KEYCLOAK

AI時代身份治理平台從SSO、Zero Trust到AI Agent Security的下一代架構

資深技術顧問
Andy Chung
tichung@htic.com.tw

August 2026

鋐迪資訊
https://www.htic.com.tw

介紹

鍾迪 Andy Chung

- 現職：
  - 鋐迪資訊股份有限公司-資深技術顧問
- 學歷：
  - 加州長堤大學 CSULB 碩士
- 專案建置經歷 (23年)：
  - 政府、電信、航空、學校、海外：60個專案+
- 專業能力：
  - 身份管理系統規劃與建置
  - 帳號整合同步規劃與建置
  - 單一簽入系統規劃與建置
  - 特權管理系統規劃與建置
  - 多因素認證系統規劃與建置
  - Kubernetes系統規劃與建置
  - Java / SQL/ LDAP/ Scripting 開發

台大網路與資安技術研討會:

2022 - 運用零信任策略落實身份治理與單一簽入整合。

2023 - 透過單一簽入解決方案整合地端應用系統與雲端服務認證。

2024 - 深入探討特權帳號管理系統整合運用。

2025 - Kubernetes 私有雲實作指南：企業 IT 現代化的第一步。

2026 - AI時代身份治理平台從SSO、Zero Trust到AI Agent Security的下一代架構

課程大綱

1. AI 時代為何重新定義身份治理
2. Keycloak 核心能力與架構解析
3. Demo Showcase
4. 政府與校園導入情境
5. 現代 IAM 與 Zero Trust 架構演進
6. 身份治理的未來趨勢
7. 總結

AI 時代
為何重新定義身份治理

AI是什麼

AI 是一個能理解、思考、決策，並且可以替人執行工作的數位工作者

傳統程式

人
↓
點選按鈕
↓
系統執行

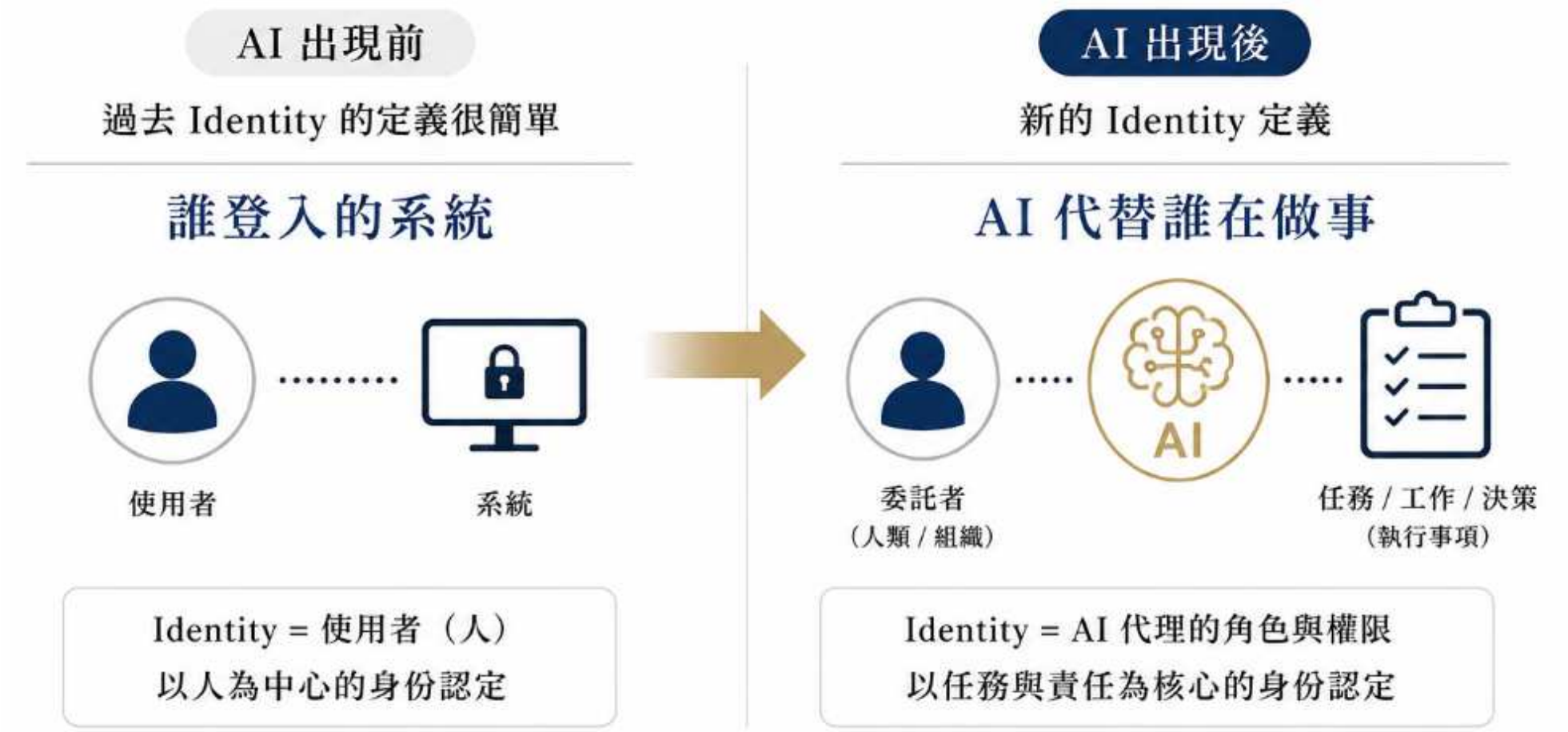
每一步都是工程師預先設計好的

AI

人
↓
自然話
↓
AI 理解
↓
AI 思考
↓
AI 自己決定流程
↓
AI 執行工作

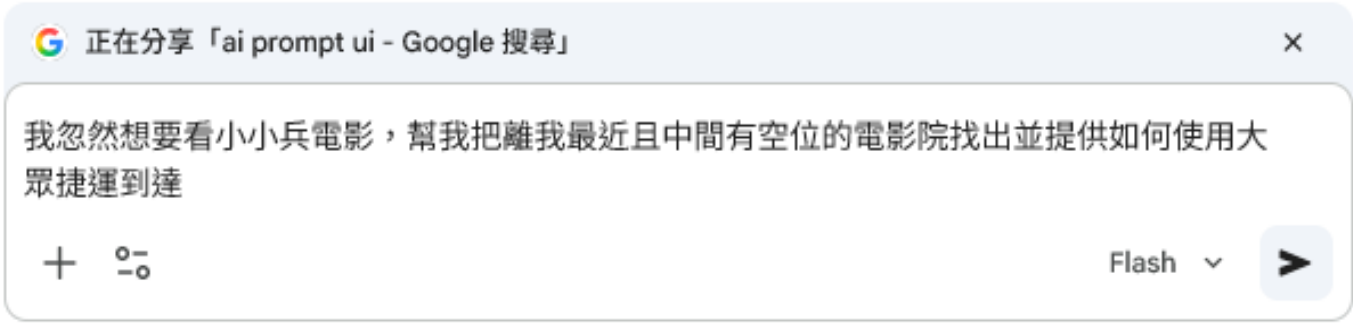
可能每次都會給有微差異的結果

AI 為何重新定義了 Identity



從「誰登入」到「AI 代替誰」，
Identity 的核心，從「人」轉向「責任與行為」。

有無AI比較



沒有AI時

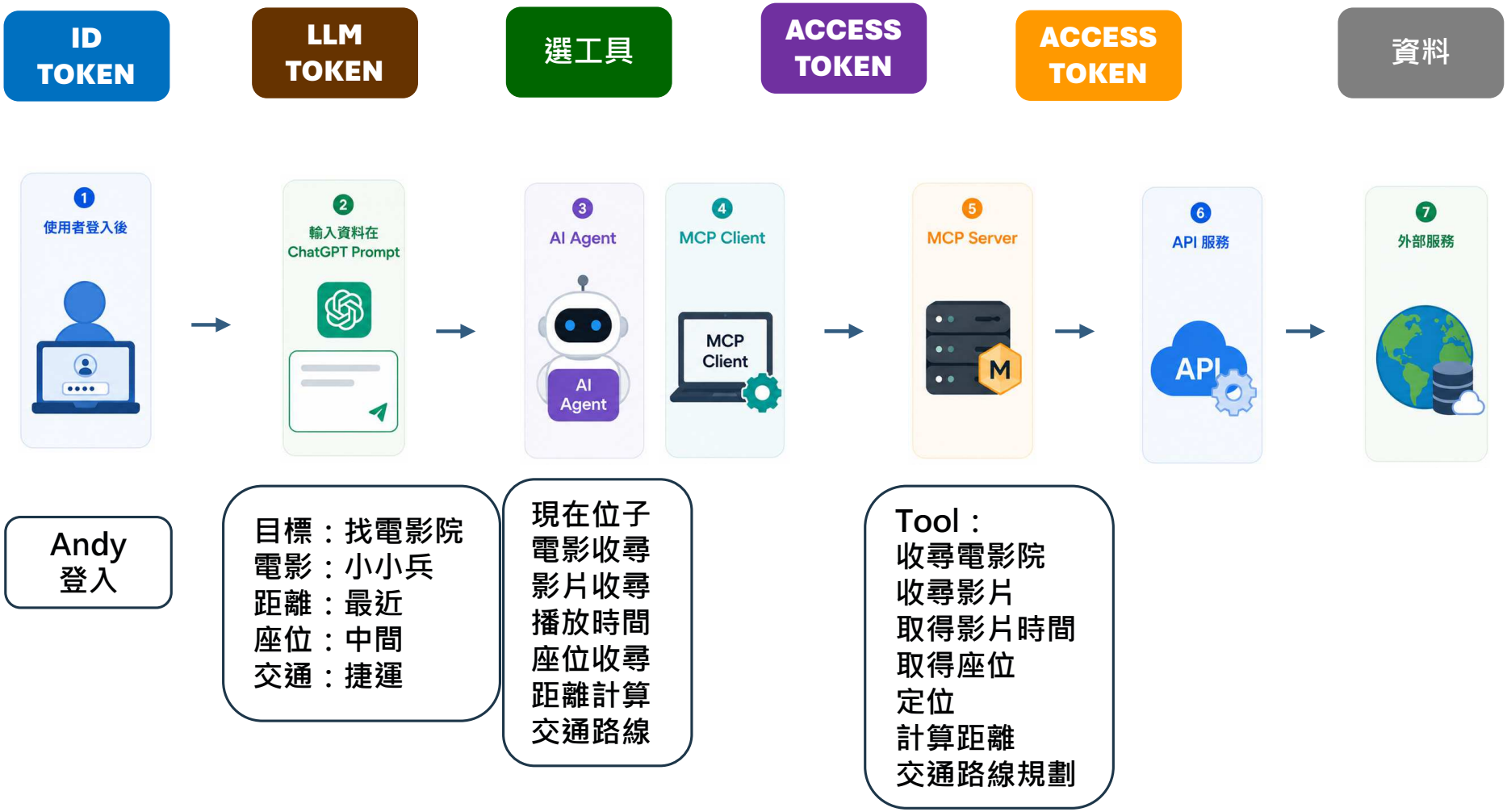
自己

1. 找出有播放小小兵的電影院
2. 確認電影院地址
3. 確認是否有中間的位子
4. google map電影院 by 捷運

有AI時

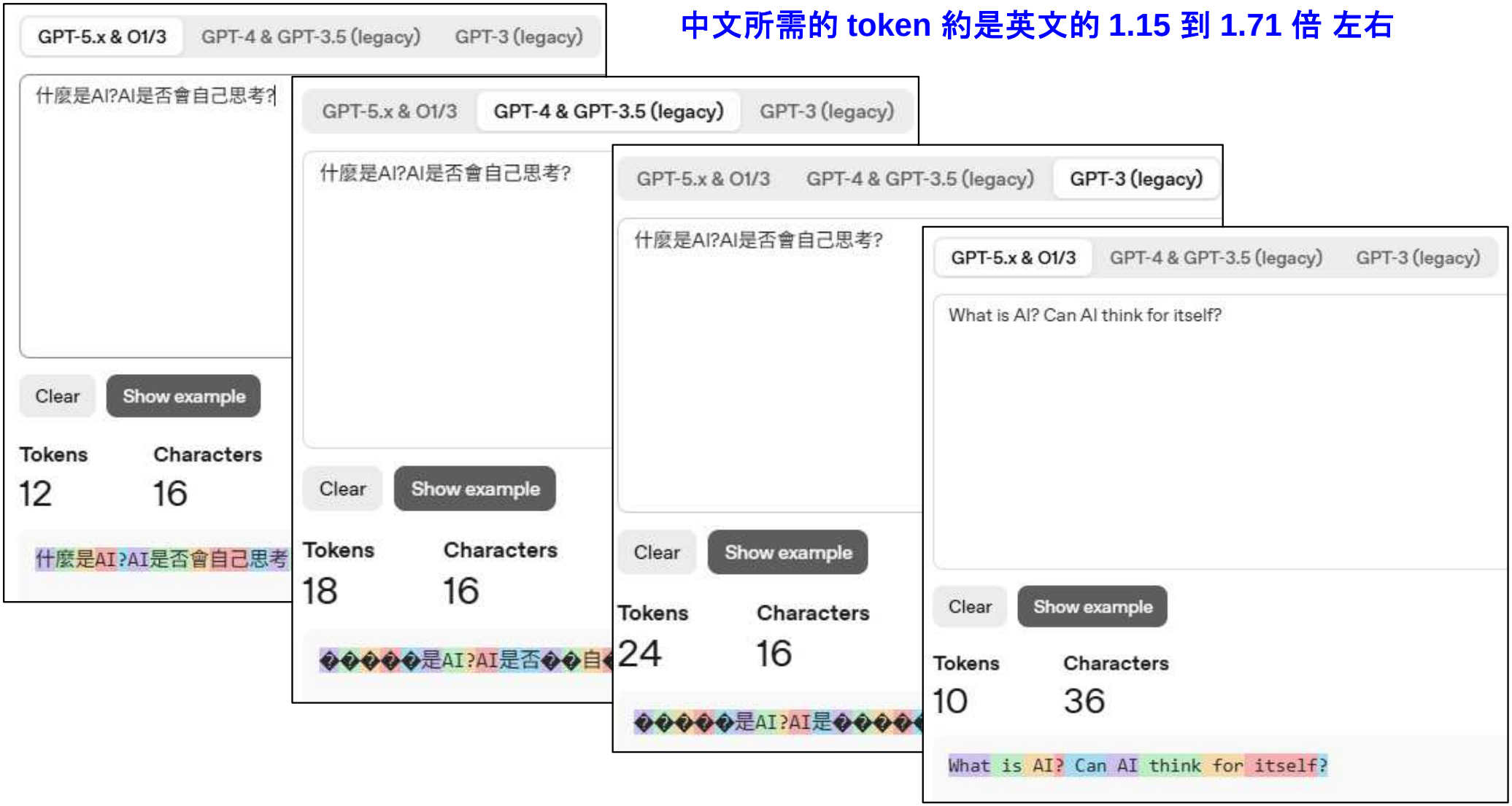
AI

AI非常依賴TOKEN機制



LLM Token

中文所需的 token 約是英文的 1.15 到 1.71 倍 左右



ChatGPT Tokenizer

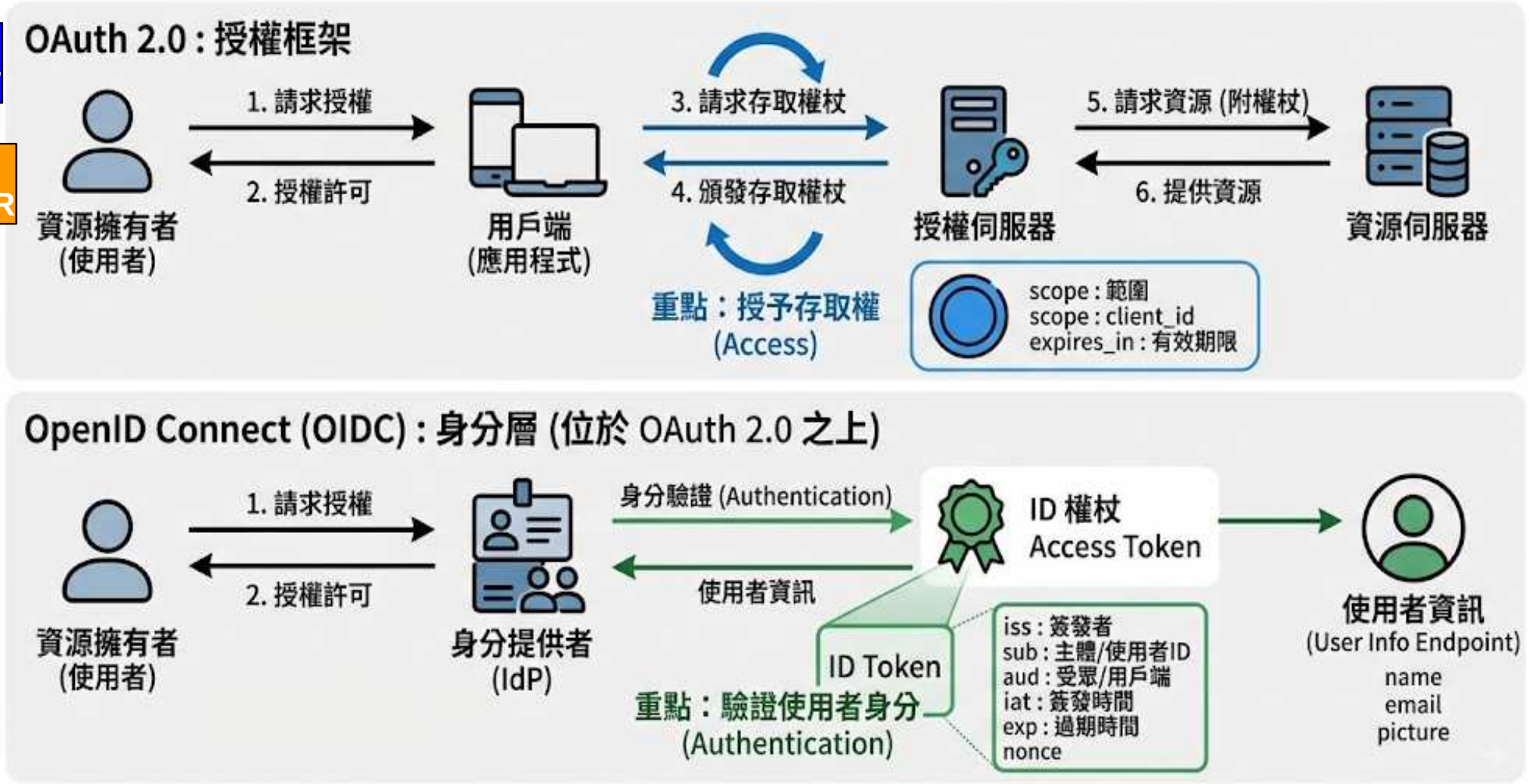
認證與授權的Token

AI AGENT

MCP SERVER

API

使用者登入



Token是AI的數位身份證

Token 是 AI 的「數位身份證」，沒有 Token，AI 無法安全地證明自己是誰、代表誰，以及被允許做什麼


```
{
  "header": {
    "alg": "RS256",
    "typ": "JWT",
    "kid": "bZGn9I9gCukorhKVKy2lPBWcy662-Ys7RRvdaLOio-Q"
  },
  "payload": {
    "exp": 1786411908,
