

國立臺灣大學電資工程學院資訊工程所

碩士論文

Department of Computer Science and Information Technology

College of Engineering

National Taiwan University

Master Thesis

基於 OurChain 的自主身分系統設計與實作

Design and Implementation of Autonomous Identity
System Based on OurChain

林俊佑

Jun You, Lin

指導教授：薛智文 博士

Advisor: Chih-Wen (Steven) Hsueh Ph.D.

中華民國 113 年 7 月

July, 2024

國立臺灣大學碩士學位論文

口試委員會審定書



基於 OurChain 的自主身分系統設計與實作

Design and Implementation of Autonomous Identity
System Based on OurChain

本論文係林俊佑君（R11922114）在國立臺灣大學資訊工程所完成之碩士學位論文，於民國 113 年 7 月 25 日承下列考試委員審查通過及口試及格，特此證明

口試委員：_____

（指導教授）

_____	_____
_____	_____
_____	_____
_____	_____

所 長：_____



[illegible]





摘要

隨著網路科技的快速發展，數位身分的管理與認證變得愈加重要。在傳統的身分管理系統中，中心化機構負責儲存與管理用戶的身分資料，但這樣的做法存在諸多安全風險，如資料洩漏和身分盜用。問題的根源在於中心化身分管理系統最終依賴於各個機構的自律性。當機構辜負用戶的信任，濫用資料謀取私利，或因資訊安全疏忽導致資料洩漏時，用戶往往無法即時得知，只能被動地接受損失。此時，所謂信任顯得如此蒼白無力！

我們針對身分管理系統中存在的安全風險和信任問題進行深入研究，並提出了一種全新的解決方案——自主身分識別。傳統的身分管理系統雖然已經足夠實用，但其對中心化中介者的依賴造就了更高的安全風險與更低的可信任性。本研究的核心概念是將身分識別的控制權交還給用戶，使其能夠主動地管理自己的身分資訊，從而提升系統的安全性與可信任性。

然而單純的透過分散式系統來完成身份認證無法解決多節點的信任性問題，所以我們基於 OurChain 實作了基於智能合約的解決方案，讓 OurChain「一個去中心化的區塊鏈」得以提供一個可信任的執行環境作為「自主身分識別」系統的運作基礎。

關鍵字：自主身分、管理、認證、隱私、區塊鏈





Abstract

Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Ab-
stract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract
Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract
Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract
Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract
Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract
Abstract Abstract Abstract Abstract

Keywords: Autonomous Identity, Management, Authentication, Privacy, Blockchain

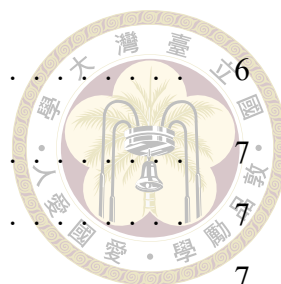


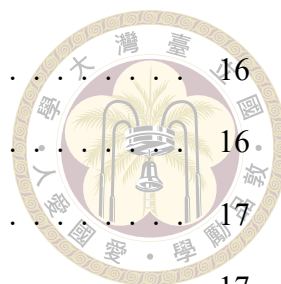


目錄

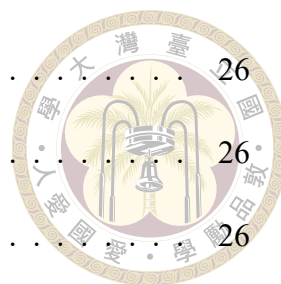
	Page
口試委員審定書	i
致謝	iii
摘要	v
Abstract	vii
目錄	ix
圖目錄	xv
表目錄	xvii
符號列表	xix
第一章 緒論	1
1.1 研究背景	1
1.2 研究目的	2
1.3 研究意義	2
1.4 論文架構	3
第二章 文獻探討	5
2.1 身分管理技術的發展	5
2.1.1 一般身分	5
2.1.1.1 定義與特徵	5
2.1.1.2 優勢與局限	6

2.1.1.3	應用實例	6
2.1.2	聯合身分	7
2.1.2.1	定義與原理	7
2.1.2.2	技術實現	7
2.1.2.3	優勢與挑戰	7
2.1.3	使用者中心的身分	8
2.1.3.1	定義與特點	8
2.1.3.2	技術原理	8
2.1.3.3	版本演進	9
2.1.3.4	優勢與局限	9
2.1.4	去中心化身分	10
2.1.4.1	定義與核心理念	10
2.1.4.2	技術基礎	10
2.1.4.3	實現方案	11
2.1.4.4	優勢與挑戰	11
2.2	近代身分驗證資安技術	12
2.2.1	無密碼登入	12
2.2.1.1	定義與原理	12
2.2.1.2	優勢與挑戰	12
2.2.1.3	實現技術	13
2.2.2	基於風險的驗證	13
2.2.2.1	原理與工作流程	13
2.2.2.2	風險因素	14
2.2.2.3	優勢與局限性	14
2.2.3	OWASP	15
2.2.3.1	OWASP Top 10	15
2.2.3.2	OWASP 身份驗證指南	15
2.2.3.3	對身份驗證安全的影響	16

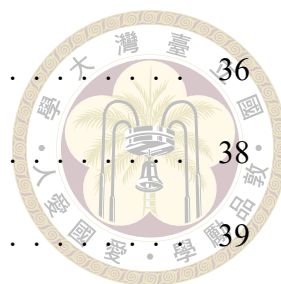




2.2.4	多因素驗證	16
2.2.4.1	驗證因素類型	16
2.2.4.2	常見 MFA 方法	17
2.2.4.3	優勢與挑戰	17
2.2.5	裝置指紋	18
2.2.5.1	工作原理	18
2.2.5.2	常見的指紋特徵	18
2.2.5.3	應用場景	19
2.2.5.4	優勢與挑戰	19
2.3	文獻探討總結	20
2.3.1	便捷性	20
2.3.2	安全性	20
2.3.3	公平性	21
2.3.4	隱私性	22
第三章	系統設計	23
3.1	系統的重要進步	23
3.1.1	別名優先機制	23
3.1.2	基於用戶時空的分析方法	24
3.1.3	基於多因素驗證的可信度測量	24
3.1.4	基於危險程度的驗證機制	24
3.1.5	極致多因素認證	24
3.1.6	半匿名機制	24
3.1.7	基於區塊鏈的自主轉址	26
3.2	系統架構	26



3.2.1	整體說明	26
3.2.2	AID Server	26
3.2.3	Wallet	26
3.2.4	Consensus Core	26
3.3	資料結構	26
3.3.1	整體說明	26
3.3.2	AID Server	26
3.3.3	Wallet	26
3.3.4	Consensus Core	26
3.4	AID 的威脅模型	26
3.5	商業流程	26
3.6	本章小結	26
第四章	系統實作	27
4.1	各模塊說明	27
4.1.1	Wallet	29
4.1.2	AID Server	30
4.1.3	Consensus Core	32
4.2	重要流程分析	33
4.2.1	AID 生成	33
4.2.1.1	用戶自主生成 AID	33
4.2.1.2	服務提供商生成 AID	34
4.2.1.3	AID 生成的通用原則	34
4.2.2	AID 中基於 RBA 的 MFA 驗證	35



