國 立 交 通 大 學

網 路 工 程 研 究 所

碩 士 論 文

初 稿

基於區塊鏈的身份識別與存取控制用於開放銀行

Blockchain-based Identification and Access Control for Open Banking Ecosystem

研 究 生：鄭人豪

指導教授：袁賢銘 博士

中 華 民 國 110 年 8 月

基於區塊鏈的身份識別與存取控制用於開放銀行

Blockchain-based Identification and Access Control for Open Banking Ecosystem

研 究 生：鄭人豪 Student: Jen-Hao Cheng
指導教授：袁賢銘 Advisor: Shyan-Ming Yuan

國 立 交 通 大 學
網 路 工 程 研 究 所

碩 士 論 文 初 稿

A Thesis Draft
Submitted to Institute of Network Engineering
College of Computer Science
National Chiao Tung University
in partial fulfilment of the requirements
for the Degree of
Master
in
Computer Science

Aug 2021

Hsinchu, Taiwan

中 華 民 國 110 年 8 月

基於區塊鏈的身份識別與存取控制用於開放銀行

學生：鄭人豪　　　　　　　　　　　　指導教授：袁賢銘 博士

國立交通大學 網路工程研究所

摘　　要

隨著社群平臺愈來愈普及，多數網站提供第三方登入 (social login)，讓初次使用的用戶可以用第三方平臺現有的帳號完成註冊及登入，幫助用戶免於記下不同網站的帳號及密碼，亦不用填寫繁雜的註冊表單。對於用戶而言，可以達到更好的用戶使用體驗；對於應用程式開發人員而言，不必自行管理個資、建立會員系統，由第三方平臺負責管理，而當需要在提供多種不同的服務時，可以使這些服務支援同一種第三方平臺驗證方式，即可達到單一登入 (SSO) 功能。但第三方登入系統仍屬於中心化系統，用戶的數位身分及資料皆屬於第三方平臺，因此用戶在使用服務的同時，也提供個人資料給服務提供者。

然而，以太坊區塊鏈擁有防偽造、防竄改及去中心化的特性，可以安全、有效存放紀錄於鏈上，達到透明性且安全性。透過以太坊虛擬機及以太坊智能合約可以建構去中心化的應用程式，提供更完整、多樣的功能。

本篇論文提出使用以太坊區塊鏈技術應用於開放銀行 (Open Banking)，使銀行機構與第三方服務業者 (TSP) 以安全、透明方式合作，透過區塊鏈技術管理並整合用戶身分，方便用戶管理及存取資料，且提供第三方登入的優點，並使得去中心化平臺成為信任的第三方，負責管理對應用戶身分並且提供存取控制之功能，同時兼具區鏈特性，提升安全性。除了驗證用戶身分功能外，亦提供用戶授權功能，用戶可以自行決定授權範圍及對象，授權對象包含第三方服務 (TSP) 業者。

關鍵字: 區塊鏈、智能合約、第三方服務提供者、開放銀行

Blockchain-based Identification and Access Control for Open Banking Ecosystem

Student: Jen-Hao Cheng Advisor: Dr. Shyan-Ming Yuan

Institute of Network Engineering
National Chiao Tung University

Abstract

Open Banking has become a trend in the financial industries for providing innovative and diverse services. The concept of open banking aims to give third-party service providers (TSPs) access to customer's financial data for further analysis uses, thereby helping them to get a better deal and improve the customer experience.

Since an open banking ecosystem serves as a platform for various participants and establishes trust through trust third party, there has been increasing attention paid to personal information privacy and identity theft. Previous studies have examined blockchain technology applied to open banking for protecting customer's privacy, but identity verification and integration have been lacking.

The objective of our research was to formulate the protocols that blockchain-based identification and account integration. Utilizing the protocols we proposed, the customers view their data securely on TSPs and have the capability to authorize particular TSP to access their financial data. In this paper, the Ethereum blockchain was adopted to realize data sharing schema and access control management. Our experiment results show the scalability and usability of our system prototype.

Keywords: Blockchain, Smart contract, Third-party providers, Open banking

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In recent years social login services have become ever more prevalent such as Google, Facebook, or Twitter. The user can sign into a third party website without creating a new account. Those services provide unified identity management through social login and allow the user to confirm the access permission. Although social login seems to have a lot of benefits, e.g., security, convenience, and ease of use, there has a concern about those service providers may collect sensitive information about the user [1]. Besides, those social login services are mainly based on the centralized systems that enforce data permit across different parties. If one third party application sends a retrieval request for the data, it must obtain permission from the centralized party.

Open Banking is a hot topic in the financial industry nowadays. It aims to share customer's financial data with different organizations and required consent. The banking institutions disclose APIs to third-party service providers for creating new services, analytics, financial products to improve customer experience. It is not only meeting customer needs but also help third-party service provider to create innovate activity for exploring prospective customers and accelerate financial inclusion. In recent years, Open Banking has been adopted in various stages in countries around the world. Three phases have been defined by Open Banking and each stage represents sharing data scope. In Taiwan, Financial Supervisory Commission (FSC) has approved several banks allowing to join in the second phase of Open Bnaking [2].

- **Phase 1 Public information:** interest rates, exchange rates, mortgage rates, foreign currency exchange rates, etc.
- **Phase 2 Customer data:** accounts information, loan, deposit, credit cards, personal information, etc.

- **Phase 3 Transaction information:** account integration, payment, debit authorization, settlement of the loan, etc.

A key issue is the privacy of customer's personal data including customer's deposits, loans, investments, and account information. When financial institutions disclose APIs to TSPs, the system has numerous critical concerns such as malicious attacks, tampering. Once the attacker hacks the system, the customer data will be exposed and may cause enormous losses. Blockchain technology can provide data storage, access control, transaction security, and tamper-proof data so that it can protect customer data privacy [3].