    "iat": 1786411608,
    "auth_time": 1786411607,
    "jti": "onrtac:40ea2cc8-1215-4888-e451-26e8ed5d6c2e",
    "iss": "https://keycloak.apps.ocp.iam-guru.com/realms/iam-guru",
    "aud": "account",
    "sub": "40c7d4a1-8b41-428b-8d23-384d66cf9d8d",
    "typ": "Bearer",
    "azp": "auth-client",
    "sid": "w_xTRXvpS9YqmY8V8MBkv_zd",
    "acr": "1",
    "allowed-origins": [
      "http://localhost:8080"
    ],
    "realm_access": {
      "roles": [
        "offline_access",
        "uma_authorization",
        "default-roles-iam-guru"
      ]
    },
    "resource_access": {
      "account": {
        "roles": [
          "manage-account",
          "manage-account-links",
```


Access Token


```
{
  "header": {
    "alg": "RS256",
    "typ": "JWT",
    "kid": "bZGn9I9gCukorhKVKy2lPBWcy662-Ys7RRvdaLOio-Q"
  },
  "payload": {
    "exp": 1786411908,
    "iat": 1786411608,
    "auth_time": 1786411607,
    "jti": "a4eb5498-4bbf-5194-5d1e-6d1c39d9c3d6",
    "iss": "https://keycloak.apps.ocp.iam-guru.com/realms/iam-guru",
    "aud": "auth-client",
    "sub": "40c7d4a1-8b41-428b-8d23-384d66cf9d8d",
    "typ": "ID",
    "azp": "auth-client",
    "nonce": "ae7IeOl_DA0g1sRjiPeZaHvq",
    "sid": "w_xTRXvpS9YqmY8V8MBkv_zd",
    "at_hash": "8ph-rpNyrwp5ej8v1GocgQ",
    "acr": "1",
    "email_verified": false,
    "name": "迪 鍾",
    "preferred_username": "andychung",
    "given_name": "迪",
    "family_name": "鍾",
    "email": "andychung@iam-guru.com"
  },
```


ID Token

Keycloak
核心能力與架構解析

什麼是 Keycloak?

一套開源的身分與存取管理（IAM）平台

核心功能

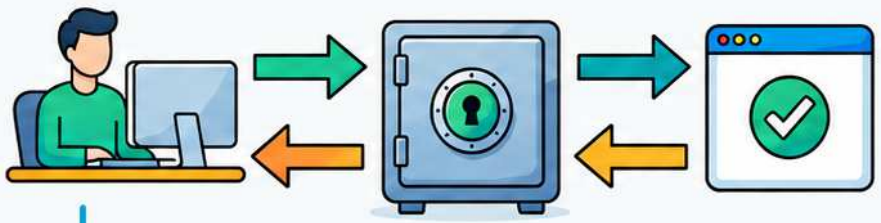
單一登入
(Single Sign-On, SSO)
使用者只需登入一次，即可存取所有應用程式。