4.2.3	別名在服務器上註冊	36
4.2.4	別名在服務器上登入	38
4.2.5	AID 的批次創建與綁定	39
4.2.6	AID 遷移	40
4.3	與現有方法之比較	41
4.3.1	AID 系統對 GDPR 合規性的創新解決方案	41
4.3.1.1	數據最小化和存儲限制	41
4.3.1.2	被遺忘權的實現	42
4.3.2	AID 系統的身份冒用防護與救援機制	43
4.3.2.1	身份冒用防護機制	43
4.3.2.2	創新的身份救援機制	45
4.3.2.3	安全性與可用性的平衡	46
第五章	結果與未來展望	47
5.1	結論	47
5.2	未來展望	47
	參考文獻	49
	附錄 A — 模型參數表	51
A.1	模型一	51
A.2	模型二	51
	附錄 B — BraTS 2021 分割結果圖	53
B.1	編號 0001 0050	53
B.2	編號 0051 0100	53





圖目錄





表目錄

4.1 系統各模塊支持的核心操作	27
------------------------	----





符號列表

\mathfrak{a} \mathfrak{a} 符號解釋

\mathfrak{j} \mathfrak{j} 符號解釋

\mathfrak{v} \mathfrak{v} 符號解釋





第一章 緒論

自主 (Autonomy) 的核心概念可以簡單地定義為「成為自己的主人」。在身份系統中，我們致力於實現這一目標。其中，自主身份 (Autonomous Identity, AID) 即為「成為自己身份的主人」，這一理念的實作。

1.1 研究背景

在數位時代，身份管理已成為一個日益重要且複雜的議題。隨著在線服務的普及，個人數據的價值和風險都在不斷增加。傳統的身份驗證方法，如用帳戶名和密碼，已經顯得不夠安全和便捷。同時，中心化的身份管理系統也面臨著壟斷、隱私侵犯等問題。

近年來，區塊鏈技術的發展為去中心化身份管理提供了新的可能性。自主身份系統 (AID) 正是在這一背景下應運而生。AID 旨在將身份的控制權交還給個人，使用戶能夠完全掌控自己的數位身份和個人數據。這一概念不僅有助於增強隱私保護和數據安全，還能夠簡化身份驗證流程，提高用戶體驗。

然而，實現真正的自主身份系統仍面臨諸多挑戰，包括技術實現、法律法規、用戶習慣等方面的問題。因此，深入研究自主身份系統的可行性和潛在影響具有重要的理論和實踐意義。



1.2 研究目的

本研究旨在探討自主身份 (Autonomous Identity, AID) 作為解決當前身份管理問題的潛在方案。具體而言，本文將：

1. 分析當前數位身份系統的主要挑戰和限制。
2. 探討自主身份系統的核心概念和技術基礎。
3. 評估自主身份系統在便捷性、安全性、公平性與隱私性方面的優勢。
4. 討論自主身份系統在實際應用中可能面臨的挑戰和解決方案。

1.3 研究意義

自主身份系統代表了身份系統領域的一個重要範式轉移。過去的身份管理系統主要由機構控制，每個用戶被由上而下的方式賦予一個身份。而自主身份系統則轉向由用戶自己控制自己的身份，由下而上的組織成群，形成一個又一個的身份互動節點，最終透過去中心化的方式連結一個又一個身份節點而形成更大的身份網絡。

通過賦予用戶更多對自身數位身份的自主權，AID 有潛力改變個人、組織和社會之間的互動方式。本研究的結果將為政策制定者、技術開發者和最終用戶提供寶貴的見解，有助於推動更進步的身份管理創新。

1.4 論文架構



為了要基於自主的理念設計出一個完整的身分驗證系統，我們需要先了解現有的身分驗證技術，並且對於這些技術進行分析，找出其缺點 (chapter2)。接著提出我們的系統設計，並且說明其架構、資料結構、威脅模型以及商業流程 (chapter3)。最後，我們會透過實作來驗證我們的系統設計 (chapter4)。





第二章 文獻探討

在本節中,我們不僅會分析身份管理技術的發展歷程,還會探討近代的身份驗證資安技術。最後,我們將總結自主身份系統如何透過自主的思想解決現有技術的缺點與限制。

2.1 身分管理技術的發展

身分管理技術的演進反映了數位時代對安全、隱私和用戶自主權的不斷追求。本節將詳細探討從最初的中心化身分管理到最新的去中心化身分概念的發展歷程。

2.1.1 一般身分

一般身分,也稱為中心化身分 (Centralized Identity), 是數位身分管理的初始階段。

2.1.1.1 定義與特徵

中心化身分系統由單一權威機構管理和控制,通常是大型組織或政府機構。這種系統的主要特徵包括:



- 集中式控制: 所有身分信息都存儲在中央數據庫中。
- 層級結構: 採用自上而下的管理方式。
- 高度依賴: 系統的安全性和可靠性完全依賴於中央機構。

2.1.1.2 優勢與局限

中心化身分系統的優勢在於其管理的簡單性和效率,但同時也存在顯著的局限:

- 優勢:
 - 統一管理, 便於維護
 - 標準化的身分驗證流程
- 局限:
 - 單點故障風險高
 - 用戶隱私保護不足
 - 跨系統互操作性差

2.1.1.3 應用實例

典型的中心化身分系統包括早期的企業內部身分管理系統和政府 issued ID 系統。例如, 傳統的 Windows Active Directory 就是一個典型的中心化身分管理系統。



2.1.2 聯合身分

聯合身分 (Federated Identity) 是對中心化身分系統局限性的一種回應,旨在提高系統間的互操作性。

2.1.2.1 定義與原理

聯合身分允許不同組織之間共享身分信息,使用戶能夠使用單一 set 的憑證訪問多個系統。其核心原理包括:

- 信任關係: 參與組織之間建立互信機制。
- 身分 federation: 實現不同域之間的身分信息共享。
- 單點登錄 (SSO): 用戶只需登錄一次即可訪問多個服務。

2.1.2.2 技術實現

聯合身分的技術實現主要依賴於以下協議和標準:

- SAML (Security Assertion Markup Language)
- WS-Federation (Web Services Federation)
- OAuth 2.0 (用於授權)

2.1.2.3 優勢與挑戰

聯合身分相比中心化身分有顯著進步,但仍存在一些挑戰:

- 優勢:



- 提高用戶體驗, 減少多重登錄的需求
- 降低身分管理的複雜性和成本
- 增強系統間的互操作性

- 挑戰:

- 隱私問題: 身分提供者可能獲取過多用戶信息
- 安全風險: 如果身分提供者被攻破, 可能影響多個系統
- 標準化問題: 不同聯盟間的兼容性仍然存在挑戰

2.1.3 使用者中心的身分

常見的實作為 OpenID , 是一種開放標準和去中心化的身份驗證協議, 代表了向用戶中心身份管理過渡的重要一步。

2.1.3.1 定義與特點

OpenID 允許用戶使用現有賬戶登錄多個網站, 而無需創建新賬戶。其主要特點包括:

- 去中心化: 沒有單一的身份提供者控制整個系統。
- 用戶控制: 用戶可以選擇信任的身份提供者。
- 開放標準: 任何網站都可以使用 OpenID 來登錄用戶。

2.1.3.2 技術原理

OpenID 的工作原理基於以下步驟:



1. 用戶嘗試登錄支持 OpenID 的網站（依賴方）。
2. 用戶提供其 OpenID 標識符。
3. 依賴方將用戶重定向到其 OpenID 提供者。
4. 用戶在 OpenID 提供者處進行身份驗證。
5. OpenID 提供者將用戶重定向回依賴方, 並提供身份驗證結果。

2.1.3.3 版本演進

OpenID 標準經歷了多次演進:

- OpenID 1.0 和 1.1: 最初版本, 引入基本概念。
- OpenID 2.0: 增加了安全性和功能性改進。
- OpenID Connect: 基於 OAuth 2.0 構建, 提供更豐富的身份信息。

2.1.3.4 優勢與局限

OpenID 在推動去中心化身份管理方面取得了重要進展, 但仍存在一些局限:

- 優勢:
 - 簡化用戶體驗, 減少密碼疲勞
 - 提高網站對用戶身份的信任度
 - 降低網站實現用戶認證的成本
- 局限:

- 用戶隱私仍可能受到 OpenID 提供者的影響
- 部分用戶可能難以理解 OpenID 的工作原理
- 大型身份提供者（如 Google, Facebook）可能主導市場



2.1.4 去中心化身分

去中心化身分 (Decentralized Identity) 代表了身份管理技術的最新發展方向, 旨在賦予用戶對其數字身份的完全控制權。

2.1.4.1 定義與核心理念

去中心化身分是一種允許個人或組織創建並完全擁有其數字身份的方法, 無需依賴任何中央機構。其核心理念包括:

- 自主權: 用戶完全控制自己的身份信息。
- 可移植性: 身份可以在不同系統間自由移動。
- 隱私保護: 用戶可以選擇性地披露身份信息。
- 安全性: 利用密碼學技術確保身份信息的完整性和真實性。

2.1.4.2 技術基礎

去中心化身分主要基於以下技術:

- 區塊鏈: 提供不可篡改的分布式賬本。
- 分布式標識符 (DID): 用於唯一標識去中心化數字身份。

- 可驗證證明 (Verifiable Credentials): 用於證明身份屬性。
- 零知識證明: 允許在不披露具體信息的情況下證明某一屬性。



2.1.4.3 實現方案

目前已有多個項目致力於實現去中心化身分:

- Sovrin: 基於 Hyperledger Indy 的開源身份網絡。
- uPort: 基於以太坊的身份管理解決方案。
- Microsoft ION: 基於比特幣區塊鏈的去中心化身份系統。

2.1.4.4 優勢與挑戰

去中心化身分提供了許多優勢,但在實際應用中也面臨一些挑戰:

- 優勢:
 - 增強用戶隱私和數據控制
 - 減少身份盜竊和欺詐風險
 - 提高身份信息可移植性和互操作性
- 挑戰:
 - 技術複雜性可能影響用戶採用
 - 法律和監管框架需要適應新的身份範式
 - 需要建立廣泛的生態系統支持



2.2 近代身分驗證資安技術

隨著數位技術的快速發展,身分驗證的安全性面臨著越來越多的挑戰。本節將探討本研究中會持續提到的幾種近代資安技術,包括無密碼登入、基於風險的驗證、OWASP 安全標準、多因素驗證以及裝置指紋技術。

2.2.1 無密碼登入

無密碼登入 (Passwordless Authentication) 是一種新興的身分驗證方法,旨在解決傳統密碼所帶來的安全風險和使用不便。

2.2.1.1 定義與原理

無密碼登入是指在不使用傳統密碼的情況下完成用戶身份驗證的過程。主要原理包括:

- 使用生物特徵(如指紋、面部識別)進行身份驗證
- 利用硬件安全密鑰(如 YubiKey)
- 通過手機應用或郵箱接收一次性驗證碼

2.2.1.2 優勢與挑戰

無密碼登入技術提供了多項優勢,但也面臨一些挑戰:

- 優勢:
 - 提高安全性,減少密碼相關的攻擊風險



- 改善用戶體驗, 無需記憶複雜密碼
- 降低密碼重置和管理的成本

- 挑戰:

- 需要額外的硬件或軟件支持
- 用戶習慣的改變可能需要時間
- 某些生物特徵識別技術可能引發隱私顧慮

2.2.1.3 實現技術

常見的無密碼登入實現技術包括:

- FIDO2 (Fast IDentity Online) 標準
- WebAuthn API
- 移動推送通知認證

2.2.2 基於風險的驗證

基於風險的驗證 (Risk-Based Authentication, RBA) 是一種動態調整身份驗證強度的方法, 根據每次登錄嘗試的風險級別來決定所需的驗證步驟。

2.2.2.1 原理與工作流程

RBA 系統通常遵循以下工作流程:

1. 收集上下文信息: 如 IP 地址、地理位置、設備信息等

2. 風險評估: 根據收集的信息評估當前登錄嘗試的風險級別
3. 動態調整: 基於風險評估結果, 決定是否需要額外的驗證步驟
4. 持續學習: 系統不斷學習和更新風險模型



2.2.2.2 風險因素

RBA 考慮的風險因素通常包括:

- 地理位置異常
- 設備變更
- 登錄行為模式變化
- 網絡環境 (如使用 VPN)
- 敏感操作 (如大額交易)

2.2.2.3 優勢與局限性

RBA 在提高安全性的同時也帶來了一些挑戰:

- 優勢:
 - 提供針對性的安全保護
 - 減少對低風險用戶的干擾
 - 能夠適應不斷變化的威脅環境
- 局限性:



- 可能產生誤判, 影響用戶體驗
- 需要處理大量數據, 增加系統複雜性
- 隱私問題: 需要收集和分析用戶行為數據

2.2.3 OWASP

OWASP (Open Web Application Security Project) 是一個非營利基金會, 致力於提高軟件的安全性。雖然 OWASP 不是一種特定的技術, 但它提供了重要的安全指南和最佳實踐, 對身份驗證安全有重要影響。

2.2.3.1 OWASP Top 10

OWASP Top 10 是一份關於 web 應用程序最關鍵安全風險的報告, 其中包括與身份驗證相關的多個項目:

- 身份認證失效 (Broken Authentication)
- 敏感數據洩露 (Sensitive Data Exposure)
- 失效的訪問控制 (Broken Access Control)

2.2.3.2 OWASP 身份驗證指南

OWASP 提供了詳細的身份驗證安全指南, 包括:

- 密碼策略建議
- 多因素驗證實施指南



- 會話管理最佳實踐
- 防範常見攻擊 (如跨站腳本攻擊,SQL 注入) 的方法

2.2.3.3 對身份驗證安全的影響

OWASP 的工作對提高身份驗證安全性有重要貢獻:

- 提高了開發者和組織對身份驗證安全的認識
- 提供了實用的安全實施指南
- 促進了安全最佳實踐的共享和標準化

2.2.4 多因素驗證

多因素驗證 (Multi-Factor Authentication, MFA) 是一種要求用戶提供兩種或更多驗證方式的安全機制,大大提高了身份驗證的安全性。

2.2.4.1 驗證因素類型

MFA 通常涉及以下幾類驗證因素:

- 知識因素: 用戶知道的東西,如密碼、PIN 碼
- 所有因素: 用戶擁有的東西,如手機、硬件令牌
- 固有因素: 用戶的生物特徵,如指紋、虹膜
- 位置因素: 用戶的地理位置
- 行為因素: 用戶的行為模式,如打字節奏



2.2.4.2 常見 MFA 方法

實際應用中常見的 MFA 方法包括：

- SMS 一次性密碼 (OTP)
- 移動應用程序生成的 TOTP (基於時間的一次性密碼)
- 硬件安全密鑰 (如 YubiKey)
- 生物識別 (如指紋、面部識別)
- 推送通知

2.2.4.3 優勢與挑戰

MFA 雖然大大提高了安全性,但也面臨一些挑戰：

- 優勢：
 - 顯著提高賬戶安全性
 - 減少身份盜竊和未授權訪問的風險
 - 滿足許多法規和合規要求
- 挑戰：
 - 可能增加用戶登錄的複雜性
 - 某些 MFA 方法 (如 SMS) 可能被攔截
 - 實施和維護成本可能較高