Blockchain technology brings numerous benefits in a variety of industries, providing more security in trustless environments. The blockchain is a distributed digital ledger that storing records or data in blocks and those blocks are linked through cryptographic proofs so that the attacker can not temper any data. In Ethereum blockchain, it provides smart contract to us to deploy autonomous applications without third party and interact with smart contract on Ethereum network.

1.2 Objective

In order to address these problems, we propose blockchain-based identification and access control system for Open Banking ecosystem in this thesis. The system allows organization administrators to interact with blockchain for register the digital identity of the customer by the actual identity, and customer also can manage their digital identity and control their data access by calling smart contract functions directly. If any organization or financial institution wants to participate in existed blockchain-based Open Banking ecosystem, the system platform will create an Ethereum account for them so that they can prove they have ownership for Ethereum address. Customer registers their bank account and then bind their actual identity to their Ethereum address for enabling blockchain-based third-party authentication. After binding, customers have a unique digital identity on blockchain, and they can log in to another financial institution or third-party service provider without filling any form. In addition to binding, customers can also integrate existed accounts into their unique digital identity on blockchain.

With blockchain technology, the system is more secure, traceable, transparent, and tamper-resistant. TSP also does not have to create authentication systems and it can ensure customer's digital identity doesn't have tampered with. TSP is required to invoke smart contract function to confirm their access permission and access scope. After the financial institution successfully authenticating the TSP, it issues access tokens to TSP. TSP can request customer data with access token through API.

This thesis is organized as follows nine chapters. Chapter 1 and 2 introduce the background knowledge of blockchain and related work in Open Banking ecosystem. Chapter 3 describes the system overview, its scenario and explains how it work. Chpater 4 presents the smart contract used in our system and its workflow. Chapter 5 illustrates case studies and experimental validation. Chapter 6 presents demonstration of our system. Chapter 7 describes the evaluation of our system performance, including gas consumption and its throughput. Finally, we summarize our system functions, our finding, and discussion in Chapter 8 and 9.

Chapter 2

Background

2.1 Ethereum

2.1.1 Smart Contract

Ethereum [4] was proposed in 2014 by Vitalik Buterin, it is a decentralized, open-source blockchain, and it also supports smart contracts. Because Ethereum enables smart contracts, the programmers can build their distributed applications by writing smart contract programs, e.g., Solidity. With smart contract technology, they can build self-enforce, self-verify and tamper-proof systems, such as voting systems, healthcare, supply chain, financial service, and so on.

In the Etheruem platform, the block not only stores transaction records, but also stores smart contract so that Ethereum has the capability to compute business logic. The blockchain uses Merkle tree to store the transactions in every block.

2.1.2 Merkle tree

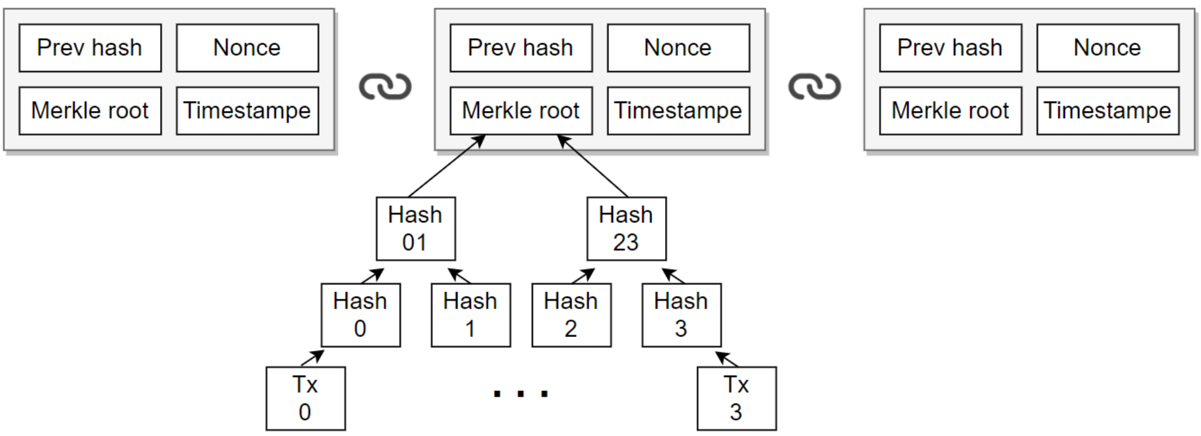

Figure 2.1: Merkle tree in Bitcoin blockchain

Merkle trees are important in blockchain technology. A Merkle tree is a tree in which every node is stored with the hash of data. Each node is the hash of his leaves. In Bitcoin blockchain, it uses the Merkle tree as proof to make sure the data block can't be tampered with, it is not possible to modify the data after the data written in the blockchain as shown Figure 2.1.

