

國立政治大學資訊科學系研究所
碩士學位論文

去中心化數位貨幣交易記錄與查詢服務：設計
與以太坊實作

A Decentralized Digital Currency Tracing Service: Design and
Implementation on Ethereum

指導教授：郭桐惟 教授

研究生：朱奕寧 撰

中 華 民 國 一〇九年 十一月

摘 要

在這個數位崛起的時代，區塊鏈是這個時代所造就的新興科技。隨著區塊鏈的發展日漸成熟，該技術也逐漸在金融、物流、穩定幣等各個層面衍生出多樣化的應用與服務。它透過密碼學與分散式系統技術，使區塊鏈中的所有節點彼此交換訊息並共同維護鏈上資料(如交易帳本)。隨著時間的推進，區塊鏈中的資料與交易量不斷成長，因此，如何於龐大的區塊鏈中取得數位貨幣交易紀錄，已成為實務應用上必須面對的問題。為此，本論文設計一去中心化數位貨幣記錄與查詢服務。此服務實作於私有鏈之智能合約中，利用跨鏈技術存取於公有鏈上之交易資料，並於智能合約中使用 Binary Search 演算法，提升「區塊範圍查詢」、「時間範圍查詢」功能之效率。

關鍵字：區塊鏈、智能合約、跨鏈技術

ABSTRACT

In this era of digital rise, blockchain is an emerging technology created in this era. As blockchain technology matures, it has gradually derived diversified applications and services at various levels such as finance, logistics, and stable coins. It uses cryptography and decentralized system technology to enable all nodes in the blockchain to exchange messages with each other and jointly maintain chain data (such as transaction ledgers). As time progresses, the data and total amount of transactions in the blockchain for sustained growth. Therefore, how to obtain digital currency transaction history in blockchain has become a problem that must be faced in practical applications. For this reason, this paper designs a decentralized digital currency tracing service. This service is implemented in the private blockchain smart contract, uses cross-chain technology to access transaction data on the public blockchain, and uses the binary search algorithm in the smart contract to improve effectiveness of "block range query" and "time range query".

Keywords: blockchain, smart contract, cross-chain technology

目 次

摘 要	I
ABSTRACT	II
目 次	III
圖 次	IV
第一章 緒論	1
第二章 技術背景	3
2.1 區塊鏈	3
2.2 以太坊區塊鏈	5
2.3 ORACLIZE (PROVABLE)服務	6
第三章 相關研究	7
3.1 BIGCHAINDB	7
3.2 BLOCKIPFS - BLOCKCHAIN-ENABLED INTERPLANETARY FILE SYSTEM FOR FORENSIC AND TRUSTED DATA TRACEABILITY [13]	8
3.3 ETHEREUM QUERY LANGUAGE	9
3.4 ETHERSCAN API	10
第四章 系統設計實作	11
4.1 實作系統架構	11
4.2 ORACLIZE 與 ETHEREUM-BRIDGE 介紹	12
4.3 資料處理方式	14
4.4 QUERY 功能介紹	17
4.5 系統介面與操作流程	17
第五章 討論與結論	28
5.1 討論	28
5.2 結論	29
參考文獻	30

圖 次

圖 2.1：ORACLIZE(PROVABLE)服務示意圖	6
圖 3.1：BIGCHAINDB 系統架構圖	8
圖 3.2：BLOCKIPFS 之系統流程圖	9
圖 3.3：EQL 語法範例	10
圖 4.1：實作系統架構圖	12
圖 4.2：服務介面之網頁架構圖	17
圖 4.3：數位貨幣交易記錄與查詢服務之首頁畫面	18
圖 4.4：註冊主頁	18
圖 4.5：代幣註冊之畫面	19
圖 4.6：完成註冊服務之畫面	20
圖 4.7：查詢頁面確認註冊完成之畫面	21
圖 4.8：修改代幣資訊之畫面	22
圖 4.9：顯示已同步之所有交易紀錄之畫面	23
圖 4.10：詳細交易內容之畫面	24
圖 4.11：訂定區塊範圍篩選之畫面	24
圖 4.12：區塊範圍篩選結果之畫面	25
圖 4.13：訂定日期範圍篩選之畫面	25
圖 4.14：日期範圍篩選結果之畫面	26
圖 4.15：特定位址篩選之畫面	26
圖 4.16：特定位址篩選結果之畫面	26
圖 4.17：比較 BINARY SEARCH 搜索效能	27

第一章 緒論

隨著網際網路的普及，已有各式各樣的數位化商品與服務陸續出現，隨之改變大眾的生活習慣。然而，因近年來區塊鏈技術日漸成熟，貨幣的數位化才得以被實現。除了數位貨幣之外，也衍生出金融性商品、物流追蹤、產品保固卡等各個層面的應用與服務。重要的是，區塊鏈具備下列三大特性：(一)交易紀錄不得竄改、(二)可追溯性、(三)去中心化，下列分別介紹此三項特性。

(一)交易紀錄不得竄改：採用密碼學原理將交易紀錄儲存於區塊鏈中，並透過時間順序排序，一旦數據寫入至區塊鏈中，任何人都無法輕易擅自更改數據內容。

(二)可追溯性：區塊鏈系統為塊鏈式結構，鏈上數據依時間順序環環相扣，造就了區塊鏈之可追溯性。

(三)去中心化：因這個特性讓區塊鏈被稱之為「分散式帳本」，由系統中所有節點共同維護與更新同一本帳本，且沒有中央伺服器來管理秩序。

區塊鏈便是能同時滿足交易紀錄不得竄改、可追溯性且去中心化的安全系統。隨著區塊鏈技術的演進，現今也發展出了智能合約 [1]，其合約內容與代碼皆由程式所編寫而成，是區塊鏈中一種制定合約時所使用的特殊協議，主要用於提供驗證及執行智能合約內所訂定的條件。目前大多應用於發行數位資產(如 ERC-20 Token [2])、儲存資料或是開發金融類型協議(如 DeFi [3])。智能合約也具有高交易效率，合約的流程幾乎為自動化，使得交易效率提高；也能透過客製化合約，依照用戶的需求進行修改。

因應不同使用者需求及應用場景的不同，區塊鏈主要可分為公有鏈和私有鏈。公有鏈為任何人都能參與的區塊鏈，不需要通過授權，即可自由加入或退出，如比特幣 [4]或以太幣 [5]等；私有鏈則是需要受到授權才能參與的區塊鏈，並非任何人都能加入，像是 Facebook Libra [6]、R3 Corda [7]等。因參與私有鏈的節

點有權限制和可控制的區塊鏈系統設置，因此，私有鏈往往可以有較快的交易處理速度、每個區塊可容納更多的交易量且能夠節省需花費的交易手續費。本論文將聚焦於私有鏈上。

隨著時間的演進，區塊鏈中的資料與交易總量不斷成長，因此，如何於龐大的區塊鏈中取得數位資產的交易紀錄，已成為實務應用上必須面對的問題。如今也有少數的現有技術與參考文獻來解決此問題，像是 BigchainDB [8]、EQL [9]、Etherscan [10]。然而，BigchainDB 主要是針對分散式資料庫與區塊鏈的結合，但無提供智能合約的內部交易資訊的查詢(例如 ERC-20 Token 代幣之交易數量)；另一方面，EQL 提供使用者可透過類似 SQL [11]的查詢語言檢索區塊鏈中的數據，但實際上，EQL 與 BigchainDB 皆無法達成這項檢索；雖然 Etherscan 為現今最為成熟的 Ethereum Block Explorer，其檢索區塊鏈中的數據功能齊全，也提供智能合約的內部交易資訊，但沒有提供區塊範圍查詢、時間範圍查詢的功能。本論文欲設計一去中心化數位貨幣交易記錄與查詢服務。更明確地說我們希望在不遺漏任何交易紀錄、無誤的交易內容且交易來源可信任的情況下，設計一查詢數位貨幣之歷史交易服務。重要的是，該服務提供了區塊範圍查詢與時間範圍查詢功能。以下是本篇論文的研究目標：

- 設計一去中心化數位貨幣交易紀錄與查詢服務，並保證不遺漏任何交易紀錄、無誤的交易內容且交易來源可信任
- 將該服務實作於以太坊私有鏈上，並以智能合約開發其服務
- 結合 Oraclize(Provable) [12]服務於智能合約中，作為公有鏈與私有鏈間之資料存取之跨鏈技術
- 實作 Binary Search 於智能合約中，提升使用者查詢速度

第二章 技術背景

本研究以探討去中心化數位貨幣交易紀錄與查詢服務為目標，將以太坊區塊鏈作為開發平台，並以智能合約結合 Oraclize(Provable)服務實作於區塊鏈跨鏈互操作性應用，將存在於公有鏈上之數位貨幣歷史交易儲存至私有鏈中，使用者可針對儲存於私有鏈中之歷史交易進行查詢。本章共分三小節，首先簡介區塊鏈之相關技術背景，接著針對以太坊區塊鏈平台及其智能合約簡述其相關特性，最後則是介紹 Oraclize(Provable)服務是如何可信地與區塊鏈外部溝通。

2.1 區塊鏈

「區塊鏈」源自於中本聰(Satoshi Nakamoto)的比特幣，透過複雜的密碼學加密資料，再利用數學分散式演算法，成為「去中心化的系統」，也是作為比特幣的底層技術，或稱為「去中心化的分散式資料庫」。區塊鏈將舊有的資料庫重新改造，把資料分成不同的區塊，每個區塊透過特定的協議鏈結到上一區塊，將區塊互相接連呈現一完整的歷史數據。區塊鏈上的每筆交易，都能由區塊鏈的結構設計達到溯源的功能，並能針對每筆交易進行驗證。在區塊鏈的每一個區塊當中，皆會擁有一時間戳進行紀錄，以表示該區塊是由當時的時間所生成的，造就了不可篡改、不可偽造的系統。區塊鏈中所有參與的節點在沒有中心的情況下，節點們透過去中心化的共識機制、遵循著規則驗證訊息內容，並寫入時間戳後生成區塊，再將其廣播至所有節點。透過分散式的方式達成數據儲存、交易驗證、訊息傳遞即為區塊鏈的核心技術。

