

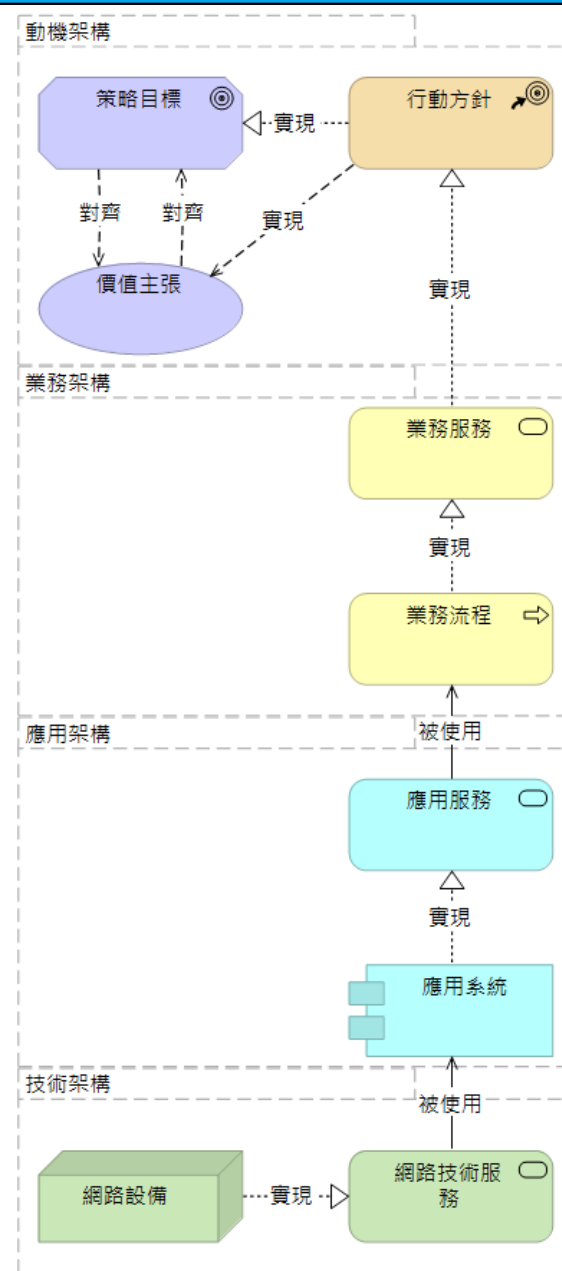
數位轉型中的ArchiMate應用 (ArchiMate Applications in Digital Transformation)

南山人壽 資深企業架構顧問
前台北市政府 數位創新辦公室 企業架構總監
社團法人中華企業架構師學會(a|CEA) 榮譽理事長
國際企業架構師協會(a|EA) 台灣分會 副理事長
工業技術研究院 產業學院 首席架構師講座
美國海軍研究院 電腦科學系 博士