2.2.5 裝置指紋

裝置指紋 (Device Fingerprinting) 是一種通過收集設備特徵信息來唯一識別或驗證用戶設備的技術，常用於增強身份驗證和防欺詐系統。

2.2.5.1 工作原理

裝置指紋技術通過以下步驟工作：

1. 數據收集：收集設備的各種特徵信息
2. 特徵提取：從收集的數據中提取關鍵特徵
3. 指紋生成：基於特徵創建唯一的設備標識符
4. 比較與驗證：將當前指紋與已知指紋進行比較

2.2.5.2 常見的指紋特徵

裝置指紋可能包括以下特徵：

- 硬件信息：CPU 型號、內存大小、硬盤序列號等
- 軟件環境：操作系統版本、已安裝的字體、瀏覽器插件等
- 網絡特徵：IP 地址、TCP/IP 配置參數等
- 瀏覽器設置：用戶代理字符串、語言設置、時區等
- 運行時特徵：Canvas 指紋、WebGL 指紋等



2.2.5.3 應用場景

裝置指紋技術在身份驗證和安全領域有廣泛應用：

- 增強型身份驗證: 作為額外的驗證因素
- 防欺詐: 識別可疑設備或行為模式
- 持續身份驗證: 在會話期間持續驗證用戶身份
- 風險評估: 作為風險基礎驗證的一個輸入因素

2.2.5.4 優勢與挑戰

裝置指紋技術提供了獨特的安全優勢,但也面臨一些挑戰:

- 優勢:
 - 提供無需用戶交互的額外安全層
 - 可以檢測異常或可疑的設備行為
 - 有助於識別帳戶共享或未經授權的訪問
- 挑戰:
 - 隱私問題: 可能被視為對用戶隱私的侵犯
 - 準確性: 設備更新或配置變化可能影響指紋的穩定性
 - 法律合規: 在某些司法管轄區可能面臨法律限制

總結來說,近代身分驗證資安技術正朝著更安全、更便捷、更智能的方向發展。無密碼登入、基於風險的驗證、OWASP 安全標準、多因素驗證和裝置指紋等技術的結合,為應對當前複雜多變的網絡安全威脅提供了強有力的工具。



2.3 文獻探討總結

我們將從便捷性、安全性、公平性與隱私性四個方面進行分析，探討自主身份系統如何解決現有技術的缺點與限制。

2.3.1 便捷性

現有技術分析 傳統的身份驗證模式要求用戶為每個組織創建獨立的帳號，甚至在同一組織內可能需要管理多套身份方案。這種方式雖然使得組織能夠完全控制用戶身份，但也導致了以下問題：

- 用戶需要記憶多組登錄憑證，增加了管理難度。
- 不同系統間的身份信息難以互通，降低了用戶體驗。
- 身份創建過程耗時耗力，降低了效率。

AID 系統解決方案 我們提出的 AID 系統通過以下方式解決這些問題：

- 用戶管理單一身份，通過別名機制靈活地加入不同組織。
- 實現一個身份多處使用的模式，大幅提高便捷性。
- 各組織成為 AID 系統的子集，簡化了用戶的身份管理流程。

2.3.2 安全性

現有技術分析 傳統身份驗證模式下，機構建置了許多安全的身份驗證與權限管理機制讓用戶選用

- 身份驗證技術的瓶頸，



AID 系統解決方案 AID 系統通過以下機制提高安全性：

- 用戶可自主控制與限制自身權限與數據訪問範圍。
- 採用加密技術確保數據安全，減少對平台運營者的依賴。
- 分布式存儲降低了單點故障風險。

2.3.3 公平性

現有技術分析 在傳統模式下，大型組織掌握大量用戶資源，容易形成壟斷，導致：

- 用戶處於弱勢地位，難以對平台政策產生影響。
- 數據壟斷阻礙了市場公平競爭。
- 用戶難以自由選擇或轉換服務提供商。

AID 系統解決方案 AID 系統通過以下方式提升公平性：

- 每個用戶都有權利管理自己的身份，不受特定組織的限制。
- 用戶可以自主地在 AID 伺服器之間進行身份遷移。
- 通過去中心化技術，減少對單一大型平台的依賴。



2.3.4 隱私性

現有技術分析 在 GDPR 的隱私保護法規有幾項現有的身份驗證技術難以達到的要求：

- 被遺忘權，用戶有權要求系統刪除自己的數據，但是很難徹底清除。
- 數據最小化，系統僅可以存取與使用必要的數據。
- 積極授權，用戶要明確授權系統使用自己的數據，數據才會被使用。

AID 系統解決方案 AID 系統通過以下方式保護用戶隱私：

- 利用可否認權來解決被遺忘權問題，即使用戶的資料無法徹底刪除，但用戶可否認自己的資料，沒有人可以證明這些資料是屬於用戶的。
-

總結來說，AID 系統通過自主管理的方式，在保留傳統身份驗證系統的優點（如組織對用戶的管理能力）的同時，有效解決了現有技術的缺點，提高了便捷性、安全性、公平性與隱私性。



第三章 系統設計

本章節將會介紹我們設計的系統，並且說明其架構、資料結構、威脅模型以及商業流程。

3.1 系統的重要進步

為了實現用戶對其身份的完全自主權，本研究提出了許多創新的機制，希望解決既有的技術困境。

3.1.1 別名優先機制

在過去的身份驗證系統中，會要求使用者提供一個唯一的帳號作為登入的識別。然而，這樣的設計無形中讓用戶對外交出了個人名稱的自主權，連名稱都無法自主的系統，顯然難以達成本研究的設計目標。因此，我們提出讓用戶在註冊身份時優先使用自己喜歡的別名，接著透過 UUID 的機制產生唯一識別號，日常的操作中優先讓用戶使用別名作為帳號，只有當別名因為重複而難以識別時會要求用戶使用和 UUID 關聯的機制做識別。當然，何時該使用別名，何時該使用 UUID，這顯然是個複雜的問題，因此我們提出了”基於用戶時空的分析方法”試圖系統性的進行分析。



3.1.2 基於用戶時空的分析方法

提到物自體的概念..... 這樣的機制可以讓用戶在不同的情境下使用不同的身份，進而保護用戶的隱私。

3.1.3 基於多因素驗證的可信度測量

提到物自體的概念.....

3.1.4 基於危險程度的驗證機制

人為自然立法的概念..... 這樣的機制可以讓用戶在不同的情境下使用不同的身份，進而保護用戶的隱私。

3.1.5 極致多因素認證

人為自然立法的概念.....

3.1.6 半匿名機制

去中心化不代表無中心,.... 解決被遺忘權, 單向追蹤等等,.....





