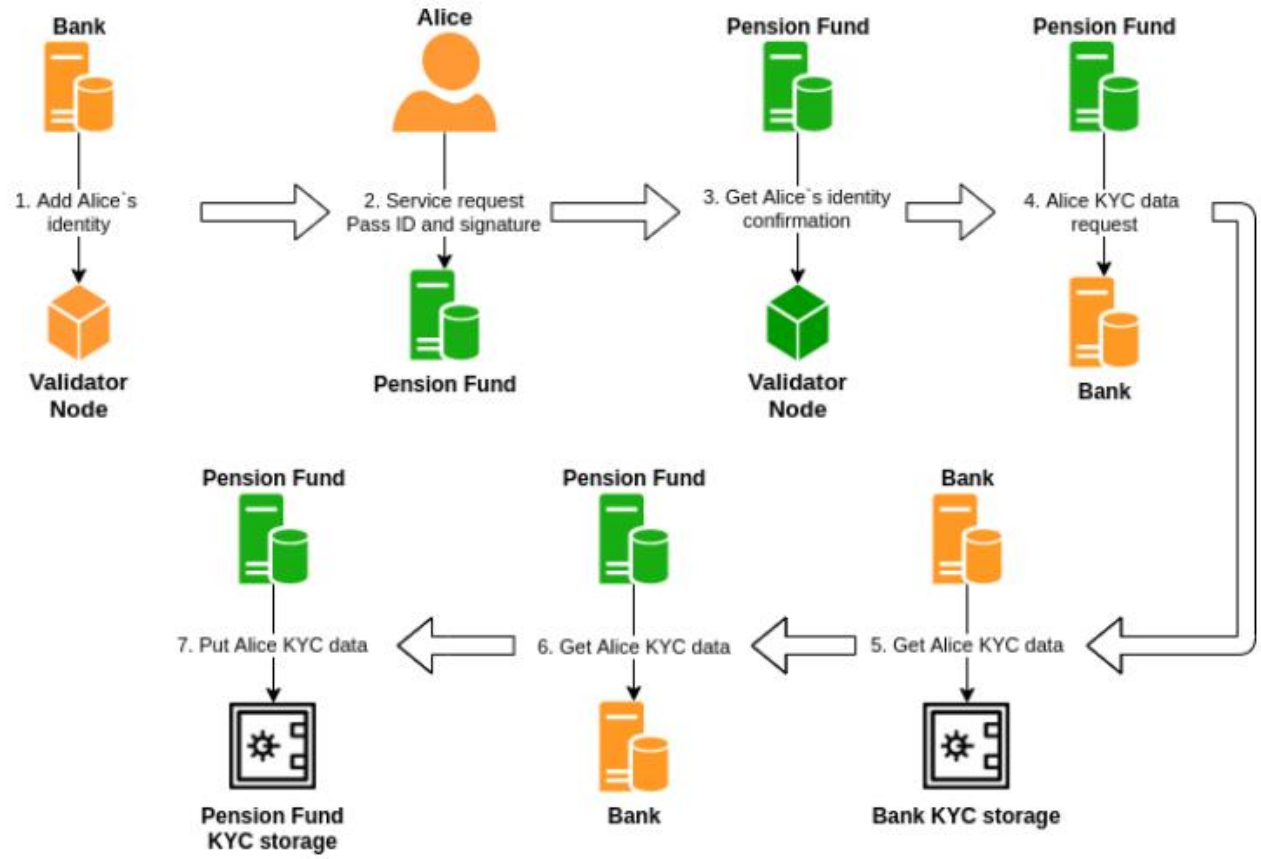

Рисунок 2.4 - Процес отримання доступу до сервісів при використанні отриманого ідентифікатора

1. Попередньо банк додає подію ідентифікації Аліси в розподілену базу даних (описана вище).

2. Аліса хоче взаємодіяти з новою системою (Pension Fund). Вона не зареєстровано в ней, але нова системою також є вузлом в системі ідентифікації. Аліса формує запит до Pension Fund, в який включає свій ідентифікатор

(Відкритий ключ), і підписує запит особистим ключем, який підтверджує право власності на ідентифікатор.

3. Pension Fund перевіряє підтвердження ідентифікатора Аліси в розподіленої базі. Якщо Pension Fund довіряє банку щодо ідентифікації та сертифікації користувачів, він повинен надати таку ж послугу Алісі.

4. Якщо Pension Fund недостатньо довіряє банку (наприклад, банк є компанією з недостатньою репутацією), він може запросити ідентифікаційні дані Аліси у банку (якщо Аліса дає свою згоду).

5. У цьому випадку банк отримує дані зі сховища і передає їх в Pension Fund. При цьому Аліса повинна надати дозвіл на доступ до її персональних даних під час формування запиту до Pension Fund.

6. Pension Fund отримує дані і може перевірити їх достовірність (геш даних, що зберігаються в розподіленій базі), після чого Pension Fund надає Алісі доступ до його послуг.

7. Pension Fund зобов'язаний зберігати цю інформацію, тому він повинен відразу отримати її і зберегти у своєму власному сховищі.

За допомогою технології блокчейн можливо організувати систему ідентифікації для використання у різних областях: в сфері охорони здоров'я, в промисловості та інших. На рисунку 2.5 представлена схема ідентифікації з використанням датчиків IoT автомобіля користувача.

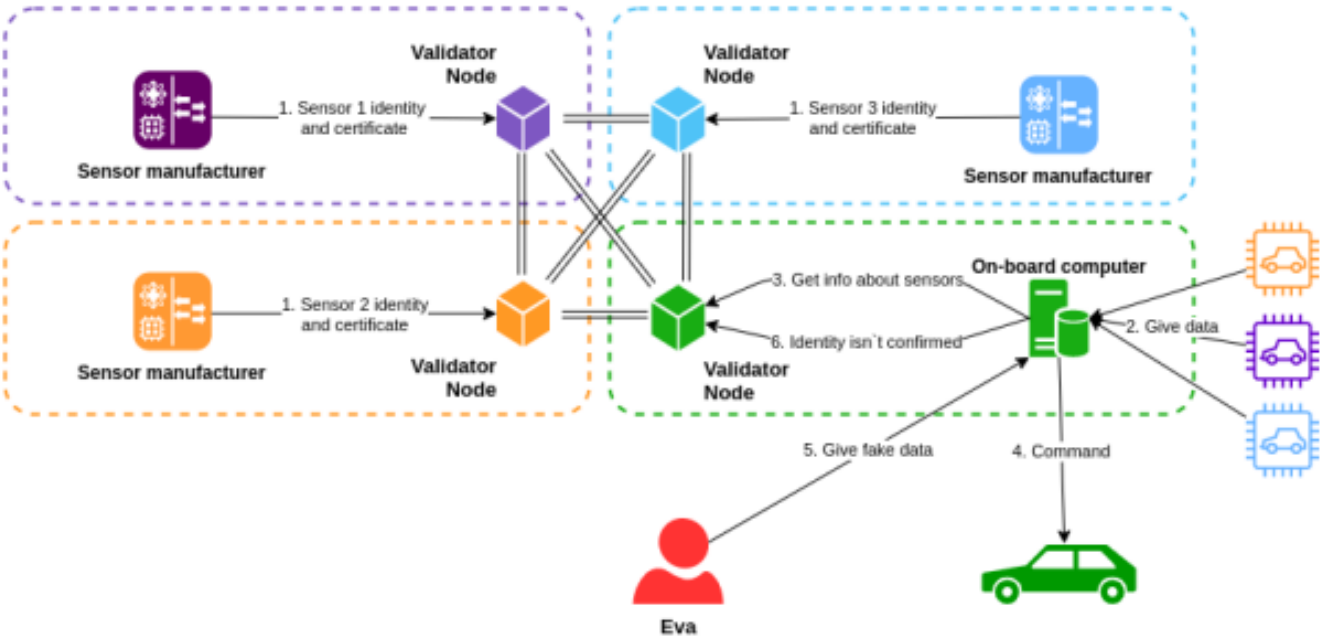

Рисунок 2.5 – Система ідентифікації з використанням датчиків IoT автомобіля

Процедура ідентифікації в даній системі складається з наступних кроків:

- виробники датчиків розміщують ідентифікатори і відкриті ключі своєї продукції в системі ідентифікації;
- датчики передають інформацію з бортового комп'ютера автомобіля (при цьому підписуючи всі повідомлення);
- бортовий комп'ютер звертається до розподіленого реєстру і перевіряє дійсність відкритих ключів датчиків;
- на підставі отриманої інформації бортовий комп'ютер подає команди на механізми автомобіля;

У випадку, якщо користувач (Єва) хоче відправити неправдиву інформацію від імені одного з датчиків, комп'ютер запитує відкритий ключ, який відповідає отриманому ідентифікатором. Оскільки в системі відсутня identity Єви (а якщо і присутня, то до неї прив'язана інше значення відкритого ключа), бортовий комп'ютер не буде обробляти отриманий запит.

Перевагами такої децентралізованої системи є те, що ній взагалі немає такого поняття, як злом або витік даних [17]. Інформація про зареєстровані

імена публічна й доступна всім користувачам мережі. Вкрасти можна тільки саме ім'я, але завдяки криптографічним основам блокчейна це можливо тільки в тому випадку, якщо зловмисник отримає в руки приватний ключ користувача. Оскільки приватний ключ зберігається тільки на стороні користувача, і ключ зашифровано за допомогою його майстер-пароля, ця ситуація рівноцінна компрометації одночасно сховища користувача і його майстер-пароля, а не системи в цілому.

Існує велика кількість різноманітних проектів і стартапів, які працюють з технологією Blockchain. Блокчейн-рішення та напрацювання в цій сфері мають такі IT-монстри як Microsoft (Azure Blockchain as a Service), SAP, Samsung, IBM та інші. Крім них, безліч блокчейн-проектів запускають різні стартапи. При цьому найбільш перспективними з них є ті, що забезпечують ідентифікацію особистості і захист її персональних даних. Тому виникає завдання проведення аналізу платформ і проектів, які володіють необхідним функціоналом для ідентифікації персональних даних (proof of identity) і їх захисту.

3 АНАЛІЗ BLOCKCHAIN ПРОЕКТІВ З ІДЕНТИФІКАЦІЇ ОСОБИСТОСТІ

Для того, щоб ідентифікувати себе в мережі, необхідно надати інформацію про себе, щоб довести, що користувач саме той, за кого себе видає.

Ідентифікація користувачів в фінансовому світі відбувається за рахунок процедури KYC – «Know Your Customer», покликаної відстежувати контрагентів і їх транзакції для боротьби з відмиванням коштів, шахрайством, а також для поліпшення безпеки фінансових операцій. При проходженні процедури KYC користувачі передають частину своєї особистої інформації третій стороні, після чого більше не мають можливості контролювати свої конфіденційні дані.

Відмова від надання особистих даних позбавляє користувача можливості використовувати сервіси або послуги, що вимагають процедури ідентифікації особистості. Дані, що надаються централізованим органам, не можуть бути в безпеці, навіть в компаніях, що забезпечують високий рівень безпеки. Тому, збереження своїх конфіденційних даних під контролем є одним з найважливіших напрямків розвитку на даний момент.

Проведемо аналіз платформ і проектів, які побудовані для ідентифікації персональних даних.

3.1 Аналіз Блокчейн-проектів глобальних корпорацій

3.1.1 Блокчейн-проекти компанії Microsoft

На сьогодні на базі Azure Blockchain as a Service від Microsoft розроблено вже безліч блокчейн-рішень для різних індустрій.

Ще в 2017 році, Microsoft спільно з Accenture і Avanade розробили систему баз даних на основі блокчейн для загального доступу до одних і тих самих даних. А в 2018 році, Microsoft представила DID - систему децентралізованої ідентифікації під назвою ION (Identity Overlay Network), що працює на блокчейне

Біткоіна

(рис. 3.1). Її запуск Microsoft здійснила в травні 2019 року.

