Communications of the CCISA
Vol. 24 No. 4 Oct. 2018

基於區塊鏈實作分散式金融 KYC 平台

楊金祥[1]、左瑞麟[2*]
[1, 2]國立政治大學資訊科學系
[1] 104971005@nccu.edu.tw、[2]raylin@cs.nccu.edu.tw

摘要

KYC(Know Your Customer)是銀行非常重要的一項程序，除了法規上的遵循外，也同時要滿足控制客戶投資風險和防制洗錢的目的，但各銀行重複花費大量的金錢和時間成本在相同客戶 KYC 資料的收集和驗證上。另一方面，區塊鏈是一個採用密碼學及共識演算技術來確保交易資料無法被竄改的分散式帳本系統，被稱為無須中間人的信任機器。

本研究透過區塊鏈做為銀行之信任基礎，在不改變既有的 KYC 流程下，提出一套“金融 KYC 平台” 架構，讓不同銀行在提供金融服務時，每個客戶的 KYC 資料收集和驗證的程序只要進行一次。此平台不使用集中式資料庫，KYC 原始資料存放在客戶資料註冊銀行本地資料庫中，銀行透過區塊鏈驗證 KYC 資料之正確及完整性，並透過區塊鏈上紀錄的相關資訊，進行銀行間的資料的授權及同步作業。本研究實際建置了七個金融 KYC 平台的使用情境，驗證原先的各項程序假設，都能在兼顧安全和效率下完成。

關鍵詞：區塊鏈、KYC、智能合約、Dapp

The Decentralized Banking KYC Platform Based on Blockchain Technology

King-Hsian Yang[1], Raylin Tso[2*]
[1, 2]Department of Computer Science, National Chengchi University
[1]104971005@nccu.edu.tw, [2]raylin@cs.nccu.edu.tw

Abstract

KYC (Know Your Customer) is a very important procedure for banks not only for regulatory compliance, but also to meet the controls of customer investment risk and the purpose of money laundering control. It takes a lot of money and time in the process of verifying the same customer’s information for each bank. On the other hand, Blockchain is a decentralized ledger platform that uses cryptography and consensus algorithm to ensure that transactional data cannot be tampered with, and is known as a “Trust Machine”. Without

*通訊作者 (Corresponding author.)

changing current KYC procedures, this study build a "Banking KYC Platform" based on blockchain technology. And let the KYC validation be only conducted once for each customer between banks. This platform does not use centralized database. The original KYC data is stored in bank's local database. Banks verify customer KYC data's correctness and completeness through blockchain and synchronize them based on the information recorded in blockchain. This research shows the result of verifying seven KYC use cases with this Banking KYC Platform. It is a complete demonstration for reaching the goal of solving security, efficiency and cost reduction problems.

Keywords: Blockchain, KYC, Smart Contract, Dapp

壹、前言

KYC(Know Your Customer)客戶識別主要是針對金融機構想要提供金融服務時，要先清楚的了解客戶的資訊和可承受的風險，所包含的一系列相關文件和驗證的程序，並且在非常嚴格的國內和國外的法律規範下被要求確實執行，以防止洗錢和資恐的事件發生。KYC 在金融服務上，扮演非常重要的角色，不管是從客戶投資的適切性，或是從防制洗錢或是資恐的角度來看，一家銀行若沒有做好充分認識、驗證客戶資訊的動作，可能會造成客戶極大的損失，或是為公司帶來營業上的困境，有些公司因此還受到主管機關的相關罰則。但目前各家銀行卻沒有一致 KYC 的做法，不同的銀行各自儲存自定的客戶 KYC 資訊。當一個客戶在不同銀行要進行金融相關交易時，都必須分別填寫相關的評估表格，然後花費相當的時間去做資料驗證的動作。而當客戶在某家銀行有最新的資訊異動的時候，其他家銀行所儲存的 KYC 資料，可能就變成舊的資訊，失去參考價值。這樣的問題會導致銀行浪費非常多的人力和物力在 KYC 資料的確認和同步上，不僅效率不彰，導致銀行花費巨額的人力物力成本，最重要的是不正確的資訊可能還會導致金融秩序受到影響。

1.1 研究動機

KYC 目前的程序非常昂貴、耗時和沒有效率。一個大型的公司要通過必要的驗證和確認，可能需要花數十天的時間。這不只對客戶來說是一個很大等待時間，對金融機構來說，更是花費鉅大的內部成本在進行所謂的“無營收”程序。但是，若沒有做好客戶驗證的程序，不僅可能導致公司的損失，更可能遭受到鉅額的罰款。例如兆豐銀行紐約分行 2016 年就因為在防制洗錢和 KYC 上有缺失遭美國紐約州金融廳（NYDFS）重罰 1.8 億美元。因此，如何建立一個不同銀行間都信任認可的 KYC 架構，降低在進行相關

金融業務(如基金投資或跨境匯款)時，所需的 KYC 驗證程序，提升客戶的滿意度，降低銀行的驗證成本，是本研究要解決的問題。

本研究透過私有/聯盟區塊鏈作為 KYC 的底層，透過區塊鏈之去中心化、無須信任系統、不可竄改和加密安全性的特性，作為一個可信賴的分散式客戶資料儲存系統，但此系統只儲存 KYC 的 hash 和簽章資訊，以及各項資訊最新的所屬銀行，用來作為銀行間 KYC 資料之正確性及完整性驗證使用。