3.1.7 基於區塊鏈的自主轉址

3.2 系統架構

3.2.1 整體說明

3.2.2 AID Server

3.2.3 Wallet

3.2.4 Consensus Core

3.3 資料結構

3.3.1 整體說明

3.3.2 AID Server

3.3.3 Wallet

3.3.4 Consensus Core

3.4 AID 的威脅模型

3.5 商業流程

3.6 本章小結

系統如何解決所有缺點，並且保留所有優點。



第四章 系統實作

本研究已成功實現了系統的核心功能，並在受控的測試環境中進行了全面的概念驗證（Proof of Concept, PoC）。系統架構由三個關鍵模塊組成：AID Server、Wallet 和 Consensus Core

4.1 各模塊說明

本節將詳細介紹系統的核心架構, 包括 AID Wallet、AID Server 和 Consensus Core 三個主要模塊。表4.1概述了每個模塊支持的核心操作。

Table 4.1: 系統各模塊支持的核心操作

AID Wallet	AID Server	Consensus Core
AID Generation	AID Register	Set AID Owner
MFA Login	AID Login	Set AID Manager
Set User Data	MFA	Get AID Owner
Load User Data	RBA	Get AID Manager
Wallet Migration	AID Migration	

AID Wallet 模塊是用戶與系統交互的主要界面。它支持以下核心功能：

- **AID Generation:** 生成新的去中心化身份標識符。
- **MFA Login:** 實現多因素認證登錄。
- **Set User Data:** 允許用戶設置和更新存在個人設備的數據。

- **Load User Data:** 從個人設備中加載用戶數據。
- **Wallet Migration:** 支持錢包遷移功能, 確保用戶數據的可攜性。



AID Server 作為系統的中央服務器, 負責處理身份驗證和數據管理。其主要功能包括：

- **AID Register:** 註冊新的去中心化身份。
- **AID Login:** 處理用戶的登錄請求。
- **MFA:** 實現多因素認證機制。
- **RBA:** 執行基於風險的認證。
- **AID Migration:** 管理身份遷移過程。

Consensus Core 模塊負責維護系統的一致性和安全性。它提供以下關鍵操作：

- **Set AID Owner:** 設置或更新 AID 的所有者。
- **Set AID Manager:** 指定 AID 的管理者。
- **Get AID Owner:** 檢索 AID 的當前所有者信息。
- **Get AID Manager:** 獲取 AID 的管理者信息。

這三個模塊協同工作, 共同構成了一個安全、可靠且靈活的去中心化身份管理系統。AID Wallet 提供用戶友好的用戶端接口, AID Server 作為後端服務的一部分處理核心的身份驗證和數據管理任務, 最後 Consensus Core 則確保整個系統的一致性和可信度。



4.1.1 Wallet

本研究中的 Wallet 模塊是一個基於 Flutter 框架開發的跨平台網絡前端應用，為用戶提供了一個直觀且功能豐富的圖形用戶界面 (GUI)。該模塊的設計和實現不僅展示了系統的可擴展性和互操作性，還凸顯了與其他系統組件的無縫集成能力。

Wallet 應用與 AID Server 進行了深度集成，這種集成體現了系統的模塊化設計理念，主要表現在以下幾個方面：

- **身份驗證機制：**利用 AID Server 提供的去中心化身份驗證機制，實現了符合 W3C 標準的安全可靠的用戶登錄流程。這種方法不僅提高了系統的安全性，還增強了用戶隱私保護。
- **別名系統：**實現了基於別名的登錄功能，這種設計不僅增強了用戶體驗的便捷性，還提供了額外的隱私保護層，使用戶能夠在不暴露真實身份的情況下進行身份驗證。
- **去中心化身份管理：**用戶可以在應用內創建新的去中心化身份 (AID) 或註冊新的別名，這種設計體現了系統的自主性和靈活性，使用戶能夠更好地控制自己的數字身份。

Wallet 應用的核心功能模塊包括：

1. **用戶認證模塊：**實現了基於 AID Server 的安全登錄機制，支持多因素認證和風險基礎認證 (RBA)。
2. **AI 對話模塊：**集成了符合微軟 AI 聊天規範的對話界面，為用戶提供了基於自然語言處理 (NLP) 的智能交互體驗。

3. **去中心化身份管理模塊**: 允許用戶創建、管理和恢復多個去中心化身份 (AIDs), 實現了符合 W3C DID 規範的身份管理功能。


4. **別名註冊與管理模塊**: 提供了直觀的界面, 用於註冊和管理用戶別名, 增強了系統的可用性和用戶隱私保護。

綜上所述, Wallet 模塊作為 AID 系統的前端範例, 不僅通過與 AID Server 的緊密集成, 為用戶提供了一個安全、便捷的去中心化身份驗證平台, 還成功整合了最新的 AI 技術規範。這一實現不僅證明了 AID 身份驗證系統在實際應用中替代當前中心化第三方登入系統的可行性, 還為未來的去中心化身份管理和 AI 輔助服務的結合提供了寶貴的實踐經驗。

4.1.2 AID Server

本研究中的 AID Server 是一個基於 Golang 開發的高性能 Web API 服務器, 其設計旨在提供自主身份認證 (Autonomous Identity, AID) 系統的核心功能。AID Server 的實現不僅體現了 AID 系統的特殊交互機制, 還嵌入了第三方服務的後端功能, 以驗證 AID 系統替代當前主流 OpenID 驗證系統的可行性。選擇 Golang 作為主要開發語言基於其卓越的並發處理能力、跨平台兼容性以及豐富的標準庫。特別是, Golang 的協程 (Goroutine) 機制能夠有效處理高並發請求, 確保系統在高負載環境下保持穩定性能。

AID Server 實現了一系列核心功能, 涵蓋自主身份 (Autonomous Identity, AID) 管理、數據存儲、人工智能集成以及用戶行為分析。在 AID 管理方面, 系統支持創建、存儲和管理多重身份, 每個身份可關聯多個別名, 從而增強系統靈活性和用戶隱私保護。數據存儲採用了嵌入式數據庫解決方案, 結合 LevelDB 和 SQLite, 既簡化了部署流程, 又為未來向完全去中心化應用擴展奠定了基礎。



LevelDB 主要用於高性能鍵值存儲，而 SQLite 則處理需要複雜查詢的結構化數據。通過集成微軟的 OpenAI API，AID Server 提供了符合微軟 AI 聊天規範的接口，使 Wallet 模塊能夠無縫接入先進的自然語言處理功能。此外，系統還記錄和分析用戶的時空操作數據，為實施風險基礎認證（Risk-Based Authentication，RBA）算法提供了堅實的數據基礎。

在安全機制方面，AID Server 實現了多層次的保護策略。基於收集的用戶行為數據，系統能夠實時評估每次操作的風險程度，實現動態的 RBA。當檢測到高風險操作時，系統會自動觸發額外的身份驗證流程。AID Server 還支持多種多因素認證（Multi-Factor Authentication，MFA）方法，包括但不限於硬體驗證和基於時間的一次性密碼（Time-based One-Time Password，TOTP）等。系統提供了靈活的 API 接口，便於未來集成新的 MFA 方法。此外，所有敏感數據在存儲前均經過加密處理，確保即使在數據泄露情況下，用戶的隱私信息也不會輕易被獲取。

在去中心化架構方面，AID Server 通過與 Consensus Core 模塊的緊密集成，實現了 AID 所有權和管理權的分布式設置與查詢。通過 Consensus Core 提供的 API 接口，AID Server 能夠建立跨節點的 AID 關聯，實現了 AID 的分布式管理。這種設計不僅增強了系統的去中心化特性，還為未來的橫向擴展提供了技術基礎。

總括而言，AID Server 作為 AID 系統的核心組件，不僅提供了高效、安全的身份管理功能，還通過集成先進的安全機制，為去中心化身份認證系統的發展奠定了堅實的理論和技術基礎。