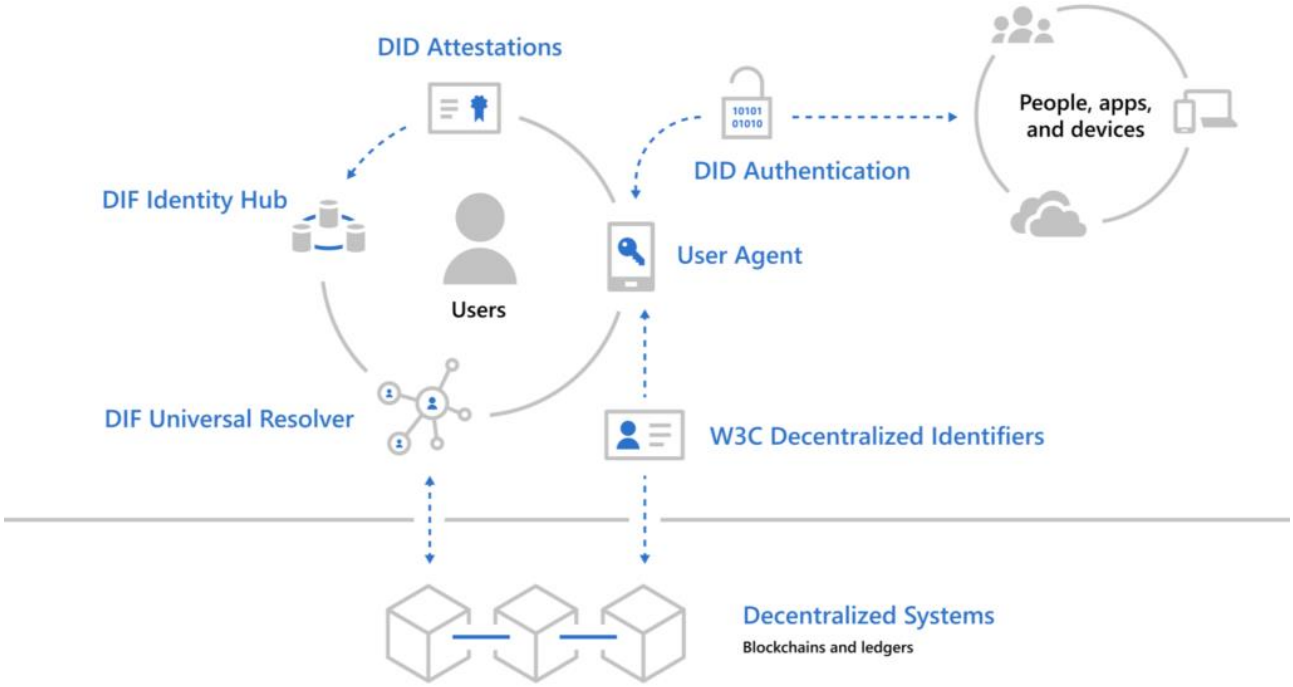

Рисунок 3.1 – Система децентралізованої ідентифікації від Microsoft

ION є мережею другого шару з власною структурою нод (Node, активний електронний пристрій в складі децентралізованої мережі) і відкритим вихідним кодом.

За аналогією з підписанням транзакцій в мережі біткоіна, DID є доказом володіння. Індивідуальні ноди ION відповідають за моніторинг DID і внесення часових міток в блокчейн біткоіна.

Так як керовані ION ідентифікатори децентралізовані, в разі видалення облікового запису користувача доступ до прив'язаних до аккаунту сервісів зберегається.

ION просувається як практична альтернатива автентифікаційній службі Facebook Login для користувачів, що завантажуються в різні додатки на iOS, Android, Windows 10 і інших платформах [18].

Microsoft вважає, що ідентифікатори і публічні криптографічні ключі можуть бути прив'язані до блокчейнів або регістрів різних криптовалюта, таких як Bitcoin або Ethereum.

За інформацією компанії [18], в ION вирішені проблеми недостатньої продуктивності існуючих блокчейн, що перешкоджають створенню глобального сервісу децентралізованої ідентифікації. Вона забезпечує десятки тисяч операцій в секунду і наближається за можливостями масштабування до систем Azure Active Directory, використовуваним співробітниками самої Microsoft для входу в корпоративні мережі.

У проекті також беруть участь виробник апаратних нод Casa, процесинговий сервіс BitPay, криптовалютна біржа Gemini, розробник платіжного рішення Fold і кілька інших компаній.

Поліпшення швидкодії стало можливо при використанні протоколу створення мереж DID - Sidetree, над яким продовжує працювати Microsoft з партнерами. Компанія вважає, що на даному етапі Sidetree ще не готовий для тестування, проте, на малопотужному споживчому обладнанні він вже забезпечує «десятки тисяч DID-операцій в секунду». Граничним завданням розробників протоколу є підтримка децентралізованої ідентифікаційної системою «мільйонів організацій, мільярдів людей і незліченних пристроїв» [18].

На базі блокчейн від Microsoft запускають різні стартапи.

Датська компанія Maerskc спільно з Microsoft, EY і Guardtime створили блокчейн-рішення для забезпечення безпеки і оптимізації страхування в сфері морських вантажоперевезень.

Consensys спільно зі стартапом BlockApps на Azure розробили систему відстеження пересування сировини для однієї з найбільших в світі гірничодобувних компаній BHP Billiton і її постачальників.

Онлайн турагентство Австралії і Нової Зеландії Webjet за підтримки Microsoft створили унікальне блокчейн-рішення, яке може стати новим

галузевим стандартом. Рішення оптимізує процес бронювання та оплати поїздок для користувачів, а також підвищує безпеку і прозорість для Webjet і його партнерів при проведенні транзакцій.

Ізраїльський Bank Hapoalim на базі технології Microsoft створили блокчейн-рішення для спільної з клієнтами роботи з документами. Ця система дозволяє безпечно оновлювати інформацію без особистої присутності користувача в банку.

Можливості Azure активно використовуються Фінтех консорціумом Тайвані і Центральним банком Сінгапуру - Monetary Authority of Singapore (MAS). Більш того, MAS в партнерстві з R3, Ledger Technology і консорціумом фінансових установ (11 фінансових інститутів і 5 технологічних компаній) розробили прототипи програмного забезпечення трьох різних моделей для децентралізованих міжбанківських платежів і розрахунків. Ці рішення були створені в Azure Blockchain as a Service при безпосередній технологічної підтримки Microsoft.

3.1.2 Блокчейн-проекти IBM

Компанія IBM практично відразу після виходу блокчейна на ринок Фінтех-послуг усвідомила перспективність цієї технології і зайнялася вивченням її можливостей. У компанії працює близько 1,5 тис. фахівців у сфері блокчейна. У березні 2018 року IBM повідомила про реалізацію понад 400 блокчейн-проектів в самих різних галузях, в т.ч. в ритейлі, сфері фінансових послуг, транспорті, державному секторі, охороні здоров'я, ЗМІ, логістиці.

Так, спільно з AIG IBM працює над створенням «розумного» страхового договору на базі блокчейна для управління міжнародним страховим покриттям.

Через блокчейн-платформу we.trade 9 європейських банків, в т.ч. Rabobank, HSBC, Deutsche Bank, Societe Generale, Natixis, KBS, UniCredit і ін., працюють з IBM для полегшенням внутрішньої та транскордонної торгівлі малих і середніх компаній.

Компанія Sony розробила систему зберігання даних про освіту на базі хмарного сервісу захищеної блокчейн-мережі IBM і блокчейн-платформи Hyperledger Fabric. Нова система повинна допомогти боротися з шахрайством при проходженні співбесід.

Виробники продуктів харчування і ритейлери, серед яких Dole, Driscoll's, Golden State Foods, Kroger, McCormick and Company, McLane Company, Nestlé, Tyson Foods, Unilever і Walmart, використовують блокчейн-рішення IBM для контролю за ланцюжком поставок продуктів харчування. Це дозволяє швидше відстежувати різного роду проблеми і знаходити їх джерела.

Над створенням торгової блокчейн-платформи для глобальної логістичної індустрії IBM працює разом з оператором контейнерних перевезень Maersk. Як очікується, нове рішення дозволить використовувати цифрові ланцюжки поставок і відслідковувати рух вантажу по всьому світу в режимі реального часу.

Створення блокчейн-рішення для торгівлі нафтою IBM анонсувала спільно з компаніями Natixis і Trafigura. Воно було створено на замовлення банку Natixis, який веде бізнес, пов'язаний з сирою нафтою.

Спільно з SecureKey IBM розробляє систему цифрової ідентифікації на базі блокчейн. Вона повинна прискорити і поліпшити процес розпізнавання клієнтів при доступі до таких послуг, як відкриття банківського рахунку, отримання водійських прав і оплата комунальних послуг.

3.1.3 Блокчейн-проект Samsung

У квітні 2018 року Samsung Electronics анонсувала впровадження блокчейн-системи для управління гігантською глобальною мережею поставок, яка є у компанії.

Логістичне блокчейн-рішення для Samsung розробила дочірня компанія - Samsung SDS. Samsung вирішила використовувати блокчейн для скорочення

часу і підвищення ефективності процесів запуску продуктів і їх поставок кінцевим користувачам.

Передбачається, що нововведення зменшить витрати на доставку товару на 20%.

3.2 Blockchain системи з ідентифікації особистості

3.2.1 Система Civic

Система Civic - blockchain-платформа, що дозволяє управляти ідентичністю в онлайн просторі. Користувачі можуть реєструвати, підтверджувати, а також приховувати свої особисті дані для запобігання шахрайських дій [19].

Civic є децентралізованою екосистемою ідентифікації на основі Ethereum, яка надає можливість перевіряти особистість користувача за запитом. Використовуючи платформу Civic, користувачі можуть створювати власну цифрову особистість і зберігати особисту інформацію про неї на пристрої. Досягається така можливість за рахунок технології блокчейн, що робить такий підхід безпечним і зручним.

В екосистемі сервісу Civic головна роль відводиться надійним постачальникам інформації, необхідної для посвідчення особи. До них відносяться банки, організації комунальної та соціальної сфери. Ці структури повинні перевіряти справжність відомостей про фізичну або юридичну особу, потім підтверджуючи їх внесенням в блокчейн.

Всякий раз клієнт мережі буде отримувати повідомлення, коли його інформація використовується мережею. У сукупності ж це призводить до створення децентралізованої і надійної екосистеми, що служить для перевірки автентичності.

Платформа Civic надає можливість проходження KYC один раз, з подальшим наданням необхідних даних на інших платформах. Клієнтом може

стати будь-який учасник мережі, що заніс свої дані в блокчейн за допомогою додатка Secure Identity. Перевірку особистості здійснюють валідатори, які надають дані постачальникам послуг для верифікації користувачів, отримуючи за це винагороду в токенах криптовалюти CVC. Смарт-контракти використовуються для контролю фінансових операцій і атестації.

Структура Civic (CVC) пропонує послугу миттєвої перевірки цифрової ідентифікації на вимогу і за зниженою ціною.

Архітектура мережі має три різних об'єкта (рис. 3.2):

- користувач (має особисту інформацію);

заявник ідентичності;

валідатор (установа, банк, держава, компанія, постачальник послуг).

