

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 8 月

August, 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：_____





致謝文打在這裡。致謝文打在這裡。致謝文打在這裡。致謝文打在這裡。

致謝文打在這裡。致謝文打在這裡。致謝文打在這裡。致謝文打在這裡。致謝文打

在這裡。致謝文打在這裡。致謝文打在這裡。致謝文打在這裡。致謝文打在這

裡。致謝文打在這裡。致謝文打在這裡。致謝文打在這裡。致謝文打在這裡。致

謝文打在這裡。致謝文打在這裡。致謝文打在這裡。致謝文打在這裡。致謝文打

在這裡。致謝文打在這裡。致謝文打在這裡。致謝文打在這裡。致謝文打在這

裡。致謝文打在這裡。致謝文打在這裡。致謝文打在這裡。致謝文打在這裡。致

謝文打在這裡。致謝文打在這裡。致謝文打在這裡。致謝文打在這裡。致謝文打

在這裡。致謝文打在這裡。致謝文打在這裡。致謝文打在這裡。致謝文打在這

裡。致謝文打在這裡。致謝文打在這裡。致謝文打在這裡。致謝文打在這裡。致

謝文打在這裡。致謝文打在這裡。致謝文打在這裡。致謝文打在這裡。





摘要

現代數位身分系統面臨嚴峻挑戰：身分驗證漏洞威脅用戶安全，中央化數據儲存易遭攻擊導致大規模個資外洩，大型組織壟斷關鍵服務造成權力失衡。這些問題不僅危及個人權益，更阻礙了數位社會的發展。本研究將完善「自主身分」系統，旨在徹底重塑數位身分管理。我們從身分認證、資料管理和信用評分三個關鍵領域著手，設計了一套去中心化解決方案。這一系統將數位身分的控制權從大型機構手中歸還給個人用戶，顯著提升了用戶自主權。我們還基於區塊鏈 OurChain 進行了概念驗證，成功證實了 AID 系統的可行性。我們認為「自主身分」系統有潛力徹底改變人們與數位世界的互動方式，為建立一個更安全、公平和自由的數位社會鋪平道路。

關鍵字：自主身分、驗證、憑證、隱私、區塊鏈





Abstract

Today's digital identity systems have big problems: weak security puts users at risk, central data storage can be easily hacked, and big companies have too much control. These issues hurt people and slow down digital progress. Our research improves the 'Autonomous Identity' (AID) system to fix these problems. We created a new solution that focuses on three main areas: checking who you are, managing your data, and scoring your reputation. This system gives control back to users, not big companies. We tested AID using OurChain blockchain and showed it works. We consider AID can change how people use the digital world, helping create a safer, fairer, and freer online society.

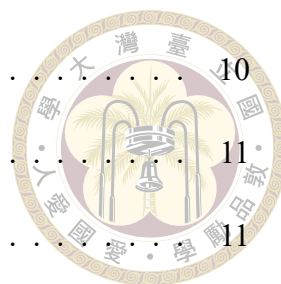
Keywords: Autonomous Identity, Authentication, Certification, Privacy, Blockchain





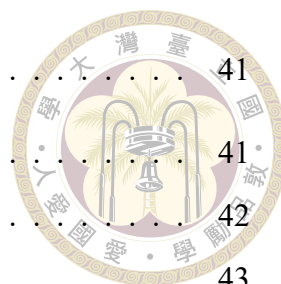
目錄

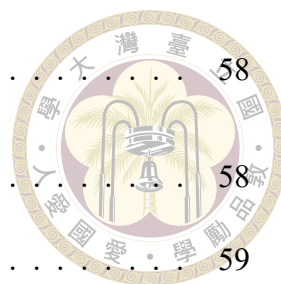
	Page
口試委員審定書	i
致謝	iii
摘要	v
Abstract	vii
目錄	ix
圖目錄	xv
表目錄	xvii
第一章 緒論	1
1.1 研究動機	2
1.2 主要貢獻	3
1.3 論文架構	3
第二章 文獻探討	5
2.1 身分系統的起源	5
2.2 身分系統的迭代	7
2.2.1 中心化身分	7
2.2.2 聯合身分	8
2.2.3 使用者中心的身分	9



2.2.4	自治身分	10
2.2.5	未來展望	11
2.3	AID 系統的發展	11
2.3.1	最初的自主身分	11
2.3.2	自主憑證機制	13
2.4	身分系統的挑戰	14
2.4.1	使用者體驗	14
2.4.2	使用者認知	16
2.4.3	隱私保護	17
2.4.4	平等信任	18
2.4.5	法律合規性	19
2.4.6	公認原則	20
2.5	本章總結	20
第三章	系統設計	23
3.1	系統的新設計	23
3.1.1	自主認證	24
3.1.1.1	最簡自主認證	24
3.1.1.2	MFA 的加入	26
3.1.1.3	AID Server 的加入	27
3.1.1.4	自主憑證的使用	29
3.1.1.5	自主認證流程	30
3.1.2	數據自主	31
3.1.2.1	數據被遺忘權	33
3.1.2.2	數據明確授權	35
3.1.2.3	數據可驗證性	38

3.1.2.4	無特權的執行	41
3.1.3	信用評分	41
3.1.3.1	信用評分機制	42
3.1.3.2	生態系的營運	43
3.2	系統架構設計	43
3.2.1	系統結構概覽	44
3.2.2	層次與角色對應	45
3.2.2.1	共識層與共識核心	45
3.2.2.2	服務層與服務提供者	45
3.2.2.3	數據層與終端使用者	45
3.2.3	共識層	46
3.2.4	服務層	46
3.2.4.1	身分管理	47
3.2.4.2	憑證管理	48
3.2.4.3	數據管理	48
3.2.5	數據層	49
3.2.5.1	身分管理	50
3.2.5.2	數據管理	50
3.2.5.3	憑證管理	50
3.2.5.4	數據存儲	51
3.3	系統設計細節	52
3.3.1	區塊鏈憑證機制	52
3.3.1.1	自主憑證	52
3.3.1.2	數據憑證	54
3.3.2	身分識別問題	55
3.3.2.1	基於使用者時空的分析方法	56
3.3.2.2	基於危險程度的驗證機制	57

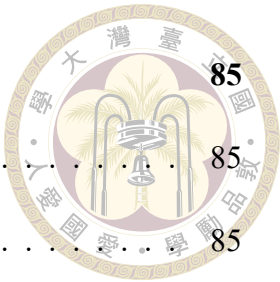




3.3.3	密碼救援問題	58
3.3.3.1	極致多因素驗證	58
3.3.4	混合數據管理	59
3.3.5	組織使用者控管	60
3.4	資料結構	61
3.4.1	共識核心	61
3.4.2	AID Server	61
3.4.3	Wallet	62
3.5	本章總結	63
第四章	系統實作	65
4.1	系統架構	65
4.2	實現細節	66
4.3	流程分析	68
4.3.1	產生新的 AID 與自主憑證	68
4.3.2	進入支付服務獲取收據	69
4.3.3	使用 AI 服務對話	70
4.4	本章總結	71
第五章	結論與未來展望	73
參考文獻		75
附錄 A — 實際操作介面		83
A.1	AID 錢包	83
A.2	AI 聊天軟體	83

附錄 B — 系統 UML 圖

B.1	自主憑證流程	85
B.2	數據憑證流程	85







圖目錄

2.1	中心化身分	8
2.2	聯合身分	8
2.3	使用者中心身分	9
2.4	自治身分	10
2.5	自主身分	12
2.6	AID 系統的自主憑證機制 (AC)	13
3.1	最簡自主認證	26
3.2	過去的自主認證	28
3.3	自主認證 (加入憑證機制)	29
3.4	自主身分	31
3.5	過去的 AID 資料結構	32
3.6	數據憑證 (資料傳出)	37
3.7	當前網路後端設計	39
3.8	跨服務共識流程	40
3.9	自主身分系統結構圖	44
3.10	自主憑證流程	53
3.11	數據憑證流程	55
4.1	AID 概念驗證架構簡圖	65
4.2	產生新的 AID 與自主憑證	68
4.3	進入支付服務獲取收據	69
4.4	產生新的 AID 與自主憑證	70





表目錄

2.1 身分系統需求比較表	7
3.1 數據自主的四個難題	33
3.2 自主身分系統層次與角色對應	44
3.3 各代身分管理系統比較	63
4.1 系統分層結構及介面	66





第一章 緒論

在當今數位世界中，身分系統面臨著諸多挑戰，如身分盜竊、數據壟斷和隱私侵犯等問題。為了應對這些挑戰，本研究完善了自主身分（Autonomous Identity, AID）系統。這個系統不僅在理論上具有深遠意義，更有望在實踐中徹底改變人們與數位世界的互動方式。

AID 系統的核心理念是將「自主」的哲學概念融入數位身分管理中。在這個創新的框架下，使用者能夠完全掌控自己的數據，自主發起每個數位行為。想像一個世界，人們可以輕鬆地在不同服務間切換身分，自由決定數據的流動，這種程度的控制在當前的數位環境中幾乎難以想象。

更令人興奮的是，AID 系統不僅僅是一個技術創新，它還通過營造具有道德意識的環境，為構建一個更加公平、透明的數位社會鋪平了道路。這意味著可能即將進入一個新的數位時代，在這個時代中，使用者的權益將得到充分尊重，數據隱私和安全性將得到顯著提升。

本研究的目標是完善 AID 系統的理論基礎，設計出一個更具體的系統架構，並通過實作驗證其可行性。我們相信，AID 系統將為數位身分管理帶來一場變革，為建立一個更加安全、公平和自由的數位社會做出貢獻。



1.1 研究動機

自主 (Autonomy) 概念源於 18 世紀啟蒙運動，標誌著人類開始質疑對王權與神權的依賴，轉而追求通過科學與理性實現思想獨立與自由。這一概念的演變不僅塑造了現代社會對自主的理解，也為重新思考數位時代的身分管理提供了重要的理論基礎。

Campbell[11] 剖析了啟蒙思想家康德 (Kant) 對自主的理解：「意志藉由自身成為自己的法則的那種特性」(“that property of the will by which it [the will] is a law to itself”)。換言之，自主可以被理解為「個體自由地遵循符合自己道德標準的法則」。這一解釋揭示了自主不僅涉及個體的自由意志，還關乎整個社會的道德基礎。

基於康德的觀點，可以衍伸出一個關鍵問題：如何在數位世界中形塑道德標準？這並非一個簡單的問題。本研究設想的做法是在身分系統中營造出一個孕育道德觀的空間，通過使用者之間的真實互動，逐漸形成共同認知的道德標準。這樣的系統將成為一個真正自主的身分系統，而非僅僅是當前普遍存在的資料庫式身分系統。

基於這些洞見，本研究提出的「自主身分」(Autonomous Identity, AID) 概念與目前主流探討的「自治身分」(Self-sovereign identity, SSI) 系統有著本質的區別。SSI 允許使用者參與身分管理系統的經營，賦予使用者對其數位身分的一定控制權。然而，AID 則更進一步，將「自主」的哲學思想融入數位身分管理的實踐中，使每位使用者在具備動態道德標準的身分系統中自由的管理自己。這種模式不僅賦予使用者更大的自主權，還在系統設計中納入了道德考量，以確保個人自主不會損害社會整體利益。



1.2 主要貢獻

本研究透過完善實驗室學長 Yuxuan[30] 提出的「自主身分」(Autonomous Identity, AID) 系統。從自主認證、數據自主與信用評分三個方面出發，不僅建立更完整的理論基礎，更解決了過去並未關注到的重要議題。此外，通過在區塊鏈 OurChain[37] 上實作 AID 系統，驗證了其在實際應用中的可行性和效果。本研究的主要貢獻包括：

- **構建更完整的 AID 理論基礎：**在 Yuxuan 的研究基礎上，進一步完善了 AID 系統的理論框架，涵蓋自主認證、數據自主和信用評分等多個方面。
- **設計更具體的 AID 系統架構：**提出了一種符合最新發展趨勢的 AID 系統架構，成功解決了資訊安全、隱私保護、使用者體驗等多方面的技術難題。
- **實作基於區塊鏈的 AID 系統原型：**在區塊鏈上實作了 AID 系統的原型，並通過實際應用驗證了其可行性和效果。

1.3 論文架構

為了基於自主的理念設計出一個完整的身分驗證系統，本研究首先探討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（第二章）。接著提出 AID 的系統設計，並且說明其架構細節與資料結構（第三章）。然後，透過實作來驗證本研究的系統設計（第四章），最終提出結論與對 AID 系統的未來展望（第五章）。





第二章 文獻探討

「數位身分」可以被用於在網路上標識一個人，一種物體，甚至一個機構。更準確的說，數位身分是一組屬性，包含了對應實體的資訊與唯一識別號 (identity, ID)。而「數位身分系統」是一套用於創建、管理和驗證「數位身分」的技術和流程集合。本研究探討的 AID 系統即是一種數位身分系統，旨在解決當前身分系統中存在的問題，並提出一套完整的解決方案。

本章節闡述了理解自主身分 (Autonomous Identity, AID) 系統所需的關鍵背景知識。首先，本章將探討身分系統的起源，以此更進一步說明「自主」的設計理念。其次，本章將回顧身分系統的迭代過程，從中心化模式到自治身分，分析各代系統的特徵及其優劣。繼而，本章將聚焦於 AID 系統的發展軌跡，探討其起源與演變。最後，本章將詳細闡述身分系統設計面臨的多維度挑戰，包含使用者體驗、使用者認知、隱私保護、平等信任、法律合規性和公認原則等方面。通過這些多角度的探討，本章為讀者構建一個全面的理論框架，為後續對 AID 系統的深入分析和評估奠定基礎。

2.1 身分系統的起源

網際網路 (Internet) 起源於 20 世紀 70 年代，逐漸發展成為現代網際網路的基礎架構。最初，Internet 的設計源自美國國防部的 ARPANET 計畫，旨在滿足軍



事需求下的網路可用性和穩定性。因此，其初期設計基於以下假設[36]：

- 最終使用者至少在最低程度上相互信任。
- 網路由於潛在的物理攻擊而本質上不可靠。

這些假設在當時的網路環境中是合理的。然而，隨著網路的普及和應用範圍的擴大，這些假設已不再適用[13]。Internet 從一個研究驅動的项目演變為社會的重要組成，新的需求不斷湧現，不僅挑戰了原有的設計原則，還促使人們重新審視既有的假設。

在現代網路環境中，使用者之間的信任關係變得日益複雜。人們需要可靠的方式來識別自己和他人，以便在網路上進行交流和交易。因此，各種身分系統應運而生，以滿足網路環境中的身分識別需求。但正如 Cameron[10]所指出的，統一而理想的身分管理系統實際上並不存在。身分系統涉及的範疇廣泛而複雜，個人與組織之間存在多樣化且往往相互衝突的需求，試圖通過單一標準來限制或規範這些需求是不切實際的。

例如，終端使用者可能希望自由地訪問和分享信息，而內容提供商和知識產權持有者則希望保護其知識產權。政府可能希望監管某些網絡活動以維護社會秩序，而使用者和隱私倡導者則強調個人隱私的重要性。這種複雜的利益衝突導致了諸如網絡中立性、數據隱私、內容審查等一系列熱點問題的出現[50]。

面對這種複雜的需求，Blumenthal[8]認為確保設計上的一般性、彈性與開放性至關重要。他設想的未來系統應能容納衝突並逐漸改善：企業的管理者和被管理者爭論分紅，垃圾郵件的發送者和接收者爭論各自的困難。在這個網路世界中，不同身分的參與者沒有絕對的贏家，也沒有天生的失敗者。理想的系統應該通過所有使用者不斷的爭論和互動逐漸形成。



總而言之，現代網路環境的變遷對身分系統提出了新的挑戰。為了應對這些挑戰，本研究期望透過自主的設計理念，設計出一套能讓每個系統參與者自由互動且在彼此的影響下逐漸改進的身分系統。

2.2 身分系統的迭代

Table 2.1: 身分系統需求比較表

需求	中心化身分	聯合身分	使用者中心的身分	自治身分	自主身分
數據集中問題	✓	×	✓	✓	✓
資料孤島化	×	△	△	✓	✓
組織身分控管	✓	✓	△	×	✓
便於系統管理	✓	✓	✓	×	△
系統管理一致性	✓	✓	✓	×	△
減少多重身分認知負擔	×	△	△	✓	✓
實現單點登錄 (SSO)	×	✓	✓	✓	✓
促進組織間協作	×	△	△	×	✓
降低運營成本	✓	×	✓	△	✓
增強隱私保護	×	×	△	✓	✓
使用者身分資料控制權	×	×	△	✓	✓
身分提供者選擇靈活性	×	×	✓	✓	✓
服務數據控制權	×	×	×	×	✓
身分可攜性	×	×	△	✓	✓
使用者與服務商平等性	×	×	×	✓	✓
抗審查特性	×	×	×	✓	✓
符合現有法律框架	△	△	△	×	✓
高易用性	✓	✓	✓	×	✓
系統互操作性	×	×	✓	✓	✓

身分系統的設計經歷了多個階段的演變，每個階段都試圖解決特定的問題，同時也帶來了新的挑戰如表2.1。本節將介紹不同世代的身分系統設計，以說明彼此衝突的需求和技術限制，並為後續討論提供背景。

2.2.1 中心化身分

中心化身分系統如圖2.1是最初期的身分管理方案，由同一位管理者操作和存儲所有使用者資訊，在企業和政府機構中廣泛應用。典型例子包括活動目錄 (Windows Active Directory, AD) 和輕量級目錄訪問協議 (Lightweight Directory

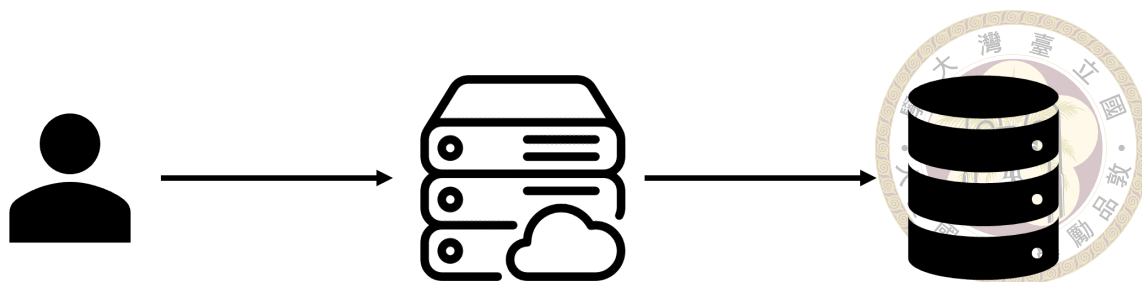


Figure 2.1: 中心化身分

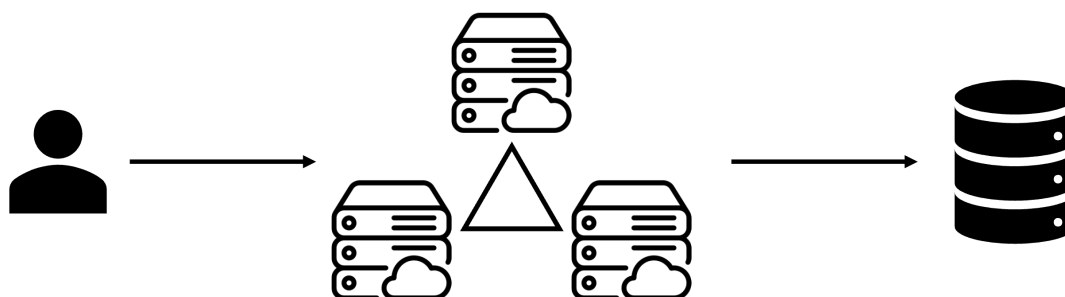


Figure 2.2: 聯合身分

Access Protocol, LDAP) [33, 43]。這類系統的主要優勢在於其集中管理的特性，便於系統管理員進行使用者管理和權限控制，同時確保組織內部身分信息的一致性和及時更新。然而，中心化身分系統也面臨著諸多挑戰，如單點故障風險、隱私保護問題，以及跨組織可遷移性差等。使用者通常需要為每個服務創建單獨帳戶，這不僅增加了認知負擔 [25]，還導致使用者身分被服務提供商完全控制，缺乏自主權。