4.1.3 Consensus Core



本研究中的 Consensus Core 模塊基於 Bitcoin 區塊鏈的硬分叉 OurChain 平台開發而成,是一套創新的智能合約系統。其設計旨在為自主身份 (Autonomous Identity, AID) 管理系統提供分布式一致性和安全性保障。OurChain 是由我們開發出來的強大的區塊鏈平台,具有高性能、低功耗和高安全性的特點。Consensus Core 模塊通過與 OurChain 平台的深度集成,為 AID 系統提供了一個安全、可靠的共識機制,確保了整個系統的一致性和可信度。

在 AID 生成過程中,Consensus Core 利用 OurChain 的共識機制確保每個 AID 的唯一性和不可篡改性。這種基於區塊鏈的共識過程具有以下優勢:

- 有效抵禦潛在的網絡攻擊
- 維護整個系統的完整性和可信度
- 為跨節點的 AID 操作提供可靠的信任基礎

Consensus Core 作為 AID 系統的信任錨點,使用戶能夠自主控制其數字身份在多個節點間的遷移和存在。這種設計理念帶來了多重效益:

1. 增強了系統的去中心化特性
2. 賦予用戶對其數字身份的完全自主權
3. 通過區塊鏈技術實現分布式信任
4. 有效解決傳統中心化身份管理系統面臨的單點故障和信任集中化問題

Consensus Core 的設計還考慮到了系統的可擴展性和互操作性。通過標準化的智能合約接口，該模塊為未來整合不同 AID Server 的實作預留了可能性，進一步增強了 AID 系統的適應性和兼容性。



最後，Consensus Core 還透過智能合約實作類似內容分發網路（Content Delivery Network, CDN）的分散式快取伺服器（Distributed Cache Servers）架構。這些快取伺服器儲存了用戶網際協定（Internet Protocol, IP）位址與代管 AID 伺服器地址的映射關係。此設計顯著提升了跨 AID 伺服器時空分析演算法（Cross-server Spatio-temporal Analysis Algorithm）的運行效率。在獲取用戶的時空資訊後，系統無需遍歷所有的 AID 伺服器來確定可能的 AID 代管者，而是通過查詢快取伺服器即可迅速鎖定目標。這種優化措施大幅降低了系統延遲，提高了整體性能，同時保持了 AID 系統的分散式特性和隱私保護能力。

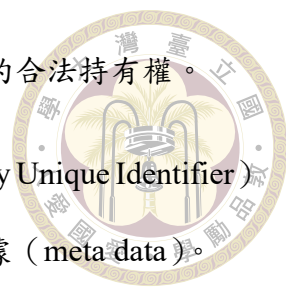
4.2 重要流程分析

4.2.1 AID 生成

在自主身份（Autonomous Identity, AID）系統中，AID 作為普適性的數字身份標識符，具有高度的靈活性和適應性，萬事萬物皆可以用 AID 來識別。根據實際應用需求，AID 的生成方式可以有顯著差異。本實作重點驗證了兩種主要的 AID 使用場景：

4.2.1.1 用戶自主生成 AID

在這種場景下，用戶可以在個人設備上直接生成並管理多個別名關聯的 AID。這種方法的關鍵特徵如下：

- 
- **多因素認證 (MFA)**：用戶需要通過 MFA 來證明對 AID 的合法持有權。
 - **元數據嵌入**: AID 生成過程中，除了基於 UUID (Universally Unique Identifier) 規範產生的識別號外，還需要嵌入輔助 MFA 驗證的元數據 (meta data)。
 - **元數據類型**: 典型的元數據包括但不限於：
 - 手機號碼
 - 電子郵件地址
 - 其他可用於身份驗證的資料...

4.2.1.2 服務提供商生成 AID

在另一種場景中，當用戶在特定服務中註冊別名時，供應商為了方便管理相同別名的用戶，會額外生成 AID 在內部取代別名用來完成用戶識別：

- **服務端管理**: AID 由服務提供商生成並管理。
- **用戶便利性**: 用戶只需記住自己的別名，無需直接管理 AID。
- **責任劃分**: 服務提供商承擔 AID 管理的責任，簡化了用戶的操作流程。

4.2.1.3 AID 生成的通用原則

無論採用哪種生成方式，AID 的創建過程都遵循以下原則：

1. **基於 UUID 規範**: AID 的核心識別號需遵循 UUID 規範，確保全局唯一性。
2. **情境相關元數據**: 根據具體使用情境，在生成過程中同步創建所需的元數據。



3. 區塊鏈記錄: 將 AID 中需要公開的部分信息通過區塊鏈智能合約進行記錄。

這一步驟旨在：

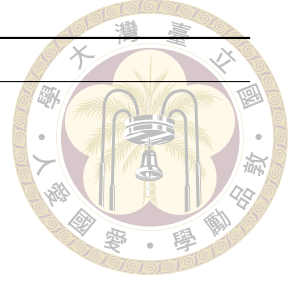
- 確保 AID 的唯一性
- 保證 AID 信息的不可篡改性
- 提供去中心化的公開驗證機制

通過這種設計，AID 系統能夠在保證安全性和可信度的同時，為不同的應用場景提供靈活的身份管理解決方案。這種方法不僅滿足了現代數字身份管理的需求，還為未來更複雜的身份認證場景提供了可擴展的框架。

4.2.2 AID 中基於 RBA 的 MFA 驗證

在本實作中，系統對所有用戶的登入操作以及對應用程式中人工智慧（Artificial Intelligence，AI）功能的訪問進行全面監控。系統記錄包括裝置指紋（Device Fingerprint）與網際協定（IP）地址等元數據（Metadata）。這些數據將被應用於風險基礎認證（Risk-Based Authentication，RBA）演算法，以評估每次操作的風險程度。當計算得出的風險程度超過預設閾值時，系統將自動啟動多因素認證（Multi-Factor Authentication，MFA）流程，要求用戶進行額外的身份驗證。此外，當系統檢測到用戶執行高風險行為時，會通過持續要求 MFA 驗證來提高用戶的可信度。

一個典型的風險評估範例如下：當用戶嘗試登入時，系統讀取當前用戶的裝置指紋和 IP 地址，並檢索該用戶前幾次登入的同類數據。通過比對這些數據，系統可以識別潛在的異常行為，例如 IP 不應該在不同國家或是裝置不該一直切換。若發現顯著差異，系統將要求額外的 MFA 驗證。以下提供了一個簡化的 MFA 觸發演算法示例：



Algorithm 1 MFA 觸發決策演算法範例

```
1: function NEEDMFA(uMeta, opMeta, preRec)
2:   if uMeta  $\equiv$  preRec.uMeta then
3:     if opMeta  $\approx$  preRec.opMeta then
4:       return false
5:     else
6:       return true
7:     end if
8:   end if
9:   return true
10: end function
```

其中， \equiv 表示完全等同， \approx 表示近似相等。此演算法通過比較用戶空間元數據和當前操作元數據與先前記錄的數據來決定是否需要觸發 MFA。此方法能有效地平衡安全性和用戶體驗，為系統提供了靈活且強大的身份驗證機制。

4.2.3 別名在服務器上註冊

本研究提出了一種創新的匿名身份（AID）註冊和關聯機制。這種機制為使用者提供了極為便捷的服務註冊方式。他們只需要定義首選別名和簡單的數位個人識別碼（Personal Identity Number, PIN）即可完成註冊。該系統允許重複別名和 PIN 碼，從而顯著減輕使用者的認知負擔。我們開發的時空分析演算法使 AID 伺服器能夠透過將使用者目前上傳的設備資訊與網際網路協定（IP）位置進行比較來分析和識別潛在的關聯 AID。在此過程中，我們實現了基於風險的身份驗證（RBA）的多重身份驗證（MFA）機制，要求用戶進行額外的驗證以增加其 AID 的可信度來完成 AID 關聯過程。這種方法的獨特之處在於它不需要使用者明確傳遞 AID。此外，由於 AID 可以綁定到另一個 AID 伺服器進行託管，基於 AID 系統的特殊機制甚至無法追蹤使用者的真實 AID，從而在便利性和隱私保護之間取得了平衡。