貳、背景與相關研究探討

2.1 區塊鏈與智能合約

區塊鏈是一個分散式的帳本系統，採用密碼技術來確保交易的正確性，不同的區塊鏈技術採用不同的共識機制。

區塊鏈的技術包含分布儲存、點對點傳輸、共識機制、密碼演算法等等不同技術的集合和應用，被認為是一個可以信賴的消除中間人的資訊傳輸技術。近年來已成為全世界國家、貨幣組織、學術界討論研究的重點，許多的產業界也紛紛地投入到這一個新技術的發展，希望能夠拿到新一輪變革主導的力量。目前，區塊鏈的應用已延伸到物聯網(IoT)、智能製造、供應鏈管理、數位資產交易等等不同領域中。

最早使用區塊鏈這個技術的例子即是比特幣的交易系統(Andreas M. Antonopoulos, 2014)，比特幣參與者們集體維護一個具時序性的分散式帳本系統。其中的每一個區塊網路之參與者都是一個節點，一套完整的帳本因為這些節點而得以保存，帳本中記錄了所有的歷史帳戶訊息，任何一個節點要發起一個交易行為都需要將交易行為訊息傳遞到區塊網路中其他的每一個節點，如此可以確保保存於所有節點上的帳本能精確地更新且驗證這一筆交易行為。區塊鏈類型一般來說可分為下列幾種，金融的相關應用以分類來看，較適合使用執行在私有/聯盟鏈，這也是目前發展的趨勢，這份研究的底層區塊鏈架構也將架設在一個私有/聯盟的環境中來實現。

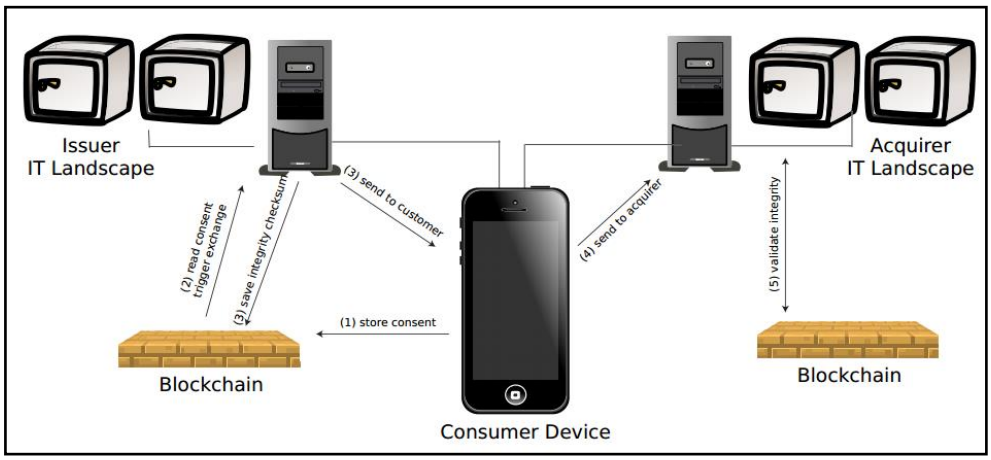
	Public	Private (Consortium)
存取	Permissionless ledger-anyone can use it and innovate with it	Permissioned ledger-only a closer group of organizations can participate
共識算法	'Proof of work' consensus	Custom consensus engine-rules set by the participating organizations
成員	Public nodes	Private nodes (closed group)
貨幣	Cryptocurrency token	Optional token
使用	Open wallet access/internet	Closed wallet access
門檻	Cost of using it is low	All running costs need to be met by the participating organizations

圖一：區塊鏈分類架構

另一方面，智能合約就是一個電腦化的交易協議，用來執行合約條款，並能在區塊鏈中執行，如此，能讓區塊鏈不僅僅存放交易的資料，也能進行資料邏輯的運算，藉以達到更複雜的商業應用。乙太坊(Ethereum)即為一個開源含智慧合約(smart contract)功能之公共區塊鏈平台，由 Vitalik Buterin 受比特幣啟發後提出。目前在上面主流的程式語言為 Solidity [2]。

2.2 KYC 相關研究

Djuri Baars 於 2016 年在 KYC on Blockchain [1]中提出一個 POC 的系統架構如下圖：

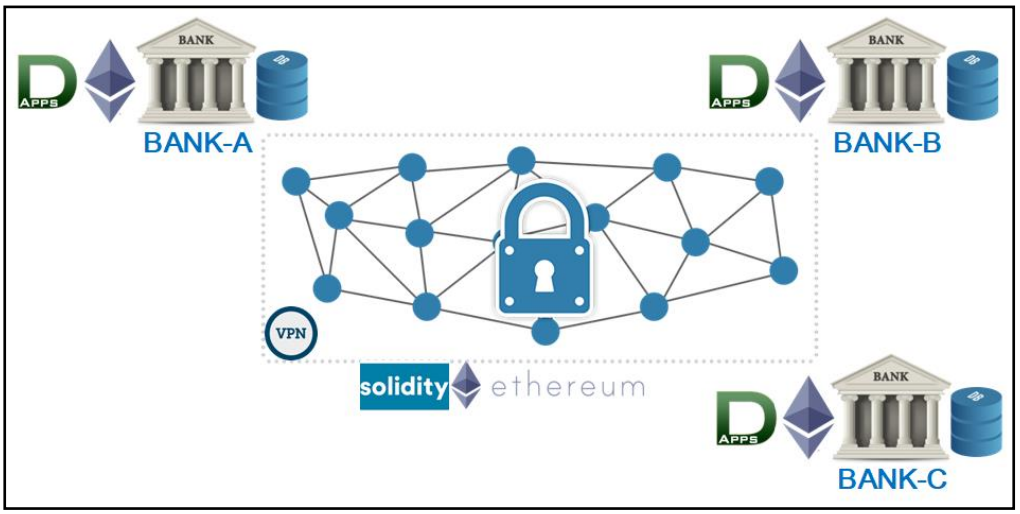


圖二： KYC on Blockchain high level architecture [1]