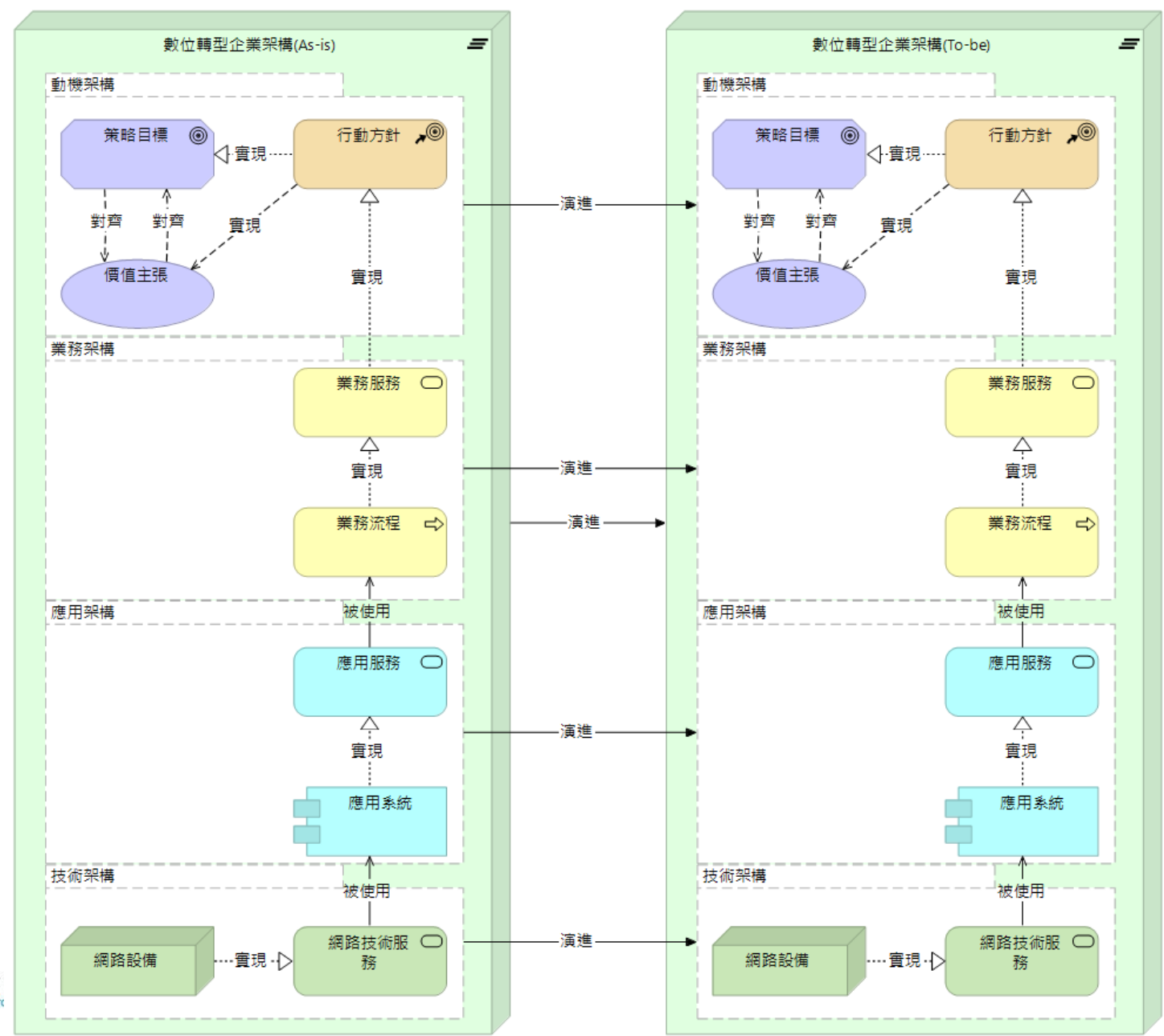
韓孟麒



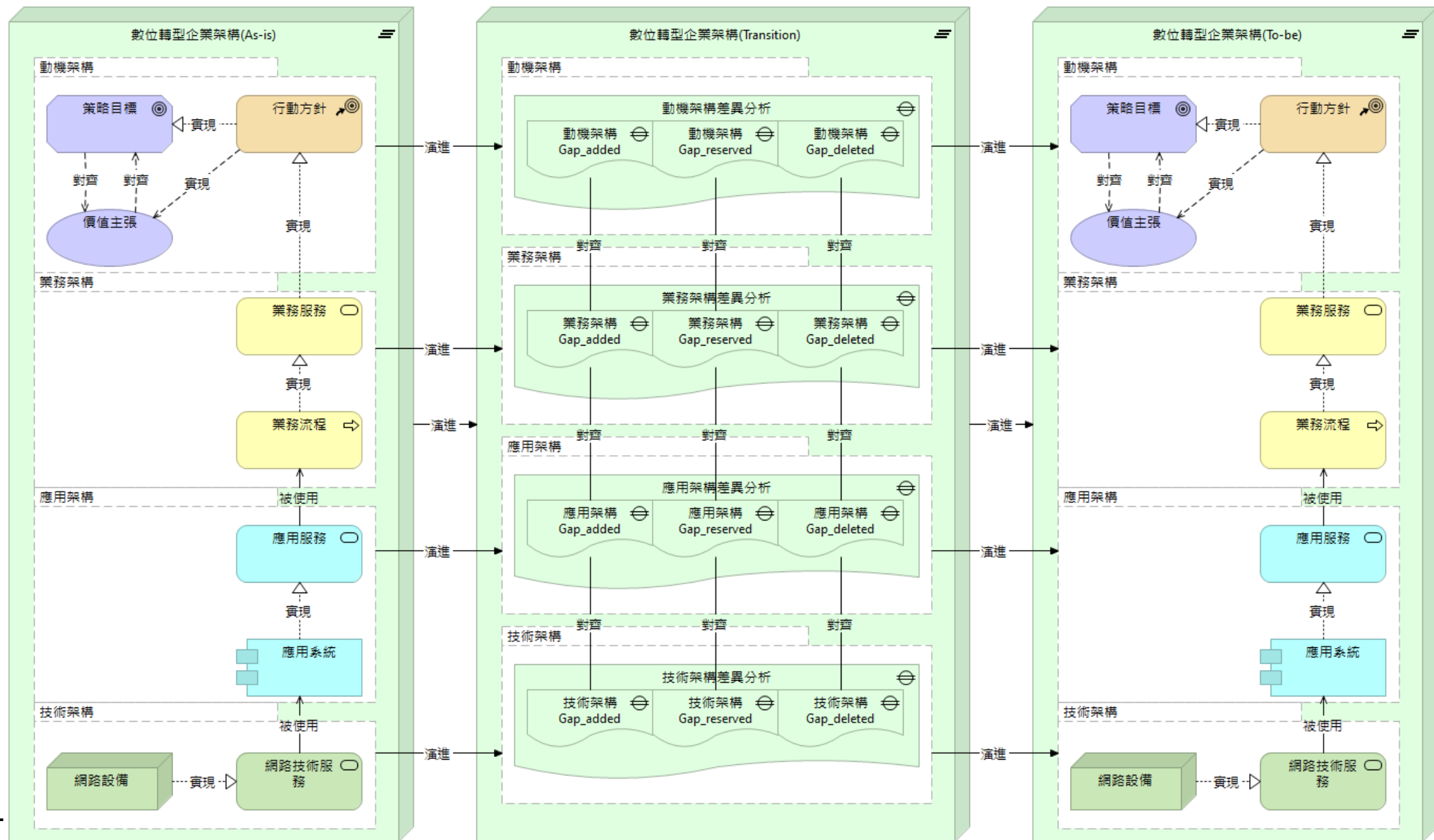
Frameworks in Digital Transformation