身分識別中介
(Identity Brokering)
支援透過第三方身分提供者進行登入。

集中化使用者管理
(Centralized User Management)
在同一處管理所有使用者帳戶和權限。

運作原理

1. 使用者請求存取應用程式
2. 應用程式重新導向至 Keycloak 進行驗證



3. 使用者輸入認證資訊（使用者名稱／密碼）
4. Keycloak 驗證成功並發送安全權杖 (Token) 給應用程式

TOKEN

優點

增強安全性
集中化管理，降低密碼洩露風險。

提升使用者體驗
減少登入次數，簡化存取流程。

降低開發成本
無需為每個應用程式重複開發身分驗證邏輯。

擴充性與相容性
支援 OpenID Connect、OAuth 2.0, SAML 2.0 等標準。

Keycloak 十大核心能力

01 Identity Management（身份管理）

- 使用者管理
- 群組管理
- 角色管理
- 組織管理
- 聯合身分管理

02 Authentication（身分驗證）

- 多種登入方式
- 身分驗證流程
- 多因子驗證（MFA）
- 自訂驗證機制

03 Authorization（授權）

- 角色與權限管理
- 資源與範圍管理
- 權限政策（Policy）
- 授權服務

04 Single Sign-On（單一登入）

- 單一登入（SSO）
- 單一登出（SLO）
- 工作階段管理

05 Token Service（Token 服務）

- Access Token
- ID Token
- Refresh Token
- Token Exchange
- Offline Token

06 Client Management（Client 管理）

- 客戶端管理
- Client Scope
- Protocol Mapper
- 認證設定
- Redirect URI 管理

07 Identity Federation（身份聯邦）

- 整合企業目錄
- 社群身分
- SAML／OIDC 整合

08 User Federation（使用者聯邦）

- 使用者資料不落地
- 即時同步外部目錄
- 支援多種 User Storage Provider

09 Identity Brokering（身份代理）

- 身分代理（Broker）
- 整合外部 IdP
- 社群帳號登入
- 多重身分來源整合

10 Admin API

- RESTful Admin API
- Java Admin Client
- 自動化管理
- Terraform／Operator 整合

Keycloak by the Numbers

KEYCLOAK

全球領先的開源身份與存取管理解決方案，受到全球社群與企業的信任與採用。

30,000+
GitHub 星標
GitHub 上最受歡迎的身份與存取管理開源專案之一

12,985
貢獻者
全球社群持續貢獻與創新

2,675
組織參與
來自企業與組織共同參與並貢獻 Keycloak 專案

10+
發展年數
超過十年的成熟、穩定與持續演進

100%
開源專案
採用 Apache 2.0 授權完全免費使用、延伸與分享

數百萬
使用者
在全球數百萬個身份中提供安全保護

CNCF 孵化專案
雲原生運算基金會孵化專案

健康評分 極佳
健康評分 81/100

活躍開發 持續精進
定期版本發布與長期技術支援

全球採用 廣泛應用
被企業、政府與雲端服務供應商廣泛採用

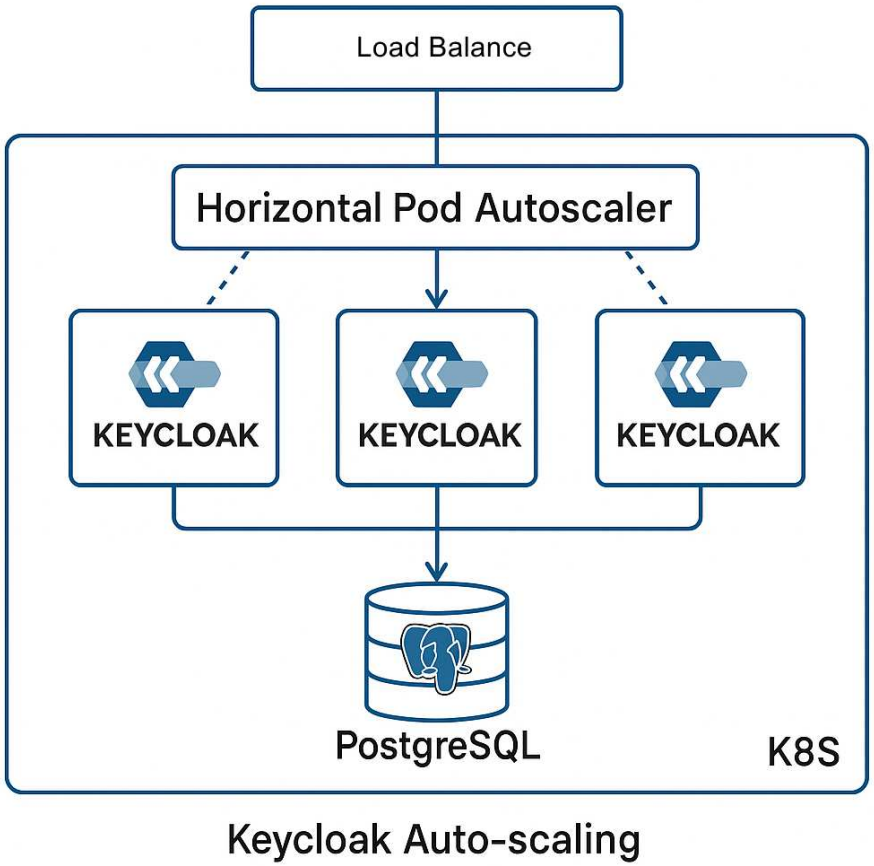
企業就緒 安全可靠
內建安全性、可擴充性與合規性設計

受到全球創新組織的信任

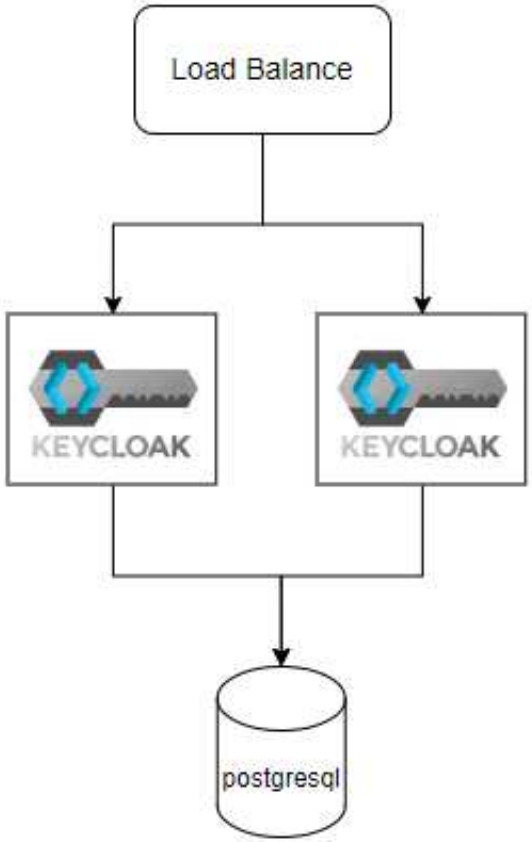
cisco HITACHI accenture CERN Red Hat Quest OHIO Ohio Supercomputer Center

Keycloak支援平台與協定

Keycloak架構



Keycloak Auto-scaling



Keycloak 可運行於：

- Linux
- Windows Server
- 容器化環境

資料庫平台：

Keycloak 可支援的資料庫：

- PostgreSQL（推薦使用）
- MySQL / MariaDB
- Oracle Database
- Microsoft SQL Server

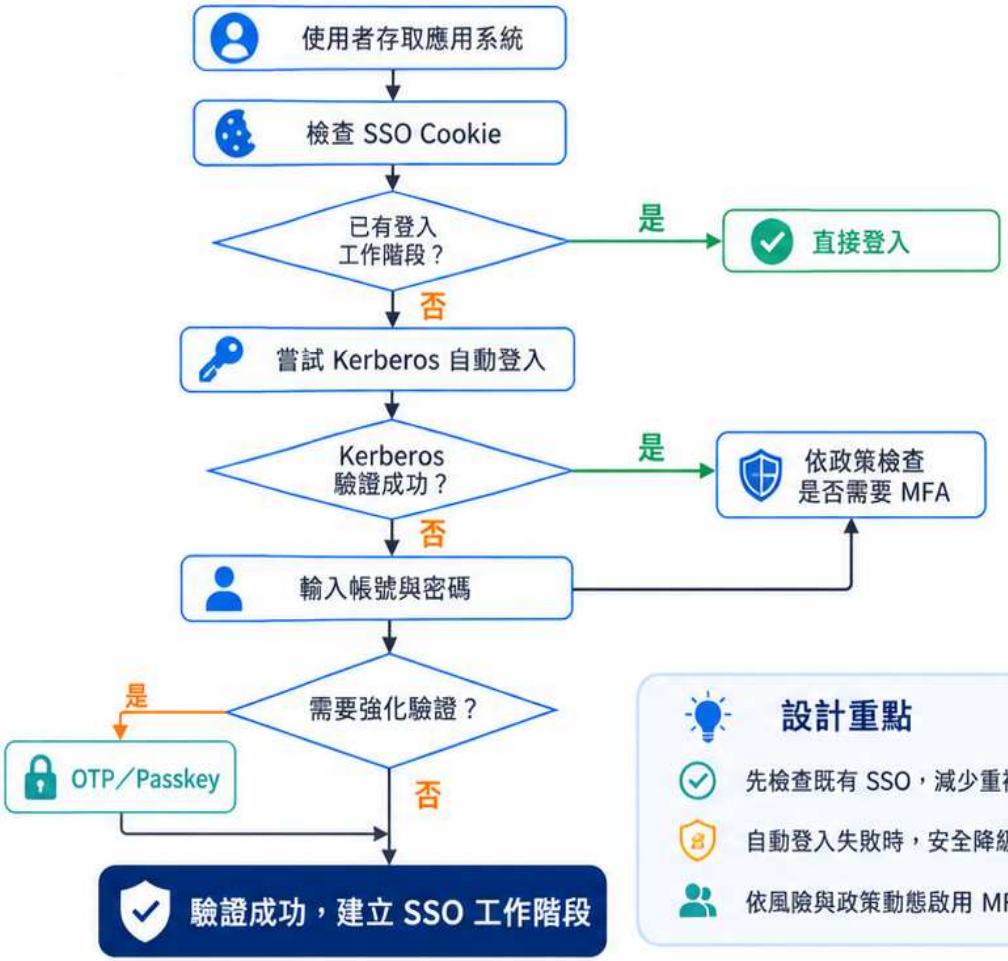
Keycloak 內建認證模組與認證流程

彈性組合多種驗證方式，打造符合企業需求的登入流程

內建認證模組

1. **帳號密碼**
   最基礎的第一因素驗證
2. **OTP 多因素驗證**
   支援 TOTP 與 HOTP
3. **Passkey／WebAuthn**
   支援無密碼與生物辨識
4. **Kerberos**
   整合 AD 實現自動登入
5. **X.509 憑證**
   使用用戶端憑證驗證
6. **外部身分提供者**
   整合 OIDC、SAML 與社群登入

企業認證流程範例



設計重點

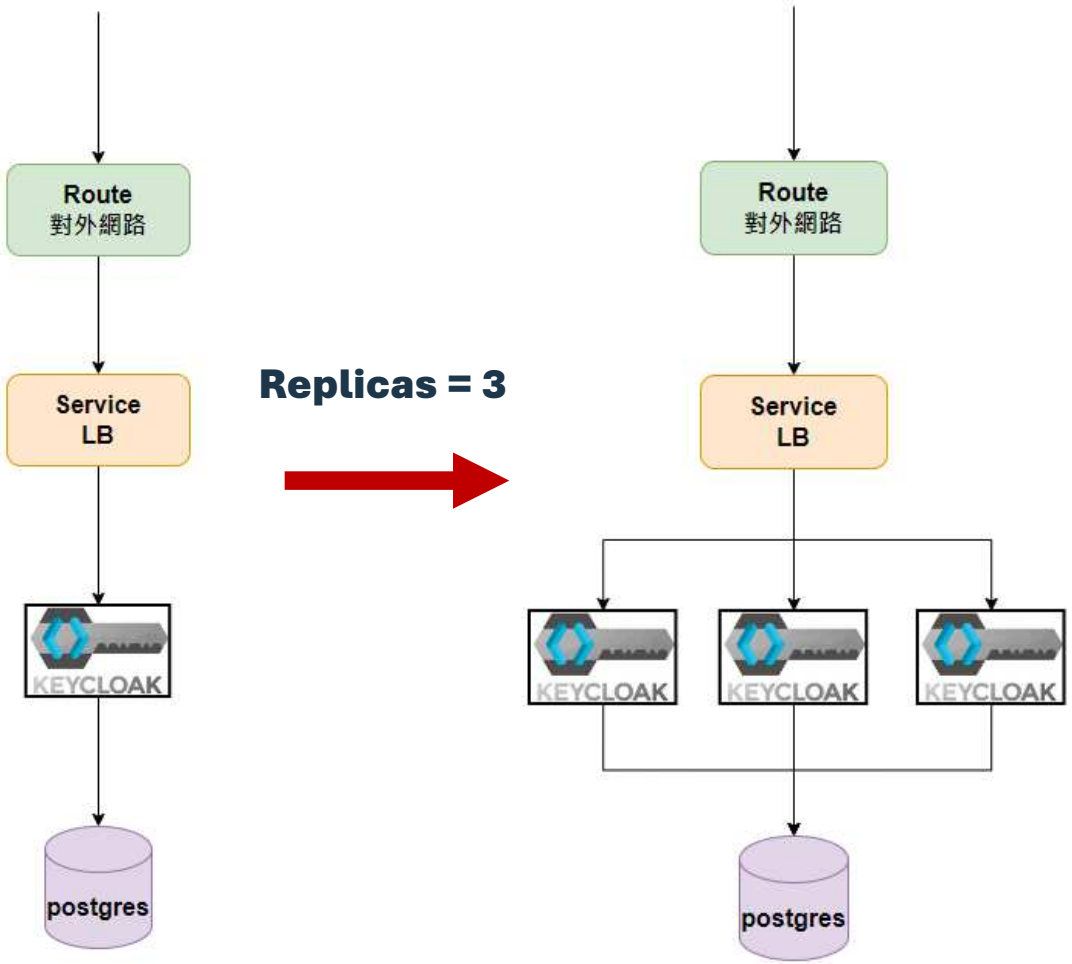
- 先檢查既有 SSO，減少重複登入
- 自動登入失敗時，安全降級至帳號密碼
- 依風險與政策動態啟用 MFA

商業產品比較

Feature	Keycloak	Microsoft Entra ID	Okta	Ping Identity	IBM Verify
OAuth2	✓	✓	✓	✓	✓
OIDC	✓	✓	✓	✓	✓
SAML	✓	✓	✓	✓	✓
LDAP	✓	Limited	Limited	✓	✓
AD	✓	✓	✓	✓	✓
Kubernetes	✓	Cloud	Cloud	Mixed	Mixed
SPI Extension	⭐⭐⭐⭐⭐	⭐	⭐	⭐⭐⭐	⭐⭐⭐
Source Code	✓	✗	✗	✗	✗
License Fee	None	Yes	Yes	Yes	Yes

Keycloak
Demo Showcase

Keycloak in Container架構功能展示



- 負載平衡
- 高可用性
- 彈性擴充
- 滾動更新
- 自動復原

Moodle與Keycloak整合認證展示

登入到IAM-Guru-Moodle

帳號或電子郵件

Enter your username or email

密碼

Enter your password

Forgot password?