Djuri Baars 所提出的架構由三個部分組成：Blockchain, Server Application, Customer smartphone application.整個 KYC 的交易都透過 smartphone 來進行文件的申請和上傳，application server 會連結到銀行後端的系統，blockchain(POC 環境是跑在 bitcoin testnet 上)中存放使用者的共識資訊(consent)和資料簽章(Integrity value，每個使用者有自己的 Private Key，存放在 smartphone 裡面)。KYC 資訊的共享都是透過 smartphone 上的 application 向 Server application 進行交換已經認證過的申請文件檔案(檔案透過區塊鏈來確認沒有被異動)。這份概念性的研究簡化了整個 KYC 流程，只說明了透過 smartphone 進行第一次 onboard 和如何透過 smartphone share KYC 文件的步驟。這份論文中使用者的手機需要保留申請的 KYC 文件資料來傳送給第二家銀行時到區塊鏈確認是否有通過驗證，以及使用者的公私鑰需要保留在手機上進行資料加密簽章這件事情，我認為由於手機常常會遺失或重新安裝，資料無法進行長時間的保存，實務上並不可行，但論文提到資料各銀行自行存放的私密性及區塊鏈中僅存放 KYC 文件的簽章資訊的概念的確值得借鏡。

參、方法

本研究假設參與金融 KYC 聯盟區塊鏈的各銀行，透過虛擬私有網路通道(VPN)或專線互相串連，所有鏈上跟鏈下的訊息溝通，都是透過此網路進行。每間銀行都需執行至少一個 ethereum 的節點，結合成為一個 ethereum 的聯盟鏈。每間銀行有自己的 local DB(mongoDB)存放客戶 KYC 原始資料，並在該節點上面佈建 Dapp，作為金融 KYC 的操作平台，相關示意如圖三。



圖三：金融 KYC 平台聯盟鏈架構示意圖

另外，本研究的 KYC 資料，是以富邦投信之“客戶投資適性分析表及風險預告書”為範本做一個展示，並假設所有聯盟鏈中各銀行的 KYC 資料格式都是標準一樣的。將其包含的 KYC 資訊，原始資料存放在銀行，而 Hash/簽章等相關的資料放置於區塊鏈中以方便 Dapp 進行資料的比對。

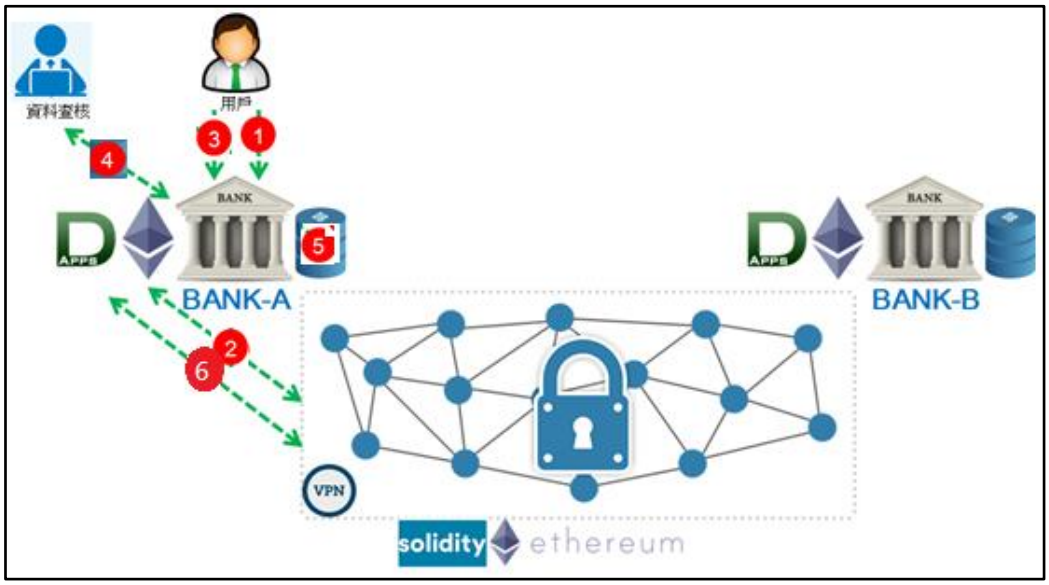
本研究設計了七個 KYC 的使用情境，來實際驗證此金融 KYC 平台，能夠在不同的假設下，達到 1. 不改變既有銀行的 KYC 作業程序。2. 不揭露所有用戶資訊。3. 不需重複驗證。4. 客戶資料異動，銀行間能同步最新資訊等四項假設前提。

3.1 使用者第一次開戶(Bank-A)：

(1). 使用者到銀行(Bank-A)櫃台，提供身分證明文件
(2). Bank-A 在 Dapp 介面，透過使用者的 ID 進區塊鏈查詢是否有資料(區塊鏈中以 Hash(ID)作為儲存使用者 KYC 資料的 index)
- ✓ 確認在區塊鏈中並沒有儲存使用者的 KYC 資訊