自從比特幣的出現後，區塊鏈的技術逐漸受到重視，公開透明、公平競爭。隨後世界便開始探索區塊鏈技術在各行各業中發展的可能性，因此，除了常見的

公有鏈以外，隨著應用場景的不同，更衍伸出適合企業、產業界使用的私有鏈與聯盟鏈。依照使用者對區塊鏈的權限程度不同，種類可略劃分為三種類型：

- (1)公有鏈：任何人都可以訪問，發送、接收、驗證交易，並參與共識過程的區塊鏈。我們最熟悉的區塊鏈，大多屬於公有鏈，像是比特幣、以太幣等。
- (2)私有鏈：區塊鏈的權限被一定程度地進行了限制，須受到授權才能成為節點，並非任何人都能參與。例如摩根大通(JP Morgan)引領的 Quorum 就是私有鏈的代表之一。
- (3)聯盟鏈：聯盟鏈與私有鏈相似，區塊鏈的開放程度與權限也是有所限制的，而授權的節點通常為企業與企業間有合約的關係等。舉例來說，Hyperledger Fabric，以及 R3 Corda 皆是聯盟鏈的一種。

此外，區塊鏈上的交易是在於打包交易並出塊，通過驗證與確認交易是否有效，使交易順利完成，並讓區塊鏈中的所有節點進行更新、以及擁有相同的帳本內容，這樣打包交易並出塊的機制就是「挖礦」。在以太坊區塊鏈中，Ether(ETH)是其中的燃料，在區塊鏈上進行交易時，必須經由礦工運算打包後上鏈，使用者必須支付礦工該交易的手續費，其交易手續費是以 Gas 計算，並以 Ether 支付。於以太坊主網當中，每個區塊都有 800M 的大小限制(Block Gas Limit)，而一般標準交易的 Gas Limit 約為 21,000，一個區塊約可以容納 380 筆交易。交易的 Gas Limit 會依據交易的內容而有所不同，每筆交易的發起方都須設置交易的 Gas Limit 和 Gas Price，不同的操作會產生不同的 Gas 成本，若是交易需花費的 Gas Limit 太高或是超出區塊鏈所設定的大小限制，則可能發生交易無法打包至區塊中，而滯留在交易池(Transaction Pool)的狀況。

整體而言，區塊鏈的應用技術不斷創新，仍有必要持續探索區塊鏈此一新興科技的潛能，在實務應用上，則必須解決區塊鏈的交易擁塞的情形、須支付的 Gas Fee 過高，以及區塊的 Gas Limit 的限制。有鑑於此，為了實現數位貨幣溯源的服務，本研究將實驗環境建立於私有鏈當中，為解決沒有足夠的 Ether 予以支付

交易手續費的狀況、打包交易並出塊的速度、以及自定義足夠大的區塊 Gas Limit，且利用智能合約的多元功能，進行數位貨幣溯源的探討與實驗，並針對以太坊區塊鏈的私有鏈平台與 Oraclize(Provable)的服務應用結合，儲存以太坊公有鏈上之數位貨幣之相關交易進行溯源，透過跨鏈互操作性的方式傳遞歷史交易，並開發實驗系統進行探討與實作，利於後續搜索區塊鏈之實務應用發展的需求之參考。以下將針對以太坊區塊鏈與 Oraclize(Provable)服務分別簡述其相關特性。

2.2 以太坊區塊鏈

以太坊區塊鏈在比特幣區塊鏈的基礎上，衍伸了一個具開源性與開發性高的區塊鏈平台，且支援開發多元化區塊鏈分散式應用程式，稱之為智能合約(Smart Contracts)，並透過其原生貨幣以太幣(Ether)執行去中心化的運行。以太坊區塊鏈也具有圖靈完備(Turing Complete)的特性，使用者能透過 JavaScript 和 Python 等現有語言在以太坊虛擬機(Ethereum Virtual Machine，EVM)上開發分散式應用程式，或稱為 Dapps，並共同維護以太坊區塊鏈之完整性。

智能合約是區塊鏈中的一種特殊協議，其合約內容與代碼皆由程式所編寫而成，可用於發行客製化代幣、紀錄資料或是金融交易等功能。智能合約具有高安全性、高交易效率與可客製化等，我們也能透過智能合約儲存資料，且本身也能傳遞資料訊息到外界。智能合約就像是個可以被信任的第三方角色，能在沒有中介的情況下，進行交易。智能合約使得區塊鏈在未來能有更多的可能性。

ERC-20(Ethereum Request for Comment)是一個基於以太坊智能合約的一種 Token 標準協議(EIP，Ethereum Improvement Proposal)，所有的 ERC-20 代幣都能於以太坊中進行交易、追蹤或是監測等。以太坊中的代幣交易皆須透過發起交易(Transaction)才能完成動作，而這些交易都會被記錄在區塊鏈上，使交易都具有可追溯性。

2.3 Oraclize (Provable)服務

Oraclize(Provable)作為區塊鏈外部世界之資料提供者，能夠在區塊鏈上提供可信的數據傳送服務，其目的是在區塊鏈建立一條可與外部溝通的橋樑，為了解決智能合約取得數據的限制，在保證可信的情況下，使得智能合約具有取得外部數據的能力。此服務也不僅局限於以太坊，目前也可以使用在其他區塊鏈平台上，例如 Bitcoin、EOS、Rootstock、Fabric。

Oraclize(Provable)在以太坊上部署了名為 usingOraclize 的智能合約，只需要在自己的智能合約當中引用，並依據 Web API 的使用方法進行調用即可。Oraclize(Provable)主要是透過監聽特定 Event 來接收合約所發出的 Query 請求，處理完成後，再由 Oraclize(Probable)主動呼叫合約的 callback function 將資料回傳。此外，Oraclize(Provable)也提供了五種資料來源，使智能合約取得外部資料，包括 URL 資料來源、WolframAlpha 資料來源、IPFS 資料來源、Random 資料來源、以及 Computation 資料來源。

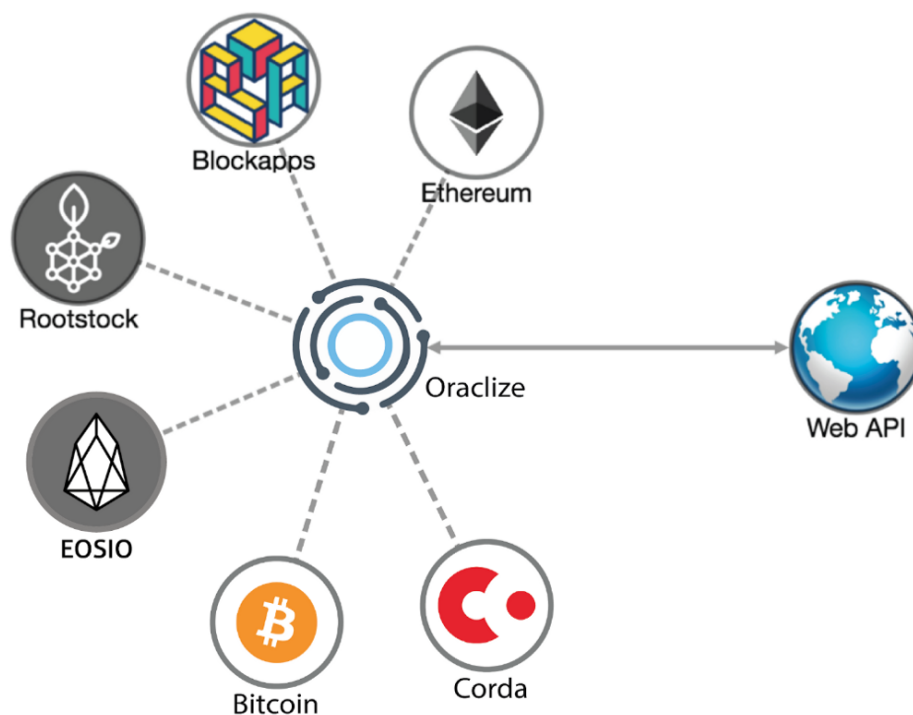


圖 2.1：Oraclize(Provable)服務示意圖

第三章 相關研究

本論文欲設計一建構於區塊鏈智能合約之數位貨幣交易記錄與查詢服務，以參考現有相關技術，並針對該服務之需求進行研究。本論文將以此作為出發點，進行該題目之探討與延伸，並開發該服務系統作為依據。本章共分為四小節，第一小節為 BigChainDB，該論文則是針對分散式資料庫與區塊鏈的結合，以達成兩者之特性；第二小節為 BlockIPFS，該論文針對分佈式文件系統與區塊鏈整合，並未利用利用區塊鏈作為儲存資料之工具；第三小節為 EQL，該論文針對區塊鏈查詢的功能，實作出一查詢語言 EQL(Ethereum Query Language)，此工具可針對區塊(Block)與交易(Transactions)進行查詢，但尚未完成智能合約內部數據的查詢；第四小節為 Etherscan，是一個 Ethereum Block Explorer，可以查看區塊鏈上所有發生的交易及其交易狀態等。

3.1 BigChainDB

BigChainDB 是一個可用的去中心的資料庫。它的設計起源於分散式資料庫，加入了很多區塊鏈的特性，像是去中心控制、不可改變性、數字資產的建立和移動。BigChainDB 繼承了現代分散式資料庫的特性：吞吐量和容量都是與節點數量線性相關、功能齊全的 NoSQL 查詢語言，以及高效的查詢和許可權管理。因建構在已有的分散式資料庫上，它在程式碼層面也繼承了企業級的健壯性。下圖為 BigChainDB 之系統架構圖，根據現版本之設計，每個節點都有一 mongoDB 資料庫用於儲存鏈上資料之用，以此使用者皆能透過常規之 SQL 語言進行鏈上資訊(Block、Transaction 等)之搜索，即不包含智能合約內部數據查詢。