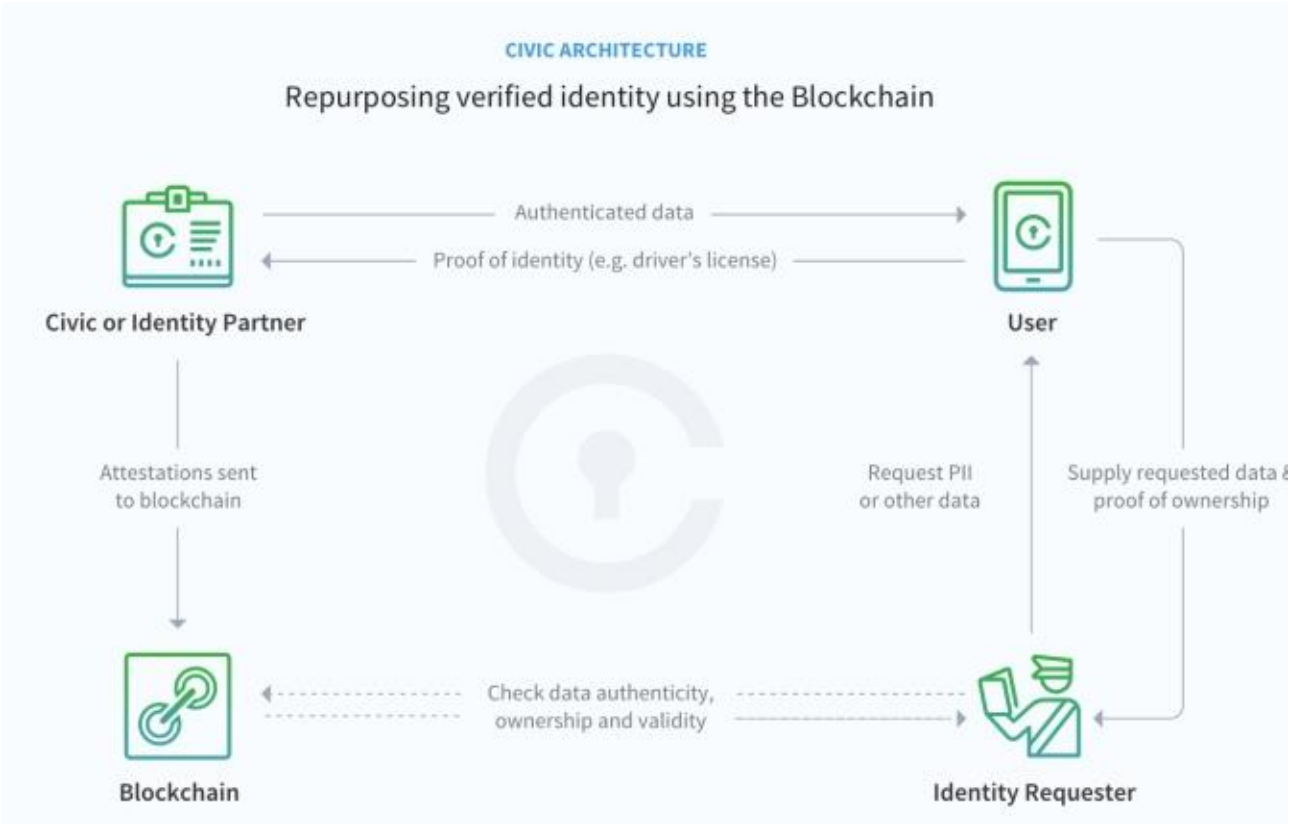

Рисунок 3.2 – Архітектура системи Civic

Основна ідея Civic полягає в створенні єдиного цифрового посвідчення для кожного користувача з можливістю отримання доступу до різних служб,

сайтів, сервісів і організацій. Позитивною стороною такого рішення є повний контроль над наданими даними з боку користувача. Дані не зберігаються в блокчейн, замість цього в розподіленому реєстрі зберігаються посилання з доступом до атестацій. Це означає, що зареєстрований в мережі ідентифікатор і є доказом його справжності.

Щоб позбавити користувачів від частого введення логінів і паролів, які до того ж можна зламати, ідентифікація в Civic відбувається з надійними біометричними даними: відбитками пальців, 3D-розпізнаванням особи, голосу, серцевого ритму. Біометричні дані також не збираються централізовано і зберігаються на пристрої користувача в зашифрованому вигляді.

Доступ до мережі відкритий для всіх, і створення призначених для користувача інтерфейсів є безкоштовним. CVC - це криптовалюта, пов'язана з мережею; вона забезпечує економічну життєздатність екосистеми і регулює внутрішні обміни протоколом.

Система Civic пропонує мобільний додаток для смартфонів IOS і Android.

Недоліками системи Civic є

- слаборозвинених програмну інфраструктуру;
- недостатнє зростання кількості партнерів;
- залежність активності використання токену CVC від спекулятивних факторів.

3.2.2 Система ідентифікації uPort

uPort є спеціалізованою системою, призначеною для ідентифікації та автентифікації на базі Blockchain. uPort спільно з Microsoft створюють захищену і доступну у використанні широкому колу осіб систему для автономної суверенної ідентичності на основі Ethereum для різних аспектів життя людини (банківська діяльність, акаунти в соціальних мережах, професійна діяльність). Особистість можна буде використовувати як єдиний

аккаунт для автентифікації, підписувати нею заяви, виставляти напоказ свій публічний профіль.

Система побудована на основі розумних контрактів на платформі Ethereum. Це дає творцям багаті можливості по використанню вже існуючого коду, вбудованого в платформу.

Система uPort має три основних компоненти [20]:

1. Мобільного клієнтського додатка, що зберігає приватний ключ користувача.

2. Бібліотек, за допомогою яких розробники можуть підключити автентифікацію до свого сайту.

3. Смарт-контрактів Ethereum, що містять логіку програми.

У найзагальнішому вигляді, ідентифікатор, він же цифровий паспорт uPort, являє собою адресу (відкритий ключ) в блокчейне Ethereum. uPort - цифрове представлення особистості (а також додатку, організації, гаджета або бота), здатне робити підтверджені заяви при взаємодії зі смарт-контрактами та іншими ідентифікаторами uPort, як в блокчейне, так і офф-чейн.

Головними недолікамм системи uPort є

- частково закритий код мобільних додатків uPort, що не дає повністю зрозуміти механізм роботи;
- достатньо нестабільна робота системи.

Перевагами системи uPort є:

– універсальність;

– анонімність;

– відсутність довіри до третьої сторони;

– зручність для користувача. З усіх систем ідентифікації на основі blockchain uPort має найбільш зручний мобільний додаток-клієнт.

3.2.3 Система ідентифікації emcSSL

Інфраструктура emcSSL базується на блокчейні криптовалюти EmerCoin, використовуючи останній як децентралізоване довірене сховище геш-сум клієнтських SSL-сертифікатів. Самі сертифікати можуть бути згенеровані клієнтами локально, без участі будь-яких зовнішніх сервісів авторизації, і швидко замінені у міру необхідності, що робить ефективними як планову заміну, так і швидкий відзив скомпрометованих сертифікатів [21].

Також запропонована система InfoCard - децентралізованих розподілених «візитних карток», з можливістю організації інформації в ієрархічну структуру, що може бути корисним для швидкого оновлення вмісту карток членів компаній або інших організацій.

Спільне використання запропонованих сервісів дозволяє швидко створювати і оновлювати облікові записи, а також мати безпарольний логін і захищене з'єднання з необмеженим безліччю серверів.

Новизна пропозиції полягає в повній децентралізації системи, тобто відсутності якоїсь групи серверів авторизації під єдиним управлінням, що має місце в системах Kerberos, OpenID і їм подібним.

EmerCoin має розподілене сховище загального призначення з відкритим інтерфейсом - EmerCoin NVS. Це дозволяє зберігати в блокчейне інформацію будь-якого роду.

Одним із застосувань такого блокчейна є випуск SSL-сертифікатів. На них і будується ідентифікація користувачів за допомогою EmerCoin. На відміну від Blockstack і uPort, які розробляють власні клієнтські протоколи автентифікації, написані на Javascript і запущені в браузері, EmerCoin використовують існуючу технологію автентифікації за допомогою клієнтських SSL-сертифікатів.

Різниця між звичайними клієнтськими сертифікатами та сертифікатами emcSSL в тому, що в якості центру сертифікації виступає не деяка третя довірена сторона, а сам блокчейн. Головний секрет SSL – приватний ключ

центру сертифікації - в emcSSL є відкритим, і валідація сертифіката здійснюється не через підпис від CA, а через геш в блокчейне.

Така система не розкриває секрет користувача в процесі автентифікації сервера і використовує децентралізоване зберігання облікових записів. Як і у Blockstack і uPort, у emcSSL є можливість прикріпити до імені довільну інформацію у вигляді профілю [21].

Так як в системі emcSSL відсутній центр сертифікації, а випуском сертифікатів займаються самі користувачі, то випуск сертифіката безкоштовний. Блокчейн EmerCoin виступає тільки як публічне довірене сховище гешів SSL-сертифікатів і забезпечує унікальність Serial, який і є унікальним UserID.

Таким чином, в системі emcSSL успішно вирішені обидві проблеми - як нерозголошення таємниці, так і децентралізації, що дозволяє масштабувати систему до загальносвітового рівня. Відповідно, в системі неможливі атаки, які призводять до масової компрометації облікових записів, бо приватні ключі генеруються користувачами, і ніколи не покидають їх комп'ютерів, і просто не існує такого центрального місця, яке можна скомпрометувати.

В особливо важливих випадках доцільно використання emSSL спільно з паролем в рамках двухфакторной авторизації. При цьому emcSSL авторизує пристрій і забезпечує безпечний канал зв'язку з сервером, а пароль авторизує оператора.

Також блокчейн-архітектура emcSSL ефективно і безпечно вирішує проблему відкликання скомпрометованого сертифіката і його швидкої заміни, чим вигідно відрізняється від CRL і протоколу OCSP, уразливого до атаки MItM.

Як було зазначено вище, система клієнтських SSL-сертифікатів вирішує проблему повного розкриття секрету. Однак по ряду причин, вона не дуже добре масштабується, і широкого поширення не отримала.

Переваги системи emcSSL:

– універсальність;

– відкритість;

– анонімність;

– використання існуючого протоколу (автентифікація через клієнтський SSL-сертифікат). Це серйозно полегшує життя розробникам сайтів, на яких буде використовуватися така модель автентифікації;

– відсутність довіри до третьої сторони.

Недоліком системи emcSSL є спрямованість на специфічну цільову аудиторію технічних фахівців. На даний час у системи немає достатньо зручного способу для кінцевих користувачів зареєструвати ім'я і використовувати його як засіб автентифікації. Користувачі змушені пройти процедуру генерації сертифіката і завантаження його в браузер [21], і хоча розробники надають докладну інструкцію для здійснення цих дій, вона поки ще спрямована на фахівців в області криптографії.

3.2.4 Децентралізована система 3Box

3Box - це децентралізована система зберігання призначених для користувача даних нового покоління. Вона дозволяє розробникам виконувати різні операції з ідентифікації користувачів і облікових записів, на зразок отримання ідентифікатора (DID) користувача, зв'язування нових адрес з DID і додавання нових методів автентифікації [22].

Кожен обліковий запис 3Box має унікальний DID, званий 3ID, що дозволяє користувачам управляти своїми даними та інформацією в децентралізованих мережах. Для створення або відновлення облікового запису 3Box користувачі проходять автентифікацію, підписуючи повідомлення своєю парою ключів. Якщо додатки або служби хочуть взаємодіяти з обліковим записом користувача будь-яким іншим способом, окрім читання загальнодоступних даних, таких як запис або видалення даних, їм потрібно буде запросити, щоб користувач підписав повідомлення про згоду своїми

перевіреними ключами, оскільки тільки ключі власника облікового запису можуть управляти децентралізованої ідентифікацією.

Завдяки своїй децентралізованої структурі 3Box дає можливість розробникам усунути значну частину відповідальності, пов'язаної із забезпеченням безпеки і захистом призначених для користувача даних. Дані зберігаються у користувачів, що дає їм більше контролю над своїми конфіденційними даними, а також над їх передачею і використанням в інших додатках. Можливість зберігання даних у користувачів дозволяє їм легко використовувати ці дані в інших додатках, службах або мережах без необхідності створення нових даних.

3.2.5 Децентралізована система Blockpass

Додаток Blockpass для ідентифікації особи, за допомогою якого користувачі можуть створювати і зберігати цифрові посвідчення, дозволяє користувачам створювати цифрові ідентифікатори з можливістю підключення до банків, бірж, торгових платформ та інших регульованим сервісів [22].

Перевірку документів здійснюють сторонні особи, але лише за умови їх надання користувачем. Обираючи послуги продавця, що вимагає підтвердження особистості, користувач сам вирішує: надавати продавцю свої дані або відмовитися від його послуг.

Система Blockpass має в своєму складі протоколи Know Your Device (KYD) і Know Your Object (KYO) для перевірки ідентичності в «інтернеті речей» і автономного протоколу ідентифікації особистості для «інтернету всього» (IoE).

3.2.6 Система ідентифікації Bloom

Bloom - це рішення для створення безпечної ідентифікації особистості і кредитного рейтингу на основі технології блокчейн. Протокол складається з трьох основних компонентів (рис. 3.3):

- BloomID (Identity Attestation), який забезпечує глобальну безпечну ідентифікацію, дозволяючи кредиторам пропонувати відповідні кредити по всьому світу;
- BloomIQ (Credit Registry), який є системою звітності та відстеження поточних і минулих боргових зобов'язань, прив'язаних до BloomID користувача;
- BloomScore, що є показником кредитоспроможності споживачів. Така децентралізована оцінка аналогічна оцінці системи рейтингу позичальників FICO або VantageScore.