(3). 使用者填寫相關 KYC 表單及針對 KYC 表單勾選那些資訊可以公開
(4). Bank-A 經過銀行端標準 KYC 審核認證流程進行使用者資料驗證
(5). Bank-A 透過 Dapp 介面，將使用者的 KYC 資料存放在銀行端的資料庫中
(6). Bank-A 透過 Dapp 介面，將使用者的資料登錄到區塊鏈中
- ✓ 新增
- ✓ 儲存各屬性的 hash 值、異動銀行(Bank-A)、屬性 hash 值用銀行私鑰簽章、及該屬性是否公開查詢等相關訊息

流程示意圖如下：



圖四：金融 KYC 平台使用情境一流程示意圖

3.2 使用者在第二家銀行開戶(Bank-B) (資料已存在，不進行資料更新)

(1). 使用者到銀行(Bank-B)櫃台，提供身分證明文件

(2). Bank-B 在 Dapp 介面，透過使用者的 ID 進區塊鏈查詢是否有資料(區塊鏈中以 Hash(ID)作為儲存使用者 KYC 資料的 index)

- ✓ 確認在區塊鏈中已存在使用者的 KYC 資訊

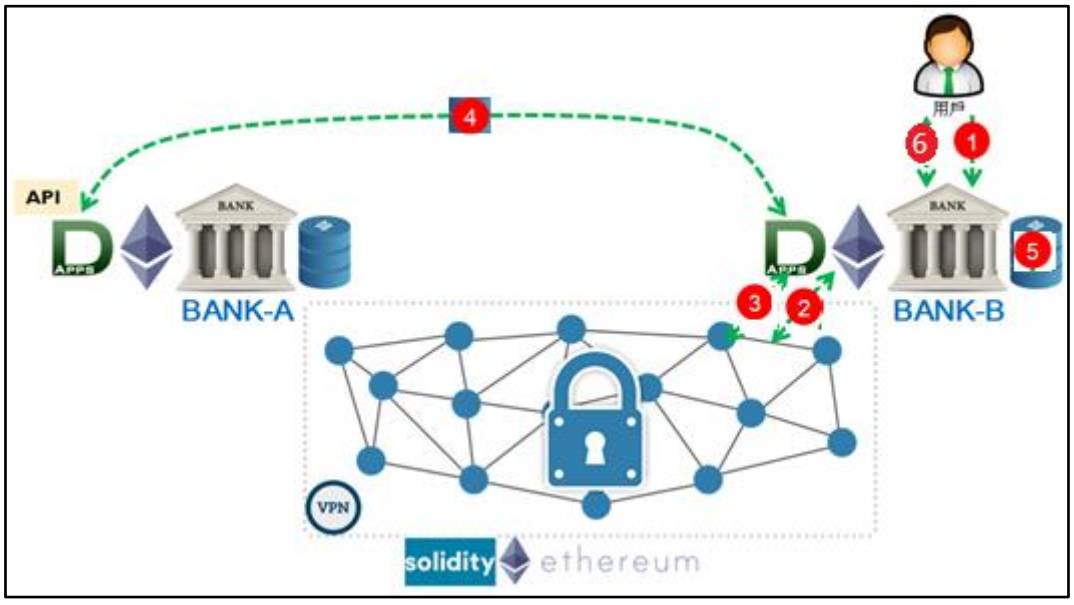
(3). Bank-B 比對區塊鏈中使用者 KYC 資訊的屬性，發現全部的屬性都是公開查詢，所有的屬性更新銀行都是 Bank-A

(4). Bank-B 透過平台紀錄的 Bank-A API 介面位址，向 Bank-A 撈取所有屬性的資料

- ✓ Bank-B 以 Hash(ID)搜尋區塊鏈確認那些屬性可以公開查詢
- ✓ Bank-B 依每個公開屬性，透過 API 向 Bank-A 撈取這些公開屬性的實際資料
- ✓ Bank-B 用私鑰簽章" 查詢屬性" (id+屬性+from 銀行+to 銀行)放到 API 中，Bank-A 收到後，用 Bank-B 的公鑰驗證此簽章是否正確，正確才允許查詢此屬性資料
- ✓ Bank-B 將回傳的的屬性值，和區塊鏈查詢的資料進行 Hash 值比對，並確認每個屬性簽章用 Bank A 公鑰解開的值，和 Hash 值相同，若相同表示資料無誤

(5). Bank B 將使用者的 KYC 資料存放在銀行端的資料庫中

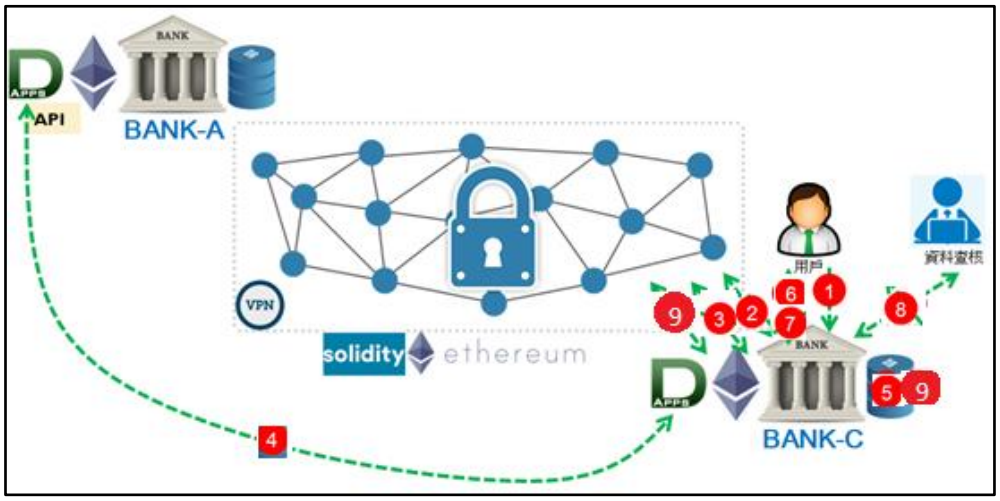
(6). 銀行提供此平台可公開之 KYC 資訊給使用者確認，使用者確認沒有資料需要更改