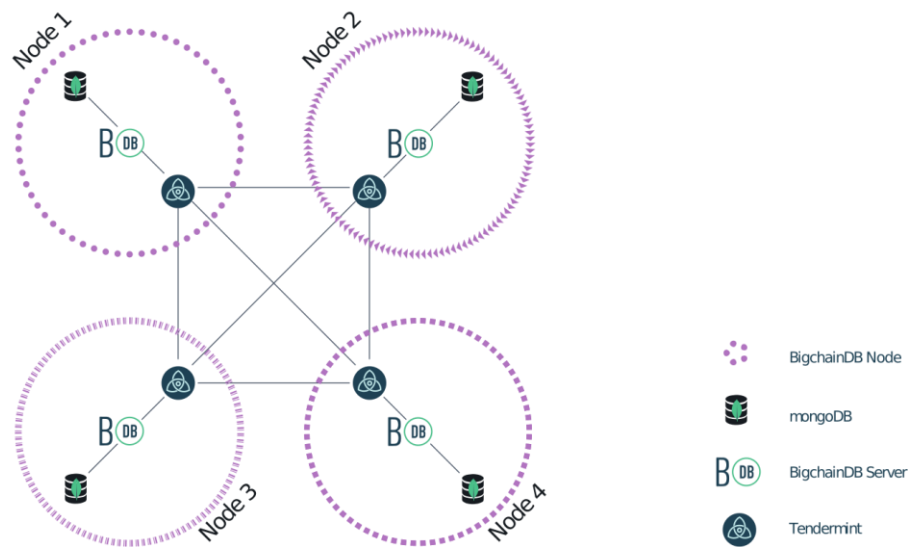


圖 3.1：BigchainDB 系統架構圖

3.2 BlockIPFS - Blockchain-enabled Interplanetary File System for Forensic and Trusted Data Traceability [13]

在此論文中，探討了將區塊鏈與分佈式文件系統(IPFS，Interplanetary File System)整合，該系統利用區塊鏈(該論文實作於須身份驗證之區塊鏈 - Hyperladder Fabric)既有的安全性和可追溯性，其溯源性可用於審核和保護文件的作者身份，並可作為調查的證據和爭議解決。開創一個更安全的文件共享平台，該平台具有清晰的審核記錄，以提高數據的可信度以及作者身份保護，方法提供了一條明確的途徑來追溯與給定文件相關的所有活動（如添加文件、訪問文件等）。下圖為論文中 BlockIPFS 之架構，用於說明文件上傳之步驟，其文件本身由 IPFS 加密和管理，而解密文件的密鑰存儲在區塊鏈上的特定通道中。

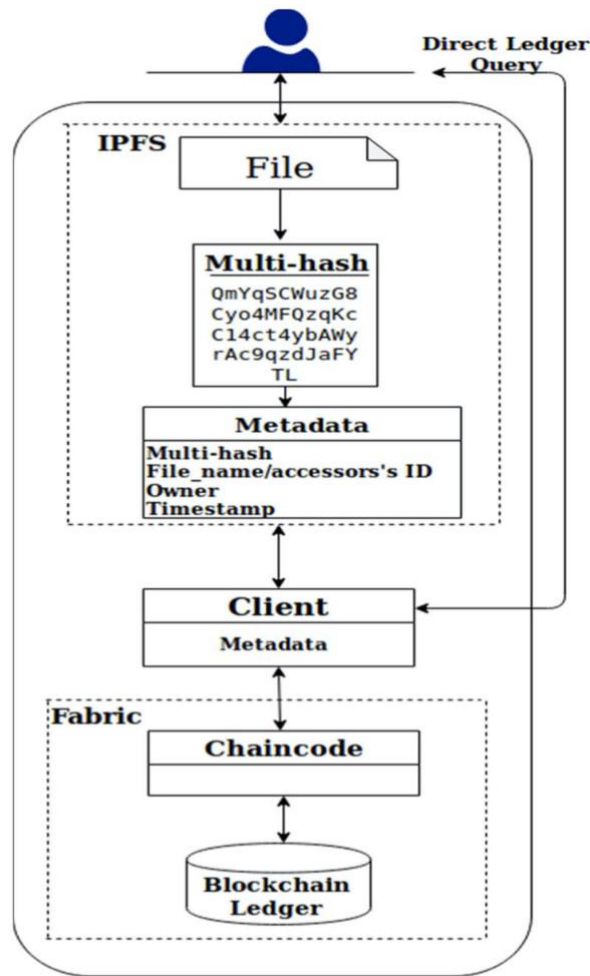


圖 3.2：BlockIPFS 之系統流程圖

3.3 Ethereum Query Language

此論文提出了以太坊查詢語言(EQL，Ethereum Query Language)，一種允許用戶通過編寫類似 SQL 的查詢從區塊鏈中檢索信息的查詢語言。查詢提供了豐富的語法，可以指定數據元素來搜索分散在多個記錄中的信息。EQL 使從區塊鏈中搜索、獲取、格式化和呈現信息變得更加容易。雖然提供了使用者易於查詢區塊鏈資訊的方法，但還缺乏取得智能合約內部資訊的功能。下圖是 EQL 對 Block 查詢的語法範例。

```
1 SELECT block.parent.number, block.hash,  
    block.timestamp, block.number,  
    block.amountOfTransactions  
2 FROM ethereum.blocks AS block  
3 WHERE block.timestamp BETWEEN date('2016-01-01')  
    AND now() AND block.transactions.size >10  
4 ORDER BY block.transactions.size  
5 LIMIT 100;
```

圖 3.3：EQL 語法範例

3.4 Etherscan API

Etherscan，是一個 Ethereum Block Explorer，可以查看區塊鏈上所有發生的交易、交易狀態或是查詢 ETH 錢包餘額等功能。其中，Etherscan 也能瀏覽相關 ERC20、ERC721 合約之代幣價格、相關交易以及代幣持有者等。該網站也提供了統計圖表和數據，進而分析供應量的增長、代幣價格的漲幅或是交易頻率等服務。

第四章 系統設計實作

本研究依前章所述之相關研究作為延伸，於以太坊區塊鏈平台之智能合約實作數位貨幣溯源與搜索服務，實踐合約導向程式語言 Solidity [14]建構資料儲存、資料搜索與資料追溯的機制，使用者能直接對於該智能合約進行搜索。透過智能合約結合跨鏈技術作為提供區塊鏈交易溯源與搜索之功能，利於後續搜索區塊鏈之實務應用發展的需求之參考。

本章共分為五小節，第一小節闡述數位貨幣溯源服務之系統架構；第二小節介紹 Oraclize(Provable)服務與 Ethereum-Bridge；第三小節說明擷取之交易資料處理方式；第四小節描述 Query 之功能；第五小節展示系統介面與操作流程。

4.1 實作系統架構

本實驗架構主要由以太坊公有鏈(Mainnet)與私有鏈(Private)、多位使用者、一個 Time Oracle、以及建立服務於私有鏈之智能合約並結合 Oraclize(Provable)服務、以及提供使用者 Web UI 操作本服務，圖 4.1 為實作系統架構圖。

為實現數位貨幣溯源之服務於以太坊區塊鏈，使用 go-ethereum (geth) [15]工具建立數個節點(enode)，其中一個節點架設 NodeJS [16]伺服器負責區塊鏈與使用者之溝通。利用以太坊智能合約設計其系統，其智能合約再透過 Oraclize(Provable)服務以利用 Etherscan 提供之 API 取得相關歷史交易資訊，並將其資訊儲存至合約當中，以利使用者之後續查詢。

Time Oracle 為獨立於以太坊區塊鏈外的 NodeJS 伺服器，每隔一段時間，負責觸發合約中之 Oraclize(Provable)服務，以調用 API 獲取公有鏈(Mainnet)上之交易記錄，以確保數位貨幣追溯交易之功能不會中斷。

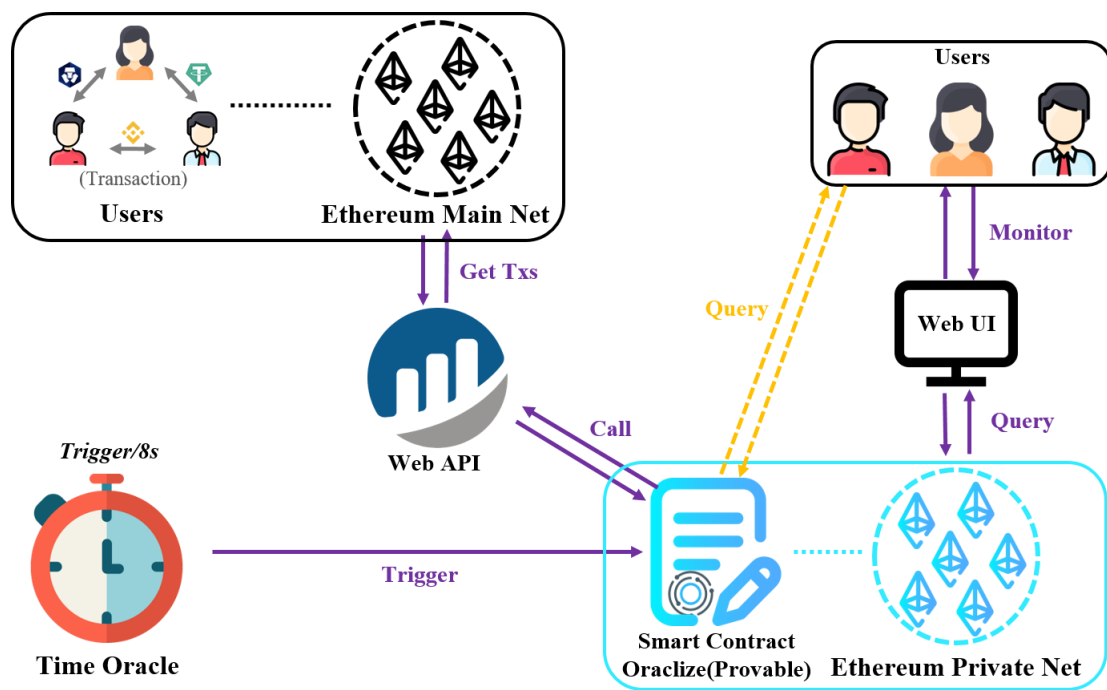


圖 4.1：實作系統架構圖

4.2 Oraclize 與 Ethereum-Bridge 介紹

4.2.1 Oraclize

Oraclize(Provable)是一個兼容以太坊智能合約的數據傳送服務，能夠自定義智能合約，在區塊鏈與真實世界之間建立一可信的通道，在保證可信的情況下，使其可透過 API 取得以太坊外部真實世界的資訊。

```

1. contract tokenTracer is usingProvable, Parser {
2.     // oraclize results
3.     function __callback(bytes32 myid, string memory _result) public {
4.         if (msg.sender != provable_cbAddress()) revert();
5.         // 更新合約餘額
6.         tracerBalance = address(this).balance;
7.
8.         oraclizeIsRunning = false;
9.         // 檢查是否有回傳值
10.        if (bytes(_result).length != 0) {