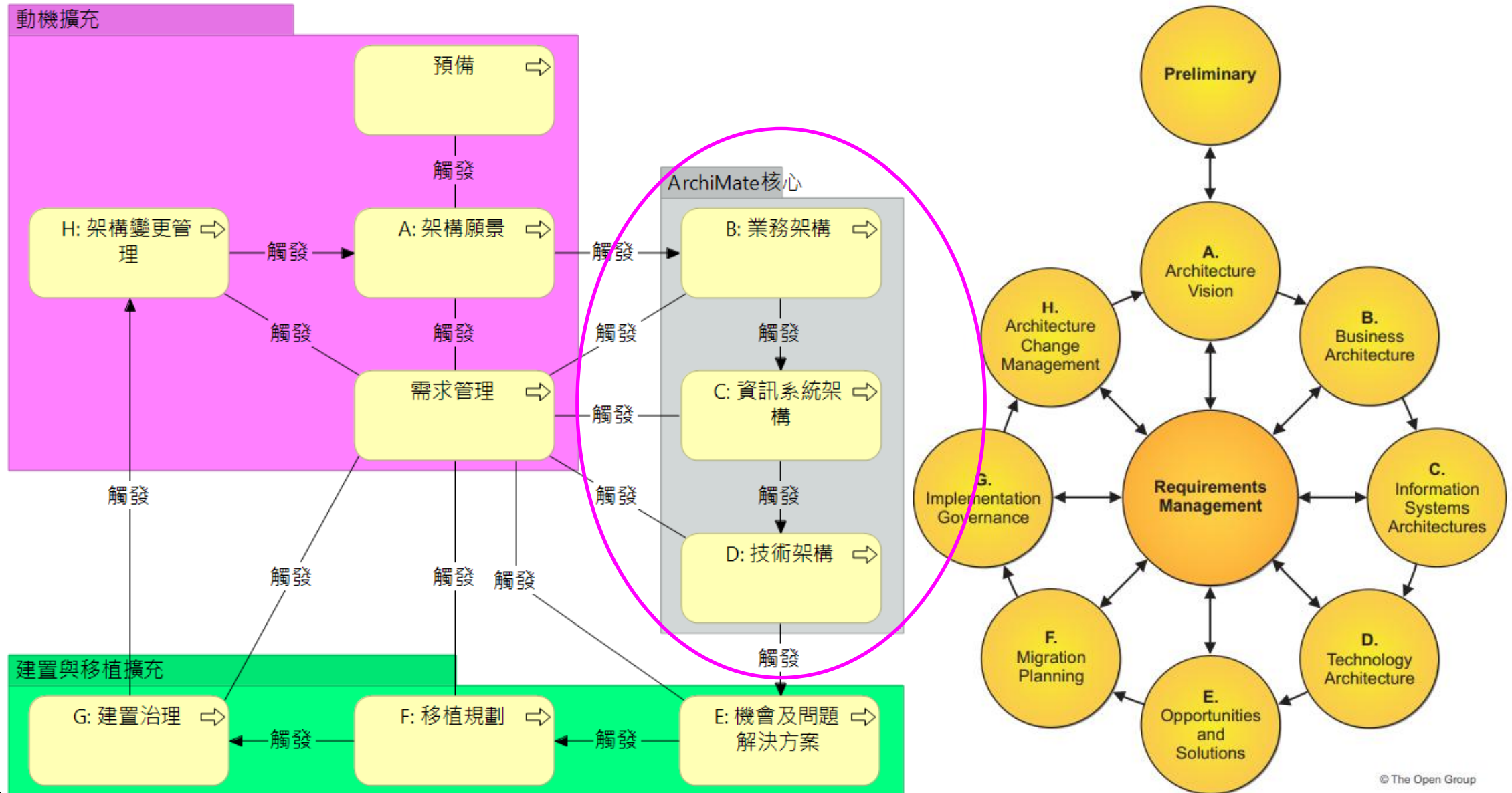
As-is and To-be Frameworks in Digital Transformation