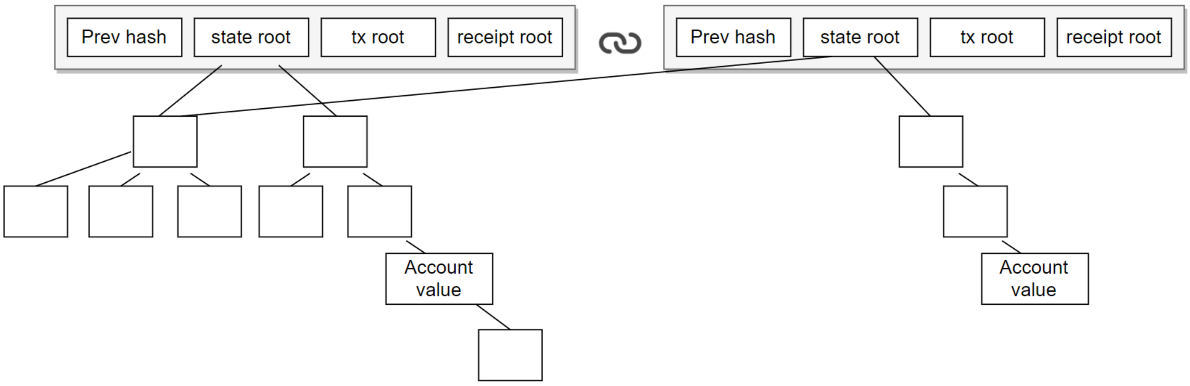

Figure 2.2: Merkle tree in Ethereum

The Merkle tree in Ethereum is as Figure 2.2. Every block header has not only one tree, but three trees for transactions, receipts, and state. The state root is the root hash of the Merkle Patricia tree that used to store the entire state of the Ethereum blockchain, like account balances, contract storage, and contract code. Unlike Bitcoin blockchain, Ethereum uses Merkle Patricia tree as the state tree, it consists of a map struct, the keys are account address and the values are account declarations such as the balance, nonce.

2.1.3 Elliptic Curve Cryptography (ECC)

Elliptic Curve Cryptography is public key cryptography based on elliptic curves. In Ethereum, they use ECC to generate the key pair. Initially, the base point G on elliptic curve $CURVE$ must be provided, and then randomly generate 256-bits integer m as a private key which is used to sign message or transaction. If given m, it is easy and fast to find 512-bits public key P. But if given P, it is impossible to find m because of Elliptic Curve Logarithm Problem. The Ethereum address is the last 20-bytes of Keccak-256 hash of the public key.

$$P = [m]G \tag{2.1}$$

Elliptic Curve Digital Signature Algorithm (ECDSA) is used to create a digital signature of data that uses elliptic curve cryptography. ECDSA isn't used to encrypt any data, it makes sure that the data was not tampered with. In the Ethereum blockchain, it used to prove ownership of an address without revealing private key.

After signing message using ECDSA through private key, the digital signature consists of three values: r, s, v. r and s are the values used in standard ECDSA signatures, and v means recover id that used to recover signed message. In blockchain (e.g., Bitcoin, Ethereum), we called it public key recovery. If we get (r, s) and message, we can compute the public key to verify message.

2.2 MetaMask

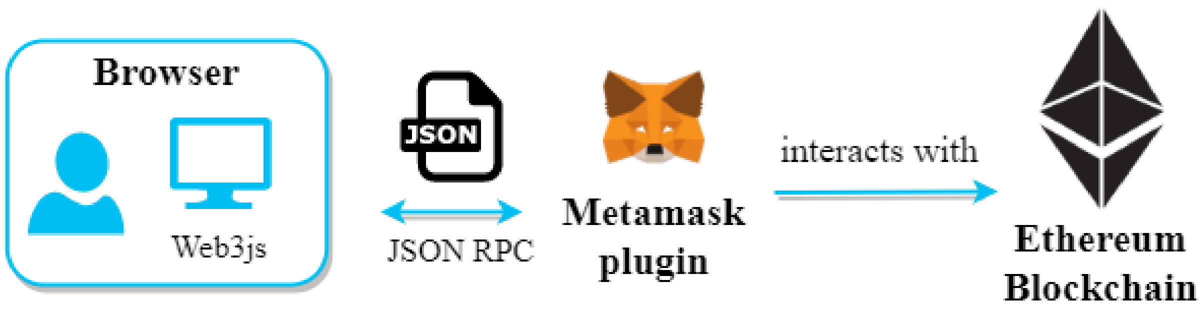

Figure 2.3: Architecture of DApp

MetaMask [5] is the most popular crypto wallet for accessing Ethereum distributed application (DApp). This tool can enable web3 API in website so that users can interact with various Etehereum blockchain from Javascript [6], e.g., Mainnet, Testnet. It also creates accounts by the user themself. The user of MetaMask can create and manage their accounts; moreover, MetaMask provides an interface that user can perform a transaction to the connected blockchain.

Because the user securely manages owned Ethereum account through MetaMask, the user can use their private key to sign a transaction or sign data to prove ownership of an account. Using this crypto wallet, the developers of decentralized applications can build their own cryptocurrencies and focus on designing and implementing functions of smart contracts. There are several different architectures for distributed apps [7], the most common way is the developer design frontend that can allow the users to interact with the business logic by MetaMask. Or, the user can send a transaction to blockchain directly.

In the MetaMask wallet, it seems to have a lot of benefits. Firstly, the keys are stored in the user’s browser and it doesn’t store on wallet provider’s server, so the user can manage his/her private key and public key without server. Secondly, it provides an easy to use interface, every user can send and receive cryptocurrency or token.

Regarding the architecture of Dapp, Figure 2.3 shows that the web3.js libraries can enable user’s browser to interact with blockchain so that users can read and write data from smart contracts, send transactions between accounts.

2.3 OAuth

- OAuth 2.0 flow and explain

2.4 Trust Service Provider (TSP)

- Open Banking flow - Current TSP in Taiwan - Three stage table

2.5 Related works

In this section, we will provide an overview of related literature about blockchain-based access control, identity management, data sharing, and Open Banking ecosystem.

In traditional access control management, the most common solution is PKIs, but it has some concerns about scalability and granularity. Paillisse *et al.* [8] presented a blockchain-based approach to address these problems. They take advantage of blockchain to record and distribute access control policies. Daraghmi *et al.* [9, 10] described a blockchain-based system for electronic medical records and academic records, using blockchain smart contract to manage the data access permission securely and effectively. It also utilizes advanced encryption techniques to protect user's privacy. Rouhani *et al.* [11] proposed a distributed Attributed-Base Access Control (ABAC) system that can provide auditing of access attempts. This work has focused on addressing audit and scalability, moreover, they apply the solution to the digital library and improve a lot. Fu *et al.* [12] proposed a user rights management system that aims to protect user privacy through enforcing executable sharing agreement. They also adopted multi-layer blockchain architecture to satisfy CAP (consistency, availability, and partition tolerance) theorem.