```

```

11.         savingTx(_result);
12.     }
13. }
14.
15. // call oraclize
16. function traceTx() payable public {
17.     uint gasLimit = 50000000;
18.     oraclizeIsRunning = true;
19.
20.     // 設定為每次 Oraclize 取得的交易筆數[:Count]
21.     string memory apiStr1 = "json(https://api.etherscan.io/api?module=logs&action=getLogs&fromBlock=";
22.     string memory apiStr2 = "&toBlock=latest";
23.     string memory apiStr3 = "&address=0x";
24.     string memory apiStr4 = "&topic0=0xddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef";
25.     string memory apiStr5 = "&apikey=HTI3IX924Z1IBXIIN4992VRAPKHJI149AX).result[";
26.     string memory apiStr6 = "][transactionHash, blockNumber, timeStamp, topics, data]";
27.     string memory apiUrl = string(abi.encodePacked(apiStr1, uint2str(syncBlockHeight), apiStr2, apiStr3, parseAddressToString(tokenContract), apiStr4, apiStr5, uint2str(syncIndex), ":", uint2str(syncIndex + 50), apiStr6));
28.     provable_query("URL", apiUrl, gasLimit);
29. }
30. }

```

如程式碼所示，智能合約調用了 `provable_query` 為 `oraclize(Provable)` 提供之方法，並於區塊鏈外調用這個 `query`，得到結果之後，則會調用智能合約中的 `callback` 功能將結果作為參數傳入。有了 `Oraclize(Provable)` 服務，當以太坊內部需要外部資訊時，只需使用 API 呼叫，並寫好相對應之 `callback` 方法即可。

4.2.2 Ethereum-Bridge

本實驗環境為以太坊私有鏈，我們必須使用 Ethereum-Bridge [17] 作為 Oraclize(Provable)對外部世界溝通的橋樑。啟動 Ethereum-Bridge 後，它則會開始監聽鏈上 Oraclize(Provable)發出的 Event。作為橋樑的角色，每當它接收到 Event 時，於區塊鏈外部執行該 API 請求，並將最終結果傳回該智能合約中。

4.3 資料處理方式

透過 oraclize(Provable)服務所取得之區塊鏈外部資訊後，我們必須將資料分析後再將其儲存至智能合約中。於我們所開發之智能合約中繼承之「jsmnSol」功能(此工具為 GitHub 上之開源專案，並以此進行資料處理，且為 Solidity 語言所開發之 JSON parser) [18]，以利處理從區塊鏈外部傳回之資訊內容。

```
1. contract tokenTracer is usingProvable, Parser {
2.     bytes32[] transactionHash;
3.     address[] sender;
4.     address[] receiver;
5.     uint[] value;
6.     uint[] blockNumber;
7.     uint[] timeStamp;
8.
9.     // 取得已儲存之交易筆數
10.    uint public transactionCount;
11.
12.    mapping(bytes32 => bool) private isExist;
13.
14.    function savingTx(string memory json) internal {
15.        uint returnValue;
16.        JsmnSolLib.Token[] memory tokens;
17.        uint actualNum;
18.
19.        // (returnValue, tokens, actualNum) = JsmnSolLib.parse(json, 9);
20.        (returnValue, tokens, actualNum) = JsmnSolLib.parse(json, 450);
21.    }
```

```

22.         // 每次 Oraclize 取回之交易筆數
23.         for (uint i = 0; i < 50; i++) {
24.             JsmnSolLib.Token memory a = tokens[1 + 8*i];
25.             bytes32 _transactionHash = parseStringTo32Bytes(JsmnSolLib.getBytes(
                json, a.start, a.end));
26.             // 避免重複紀錄同一筆交易
27.             if (!isExist[_transactionHash] && _transactionHash != "") {
28.                 isExist[_transactionHash] = true;
29.                 transactionHash.push(_transactionHash);
30.                 JsmnSolLib.Token memory b = tokens[2 + 8*i];
31.                 uint _blockNumber = parseHexToUint256(JsmnSolLib.getBytes(
                    json, b.start, b.end));
32.                 blockNumber.push(_blockNumber);
33.                 JsmnSolLib.Token memory c = tokens[3 + 8*i];
34.                 timeStamp.push(parseHexToUint256(JsmnSolLib.getBytes(json, c
                    .start, c.end)));
35.                 JsmnSolLib.Token memory d = tokens[6 + 8*i];
36.                 sender.push(parseAddr(subString(JsmnSolLib.getBytes(json, d.
                    start, d.end), 26, 66)));
37.                 JsmnSolLib.Token memory e = tokens[7 + 8*i];
38.                 receiver.push(parseAddr(subString(JsmnSolLib.getBytes(json,
                    e.start, e.end), 26, 66)));
39.                 JsmnSolLib.Token memory f = tokens[8 + 8*i];
40.                 value.push(parseHexToUint256(JsmnSolLib.getBytes(json, f.sta
                    rt, f.end)));
41.
42.                 // 更新下回開始搜尋之 blockNumber, 需避免同一 block 有多筆交易
43.                 syncBlockHeight = _blockNumber;
44.                 syncIndex = 0;
45.             } else if (_transactionHash == "") {
46.                 syncBlockHeight = realBlockHeight;
47.                 syncIndex = 0;
48.                 break;
49.             }
50.         }
51.         if (transactionCount == transactionHash.length) {
52.             syncIndex += 50;
53.         }

```

```
54.         transactionCount = transactionHash.length;
55.     }
56. }
```

4.4 Query 功能介紹

本實驗為實現數位貨幣溯源之服務，為此我們也提供了 Query 功能，以利使用者可對儲存於智能合約中之數位貨幣歷史交易進行搜索。於該智能合約中擁有「區塊範圍搜索」與「時間範圍搜索」兩大功能，以及針對交易發送方或接收方進行查找，並於智能合約實作 Binary Search 演算法提供使用者查詢速度。