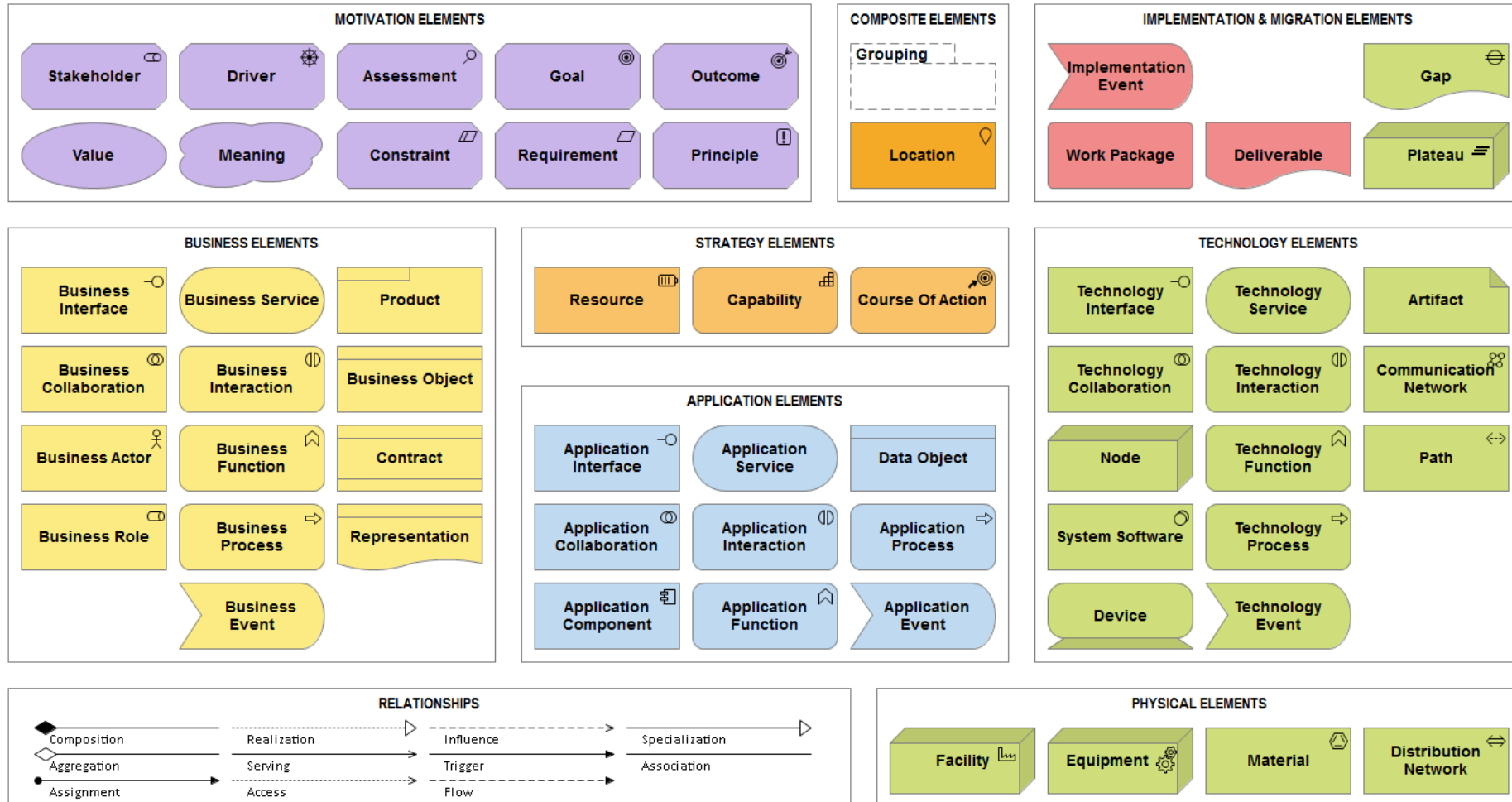
Enterprise Architecture of Digital Transformation