More recent attention has focused on data sharing. The common scenario is Open Banking system since it needs secure identity authentication and the perfect mechanism of user data privacy. With blockchain technology, it can solve the shortcoming of the centralized system and prevent user's data from being breached.

Chapter 3

System Design

3.1 Overview

Table 3.1: Notations

Notation	Description
C_i	Customer / User
Org_i	Organization
FR	Financial Regulation
D_{ij}	Data of C_i in Org_j
TSP_i	Third-party Service Provider
Acc_{ij}	Org_j Account of C_i
ID_i	Identification card number of C_i
Add_i	Ethereum address of i
Pri_i	Private key of Add_i
DI_i	Digital Identity of C_i
R_{ijk}	Access right of TSP_i to access Org_j with C_k consent
$ACMgr_i$	Access Control Manager contract of C_i
$OMgr$	Organization contract
N	Nonce

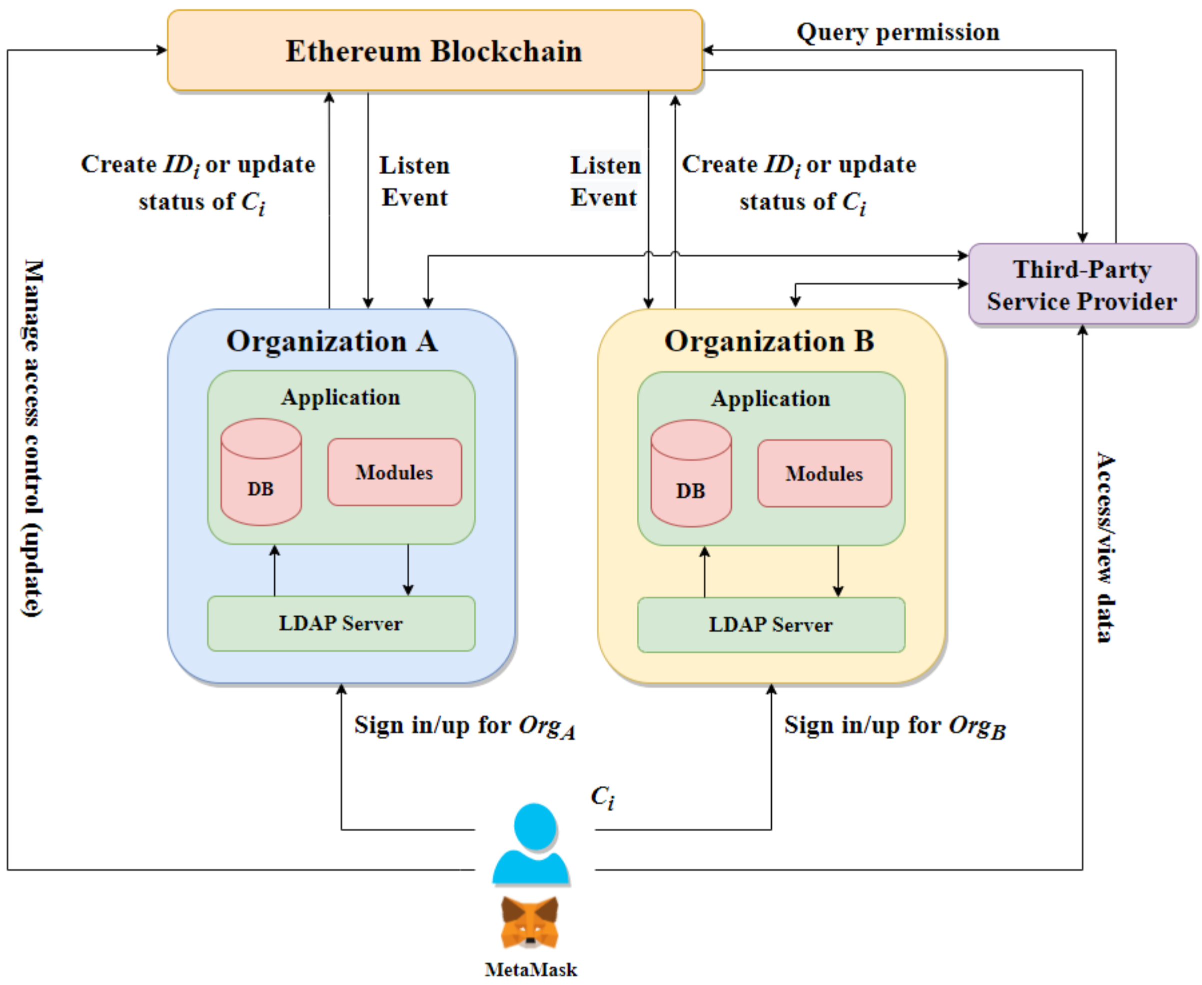

Figure 3.1: System Architecture

The architecture of our system is given in figure 3.1. Our proposed system solved the account integration and identity verification. And it can apply to various scenarios which need data sharing and distributed access control management such as open banking, medical records, and academic records. At the initial stage, each Org or TSP gets the exclusive Add and the corresponding Pri to take part in our open banking ecosystem. They must put the safety of Pri first and be careful; otherwise, identity theft may occur. There are several parties in our system: blockchain network, organization, third-party service provider, and user. The interpretation of these parties is stated as following:

- **Blockchain network:** The Ethereum blockchain network is used to store the user's DI and data access rights and deploy smart contracts, including identity creation and storage. Every node on blockchain has a copy of the ledger and ensures data on the blockchain cannot be tampered with. With decentralized technology, the management of data access rights doesn't have any governing authority to monitor. There have two kinds of contracts to realize that account integration and access manager, detailed in Sect. 4.