典型的註冊流程如下：使用者嘗試註冊使用本實現錢包中的 AI 聊天軟體，並

選擇別名和 PIN 碼。AID 伺服器收到請求後，會檢查使用者提交的裝置資訊和 IP 位址，從而通過 Consensus Core 快取服務獲知該 IP 位址最近的操作是由哪些 AID 伺服器管理的，然後發布請求給這些 AID 伺服器。這些伺服器根據使用者提交的資訊執行 RBA 分析。以我們的實作為例，註冊被認為是一項重要的活動，所有請求最初都會被對應的 AID 伺服器拒絕。隨後，使用者直接連接的 AID 伺服器將要求使用者對其想要使用的 AID 進行 MFA 認證。在實作中，以一次性密碼（OTP）為例。使用者接收並輸入 OTP 後，提高了對應 AID 在特定時間和空間內的可信度。前端協助用戶再次發出註冊請求，此時唯一的高可信度 AID 伺服器接受用戶的註冊請求，並將用戶的別名與 AID 關聯起來。值得注意的是，在這個實作中，系統在建立別名時會建立一個 AID，並在系統內部使用 AID 來辨識別名。因此，您實際上不是將別名與 AID 關聯，而是將新的 AID 與 AID 關聯。下面提供一個簡化的處理演算法：

Algorithm 2 AID 註冊與關聯流程

```
1: function REGISTERALIAS(alias, pin, ip, deviceInfo)
2:   aid ← generateAID(alias, pin)
3:   aidServerList ← remoteConsensusCore.cache(ip)
4:   for aidServer in aidServerList do
5:     if aidServer.askForBind(aid, ip, deviceInfo) then
6:       return "Registration successful"
7:     end if
8:   end for
9:   return "Registration Failed please try MFA"
10: end function
11: function MFAinWALLET(aid, ip, deviceInfo)
12:   saveRequest(ip, deviceInfo)
13:   return "MFA OK"
14: end function
15: function MAIN
16:   registerAlias(...)                                ▷ First attempt, should fail
17:   MFAinWallet(...)                                  ▷ User performs MFA
18:   registerAlias(...)                                ▷ Second attempt, should succeed
19: end function
```

此演算法展示了註冊過程中的主要步驟，包括 AID 生成、伺服器查詢、MFA 驗證以及最終的註冊確認。這種設計不僅確保了用戶註冊的便利性，還通過多層

次的驗證機制保障了系統的安全性和用戶的隱私。



4.2.4 別名在服務器上登入

本研究提出了一種創新的別名登入機制，該機制結合了時空分析演算法 (Spatio-temporal Analysis Algorithm)、風險基礎認證 (Risk-Based Authentication, RBA) 以及多因素認證 (Multi-Factor Authentication, MFA)，以解決直接使用別名進行登入這一具有挑戰性的問題。

我們的實作不僅顯著提升了用戶的登入體驗，還確保了系統的安全性和可信度。典型的登入流程如下：

1. 用戶在應用程式介面中輸入別名 (alias) 和個人識別碼 (Personal Identification Number, PIN)，隨後提交登入請求。
2. 匿名身份 (Anonymous Identity, AID) 伺服器接收到請求後，可能會遇到多組相同別名與 PIN 碼的情況。
3. 時空分析演算法根據用戶的裝置資訊和網際協定 (Internet Protocol, IP) 位址，分析並識別最可能對應的幾組別名，進而提取這些別名的 AID。
4. 系統通過共識模組 (Consensus Module) 中的快取查詢功能，定位這些 AID 所關聯的用戶 AID 在相應 AID 伺服器的位址。
5. 主要的 AID 伺服器向這些關聯的 AID 伺服器發送請求，要求進行 RBA 分析。

若 RBA 分析結果顯示風險程度較高，可能出現兩種情況：

- 該別名並非用戶登入的目標別名

- 該別名當前進行別名登入的風險程度過高



基於這兩種條件，系統可能面臨三種情境：

1. 沒有別名通過 RBA 分析，此時系統將要求用戶進行 MFA 驗證。
2. 僅有一個別名通過 RBA 分析，用戶可直接登入。
3. 多個別名通過 RBA 分析，表明當前難以準確識別用戶，系統將要求用戶進行額外的 MFA 驗證以增加可信度，這將有助於提高後續登入時 RBA 分析的通過機率。

本實作的創新之處在於巧妙地結合了時空分析、風險評估和多因素認證，在保障系統安全性的同時，為用戶提供了便捷的登入體驗。這種方法不僅解決了別名重複使用的問題，還通過動態風險評估和額外的身份驗證機制，確保了整個登入過程的安全性和可靠性。

4.2.5 AID 的批次創建與綁定

本實作中，我們成功實現了一個創新的身份管理流程：單一用戶可批次創建多個 AID (Autonomous Identity)，並允許多個用戶將這些 AID 綁定到其個人帳戶下。這種設計不僅展示了 AID 系統作為當前中心化第三方登入系統替代方案的可行性，還證明了其應對複雜身份管理需求的能力。一個典型的應用場景是購票系統，其中一位用戶可以為自己和家人一次性購買多張票，之後每個家庭成員可以使用自己的個人 AID 進行登入和票務管理。

本實作的簡化流程如下：用戶購票成功後，購票網站的 AID Server 創建對應張數的 AID 作為票券，並將這些 AID 關聯到各個家庭成員的 AID。因此，當家庭

成員需要使用票券或登入時，只需使用自己的 AID 即可。登入過程中，系統會根據關聯的用戶 AID 執行相應的風險基礎認證 (RBA) 和多因素認證 (MFA)。這種實現方式不僅簡化了用戶體驗，還保證了每個家庭成員的身份安全和隱私保護。




4.2.6 AID 遷移

為了實現用戶對其身份的完全自主權，本研究提出並實現了一種創新的 AID (Autonomous Identity) 遷移機制。此機制允許用戶自由選擇並更換其信任的 AID 伺服器作為個人 AID 的代管者，從而增強了系統的靈活性和用戶的控制權。

本節詳細闡述了 AID 遷移機制的實現過程及其重要性：

1. **用戶自主權:** 通過允許用戶自由選擇 AID 伺服器，本機制確保了用戶對其數位身份的完全控制權。這不僅提高了系統的透明度，還增強了用戶對系統的信任度。
2. **遷移機制流程:** 本實作的 AID 遷移過程包含以下關鍵步驟：
 - 用戶通過 Wallet 應用程式發起遷移請求，指定目標 AID 伺服器地址。
 - Wallet 應用程式將遷移請求（包含目標 AID 伺服器地址和用戶的 AID）傳送至 Consensus Core 模組。
 - Consensus Core 模組利用智能合約技術，將用戶的 AID 記錄更新至區塊鏈。
 - 系統通知目標 AID 伺服器接收用戶的 AID，同時要求原 AID 伺服器刪除相關用戶資料。
3. **基於共識的操作轉移:** 遷移完成後，所有針對用戶 AID 的操作都會自動轉向新的 AID 伺服器。這是因為系統設計確保所有操作都從 Consensus Core 模