2.2.2 聯合身分

主要為了解決同一個使用者擁有太多身分的認知負擔，聯合身分系統如圖2.2允許不同組織間共享身分信息，代表性技術包括安全斷言標記語言（Security Assertion Markup Language, SAML）和 WS-Federation [18, 38]。這種模式的出現大大改善了使用者體驗，實現了單點登錄（Single sign-on, SSO），減少了密碼疲勞問題。聯合身分系統促進了組織間的協作和資源共享，同時也降低了重複身分管理的運營成本。但是，這種模式也帶來了隱私方面的挑戰 [2]，如使用者信息



Figure 2.3: 使用者中心身分

在多個服務提供商間共享可能違反《通用資料保護規則》（General Data Protection Regulation, GDPR）[15] 等隱私法規。此外，實施和維護聯合身分系統的技術複雜度較高，參與組織之間需要建立並維護信任關係。

2.2.3 使用者中心的身分

為了解決隱私方面的問題，使用者中心的身分系統如圖2.3所示逐漸興起。這種系統允許使用者透過單一的身分供應者登入多個獨立的服務，同時每個服務各自掌握使用者在其內部的資料。隨著這種需求的增長，新的技術標準應運而生。OpenID 和 OAuth 等協議的出現 [21, 41]，標誌著身分管理向使用者賦權的重要轉變。這種模式增強了使用者對個人身分信息的控制權，提供了更大的靈活性，允許使用者選擇不同的身分提供者。

使用者中心的身分系統實現了以服務供應商為單位的資訊範圍控制，在一定程度上改善了隱私保護。然而，這種方式也面臨著一些挑戰。首先是身分碎片化的問題，多個身分提供者的存在可能導致使用者體驗的不一致。其次，安全風險



Figure 2.4: 自治身分

如釣魚攻擊和身分提供者數據洩露等仍然存在 [47]。此外，儘管使用者獲得了更多控制權，但他們仍然在某種程度上依賴中心化的身分提供者，因此使用者的自主性還擁有很大的提升空間 [4]。

2.2.4 自治身分

自治身分如圖2.4所示，是身分管理系統的最新發展 [39]。這種方式透過區塊鏈技術取代中心化的身分供應商，其代表性例子包括基於以太坊的 uPort 和微軟的自治身分覆蓋網絡 [31, 32]。自治身分系統賦予使用者對自身身分的控制權與對身分管理的治理權，同時提高了身分的可攜性和一致性。此外，它還增強了使用者與服務提供商之間的平等性，並提供了抗審查的特性。

然而，作為一種新興技術，自治身分系統也面臨著諸多挑戰 [42]：

- **法律框架衝突**：它可能與現有法律框架存在潛在衝突，例如與 GDPR 中的被遺忘權不相容 [16]。
- **技術複雜性**：自治身分系統的技術複雜性可能影響普通使用者的使用體驗，

降低其易用性 [27]。

- 其他問題：還有諸如隱私保護、系統互操作性等問題需要解決。



2.2.5 未來展望

綜上所述，身分系統的發展經歷了多個階段，每個階段都試圖解決特定的問題。從中心化到使用者自治，身分系統的設計逐漸向使用者賦權，提高了使用者對自身身分的控制權。但是，即使到了使用者自治階段，人們依舊不認為找到了理想的解決方案。如同 Schardong 等人 [42, 46] 所說，當今的自治身分系統仍面臨著許多挑戰，包括安全性、隱私保護、易用性、信任建立等問題。因此，本研究認為身分系統的設計仍有很大的改進空間，需要更多的研究和實踐來不斷優化。

2.3 AID 系統的發展

自主身分系統的發展源於對傳統身分系統的挑戰，本節將介紹 AID 系統的發展軌跡，從最初的自主身分到後續的自主憑證機制，探討其設計理念和技術特點。

2.3.1 最初的自主身分

最初的自主身分 (Autonomous Identity, AID) 系統由學長 Yuxuan[30] 設計，其主要目的是在保障使用者自主權的前提下，建立一個跨網路服務的統一身分框架。這裡的「自主」概念可以被理解為使用者能夠完全掌控自己的數據和隱私。為實現這一目標，AID 系統加入以下幾個核心要素：

- 數據自主：將大部分使用者數據存儲在本地設備上，以確保數據隱私和控制

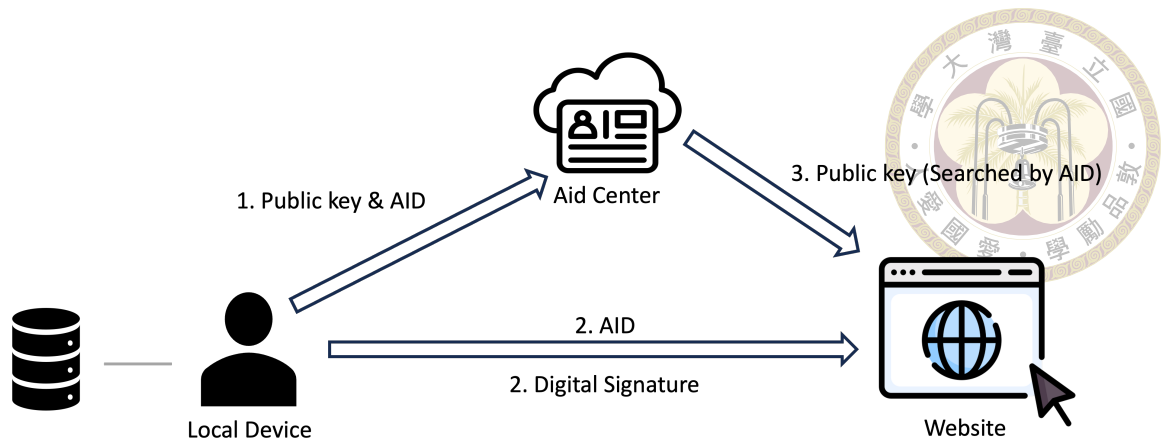


Figure 2.5: 自主身分

權。

- **自主認證：**採用數位簽章方案來實現身分驗證，確保身分認證不依賴外部系統。
- **信任評估：**引入基於評論的評估機制，以建立信任體系，幫助服務提供商評估使用者的可信度。

明確的描述 AID 系統的自主認證機制如圖2.5所示。使用者在個人設備產生隨機 UUID 與公私鑰對，並將公鑰與 UUID 上傳至中心化的註冊機構（AID Server），以註冊 AID。當使用者需要進行身分認證時，他們可以通過私鑰對數據進行簽名，並將簽名與數據一起發送給服務提供商，服務提供商可以通過向 AID Server 下載公鑰來驗證簽名的有效性，從而確認使用者的身分。此外，學長還提出可以讓使用者輸入帳號與密碼作為配合 UUID 成為生成公私鑰對的隨機種子，以簡化使用者需要記憶的資訊。

關於使用者數據自主的機制，學長提出角色資料（actor profile）的概念，每個 AID 都會在個人設備上存儲多個角色資料，內部包含了該角色的基本資訊與在使用服務的過程中產生的數據。當使用者需要使用服務時，他們可以選擇將特定角色資料上傳至服務提供商，從而實現數據的自主。

最後，基於 AID 的評分機制，透過讓服務提供商對使用者在服務中的行為在

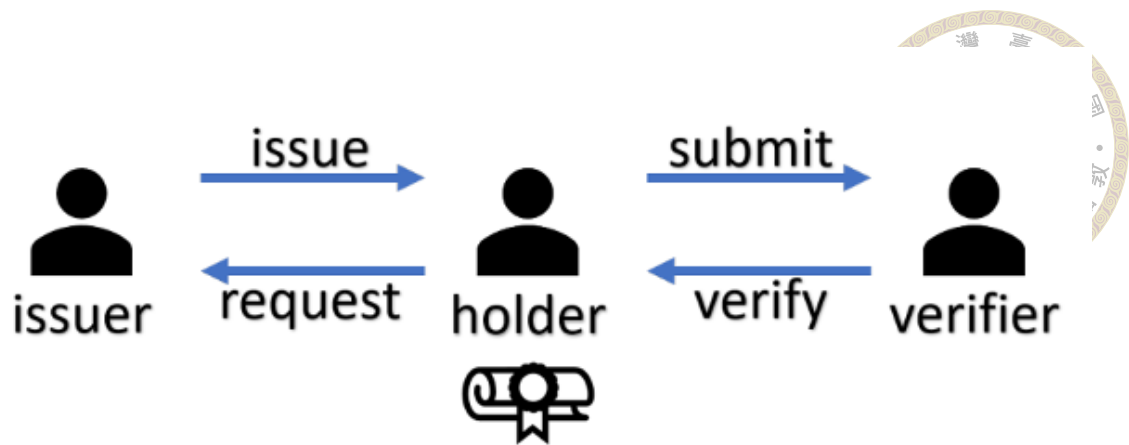


Figure 2.6: AID 系統的自主憑證機制 (AC)

AID Server 中評分，並將評分結果與 AID 綁定，從而實現對 AID 的可信度與真實性的量化。這種機制可以幫助服務提供商更好地評估使用者的可信度與真實性。

2.3.2 自主憑證機制

本研究採用的 AID 系統自主憑證機制 (Autonomous Certificate, AC) 源自學長 Tze-Nan[51] 的研究成果。該機制旨在增強 AID 系統的數據自主性，並提升用戶隱私保護水平。Tze-Nan 不僅闡述了 AC 機制的應用場景，還詳細闡釋了其設計原則與實現方法。

AC 機制如圖2.6所示可以看作是建構於 AID 系統之上的公鑰基礎設施 (Public Key Infrastructure, PKI) 擴展方案。使用者可以尋求 AC 機構簽發包含使用者特定資訊的憑證，其格式應類似傳統的 X.509 憑證 [23]。這些憑證包含了簽發 AC 機構的數位簽名、AC 有效期限與對應 AID 身分資訊等等。

這種自主憑證機制有效替代 AID 系統原設計中的角色資料 (actor profile)。它使得使用者能夠根據具體需求，自行定義在特定服務中欲揭露的個人資料範疇，並尋求特定 AC 機構的背書。這一機制不僅提升了用戶提供之角色資訊的可信度，還增強了整體系統的靈活性。此外，AC 機制亦提供自行簽署的功能，允許使用

者在無需 AC 機構背書的情況下自主簽署憑證。這一設計確保了 AID 系統不會因引入 AC 機制而損及其原有的自主性原則。



為了闡明 AC 機制的實際應用，我們可以考慮以下情境：某學生用戶在線購買教科書時，希望利用學生身分享受優惠，但同時也想最小化個人資訊的披露。在 AC 機制下，該用戶可向指定的 AC 機構申請學生身分憑證，並將其提供給電商平台。平台通過驗證 AC 的有效性即可確認用戶的學生身分，而無需獲取其他個人資訊。此過程體現了 AC 機制支持用戶根據具體需求選擇性地提供身分資訊，從而實現最小數據揭露原則，有效保護用戶隱私。

2.4 身分系統的挑戰

為了設計出一個理想的身分系統，在本節中，會從不同維度的多個方向來探討身分系統的困境。這些討論能釐清身分系統的核心特徵，以此為未來的設計提供參考。

2.4.1 使用者體驗

使用者體驗在身分系統設計中扮演著關鍵角色，直接影響系統的可用性和採納率。然而，Hamme 等人 [20] 的研究闡明了使用者體驗、安全性和隱私保護之間的複雜關係。該研究指出了一個普遍存在的現象：使用者傾向於選擇最簡單的方式來設置和使用身分系統，這種傾向可能導致系統安全性和隱私保護程度的降低。

這種情況產生了一個兩難困境，為了提高安全性而強制使用者採用複雜的身分驗證方式可能會適得其反。例如 Zhang 等人 [52] 的研究表明，要求使用者定期

更改密碼往往導致使用者僅修改特定字元，反而造成更大的安全隱患。同樣地，為了增強隱私保護而要求使用者完成詳細的隱私設置也可能降低使用者體驗。Acquisti 等人 [1] 的研究發現，複雜的隱私設置過程往往讓使用者感到困惑和沮喪，甚至導致他們放棄設置而選擇默認選項，從而降低了隱私保護水平。

為解決這一困境，研究者提出了「無摩擦驗證」(Zero Friction Authentication) 的概念，旨在最小化使用者在設置和使用過程中遇到的困難，同時維持適當的安全性和隱私保護水平。Hamme 等人 [20] 強調，無摩擦驗證的核心目標是在保護使用者安全和隱私的同時，顯著降低使用者的操作負擔。這種平衡對於現代身分系統的設計至關重要，因為它直接影響系統的使用率和效能。

為實現無摩擦驗證，近年來安全領域出現了多種新技術。如 Ghorbani 等人 [17] 研究了無密碼登入（如 FIDO2）的可用性，發現這種方法能透過硬體密鑰在手機上跨裝置完成高安全性的驗證，且使用者普遍認為方便並願意持續使用。Wiefling 等人 [49] 探討了基於風險的驗證（Risk-Based Authentication, RBA），該方法透過追蹤身分與系統互動的歷史數據，在每次服務請求時動態判斷危險性，並在危險時採用更安全的多因素驗證（Multi-Factor Authentication, MFA） [9]。Alaca 等人 [3] 關於裝置指紋的研究顯示，透過在每次使用者對服務發出請求時記錄並比對裝置指紋，可以有效辨識部分惡意行為，且不會增加使用者的操作負擔。

另外，為解決複雜隱私設置帶來的問題，Acquisti 等人 [1] 提出了「隱私設計」(Privacy by Design) 的概念。這種方法將複雜的設置過程分解為多個簡單步驟，並在使用者使用系統的不同階段逐步引導使用者完成設置。研究表明，這種方法不僅能提高使用者的隱私保護水平，還能顯著改善使用者體驗。

這些新技術的應用表明，在不影響使用者體驗的前提下提供更高的安全性和更好的隱私保護是可能的。然而，如何在自主身分系統中實現真正的無摩擦驗證，

以及如何有效地平衡使用者體驗、安全性和隱私保護的需求，仍然是一個值得深入研究的課題。



2.4.2 使用者認知

使用者認知在數位身分管理中扮演著至關重要的角色，直接影響到系統的安全性和有效性。LastPass[28] 的研究揭示了使用者認知與實際情況之間存在顯著差距：使用者平均估計自己擁有 20 個線上帳號，而實際上平均擁有 37 個以上的帳號。這種認知偏差背後反映了使用者使用上的不便，例如經常因為忘記密碼而無法登入帳號，或者因為各個帳號資料不互通而需要耗費大量時間來管理。

Dhamija 等人 [14] 的研究進一步指出，使用者對身分管理系統的認知和理解程度直接影響其安全行為。隨著需要管理的使用者名和密碼數量增加，使用者往往感到困惑，進而採取不安全的行為，如使用弱密碼或在多個平台使用相同密碼等。此外，使用者對身分管理系統的認知不足也會導致他們無法有效應對釣魚攻擊、社交工程等安全威脅。

為了解決這個問題，未來的身分管理解決方案應該朝多個方向發展。首要任務是簡化多層次、多維度的使用者身分管理，允許單一的身分管理多樣的別名，以適用於不同的場景。例如，使用者可以用唯一的帳戶創建三個別名，分別對應自己的三種社會身分：在家中是家長，在工作中是員工，在社交場合是朋友。甚至針對單一的服務，使用者也可以擁有多種別名，如在論壇中既可以以專家身分發表權威言論，也可以作為普通使用者表達個人觀點。遵循上述的設計可以幫助使用者在盡可能不增加認知負擔的情況下有效管理自己的身分。

然而，真正簡化使用者認知並非易事。即使是宣稱已解決這個問題的使用者中心身分系統，實際上也未能完全做到。以 Google 的組織管理文件 [19] 為例，為

了確保不同組織擁有不同的安全限制與規定，系統仍然要求使用者在不同組織間創建不同的身分。這表明，同時簡化使用者認知與滿足組織需求，仍然是一個有待解決的挑戰。



2.4.3 隱私保護

在數位時代，隱私保護已成為身分系統設計的核心考量之一。歐盟制定的《通用數據保護條例》(GDPR) [15] 代表了目前全球最嚴格的隱私保護標準。本研究認為，一個理想的身分系統應當能夠全面符合 GDPR 的要求，從而確保使用者隱私得到最大程度的保護。然而，近年來的 GDPR 違規案例表明，即便是大型企業也面臨著遵守某些 GDPR 規定的挑戰。

基於 Schardong 等人的研究 [42]，本研究發現當前身分系統中存在兩個尤為突出的關鍵問題。首先是使用者實現明確授權的困難。Saemann[40] 的研究強調，在當前的身分系統框架下，企業難以實現使用者對數據使用的明確授權。具體而言，企業難以證明其對數據或權限的使用行為已獲得使用者授權，而使用者也缺乏有效途徑證明自己的數據或權限被不當使用。這種情況不僅增加了企業的法律風險，也削弱了使用者對系統的信任。

第二個問題是實現被遺忘權的困難。Smirnova[45] 指出，滿足使用者的被遺忘權在當前身分系統中存在著一定的挑戰。使用者數據在系統中往往呈分散狀態，即使刪除核心使用者的資料，仍可能保留使用者的系統日誌或與其他使用者的互動數據。這種情況使得完全實現被遺忘權變得複雜而困難，可能導致使用者隱私無法得到全面保護。

基於以上分析，本研究提出符合現代隱私保護要求的身分系統應該具備兩個關鍵特點。首先，系統應提供合理的機制，讓使用者或企業能夠證明其數據使用

行為是否符合授權，這將有助於提高系統的透明度和可信度。其次，系統應提供有效的方法讓使用者行使被遺忘權，確保使用者不會被難以刪除的數據綁架。這意味著系統需要設計更精細的數據管理和刪除機制。在後續研究中，將詳細探討如何在自主身分系統中實現這些特性，並提出相應的技術解決方案。

2.4.4 平等信任

身分系統中的平等信任問題是一個複雜的多方利益平衡問題，涉及系統的公平性和可信度。Alex 等人 [39] 強調了身分系統中各方利益的衝突，主要表現在使用者之間的權益差異、不同系統間的互操作性問題，以及使用者與系統供應者之間的利益衝突。例如，身分系統供應者可能希望獲取更多使用者個人資料以獲取利益，而使用者則希望保護自己的隱私。這種利益衝突如果處理不當，可能導致系統環境惡化、使用者權益受到侵犯，以及市場壟斷和不公平競爭。Zuboff[52] 的研究進一步指出，這種數據收集和利用的不平等可能導致所謂的「監視資本主義」，對個人自由和社會公平造成深遠影響。