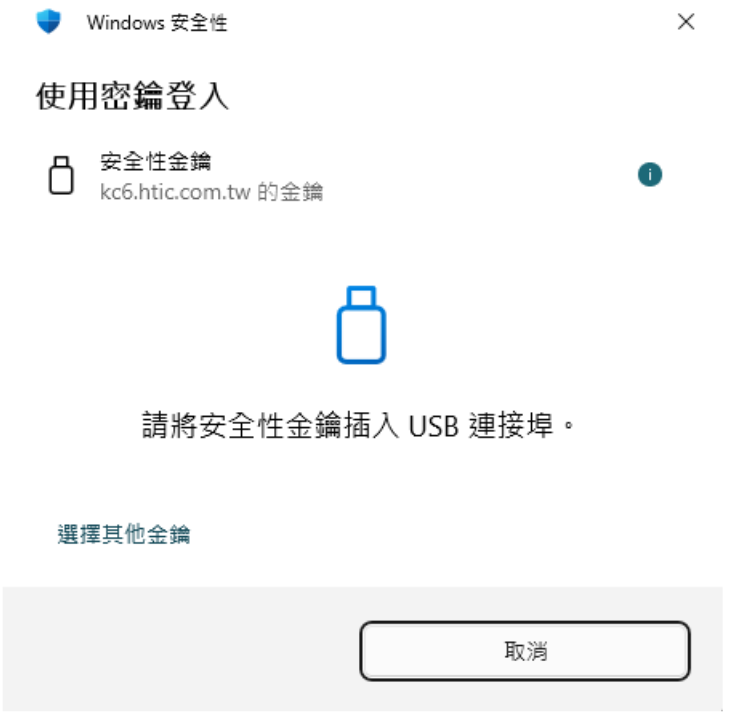
登入

OR

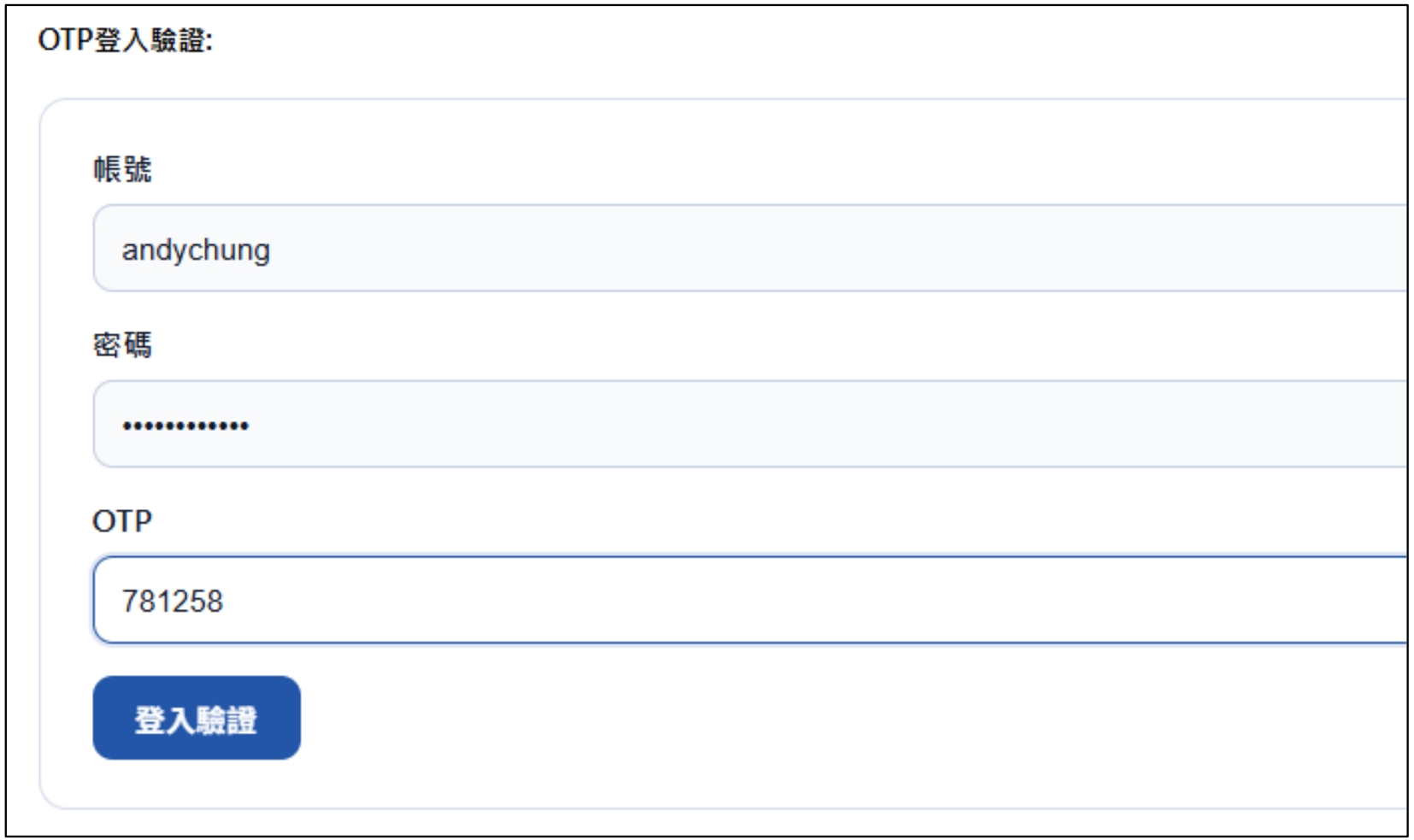
Log in with OIDC Keycloak

Log in with OAuth2 Keycloak

Keycloak MFA認證展示



自己的登入頁面 + Keycloak OTP驗證



升級驗證


提權驗證展示
Step-Up Authorization Code with PKCE:
此區會依設定頁的 Endpoint、Client_Id 與 redirect_uri 產生提權登入連結，並固定帶入 acr_values。
https://keycloak.apps.ocp.iam-guru.com/realms/iam-guru/protocol/openid-connect/auth?client_id=auth-client&redirect_uri=http://localhost:8080/callback&response_type=code&response_mode=query&scope=openid%20profile%20email&state=FiAh-SEEPtu6jD4hSej6sI04&nonce=GAt1GRiA71PPT8UzphZ9CcZi&code_challenge=86B2yX3tkt1ktIWlu7bzewLz51BzRNtNGM_FGCp8y0E&code_challenge_method=S256&acr_values=up
client_id = auth-client
redirect_uri = http://localhost:8080/callback
response_type = code
response_mode = query
scope = openid profile email
state = FiAh-SEEPtu6jD4hSej6sI04
nonce = GAt1GRiA71PPT8UzphZ9CcZi
code_challenge = 86B2yX3tkt1ktIWlu7bzewLz51BzRNtNGM_FGCp8y0E
code_challenge_method = S256
acr_values = up
點擊認證一
電子郵件
andychung@iam-guru.com
姓名
迪 鍾
修改郵件(OTP)
修改姓名(FIDO)
登出

API呼叫

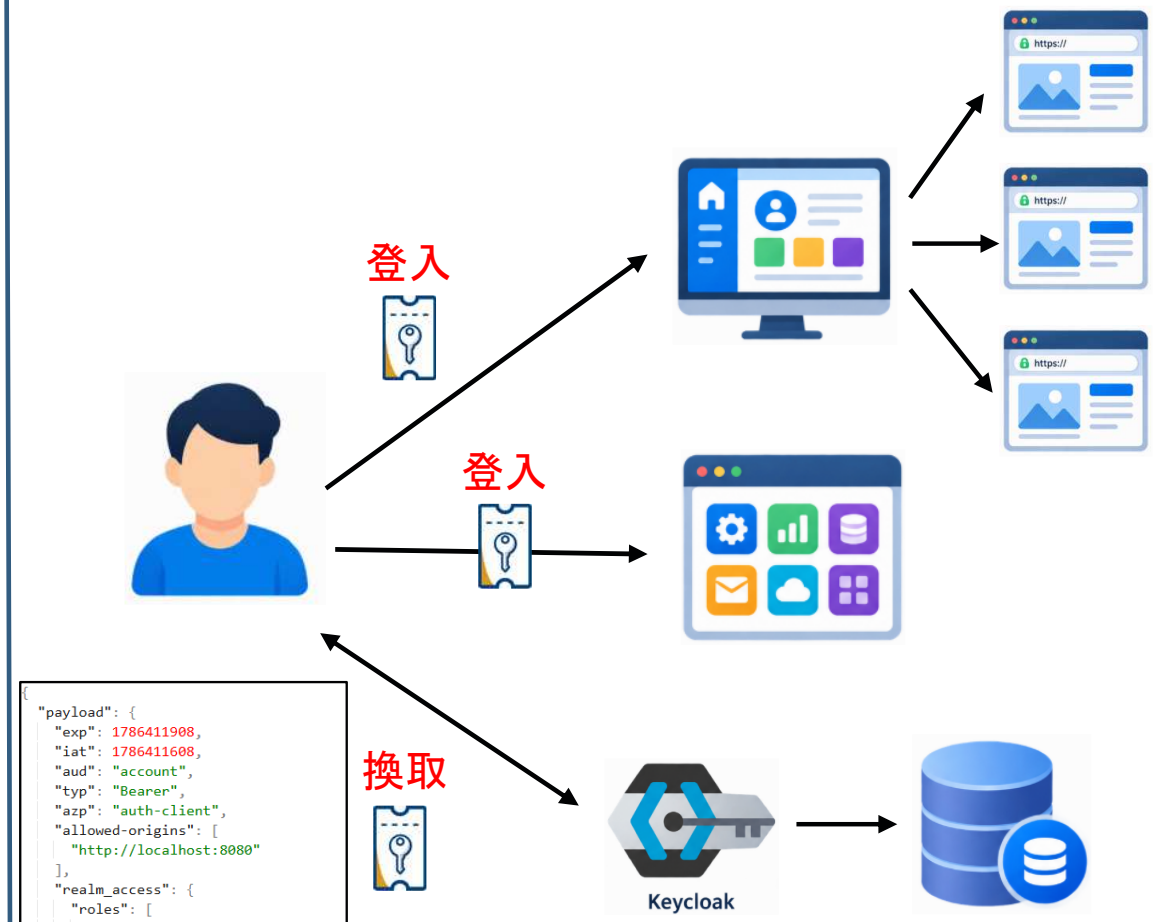

API認證展示
API Client設定:
api_client_id
api-client
api_client_secret
api_scope
query_parking
儲存
API展示:
取停車位資訊

政府與校園導入情境

人的數位身份

帳號：andychung
密碼：myP@ssword

傳統




```
{
  "payload": {
    "exp": 1786411908,
    "iat": 1786411608,
    "aud": "account",
    "typ": "Bearer",
    "azp": "auth-client",
    "allowed-origins": [
      "http://localhost:8080"
    ],
    "realm_access": {
      "roles": [
        "offline_access",
        "default-roles-iam-guru"
      ]
    },
    "resource_access": {
      "account": {
        "roles": [
          "manage-account",
          "view-profile"
        ]
      }
    },
    "scope": "openid profile email",
    "name": "迪 鍾",
    "preferred_username": "andychung",
    "email": "andychung@iam-guru.com"
  }
}
```

新世代

運用情境一：企業內部系統的單一登入（SSO）

🔍 描述：

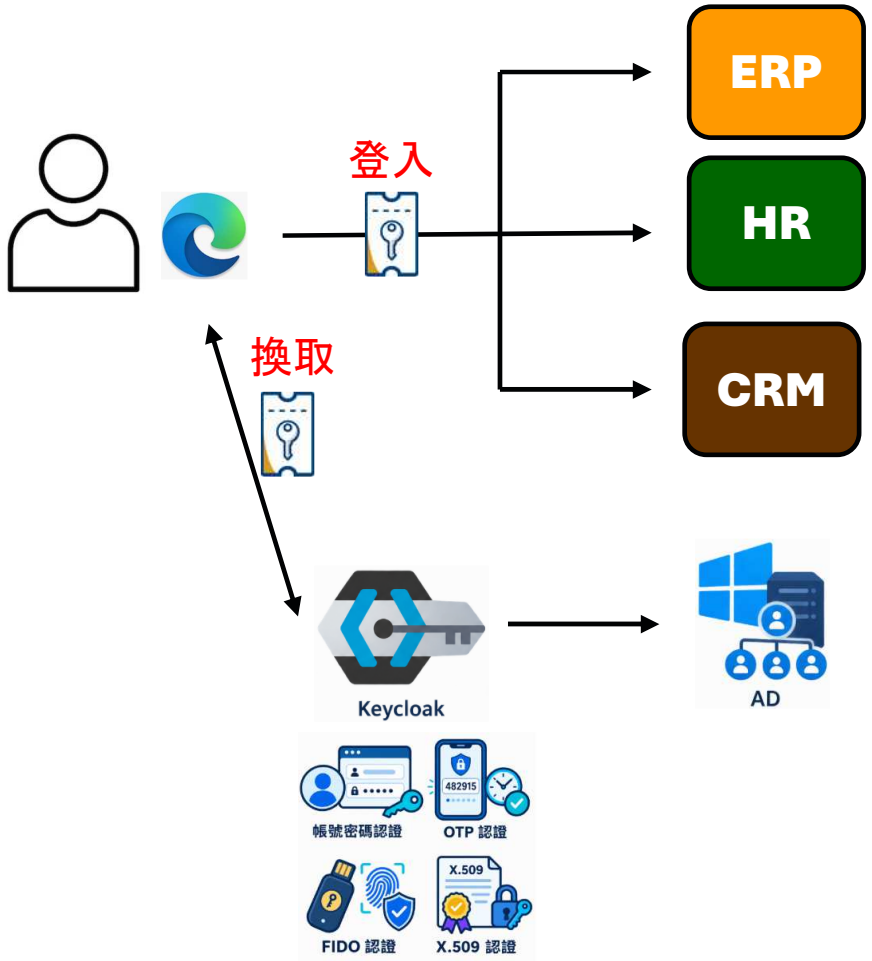
企業中有多個內部應用系統（如 ERP、HR、CRM），原本每個系統都有自己的帳號密碼管理，導致用戶體驗差、管理困難。使用 Keycloak 整合後，使用者只要登入一次即可存取所有系統。