圖五：金融 KYC 平台使用情境二流程示意圖

3.3 使用者在第三家銀行開戶(資料進行更新)(Bank-C)

- 使用者到 Bank-C 櫃台，提供身分證明文件
- 銀行在 Dapp 介面，透過使用者的 ID 進區塊鏈查詢是否有資料(區塊鏈中以 Hash(ID) 作為儲存使用者 KYC 資料的 index)
  - ✓ 確認在區塊鏈中已存在使用者的 KYC 資訊
- 銀行查詢區塊鏈中使用者 KYC 資訊的屬性，發現全部的屬性都是公開查詢，所有的屬性更新銀行都是 Bank-A
- 銀行透過平台紀錄的 Bank-A API 介面位址，向 Bank-A 撈取所有屬性的資料
  - ✓ Bank-C 以 Hash(ID)搜尋區塊鏈確認那些屬性可以公開查詢
  - ✓ Bank-C 依每個公開屬性，透過 API 向 Bank-A 撈取這些公開屬性的實際資料
  - ✓ Bank-C 用私鑰簽章"查詢屬性"放到 API 中，Bank-A 收到後，用 Bank-C 的公鑰驗證此簽章是否正確，正確才允許查詢此屬性資料
  - ✓ Bank-C 將回傳的的屬性值，和區塊鏈查詢的資料進行 Hash 值比對，並確認每個屬性簽章用 Bank-A 公鑰解開的值，和 Hash 值相同，若相同表示資料無誤
- Bank C 將使用者的 KYC 資料存放在銀行端的資料庫中
- 銀行櫃臺回報此平台之 KYC 資訊給使用者，使用者發現某些資料需要更改
- 使用者填寫相關欲變更之 KYC 表單（某些屬性及公開資訊調整）
- Bank C 經過銀行端標準 KYC 審核認證流程進行使用者欲更新的資料驗證（含公開資訊變更）
- Bank C 透過 Dapp 介面，將使用者的資料在區塊鏈和本身的資料庫進行更新
  - ✓ 變更
  - ✓ 儲存變更屬性的 hash 值、異動銀行(Bank-C)、屬性 hash 值用銀行私鑰簽章、及該屬性是否公開查詢



圖六：金融 KYC 平台使用情境三流程示意圖

3.4 在銀行(BANK-B)購買其他金融服務，發現 KYC 資料有異動，進行相關資料變更

(1).使用者進行某些金融服務，觸發到銀行端需進行 KYC 審核動作
(2).銀行(Bank-B)在 Dapp 介面，透過使用者的 ID 進區塊鏈查詢使用者資料
(3).Bank-B 比對區塊鏈所有屬性的 Hash 值和銀行端所儲存的屬性 Hash 值
- ✓ 發現某些屬性的 Hash 值不同，確認這些屬性的更新銀行欄位紀錄為 Bank-C

(4).透過平台紀錄的 Bank-C API 介面位址，向 Bank-C 撈取屬性的資料
- ✓ Bank-B 以 Hash(ID)搜尋區塊鏈確認屬性可以公開查詢
- ✓ Bank-B 依公開屬性，透過 API 向 Bank-C 撈取這些公開屬性的實際資料
- ✓ Bank-B 用私鑰簽章”查詢屬性”放到 API 中，Bank-C 收到後，用 Bank-B 的公鑰驗證此簽章是否正確，正確才允許查詢此屬性資料
- ✓ Bank-B 將回傳的的屬性值，和區塊鏈查詢的資料進行 Hash 值比對，並確認每個屬性簽章用 Bank-C 公鑰解開的值，和 Hash 值相同，若相同表示資料無誤

(5).Bank-B 將使用者的 KYC 資料存放在銀行端的資料庫中

3.5 使用者欲變更公開資訊的欄位(Bank-C)

(1).使用者透過 Bank C 臨櫃或是網站認證，提供身分證明或網站帳密，變更公開資訊的項目
(2).銀行在 Dapp 介面，透過使用者的 ID 進區塊鏈查詢使用者資料
- ✓ 比對區塊鏈所有屬性的 Hash 值和使用者欲變更的值
- ✓ 發現這些欲變更公開資訊的屬性的更新銀行欄位紀錄為 Bank-A

(3).Bank C 經過銀行端標準 KYC 審核認證流程進行使用者欲更新的資料驗證
(4).Bank C 透過 Dapp 介面，更新使用者資訊
(5).Bank C 將使用者的資料在本身的資料庫進行更新
(6).Bank C 將使用者的資料在區塊鏈進行更新
- ✓ 變更
- ✓ 儲存變更異動銀行為 Bank-C 及該屬性是否公開查詢