在當前的身分系統中，建立健全的信任模型仍然是一個重大挑戰。傳統的二元邏輯驗證模式（即完全信任或完全不信任）已不能滿足現代身分系統的需求。Schardong 等人 [42] 指出，現實世界的信任往往是模糊而不確定的，人們很難用簡單的真假邏輯分辨使用者是否可以被信任。例如，可能同時存在多個資訊來源，部分認為可信任，部分認為不可信的情況。Josang 等人 [24] 提出的主觀邏輯數學框架為處理這類多組不確定性信任的問題提供了一個系統性的解決方案。該框架將可信度表示為一個區間，從而更好地模擬了現實世界的身份信任過程。

此外，在去中心化系統中，解決身分驗證問題尤其困難。正如 Dhamija[14] 所強調的，使用者需要向系統證明自己的身分，同時系統也需要向使用者證明自己

的合法性和可信度。Tze-Nan[51] 提出的自主憑證機制為解決這一問題提供了一個新的思路。他改良了傳統的憑證簽署（Certificate Authority）技術，使憑證不僅可以由使用者自主操作，還能被所有經手者評分。如此一來，使用者和系統之間就可以在自主的前提下互相驗證並評分，從而建立起一個平等的信任關係。

然而，長期來看，一個合理的評分機制，甚至是一個長期的治理機制也是必要的。在這方面，Chohan 等人 [12] 關於 DAO（去中心化自治組織，Decentralized Autonomous Organization）治理的研究提供了核心概念，可以為自主身分系統的制度建設提供參考。

在後續的研究中，將探討如何在自主身分系統中實現技術上的互相驗證與制度上的互相信任，最終構建一個能夠平衡各方利益、促進公平競爭而不易壟斷的自主身分生態系統。這種系統將能夠在保護使用者權益的同時，也為系統供應者提供合理的發展空間，從而實現真正的平等信任。

2.4.5 法律合規性

身分系統在設計和實施過程中，除了需要應對技術上的關鍵挑戰外，還必須嚴格遵守相關的法律法規，尤其是在數據保護和隱私保護方面。這些法規旨在確保使用者的權益得到充分保護。以下是兩個具有代表性的法規和標準：

- **歐盟通用數據保護條例（GDPR） [15]**：這是當前全球最嚴格的數據保護法規之一。GDPR 要求企業在收集、存儲和使用用戶數據時，必須遵守一系列嚴格的規定，以保護用戶的隱私權和數據安全。
- **美國國家標準技術研究所（NIST）身分驗證指南 [35]**：NIST 制定了一系列身分驗證標準，涵蓋多因素驗證、風險評估、身分驗證和授權等多個方面。這些標準為身分系統的設計和實施提供了重要的技術指導。

身分系統的開發者和運營者必須充分理解並遵守這些法規和標準，以確保系統的合法性和可靠性，同時有效保護用戶的權益。



2.4.6 公認原則

在設計身分系統時，遵循身分識別領域的公認原則至關重要。這些原則為系統設計提供了重要的指導方針，有助於確保系統的合理性和有效性。包括但不限於：

- **Kim Cameron 的身分管理七大法則 [10]**：這些法則涵蓋了使用者控制、最小化披露和互操作性等關鍵概念，為身分管理系統的設計提供了全面的指導。
- **Dhamija 等人提出的身分管理七大缺陷 [14]**：這項研究指出了身分管理系統中常見的問題，包括使用者體驗、使用者認知和使用者信任等方面的挑戰，為系統設計者提供了重要的警示。
- **Allen 的自主身分十大原則 [4]**：這些原則強調了使用者控制、最小化披露和互操作性等概念，與 Cameron 的法則有所重疊，但更加聚焦於自主身分的特點。

這些公認原則雖然側重點有所不同，但都代表了各時期身分系統設計的最佳實踐，值得設計者們深入研究和遵循。

2.5 本章總結

本章回顧了現有的身分系統設計，並探討了身分系統在使用者體驗、使用者認知、隱私保護、平等信任、法律合規性和公認原則等方面面臨的挑戰。這些討

論有助於釐清身分系統的核心特徵，為未來的設計提供參考。在下一章中，將提出一個新的自主身分系統設計，以應對現有身分系統的挑戰。







第三章 系統設計

自主身分 (AID) 是由使用者個人裝置管理的網路唯一身分，其核心理念在於讓每位使用者在具備動態道德標準的身分系統中自由的管理自己，從而解決現有身分管理系統的問題。

本章節將全面探討自主身分 (AID) 系統的設計與實現。首先回顧並歸納過去的 AID 系統設計，從中汲取寶貴經驗。在此基礎上，將詳細闡述完整的 AID 設計理論，深入探討其核心概念和原則。隨後，本章節將提出一個更現代化、更符合當前技術趨勢和用戶需求的系統架構設計。

通過這種由淺入深、從理論到實踐的方法，本章節旨在為讀者提供一個全面而深入的 AID 系統設計藍圖，為未來的實現和應用奠定堅實的基礎。

3.1 系統的新設計

本研究不僅擴展了對「自主」概念的理解，更深入探討了自主性在身分系統中的應用。本研究將「自主」的定義從「使用者能夠完全掌控自己的數據和隱私」提升至「讓每位使用者在具備動態道德標準的身分系統中自由管理自我」。這一深化不僅強調了個人對數據的控制權，還引入了道德標準這一重要維度，為身分管理賦予了更深層的社會意義。

同時，本研究正面迎接身分系統與自主性之間的固有矛盾。身分的本質在於互動的識別，而 AID 系統卻致力於賦予使用者在個人裝置掌握自己的身分。這種看似不可調和的衝突，實則提供了重新思考身分管理的機會。為此，本研究進行了縝密的權衡和創新設計。提出的新方案能在保護個人自主權的同時，盡可能不損害身分系統的核心功能。

Yuxuan[30] 提出的 AID 系統可概括為三個核心設計：自主認證、數據自主和信用評分。本節將從這三個方向開始構建完整的 AID 理論體系。

3.1.1 自主認證

自主認證是 AID 系統的核心，畢竟如果使用者連登入都被其他人掌握，那麼談何自主。使用者認證流程的本質可以被理解為使用者向驗證方證明「認證因素」[5] 的擁有權。

3.1.1.1 最簡自主認證

本研究首先提出可以利用以下模型定義描述認證流程的本質：

- Function Definitions:
 - generate: $\emptyset \rightarrow (AID, token)$
 - verify: $(AID, token) \rightarrow \{0, 1\}$ // 0: fail, 1: pass
- System relationship:
 - $\forall (AID, token) = \text{generate}()$
 - $\text{verify}(AID, token) = 1$

這些函數定義了一個認證的過程，使用者可以通過 generate 函數獲取一個 AID 和 token，AID 被視作自主身分系統內的唯一識別號，而 token 伴隨 AID 產生，可以



作為 AID 的認證因素。因此 verify 函數只要檢測到 token 和 AID 的對應關係即可通過認證。

這樣的認證模式雖然能滿足基本需求，但在自主認證的背景下顯得不夠完善。自主認證期望使用者能夠在個人裝置上獨立管理認證因素，無需依賴外部系統完成身分驗證。這意味著認證所需的敏感信息（即 token）應當留存在使用者設備內，而非傳送至其他地方。

基於這一理念，本研究提出了一個新的模型，用以描述自主認證的最基本流程：

- Function Definitions:

- generate: $\emptyset \rightarrow (AID, token)$
- proof: $token \rightarrow one-time-proof$
- verify: $(AID, one-time-proof) \rightarrow \{0, 1\}$ // 0: fail, 1: pass

- System relationship:

- $\forall (AID, token) = \text{generate}()$
- $\forall one-time-proof = \text{proof}(token)$
- $\text{verify}(AID, one-time-proof) = 1$

模型中引入了 proof 函數，它能將 token 轉換為一次性的驗證證明。這一機制使得使用者可以在不直接暴露 token 的前提下完成身分認證。成功地在保障使用者密鑰和實現有效認證之間取得了平衡。

基於前述推演，本研究提出了一個符合 AID 價值觀的最簡認證方案（如圖3.1所示）。這個方案實現了在不暴露完整認證因素的情況下進行身分驗證，從而實現了 AID 系統追求的自主性目標。換言之，使用者可以獨立證明自己的身分，無需依賴任何外部系統或機構。

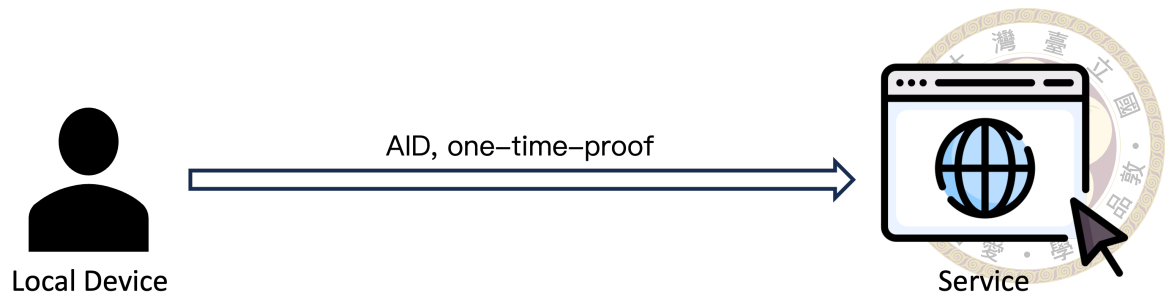


Figure 3.1: 最簡自主認證

在實際應用中，這一方案可通過多種技術來實現，如公私鑰對系統或零知識證明等。舉例來說，一種簡單的實作方式是：

- 將公鑰視為 AID
- 將私鑰視為 token
- 使用私鑰對隨機數進行簽名，並將簽名作為 one-time-proof

3.1.1.2 MFA 的加入

雖然前述的最小設計為自主認證奠定了基礎，但面對當今身分管理領域的複雜挑戰，其局限性日益顯現。Bonneau 等人 [9] 的研究強調，單一認證方式（如私鑰或帳密）已無法應對現今嚴峻的安全威脅。多因素認證（Multi-Factor Authentication, MFA）因此成為必要。這一觀點得到了權威機構的認可，其中美國國家標準暨技術研究院（NIST）在其指南中明確強調了 MFA 的關鍵作用 [35]。

基於這些考量，本研究提出了一個更為全面的自主認證方案模型。這個進階模型不僅保留了原有的自主性特徵，還支持多種認證因素的整合，為使用者提供了更強大的身分保護。

- Function Definitions:
 - generate: $\emptyset \rightarrow (AID)$
 - bind: $(AID, token) \rightarrow AID$



- $\text{proof}: \text{token} \rightarrow \text{one-time-proof}$
- $\text{verify}: (\text{AID}, \text{one-time-proof}) \rightarrow \{0, 1\}$
- System relationship:
 - $\forall \text{one-time-proof} = \text{proof}(\text{token})$
 - $\text{verify}(\text{bind}(\text{AID}, \text{token}), \text{one-time-proof}) = 1$

本研究提出的改進模型在原有基礎上做出了兩項關鍵調整：首先，將 AID 的生成過程獨立化；其次，引入了 bind 函數，用於將不同的認證因素 (token) 綁定到 AID 上，從而實現多因素認證 (MFA)。

然而，這種看似完善的設計卻帶來了新的挑戰，主要體現在系統複雜度的增加。在現有密碼學技術下，要在數學上嚴格證明某個認證因素與獨立生成的 AID 之間的關聯性仍然困難重重。因此，實現這一模型往往需要依賴更為複雜的系統設計和實作。

值得一提的是，Wang 等人 [48] 提出的門限簽名方案為解決這一問題提供了一個潛在途徑。該方案允許在生成單一公鑰的同時產生多個私鑰，理論上可以在不需額外 bind 機制的情況下實現上述模型。然而，這種方案在實際應用中面臨諸多限制，難以被視為 MFA 的完整解決方案。

3.1.1.3 AID Server 的加入

Yuxuan 設計的 AID 系統 [30] 可被視為 MFA 模型的一個實際案例（如圖3.2）。該系統的核心機制如下：

1. 產生 ID：使用者在個人裝置上生成 UUID[29] 作為唯一識別號。
2. 產生認證因素：由於 UUID 本身不具備可認證特性，使用者還需在裝置上生成一對公私鑰。

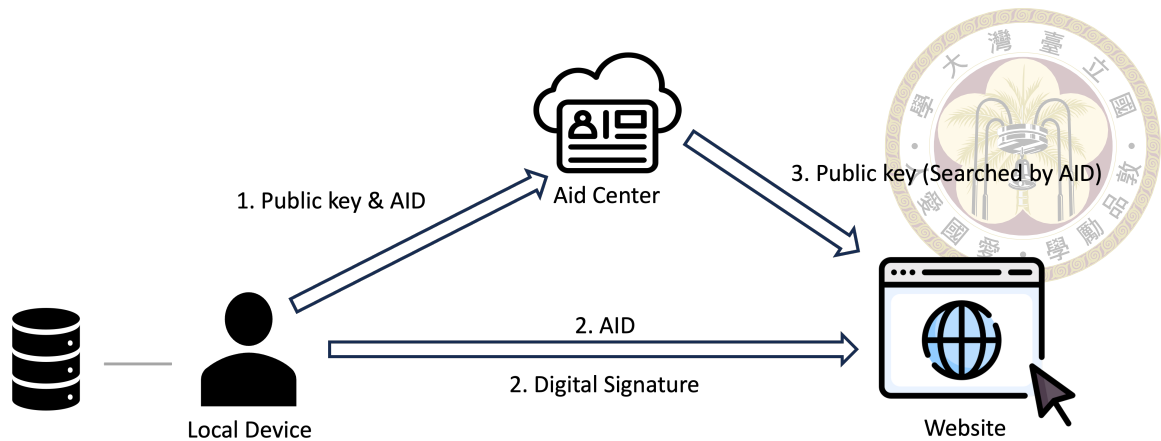


Figure 3.2: 過去的自主認證

3. **連結認證因素**：將公鑰與 UUID 一同上傳至中心化 AID Server，建立連結。
4. **身分認證**：認證時，使用者通過私鑰對隨機數據進行簽名，驗證者則可從 AID Server 獲取對應 UUID 的公鑰來驗證簽名。

這種設計不僅實現了基本的身分認證功能，還通過中心化的 AID Server 巧妙解決了兩個關鍵安全問題：

- **ID 重複問題**：儘管 UUID 重複的機率極低，但在實際場景中，惡意使用者可能刻意生成相同的 UUID，導致身分混淆。AID Server 通過確保每個傳入的 UUID 的唯一性，有效防止了這一問題，拒絕任何重複 UUID 的綁定請求。
- **身分強佔問題**：作為 ID 重複問題的延伸，身分強佔指惡意使用者可能在 UUID 真實擁有者使用特定服務前，搶先綁定該 UUID，致使真實擁有者無法使用服務。AID Server 通過在使用者進入服務時驗證 UUID 與公鑰的對應關係，有效規避了這一風險。

深入探討上述流程，本研究發現了一些問題，首先，中心化的 AID Server 雖然解決了 ID 重複問題與身分強佔問題，但是卻導致了新的問題如單點故障、可擴展性等。其次，中心化的 AID Server 也無法滿足 AID 系統的最初目標：自主認證。使用者無法完全自主管理自己的身分，而是需要依賴中心化的 AID Server 提

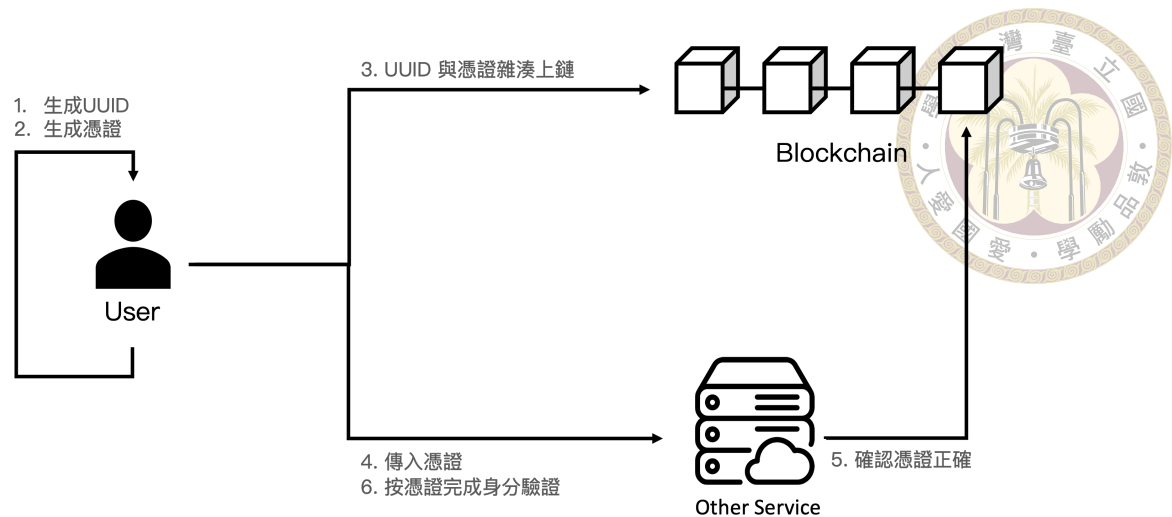


Figure 3.3: 自主認證 (加入憑證機制)


供的連結能力才能完成認證。為了解決這些問題，本研究提出以下原則希望能夠重新設計 AID 系統：

- **去中心化**：把區塊鏈技術應用到 AID 系統的實作中，從而消除中心化的 AID Server，實現系統的高可用性和可信任。
- **去依賴性**：使用者認證時不應該要求服務到 AID Server 搜索綁定 UUID 的公鑰，而應該是讓使用者自主宣告綁定的公鑰等關聯認證因素，AID Server 只負責記錄這個宣告，作為額外的保證。

3.1.1.4 自主憑證的使用

最後，單純的加入區塊鏈技術取代 AID Server 並不是沒有缺點，除了區塊鏈技術的性能限制外，還有一個更為重要的問題：**隱私**。區塊鏈技術的特性決定了所有的交易都是公開的，這將導致使用者的綁定關係被公開，進而影響使用者的隱私。

過去的設計中僅使用公鑰作為綁定因素尚無隱私問題，但 MFA 機制中常用的手機驗證碼、信箱驗證碼等都是私密資訊，這樣的設計將導致使用者的隱私受到威脅。



為解決前述挑戰，本研究對 Tze-Nan[51] 提出的「自主憑證」概念進行了簡化和優化。如圖3.3所示，新的自主認證解決方案建立在一個新機制上。在這個機制中，使用者首先將個人資訊和多因素認證（MFA）所需的驗證資訊整合到類似 X.509 的憑證結構中。隨後，系統計算該憑證結構的雜湊值，並將其上傳至區塊鏈。當使用者需要登入時，他們只需向登入對象提供完整憑證。登入對象則通過區塊鏈驗證憑證的完整性，一旦驗證通過，使用者就能用憑證中提供的 MFA 方案和資訊完成登入程序。另外，如果使用者在憑證內的個人資訊需要有第三方背書，可以透過追加簽章服務後在憑證內部加入第三方簽章來完成。