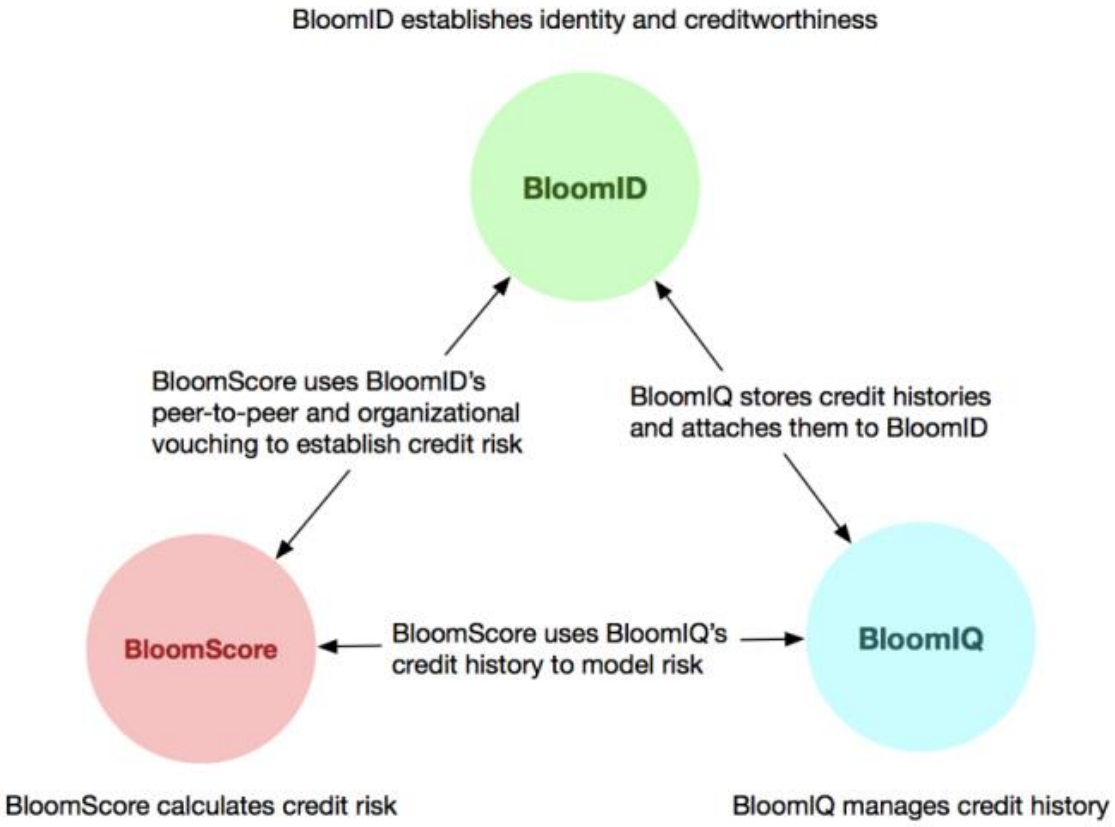

Рисунок 3.3 – Архітектура системи Bloom

Bloom дозволяє користувачам створювати кредитний портфель, доступний в будь-якій частині світу, а також забезпечує децентралізований підхід до побудови кредитної індустрії на блокчейні з підвищеною безпекою і зручністю.

Бізнес-модель Bloom була оголошена в серпні 2017 року. У неї є партнерські відносини з іншими проектами кредитування блокчейнів, такими як Ethland і Lendoit.

Але на даний час команда Bloom не розробила жодних доказів концепції - бета-версії, тому по суті цей проект все ще знаходиться на стадії паперу.

Для контрольних точок, зазначених в плані створення та розвитку Bloom, немає розрахункового періоду завершення, тому невідомо, коли протокол буде завершено і готовий до розгортання.

3.2.7 Система ідентифікації Blockstack

Нью-йоркский блокчейн-стартап Blockstack є системою безсерверних веб-додатків, яка дозволяє створювати односторінкові сайти, безкоштовно викладати їх в блокчейн замість традиційного хостингу. В процесі була розроблена концепція Blockchain ID, за допомогою якої можна створити в блокчейне публічний профіль і за допомогою нього автентифікувати себе [17].

Хоча основний упор система Blockstack концентрує на додатках, в яких користувачі є власниками своїх даних (наприклад Airtext, BentenSound, ImageOptimizer або Graphite), у blockstack також є філософія малого використання блокчейна у випадках коли це абсолютно необхідно. Їх основний аргумент в тому, що блокчейн повільний і дорогий, а значить повинен використовуватися тільки для одиночних або нечастих операцій. Решта взаємодії з додатками має відбуватися через peer-to-peer, тобто користувачі децентралізованих програм слід ділитися даними безпосередньо один з одним, а не через блокчейн (рис. 3.4).

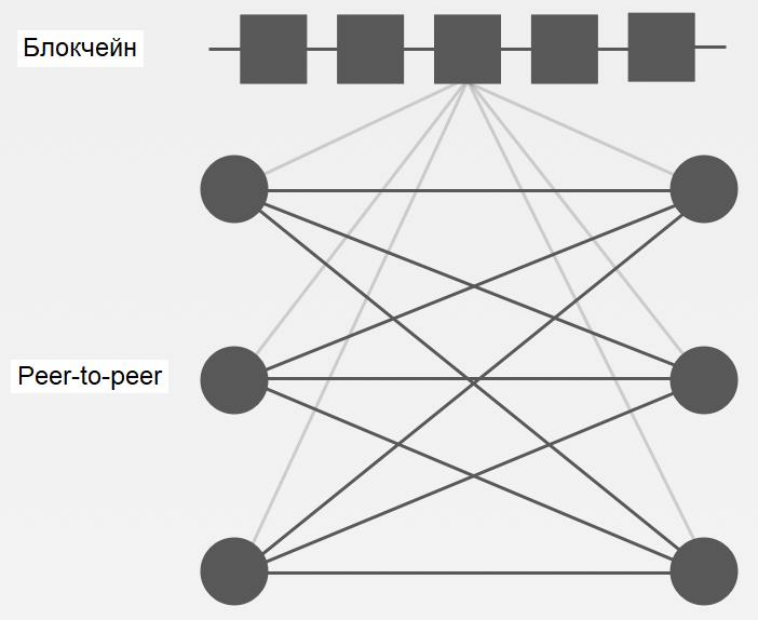

Рисунок 3.4 – Взаємодія з додатками в Blockstack

У випадку з Blockstack, в блокчейне зберігається тільки ідентифікаційна інформація користувача. Інформація про те, як отримати дані кожного користувача, зберігається в файлах зони (zone files) і поширюється через пірінгову мережу за допомогою нод. Достовірність даних, які віддають Ноди, можна перевірити, а саме їх справжність, порівнявши їх з гешами, які зберігаються в блокчейне і у інших користувачів.

У спрощеній версії системи, справжність і цілісність всіх даних, які отримуються, можна перевірити використовуючи публічні ключі і геш-кодування, що зберігаються в віртуальному ланцюгу Blockstack.

В другому випадку архітектура складніше, ніж перший підхід, і вимагає більш комплексної інфраструктури. Blockstack надає всі необхідні компоненти для створення такої децентралізованої системи (рис. 3.5).

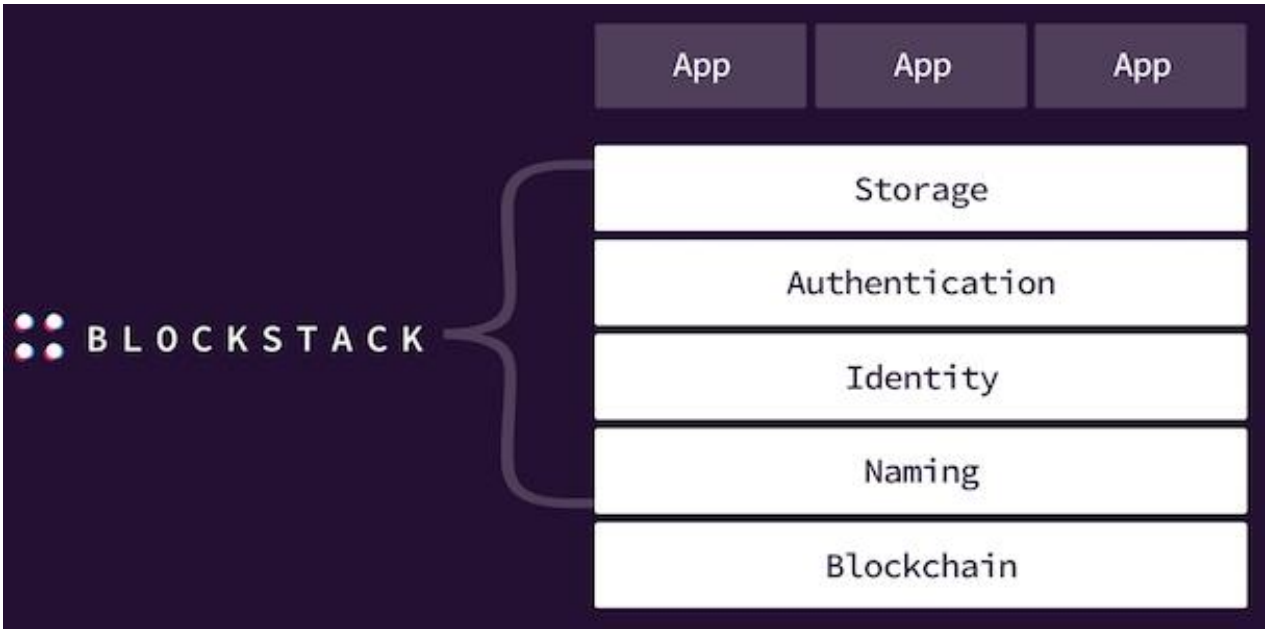

Рисунок 3.5 – Архітектура Blockstack

При такій архітектурі, в блокчейне зберігаються тільки дані, які дійсно повинні бути розподіленими і не перезаписуваними. У випадку з Blockstack, транзакції в блокчейне потрібні тільки, щоб зареєструватися і вказати, де повинні зберігатися дані. Може знадобитися більше транзакцій, якщо користувач захоче змінити щось з інформації, але подію, що не повторюється.

Більш того, логіка програми, на противагу до першого підходу, працює на стороні клієнта, а не на смарт-контрактах. Це дозволяє розробнику змінювати цю логіку без дорогих або іноді навіть неможливих оновлень смарт-контракту. А тримаючи дані і логіку програми не в блокчейне, децентралізовані додатки можуть досягти рівня продуктивності і масштабованості традиційних централізованих систем.

Система Blockstack використовує вже існуючу інфраструктуру Bitcoin. Кожна транзакція Bitcoin може містити опціональне поле OP_RETURN, в якому зберігається довільна інформація. Завдяки такій можливості, Blockstack будує на основі блокчейна Bitcoin віртуальний блокчейн (virtualchain), який складається тільки з транзакцій з непорожніми полями OP_RETURN, що

містять повідомлення потрібного формату. Такі транзакції можуть передавати інформацію про наступні операції:

– реєстрація імені, в ході операції ім'я прив'язується до біткоін-адреси, яка його зареєструвала;

– оновлення профілю. У профілі можуть бути абсолютно довільні поля;

– передачі імені іншою адресою. Таким чином, віртуальний блокчейн Blockstack грає роль, схожу на роль DNS.

Переваги системи Blockstack:

- універсальність. Зареєструвавши ім'я в системі Blockstack, користувач може бути ідентифікований під цим ім'ям на будь-яких сайтах, на яких працює протокол автентифікації Blockstack;

- відкритість. Розробка системи ведеться за принципами open source;

- анонімність. Інформація про те, коли і куди автентифікується користувач, доступна тільки користувачеві і сайту, на якому він автентифікується;

- відсутність довіри до третьої сторони. Система Blockstack працює на базі розвиненого блокчейна Bitcoin, тобто:

1) систему практично неможливо зламати: в такій системі взагалі немає такого поняття, як злом або витік даних. Інформація про зареєстровані іменах і так публічна й доступна всім. Вкрасти можна тільки саме ім'я, але завдяки криптографічним основам блокчейна це можливо тільки в тому випадку, якщо зловмисник отримає в руки приватний ключ користувача. Оскільки приватний ключ зберігається тільки на стороні користувача, і ключ зашифровано за допомогою його майстер-пароля, ця ситуація рівноцінна компрометації одночасно сховища користувача і його майстер-пароля, а не системи в цілому;

2) в системі немає єдиної точки відмови. Система може вийти з ладу тільки якщо вийдуть з ладу всі вузли блокчейна Bitcoin;

3) ніхто не може скористатися ім'ям користувача крім нього самого.

4) ніхто не може конфіскувати або заблокувати особистість користувача. Всі дії в блокчейне приймаються на основі консенсусу декількох вузлів. Конфіскація імені користувача можлива тільки при схваленні значного числа вузлів мережі, що рівноцінно компрометації декількох тисяч серверів по всьому світу.