✅ 關鍵功能：

- SSO（Single Sign-On）
- LDAP / AD 整合
- 使用者群組與角色管理
- 客製化登入頁面

🏗️ 建議架構：

- Keycloak + LDAP（例如 Active Directory）
- 每個應用系統都作為 Keycloak 的一個「Client」
- 可整合 OpenID Connect / SAML2.0



運用情境二：整合多個身分來源（Identity Brokering）

🔍 描述：

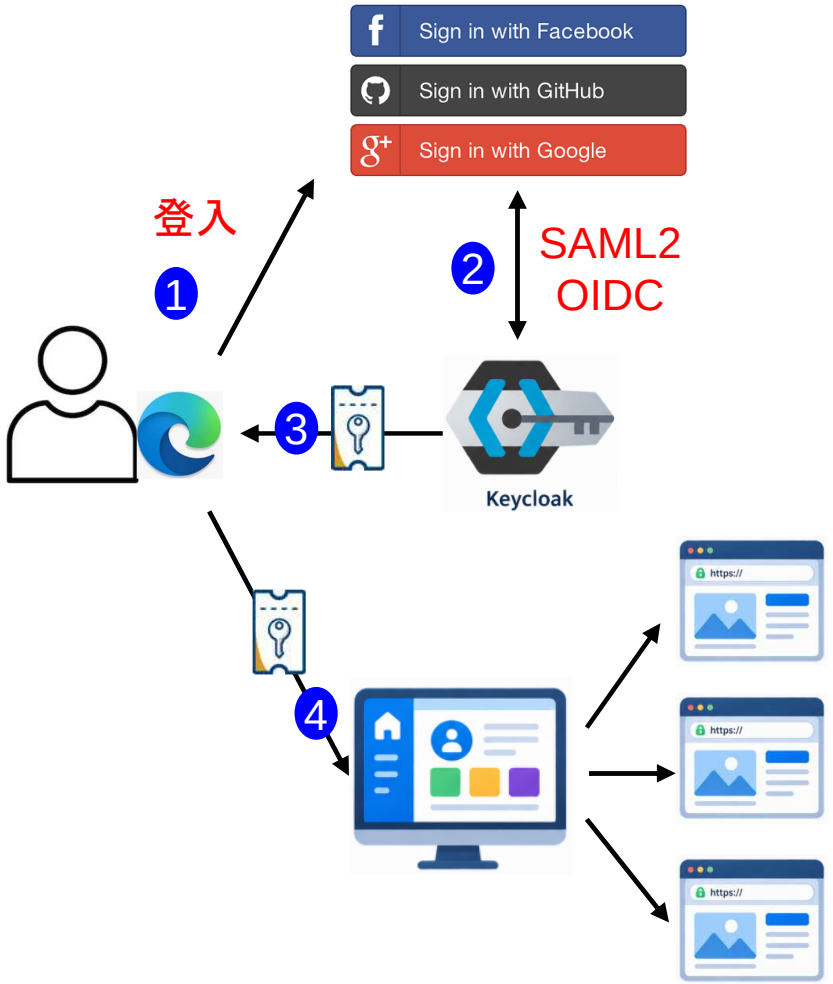
某企業有多個子公司，各自使用不同的身份系統（Apple、Facebook、Google），但希望讓使用者可統一登入 Portal。

✅ 關鍵功能：

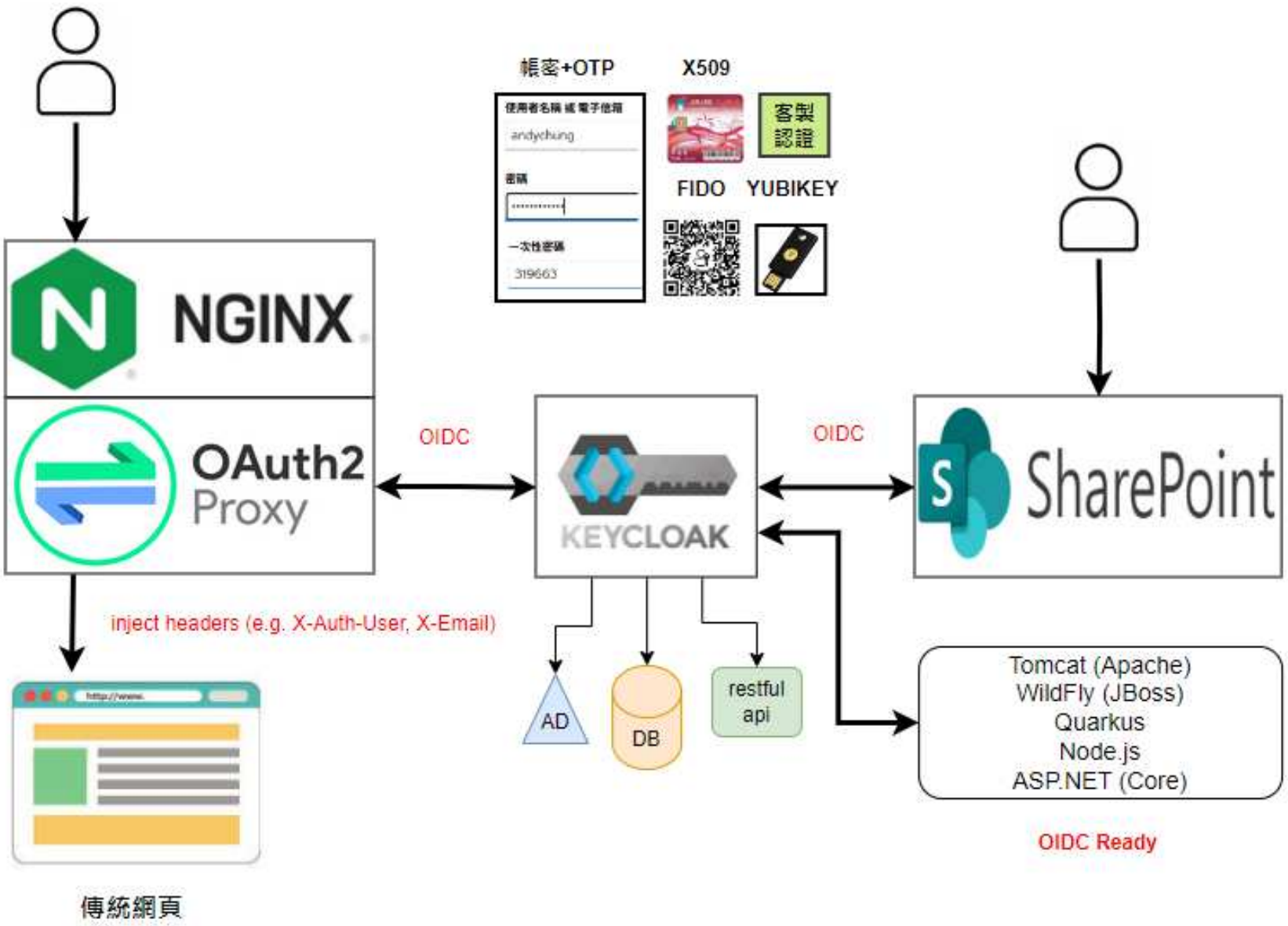
- Identity Provider Federation（整合多個身分提供者）
- 支援 SAML2 / OpenID Connect IDP
- Realm 分離管理

📐 建議架構：

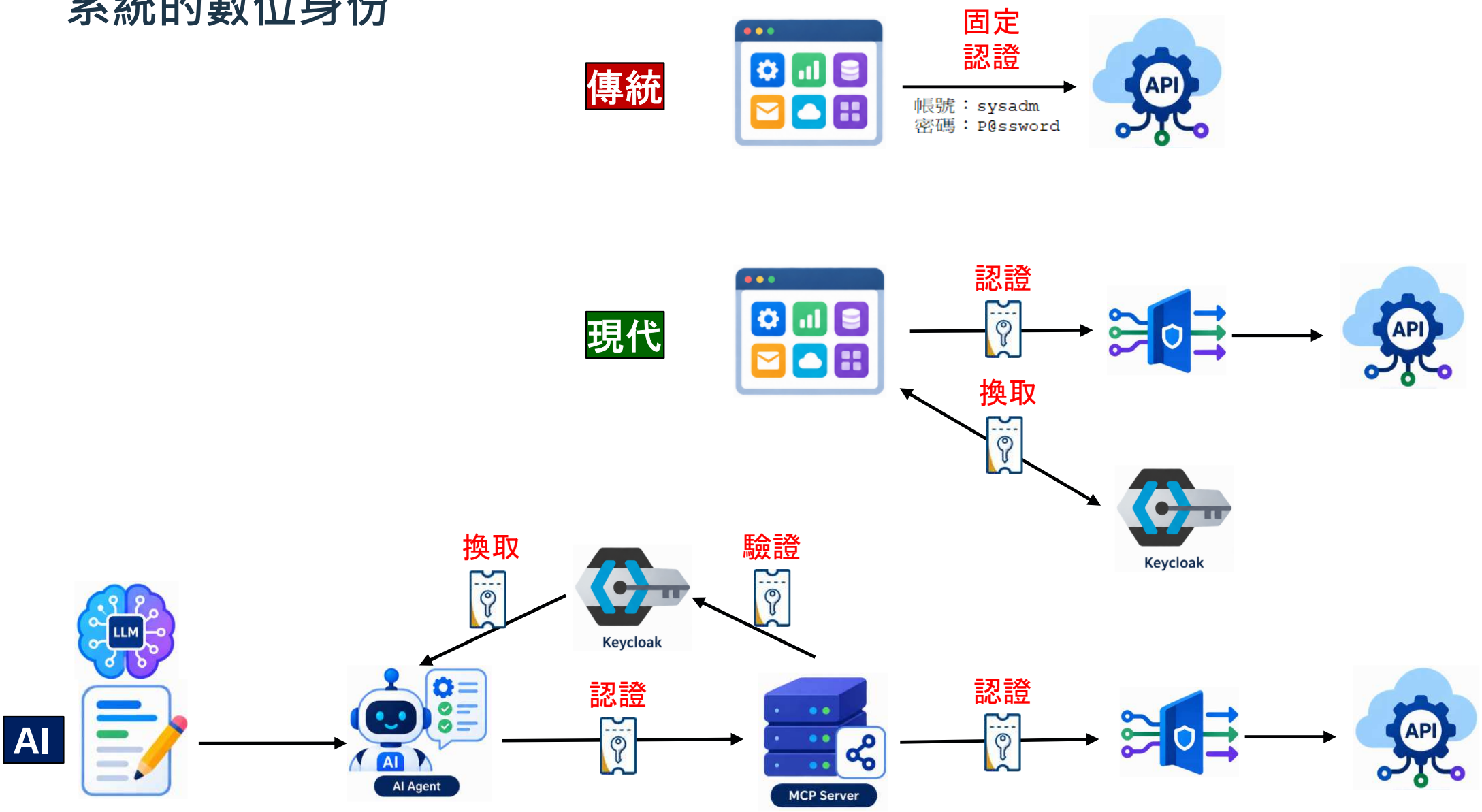
- 每個子公司作為一個 Identity Provider 接入
- Keycloak 作為 Broker，統一身份路由與授權



Keycloak整合傳統網頁



系統的數位身份


傳統
固定
認證
帳號：sysadm
密碼：P@ssword
API
現代
認證
換取
Keycloak
API
換取
驗證
Keycloak
AI
LLM
AI
AI Agent
認證
MCP Server
認證
API