這種設計極大地增強了系統的安全性和隱私保護能力。首先，由於使用者自行保管包含個人資料和驗證資訊的完整憑證，個人隱私得到了有效保護。其次，區塊鏈僅存儲憑證的雜湊值和使用者的 UUID，大幅減少了敏感資訊外洩的風險，實現了資料最小化原則。再者，通過區塊鏈驗證憑證完整性，結合 MFA 機制，確保了整個認證過程的安全可靠。最重要的是，這個方案賦予了使用者更大的自主權，使他們能夠自主決定何時、向誰提供憑證，從而增強了對個人身分資訊的掌控。

3.1.1.5 自主認證流程

這個優化後的「自主認證」方案不僅有效平衡了隱私保護和認證安全性，還為使用者提供了更大的身分資訊自主權，充分體現了 AID 系統的核心理念。以下深入描述這個方案的細節流程：

1. 使用者在個人設備上：

- (a) 產生 UUID
- (b) 選擇認證因素（如手機、信箱、金鑰等）

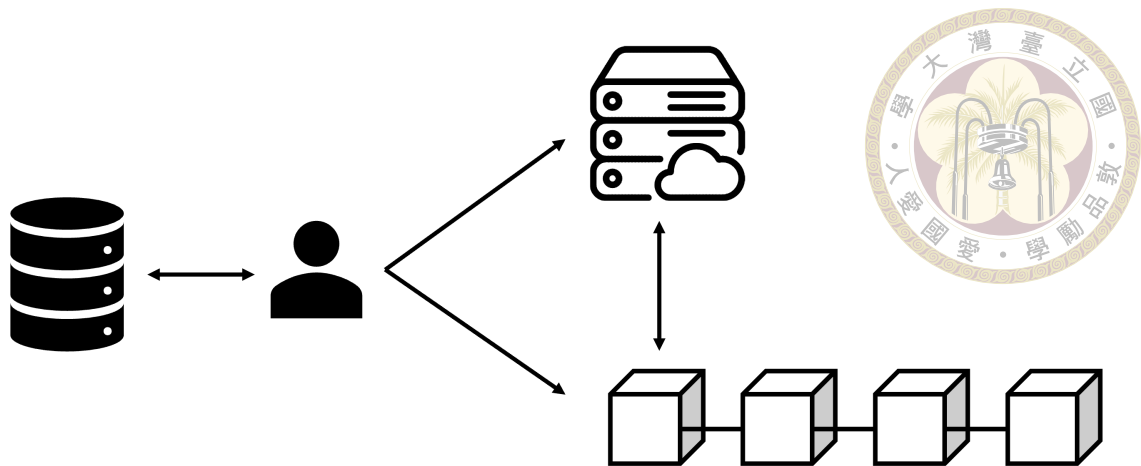


Figure 3.4: 自主身分

(c) 將對應資訊（如手機號碼、信箱地址、公鑰等）加入憑證

2. 憑證處理：

(a) 填入欲揭露的個人資訊（類似過去 AID 中的 actor profile）

(b) 計算憑證的雜湊值

(c) 將雜湊值上傳至區塊鏈

3. 登入流程：

(a) 使用者將憑證傳送給登入對象

(b) 登入對象在區塊鏈中透過檢查雜湊值驗證憑證的完整性

(c) 登入對象要求使用者使用憑證內的驗證方案完成認證

(d) 使用者依要求完成認證（例如：使用私鑰對隨機數進行簽名來證明擁有公鑰）

3.1.2 數據自主

實現數據的自由管理是一項具有挑戰性的任務。在傳統身分管理系統中，使用者數據通常由身分供應商（Identity Provider）或服務提供商（Service Provider）集中控制，這嚴重限制了使用者在個人數據上的自由。為了克服這一限制，AID 系統採用了獨特的「數據層反轉」策略如圖3.4：將使用者數據完全遷移至使用者

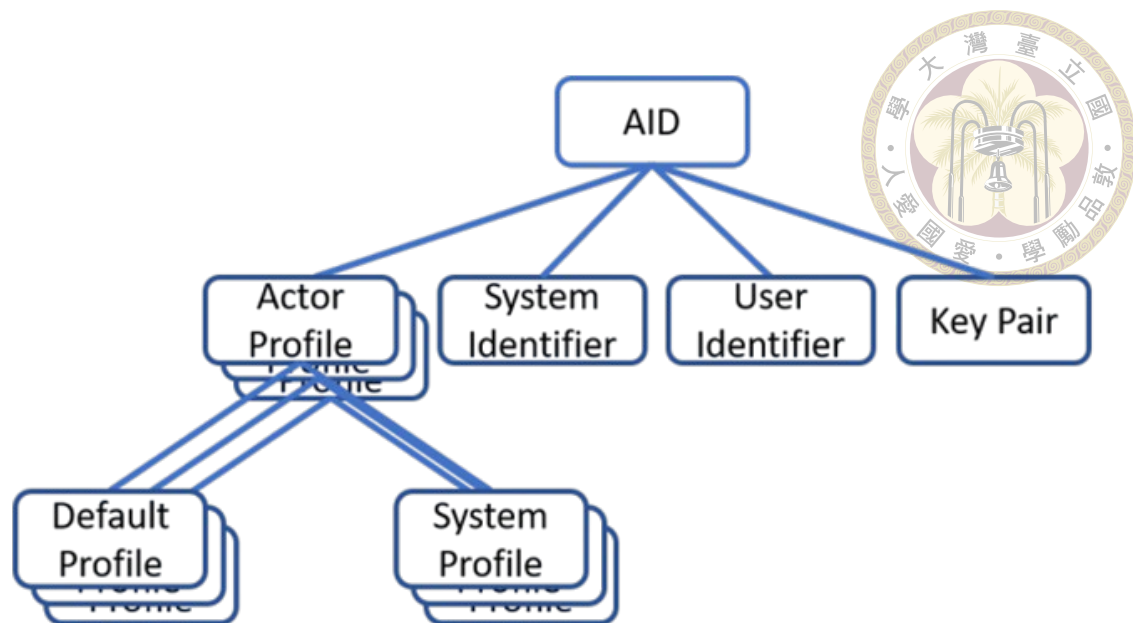



Figure 3.5: 過去的 AID 資料結構

端設備，從而實現使用者對個人數據的直接控制。

更深入的描述，在過去的 AID 系統中，在個人裝置保留角色資料（actor profile）的資料結構（如圖3.5）。其中，每個角色資料包含了一組預設資料與多組服務資料。預設資料是使用者自行提供的基本資料，如姓名、性別、年齡等。服務資料則是使用者在使用各服務時產生的資料，如交易記錄、行為日誌等。設計上，盡可能在個人裝置上使用服務，當不得不上傳資料時，使用者可以選擇上傳哪個角色的哪組資料，從而實現數據的自主權。

然而，本研究認為這樣的設計僅實現了 GDPR[15] 中強調的「數據最小揭露」，並沒有完整面對數據自主背後的艱困挑戰。就像蘋果公司 AI 服務 [6] 的隱私困境，AI 模型需要讀取使用者的隱私才能更好的協助使用者。確實存在著理想的做法，就是只使用個人設備內的小型 AI 模型產生推論，而不是將所有數據上傳到雲端進行運算。但是，這樣的做法在實際應用中卻面臨著巨大的挑戰，因為小型 AI 模型的能力有限，無法滿足複雜的應用需求。因此，蘋果公司會在個人設備的小型 AI 模型無法妥善處理的情況下，使用特殊的私有雲計算方案 [7] 來保護使



挑戰	說明	關鍵概念	技術解決方案
數據被遺忘權	使用者有權要求完全刪除其個人數據	<ul style="list-style-type: none"> 分散存儲帶來的挑戰 	<ul style="list-style-type: none"> 無狀態服務 數據不可定向性
數據明確授權	收集數據前需獲得明確同意並說明用途	<ul style="list-style-type: none"> 目的明確原則 使用者體驗設計 使用者認知負擔 	<ul style="list-style-type: none"> 數據憑證機制 數據流動的透明化
數據可驗證性	確保數據的正確性和一致性	<ul style="list-style-type: none"> 跨服務數據傳遞流程 去中心化帶來的挑戰 	<ul style="list-style-type: none"> 數據憑證機制 區塊鏈智能合約
無特權的執行	減少對特權管理者的依賴	<ul style="list-style-type: none"> 使用者自主管理 便利性與責任平衡 	<ul style="list-style-type: none"> 極限多因素驗證 暫時提權方案


Table 3.1: 數據自主的四個難題

用者的隱私。

基於這樣的背景，本研究提出數據自主權應考慮到**不得不上傳的數據與直接由服務產生的個人數據**，並直接面對表3.1幾個關鍵挑戰。

3.1.2.1 數據被遺忘權

數據被遺忘權（Right to be Forgotten）是現代數據保護法規中的一項關鍵原則，它賦予使用者要求服務供應商完全刪除其個人數據的權利。然而，在技術實現層面，這個看似簡單的要求卻面臨著巨大的挑戰。實現數據被遺忘權的主要技術難點在於使用者資料通常以各種形式分散存儲在系統的不同部分，包括主資料庫、日誌系統、備份存儲等[45]。此外，某些數據可能與其他使用者的數據或系統日誌緊密關聯，難以單獨刪除。



為了應對這些挑戰，本研究提出了結合自主身分（AID）系統和無狀態服務（Stateless Service）概念的解決方案。AID 系統的核心理念是將主要數據存儲在使用者個人裝置中，而非集中在服務供應商的伺服器上。這種去中心化的方法大大簡化了數據刪除的難度。與此同時，借鑒了 [26] 的研究，本研究提出服務供應商應採用無狀態設計原則。在這種設計下，每次交互結束後，服務供應商會刪除與使用者相關的所有臨時數據。對於必須保留的數據（如交易記錄），則由使用者下載並自行管理，在下次使用服務時重新上傳必要的數據。

這種結合 AID 系統和無狀態服務的方法具有多重優勢。首先，它極大地簡化了刪除流程，因為大部分數據由使用者自行管理，服務供應商只需處理少量臨時數據。其次，這種方法顯著增強了隱私保護，降低了數據被不當收集或利用的風險。此外，它提高了整個數據處理過程的透明度，使用者能夠清楚知道哪些數據被服務存儲，哪些由自己管理。最後，這種方法更容易滿足如 GDPR 等嚴格的數據保護法規要求。

儘管無狀態服務為實現數據被遺忘權提供了有效途徑，但這種方法也引發了一系列新的挑戰，尤其是在數據信任和適用性方面。首要問題是數據的可信度。當服務要求使用者保存具有實質價值的狀態數據時，如可兌換現金的積分或重要的交易記錄，這些存儲在使用者端的數據可能面臨被惡意修改的風險。這不僅威脅到服務的正常運作，還可能給服務供應商帶來巨大的經濟損失。

為了解決這個信任問題，本研究提出了一種基於密碼學的驗證機制。這種機制的核心是利用數字簽名技術確保數據的完整性和真實性。具體而言，服務供應商會對關鍵數據的雜湊值進行加密簽名，然後將原始數據和簽名一併返回給使用者。在使用者再次使用服務時，需同時上傳數據和對應的簽名。服務供應商通過以下步驟驗證數據的完整性：



1. 接收使用者上傳的數據和簽名。
2. 使用存儲的私鑰解密簽名，獲得原始雜湊值。
3. 對接收到的數據重新計算雜湊值。
4. 比較解密後的雜湊值和新計算的雜湊值。

如果兩個雜湊值相符，則可以確認數據在使用者端存儲期間未被篡改。這種方法既保證了數據的完整性，又維持了無狀態服務的核心優勢。

然而，本研究必須認識到，無狀態服務並非適用於所有場景。某些應用類型，如社交網絡平台或在線論壇，其核心功能就是依賴於在平台上持續存儲和展示使用者生成的內容。對於這類應用，完全採用無狀態服務模式是不切實際的。

在這種情況下，本研究提出了一種基於「不可定向性」原則的替代方案。這一方案即使在必須保留使用者數據的情況下，也能在一定程度上保護使用者隱私。其概念是將使用者上傳的數據做去識別化處理，使其無法直接與特定使用者關聯。具體而言，可以將使用者數據中的個人識別信息（如姓名、地址、實名 AID 等）替換為標識符，使用者可以單向的證明自己擁有數據，但服務沒有辦法直接證明數據與特定使用者的關聯。這種方法在一定程度上保護了使用者隱私，同時也滿足了服務的核心功能需求。

總的來說，無狀態服務雖然為數據被遺忘權提供了有力支持，但並非放之四海而皆準的解決方案。在實際應用中，需要根據具體情況選擇適當的技術方案，並在數據保護、使用者體驗和服務功能之間尋求最佳平衡點。

3.1.2.2 數據明確授權

近年來，隨著數據保護法規的不斷演進，特別是歐盟通用數據保護條例（GDPR）的實施，數據授權問題已成為企業，尤其是大型跨國企業面臨的重大挑

戰之一。Saemann[40] 的研究深入探討了 GDPR 實施後企業在數據管理實踐中遇到的困難，特別強調了目的明確（Purpose Specification Principle）的複雜性。企業必須在收集個人數據之前獲得數據主體的明確同意，並清晰闡明數據的使用目的和範圍。這一看似直接的要求，實則蘊含了多層次的法律和技术挑戰。

以 Google 和 Amazon 為例，這兩家公司儘管在數據收集過程中提供了全面的隱私設定選項，但仍被歐盟監管機構認定為未能充分獲得使用者的明確同意。2019 年，法國國家資訊自由委員會（CNIL）對 Google 處以 5000 萬歐元的罰款，指出其隱私政策缺乏透明度和具體性 [22]。同樣，2021 年盧森堡國家數據保護委員會（CNPd）因 Amazon 的廣告個性化做法違反 GDPR，對其處以 7.46 億歐元的罰款 [44]。

這些案例揭示了即便是資源豐富的科技巨頭，在實現 GDPR 合規性時也面臨著重大挑戰。相關判決強調，僅僅提供隱私設定選項是不夠的，關鍵在於這些選項的設計和呈現方式。具體而言，這些公司的做法之所以被認定為不符合 GDPR 標準，主要基於兩個方面的考量：

1. **使用者體驗設計不當：**隱私設定介面往往缺乏直觀性和易用性。複雜的選項結構和專業術語的使用增加了使用者理解和操作的難度。介面設計對使用者的隱私決策有顯著影響，不恰當的設計可能導致使用者做出與其真實意願不符的選擇。
2. **使用者認知負擔過重：**大多數使用者面對繁複的隱私設置時，要麼難以充分理解每個選項的含義及其潛在影響，要麼不願投入大量時間來仔細配置這些選項。即使是受過教育的使用者也可能因為認知偏差和決策疲勞而做出次優的隱私決定。

這些問題凸顯了在設計符合 GDPR 要求的數據收集機制時，不僅需要考慮法律合

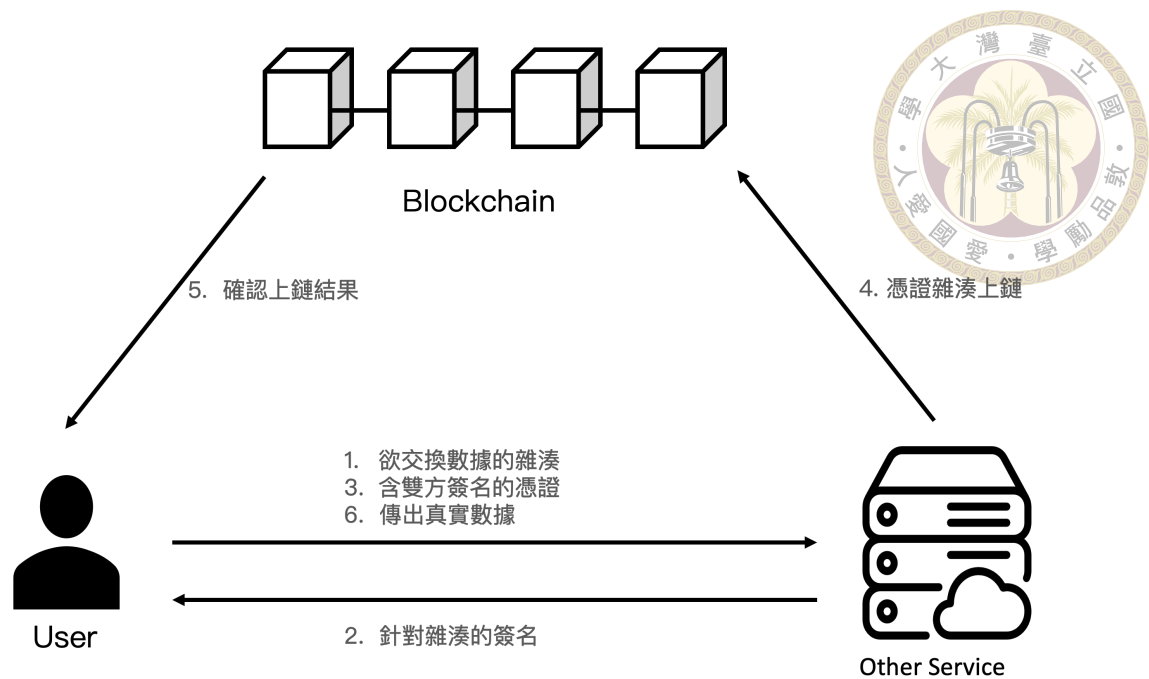



Figure 3.6: 數據憑證（資料傳出）

規性，還需要從使用者角度出發，提供直觀、易用的隱私設定選項。

因此，本研究提出「數據憑證」(Data Certificate) 概念（如圖3.6）。這一機制借鑒了自主憑證 [51] 的設計，為每份數據賦予一個包含多重驗證要素的電子憑證。具體而言，數據憑證包含以下關鍵元素：數據所有者的數字簽名、數據使用者的數字簽名、數據內容的加密哈希值、明確定義的使用範圍及目的。為確保憑證的不可篡改性和公開可驗證性，本研究進一步建議將憑證的雜湊值記錄於區塊鏈網絡中。

在此機制下，數據傳輸遵循嚴格的授權流程：僅當數據所有者和使用者雙方完成憑證簽署，且憑證雜湊值成功記錄於區塊鏈後，實際的數據內容才會被傳輸。這種設計明確定義了數據的內容與授權範圍，為數據使用提供了雙向保障。一方面，當發生數據濫用時，數據所有者可通過公開憑證證實濫用行為；另一方面，服務提供者也可藉由憑證證明其數據使用的合法性。結合優化的使用者互動流程，此機制有望實現更為明確和透明的數據授權過程。



此外，通過在區塊鏈上額外記錄數據特徵，本機制為個人數據流動提供了追蹤能力，從而增強了數據管理的「透明度」，這是保障隱私的關鍵環節之一。然而，值得注意的是，儘管此方法能夠明確記錄數據流動，但其有效性在很大程度上仍依賴於系統參與者的誠信和遵守規則的意願，可能在實際應用中面臨執行挑戰。鑒於此，本研究建議將數據憑證機制視為一個基礎框架，需要與其他輔助機制結合使用以增強其實際效力。例如，引入基於信用評分系統可以提升使用者對服務提供者的信任度，彌補單純依賴區塊鏈記錄的潛在不足。