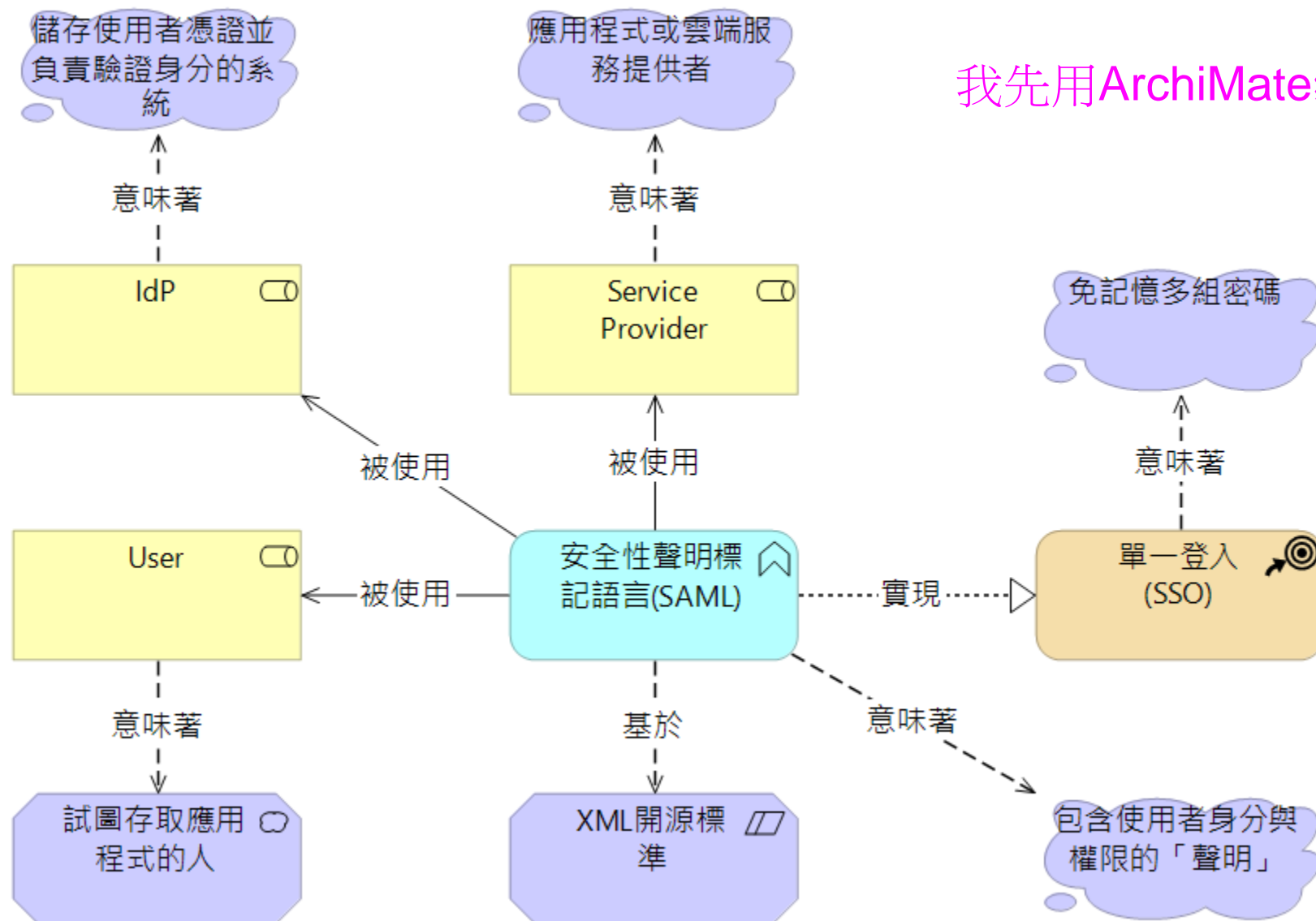
The Open Group Architecture Framework (TOGAF): ADM