Недоліки системи Blockstack:

– новизна. Протокол все ще перебуває в стадії розробки, тому система може працює нестабільно, з помилками, відсутня важлива функціональність;

– незручність для користувача. Хоча розробники прагнуть виправити цю проблему, прямо зараз робота з протоколом досить незручна. Користувачеві необхідно завантажити, запустити і налаштувати локально додаток-клієнт, який користувачі Windows і Linux змушені завантажувати і налаштовувати частини системи вручну. Користувачеві також необхідно замовити собі хоча б одне ім'я; в силу особливостей протоколу, система вимагає від користувача невелику плату за кожну операцію з віртуальним блокчейном, в тому числі за реєстрацію імені. Це означає, що користувачеві треба купити біткоіни і передати їх на свою адресу. Немає повноцінних бібліотек для вбудовування автентифікації на веб-сайті.

Таким чином, децентралізована ідентифікація надає можливість використовувати єдиний цифровий ідентифікатор на різних майданчиках та сервіси, без необхідності повторного занесення даних. Блокчейн гарантує безпеку і незмінність цих даних, а користувачі отримують повний контроль над своєю особистою інформацією. Результати проведеного аналізу систем децентралізованої ідентифікації показали, що в якості системи для реалізації можливо обрати протокол Blockstack.

4 РЕАЛІЗАЦІЯ ПРОТОКОЛУ ІДЕНТИФІКАЦІЇ BLOCKSTACK

Система Blockstack виконує наступні цілі:

1. Забезпечує безпечні і розподілені методи зберігання, реєстрації та передачі імен користувачів.

2. Забезпечує локальне зберігання приватного ключа користувача, з допомогою якого той може довести своє володіння ім'ям.

3. Надає можливість користувачеві підписувати своїм приватним ключем запити на автентифікацію і таким чином заходити під своїм ім'ям на сайти, що підтримують даний протокол.

4. Містить бібліотеки, за допомогою яких розробники можуть легко налаштувати автентифікацію користувачів на своєму веб-сайті.

Система складається Blockstack з наступних компонентів:

1. Інфраструктура Blockstack:
    a. Virtualchain
    b. Blockstack Core
    c. Blockstack Portal
    d. Onename

2. Бібліотека на Python, за допомогою якої можна створювати і підписувати Authentication Request і Authentication Response об'єкти.

3. Додаток для фреймворка Django, що реалізує автентифікацію з використанням протоколу Blockstack на стороні сервера.

4.1 Інфраструктура Blockstack

Blockstack - це комплекс програмного забезпечення, що знаходиться в розробці, кінцева мета якої - створення нового децентралізованого Інтернету. Цей стартап співпрацює з Microsoft [23].

Метою компанії є побудова в блокчейне окремої системи імен, схожою на DNS. Це дозволяє перевести такий важливий і фундаментальний рівень всесвітньої павутини, як система імен, в децентралізовану область. Якщо перевести все імена в блокчейн, то будуть усунути сторони, яким необхідно довіряти, такі як кореневі DNS-вузли і Identity Providers. Імена Blockstack можна використовувати як для адресації веб-сайтів, так і для автентифікації користувачів. Фактично, ім'я в блокчейне може бути як ім'ям сайту, так і ім'ям користувача, або і тим, і іншим.

Додатки Blockstack - додатки без сервера, односторінкові веб-сайти, що складаються з статичних файлів (HTML, CSS, Javascript). Ці статичні файли розміщуються в особистому хмарному сховищі розробника (Google Drive, Dropbox і т.п.) [24].

Інфраструктура Blockstack розбивається на кілька шарів (рис. 4.1) [25].

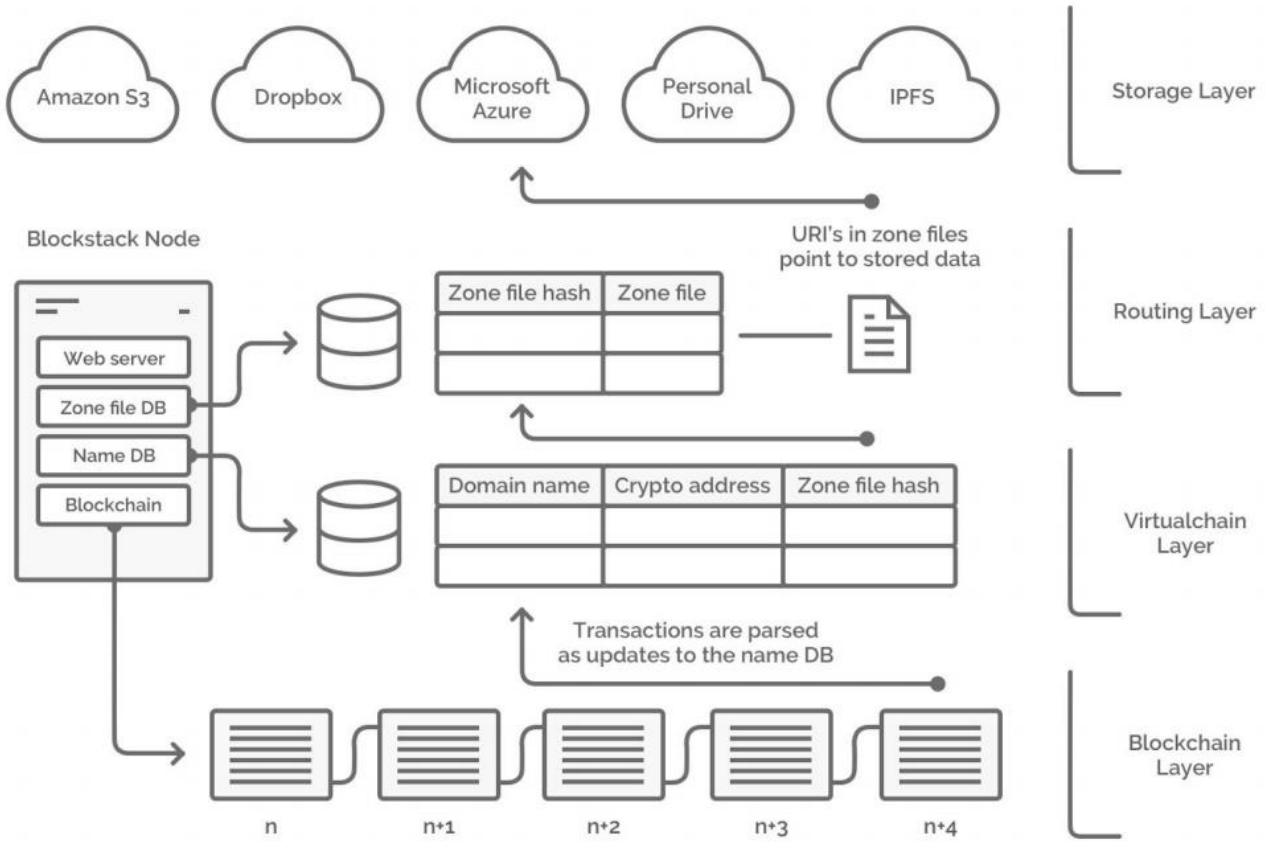

Рисунок 4.1 – Інфраструктура Blockstack

Ключовою технологією інфраструктури Blockstack є блокчейн Bitcoin, на якому будується система альтернативних кореневих DNS-серверів.

Blockchain Layer інфраструктури Blockstack - це блокчейн Bitcoin; завдяки сильній криптографічній основі, децентралізованій природі та механізмам консенсусу він забезпечує розподіленість даних, безпеку і той факт, що у користувача не можна відібрати ім'я без компрометації його приватного ключа.

Virtualchain Layer - надбудова над блокчейном, яка дозволяє зберігати в блокчейне інформацію, пов'язану з протоколом Blockstack: імена, їх власників, операції над іменами, інформацію про імена.

До кожного імені додається Zone File в такому ж форматі, в якому він використовується в DNS. Він містить посилання на інформацію про ім’я, наприклад, профіль (якщо це ім'я користувача) або статичні файли [26, 27]. Zone Files складають Routing Layer.

На трьох перших рівнях працює Blockstack Node. Це, по-перше, звичайний майнер Bitcoin, тобто Bitcoin Full Node, який індексує блокчейн. По-друге, такий сервер запускає у себе Blockstack Core: основну бібліотеку для роботи з Blockstack. Ця бібліотека запускається як сервер, що індексує Virtualchain і розпізнає Zone File.

Посилання в Zone File ведуть на деяке віддалене сховище, яке може містити як легковажні текстові описи профілів, так і великовагові статичні файли. Цей останній шар називається Storage Layer.

Blockstack Portal - користувальницький додаток, який відповідає за доступ користувачів до Blockstack-сайтам і автентифікацію.

На даний момент ще не в повній мірі працює частина системи, що дозволяє користувачу по імені сайту отримати до нього доступ, тобто завантажити статичні файли по шляху в Zone File. Однак зараз можна випробувати частину системи, яка відповідає за індивідуальні аккаунти і їх автентифікацію на будь-яких сайтах. Пропонується використати цю можливість

для створення системи ідентифікації та автентифікації в застосуванні до звичайних серверних сайтів [28].

4.1.1 Virtualchain

Розробники Blockstack ввели нове поняття віртуального блокчейна. Це шар програмних компонентів поверх звичайного блокчейна, який вводить нову функціональність без змін блокчейна, що лежить в основі. Всі нові операції вводяться на рівні віртуального блокчейна і кодуються в мета-дані транзакцій звичайного блокчейна. І хоча звичайні вузли блокчейна теж бачать цю інформацію, логічне значення вона має тільки для вузлів віртуального блокчейна [29].

Прикладами операцій є передзамовлення імені, реєстрація імені, зміна власника, відгук імені, продовження реєстрації імені та оновлення Zone File.

В якості блокчейна, на якому працює система імен Blockstack, був обраний блокчейн Bitcoin як найбільш розвинений блокчейн мережі. Це пов'язано з тим, що блокчейни з невеликою кількістю майнерів, сильніше схильні до ризику атаки, так як зловмиснику необхідні менші обчислювальні потужності в порівнянні з великим блокчейном.

Крім того, маленькі блокчейни (такі як Namecoin, який так само використовується в якості системи альтернативних DNS-серверів) уразливі перед атакою 51%, коли один майнер або група майнерів в змові контролюють більше 51% обчислювальної потужності блокчейна, що веде до захоплення контролю над блокчейном однією стороною [30].

Інформація про операції Blockstack вбудовується через поле OP_RETURN транзакцій Bitcoin. Згідно сайту OP_RETURN Stats, що збирає інформацію про кастомні протоколи на основі блокчейна Bitcoin, протокол Blockstack є найбільшим за обсягом транзакцій серед нефінансових додатків [31].

4.1.2 Blockstack Core

Blockstack Core - це основний CLI-додаток, написаний на Python, що здійснює значну частину роботи з протоколом, здійснюваної на сервері.

Blockstack Core включає в себе:

1. Blockstack API. Надає REST API для отримання інформації про інфраструктуру Blockstack: інформація про імена, гаманці, профілі, вузли і т.п. Ця програма необхідна для користувача, так як API використовується в Blockstack Portal. Blockstack Core, запущений в режимі API, не перетворює локальну машину користувача в повний вузол Blockstack і таким чином не завантажує локально весь блокчейн. Замість цього він покладається на повний вузол, доступний за адресою node.blockstack.org.

2. Blockstack Client. Консольна утиліта, за допомогою якої користувач може керувати своїм гаманцем і прив'язаними до нього іменами. До консольного клієнту прив'язані два Bitcoin-гаманця: owner і payment. Owner-гаманець володіє іменами, саме його адреса вказується при передачі імені аккаунта користувача. На payment-адреса нараховуються біткоіни, щоб за допомогою них можна було платити невелику технічну плату за кожну транзакцію (реєстрація імені, оновлення Zone File, передача імені).

3. Blockstack Registrar. Бібліотека для Identity Provider'ов, за допомогою якої можна легко реєструвати нові імена, заповнювати профілі користувачів і відправляти ці імена на їх особисті адреси.

4. Blockstack Core. Повний вузол Blockstack, що індексує транзакції блокчейна Bitcoin і віртуального блокчейна.

4.1.3 Blockstack Portal

Blockstack Portal - додаток-сервер, що запускається локально на комп'ютері користувача (рис. 4.2) [32, 33]. За допомогою нього користувач може:

1. Завести собі новий owner-гаманець.