3.1.2.3 數據可驗證性

在自主身分（Autonomous Identity，AID）系統中，使用者自行存儲和管理數據。這種去中心化方法雖然增強了使用者對個人數據的控制，卻也為跨服務的數據操作帶來了新的挑戰，特別是在數據的正確性（correctness）和一致性（consistency）方面。雖然數據正確性問題可以通過數位簽名技術解決，但這種方法無法解決數據一致性問題，即無法確保當前使用的數據版本是最新的，而非過時或已失效的版本。

為此，本研究提出「數據憑證」機制的不同用法，借鑒了區塊鏈在分布式系統中維護一致性的優勢。具體實施過程如下：當需要進行跨服務數據傳遞時，原始服務首先生成一個數據憑證，其中包含數據雜湊、時間戳和使用範圍等信息，並用私鑰進行簽名。隨後，原始服務將該數據憑證的雜湊值存入區塊鏈的智能合約中。這個智能合約支持即時狀態更新，確保數據的最新狀態可被追蹤。在跨服務傳遞數據時，使用者同時提供數據本身和對應的數據憑證。接收服務通過驗證憑證簽名來確認數據的正確性，並通過查詢區塊鏈上的智能合約來驗證數據的一致性。當數據發生變化時，相關服務可以即時更新智能合約的狀態，從而持續確保數據的一致性。

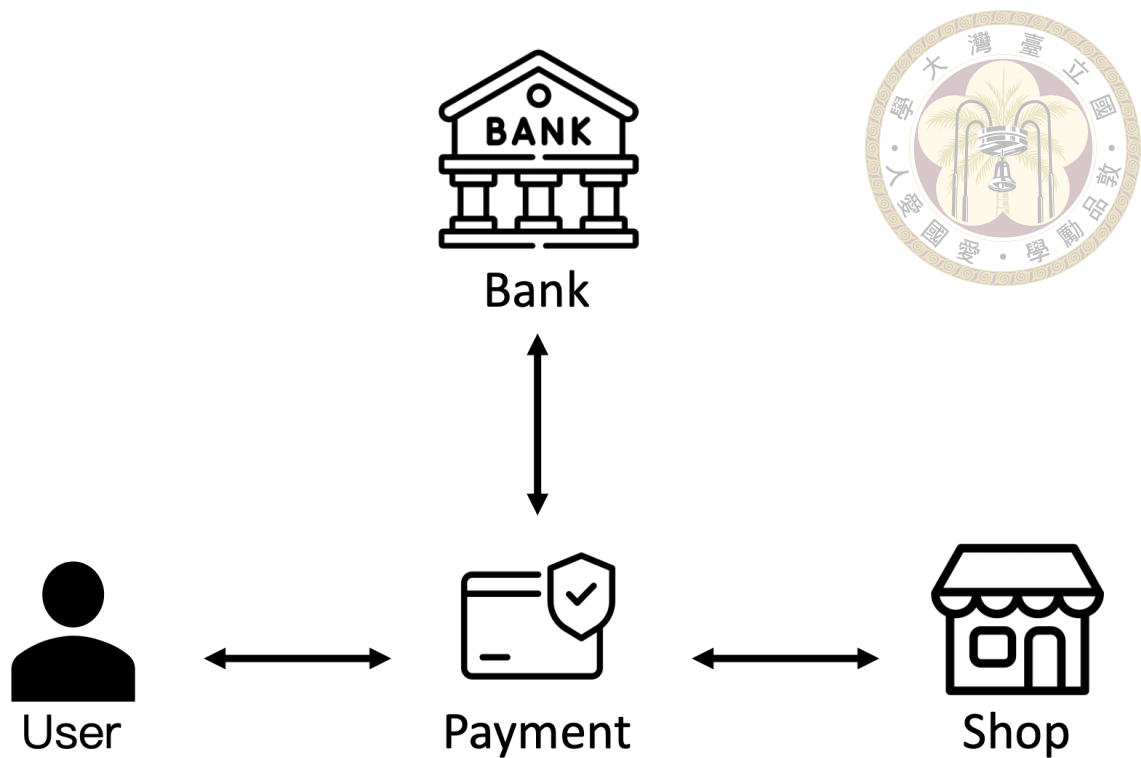


Figure 3.7: 當前網路後端設計

這種設計不僅確保了數據的「可驗證性」(Verifiability)，還為 AID 系統中的數據提供了「可移動性」(Portability)。可驗證性使接收服務能夠驗證數據的來源、完整性和時效性，確保數據的正確性和一致性；而可移動性則允許數據在不同服務間自由傳遞，無需依賴中央認證系統，同時保持可驗證性。這種方法具有高度安全性、實時性、可審計性和隱私保護等優勢，特別適用於對數據完整性和即時性要求較高的應用場景，如金融交易系統、醫療信息交換平台和供應鏈管理系統等。

舉例來說，實作一個使用第三方支付服務的購票系統時，本研究的方法與現行的網路後端設計3.7有所不同。現有設計中，支付系統全權負責與相關服務的串接，使用者僅需與支付系統溝通。而在本研究提出的架構中如圖3.8，流程如下：

1. 購票系統要求使用者到銀行服務完成支付並取得收據。
2. 銀行服務將收據的雜湊寫入區塊鏈。
3. 使用者向購票系統提交收據和購買請求。

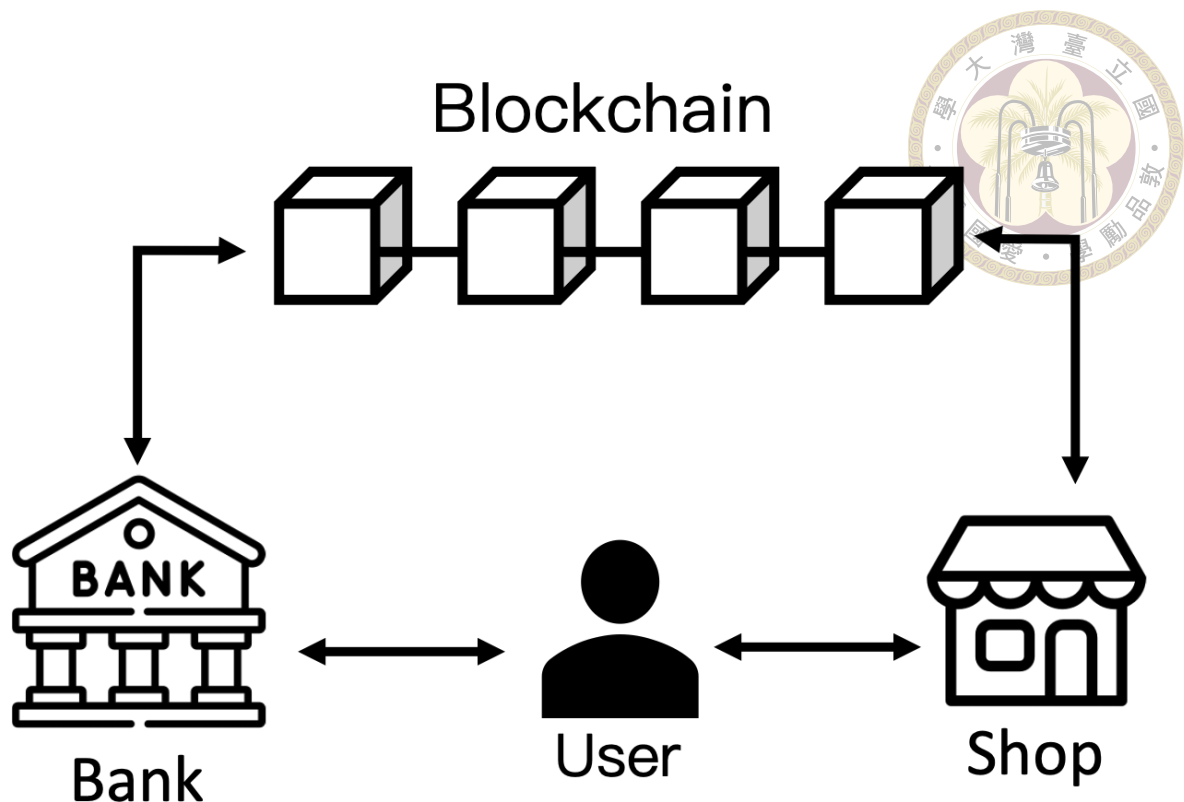


Figure 3.8: 跨服務共識流程

4. 購票系統在區塊鏈上驗證收據的真實性。

5. 驗證通過後，完成購票流程。

這種設計方式將使用者置於服務提供者之間，使得使用者能夠自主控制整個支付流程，充分展現了自主身分系統的核心理念。

儘管這種方法為基於 AID 的微服務架構中的數據管理提供了安全且靈活的解決方案，但仍存在一些需要進一步研究的問題。例如，在大規模系統中如何優化區塊鏈性能以支持高頻率的數據更新，如何使這種技術方案符合不同國家和地區的數據保護法規，以及如何讓使用者的個人裝置成為橋樑，主動從眾多服務中選出目標並進行串連。這些都是未來研究的重要方向。

總的來說，通過結合數據憑證和區塊鏈技術，這種方法為微服務架構中的數據管理提供了一個既安全又靈活的解決方案，尤其適用於需要高度數據完整性和

即時性的應用場景。隨著技術的不斷發展和完善，這種方法有望在未來的分布式系統設計中發揮更大作用，為數據的安全傳輸和管理提供新的模式。



3.1.2.4 無特權的執行

本研究認為無特權的執行也是 AID 系統必要的一環，因為即使服務提供者有著最好的意圖，但是只要存在特權管理者，就意味著使用者還是被迫要信任他人。但是，要求使用者擺脫對管理者的依賴絕非易事，因為這會導致使用者需要完全對自己的行為負責，例如不可以遺失密碼、不可以被盜用等，這樣的要求對於大多數使用者來說是不切實際的。

因此，本研究提出基於「極限多因素驗證」的暫時提權方案，當使用者需要執行某些特權操作時，服務會要求使用者用更多個驗證因素證明自己，以此達成更高的可信度。這樣的設計不僅可以讓使用者自主管理自己，也可以盡可能維持使用者的便利性與認知範圍。

3.1.3 信用評分

在過去的 AID 系統設計中，每個 AID 的角色（actor）都會在 AID Server 中被服務提供者評價，而所有系統參與者都可以查詢這些評價，藉此判斷對方的信用與真實性。然而，本研究認為這樣的做法存在問題如下：

1. 整個系統的信任，而非對使用者信任：評價機制僅針對使用者的行為，忽視了服務提供者的信用，而後者實際上更需要受到關注。
2. 尊重個人的道德標準：系統參與者無法自主決定如何被評價或如何評價他人，評價方法與標準完全由 AID Server 制定。這與 AID 中「自主量化」的目標相悖。

3. **AID 生態的建立**：整個信用系統的維運方式存在問題。AID 系統需要一個完整的生態系統才能長期運行這樣的信用評分機制。



為了解決這些問題，本研究提出了一個基於區塊鏈的信用評分機制，這個機制將信用評分的權力交還給使用者，讓每個參與者都可以自主決定自己的信用標準。這樣的設計不僅可以提高系統的透明度和公正性，還可以激勵參與者更積極地參與到 AID 生態中來。

3.1.3.1 信用評分機制

本研究把整個自主身分系統視為一個去中心化自治組織 (DAO)。每個系統參與者（包含服務供應者和終端使用者）都擁有自己的標準來基於評價判斷信譽，同時每個參與者也有權決定自己應該如何被評價。為了實現這一目標，本研究提出了完整的流程來描述評價與信譽轉換：

1. **產生憑證**：系統參與者創建「數據憑證」或「自主憑證」後，可以把對應雜湊放在自己決定的智能合約中上鏈，這個智能合約可以自訂許多規則，如是否允許別人對憑證進行評價，又或是評價的格式或條件等。所以，在本研究設計的 AID 系統中，使用者如果要允許評價，必須要把個人不同的角色資料 (actor profile) 轉換成憑證上鏈。
2. **產生評價**：取得明文憑證的人可以在區塊鏈上找到對應的智能合約，根據合約的規則對憑證進行評價。評價的內容必須遵循智能合約的規則，否則無法上鏈。
3. **信譽轉換**：用有明文憑證的人找到對應智能合約後可以讀取評價列表，之後根據自己認同的計算標準或演算法將評價轉換為信譽值。這個信譽值可以用於後續的信任判斷。

進一步來說，一些使用者可能希望根據評價者的信譽值來調整每條評價的權重，以更有效地防範惡意評價。但是，設計此功能並非易事。它可能需要引入評價者作為系統中的新角色，並考慮如何綜合多個評價者的評分結果[24]。最後，還需要考慮如何通過智能合約來實現這些複雜的功能。

3.1.3.2 生態系的營運

智能合約作為 AID 核心的信任機制載體，需要同時思考 DAO 治理的問題，這涉及到區塊鏈的經濟模型。本研究提議在區塊鏈中發行一種逐漸增加的自主身分統治代幣。這種代幣可被抵押，用於創建使用者的自主身分，藉此防止惡意使用者大規模創建惡意帳戶。每個自主身分（相當於其背後的代幣）可參與定期投票，討論評價機制的調整、新的智能合約協議等議題。為確保區塊鏈的長期運作，本研究建議對失去信任的使用者實施懲罰，同時獎勵基礎設施的運作。因此，用於創建自主身分的代幣抵押後不可贖回，被鎖定在區塊鏈上。因此，可以確保使用者不會輕率創建自主身分，並激勵使用者維護自身自主身分的信譽。此外，本研究建議對使用者在鏈上的每項操作採取使用者付費模式，使區塊鏈的維護者能獲得報酬，從而保證區塊鏈的持續運作。

3.2 系統架構設計

基於上述核心機制，本研究提出了一個完整的自主身分系統架構。這個架構融合了系統的技術層次和參與者角色，形成了一個統一且高效的生態系統。本節將詳細介紹這個架構的結構和設計理念。

層次	角色	主要功能
共識層	共識核心	提供可信賴的數據讀寫機制
服務層	服務提供者	提供特定服務, 不直接儲存使用者資料
數據層	終端使用者	管理個人數據, 使用服務

Table 3.2: 自主身分系統層次與角色對應

3.2.1 系統結構概覽

自主身分系統架構包含三個層次，由上到下分別是共識層、服務層、數據層，每個層次對應一種關鍵角色分別是共識核心、服務提供者、終端使用者。這種對應既反映了系統的技術架構，也體現了各參與者在系統中的功能和職責。圖3.9展示了這種層次-角色對應關係，表3.2則展示了各自的存在目的。

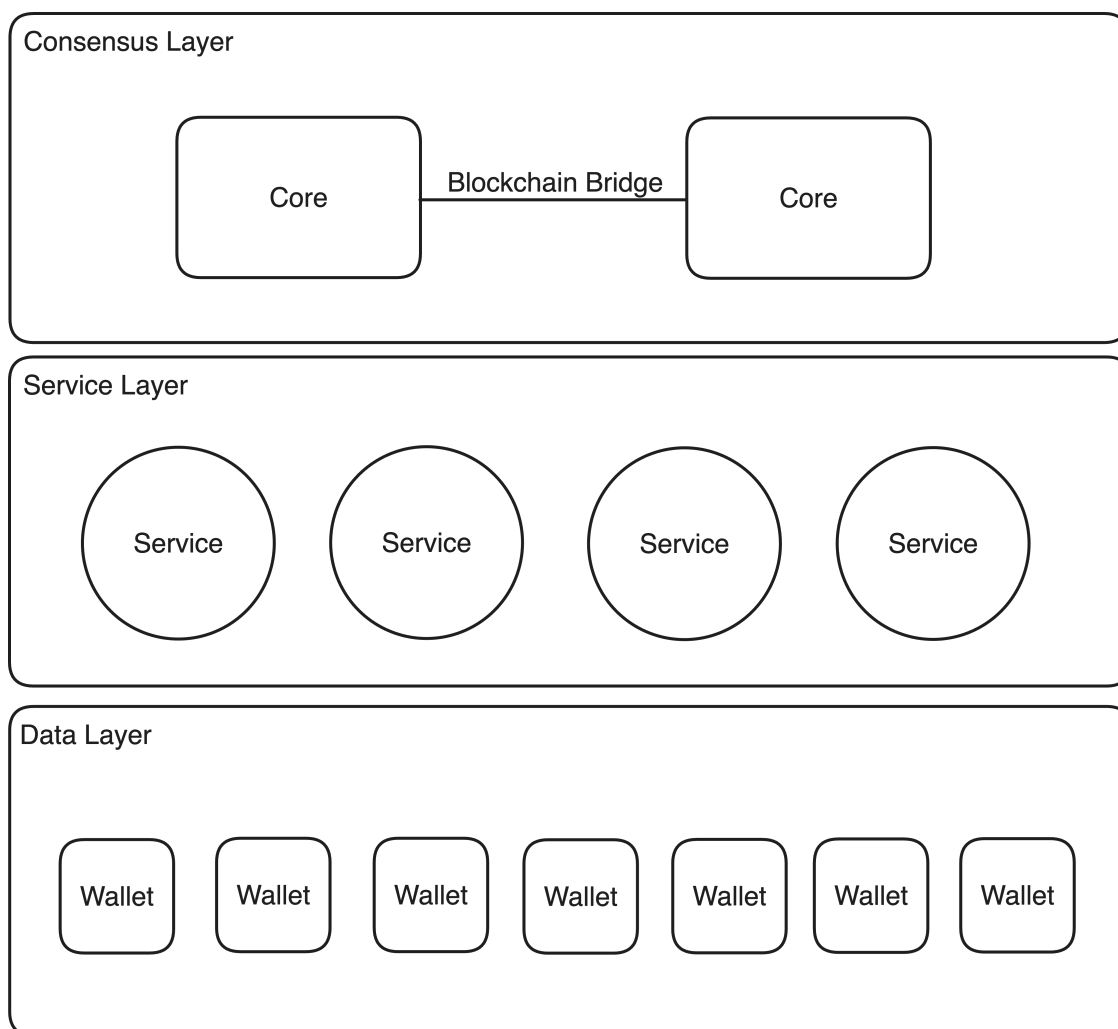


Figure 3.9: 自主身分系統結構圖



3.2.2 層次與角色對應

自主身分系統是個龐大的系統，每個層次內可以包含多個能平行擴展的角色實作，因此為了確保各個層次的模組化、易抽換、可擴展、靈活、安全和互操作等優勢，本研究明確訂出了各層的功能和職責。

3.2.2.1 共識層與共識核心

共識層在自主身分系統中扮演著基礎設施的角色，其主要參與者是共識核心。共識核心通過提供可信賴的數據讀寫機制，確保了「數據憑證」與「自主憑證」等機制在達成共識方面的可能性。其設計目的是確保數據的一致性和可信度，為整個系統提供堅實的信任基礎。共識層的實現可以靈活選擇，既可以採用區塊鏈技術，也可以使用其他形式的共識機制，以滿足不同場景的需求。

3.2.2.2 服務層與服務提供者

服務層是系統的中間層，對應的參與者是各個服務提供者。每個服務提供者都提供特定的服務，並且可以通過符合 AID 系統標準的介面加入系統。服務提供者不直接儲存使用者資料，而是作為使用者聚集的節點，使用者能夠自由地利用 AID 系統使用所需的服務。這種設計既保護了使用者數據隱私，又提供了靈活的服務串接機制。