4.5 系統介面與操作流程

為了讓使用者容易操作其去中心化數位貨幣交易記錄與查詢服務，本研究也為此 Dapp 設計其介面，提升使用時之便利性。

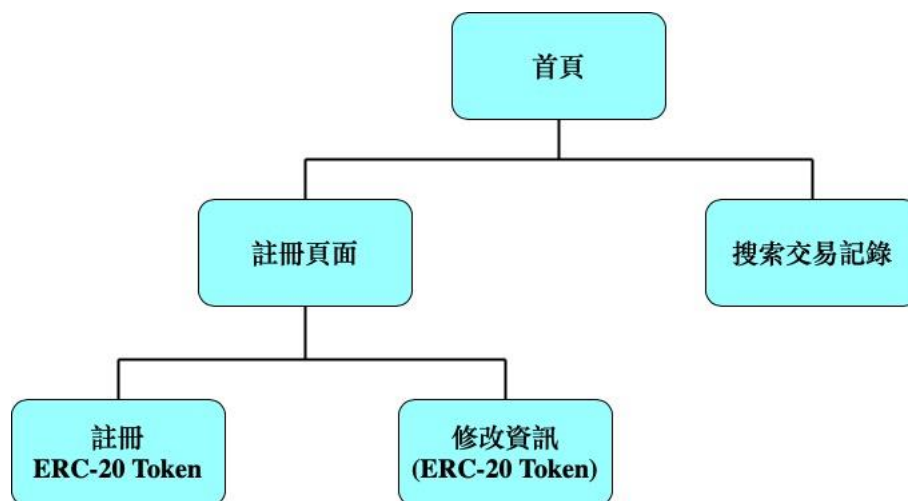


圖 4.2：服務介面之網頁架構圖

以下將分別介紹「數位貨幣交易記錄與查詢服務」功能，實作示範註冊數位貨幣之合約位址與相關資訊、修改數位貨幣之相關資訊和數位貨幣之交易記錄查找的實作結果做說明：

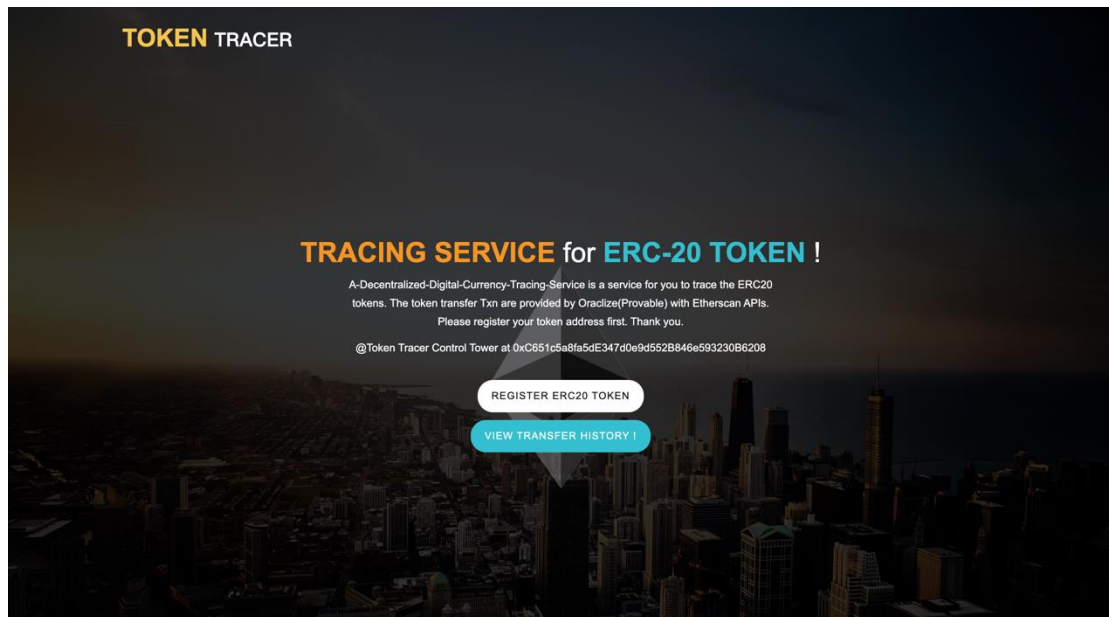


圖 4.3：數位貨幣交易記錄與查詢服務之首頁畫面

4.5.1 註冊數位貨幣之合約位址與相關資訊

本節將說明使用數位貨幣交易記錄與查詢服務之流程，其註冊之操作流程簡述如下：

(1)透過首頁之「REGISTER ERC20 TOKEN」之按鈕，進入註冊服務主頁，如下圖。

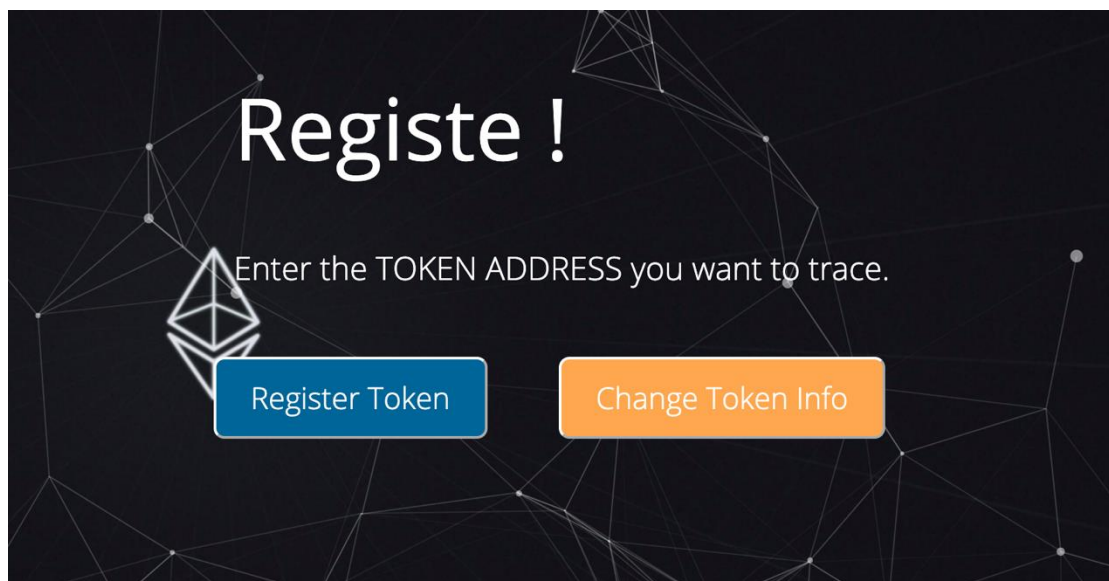
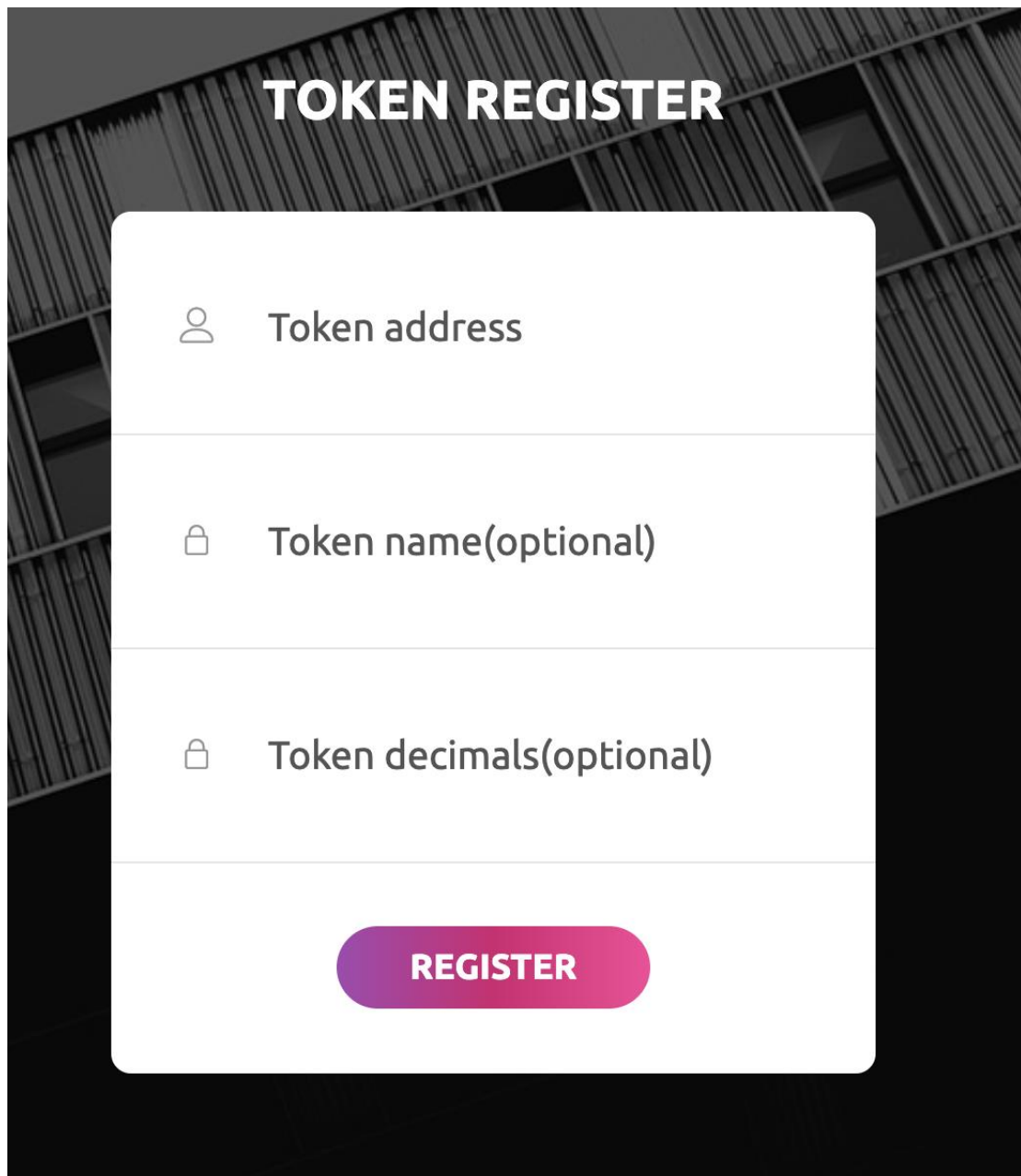



圖 4.4：註冊主頁


(2)按下註冊主頁上之「Register Token」之按鈕，進入代幣註冊頁面，如下圖。




TOKEN REGISTER



Token address



Token name(optional)



Token decimals(optional)