2. Зайти зі старого owner-гаманця за допомогою backup phrase.

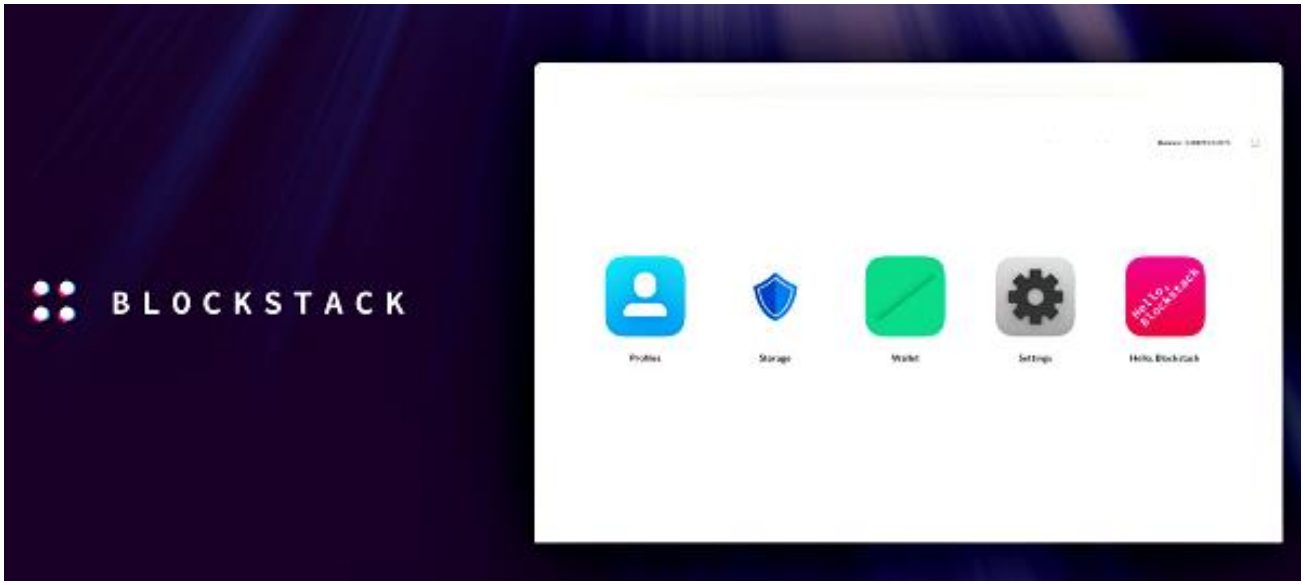

Рисунок 4.2 – Домашня сторінка Blockstack Portal

3. Придбати собі ім'я.

4. Заповнити профіль для цього імені.

5. Приймати запити на автентифікацію, ініціалізацію на веб-сайтах.

Щоб запустити у себе Blockstack Portal, користувачеві необхідно:

1. Встановити Blockstack Core.

2. Налаштувати Blockstack Core, створивши пару owner- і payment-гаманців.

Через особливості програмного забезпечення Blockstack на даний момент owner-гаманець зі Blockstack Core і owner-гаманець зі Blockstack Portal несумісні, при налаштуванні Blockstack Portal користувачеві доведеться згенерувати новий owner-гаманець.

3. Запустити Blockstack Core в режимі API, підключившись до повного вузлу Blockstack.

4. Завантажити Blockstack Portal.

5. Запустити CORS-проксі.

6. Запустити Blockstack Portal.

7. Завести в Blockstack Portal новий акаунт (тобто згенерувати owner-гаманець) або ввести backup phrase з 24 слів, відновивши доступ до старого owner-гаманцю.

8. Включити в налаштуваннях Blockstack Portal доступ до протоколу автентифікації.

Перш, ніж автентифікуватись в будь-якому додатку, користувач повинен придбати собі ім'я. Зробити це можна одним з наступних способів:

1. Завести біткоіни на свій payment-гаманець, через інтерфейс Blockstack Portal замовити собі ім'я.

2. Придбати ім'я іншим способом (через сайт onename.com або через консольний клієнт Blockstack Core) і передати ім'я на owner-гаманець в Blockstack Portal.

На даний момент процес установки клієнта і придбання імені в системі Blockstack досить незручний для кінцевого користувача. В майбутньому розробники планують значно його доопрацювати.

Після придбання імені процес автентифікації проходить для користувача тривіальним чином:

1. З запущеним клієнтом він заходить на сайт програми та намагається автентифікуватися (рис. 4.3).

Рисунок 4.3 – Домашня сторінка веб-сайту "Hello, Blockstack"

2. Додаток генерує запит на автентифікацію і перенаправляє користувача в Blockstack Portal (рис. 4.4).

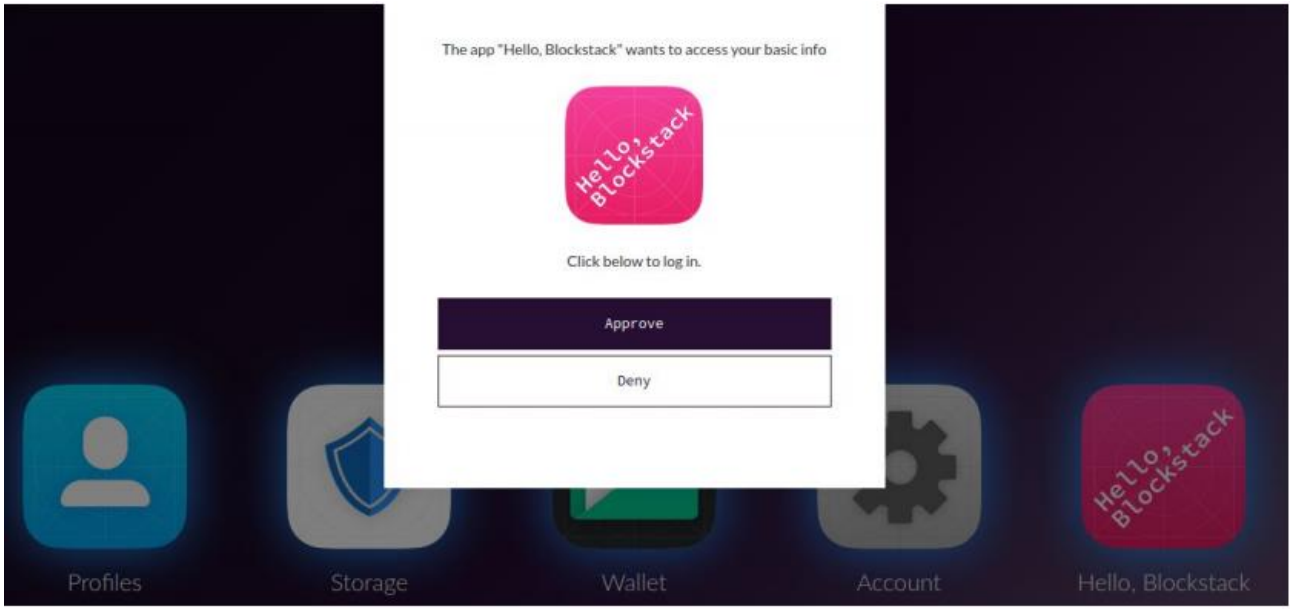

Рисунок 4.4 – Інтерфейс Blockstack Portal при запиті на автентифікацію

3. Користувач кліком підтверджує запит, Blockstack Portal генерує відповідь на запит автентифікації і перенаправляє користувача назад в додаток. Додаток автентифікує користувача.

4.1.4 Сайт Onename

Identity Provider від Blockstack, розташовується на сайті Onename за адресою https://onename.com.

За допомогою цього сайту можна зареєструвати собі ім'я, редагувати профіль, верифікувати своє ім'я в Facebook, Twitter, а також перенести створене ім'я на локальний owner-гаманець. Будь-яка плата за транзакції сплачується самим сайтом.

Через особливості роботи з блокчейном, ім'я може досить довго перебувати в процесі реєстрації (до декількох днів): воно буде відображатися на

сайті Onename, але не буде фактично зафіксовано в блокчейне. Це пов'язано з тим, що будь-які транзакції в блокчейне спершу повинні бути об'єднані в блок і пройти процедуру підтвердження іншими майнерами.

4.2 Python-бібліотека

Використання Python-бібліотеки було запропоновано в рамках цієї роботи.

Бібліотека дозволяє:

1. Згенерувати запит на ідентифікацію з боку сервера і відповідь на цей запит відповідно до заданого JSON-формату.

2. Підписати запит ключем сервера.

3. Підписати відповідь на запит ключем користувача.

4. Перетворити запит і відповідь на запит в форму JWT і назад в JSON.

5. Верифікувати запит на ідентифікацію.

6. Верифікувати відповідь на запит ідентифікацію.

Оскільки Blockstack розроблявся як мережа для безсерверних додатків, основна бібліотека для генерації запитів на автентифікацію і відповідей на запит автентифікації - бібліотека на Javascript, код якої виповнюється на стороні клієнта. Однак в застосуванні до традиційних серверних додатків клієнтська бібліотека не підходить з міркувань безпеки. додавши на сторінку свій скріпт, зловмисник може добитися неправильної роботи додатку, підмінивши всю логіку автентифікації. Таким чином можна, наприклад, автентифікуватись під чужим ім'ям і отримати доступ до чужих даними. Щоб такого не сталося, була розроблена бібліотека blockchainauth мовою Python, що виконується на рівні сервера, що відповідає за автентифікацію користувачів.

Фрагменти коду бібліотеки представлені в додатку А.

Запити на автентифікацію в рамках бібліотеки представлені у вигляді об'єктів AuthRequest. AuthRequest зберігає наступну інформацію:

1. Закритий ключ додатки (private_key).

2. Доменне ім'я додатка (domain_name).

3. Посилання, по якому можна отримати файл з інформацією про програму (Manifest_uri). Значення за замовчуванням: доменне ім'я + "/manifest.json".

За посиланням повинен повертатися JSON в наступному форматі:

{

"name": "Hello, Blockstack",

"start_url": "localhost: 5000",

"description": "A simple demo of Blockstack Auth",

"icons": [{

"src":

"https://helloblockstack.com/icon-192x192.png",

"sizes": "192x192",

"type": "image / png"

}]

}

4. Посилання, на яке клієнтський додаток відправляє відповіді на запити автентифікації (redirect_uri). Значення за замовчуванням - доменне ім'я.

5. Список дозволів (scopes) - тимчасово не використовується частина протоколу. Значення за замовчуванням - порожній масив.

6. Unix-час, через яке закінчується термін дії цього запиту (Expires_at). Значення за замовчуванням: через годину.

За допомогою бібліотеки запити можна верифікувати. Запит проходить верифікацію, якщо:

1. Він містить правильний підпис, згенерувала на основі даних з payload відкритим ключем зі списку public_keys.

2. Адреса поля iss - похідний відкритого ключа з public_keys.

3. Час iat вже пройшов.

4. Час exp ще не настав.

Відповідь на запит автентифікації в рамках бібліотеки представлені у вигляді об'єктів AuthResponse. AuthResponse зберігає наступну інформацію:

1. Закритий ключ додатку (private_key).

2. Профіль користувача в форматі JSON (profile).

3. Ім'я, по якому користувач автентифікується (username).

4. Unix-час, через яке закінчується термін дії цього запиту (Expires_at). Значення за замовчуванням: через місяць.

За допомогою бібліотеки відповіді на запити можна верифікувати. Відповідь проходить верифікацію, якщо виконуються всі ті ж умови, що і для самих запитів, а також якщо відповідь проходить додаткову верифікацію: адреса з поля iss повинна збігатися з публічною адресою власника цього імені. Для перевірки власника імені використовується або локально запущений Blockstack Core API, або публічний, запущений авторами Blockstack.

4.3 Django-додаток

Django-додаток – це зручний спосіб вбудувати на веб-сайт ідентифікацію та автентифікацію користувачів через протокол Blockstack. Використовуючи можливості Python-бібліотеки, додаток дозволяє:

1. За допомогою першого URL налаштувати процес автентифікації: при переході з цього URL, сервер за допомогою Python-бібліотеки генерує запит на автентифікацію і перенаправляє користувача до Blockstack Portal.

2. За допомогою другого URL (callback URL) додаток приймає відповідь на запит автентифікації, верифікує його і в разі успіху автентифікує користувача.

3. На ім'я користувача отримати його Blockstack-профіль.