3.2.2.3 數據層與終端使用者

數據層是系統的應用層，直接面向終端使用者。在這一層中，每個終端使用者擁有自己的數據存儲空間，並且系統提供了一個統一的數據管理介面。使用者可以自由地管理自己的數據與身分，並且通過「數據憑證」與「自主憑證」機制

來獲得他人對個人身分或數據的信任。



3.2.3 共識層

共識層是自主身分系統的基礎建設，主要由無數個共識核心組成。共識核心內包含無數個憑證，不管是「數據憑證」還是「自主憑證」。當共識核心由區塊鏈實作時，會使用智能合約來實現憑證的寫入、讀取、更新和評價等功能，分別介紹：

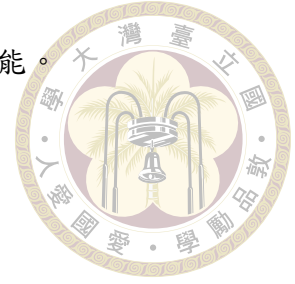
- **寫入**：用於合約初始化，將數據對應的雜湊與公開資訊寫入合約狀態。
- **讀取**：讀取合約狀態中的雜湊或公開資訊，用於驗證或查詢。
- **更新**：由憑證擁有者執行，用於即時變更憑證狀態，如撤銷或停用。
- **評價**：允許使用者根據智能合約規則，以特定格式讀取或寫入評價內容。

為了更好地理解共識層的運作，以學歷驗證為例。在這個場景中，使用者向學校申請學歷證明，學校將證明的數位雜湊寫入區塊鏈。任何需要驗證該學歷的服務都可以通過讀取區塊鏈上的雜湊來確認其真實性。若使用者的學歷狀態發生變化，學校可以即時更新區塊鏈上的雜湊狀態，確保驗證方獲得最新資訊。此外，如果某企業對學歷證明的可信度存疑，可以在區塊鏈上留下評價，供其他驗證方參考。

3.2.4 服務層

服務層是自主身分系統的應用層，負責提供各種服務。自主身分系統不包含具體服務的實作，只是提供名為 AID Server 的後端 SDK 規格，讓服務開發者能夠輕鬆地把自己的應用串接自主身分系統。本研究設計了 AID Server 的幾個關鍵功能：

- **身分管理**：提供服務加入、登入、登出等基本身分管理功能。
- **憑證管理**：提供憑證的創建、更新、讀取、評價等功能。
- **數據管理**：提供使用者數據的導入、導出等功能。



以下將分別介紹這幾個功能的設計細節。


3.2.4.1 身分管理

對服務提供者而言，新的自主身分加入必須先由使用者上傳「自主憑證」。此時，服務提供者可選擇是否接受該新自主身分。多數公開服務可能接受任何自主身分，但私人服務或有更高要求，如僅接受特定機構簽章的自主憑證，或基於評價機制被充分信任的自主身分。一旦服務提供者接受新的自主身分，該身分即可開始使用服務。

為了在確保安全的同時提供便利，自主身分系統提供了兩種登入方式：簡易登入和多因素驗證登入。簡易登入是指使用者僅需提供少量資訊即可登入，如使用者別名和密碼。多因素驗證登入則要求使用者提供更多資訊，如電子郵件驗證碼、手機簡訊驗證碼或私鑰等。服務提供者可根據自身需求選擇是否啟用簡易登入功能。

另外，儘管自主身分（AID）的生成是透過 UUID 機制 [29] 產生唯一識別號，但 AID 系統允許使用者在服務中設定自己偏好的別名，而非如傳統身分服務限制使用唯一的電子郵件地址作為使用者名稱。在日常操作中，系統優先讓使用者使用別名作為帳號完成簡易登入，只有當別名難以被識別時，才會要求使用者使用與 UUID 關聯的多因素驗證登入機制進行識別。本研究認為該設計更能表現出使用者對自己身分的自由控制權。

最後，服務提供者應當實作完善的登出功能，讓使用者能夠有效管理自身的



在線狀態。這一功能不僅允許使用者標示自己的當前狀態，更應包含請求服務端遺忘所有數據和終止會話（session）等選項。因為在自主身分中，使用者擁有自己的數據，服務提供者僅提供服務，因此每個使用週期結束後，服務提供者應當刪除所有與該使用者相關的數據。通過提供這種全面的登出機制，服務提供者能夠進一步強化使用者對其身分資訊的掌控，同時滿足使用者在資訊安全和隱私保護方面的需求。

3.2.4.2 憑證管理

服務層中的憑證管理包含兩個部分：分別是「自主憑證」和「數據憑證」。在「自主憑證」方面，大多數都是由使用者直接上傳，服務可以據此找到區塊鏈上使用者身分憑證的狀態與雜湊來完成身分驗證。此外，其他實體（如服務提供者或其他使用者）可基於自身經歷，在使用者憑證對應的鏈上合約中留下評價。

在「數據憑證」方面，當使用者申請共享特定數據時，服務提供者可按照預定的智能合約協議創建憑證，並將憑證發布到區塊鏈上。如果數據發生變動，服務提供者可即時更新憑證狀態，確保使用者數據的即時性和真實性。當另一個服務收到使用者在區塊鏈外提交的完整數據後，除了可以通過區塊鏈上的憑證驗證數據的真實性外，還可以在區塊鏈上留下評價，為其餘服務提供者提供參考。

3.2.4.3 數據管理

因為自主身分系統的核心在於賦予使用者控制自身數據的權利，數據管理在服務層相對簡單。理論上，服務提供者的最低要求是確保使用者能夠：

1. 將當次操作所必需的數據導入服務中
2. 在操作結束後將數據導回使用者端



然而，這種簡單設計可能大幅降低使用者體驗。例如：

- 若使用者每次操作都需要導入和導出數據，許多優化使用者體驗的功能將變得不可行，因系統無法追蹤使用者歷史數據。
- 頻繁的大量數據導入導出會顯著增加操作延遲和網路成本。

因此，「混合數據管理」模式成為必然選擇，即部分數據由使用者保管，部分由服務提供者保管。這種設計在保護使用者隱私的同時，也確保了操作的便利性。然而，「混合數據管理」模式仍面臨諸多需要服務提供者與使用者達成共識的問題，包括：

- 哪些數據勢必要導入服務端才能完成操作？
- 哪些數據可以持續保留在使用者端？
- 服務提供者管理的數據保留時間？
- 數據的使用權限？

這些問題涉及對隱私和資安的權衡，在下一節中將進一步討論。

3.2.5 數據層

數據層是自主身分系統的存儲層，負責存儲使用者個人數據。本研究設計了名為 Wallet 的前端 SDK 規格，讓數據層應用開發者能夠輕鬆地把自己的應用接入自主身分系統。Wallet 的幾個關鍵功能如下：

- **身管理**：提供自主身分的創建功能。
- **數據管理**：提供使用者數據的上傳、下載等功能。
- **憑證管理**：對共識雜湊的更新、讀取、評價等功能。
- **數據存儲**：儲存使用者的數據。

以下將分別介紹這幾個功能的設計細節。



3.2.5.1 身分管理

Wallet 的身分管理主要提供一個核心功能：創建自主身分。這裡的創建並非指使用者加入某個服務，而是包含兩個主要功能：

1. 使用者直接在本地裝置透過隨機產生的 UUID 創建自己的自主身分
2. 透過「自主憑證」機制創建新的憑證

在本研究的設計中，使用者可依據需求使用個人裝置創建不同的 AID，並為不同需求生成不同的身分憑證，再利用這些憑證證明 AID 的真實性。當使用者需要向服務證明自己是 AID 的真實持有者時，僅需透過 Wallet 完成 AID 上標示的多因素驗證方案。

3.2.5.2 數據管理

Wallet 的數據管理功能主要在服務需要時，將特定數據自主上傳至服務中，並在服務結束後自主下載回本地裝置。這裡的「自主」指使用者可自由選擇是否上傳或下載數據，並可自由選擇上傳或下載的數據內容。此的設計不僅更徹底地保護了使用者的安全與隱私，更因為由使用者自行攜帶數據在服務間移動，而徹底解決了數據孤島的問題。

3.2.5.3 憑證管理

Wallet 的憑證管理功能主要對共識層內憑證進行更新、讀取、評價等操作。基於「數據憑證」與「自主憑證」機制，使用者與服務需要在共識層中的區塊鏈上基於智慧合約留下 AID 雜湊，讓人們可在區塊鏈上驗證 AID 的真實性。此外，

所有使用者（包含服務提供者）都可在區塊鏈上對智能合約進行操作，以留下評價，藉此形成共識達成信任。



在 Wallet 中實作的憑證管理功能，使用者可輕易從 Wallet 中直接讀取共識層內的信任關係，並直接在共識層中更新自己身分憑證的狀態，以及對他人或服務的憑證評價。此設計不僅提高了使用者的自主權，更讓使用者能更直接地參與共識層的運作。

3.2.5.4 數據存儲

Wallet 的數據存儲功能主要用於儲存使用者的各類數據，包括憑證、公私鑰、個人資料以及與各服務的交互紀錄等。這種設計使 Wallet 成為使用者的個人化數據中心，實現統一管理。然而，將個人移動設備轉化為數據中心需解決備份、遷移和雲端儲存等問題。雖然本研究鼓勵使用者自主選擇解決方案，但仍提供以下建議實作：

每個採用自主身分系統的應用都應包含 Wallet 模塊，並利用設備的嵌入式資料庫存放數據。使用者可設置各 Wallet 的同步策略，包括自動同步（按需獲取數據）、完全同步（複製全部使用者數據）和手動同步（使用者指定數據複製）。這種設計便於實現備份、遷移和雲端儲存等功能，提升使用者數據管理的便利性。具體實現概念如下：

- **遷移**：新 Wallet 可通過手動輸入舊 Wallet 的公開地址或直接連接同一設備上的已啟動 Wallet 來建立連結。遷移後，新 Wallet 可設置對舊 Wallet 的同步策略。
- **雲端儲存**：考慮到在單一移動設備上存儲全部個人數據的安全風險，以及終端使用者難以維護家庭點對點（Point-to-Point,P2P）集群的現實，專業雲端

服務供應商可提供執行完整 Wallet 的服務。使用者支付費用後，可讓其他 Wallet 連接到此雲端 Wallet。

- **備份：**當使用者在多個設備上維護多個 Wallet，並重複存儲每項數據時，自然形成了數據備份機制。

這種多元化的數據管理策略不僅提高了數據安全性，也增強了系統的靈活性和使用者體驗。

3.3 系統設計細節

自主身分系統是個龐大的系統，涉及多個技術層次和參與者角色。為了更好地理解系統的運作，本節除了會深入介紹幾個關鍵技術模塊外，還會對特殊的系統流程進行詳細說明。

3.3.1 區塊鏈憑證機制

區塊鏈憑證機制是自主身分系統的重要組成部分，分成兩種憑證：「自主憑證」和「數據憑證」。能在確保隱私的同時，提供可靠的身分驗證和數據真實性驗證。以下會詳細介紹這兩種憑證的設計細節。

3.3.1.1 自主憑證

本研究參考 Tze-Nan[51] 提出的自主憑證概念，設計了基於區塊鏈技術的自主身管理流程。主要概念是由使用者自行生成憑證後根據個人需求自定義憑證的內容和權限，並且擁有隨時撤銷憑證的權力。另外，憑證內可以加入使用者自訂數據，以滿足不同場景的需求。還可以把憑證傳給簽章者進行實名認證，簽章

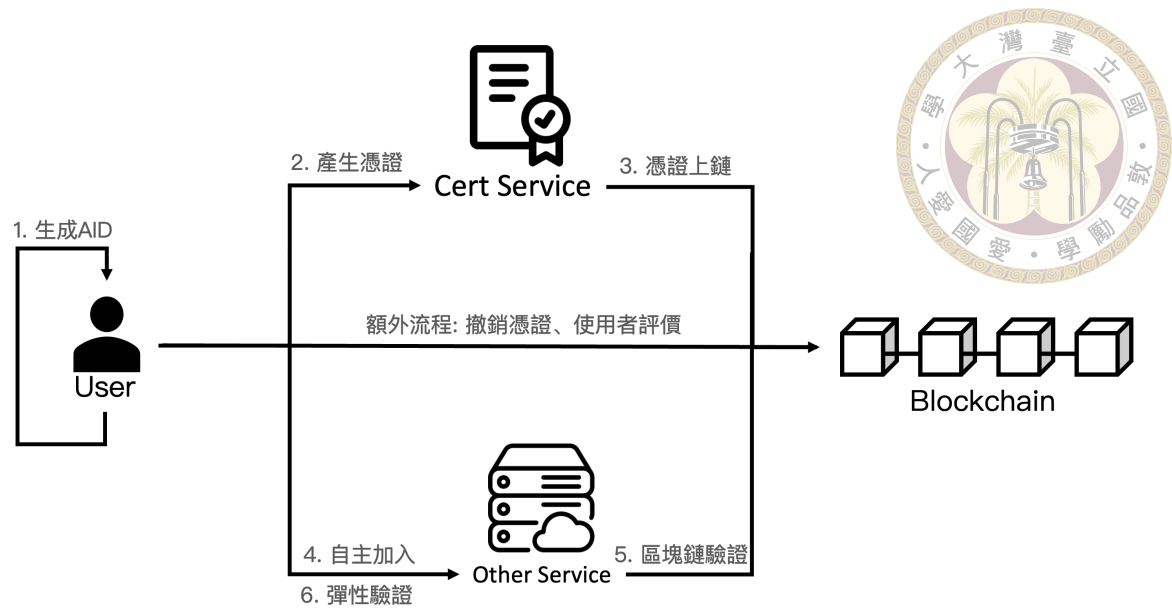


Figure 3.10: 自主憑證流程

者可以把簽名放入憑證中，以確保憑證的真實性。本研究提出的身分管理流程包含以下關鍵步驟如圖3.10：

1. **生成 AID**：透過 UUID[29] 機制離線產生唯一編號，使用者可以根據需求設置 AID 內含的詮釋資料 (metadata)，最後儲存在個人裝置中。
2. **產生憑證**：對於實名制憑證，使用者可以將 AID 傳給簽章者，簽章者將 AID 與自己的簽名一起放入憑證中，並將憑證傳回使用者。而對於非實名制憑證，使用者可以直接在本地裝置上生成憑證，不用放入任何簽名。
3. **憑證上鏈**：AID 的雜湊 (hash) 作為憑證，會被簽章者記錄在區塊鏈上。
4. **自主加入**：使用者在加入服務時，主動向服務提供者提交自己的 AID，表明自己的身分。
5. **區塊鏈驗證**：服務提供者可以在區塊鏈上取得憑證得知 AID 的真實性。
6. **彈性驗證**：使用者在登入服務時，可以根據 AID 自主選擇多因素驗證 (MFA) 的方式，來取得服務的信任。
7. **使用者評價**：區塊鏈上的憑證是 AID 的雜湊被放在智能合約中，相關參與者可以對憑證進行評價，從而形成該使用者的信譽。

8. **撤銷憑證**：使用者可以隨時撤銷憑證，並在區塊鏈上自行操作。這種機制可以應用於 AID 外流等情況。



此外，本系統的 AID 除了必須的唯一識別號，支持多樣化的 metadata，以下舉例說明：

- 自定義多因素驗證選項：滿足使用者登入的驗證所需。
- 選擇性資訊揭露：依照服務場景，使用者可以自行選擇要揭露的資訊。
- 設置憑證有效期限：使用者可以設定憑證的有效期限，以保護自己的隱私。
- 指定特定的驗證條件（如特定設備，地點或時間）：增強使用者對憑證的控制。
- 指定特定的驗證規則（設備和網路使用限制）：增強使用者對憑證的控制。
- 資源存取權限控制（如檔案存取權限）：增強使用者對憑證的控制。

總而言之，通過這種機制提升了系統對使用者隱私和安全的保護程度，讓使用者能夠更自主地管理其身分驗證流程，保持對個人身分驗證的控制權。

3.3.1.2 數據憑證

自主身分系統中，使用者在個人設備上自行管理數據的模式引發了數據共識問題。與傳統身分系統中服務可直接調用其他服務獲取數據不同，自主身分系統在處理跨服務數據共享需求時，需要一種機制來確保數據的一致性和可信度。為此，本研究擴展了自主憑證的概念，提出了數據憑證機制。數據憑證機制的運作方式如圖3.11：

1. **申請憑證**：當使用者需要向特定服務提交某個數據時，首先向能夠證明該數據真實性的其他服務提供者申請數據憑證。
2. **憑證上鏈**：接受申請的服務提供者在區塊鏈上提交該使用者數據的校驗雜

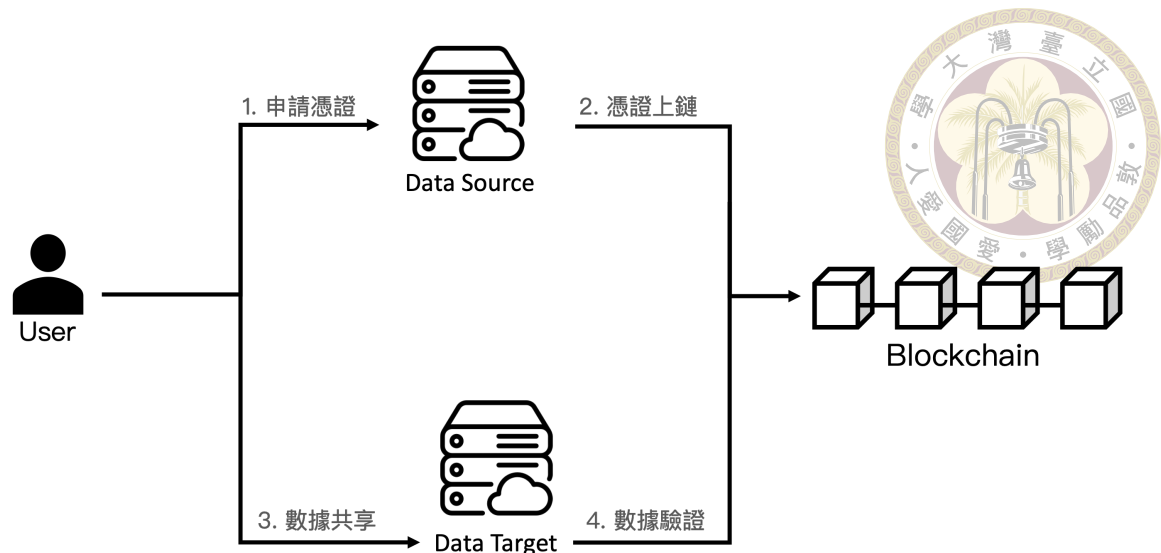


Figure 3.11: 數據憑證流程

湊。

3. **數據共享**: 使用者在向特定服務提交數據。

4. **數據驗證**: 接收數據的服務通過區塊鏈上的憑證完成數據的校驗，從而確認數據的真實性。

這種設計讓使用者能夠在不同的自主身分服務中安全地共享數據，同時向其他使用者保證數據的一致性和可信度。

此外，數據憑證的雜湊被放置在智能合約中，允許相關使用者對該筆數據進行評價。這一做法將道德標準的概念從身分層面擴展到數據層面，進一步提高了數據的可信度和安全性。

3.3.2 身分識別問題

以自治身分為例，為了確保身分系統的去中心化，大幅度使用了區塊鏈相關技術，卻因此導致使用者體驗大幅度偏離一般使用者的需求。本研究為了解決這個問題，在自主身分系統的共識層中定義了兩種登入方法：簡易登入和多因素驗證登入。期望可以根據不同情境，讓使用者選擇登入方式，以同時滿足安全性和