REGISTER

圖 4.5：代幣註冊之畫面

(3)填寫欲進行數位貨幣交易記錄與查詢服務之 ERC20 代幣之合約位址、名稱和單位並送出，其完成註冊服務之畫面如下圖。

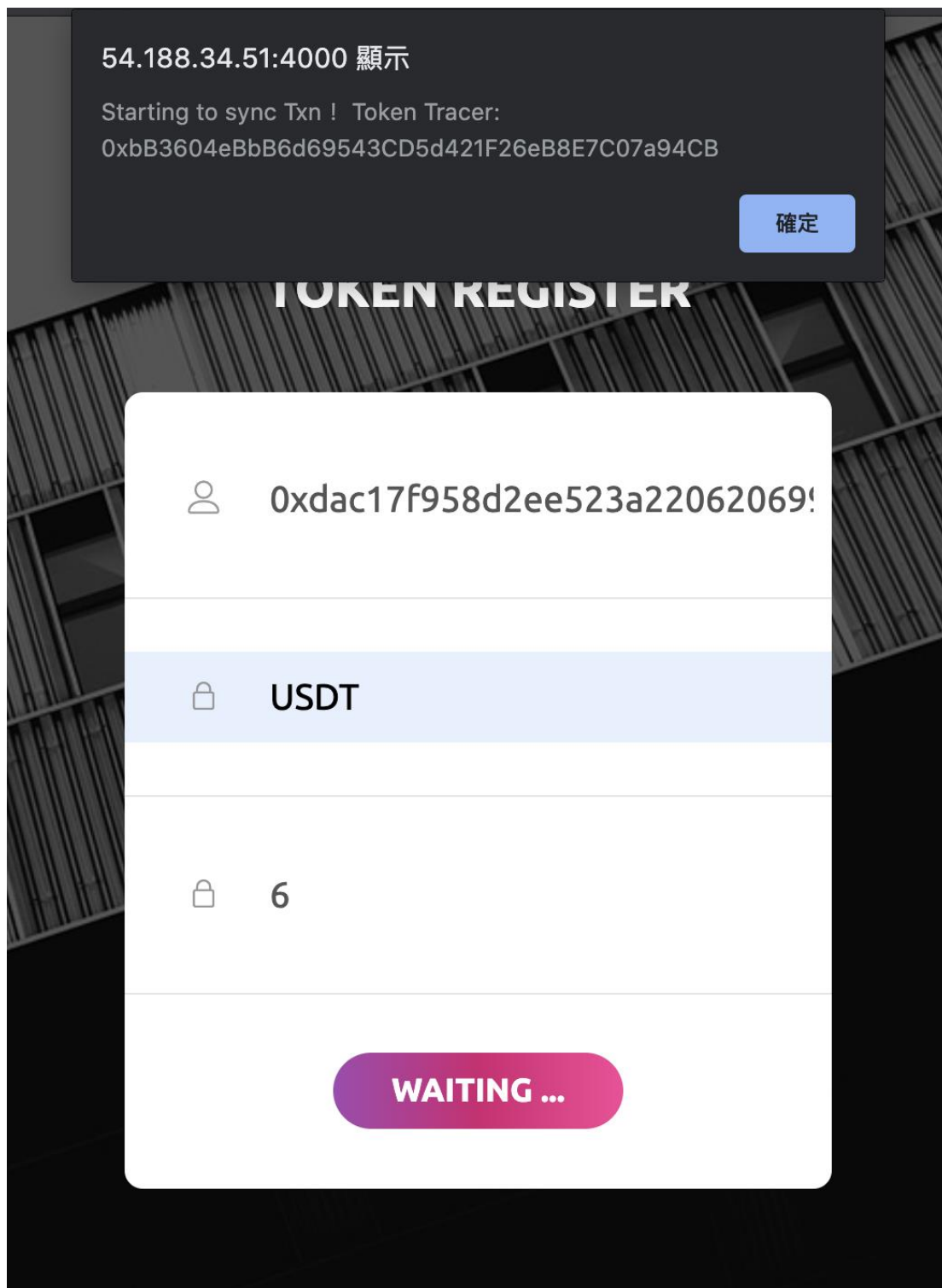


圖 4.6：完成註冊服務之畫面

(4)註冊完成後，可至交易查找之頁面進行確認，如下圖。

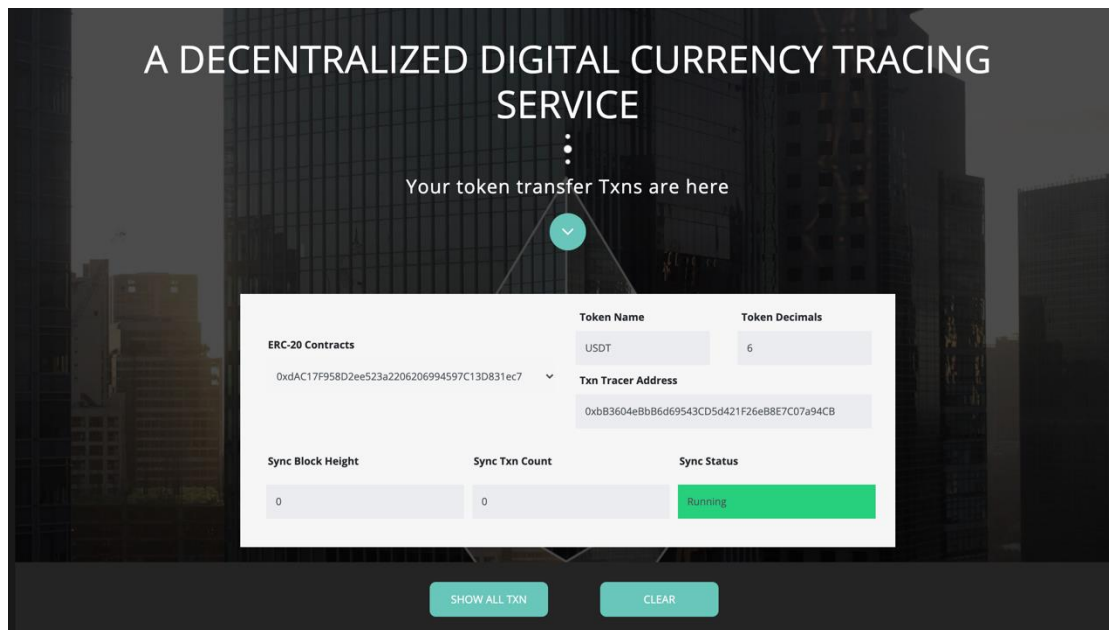


圖 4.7：查詢頁面確認註冊完成之畫面

4.5.2 修改數位貨幣之相關資訊

使用者完成註冊服務後，仍可針對代幣之名稱與單位進行變更，其操作流程簡述如下：

(1)透過註冊主頁上之「Change Token Info」之按鈕進入後，顯示代幣資訊修改頁面，並將其內容更新後按下送出即可，如下圖。

Change Token Info

ERC-20 Tokens

0xdAC17F958D2ee523a22062069 ✓

Name

USDT

Decimal

6

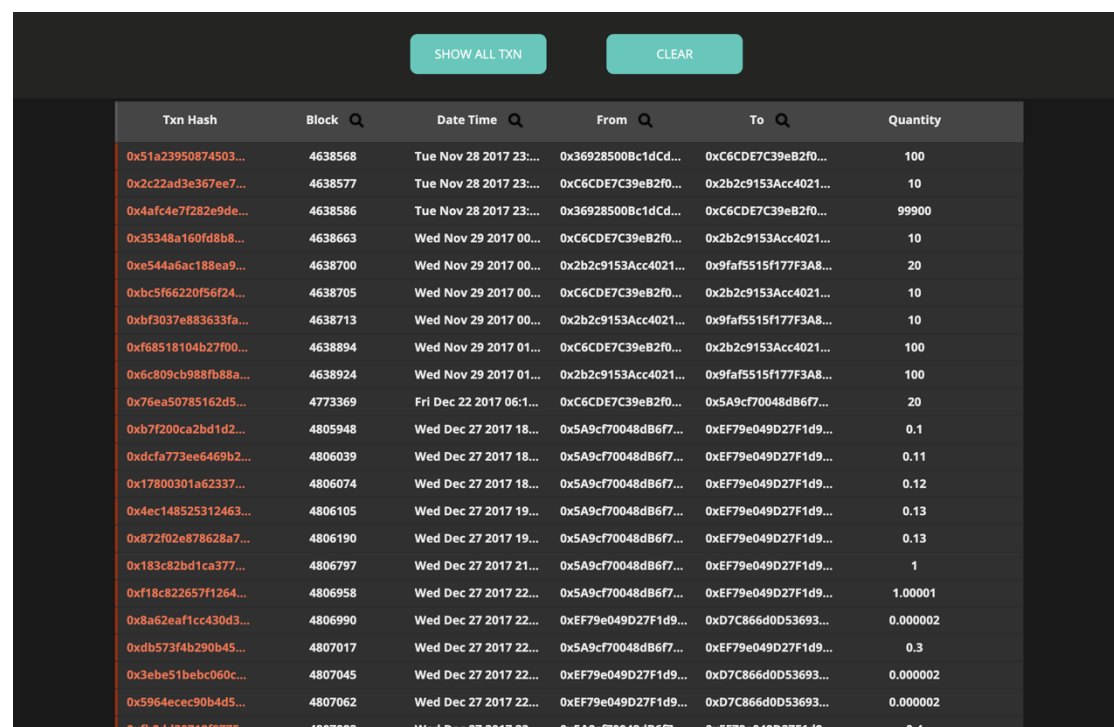
CONFIRM

圖 4.8：修改代幣資訊之畫面

4.5.3 數位貨幣之交易記錄查找

使用者完成註冊後，服務則開始同步其數位貨幣之交易記錄，使用者即可針對當前所同步之交易進行查找，也可額外對於「區塊範圍」、「日期範圍查」、「發送方位址」和「接收方位址」等功能進行交易篩選，其功能說明如下：

(1)交易查找：顯示目前已同步之所有交易紀錄，如圖 4.9 所示。使用者也將鼠標移到其中一列表項目中，即可查看其交易之詳細資訊，如圖 4.10 所示。



<div>SHOW ALL TXN</div> <div>CLEAR</div>						
Txn Hash	Block	Date Time	From	To	Quantity	
0x51a23950874503...	4638568	Tue Nov 28 2017 23:...	0x36928500Bc1dCd...	0xC6CDE7C39eB2f0...	100	
0x2c22ad3e367ee7...	4638577	Tue Nov 28 2017 23:...	0xC6CDE7C39eB2f0...	0x2b2c9153Acc4021...	10	
0x4afc4e7f282e9de...	4638586	Tue Nov 28 2017 23:...	0x36928500Bc1dCd...	0xC6CDE7C39eB2f0...	99900	
0x35348a160fd8b8...	4638663	Wed Nov 29 2017 00:...	0xC6CDE7C39eB2f0...	0x2b2c9153Acc4021...	10	
0xe544a6ac188ea9...	4638700	Wed Nov 29 2017 00:...	0x2b2c9153Acc4021...	0x9faf5515f177F3A8...	20	
0xbc5f66220f56f24...	4638705	Wed Nov 29 2017 00:...	0xC6CDE7C39eB2f0...	0x2b2c9153Acc4021...	10	
0xbf3037e883633fa...	4638713	Wed Nov 29 2017 00:...	0x2b2c9153Acc4021...	0x9faf5515f177F3A8...	10	
0xf68518104b27f00...	4638894	Wed Nov 29 2017 01:...	0xC6CDE7C39eB2f0...	0x2b2c9153Acc4021...	100	
0x6c809cb988fb88a...	4638924	Wed Nov 29 2017 01:...	0x2b2c9153Acc4021...	0x9faf5515f177F3A8...	100	
0x76ea50785162d5...	4773369	Fri Dec 22 2017 06:1...	0xC6CDE7C39eB2f0...	0x5A9cf70048dB6f7...	20	
0xb7f200ca2bd1d2...	4805948	Wed Dec 27 2017 18:...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	0.1	
0xdcfa773ee6469b2...	4806039	Wed Dec 27 2017 18:...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	0.11	
0x17800301a62337...	4806074	Wed Dec 27 2017 18:...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	0.12	
0x4ec148525312463...	4806105	Wed Dec 27 2017 19:...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	0.13	
0x872f02e878628a7...	4806190	Wed Dec 27 2017 19:...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	0.13	
0x183c82bd1ca377...	4806797	Wed Dec 27 2017 21:...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	1	
0xf18c822657f1264...	4806958	Wed Dec 27 2017 22:...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	1.00001	
0x8a62ea1cc430d3...	4806990	Wed Dec 27 2017 22:...	0xEF79e049D27F1d9...	0xD7C866d0D53693...	0.000002	
0xdb573f4b290b45...	4807017	Wed Dec 27 2017 22:...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	0.3	
0x3ebe51bec060c...	4807045	Wed Dec 27 2017 22:...	0xEF79e049D27F1d9...	0xD7C866d0D53693...	0.000002	
0x5964ecec90b4d5...	4807062	Wed Dec 27 2017 22:...	0xEF79e049D27F1d9...	0xD7C866d0D53693...	0.000002	
0xfb2dd30718f8775...	4807082	Wed Dec 27 2017 22:...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	0.4	

圖 4.9：顯示已同步之所有交易紀錄之畫面

SHOW ALL TXN
CLEAR

Txn Hash	Block	Date Time	From	To	Quantity
0x51a23950874503...	4638568	Tue Nov 28 2017 23:...	0x36928500Bc1dCd...	0xC6CDE7C39eB2f0...	100
0x2c22ad3e367ee7...	4638577	Tue Nov 28 2017 23:...	0xC6CDE7C39eB2f0...	0x2b2c9153Acc4021...	10
0x4afc4e7f282e9de...	4638586	Tue Nov 28 2017 23:...	0x36928500Bc1dCd...	0xC6CDE7C39eB2f0...	99900
0x35348a160fd8b8...	4638663	Wed Nov 29 2017 00:...	0xC6CDE7C39eB2f0...	0x2b2c9153Acc4021...	10
Transaction Hash: 0x35348a160fd8b86adea49ff482fd9d0185d544206fa54c31bd0e51fd4391de Block Number: 4638663 Time: Wed Nov 29 2017 00:03:30 GMT+0800 (台北標準時間) From: 0xC6CDE7C39eB2f0F0095F41570af89eFC2C1Ea828 To: 0x2b2c9153Acc4021676C165DC3e3B1c7AD1C7849D Quantity: 10					
0xe544a6ac188ea9...	4638700	Wed Nov 29 2017 00:...	0x2b2c9153Acc4021...	0x9faf5515f177F3A8...	20
0xbc5f66220f56f24...	4638705	Wed Nov 29 2017 00:...	0xC6CDE7C39eB2f0...	0x2b2c9153Acc4021...	10
0xbf3037e883633fa...	4638713	Wed Nov 29 2017 00:...	0x2b2c9153Acc4021...	0x9faf5515f177F3A8...	10
0xf68518104b27f00...	4638894	Wed Nov 29 2017 01:...	0xC6CDE7C39eB2f0...	0x2b2c9153Acc4021...	100
0xc6c809cb988fb88a...	4638924	Wed Nov 29 2017 01:...	0x2b2c9153Acc4021...	0x9faf5515f177F3A8...	100
0x76ea50785162d5...	4773369	Fri Dec 22 2017 06:1...	0xC6CDE7C39eB2f0...	0x5A9cf70048dB6f7...	20
0xb7f200ca2bd1d2...	4805948	Wed Dec 27 2017 18:...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	0.1
0xdcf773ee6469b2...	4806039	Wed Dec 27 2017 18:...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	0.11
0x17800301a62337...	4806074	Wed Dec 27 2017 18:...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	0.12
0x4ec148525312463...	4806105	Wed Dec 27 2017 19:...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	0.13
0x872f02e878628a7...	4806190	Wed Dec 27 2017 19:...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	0.13
0x183c82bd1ca377...	4806797	Wed Dec 27 2017 21:...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	1
0xf18c822657f1264...	4806958	Wed Dec 27 2017 22:...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	1.00001