Django - найпопулярніший веб-фреймворк загального призначення на Python і п'ятий за популярністю веб-фреймворк в цілому [34]. Він має вбудовану систему автентифікації і модель користувача, що відображається в базу даних. Django вимагає мінімальної настройки, щоб встановити стандартну систему зберігання паролів в зашифрованому вигляді на боці сервера. Але Django підтримує безліч інших систем автентифікації, включаючи OAuth і OpenID. Для підключення додаткової функціональності використовуються Django-додатки - бібліотеки з інтеграцією з Django.

Django-blockstack залежить від наступних пакетів: blockchainauth (Python-бібліотеки), самого django і blockstack-profiles (бібліотека з відкритим вихідним кодом для роботи з Blockstack-профілями).

Одна з головних складових програми - BlockstackAuthBackend. В термінах Django це Authentication Backend. Кожен раз, коли користувач намагається автентифікуватись і надає деяку інформацію про себе (наприклад, логін і пароль), Django викликає метод authenticate у всіх перерахованих в налаштуваннях Authentication Backend об'єктів. Користувач автентифікується, якщо хоч один з них повертає об'єкт User.

Оскільки в протоколі Blockstack користувач автентифікується за токенами, згенерованими його клієнтом, метод authenticate об'єкта BlockstackAuthBackend приймає аргумент auth_response_token замість традиційних username і password.

За допомогою бібліотеки blockchainauth токен верифікується, в разі успіху користувач знаходиться в базі або створюється по Blockstack-імені в якості username. На ім'я знаходиться профіль, з якого в базу записується інформація про ім'я користувача.

Спеціально для пошуку профілю існує функція fetch_profile. Спершу вона намагається отримати профіль з локально запущеного Blockstack Core API.

Якщо сервер API не запущені або профіль з інших причин не вдалося отримати, функція намагається отримати його з віддаленого Blockstack Core API.

Django-blockstack додає три посилання на сайт. Одне з них відповідає за генерацію об'єкта AuthRequest і перенаправлення користувача до його клієнту. В якості redirect_uri для AuthRequest вказується друге посилання, яке приймає токен AuthResponse і автентифікує користувача. За третім посиланням Blockstack Portal може отримати файл manifest.json сайту.

4.4 Аналіз використання протоколу Blockstack

4.4.1 Універсальність протоколу Blockstack

Однією з незаперечних переваг протоколу є єдиний вхід від одного і того ж імені на безліч сервісів. У світі з величезною кількістю веб-сервісів це позбавляє від необхідності створювати безліч різних облікових записів для цих сервісів, кожен раз заново заповнювати всю необхідну інформацію, зберігати паролі окремо для кожного. Ім'я в системі Blockstack може використовуватися на всіх сайтах, які підключили цей протокол, що можна легко зробити за допомогою Python-бібліотек. Аккаунти в таких сервісах, як Google або Facebook використовуються для автентифікації на безлічі веб-сайтів, і вони так само зручні для користувачів тим, що дозволяють не створювати окремі аккаунти для кожного сайту.

4.4.2 Безпека протоколу Blockstack

Найбільшою проблемою, пов'язаною з централізованими сховищами облікових записів, є безпека. Оскільки аккаунти користувачів зберігаються не у самих користувачів, вони не можуть контролювати заходи безпеки, що вживаються компаніями-постачальниками імен. Сервери компаній можуть бути зламані, і через них зловмисники можуть отримати доступ до аккаунтів користувачів. Злом облікового запису на одному з сервісів не завжди означає

компрометацію всіх облікових записів користувача, але на практиці користувачі часто використовують один і той же пароль для кількох сервісів. Якщо хакери дізнаються пароль від аккаунта на якомусь сервісі, а той же самий пароль використовується, наприклад, для електронної пошти, то зловмисник отримає доступ відразу і до пошти, і до всіх прив'язаним до неї аккаунтів. Особливо небезпечні масові зломи постачальників імен, що надаються різними протоколами Single Sign On (OAuth, OpenID), або зломи провайдерів електронної пошти, так як до електронної пошти дуже часто прив'язуються інші аккаунти. Це створює принципово дуже небезпечну ситуацію, при якій великі постачальники імен (Google, Facebook, Twitter, Microsoft, Yahoo) є гігантською точкою відмови всієї мережі. Щоб оцінити масштаби такого злому, необхідно оцінити частку аккаунтів, прив'язаних до постачальникам особистостей. І хоча повноцінної статистики з цих питань немає, можна все одно простежити загальні тенденції за непрямими даними. Результати аналізу, проведеного компанією-розробником рішень автентифікації LoginRadius [35] показує, що з 160000 веб-сайтів з функцією Social Login (вхід через соціальні мережі), 93% користувачів цих сайтів надали перевагу входу через соціальні мережі звичайної реєстрації по електронній пошті (рис. 4.5).

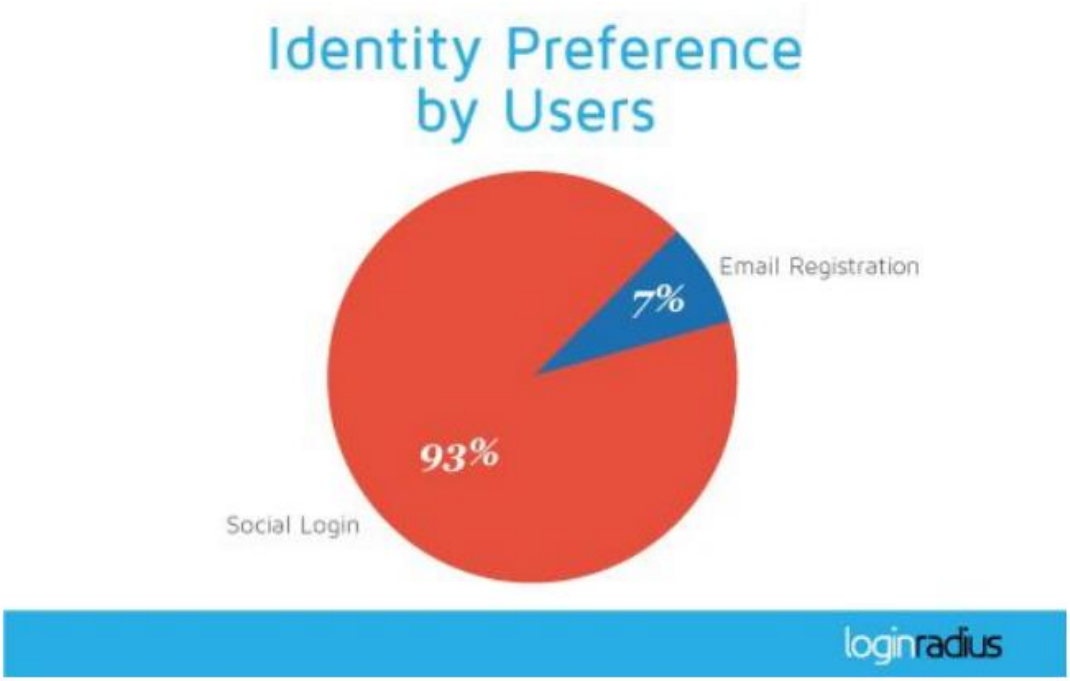

Рисунок 4.5 – Вподобання користувачів в методах автентифікації. Вхід через соціальні мережі проти стандартної реєстрації через пошту

При цьому переважна більшість користувачів, які обрали Social Login, є користувачами двох великих постачальників імен: Facebook і Google (рис. 4.6).

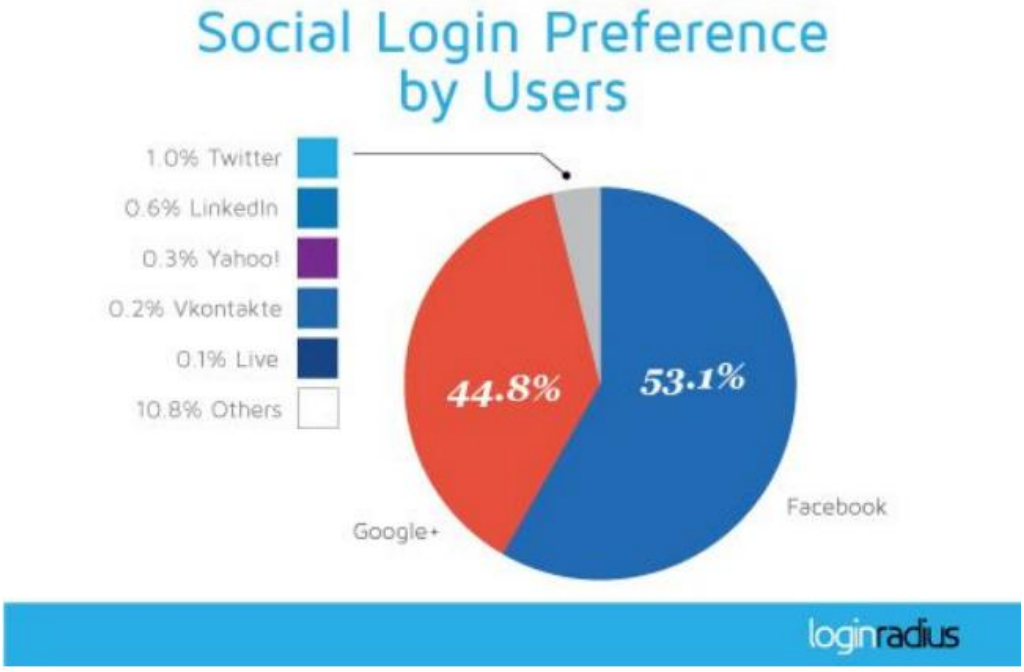

Рисунок 4.6 – Розподіл користувачів по сервісів Social Login

З цього можна зробити висновок, що всі користувачі, які скористалися можливістю автентифікації через соціальні мережі, знаходяться під загрозою крадіжки їх аккаунту в разі злому їх постачальника особистості, причому майже 98% користувачів залежать від двох сервісів, які є великими точками відмови.

Власники багатьох сайтів дозволяють своїм користувачам автентифікуватись через аккаунти великих постачальників імен. Так, через Facebook можна автентифікуватись приблизно на 14,5 мільйонах сайтів [36].

Таким чином можна зробити висновок, що з точки зору безпеки користувачі-власники аккаунтів та імен знаходяться в дуже великий залежності від декількох великих постачальників послуг і схильні до ризику компрометації своїх аккаунтів при компрометації цих сервісів.

Звичайно, запропоноване в роботі система не вирішує питання вразливостей в програмному забезпеченні. Навіть при використанні протоколу Blockstack зловмисники можуть здійснити такі атаки:

1. Атака на комп'ютер користувача. Хоча приватні ключі і зашифровані локально майстер-паролем користувача, в залежності від отриманих прав доступу зловмисник все одно може вкрасти аккаунт користувача (наприклад, отримавши у користувача пароль кейлоггером або представившись на сайті під ім'ям користувача через існуючу сесію в Blockstack Portal).

2. Атака на сервер програми, що використовує протокол автентифікації Blockstack. Зловмисник може отримати доступ до всієї призначеної для користувача інформації, що відноситься до цього додатку (наприклад, до особистого листування в соціальній мережі).

Однак при використанні протоколу Blockstack принципово неможливий той сценарій розвитку подій, коли злом одного сервісу тягне за собою крадіжку аккаунтів на іншому. Злом однієї сторони процесу автентифікації закінчується або компрометацію одного аккаунта (в разі злому локального комп'ютера

користувача), або компрометацією багатьох аккаунтів, пов'язаних тільки з цією стороною (в випадку злому сервера додатків). Постачальник імен в даному випадку замінюється блокчейном, і злом сховища імен рівноцінний компрометації гігантського блокчейна Bitcoin і контролю понад 51% вузлів мережі.

В роботі в якості протоколу ідентифікації та автентифікації був обраний протокол Blockstack через свої сильні характеристики безпеки. При активному використанні цього протоколу в мережі може зникнути проблема великих точок відмови в вигляді постачальників особистостей.

4.4.3 Децентралізованість Blockstack

При класичному сценарії зберігання імен користувачів, коли зберігання здійснюється на якійсь одній стороні (компанії-постачальнику імен), фактично ім'я контролюється не користувачем, а самою компанією.

Компанія-постачальник на свій розсуд може:

1. Повністю видалити ім'я користувача.
2. Змінити його дані.
3. Переглядати його дані і передавати третій стороні.
4. Використовувати дані користувача в комерційних цілях.

Тому і не можна говорити, що користувач в повній мірі володіє ім'ям. Іменем володіє компанія, яка дає його користувачу в безстрокову оренду на своїх умовах.