便利性的需求。

但是，系統該如何確保使用者能使用簡易登入方式呢？畢竟簡易登入可能因提供資訊不足而被攻擊者冒充或無法成功辨識身分。為此，本研究提出了「基於使用者時空的分析方法」，通過追蹤使用者每次操作時夾帶或產生的資訊來辨識使用者身分，並配合「基於危險程度的驗證機制」找出應要求多因素驗證的時機。

接著本研究會詳細介紹「基於使用者時空的分析方法」與「基於危險程度的驗證機制」這兩種特殊解決方案。

3.3.2.1 基於使用者時空的分析方法

每個身分本質上可視為一個隨時間變化的動態向量空間，其維度可謂無窮。每次對使用者的身分驗證都可視為取得特定時間僅含部分維度的向量。這方法的核心是讓使用者自主決定提供哪些維度，並在每次與系統互動時攜帶這些資訊。基於此概念，即讓使用者使用簡易登入，只要使用者自動攜帶裝置指紋、位置等資訊，系統即可通過比對這些資訊來進一步確認使用者身分。

1. 使用者 i 在時間 t 的身分向量：

$$\mathbf{i}(t) = (i_1(t), i_2(t), \dots, i_n(t), \dots), \quad n \rightarrow \infty \quad (3.1)$$

2. 使用者 i 在時間 k 登入時傳入的向量：

$$\mathbf{v}_k = (v_{k1}, v_{k2}, \dots, v_{km}), \quad m \ll n, \quad v_{kj} = i_j(t_k) \text{ 對某些 } j \quad (3.2)$$

3. 暫存的向量集合（假設 $\dim V = m$ ）：

$$\mathbf{v}_k = (v_{k1}, v_{k2}, \dots, v_{km}) \in V, \quad \dim V = m \ll n \quad (3.3)$$

4. 假設相似度函數為 s ：

$$s : \mathbb{R}^m \times \mathbb{R}^m \rightarrow [0, 1] \quad (3.4)$$

5. 身分推測方法：

$$\text{Identity} = \arg \max_{\mathbf{v}_i \in V} s(\mathbf{v}_{\text{new}}, \mathbf{v}_i) \quad (3.5)$$

舉例來說：使用者在某次登入時提供了自己的別名、性別、所在地區等資訊，而在下次登入時僅提供了性別、所在地區等資訊，系統可以通過比對這兩次向量來推測使用者的別名。這種方法使使用者在不提供完整資訊的情況下，仍能通過部分資訊完成身分驗證。

然而，這種方法也帶來了一些問題。例如，使用者提供的維度多寡會影響系統的準確性，甚至使用者提供的維度是否包含可變資訊會影響系統的安全性。因此，本研究提出維度的選擇甚至各個維度的權重應由使用者自主決定，而系統僅提供推薦機制，讓使用者在不同情境下使用不同方法，以滿足其需求。

3.3.2.2 基於危險程度的驗證機制

AID 系統允許使用者自定義各種行為的危險程度，並據此調整驗證的嚴謹度。然而，這種設計可能帶來兩個極端問題：過高的嚴謹度可能導致正常登入受阻，而過低則可能危及系統安全。為此本研究們提出「基於危險程度的驗證機制」希望系統性地協助使用者為不同場景設定合適的驗證嚴謹度。

這個機制建立在「基於使用者時空的分析方法」之上，綜合考慮三個關鍵因素：行為的危險程度、符合的維度數量，以及時間點的接近程度。這三者的互動決定了所需的驗證嚴謹程度。簡而言之，行為越危險，需要越嚴格的驗證；而越嚴格的驗證，則要求更多的維度符合和更接近的時間點。



值得注意的是，AID 系統中，使用者可以自主決定行為的危險程度，系統僅提供建議。這種設計能滿足不同情境下的需求。例如，對於個人銀行帳戶，使用者可能將所有行為視為高度危險，因此需要最嚴格的驗證。相反，對於社群媒體帳戶，閱讀文章可能被視為低風險行為，而發布內容則可能需要更嚴格的驗證。

本研究建議使用者將危險程度定義為對數據操作的敏感度：越敏感的數據越危險，越危險的數據則需要越嚴謹的驗證機制。通過這種方式，本研究期望為各種使用場景提供既靈活又安全的身分驗證解決方案。

3.3.3 密碼救援問題

密碼救援問題是自主身分系統中的一個重要問題。在傳統身分系統中，密碼救援通常是通過中心化的機制來實現的，其本質的概念就如同系統內存在著一位管理者，共同管理著使用者的身分。但這明顯違背了自主身分追求的價值觀，因此本研究提出了「極致多因素驗證」的概念。

允許使用者在創建自主憑證時設置多種驗證方案，包括難以遺失或忘記的方法，如生物特徵識別（指紋或臉部辨識）。如此，即便使用者忘記密碼，亦可通過這些驗證方式重新取回身分。進一步擴展此方案，本研究還可以將「極致多因素驗證」概念應用於「基於危險程度的驗證機制」，根據危險程度要求使用者使用多種方法完成多因素驗證，進一步提高安全性，讓使用者即使已經被攻擊者奪取身分，也能通過補充更多項驗證的方式來取回身分。

3.3.3.1 極致多因素驗證

傳統身分驗證系統中，多因素驗證主要被視為登入時的固定驗證方案。然而，在自主身分系統的設計中，為了實現使用者功能的自由管理，人們需要一種更靈

活、更強大的驗證機制，以取代傳統系統中系統管理員的角色。為此，本研究提出了「極致多因素驗證」的概念。



這一新概念的核心在於：不再將特定的多因素驗證方法視為使用者登入的唯一途徑。相反，從服務器的角度來看，每個驗證因素都應被視為使用者向服務器自主證明身分的一種方法。這種方法的靈活性體現在：


- 在高風險情況下，系統可能要求連續多種因素的驗證以確保安全。
- 在低風險環境中，可能僅需一種因素即可完成驗證。
- 即便使用者遺失了某個重要的驗證因素，系統也不應將其視為無法登入，而是要求提供更多其他因素的驗證來補償。

通過這種動態和適應性的方法，極致多因素驗證不僅提高了系統的安全性，還增強了使用者的身分管理自主權，為自主身分系統提供了一個更加靈活和強大的驗證框架。

3.3.4 混合數據管理

在自主身分的理想情境中，使用者可以完全控制自己的數據，並且能夠自由地在各個服務之間移動。然而，這種情況實際上只會在極端情況下出現，即完全靜態的網頁或應用程式。這類應用不需要與外部服務進行數據交流，使用者只需下載原始碼並連接本地數據庫即可。但在現實中，這種情況幾乎不可能發生。因此，本研究提出了「混合數據管理」的概念，旨在提供一套完整的建議，讓使用者和服務提供者能夠以最簡單的流程實現最強大的安全性與隱私保護。

這種模式的核心在於：每個應用程式都應以靜態應用為基礎啟動，所有數據交互都通過 Wallet 模組進行，以確保統一的設計。隨後，與服務層的互動應遵循以下建議來放寬權限：

- 
1. 服務層在索取數據時應提供清晰的說明，讓使用者了解數據上傳的必要性。
 2. 採用逐步詢問的方式，協助使用者設定所有簡化的資安措施或降低的隱私保護措施。
 3. 設置應從高到低、從嚴格到寬鬆平滑過渡，讓使用者能根據自身需求調整安全性和隱私保護程度。

此外，每次開啟權限產生數據上傳時，都可以通過「數據憑證」機制來確保數據的明確授權等問題。儘管採用這種方案仍可能因資訊上傳而產生隱私與資安問題，但從使用者流程的角度來看，它在盡可能保護使用者權益的同時，為服務提供者提供必要的數據支持，實現了雙方利益的平衡。通過這種「混合數據管理」模式，本研究能夠在自主身分系統中實現數據的高效管理，同時保障使用者的數據主權。

3.3.5 組織使用者控管

組織和個人使用者在身分控管需求上存在顯著差異，這給以使用者為中心的身分系統帶來了挑戰。傳統的身分供應商雖然為組織提供了專門解決方案，但這些方案往往會影響使用者自主性或需要額外創建身分，導致使用者反感和抵制。

AID（自主身分）系統通過「自主憑證」機制提供了一個創新方案。在這種機制下，組織作為憑證簽章者，可以在 AID 中明確寫入控管規則。儘管可能減少管理靈活性，但這種方法在保護使用者自主權的同時，也滿足了組織的控管需求。此外，這一機制有望推動去中心化組織的發展，為組織結構和管理模式的創新提供了新的可能性。



3.4 資料結構

本節會分別介紹自主身分系統中各角色重要物件的資料結構，以此讓讀者更清晰的了解系統的設計。

3.4.1 共識核心

共識層由不同的智能合約組成，每個智能合約都有自己的資料結構。這裡將介紹「自主憑證」和「數據憑證」機制所需智能合約的參考資料結構。

- **自主身分唯一編號**：透過 UUID 生成，用於識別使用者，此欄位在重視隱私的情況下可不填。
- **憑證操作公鑰**：對應私鑰被註記在明文憑證中，因此智能合約可以確保操作者真的有得到實際的憑證。
- **憑證雜湊**：擁有明文憑證的系統使用者可以藉此完成憑證真實性的驗證。
- **憑證狀態**：用於記錄憑證的當前狀態，可能包括「正常」、「失效」、「遺忘」等。
- **憑證評價**：用於記錄憑證的評價，可能包括「好」、「壞」、「中立」等，並且同時可以紀錄評價者的 AID。

3.4.2 AID Server

服務層中一般會暫存使用者的明文憑證與關聯服務商業邏輯的資料結構，這裡僅介紹明文憑證的參考資料結構。

- **自主身分唯一編號**：透過 UUID 生成，用於識別使用者。
- **憑證操作私鑰**：對應公鑰被註記在智能合約中，因此智能合約可以確保操作



者真的有得到實際的憑證。

- **簡易識別方案**：列出使用者想要使用的簡易登入方案與對應辨識內容，如「帳號密碼」、「IP 綁定」等。
- **多因素驗證方案**：列出使用者想要使用的多因素驗證方案與對應辨識內容，如「簡訊驗證」、「硬體金鑰」等。
- **個人資訊**：用於揭露使用者願意提供的個人資訊，如「姓名」、「性別」等。
- **服務設定**：用於記錄使用者希望自主設定的內容，如憑證的權限、有效期等。

拿簡易識別方案舉例，明文憑證就會包含「帳號」與「密碼」兩個欄位，而「IP 綁定」則會包含「IP 位址」欄位。多因素驗證方案的資料結構也類似，如「簡訊驗證」會包含「手機號碼」欄位。

另外，為了確保簡易登入的安全性，建議服務提供者暫存使用者每次識別空間與對應時間，以便透過「基於使用者時空的分析方法」來識別使用者。接著提供本研究建議的暫存資料結構。

- **使用者身分**：自主憑證明文。
- **使用者時間**：用於記錄使用者每次識別的時間。
- **使用者空間**：用於記錄使用者每次識別所拿到的所有數據，包含裝置指紋、IP 位址等。

3.4.3 Wallet

數據層中一般會存儲使用者的所有數據，這裡介紹 Wallet 中的重要資料結構。

- **自主身分列表**同一個 Wallet 可以管理多個 AID。



- 自主身分唯一編號：透過 UUID 生成，用於識別使用者。
- 憑證列表：用於記錄 AID 的所有憑證明文。
- 使用者個資：用於記錄 AID 的所有個人資訊。
- 使用者密鑰：用於記錄 AID 的所有公私鑰。
- 使用者數據：用於記錄 AID 在所有服務中的數據，以列表的形式存在，可以用服務的 AID 來索引。

3.5 本章總結

為了全面評估自主身分系統相較於傳統身分系統的優勢，本研究基於前文提出的評估標準製作了以下比較表（表3.3）。這個表格總結了本章討論的主要方面，還突出了系統和目前身分系統的差異。通過這個比較表，本研究可以看到自主身

比較項目	中心化身分	聯合身分	使用者中心身分	自治身分	自主身分
身分控制	組織控制	多組織共同控制	身分提供者控制	使用者控制	使用者完全控制
使用者體驗	簡單但缺乏靈活性	單點登錄	單點登錄	複雜的區塊鏈操作	無摩擦驗證
隱私保護	低	中等	較高	高	非常高
數據存儲	集中存儲	分散在多個組織	身分提供者與服務	區塊鏈與服務	使用者端設備
互操作性	低	中等	高	高	高
單點故障風險	高	中等	低	低	低
可擴展性	低	中等	高	高	高
法規遵循	困難	較困難	較容易	容易	容易
信任模型	中心化信任	聯盟信任	身分提供者信任	去中心化信任	自主互信
身分證明	中心化驗證	聯盟驗證	OAuth 等協議	區塊鏈驗證	自主憑證機制
數據共享	不共享	組織間共享	API 授權共享	公開數據共享	數據憑證機制

Table 3.3: 各代身分管理系統比較

分系統在多個方面的優勢。這些優勢使自主身分系統成為一種更加易用、平等和安全的分管理系統，有望在未來取代傳統身分系統，成為一種更加符合當今社

會需求的身分管理方案。





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）。系統的驗證對象包含兩個主要服務，一是遵循微軟 AI 聊天系統規格 [34] 的典型 AI 聊天服務；二是一個假想的第三方支付服務。使用者可以使用自主身分，在支付費用的前提下使用 AI 聊天服務。本章將詳細介紹系統的架構和實現細節，並通過流程分析來進一步展現自主身分系統的應用價值。

4.1 系統架構

整個自主身分系統的概念驗證分成多個模塊如圖4.1，以下按照分層架構來介紹：



Figure 4.1: AID 概念驗證架構簡圖

- **共識層**: 一條區塊鏈，用於存放「自主憑證」和「數據憑證」的智能合約。
- **服務層**: 實際與使用者產生互動的服務，包括：
 - 一個完成憑證所需的簽章服務，用於生成「自主憑證」。
 - 一個 AI 聊天服務，用於提供 AI 聊天功能。
 - 一個支付服務，用於提供 AI 聊天所需的支付功能。
- **使用者層**: 使用者的手機應用程式，一對一對應到服務層的服務。



4.2 實現細節

Table 4.1: 系統分層結構及介面

層次	實現的介面
共識層	<ul style="list-style-type: none"> - 初始化與獲取 AID 資訊 - 設置與獲取憑證評論
服務層	<ul style="list-style-type: none"> - 簽名憑證並上鏈 - 驗證憑證與管理使用者數據 - 點數驗證與證明生成 - 登出時數據處理 - 身分驗證與登入 - 生成與回傳交易憑證
使用者層	<ul style="list-style-type: none"> - GUI 開發 (Flutter) - 本地數據存儲 (Hive) - 跨應用數據共享

如4.1所示，本研究實現了系統的各個層次，並提供了相應的介面。

在共識層中，本研究挑選了 OurChain[37] 作為區塊鏈的實作，並且在其上實現了智能合約。智能合約的介面如下：

- **initAID**: 設置合約的基本資訊，本實作包含擁有者 AID 和憑證簽名。
- **getAIDInfo**: 獲取合約的擁有者 AID 和憑證簽名。
- **setComment**: 設置針對憑證的評論，本實作提供純文字紀錄。
- **getComment**: 獲取針對憑證的評論。

服務層中，本研究首先利用 Golang 調用 OurChain 的 RPC 接口來實現智能合約的操作，完成了簽章服務。簽章服務的介面如下：



- **signAID:** 使用者傳入透過「自主憑證」生成的明文憑證，服務端將其簽名後上鏈，並返回簽名後的憑證與合約地址。

雖然 AI 聊天服務和支付服務的具體實現與本研究無直接關係，但本研究主要探討如何在這些服務中嵌入自主身分系統。

在 AI 服務中，使用者必須使用自主身分登入，並擁有足夠的點數才能使用服務。因此本研究設計了以下機制：

- 驗證並暫存使用者的「自主憑證」，實現簡易登入（使用別名與 pin 碼驗證）。
- 暫存並管理使用者數據（點數、歷史紀錄）。
- 要求使用者上傳「數據憑證」以驗證點數，並生成新的點數證明。
- 登出時回傳使用者數據並清除服務端資料。

在支付系統中，同樣加入了自主身分系統。其包含以下機制：

- 基於「自主憑證」進行身分驗證和多因素驗證登入。
- 支付後生成「數據憑證」作為交易憑證。
- 不存儲使用者資訊，僅回傳「數據憑證」。

最後，關於數據層的實現，採取了以下措施：

- 基於 Flutter 框架開發跨平台網絡前端應用，為每個服務提供直觀且功能豐富的圖形使用者界面（GUI）
- 使用 Google 開發的 Hive 套件，以嵌入式資料庫的形式將所有使用者數據存儲在使用者的手機上，確保數據安全
- 利用作業系統的剪貼簿共享功能，實現使用者在不同前端應用中輕易共享數