運用情境三：微服務架構下的統一身分管理

🔍 描述：

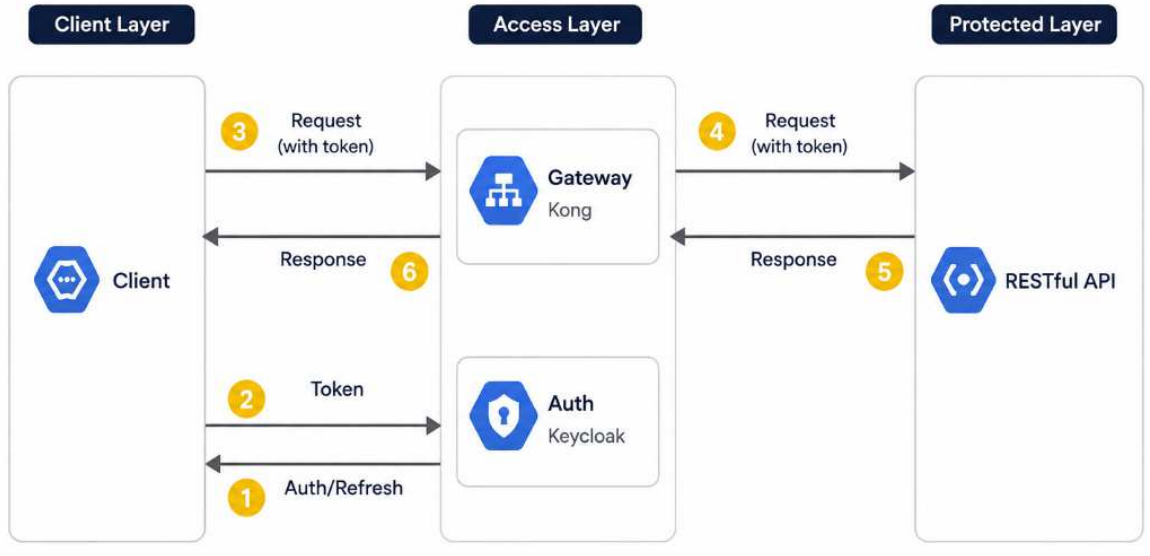
在微服務架構中，每個服務都需要驗證與授權，
如果每個服務都自己處理認證會非常複雜。
Keycloak 在微服務架構中則扮演集中式的身分提供者與 OAuth 2.0 / OIDC 授權伺服器。

✅ 關鍵功能：

- OAuth2 / OpenID Connect
- JWT Token 發行與驗證
- Fine-grained Authorization（資源級授權）

📐 建議架構：

- 微服務透過API Gateway驗證 JWT Token
- Keycloak 作為授權伺服器

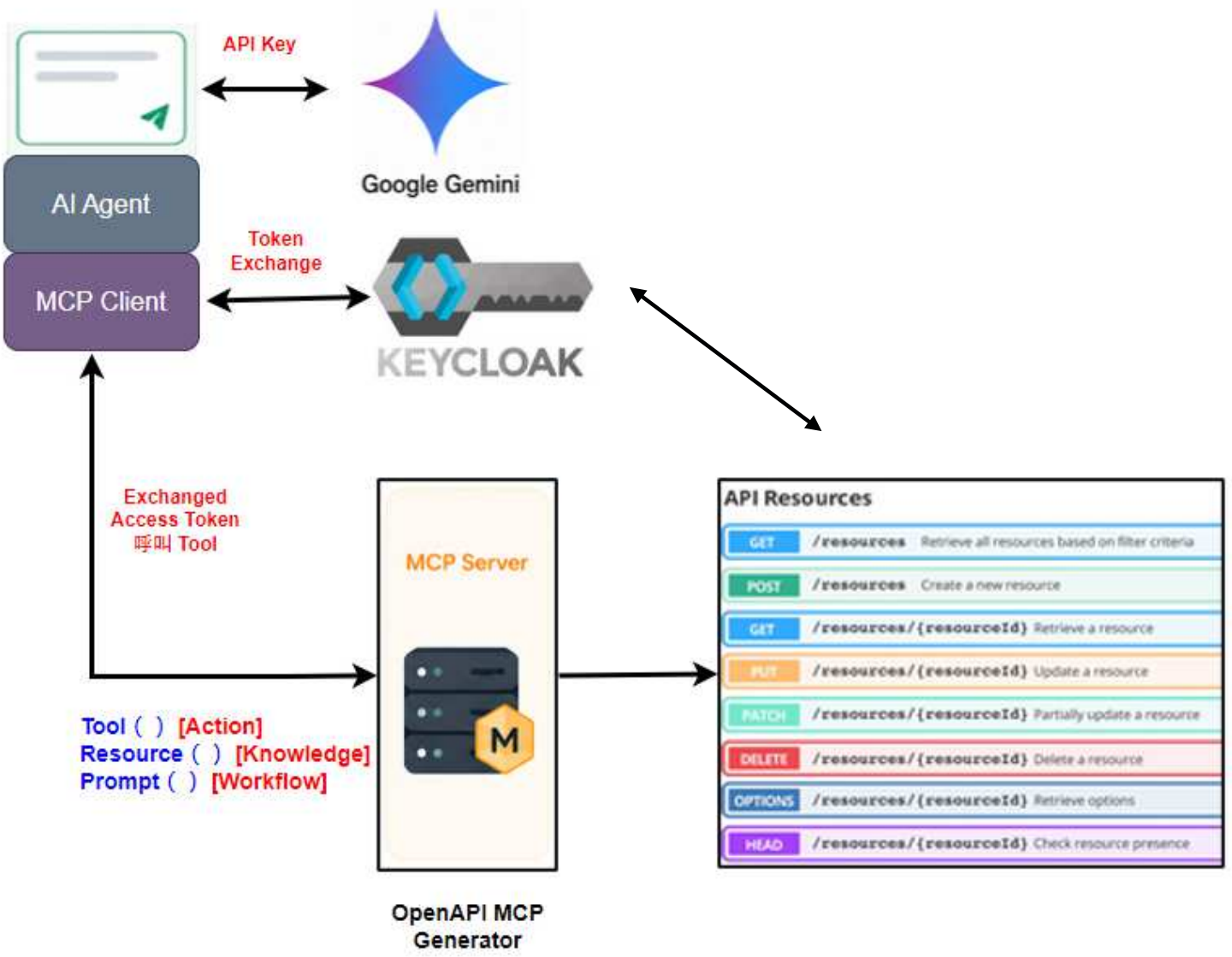


邁入AI的世界


MCP Server
M


API Resources
GET /resources Retrieve all resources based on filter criteria
POST /resources Create a new resource
GET /resources/{resourceId} Retrieve a resource
PUT /resources/{resourceId} Update a resource
PATCH /resources/{resourceId} Partially update a resource
DELETE /resources/{resourceId} Delete a resource
OPTIONS /resources/{resourceId} Retrieve options
HEAD /resources/{resourceId} Check resource presence

邁入AI的世界


API Key
Google Gemini
AI Agent
MCP Client
Token
Exchange
KEYCLOAK
Exchanged
Access Token
呼叫 Tool
MCP Server
Tool ( ) [Action]
Resource ( ) [Knowledge]
Prompt ( ) [Workflow]
API Resources
GET /resources Retrieve all resources based on filter criteria
POST /resources Create a new resource
GET /resources/{resourceId} Retrieve a resource
PUT /resources/{resourceId} Update a resource
PATCH /resources/{resourceId} Partially update a resource
DELETE /resources/{resourceId} Delete a resource
OPTIONS /resources/{resourceId} Retrieve options
HEAD /resources/{resourceId} Check resource presence
OpenAPI MCP
Generator

使用Keycloak是否需要修改程式

✅ 不一定需要修改程式的情況

如果你的系統支援 **標準身份驗證協議**（例如 OIDC 或 SAML），你可以：

- 透過 **反向代理（如 NGINX、Apache）** 配合 Keycloak 做 SSO
- SharePoint portal登入**整合**，**讓**Keycloak**當**Identity Source + MFA
- 使用 **現有的 Keycloak Adapter / SDK** 整合，通常只需要設定（少量修改）

❗ 需要修改程式的情況

以下幾種情況可能需要你修改程式碼：

1.應用程式沒有支援 OIDC/SAML 的機制

- 例如老舊系統或自訂身份驗證機制（帳密硬寫在程式中）
- 解法：需要改寫認證邏輯，改用 Token 驗證

2.需要細緻的權限控制（RBAC / ABAC）

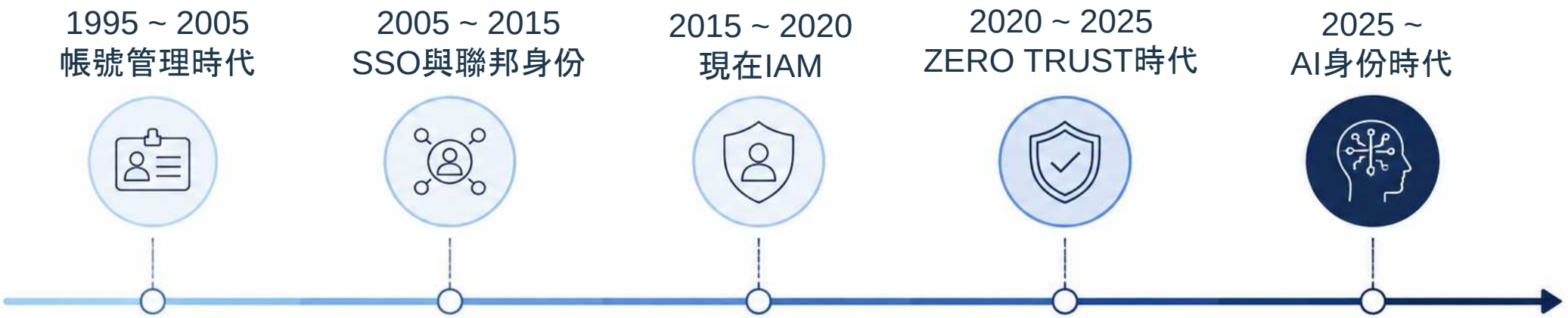
- 若你要根據 Keycloak 的角色 / 權限做授權控制，可能要在程式中加入解析 JWT 或查詢 Keycloak 的程式碼