礙於篇幅限制，後續流程圖將予以省略。

3.6 使用者欲關閉某銀行的戶頭(Bank-A)，不刪除區塊鏈中資料

(1).使用者到 Bank-A 櫃台，提供身分證明文件，請求刪除戶頭
(2).銀行透過 Dapp 將 Bank-A 銀行端資料庫進行客戶資訊刪除
- ✓ 亦可 follow 銀行客戶資訊留存規範，複製到其他資料庫備存

(3). 銀行透過 Dapp 將區塊鏈中 KYC 資料屬性 owner 屬於 Bank-A 的進行屬性資料刪除
- ✓ 該屬性會變成空值

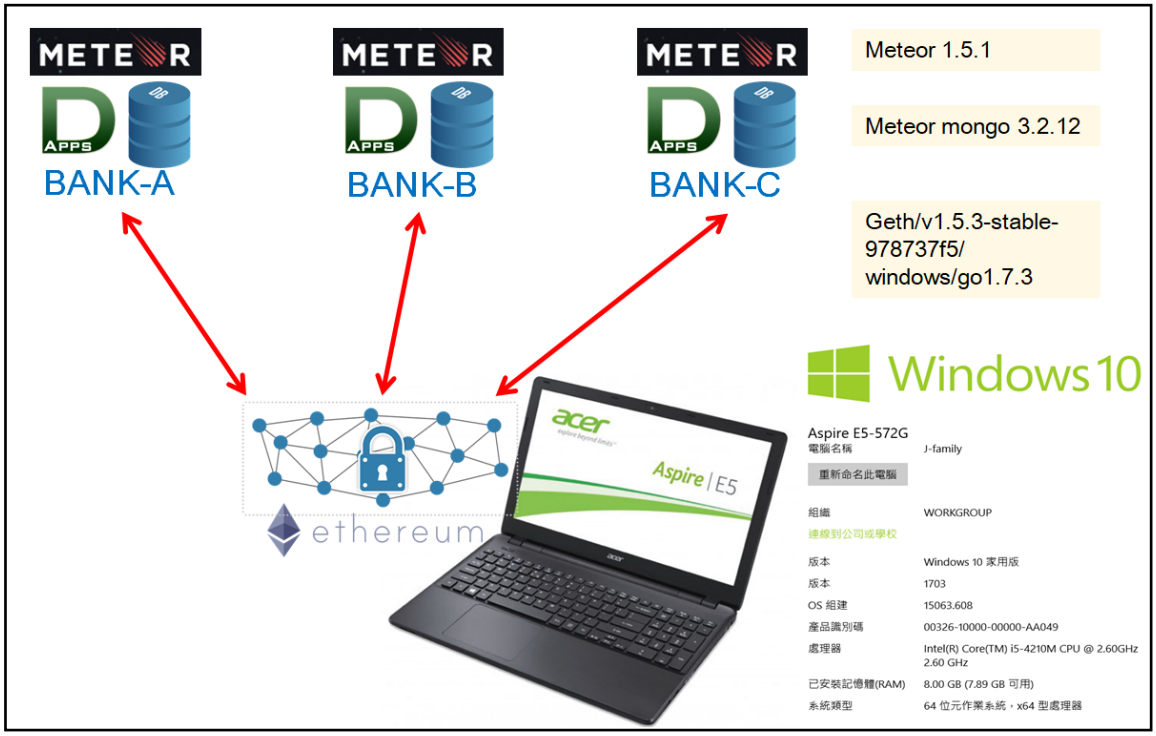
3.7 使用者欲關閉某銀行的戶頭(Bank-B)，並同時清空區塊鏈中資料

(1). 使用者到 Bank-B 櫃台，提供身分證明文件，請求刪除戶頭及申請區塊鏈資料刪除
(2). 銀行透過 Dapp 將 Bank-B 銀行端資料庫進行客戶資訊刪除
- ✓ follow 銀行客戶資訊留存規範，可能是註記已停用

(3). 銀行透過 Dapp 將區塊鏈中 KYC 資料屬性 owner 屬於 Bank-B 的進行屬性資料清空
(4). 銀行透過 Dapp 將區塊鏈中，該使用者 ID 的資料，其中 exist 屬性標記為 false
- ✓ 其他銀行已無法透過 Dapp 在區塊鏈中查詢該筆資料

肆、系統實做與情境執行結果展示

本研究以一台 Acer 的筆記型電腦，執行 geth 模擬私有聯盟區塊鏈，並在上面執行 meteor 當成是 Dapp 的平台，連結 geth 跑起來的 node.