- **Organization:** A set of organizations within an ecosystem, identified by the Add and IP addresses. Each organization represents an independent system, it owns membership software that provides an organization with functionality such as storing and editing member information. The organization not only provides services to customers depend on applications, but also collects user data by using a database. In this paper, organizations provide an interface that users decide to share data.
  - $DB_j = \{D_{1j}, D_{2j}, ..., D_{nj}\}$: Database of Org_j is used to store customer data and only owner of data can decide which TSP can access.
  - Because the organization Org_j maintains membership software, it will provide the user C_i a regular account Acc_{ij}. In the appilcation of Org_j, the user C_i can access their data D_{ij}.
- **Third-party service provider:** The third-party service provider can be any organization or entity that performs financial services to customer. In this paper, we assume that TSP doesn't manage membership software and they retrieve customer data through blockchain technology.
  - The aim of the TSP is to collect data of C_i from organizations $\{Org_1, Org_2, ..., Org_n\}$ if the organization owns the data of C_i and TSP gets the permission.
  - When the permission is revoked by triggering smart contract, the access data request is regarded as invalid even if the token exists and it is not out of date.
- **User:** A user owns an organization account and wants to participate in our proposed system. After registering an organization account, the user first needs to pass identity card authentication. Due to the digital identity is unique and important on the smart contract, the organization takes responsibility for ensuring the correct identification card number and the safety of user's personal data. Besides, the user should generate the Ethereum account themself through Metamask.

3.2 Scenario

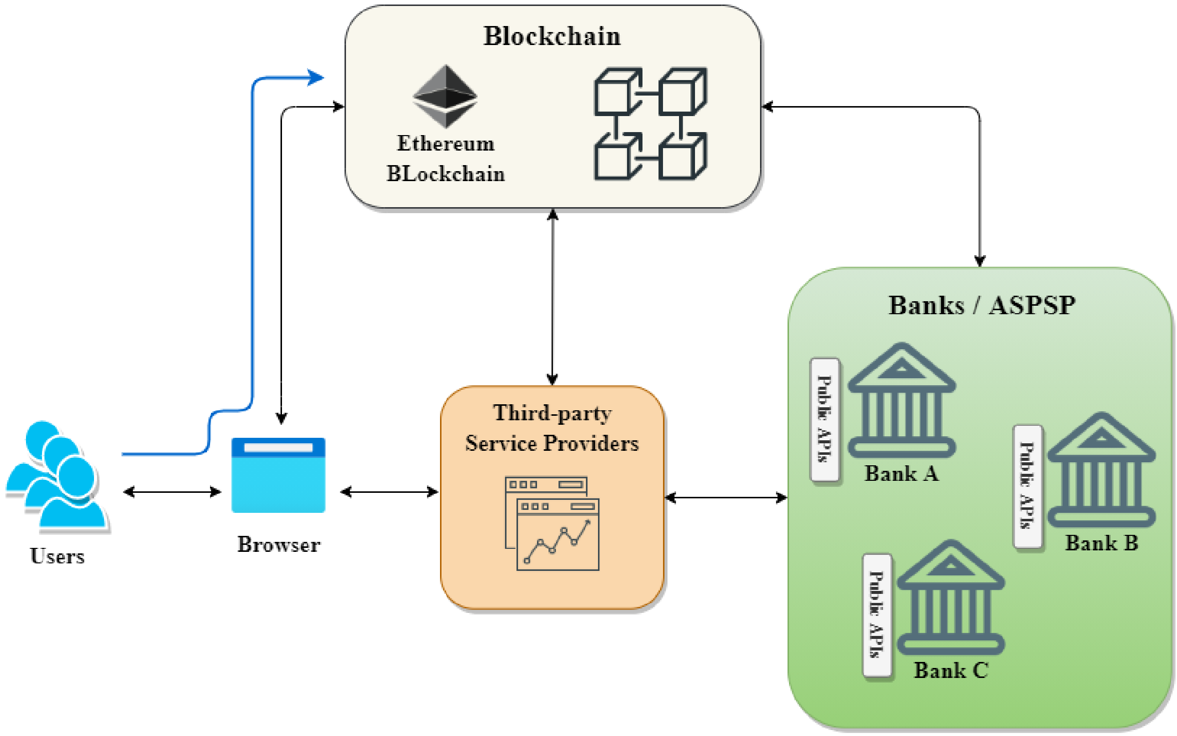

Figure 3.2: Relation between open banking roles

From the user perspective, our proposed system provides a single digital identity DI_i and access control. The access control for data sharing is constructed using smart contracts. So when banks disclose user's personal data to TSP, banks must have the user's consent through call specific user's smart contract.

In order to apply our proposed system to open banking ecosystem, we have three clearly defined roles: Customer (User), Financial institution, and Third-party services provider. Figure 3.2 gives an overview of our proposed system. It shows the relation between these roles and includes workflows, detailed in Sect. 3.3 Each customer interacts with Blockchain by using MetaMask, they not only login with MetaMask but also manage their own access manager contract. That's why customers can allow the specific party to access their data.

The user can specify the data attribute, source, destination through the smart contract to decentralized access control. And the user also can revoke access rights while he/she doesn't need the service that TSP provides.