圖 4.10：詳細交易內容之畫面

(2)區塊範圍篩選：可訂定其區塊範圍篩選交易紀錄，如圖 4.11 與圖 4.12 所示。

SHOW ALL TXN
CLEAR

Please enter the block height to query

fromBlock :

toBlock :

Txn Hash	Block	Date Time	From	To	Quantity
0x51a23950874503...	4638568	Tue Nov 28 2017 23:...	0x36928500Bc1dCd...	0xC6CDE7C39eB2f0...	100
0x2c22ad3e367ee7...	4638577	Tue Nov 28 2017 23:...	0xC6CDE7C39eB2f0...	0x2b2c9153Acc4021...	10
0x4afc4e7f282e9de...	4638586	Tue Nov 28 2017 23:...	0x36928500Bc1dCd...	0xC6CDE7C39eB2f0...	99900
0x35348a160fd8b8...	4638663	Wed Nov 29 2017 00:...	0xC6CDE7C39eB2f0...	0x2b2c9153Acc4021...	10
0xe544a6ac188ea9...	4638700	Wed Nov 29 2017 00:...	0x2b2c9153Acc4021...	0x9faf5515f177F3A8...	20
0xbc5f66220f56f24...	4638705	Wed Nov 29 2017 00:...	0xC6CDE7C39eB2f0...	0x2b2c9153Acc4021...	10
0xbf3037e883633fa...	4638713	Wed Nov 29 2017 00:...	0x2b2c9153Acc4021...	0x9faf5515f177F3A8...	10
0xf68518104b27f00...	4638894	Wed Nov 29 2017 01:...	0xC6CDE7C39eB2f0...	0x2b2c9153Acc4021...	100
0xc6c809cb988fb88a...	4638924	Wed Nov 29 2017 01:...	0x2b2c9153Acc4021...	0x9faf5515f177F3A8...	100

圖 4.11：訂定區塊範圍篩選之畫面

Txn Hash	Block	Date Time	From	To	Quantity
0x0881d7b0df1a2ef...	5000014	Tue Jan 30 2018 21:4...	0x532110357caf435...	0x05EE546c1a62f90...	217.515476
0x270210fc6287688...	5000015	Tue Jan 30 2018 21:4...	0x404c417f95Cbb77...	0x05EE546c1a62f90...	12000002.58
0x1a34b8c7e4d5b1...	5000015	Tue Jan 30 2018 21:4...	0x86940152DfbCc1...	0x05EE546c1a62f90...	9259
0x33b81ab785f43b...	5000040	Tue Jan 30 2018 21:5...	0x876EabF441B2EE5...	0x404c417f95Cbb77...	1000000.29
0xb04476ccd9fb07c...	5000043	Tue Jan 30 2018 21:5...	0x876EabF441B2EE5...	0x404c417f95Cbb77...	700000.2
0x69084d85f394f09...	5000045	Tue Jan 30 2018 21:5...	0xA29114Bdc94a74...	0x876EabF441B2EE5...	64860
0x4546105326fabd...	5000056	Tue Jan 30 2018 21:5...	0x05EE546c1a62f90...	0xA29114Bdc94a74...	29455
0xe955d70079b3fd...	5000064	Tue Jan 30 2018 21:5...	0xA29114Bdc94a74...	0x876EabF441B2EE5...	29455
0xddb5f8544cdae4...	5000188	Tue Jan 30 2018 22:2...	0xa30D8157911ef23...	0x4a782b945500ac6...	327
0x6c0fd02f1f9c20a...	5000207	Tue Jan 30 2018 22:3...	0xa30D8157911ef23...	0x4a782b945500ac6...	2600.9299
0xc8f34419e30578...	5000217	Tue Jan 30 2018 22:3...	0x05EE546c1a62f90...	0xA29114Bdc94a74...	99990
0x0c0c4b4e8f9453...	5000241	Tue Jan 30 2018 22:3...	0xA29114Bdc94a74...	0x876EabF441B2EE5...	99990
0x9c0ae8b356ec41...	5000408	Tue Jan 30 2018 23:2...	0xa30D8157911ef23...	0xD363b77A1E2CC9...	33935.120908
0x1ccb35553f8a83a...	5000504	Tue Jan 30 2018 23:4...	0xa30D8157911ef23...	0x73C8A0410c21652...	3657.87786
0x1ccb35553f8a83a...	5000504	Tue Jan 30 2018 23:4...	0xa30D8157911ef23...	0x73C8A0410c21652...	3657.87786
0x1ccb35553f8a83a...	5000504	Tue Jan 30 2018 23:4...	0xa30D8157911ef23...	0x73C8A0410c21652...	3657.87786
0xec84c18137fa69a...	5004396	Wed Jan 31 2018 15:...	0x17Bc58b788808D...	0xa30D8157911ef23...	1690
0x71426e753e59bd...	5004495	Wed Jan 31 2018 15:...	0x05EE546c1a62f90...	0x9837a1cF2E68759...	2990
0x6b165953657c01...	5004559	Wed Jan 31 2018 16:...	0x05EE546c1a62f90...	0xB2FA68E7aa90A3...	239990
0x2d9de3dc25f334...	5004577	Wed Jan 31 2018 16:...	0xB2FA68E7aa90A3...	0x876EabF441B2EE5...	239990

圖 4.12：區塊範圍篩選結果之畫面

(3)日期範圍篩選：可訂定其日期範圍篩選交易紀錄，如圖 4.13 與圖 4.14 所示。

Txn Hash	Block	Date Time	From	To	Quantity
0x51a23950874503...	4773369	Fri Dec 22 2017 06:1...	0xC6CDE7C39eB2f0...	0x5A9cf70048dB6f7...	20
0x2c22ad3e367ee7...	4805948	Wed Dec 27 2017 18:...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	0.1
0x4afc4e7f282e9de...	4806039	Wed Dec 27 2017 18:...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	0.11
0x35348a160fd8b8...	4806074	Wed Dec 27 2017 18:...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	0.12
0xe544a6ac188ea9...	4806105	Wed Dec 27 2017 19:...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	0.13
0xb5f66220f56f24...	4806190	Wed Dec 27 2017 19:...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	0.13
0xb3037e883633fa...	4806797	Wed Dec 27 2017 21:...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	1
0xf68518104b27f00...	4806958	Wed Dec 27 2017 22:...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	1.00001
0x6c809cb988fb88a...	4806990	Wed Dec 27 2017 22:...	0xEF79e049D27F1d9...	0xD7C866d0D53693...	0.000002

圖 4.13：訂定日期範圍篩選之畫面

Txn Hash	Block	Date Time	From	To	Quantity
0xb7f200ca2bd1d2...	4805948	Wed Dec 27 2017 18...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	0.1
0xdcfa773ee6469b2...	4806039	Wed Dec 27 2017 18...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	0.11
0x17800301a62337...	4806074	Wed Dec 27 2017 18...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	0.12
0x4ec148525312463...	4806105	Wed Dec 27 2017 19...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	0.13
0x872f02e878628a7...	4806190	Wed Dec 27 2017 19...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	0.13
0x183c82bd1ca377...	4806797	Wed Dec 27 2017 21...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	1
0xf18c822657f1264...	4806958	Wed Dec 27 2017 22...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	1.00001
0x8a62eaf1cc430d3...	4806990	Wed Dec 27 2017 22...	0xEF79e049D27F1d9...	0xD7C866d0D53693...	0.000002
0xdb573f4b290b45...	4807017	Wed Dec 27 2017 22...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	0.3
0x3ebe51bec060c...	4807045	Wed Dec 27 2017 22...	0xEF79e049D27F1d9...	0xD7C866d0D53693...	0.000002
0x5964ecec90b4d5...	4807062	Wed Dec 27 2017 22...	0xEF79e049D27F1d9...	0xD7C866d0D53693...	0.000002
0xfb2dd30718f8775...	4807082	Wed Dec 27 2017 22...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	0.4
0xf8bf62cb8ab0dfc...	4810661	Thu Dec 28 2017 13...	0xEF79e049D27F1d9...	0xD7C866d0D53693...	0.195827
0x90318f2cf8dda22...	4813389	Fri Dec 29 2017 00:2...	0xC6CDE7C39eB2f0...	0x876EabF441B2EE5...	1000

圖 4.14：日期範圍篩選結果之畫面

(4)發送方位址或接收方篩選：可針對其特定發送方或接收方之位址篩選交易紀錄，如圖 4.15 與圖 4.16 所示。

Txn Hash	Block	Date Time	From	To	Quantity
0x51a23950874503...	4638568	Tue Nov 28 20...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	100
0x2c22ad3e367ee7...	4638577	Tue Nov 28 20...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	10
0x4afc4e7f282e9de...	4638586	Tue Nov 28 20...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	99900
0x35348a160fd8b8...	4638663	Wed Nov 29 2017 00...	0xC6CDE7C39eB2f0...	0x2b2c9153Acc4021...	10
0xe544a6ac188ea9...	4638700	Wed Nov 29 2017 00...	0x2b2c9153Acc4021...	0x9faf5515f177F3A8...	20
0xbc5f66220f56f24...	4638705	Wed Nov 29 2017 00...	0xC6CDE7C39eB2f0...	0x2b2c9153Acc4021...	10
0xbf3037e883633fa...	4638713	Wed Nov 29 2017 00...	0x2b2c9153Acc4021...	0x9faf5515f177F3A8...	10
0xf68518104b27f00...	4638894	Wed Nov 29 2017 01...	0xC6CDE7C39eB2f0...	0x2b2c9153Acc4021...	100
0x6c809cb988fb8a...	4638924	Wed Nov 29 2017 01...	0x2b2c9153Acc4021...	0x9faf5515f177F3A8...	100
0x76ea50785162d5...	4773369	Fri Dec 22 2017 06:1...	0xC6CDE7C39eB2f0...	0x5A9cf70048dB6f7...	20
0xb7f200ca2bd1d2...	4805948	Wed Dec 27 2017 18...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	0.1
0xdcfa773ee6469b2...	4806039	Wed Dec 27 2017 18...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	0.11
0x17800301a62337...	4806074	Wed Dec 27 2017 18...	0x5A9cf70048dB6f7...	0xEF79e049D27F1d9...	0.12