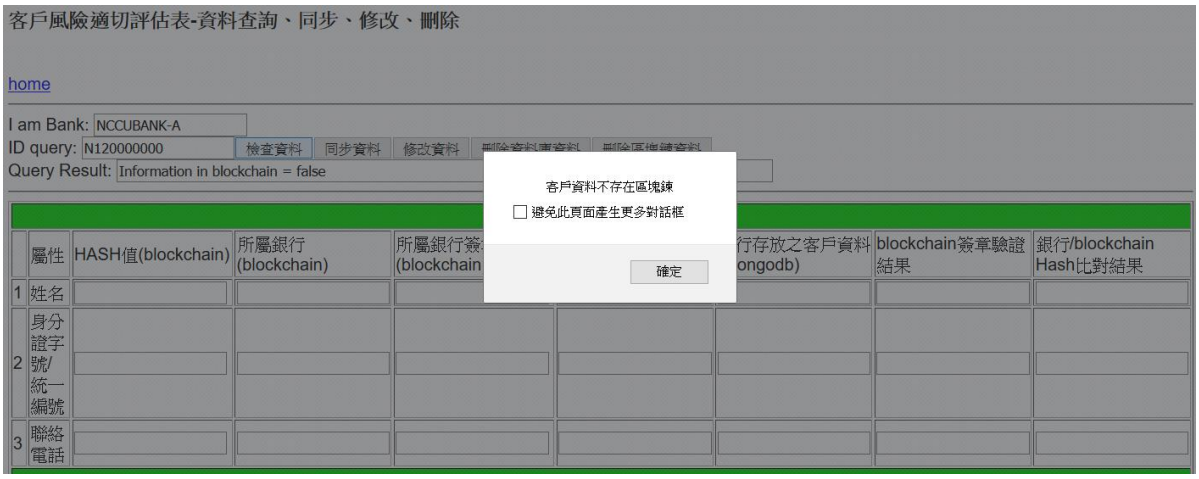
圖七：金融 KYC 平台系統執行環境示意圖

礙於篇幅限制，本篇論文僅能展示情境一之實際系統執行結果。完整版請參閱[3]。

使用情境一：使用者第一次開戶(NCCUBANK-A)

使用者(N120000000)到 NCCUBANK-A 櫃台，提供身分證明文件，經確認資料尚未在金融 KYC 區塊鏈中，填寫相關 KYC 表單及針對 KYC 表單勾選資訊可以公開與否後，銀行輸入相關 KYC 資訊於系統中。

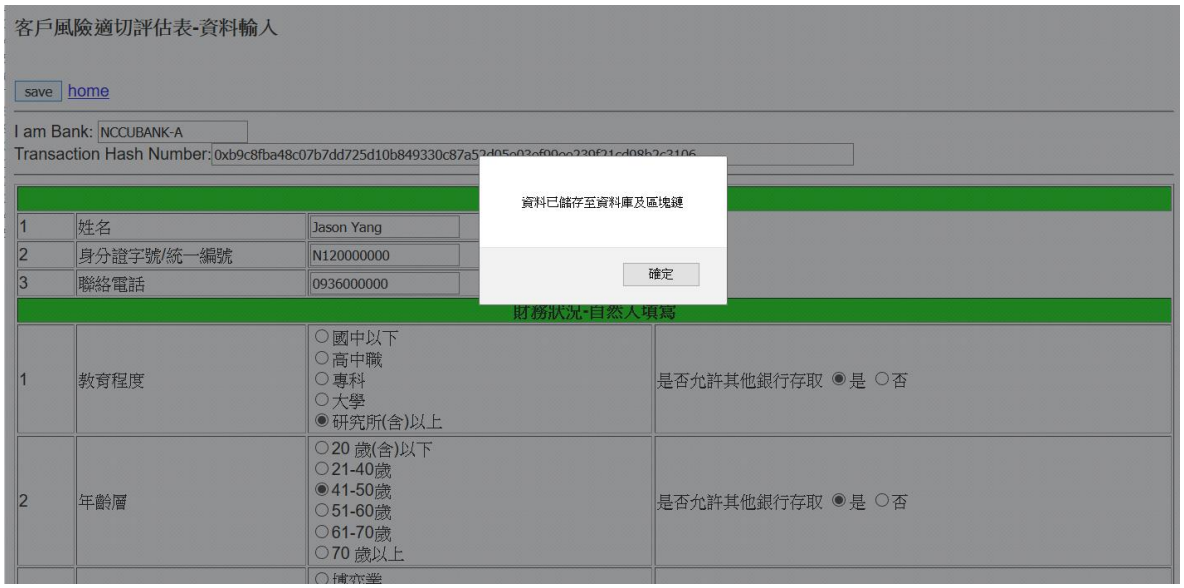
- NCCUBANK-A 透過 Dapp 介面，查詢使用者 ID： “N120000000”，確認不存在區塊鏈中