3.3 Workflow

3.3.1 Identity verification

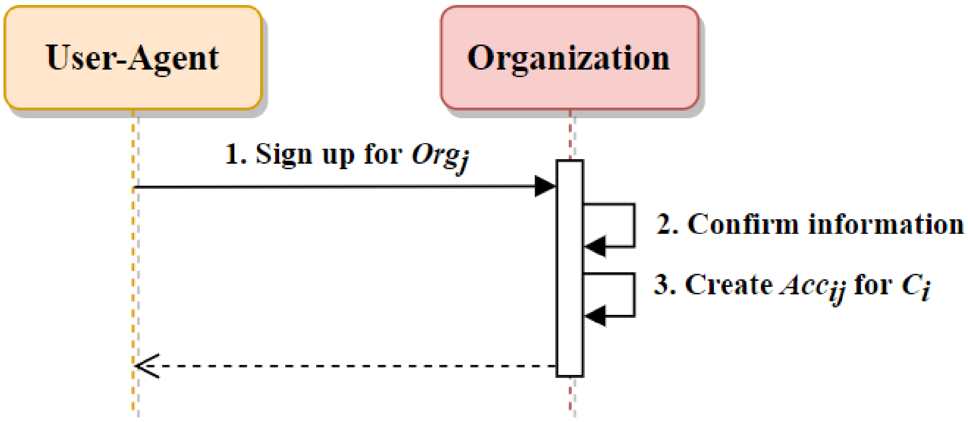

Figure 3.3: Account creation flow

User registration is commonly used by many companies or any online service which record user data. Although in recent years there has been an increase in social login, the sign-up for social media is still necessary. Figure 3.3 shows the flow for the user who visits the application (e.g., website of a bank) completes a signup form as shown in Table 3.2. The identification card number of the form is especially important for identity because it can be used to establish a digital identity on the blockchain. In this paper, we assume every organization should have its own membership software such as LDAP . And every user has multiple accounts, their personal information and data generated through organization service are belong to the organization themself.

Table 3.2: Sign-up form used in each organization

Fields	Description	required
username	Username as login account	Yes
password	The password of the account	Yes
email	User's email	No
phone	User's phone	No
id card number	Identification card number	No