3.整合 Keycloak 提供的用戶管理介面

- 例如在後台讓管理員可以 reset 密碼、指派角色 → 要使用 Keycloak Admin REST API

現代 IAM
與 Zero Trust 架構演進

身份治理演進時間軸



各系統獨立
- 帳號人工建立與管理
- 密碼分散難記
- 離職帳號難以控管

單一登入體驗
- 單一帳號登入多系統
- 身份聯邦整合
- 提升使用者體驗

集中治理與強化安全
- 身份生命週期管理
- 多因素驗證（MFA）
- 角色與權限控管（RBAC）
- 合規與稽核機制

永不信任‧持續驗證
- 身份為核心的存取控制
- 裝置、位置、風險評估
- 最小權限原則
- 微分段與持續授權

人機協作的新身份治理
- 人類、機器與 Agent 身份並存
- AI Agent 權限與治理
- 動態授權與情境感知
- 身份成為 AI 信任的基礎

Zero Trust 必須延伸到 AI

不信任任何請求，不分人或 AI；每一次存取都必須驗證與授權

核心原則 | **永不信任、持續驗證、最小權限、完整稽核**

適用對象：人、API、容器、機器、AI Agent、MCP Server

Zero Trust 在 AI 時代的實踐方式

最小權限	即時授權	身分隔離	行為監控	完整稽核
僅授予必要權限	任務結束即失效	每個 Agent 獨立身分	持續偵測異常	記錄每次存取

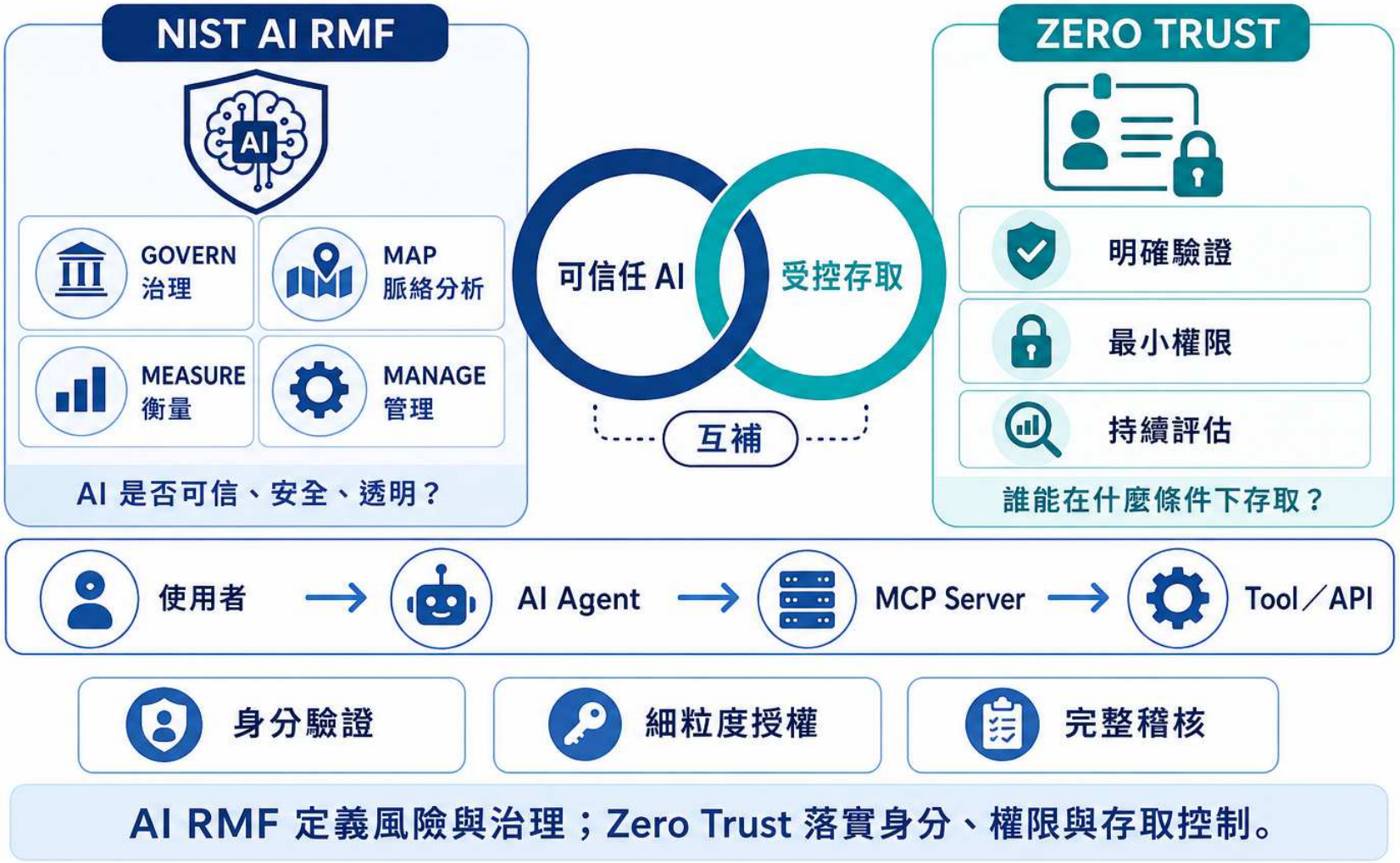
重點結論

AI 不是新的應用，而是新的「行為者（Actor）」

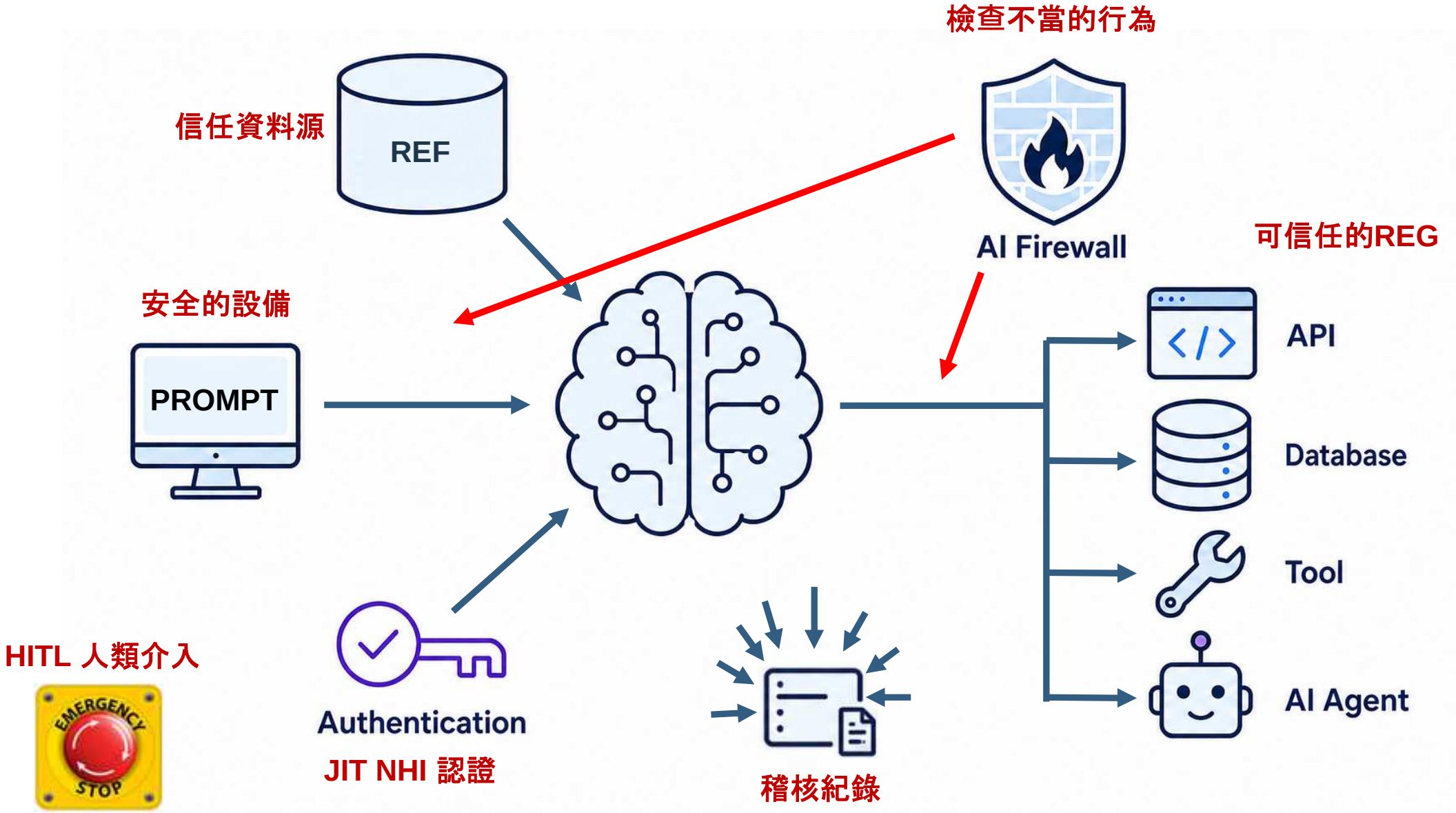
Zero Trust 必須延伸到每一個 AI Agent 與工具

NIST AI RMF × Zero Trust

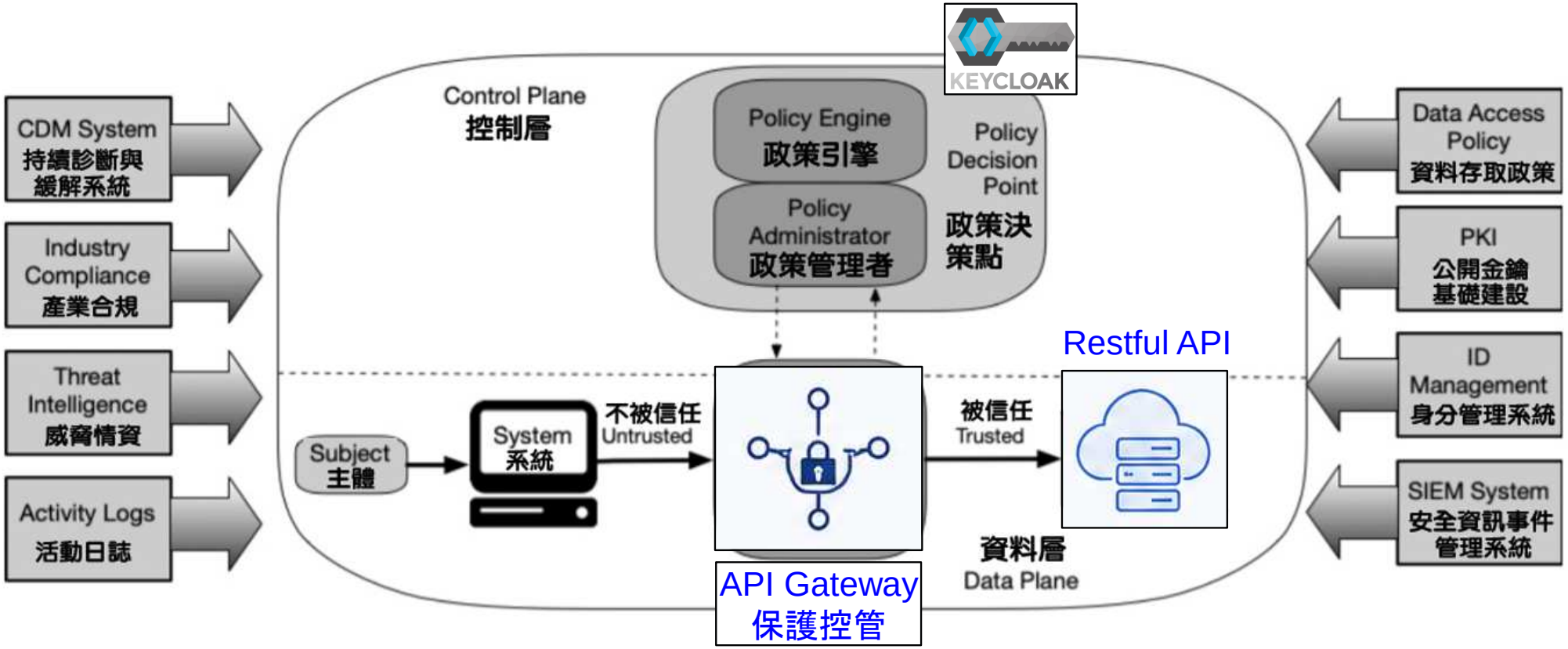
AI 風險治理與動態存取控制的互補關係



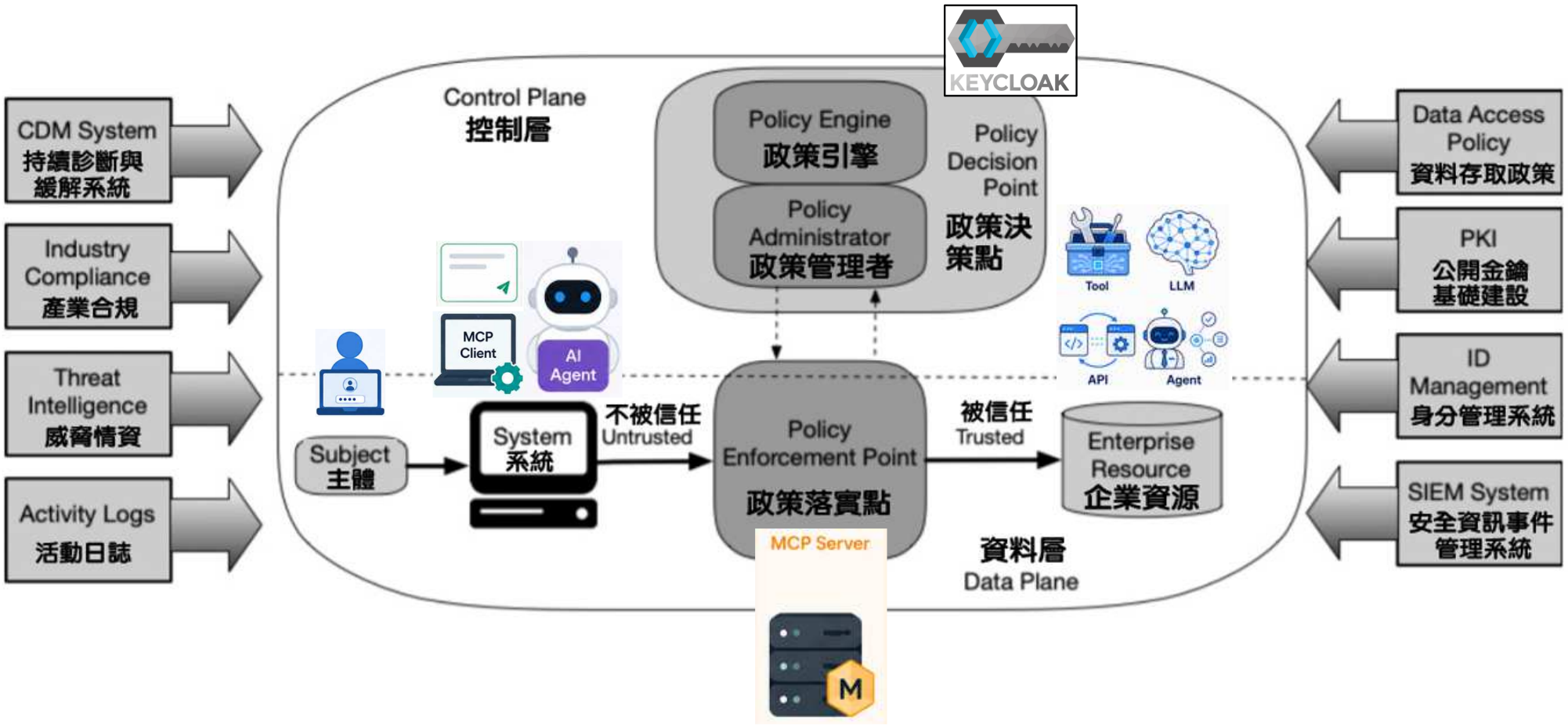
NIST AI RMF


檢查不當的行為
信任資料源
REF
AI Firewall
可信任的REG
安全的設備
PROMPT
API
Database
Tool
AI Agent
HITL 人類介入
EMERGENCY
STOP
Authentication
JIT NHI 認證
稽核紀錄