Централізоване зберігання імен користувачів може бути небезпечно ще і тим, що компанія-постачальник може просто припинити існування. Хоча в недалекому майбутньому це малоймовірний результат для великих компаній (Google, Facebook), але є і безліч дрібних і менш стабільних постачальників імен, від яких залежать користувачі.

Протокол Blockstack радикально вирішує проблему, використовуючи блокчейн-децентралізовану базу даних. Жоден вузол мережі не може повністю

контролювати систему, нові блоки додаються в блокчейн на основі консенсусу безлічі вузлів. Перераховані вище проблеми в блокчейне можуть виникнути тільки при порушенні роботи більше 51% вузлів блокчейна Bitcoin.

4.4.4 Недоліки та можливості їх подолання

Але проведений аналіз показав, що протокол Blockstack і запропонована система має свої недоліки.

Одним з головних, непереборних недоліків є невисока швидкість роботи системи в цілому. Це випливає з принципових особливостей протоколу. Оскільки всі імена зберігаються в блокчейне Bitcoin, всі операції так чи інакше пов'язані зі зміною цього сховища (реєстрація імені, оновлення zone-файлу, передача імені) проходять верифікацію і прийняття або відхилення на основі консенсусу декількох вузлів мережі.

Така транзакція може проходити від декількох хвилин до декількох годин.

Через свою новизну протокол знаходиться ще в стані апробації і не зовсім зручний для звичайних користувачів:

1. За кожну операцію, пов'язану з володінням ім'ям і зміною блокчейна користувачу доводиться платити невелику суму в біткоінах. Це випливає з того, що кожна така операція – це транзакція Bitcoin, яку необхідно підписати закритим ключем користувача. А все транзакції Bitcoin вимагають передачу хоча б невеликої суми.

2. Щоб скористатися протоколом, користувач змушений ставити додаткове програмне забезпечення (Blockstack Core, Blockstack Portal), яке знаходиться на ранній стадії експлуатації і часто не готово для зручної установки. На даний момент користувачі Windows і Linux змушені вручну збирати і запускати всі частини ПЗ, хоча розробники активно працюють над виправленням цієї проблеми.

3. Поки існує дуже мала кількість сайтів, де підключений протокол і де користувач може спробувати ідентифікацію через Blockstack.

В силу того, що протокол знаходиться на стадії розвитку, він поки не отримав широкого застосування. Проте він украй важливий хоча б як доказ того, що така важлива частина життя, як ідентифікація людини може управлятися без контролю третіми сторонами. Крім того, у протоколу є всі шанси поширитися по всьому світу завдяки підтримці Microsoft, з якими співпрацюють розробники Blockstack, щоб створити децентралізовану систему ідентифікації [16]

Головним внеском даної роботи є часткове подолання незручності, пов'язаної з роботою з протоколом. Завдяки розробленим бібліотекам все веб-сайти, які використовують Python і Django, можуть вбудувати автентифікацію користувачів через Blockstack, що сприяє більш широкому прийняттю протоколу.

Розробники Blockstack працюють над подоланням перерахованих недоліків, і є всі підстави вважати, що незабаром протокол ідентифікації та автентифікації буде більш зручний для широкого застосування.

ВИСНОВКИ

Більшість підходів до автентифікації та ідентифікації, що використовуються на даний час, можна розділити на два типи з точки зору їх переваг та недоліків:

1. Зрілі і надійні рішення, які тим не менше страждають від недоліків, через які такі протоколи можна використовувати як дійсно універсальний засіб ідентифікації у всіх сферах життя. Головним недоліком таких рішень є залежність від третьої сторони. До таких рішень відносяться зберігання зашифрованих паролів на стороні постачальника послуг, системи OAuth, OpenID.

2. Перспективні нові розробки на основі технології blockchain. Вони долають головні недоліки першого типу рішень: централізованість і залежність від третьої сторони. Всі рішення по зберіганню імені та приналежності його певній людині в таких системах приймаються консенсусом багатьох вузлів мережі.

Однак системи на основі технології blockchain знаходяться в стадії ранньої розробки і поки ще не можуть задовольнити всі запити користувачів.

У якості системи ідентифікації та автентифікації був обраний перспективний протокол децентралізованої системи імен Blockstack. За рахунок зберігання імен в блокчейне його використання вирішило більшість розглянутих недоліків, проте в силу своєї новизни був занадто незручний як для звичайних користувачів, так і для розробників веб-додатків, які могли б вмонтувати системи ідентифікації та автентифікації через Blockstack в свій додаток.

Запропонована Python-бібліотека blockchainauth призначена для зручної генерації запитів на автентифікацію і відповідей на запит автентифікації. Бібліотека може використовуватися як для ручного керування процесом автентифікації з боку розробника, так і в складі Django-додатків django-

blockstack, що дозволяє за допомогою однієї команди установки пакета і конфігураційних файлів вбудувати систему автентифікації через Blockstack на веб-сайт.

В роботі були отримані наступні результати:

1. Проаналізовані існуючі протоколи ідентифікації, їх переваги і недоліки. Було виділено клас найбільш перспективних протоколів на базі технології блокчейн.

2. Обраний для застосування протокол Blockstack, що використовує інфраструктуру блокчейн, вузли мережі, додаток-клієнт. Були вивчені його можливості, достоїнства і недоліки, детально розглянуто його застосування для ідентифікації та автентифікації користувачів.

3. Запропонована бібліотека blockchainauth на мові Python, придатна для легкої інтеграції системи автентифікації Blockstack на веб-сайти.

4. Розглянуті проблеми існуючих систем автентифікації та недоліки протоколу Blockstack та їх часткове вирішення за допомогою запропонованих бібліотек.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Как защитить свои данные от утечки, взлома, а телефон от потери или кражи. Цифровая безопасность. [Електронний ресурс]. – Режим доступу: https://mobile-review.com/articles/2020/security-phones.shtml.

2. Офіційний веб-сайт ID2020. [Електронний ресурс]. – Режим доступу: http://id2020.org/.

3. Погружение в технологию блокчейн: Децентрализованная беспарольная система безопасности. [Електронний ресурс]. – Режим доступу: https://habr.com/ru/company/microsoft/blog/316864/.

4. B. Kaliski. PKCS #5: Password-Based Cryptography Specification Version 2.0 // RFC 2898. - 2000. - С. 9-11.

5. Платформа авторизации OAuth 2.0. [Електронний ресурс]. – Режим доступу: https://tools.ietf.org/html/rfc6749.

6. OpenID Authentication 2.0. [Електронний ресурс]. – Режим доступу: https://openid.net/specs/openid-authentication-2_0.html.

7. OpenID Connect Core 1.0 incorporating errata set 1. [Електронний ресурс]. – Режим доступу: https://openid.net/specs/openid-connect-core-1_0.html.

8. Using OAuth 2.0 to Access Google APIs [Електронний ресурс]. – Режим доступу: https://developers.google.com/identity/protocols/OAuth2.

9. OAuth with the Twitter APIs [Електронний ресурс]. – Режим доступу: https://developer.twitter.com/en/docs/basics/authentication/overview/oauth.

10. Authorizing OAuth Apps [Електронний ресурс]. – Режим доступу: https://developer.github.com/apps/building-oauth-apps/authorizing-oauth-apps/.

11. The OAuth 2.0 Authorization Framework [Електронний ресурс]. – Режим доступу: https://tools.ietf.org/html/rfc6749.

12. Власов А.В., Сєвєрінов О.В., Слиш О.В. Впровадження децентралізованої системи ідентифікації // Проблеми інформатизації. Тези доповідей восьмої міжнародної науково-технічної конференції. 26-27.11.2020.

– Черкаси: ЧДТУ; Х.: НТУ «ХПІ»; Баку: ВА ЗС АР; Бельсько-Бяла: УТіГН; ДП «ПД ПКНДІ АП», 2020, Т. 1: секції 1-3. – С. 89.

13. Кравченко П. Блокчейн и децентрализованные системы: учеб. пособие для студ. заведений высш. образования: в 3 частях. Ч. 3 / П. Кравченко, Б. Скрябин, А. Курбатов, О. Дубинина. – Харьков: 2020. – 305 с.

14. Mastercard представляет модель цифровой идентификации, ориентированную на пользователя [Електронний ресурс]. – Режим доступу: https://plusworld.ru/daily/tehnologii/mastercard-predstavlyaet-model-tsifrovoj-identifikatsii-orientirovannuyu-na-polzovatelya/.

15. Блокчейн (мировой рынок) [Електронний ресурс]. – Режим доступу: https://www.tadviser.ru/index.php/.

16. Microsoft Building Open Blockchain-Based Identity System With Blockstack, ConsenSys, Bitcoin Magazine [Електронний ресурс]. – Режим доступу: https://bitcoinmagazine.com/articles/microsoft-building-open-blockchain-based-identity-system-with-blockstack-consensys-1464968713.

17. Introducing the Blockstack Identity System, Blockstack blog [Електронний ресурс]. – Режим доступу: https://blockstack.org/blog/introducing-the-blockstack-identity-system.

18. Microsoft показала сервис децентрализованной идентификации на базе блокчейн [Електронний ресурс]. – Режим доступу: https://ko.com.ua/microsoft_pokazala_servis_decentralizovannoj_identifikacii_na_baze_blokchejn_128732.

19. Обзор Blockchain проектов по идентификации личности человека [Електронний ресурс]. – Режим доступу: https://www.comnews.ru/digital-economy/content/110406/2017-11-09/obzor-blockchain-proektov-po-identifikacii-lichnosti-cheloveka.

20. Uport – цифровой паспорт пользователей Эфириума [Електронний ресурс]. – Режим доступу: https://bits.media/uport-tsifrovoy-pasport-polzovateley-efiriuma/

21. EMCSSL – Система идентификации пользователей WWW на основе подсистемы NVS криптовалюты EmerCoin и децентрализованных клиентских SSL-сертификатов [Електронний ресурс]. – Режим доступу: https://habr.com/ru/post/257605/.

22. Путешествие по DeFi вселенной: децентрализованная идентификация [Електронний ресурс]. – Режим доступу: https://3commas.io/ru/blog/puteshestvie-po-defi-vselennoj-decentralizovannaya-identifikaciya.

23. Blockstack PBC [Електронний ресурс]. – Режим доступу: https://angel.co/blockstack.

24. Офіційний сайт Blockstack [Електронний ресурс]. – Режим доступу: https://blockstack.org/intro.

25. Muneeb Ali, Jude Nelson, Ryan Shea, Michael J. Freedman. Bootstrapping Trust in Distributed Systems with Blockchains // ;login: - VOL. 41, № 3 - 2016. - С. 56.

26. Blockstack, Onename, and future applications - Identity - Blockstack Forum [Електронний ресурс]. – Режим доступу: https://forum.blockstack.org/t/blockstack-onename-and-future-applications/529.

27. Muneeb Ali, Jude Nelson, Ryan Shea, Michael J. Freedman. Blockstack: A Global Naming and Storage System Secured by Blockchains // 2016 USENIX Annual Technical Conference. - 2016. - С. 188.

28. Diploma project about Blockstack authentication protocol, Blockstack forum [Електронний ресурс]. – Режим доступу: https://forum.blockstack.org/t/diploma-project-about-blockstack-authentication-protocol/850.

29. Library for scanning blockchains and running Blockstack state engines, Github [Електронний ресурс]. – Режим доступу: https://github.com/blockstack/virtualchain.

30. Muneeb Ali, Jude Nelson, Ryan Shea, Michael J. Freedman. Blockstack: A Global Naming and Storage System Secured by Blockchains // 2016 USENIX Annual Technical Conference. - 2016. - С. 184-186.

31. OP_RETURN Stats [Електронний ресурс]. – Режим доступу: http://opreturn.org.

32. The Blockstack Browser Portal, Github [Електронний ресурс]. – Режим доступу: https://github.com/blockstack/blockstack-portal.

33. Американский блокчейн-стартап Blockstack запускает децентрализованный интернет-браузер [Електронний ресурс]. – Режим доступу: https://roskomsvoboda.org/28851/.

34. Web framework rankings, HotFrameworks [Електронний ресурс]. – Режим доступу: http://hotframeworks.com/.

35. Customer Identity Preference Trends Q2, LoginRadius [Електронний ресурс]. – Режим доступу: https://blog.loginradius.com/.

36. Facebook Connect Market Share and Web Usage Statistics, SimilarTech [Електронний ресурс]. – Режим доступу: https://www.similartech.com/technologies/facebook-connect.