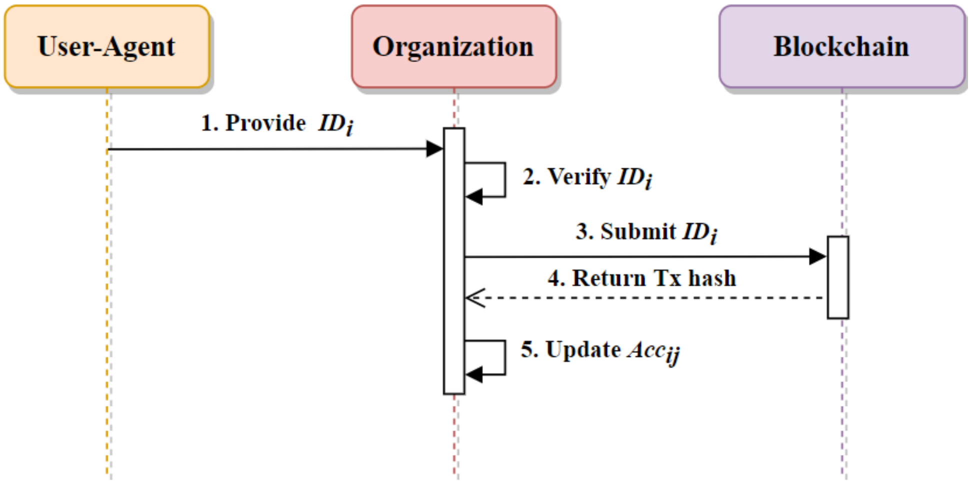

Figure 3.4: Identity verification flow

3.3.2 Account binding

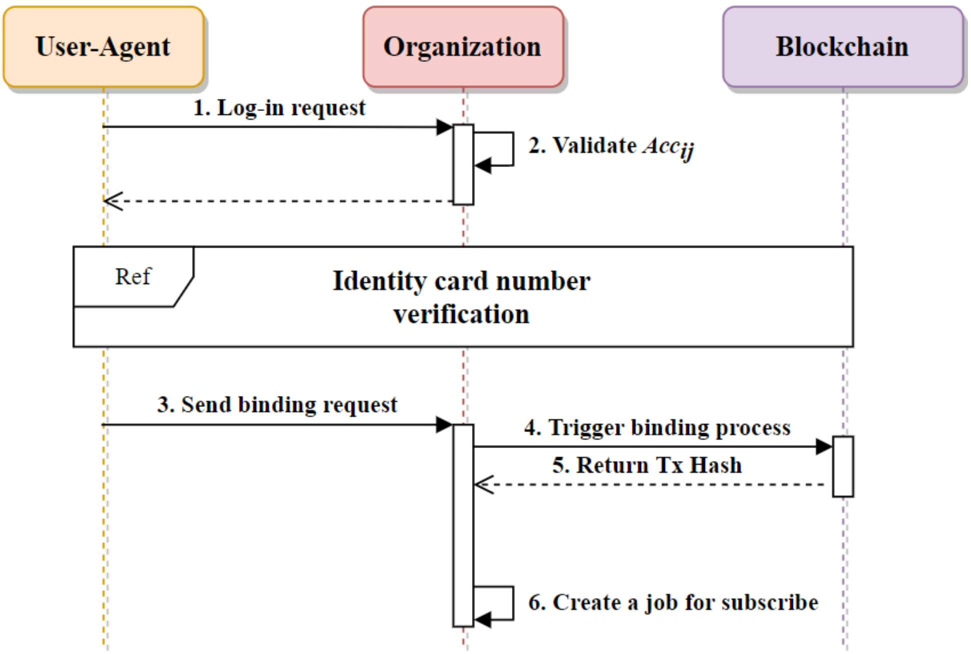

Figure 3.5: Account binding flow

3.3.3 Third-party login with Ethereum account

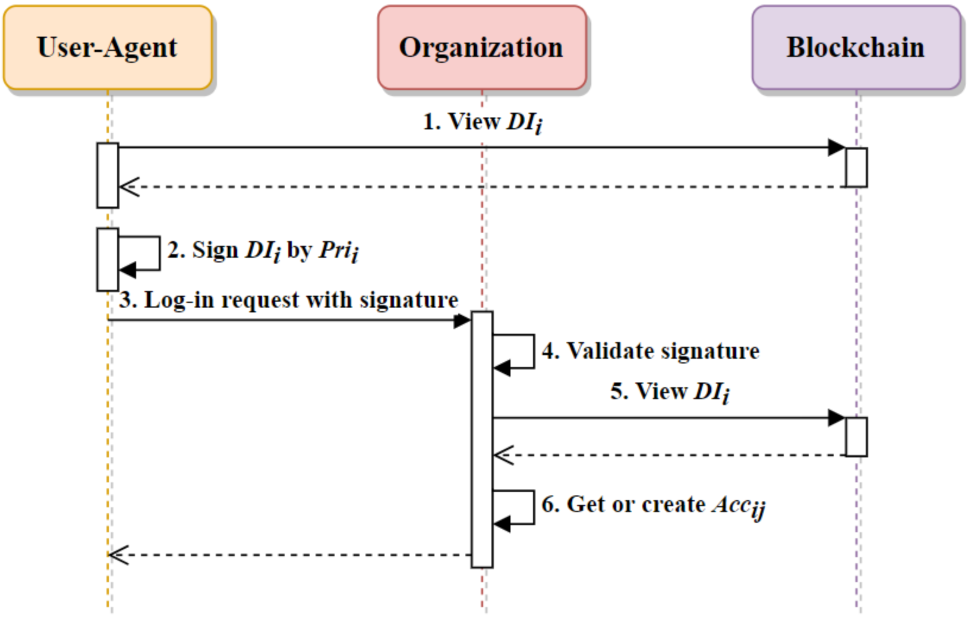

Figure 3.6: Third-party login flow

3.3.4 Data sharing

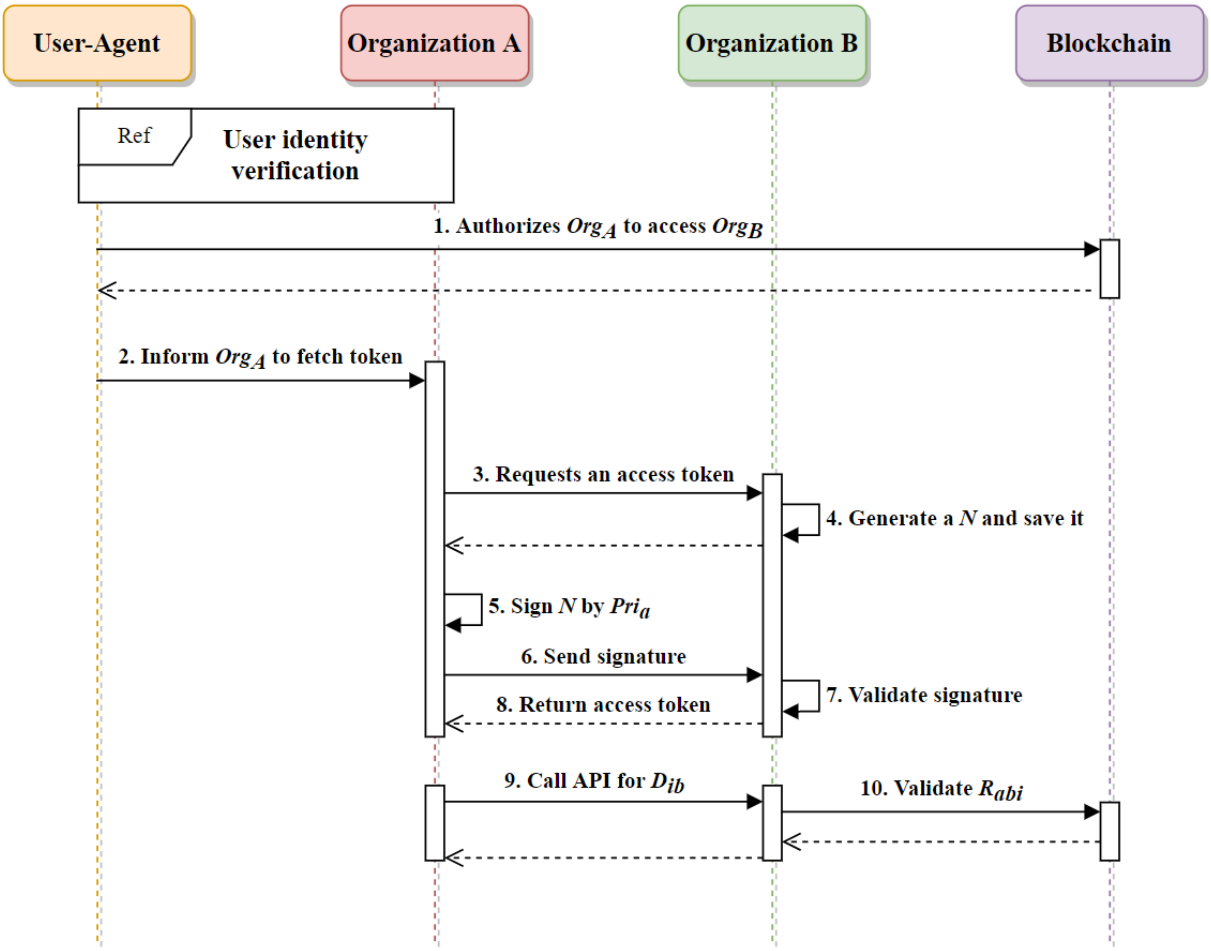

Figure 3.7: Date Sharing flow

Chapter 4

Implementation

This chapter describes the implementation of Ethereum blockchain smart contracts and defines relationships between users and organizations.