組讀取最新的 AID 伺服器資訊。

- 
4. **安全性與隱私保護:** 本機制在設計時充分考慮了安全性和隱私保護。通過使用區塊鏈技術和智能合約，確保了 AID 遷移過程的透明性和不可篡改性。同時，要求原 AID 伺服器刪除用戶資料，進一步保護了用戶隱私。
 5. **系統彈性:** AID 遷移機制顯著提高了整個身份管理系統的彈性。用戶可以根據個人需求、服務質量或隱私考慮等因素，靈活地選擇或更換 AID 伺服器。

這種 AID 遷移機制不僅體現了去中心化身份管理的核心理念，還為未來更加靈活和用戶友好的身份管理系統奠定了基礎。通過賦予用戶更大的控制權，本研究在提升用戶體驗的同時，也增強了整個系統的安全性和可信度。

4.3 與現有方法之比較

本節詳細闡述了我們的實作如何解決傳統第三方登入系統的固有問題。

4.3.1 AID 系統對 GDPR 合規性的創新解決方案

隨著數據隱私保護日益受到重視，歐盟通用數據保護條例（GDPR）成為了全球範圍內最嚴格的隱私和安全法律之一。本節深入探討了 AID（Autonomous Identity）系統如何創新性地應對 GDPR 帶來的技術挑戰，並提供了超越傳統身份驗證系統的解決方案。

4.3.1.1 數據最小化和存儲限制

GDPR 規定，個人數據的收集和存儲應該在必要的最小範圍內進行。AID 系統在這方面提供了卓越的解決方案：



- **分散式數據存儲:** AID 系統不僅像傳統第三方登入系統那樣允許服務供應商存儲最小必要的數據，還進一步創新，允許用戶將核心個人信息存儲在自己的設備上。
- **按需提交機制:** 每次登入或註冊時，用戶只需提交服務所需的最小數據集給服務供應商。這種機制顯著降低了不必要的數據暴露風險。
- **動態數據管理:** AID 系統能夠根據不同服務的具體需求，動態調整所提供的個人數據範圍，確保始終符合 GDPR 的數據最小化原則。

這種創新的數據處理方式不僅符合 GDPR 的要求，還大大增強了用戶對個人數據的控制力，為數據隱私保護設立了新的標準。

4.3.1.2 被遺忘權的實現


GDPR 賦予了用戶要求刪除個人數據的權利，即所謂的”被遺忘權”。AID 系統在實現這一權利方面提出了突破性的解決方案：

1. 傳統系統的局限性:

- 主流身份驗證系統通常依賴用戶主動要求服務供應商刪除數據。
- 即使服務供應商誠實地執行刪除操作，也無法保證數據不被其他相關方保留。
- 用戶的活動日誌和互動記錄等間接數據通常難以徹底刪除。

2. AID 系統的創新方案:

- **別名機制:** 用戶註冊新服務時使用別名，服務供應商只能獲知關聯的 AID 由哪個 AID Server 代管，而無法直接識別用戶的真實身份。

- 
- **身份解耦**: 這種設計使得即使發生數據洩露，也難以將數據準確關聯到特定真實用戶。
 - **”否認權”**: AID 系統為用戶提供了一種實質上的”否認權”，即使在極端情況下，用戶也可以合理地否認某些數據與其真實身份的關聯。

AID 系統的這種創新設計不僅技術上實現了 GDPR 的被遺忘權要求，還在概念上重新定義了數字身份的隱私保護模式。它提供了一種前所未有的隱私保護層級，即使在數據洩露的極端情況下，也能有效保護用戶的真實身份。

4.3.2 AID 系統的身份冒用防護與救援機制

在數字身份管理領域，身份冒用和身份恢復是兩個關鍵挑戰。本節詳細探討了 AID（Autonomous Identity）系統如何通過創新的”極致多因素認證”（Extreme Multi-Factor Authentication, EMFA）機制來有效應對這些挑戰，提供了超越傳統身份驗證系統的安全保障和用戶體驗。

4.3.2.1 身份冒用防護機制

傳統身份驗證系統中，用戶身份通常依賴單一或有限的識別因素（如用戶名、密碼或電子郵件地址），這種設計存在固有的安全風險：

- **單點失效風險**: 單一識別因素的洩露可能導致整個身份被盜用。
- **社會工程攻擊脆弱性**: 攻擊者可能通過欺騙或操縱獲取關鍵身份信息。
- **身份恢復機制的安全隱患**: 傳統的身份恢復過程（如密碼重置）可能被攻擊者利用。

AID 系統通過實施 EMFA 機制，有效解決了上述問題：



1. 動態多因素認證:

- 系統動態評估每次認證請求的風險級別。
- 根據風險評估結果，要求用戶提供不同數量和類型的身份因素。
- 這種方法顯著提高了身份冒用的難度，因為攻擊者需要同時掌握多種身份因素。

2. 身份因素多樣性:

- AID 系統支持廣泛的身份驗證因素，包括但不限於：
 - 知識因素（如密碼、安全問題）
 - 所有因素（如硬件令牌、智能手機）
 - 固有因素（如生物特徵）
 - 行為因素（如使用模式、地理位置）
- 這種多樣性大大增加了攻擊者成功冒用身份的難度。

3. 持續性身份驗證:

- AID 系統在整個會話過程中持續評估用戶身份的可信度。
- 如果檢測到異常行為，系統可以即時要求額外的身份驗證。

4. 去中心化身份存儲:

- 用戶的身份信息分散存儲，減少了大規模數據洩露的風險。
- 即使某些身份因素被盜，攻擊者也難以獲得足夠信息來完全冒用身份。

4.3.2.2 創新的身份救援機制



AID 系統的 EMFA 機制不僅提高了安全性，還為身份救援提供了創新解決方案：

1. 漸進式身份恢復：

- 用戶可以通過逐步提供更多身份因素來提高自己的可信度。
- 系統根據提供的因素數量和質量，逐步授予用戶更多權限。

2. 多維度身份驗證：

- 在恢復過程中，系統考慮多種身份因素的組合，而不是依賴單一“主要”因素。
- 這種方法增加了身份恢復的靈活性，同時維持了高安全標準。

3. 風險自適應恢復流程：

- 系統根據恢復請求的風險級別動態調整所需的身份因素。
- 低風險操作可能只需要少量因素，而高風險操作則要求更嚴格的驗證。

4. 社交網絡輔助恢復：

- AID 系統引入了創新的社交恢復機制，允許用戶預先指定信任的聯繫人。
- 在緊急情況下，這些信任聯繫人可以協助驗證用戶身份，增加了恢復過程的安全性和可靠性。

4.3.2.3 安全性與可用性的平衡



AID 系統的 EMFA 機制在提供強大安全保障的同時，也注重保持良好的用戶體驗：

- **上下文感知認證:** 系統根據用戶的使用環境和行為模式動態調整認證要求，減少不必要的認證步驟。
- **漸進式安全提升:** 用戶可以逐步增加其帳戶的安全級別，而不是一次性要求所有可能的身份因素。
- **用戶友好的界面設計:** 儘管背後的機制複雜，但系統為用戶提供了直觀、易用的界面，簡化了身份驗證和恢復過程。

總結而言，AID 系統通過其創新的 EMFA 機制，不僅有效解決了身份冒用問題，還提供了靈活、安全的身份救援方案。



第五章 結果與未來展望

5.1 結論

5.2 未來展望





參考文獻





附錄 A — 模型參數表

A.1 模型一

A.2 模型二





附錄 B — BraTS 2021 分割結果圖

B.1 編號 0001 0050

B.2 編號 0051 0100