Zero Trust架構 for API



Zero Trust架構 for AI



身份治理的未來趨勢

身份治理的核心概念

確保正確的資源，授予正確的人員，在正確的時間，出於正確的原因，並以正確的方式進行管理與監控

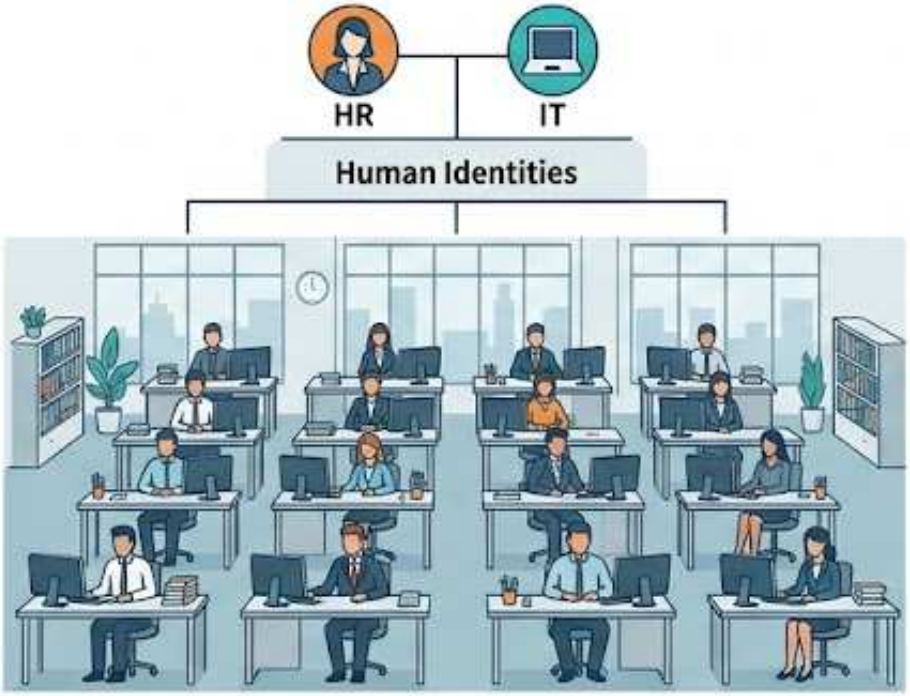
核心價值

降低風險	提升合規	營運效率	增強安全性	降低成本
減少不當存取與內部威脅	滿足法規要求與稽核標準	自動化流程，減少人工作業負擔	落實最小權限，強化整體資安防護	減少風險損失與管理成本

身份治理的演進：從「人」到「AI Agent」

過去 (PAST)

過去身份治理是在管理「人」



≈10 人

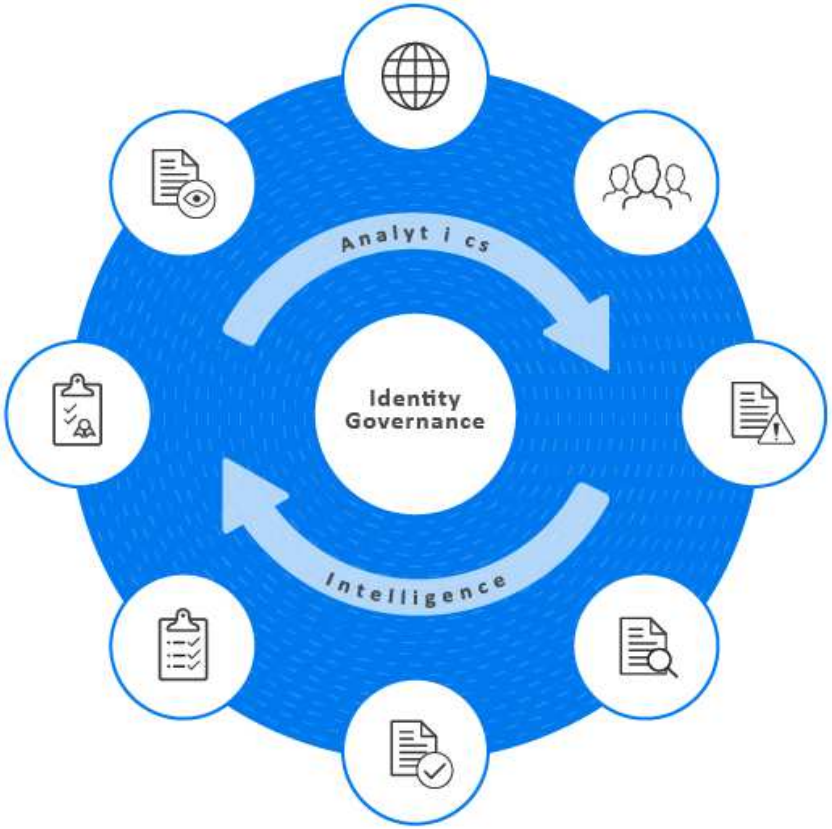
未來 (FUTURE)

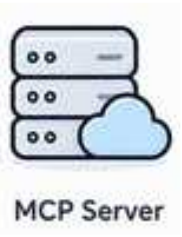
未來身份治理是在管理「所有自主行動的數位身份」

會自主行動的數位身份

比如說以前一家公司需要十個人，未來是一個人搭配數百個AI Agents

AI的Identity Governance





目前AI的Identity Governance 還停留在人類身份
目前幾乎沒有成熟流程。因此很多 AI Agent：其實沒有真正的生命週期管理。

[NOWNEWS今日新聞] 逾千名來自全球頂尖AI企業員工，包括公司內的專家與高層，近日聯署發表公開信，敦促美國政府採取行動減緩AI的開發速度，以便為安全防護與監管措施爭取時間。包括OpenAI首席科學家兼聯合創辦人、Anthropic創辦團隊成員，以及Meta與Google等科技巨頭的副總裁級高層都參與其中。

美國AI新創大廠「Anthropic」高層人士於（4）日透過官方部落格發表聯名文章，呼籲全球暫緩人工智慧（AI）研發腳步，原因在於AI技術正加速發展，甚至出現「遞迴式自我改進」的能力，可能帶來無法預期的風險。文章強調，若全球能共同放慢AI研發速度，讓社會有時間因應科技帶來的衝擊，將對人類社會有正面影響。該呼籲由共同創辦人Jack Clark與研究主管Marina Favaro執筆，並提出建立全球協議及驗證機制的具體建議。

現在能做的事情

#	議題	目前成熟度	關鍵缺口
1	AI 可以代表人類到什麼程度？	🔴 **尚未完善**	代理範圍或代理鏈治理是很難定義
2	AI 做錯事情誰負責？	🔴 **尚未完善**	這不只是 IAM 技術問題，還牽涉法律、組織責任與 AI Governance
3	AI 可以把權限交給另一個 AI 嗎？	🔴 **尚未完善**	Multi-hop Delegation、Delegation Chain 的政策與稽核仍不成熟
4	AI Agent Lifecycle	🔴 **尚未完善**	Agent 的建立、授權、變更、撤銷治理仍在發展
5	Governance Gap	🔴 **目前最大的問題**	AI 能力演進速度 > 企業治理與制度建立速度

我們已經有 OAuth、OIDC、Token Exchange、RBAC/ABAC、Workload Identity、IGA、Audit 等技術積木，但還缺乏一套成熟且普遍採用的方式，把這些能力完整組合成 AI Agent Identity Governance。

總結

AI 時代，重新定義 Identity

從「管理人」到「管理所有會做事的實體」

傳統 IAM

Human → Application

管理人類使用者存取應用程式

- 帳號 / 密碼
- 角色權限
- 存取控制

Zero Trust 時代

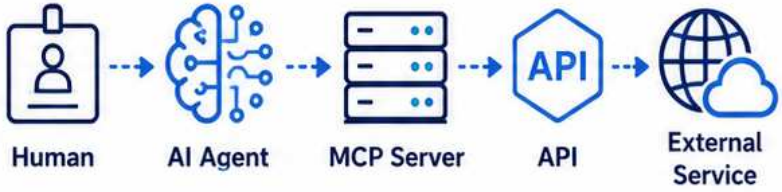
Human → Application → API

延伸信任邊界，深入到 API 層

- 身分驗證 Authentication
- 授權 Authorization
- 最小權限 Least Privilege
- 持續驗證 Continuous Verification

AI 時代

Human → AI Agent → MCP Server → API → External Service



- 多重身分：人 + 機器 + AI
- 動態授權 Just-in-Time Access
- 最小權限原則
- 完整稽核追蹤 Audit
- 全面落實 Zero Trust

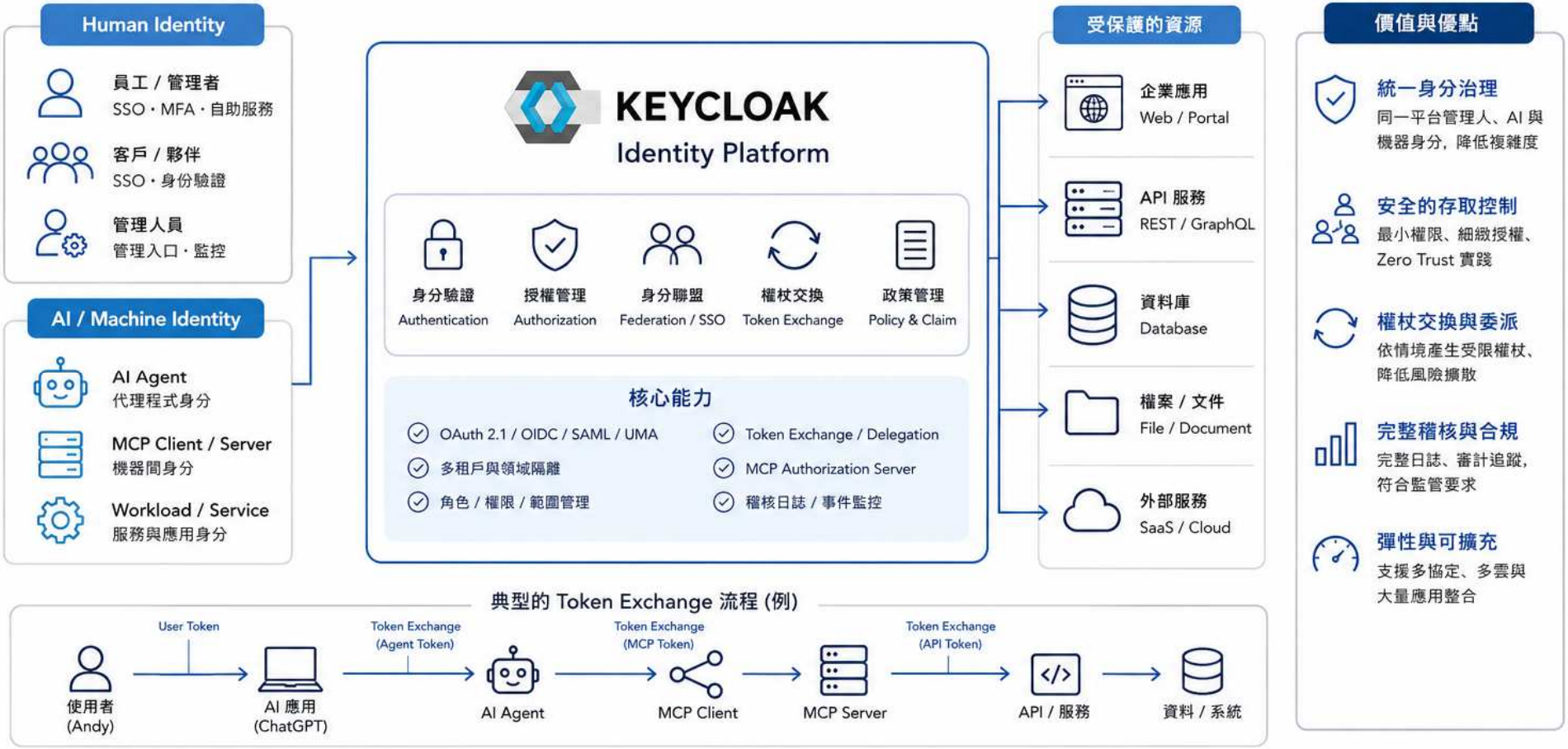
Identity 已從「管理人」演變成「管理行為者 (Actor)」

人、API、Container、Machine、AI Agent、MCP Server 都必須擁有可驗證的身分

Keycloak：統一的 Identity Platform

同時管理人與 AI 的身分與存取

一個平台，支援 Human、AI Agent、Application、Workload 的身分治理與安全存取



Identity Platform 是所有安全存取的根基

一個平台，多種身分
Human + AI + Machine 統一管理

以政策驅動安全
政策定義、強制執行、持續驗證

可見、可控、可稽核
全生命週期治理與風險降低

KEYCLOAK

THANK YOU

如果給我6個小時砍下一顆樹，我會用前面4個小時把斧頭磨利。

Give me six hours to chop down a tree and I will spend the first four sharpening the axe.

Abraham Lincoln

鋐迪資訊
https://www.htic.com.tw