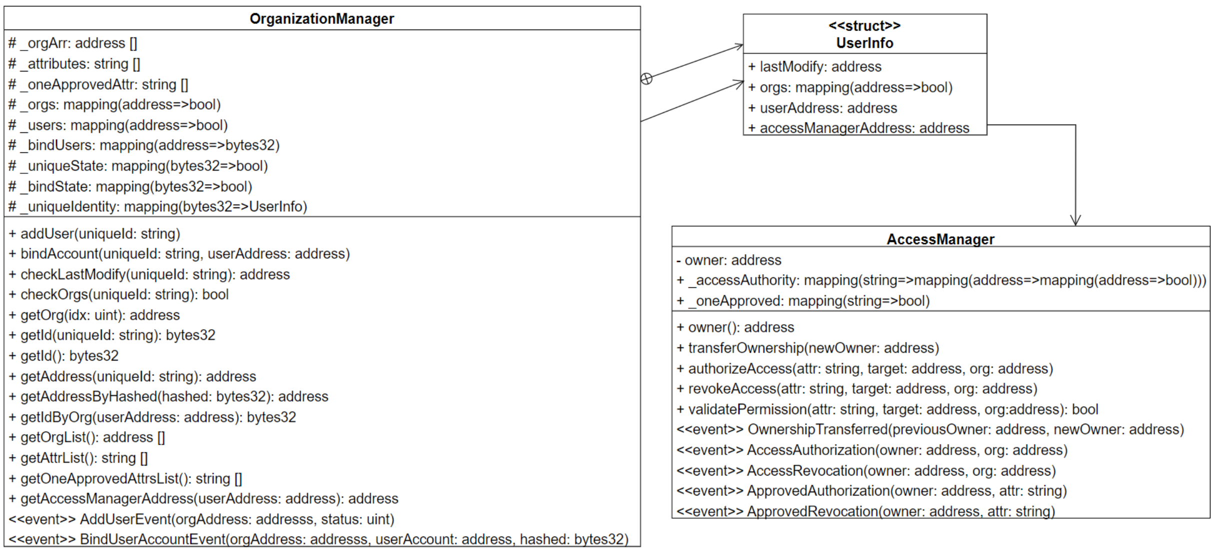

Figure 4.1: Smart Contract Diagram

4.1 Smart contract design

Organization Manager

Each organization manages mapping of user and orgs. // data structure

Access Manager

Each user manages access control list of data. // data structure

4.2 Third Party Login

4.3 Integration Account

4.4 Data Sharing

4.5 Token design

// jwt format(including iss, sub, hashed)

4.6 Data privacy protection

// which data will be save in BC and in DB.

Chapter 5

Experimental Case Study

Here are the experimental case study

Chapter 6

Demonstration

Here is the demonstration.

Chapter 7

Experimental Evaluation

Here is the evaluation.

7.1 Gas consumption

7.2 Throughput

Chapter 8

Discussion

future work: User data synchronization. How/Why? (KYC)

Chapter 9

Conclusion

Here is the conclusion.

References

[1] R. Gafni and D. Nissim, “To social login or not login? exploring factors affecting the decision,” *Issues in Informing Science and Information Technology*, vol. 11, no. 1, pp. 57–72, 2014.

[2] “Taiwan regulator allows banks to enter second stage of open banking,” *The Paypers*, Jan 2021. [Online]. Available: https://thepaypers.com/online-mobile-banking/taiwan-regulator-allows-banks-to-enter-second-stage-of-open-banking--1246579#

[3] H. Wang, S. Ma, H.-N. Dai, M. Imran, and T. Wang, “Blockchain-based data privacy management with nudge theory in open banking,” *Future Generation Computer Systems*, vol. 110, pp. 812–823, 2020.

[4] V. Buterin *et al.*, “A next-generation smart contract and decentralized application platform,” *white paper*, vol. 3, no. 37, 2014.

[5] “Metamask.” [Online]. Available: https://metamask.io/

[6] “web3.js - ethereum javascript api.” [Online]. Available: https://web3js.readthedocs.io/en/v1.3.0/

[7] F. Wessling and V. Gruhn, “Engineering software architectures of blockchain-oriented applications,” in *2018 IEEE International Conference on Software Architecture Companion (ICSA-C)*. IEEE, 2018, pp. 45–46.

[8] J. Paillisse, J. Subira, A. Lopez, A. Rodriguez-Natal, V. Ermagan, F. Maino, and A. Cabellos, “Distributed access control with blockchain,” in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–6.

[9] E.-Y. Daraghmi, Y.-A. Daraghmi, and S.-M. Yuan, “Medchain: a design of blockchain-based system for medical records access and permissions management,” *IEEE Access*, vol. 7, pp. 164 595–164 613, 2019.

[10] E.-Y. Daraghmi, Y.-A. Daraghmi, and S.-M. Yuan, “Unichain: A design of blockchain-based system for electronic academic records access and permissions management,” *Applied Sciences*, vol. 9, no. 22, p. 4966, 2019.

[11] S. Rouhani, R. Belchior, R. S. Cruz, and R. Deters, “Distributed attribute-based access control system using a permissioned blockchain,” *arXiv preprint arXiv:2006.04384*, 2020.

[12] W.-K. Fu, Y.-S. Lin, G. Campagna, D.-Y. Tsai, C.-T. Liu, C.-H. Mei, E. Y. Chang, M. S. Lam, and S.-W. Liao, “Soteria: A provably compliant user right manager using a novel two-layer blockchain technology,” *arXiv preprint arXiv:2003.10128*, 2020.

Appendix A

附錄標題

A.1 Testing