圖 4.15：特定位址篩選之畫面

Txn Hash	Block	Date Time	From	To	Quantity
0x2c22ad3e367ee7...	4638577	Tue Nov 28 2017 23:...	0xC6CDE7C39eB2f0...	0x2b2c9153Acc4021...	10
0x35348a160fd8b8...	4638663	Wed Nov 29 2017 00...	0xC6CDE7C39eB2f0...	0x2b2c9153Acc4021...	10
0xbc5f66220f56f24...	4638705	Wed Nov 29 2017 00...	0xC6CDE7C39eB2f0...	0x2b2c9153Acc4021...	10
0xf68518104b27f00...	4638894	Wed Nov 29 2017 01...	0xC6CDE7C39eB2f0...	0x2b2c9153Acc4021...	100
0x76ea50785162d5...	4773369	Fri Dec 22 2017 06:1...	0xC6CDE7C39eB2f0...	0x5A9cf70048dB6f7...	20
0x90318f2cf8dda22...	4813389	Fri Dec 29 2017 00:2...	0xC6CDE7C39eB2f0...	0x876EabF441B2EE5...	1000
0x1dfbbda58a1b32...	4951210	Mon Jan 22 2018 16:...	0xC6CDE7C39eB2f0...	0x876EabF441B2EE5...	30000000
0xce14f6760cf25ac...	4988877	Mon Jan 29 2018 00:...	0xC6CDE7C39eB2f0...	0x876EabF441B2EE5...	30000000

圖 4.16：特定位址篩選結果之畫面

4.5.4 Binary Search 效能測試

本實驗於智能合約實作 Binary Search 演算法提供使用者查詢速度。此小節我們將探討一般搜索與 Binary Search 下查詢不同區塊範圍時，觀察資料回傳時間的變化。對於此測試，我們預先儲存 100000 筆資料於智能合約當中，其儲存資料之區塊高度為 1 的有 10 筆資料、高度為 2 的有 10 筆資料，以此類推，最高的區塊高度為 10000 的有 10 資料。實驗總共進行了七種搜索測試，分別為區塊高度為 10、50、100、500、1000、5000、10000。

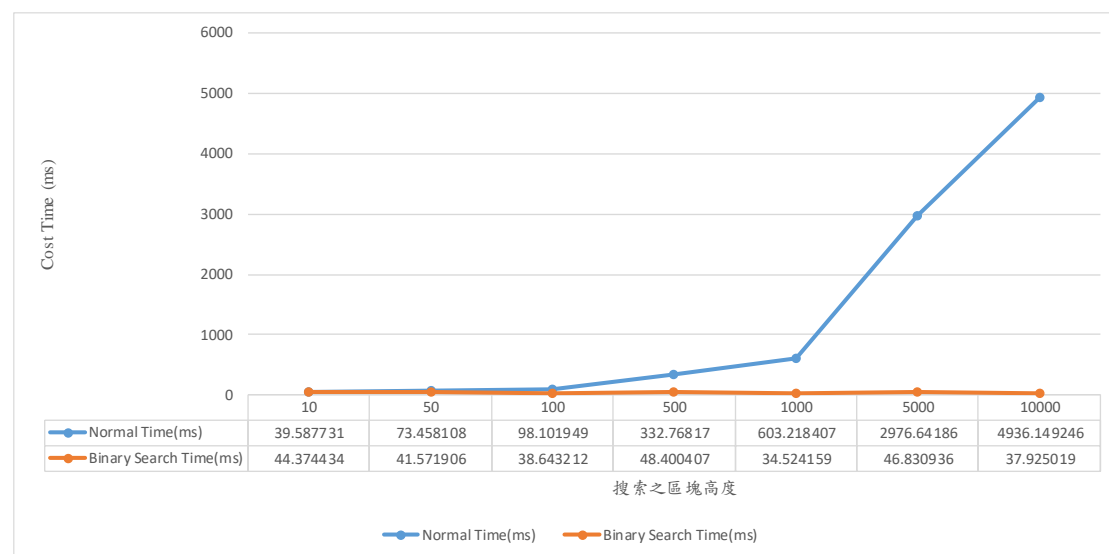


圖 4.17：比較 Binary Search 搜索效能

從圖 4.17 中可以看到使用一般搜索方式時，其查詢資料之回傳時間，在隨著搜索區塊高度的增加，回傳時間也隨之增加；然而，加入 Binary Search 搜索方法後，其實驗結果顯示，資料回傳時間不隨著搜索區塊高度的增加而增長。

第五章 討論與結論

本章節統整本研究之研究成果，共分為二小節，先是針對本研究提出整體的結論說明，以及本研究的研究貢獻；接著針對本研究之議題提出未來研究方向之建議。

5.1 討論

本研究目標在於實作以太坊區塊鏈之數位貨幣溯源服務。透過現有技術方案與相關文獻，根據溯源服務在實務上的可能作法進行研究與探討，再藉由實作與實驗，針對實務應用層面進行探討與評估，甚至找出問題並建議解決方案。

下面我們探討針對區塊鏈數位貨幣之溯源溯源服務，其影響溯源之效率的可能因素，以及實作中遭遇的難題：

- 區塊大小影響溯源效率：透過 Etherscan 所提供之 API 取得數位貨幣之歷史交易，而每筆歷史交易都有一定的容量大小，但區塊鏈中有區塊大小(Block Gas Limit)的限制，故無法一次性地將所有歷史交易同步完成。因此，本實驗架設一以太坊區塊鏈之私有鏈，並自定義其區塊大小(Block Gas Limit)，使得我們可以大量取得歷史交易，並利用批次的方式，使我們能更快速地同步所有歷史交易。
- Etherscan API 之使用次數限制影響溯源效率：我們能透過 Etherscan API 取得以太坊公鏈上之數位貨幣歷史交易，但對於一般開發者而言，使用 API 有對於每個 IP 位址每秒只能使用五次 API 的限制。
- Ethereum Bridge 處理 Oraclize(Provable)之 Query 速度影響溯源效率：由於本實驗將數位貨幣溯源服務建立於私有鏈之上，當使用 Oraclize(Provable)之 Query 時，需搭配 Ethereum-Bridge 工具監聽其智能合約所發出之特定事件(Event)，並作為中介者負責傳遞鏈外訊息。Ethereum-Bridge 並無多執行緒的

多工處理能力，因此當它接收到事件時，只能將其放入佇列中，採先進先出的方式逐一處理。

- Solidity 無提供 Json 資料型態之處理方法：數位貨幣溯源服務乃是利用以太坊區塊鏈智能合約所建立而成，將取得之歷史交易儲存於智能合約當中，以利後續之查找和追溯。不過，於智能合約獲取之歷史交易內容皆為 Json 格式之資料，而 Solidity 本身並無提供該資料的處理方式，故透過「jsmnSol」此 GitHub 上之開源工具，同時也是為 Solidity 語言所編成之 Json Parser 進行資料處理。

5.2 結論

本研究之「去中心化數位貨幣溯源服務」是一建立於以太坊區塊鏈之私有鏈平台之服務，透過私有鏈上之節點共同維護數位貨幣溯源之交易證明，以保證其公正性和安全性。該服務也利用了 Oraclize(Provable)作為可信任之交易資料提供者，為了解決智能合約取得外部數據之限制，並結合由 Etherscan 所提供之 API，以確保數據之正確性。在實務應用上，隨著區塊鏈的演進與數位貨幣的發展趨勢，可能須面臨更透明化、更嚴謹之溯源服務，甚至是政府機關之區塊鏈監管機制。因此，未來在區塊鏈數位貨幣之溯源議題時，希望能持續改進，例如：(1)數位貨幣流向追蹤：透過資料分析與追查，評估各個錢包地址是否存有洗錢之動機等。(2)納入監理機制：於每一種數位貨幣之智能合約中加入一監理者角色，並監聽該數位貨幣之所有交易內容；或是所有交易皆需經由監理者認可才能發送，藉此達到監管與治理的效果。

參考文獻

- [1] "Introductiion To Smart Contracts," [Online]. Available: <https://ethereum.org/en/developers/docs/smart-contracts/>.
- [2] "EIP-20: ERC-20 Token Standard," [Online]. Available: <https://eips.ethereum.org/EIPS/eip-20>.
- [3] "What Is DeFi?," [Online]. Available: <https://www.coindesk.com/what-is-defi>.
- [4] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system. White paper," 31 October 2008. [Online]. Available: <http://bitcoin.org/bitcoin.pdf>.
- [5] V. Buterin, "A next-generation smart contract and decentralized application platform. White Paper," 2014. [Online]. Available: <https://github.com/ethereum/wiki/wiki/White-Paper>.
- [6] "Facebook Libra," [Online]. Available: <https://libra.org/en-US/>.
- [7] "R3 Corda," [Online]. Available: <https://www.corda.net/>.
- [8] "BigchainDB," [Online]. Available: <https://www.bigchaindb.com/>.
- [9] S. Bragagnolo, H. Rocha, M. Denker and S. Ducasse, "Ethereum Query Language," in *2018 IEEE/ACM 1st International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)*, Gothenburg, Sweden, 2018.
- [10] "Etherscan," [Online]. Available: <https://etherscan.io/>.
- [11] "Introduction to SQL," [Online]. Available: https://www.w3schools.com/sql/sql_intro.asp.
- [12] "Oraclize(Provable)," [Online]. Available: <https://provable.xyz/>.
- [13] E. Nyaletey, R. M. Parizi, Q. Zhang and K.-K. R. Choo, "BlockIPFS - Blockchain-Enabled Interplanetary File System for Forensic and Trusted Data Traceability," in *2019 IEEE International Conference on Blockchain (Blockchain)*, Atlanta, GA, USA, 2019.
- [14] "Solidity," [Online]. Available: <https://solidity.readthedocs.io/en/latest/>.
- [15] "Go Ethereum," [Online]. Available: <https://geth.ethereum.org/>.
- [16] "Node.js," [Online]. Available: <https://nodejs.org/en/>.
- [17] "Ethereum-Bridge," [Online]. Available: <https://github.com/provable-things/ethereum-bridge>.
- [18] "jsmnSol," [Online]. Available: <https://github.com/chrisdotn/jsmnSol>.