圖八：客戶資料不存在區塊鏈查詢結果圖

- 透過 Dapp 介面，輸入使用者各項 KYC 資訊

圖九：客戶 KYC 資料輸入圖



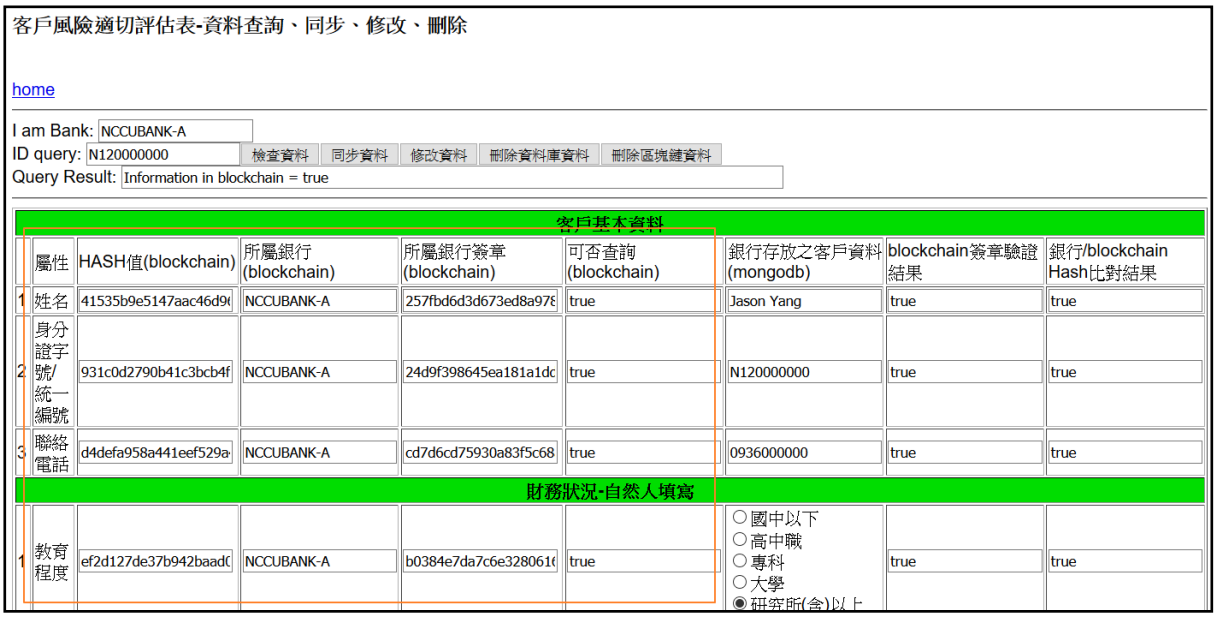
圖十：客戶 KYC 資料儲存圖

- 客戶資料儲存入資料庫中(查詢 mongo 資料庫)


```
{
        "_id" : "NaM85i4WFA7pAEhuL",
        "id" : "N120000000",
        "shaid" : "0x931c0d2790b41c3bcb4ff8234589c80eedad9f30d8bd0333aad2d1ae2d5
5faf2",
        "name" : "Jason Yang",
        "phone" : "0936000000",
        "p_education" : "5",
        "p_birthday" : "3",
        "p_career" : "3",
        "p_income_monthly" : "2",
        "p_home_income_yearly" : "2",
        "p_insurance" : "1",
        "investment_info_source" : "2",
        "investment_source" : "3",
        "investment_amount" : "3",
        "investment_type" : "2",
        "investment_knowledge_1" : "1",
        "investment_knowledge_2" : "1",
        "funding_experience" : "2",
        "investment_experience" : "2",
        "investment_purpose" : "2",
        "investment_favorite" : "2",
        "investment_risk_taken" : "2",
        "investment_of_income" : "2"
}
```


圖十一：客戶 KYC 資料資料庫查詢結果圖

- 客戶資料儲存至區塊鏈中(可透查詢確認已存在區塊鏈)



	屬性	HASH值(blockchain)	所屬銀行(blockchain)	所屬銀行簽章(blockchain)	可否查詢(blockchain)	銀行存放之客戶資料(mongodb)	blockchain簽章驗證結果	銀行/blockchain Hash比對結果
1	姓名	41535b9e5147aac46d9(	NCCUBANK-A	257fbd6d3d673ed8a97(	true	Jason Yang	true	true
2	身分證字號/統一編號	931c0d2790b41c3bcb4f	NCCUBANK-A	24d9f398645ea181a1dc	true	N120000000	true	true
3	聯絡電話	d4defa958a441eef529a	NCCUBANK-A	cd7d6cd75930a83f5c68	true	0936000000	true	true
1	教育程度	ef2d127de37b942baad(	NCCUBANK-A	b0384e7da7c6e328061(	true	○國中以下 ○高中職 ○專科 ○大學 ◉研究所(含)以上	true	true

圖十二：客戶 KYC 資料查詢存在區塊鏈及資料庫結果

伍、結論

本研究提出並實際建置驗證一個基於區塊鏈之金融 KYC 平台的作法，改善了集中資料庫式 KYC 平台隱私及安全性和效率不佳的問題，KYC 實際資料存放在銀行本地端資料庫中，並設計透過無須信任系統的區塊鏈去紀錄和追蹤每一筆 KYC 的異動資料，以確保每筆資料無法被竄改，當作是各銀行間信任的基礎，用來做為授權及是否需要在不同銀行間進行同步更新的依據。

此平台經過了各種不同可行性架構以及實際執行時作業流程遇到問題的研究，調整出本研究最終系統設計的架構，在實際系統執行驗證後，也確認能夠符合本研究一開始所進行的各項使用情境假設。

深信本研究的結果能夠成為金融界在考慮導入區塊鏈作為 KYC 資料存放平台時的參考，裡面討論到的 Solidity 變數、contract 限制，是要利用 ethereum 開發智慧合約一定會碰到問題。而 ethereum block gas limit 的限制，是在架構 ethereum 底層聯盟鏈時要特別注意的地方。本研究的分散式資料存放架構以及利用區塊鏈資訊做為各銀行間使用者授權及是否進行資料同步的機制，是一個用來解決銀行間互不信任的設計，是本研究最重要的設計理念和貢獻。

[誌謝]

本研究感謝「數位經濟推動計畫辦公室」補助科技部計畫『區塊鏈支付網路的關鍵技術與工程研發』。計畫編號 (MOST 107-2218-E-004-001-)

參考文獻

[1] D. Baars, “Towards Self-Sovereign Identity using Blockchain Technology,” *Master's thesis*, http://essay.utwente.nl/71274/1/Baars_MA_BMS.pdf, 2016.

[2] V. Tron and H. Jameson, “Ethereum Homestead Documentation,” Ethereum Community, http://www.ethdocs.org/en/latest/, 2016.

[3] 楊金祥，“應用區塊鏈之金融 KYC 平台”，*碩士論文*，國立政治大學資訊科學系，https://hdl.handle.net/11296/t97ag6，2017。