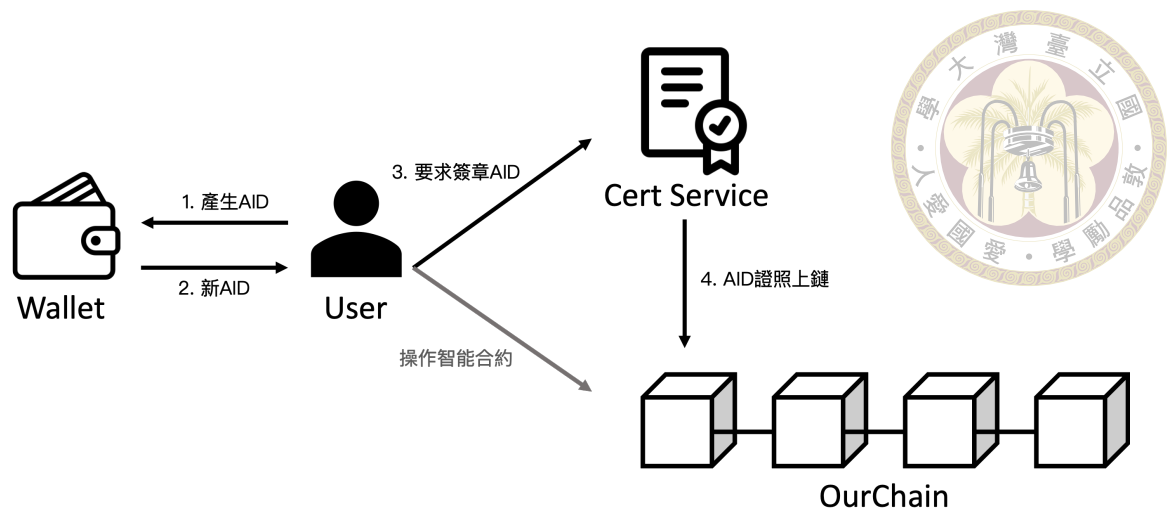


Figure 4.2: 產生新的 AID 與自主憑證

據

雖然這並不是一個完整的實作，但本研究已經成功證明了自主身分系統的可行性，接下來本研究將通過流程分析來展示自主身分系統的應用價值。

4.3 流程分析

本研究將找出三個典型的用例，配合簡圖與細節流程來展示自主身分系統的應用價值。這三個用例分別是：

- 產生新的 AID 與自主憑證
- 進入支付服務獲取收據
- 使用 AI 服務對話

4.3.1 產生新的 AID 與自主憑證

如圖4.2，以下是產生新的 AID 與自主憑證的流程：

1. 使用者在任意支援自主身分的前端應用中，點擊「產生新的 AID」

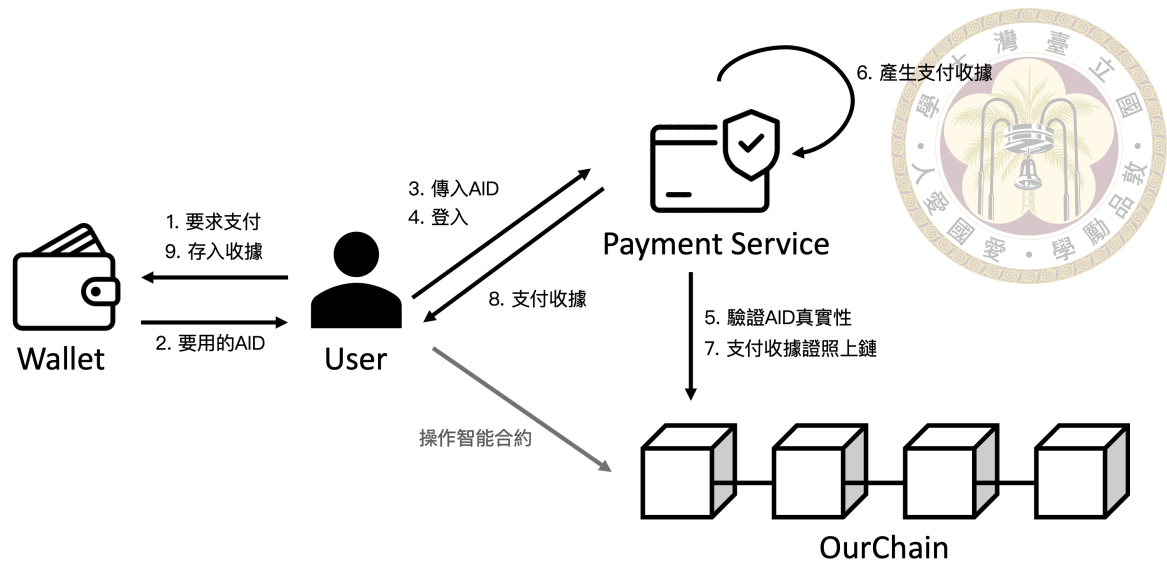


Figure 4.3: 進入支付服務獲取收據

- (a) 在裝置內部基於 UUID 機制產生唯一識別號後存儲
2. 使用者在簽章服務的前端介面中，選擇 AID 並且點擊「簽章」，產生表單讓使用者填寫資料
3. 使用者填寫個人資訊，包含別名與 pin 碼，並且點擊「確認」
 - (a) 透過使用者填入的資訊與 AID 中的識別號，生成未簽名「自主憑證」
 - (b) 把自主憑證傳入服務層產生簽名並上鏈
 - (c) 服務端回傳簽名後的「自主憑證」與合約地址，使用者存儲在個人設備

遵循以上流程，使用者自主的生成了自己的 AID 和自主憑證，整個使用者識別不依賴單一服務商，而是由使用者自己控制。

4.3.2 進入支付服務獲取收據

如圖4.3，以下是進入支付服務獲取收據的流程：

1. 使用者在支付服務的前端介面中，選擇 AID 並且點擊「支付」
 - (a) 自動取用設備上存放的「自主憑證」，上傳至服務端，觸發多因素驗證登入



Figure 4.4: 產生新的 AID 與自主憑證

- (b) 數據層的前端介面自動取用設備上的私鑰產生簽章完成登入
 - (c) 服務端驗證簽章後，完成支付，於是生成「數據憑證」並上鏈
 - (d) 服務端回傳「數據憑證」給使用者存儲在個人設備
2. 使用者在支付服務的前端介面中，點擊「查看收據」

遵循以上流程，使用者可以在支付後獲取到一份「數據憑證」，作為交易的憑證，使用者可以在任何時候查看這份憑證，而不需要依賴支付服務商。

4.3.3 使用 AI 服務對話

如圖4.4，以下是使用 AI 服務對話的流程：

- 1. 登入:
 - (a) 使用者進入 AI 服務的前端介面
 - (b) 傳入選定 AID 的「自主憑證」，觸發簡易登入
 - (c) 使用者使用憑證上的別名與密碼登入
 - (d) 登入後即開始 AI 服務中的一次對話
- 2. 對話前:



- (a) AI 服務接受數據層中上傳的使用者點數與歷史數據
 - (b) 要求使用者上傳點數的「數據憑證」，以確保點數真實性
3. 對話中:
- (a) 使用者與 AI 進行對話
 - (b) 每次對話結束後，AI 服務更新暫存的使用者點數與歷史數據
 - (c) 使用者即便離開對話，也只需使用別名與密碼簡易登入即可繼續，無需重新上傳數據
4. 登出:
- (a) 使用者選擇離開系統
 - (b) AI 服務將使用者點數、歷史數據與相關「數據憑證」回傳至個人裝置
 - (c) 系統刪除所有使用者數據

遵循以上流程，本研究實現了數據與服務分離的目標，讓使用者能夠更加方便地管理自己的數據。

4.4 本章總結

儘管本實現並非完整的商業系統，但已成功證明了自主身分系統的可行性和潛在價值。透過在 AI 聊天服務、支付系統等實際場景中的應用，展現了該系統在提升使用者隱私、資料安全和身分管理方面的巨大潛力。未來研究可進一步探討該系統在更廣泛領域的應用，以及在大規模商業環境中的實施策略。





第五章 結論與未來展望

本研究深入探討了自主身分（AID）系統，這是一個旨在解決現有身分管理系統挑戰的創新解決方案。通過將「自主」的概念融入數位身分管理，AID 系統致力於在保障動態道德標準的同時，賦予使用者對其身分信息和數據的完全控制權。研究過程中，本研究不僅提出了理論框架，還進行了系統設計和概念驗證，為未來的實際應用奠定了基礎。

本研究提出了自主身分的完整概念、設計了自主的身分管理機制、實現了高度的安全性和隱私保護、優化了使用者體驗、確保了法規遵循，並通過概念驗證展示了系統的可行性。

儘管 AID 系統展現出巨大潛力，但作為一個新興的研究領域，仍面臨諸多挑戰：

1. **艱鉅的推廣任務：** AID 系統可以說是一個發展中的身分管理系統，需要克服使用者習慣、技術標準、法律法規等多方面的障礙。未來需要進一步研究如何推廣 AID 系統，提高使用者接受度和市場競爭力。
2. **更高的開發難度：** AID 系統不但使用了區塊鏈等新技術，還引入了反轉數據層等新概念，這將對開發人員的技術水平提出更高的要求。未來需要進一步研究如何降低開發難度。
3. **大規模應用與部署：** 目前，AID 系統的概念驗證主要集中在 AI 聊天服務和

支付系統等有限的場景。未來需要進一步研究如何將 AID 系統應用於更廣泛的領域，並探討大規模部署的可行性和挑戰。



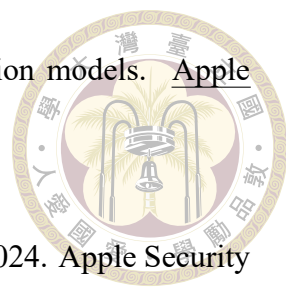
展望未來，AID 系統有望在多個領域帶來變革。在金融領域，它可以實現跨機構的無縫身分驗證，簡化開戶和貸款流程，並建立更透明、公平的個人信用評估系統。在醫療領域，AID 系統可以幫助建立患者完全掌控的電子病歷系統，實現跨機構就醫的同時保護患者隱私。在政務服務方面，基於 AID 的電子身分證系統可以實現一站式政務服務，而安全透明的電子投票系統則有助於提高公民參與度。

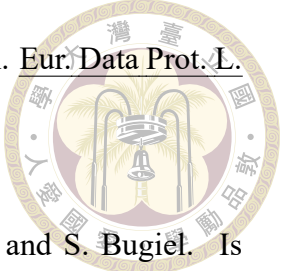
自主身分系統代表了數位身分管理的未來，有望成為下一個數位時代的代表性技術。儘管面臨挑戰，但通過持續研究，AID 系統將展現更廣闊的應用前景。未來研究將聚焦於大規模實際應用、使用者推廣、與現有系統的融合，以及系統性能和安全性的優化。我們期望更多人參與 AID 系統的探索和實踐，共同推動這一技術的發展，為數位社會的進步做出貢獻。




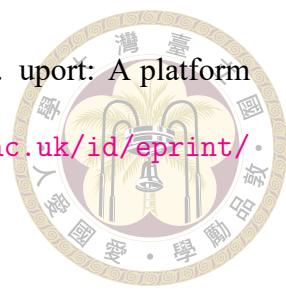
參考文獻


- [1] A. Acquisti, I. Adjerid, R. Balebako, L. Brandimarte, L. F. Cranor, S. Komanduri, P. G. Leon, N. Sadeh, F. Schaub, M. Sleeper, Y. Wang, and S. Wilson. Nudges for privacy and security: Understanding and assisting users' choices online. ACM Comput. Surv., 50(3), aug 2017. 隱私設置的研究.
- [2] G.-J. Ahn and M. Ko. User-centric privacy management for federated identity management. In 2007 International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2007), pages 187–195, 2007.
- [3] F. Alaca and P. C. van Oorschot. Device fingerprinting for augmenting web authentication: classification and analysis of methods. In Proceedings of the 32nd Annual Conference on Computer Security Applications, ACSAC '16, pages 289–301, New York, NY, USA, 2016. Association for Computing Machinery.
- [4] C. Allen. The path to self-sovereign identity, 4 2016.
- [5] A. A. S. AlQahtani, Z. El-Awadi, and M. Min. A survey on user authentication factors. In 2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), pages 0323–0328, 2021.

- 
- [6] Apple Inc. Introducing apple's on-device and server foundation models. Apple Machine Learning Research, 2024. Featured highlight.
- [7] Apple Inc. Private cloud compute. Blog post, Apple Security, 2024. Apple Security Blog.
- [8] M. S. Blumenthal and D. D. Clark. Rethinking the design of the internet: the end-to-end arguments vs. the brave new world. ACM Trans. Internet Technol., 1(1), aug 2001.
- [9] J. Bonneau, C. Herley, P. C. v. Oorschot, and F. Stajano. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In 2012 IEEE Symposium on Security and Privacy, pages 553–567, 2012.
- [10] K. Cameron. The laws of identity. White paper, Microsoft Corporation, 2005.
- [11] L. Campbell. Kant, autonomy and bioethics. Ethics, Medicine and Public Health, 3(3):381–392, 2017.
- [12] U. W. Chohan. Decentralized autonomous organizations (daos): Their present and future. March 2024. Available at SSRN: <https://ssrn.com/abstract=3082055>.
- [13] D. D. Clark, J. Wroclawski, K. R. Sollins, and R. Braden. Tussle in cyberspace: defining tomorrow's internet. SIGCOMM Comput. Commun. Rev., 32(4), aug 2002.
- [14] R. Dhamija and L. Dusseault. The seven flaws of identity management: Usability and security challenges. IEEE Security and Privacy, 6(2):24–29, 2008.
- [15] European Parliament and Council of the European Union. General Data Protection Regulation (GDPR), 2016. Regulation (EU) 2016/679.

- 
- [16] M. Finck. Blockchains and data protection in the european union. Eur. Data Prot. L. Rev., 4:17, 2018.
- [17] S. Ghorbani Lyastani, M. Schilling, M. Neumayr, M. Backes, and S. Bugiel. Is fido2 the kingslayer of user authentication? a comparative usability study of fido2 passwordless authentication. In 2020 IEEE Symposium on Security and Privacy (SP), pages 268–285, 2020.
- [18] M. Goodner and A. Nadalin. Web services federation language (ws-federation) version 1.2. OASIS Standard, 2009.
- [19] Google Cloud. Best practices for planning your identity architecture. https://cloud.google.com/architecture/identity/best-practices-for-planning#combine_cloud_identity_and_g_suite_in_a_single_account, 2024. Accessed: 2024-07-08.
- [20] T. Hamme, V. Rimmer, D. Preuveneers, W. Joosen, M. A. Mustafa, A. Abidin, and E. Argones Rúa. Frictionless authentication systems: Emerging trends, research challenges and opportunities. 09 2017.
- [21] Hard. The oauth 2.0 authorization framework. IETF RFC 6749, 2012.
- [22] G. Hub. Cnil (france) - san-2019-001. [https://gdprhub.eu/index.php?title=CNIL_\(France\)_-_SAN-2019-001](https://gdprhub.eu/index.php?title=CNIL_(France)_-_SAN-2019-001), 2023. Accessed: Tuesday 6th August, 2024.
- [23] International Telecommunication Union. ITU-T Recommendation X.509: Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks. Recommendation X.509, ITU-T, October 2019. Last accessed: 2024-08-03.

- 
- [24] A. Jøsang, S. Marsh, and S. Pope. Exploring different types of trust propagation. In K. Stølen, W. H. Winsborough, F. Martinelli, and F. Massacci, editors, Trust Management, pages 179–192, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. 信任傳播.
- [25] A. Jøsang, M. A. Zomai, and S. Suriadi. Usability and privacy in identity management architectures. In Proceedings of the Fifth Australasian Symposium on ACSW Frontiers - Volume 68, ACSW '07, pages 143–152, AUS, 2007. Australian Computer Society, Inc.
- [26] I. Krstić. Personal data in the cloud is under siege. end-to-end encryption is our most powerful defense. Lawfare, 12 2023. In cooperation with Brookings.
- [27] M. Kubach, C. H. Schunck, R. Sellung, and H. Roßnagel. Self-sovereign and decentralized identity as the future of identity management? In H. Roßnagel, C. H. Schunck, S. Mödersheim, and D. Hühnlein, editors, Open Identity Summit 2020, Lecture Notes in Informatics (LNI), pages 35–47, Bonn, 2020. Gesellschaft für Informatik. DID UX.
- [28] LastPass. Psychology of passwords: The online behavior that’s putting you at risk, 2020.
- [29] P. Leach, M. Mealling, and R. Salz. A universally unique identifier (uuid) urn namespace. 01 2005.
- [30] Y. Lin. 自主式社群網路身分的設計與實作 (the design and implementation of autonomous identity for social network). Master’s thesis, National Taiwan University, Taipei, Taiwan, 2014. Advisor: Chih-Wen Hsueh.

- 
- [31] C. Lundkvist, R. Heck, J. Torstensson, Z. Mitton, and M. Sena. uport: A platform for self-sovereign identity. https://publications.aston.ac.uk/id/eprint/42147/1/uPort_SSI_DrNitinNaik.pdf, 2017.
- [32] Microsoft. Ion - we have liftoff! Microsoft Azure Blog, 2020.
- [33] Microsoft. Active directory domain services overview. <https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview>, 2021.
- [34] Microsoft. Ai chat protocol. <https://github.com/microsoft/ai-chat-protocol/tree/main/spec>, 2024. Accessed: 2024-07-17.
- [35] National Institute of Standards and Technology. Digital identity guidelines. Special Publication 800-63-3, National Institute of Standards and Technology, Gaithersburg, MD, June 2017.
- [36] P. Nikander, A. Gurtov, and T. R. Henderson. Host identity protocol (hip): Connectivity, mobility, multi-homing, security, and privacy over ipv4 and ipv6 networks. IEEE Communications Surveys & Tutorials, 12(2):186–204, 2010.
- [37] NTU CSIE Lab408. Ourchain. <https://github.com/OurLab408/OurChain>, 2024. Accessed: 2024-07-17.
- [38] OASIS. Security assertion markup language (saml) v2.0 technical overview. Technical report, OASIS Committee Draft, 2005.
- [39] A. Preukschat and D. Reed. Self-Sovereign Identity: Decentralized Digital Identity and Verifiable Credentials. Manning Publications, May 2021. Available translations: Korean, Simplified Chinese.

- 
- [40] M. Saemann, D. Theis, T. Urban, and M. Degeling. Investigating gdpr fines in the light of data flows. Proceedings on Privacy Enhancing Technologies, 2022.
- [41] N. Sakimura. Openid connect core 1.0. OpenID Foundation.
- [42] F. Schardong and R. Custódio. Self-sovereign identity: A systematic review, mapping and taxonomy. Sensors, 22(15), 2022.
- [43] J. Sermersheim. Lightweight directory access protocol (ldap): The protocol. IETF RFC 4511, 2006. 輕量化 AD.
- [44] S. . Simmons. Amazon faces record gdpr fine. <https://www.simmons-simmons.com/en/publications/ckrus16301do70a28ptvwqy5t/amazon-faces-record-gdpr-fine>, 2021. Accessed: Tuesday 6th August, 2024.
- [45] Y. Smirnova and V. Travieso-Morales. Understanding challenges of gdpr implementation in business enterprises: a systematic literature review. International Journal of Law and Management, 66:326–344, 01 2024.
- [46] R. Soltani, U. T. Nguyen, and A. An. A survey of self-sovereign identity ecosystem. Security and Communication Networks, 2021(1):8873429, 2021.
- [47] S.-T. Sun and K. Beznosov. The devil is in the (implementation) details: an empirical analysis of oauth sso systems. In Proceedings of the 2012 ACM Conference on Computer and Communications Security, CCS '12, pages 378–390, New York, NY, USA, 2012. Association for Computing Machinery. OAuth 2.0 實作之漏洞, 建議更改.
- [48] K. Wang, Q. Xu, and G. Zhang. A secure threshold signature scheme from lattices.

In 2013 Ninth International Conference on Computational Intelligence and Security, pages 469–473, 2013.



- [49] S. Wiefling, M. Dürmuth, and L. Lo Iacono. What’s in Score for Website Users: A Data-Driven Long-Term Study on Risk-Based Authentication Characteristics. In 25th International Conference on Financial Cryptography and Data Security, FC ’21, pages 361–381. Springer, Mar. 2021.
- [50] T. Wu. Network neutrality, broadband discrimination. Journal of Telecommunications and High Technology Law, 2:141, 2003.
- [51] T.-N. Wu. The design and implementation of general autonomous certification on blockchain. Master’s thesis, National Taiwan University, Taiwan, 2021. Department of Computer Science and Information Engineering.
- [52] Y. Zhang, F. Monrose, and M. K. Reiter. The security of modern password expiration: an algorithmic framework and empirical analysis. In Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS ’10, pages 176–186, New York, NY, USA, 2010. Association for Computing Machinery.





附錄 A — 實際操作介面

A.1 AID 錢包

A.2 AI 聊天軟體





附錄 B — 系統 UML 圖

B.1 自主憑證流程

B.2 數據憑證流程