TOGAF Modeling Language: ArchiMate

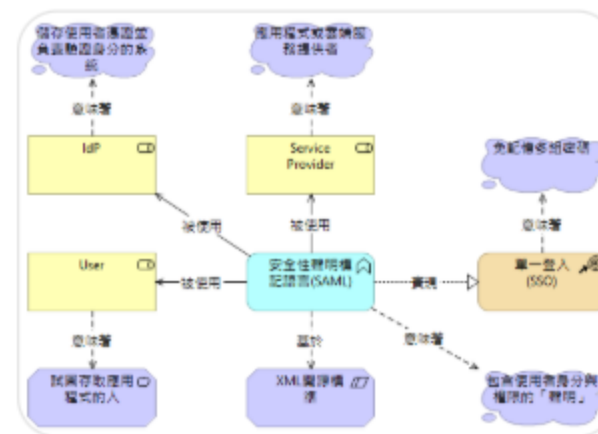


Examples for Semantic Network via ArchiMate (SNA): SAML



我先用ArchiMate繪出語意視圖

請AI Agent解析視圖之語意



此圖叫Semantic Network via ArchiMate (SNA), 請用語意網路語法, 解析此視圖。

根據您提供的圖片內容，這是一張 **Semantic Network via ArchiMate (SNA)** 語意網路圖，主要描述 **SAML (Security Assertion Markup Language)** 與 **單一登入 (SSO)** 之間的語意關係。以下使用語意網路語法進行解析。

語意網路語法格式說明

它看出SNA的語法

text

复制 下载

[節點A] --[關係]--> [節點B]

- 節點：代表概念、實體或系統
- 關係：代表語意連結（如：實現、被使用、包含、負責等）

語意網路解析

並把節點做了說明

核心節點

節點名稱	說明
IdP	儲存使用者憑證並負責驗證身分的系統 (身分提供者)
User	試圖存取應用程式的人
SP	應用程式或雲端服務提供者 (Service Provider)
SAML	記錄語言，XML 開源標準
SSO	單一登入 (Single Sign-On)
Assertion	包含使用者身分與權限的「聲明」

它依語法, 解析的正規化程式

1. IdP --[負責驗證身分]--> User
2. IdP --[儲存]--> 使用者憑證
3. IdP --[使用]--> SAML
4. SAML --[實現]--> SSO
5. SSO --[被使用]--> SP
6. SP --[提供服務給]--> User
7. User --[試圖存取]--> SP
8. SAML --[包含]--> Assertion
9. Assertion --[描述]--> 使用者身分
10. Assertion --[描述]--> 權限
11. SAML --[是一種]--> XML開源標準
12. SP --[信任]--> IdP
13. IdP --[產生]--> Assertion
14. Assertion --[傳送給]--> SP

它由視圖, 解析的自然語言語意

語意摘要

這張 SNA 圖表達的核心語意為：

SAML 是一種 XML 開源標準，作為記錄語言實現了單一登入 (SSO) 機制。IdP (身分提供者) 使用 SAML 來產生包含使用者身分與權限的「聲明 (Assertion)」，並將其傳送給 SP (服務提供者)，讓使用者只需一次驗證即可存取多個應用程式。

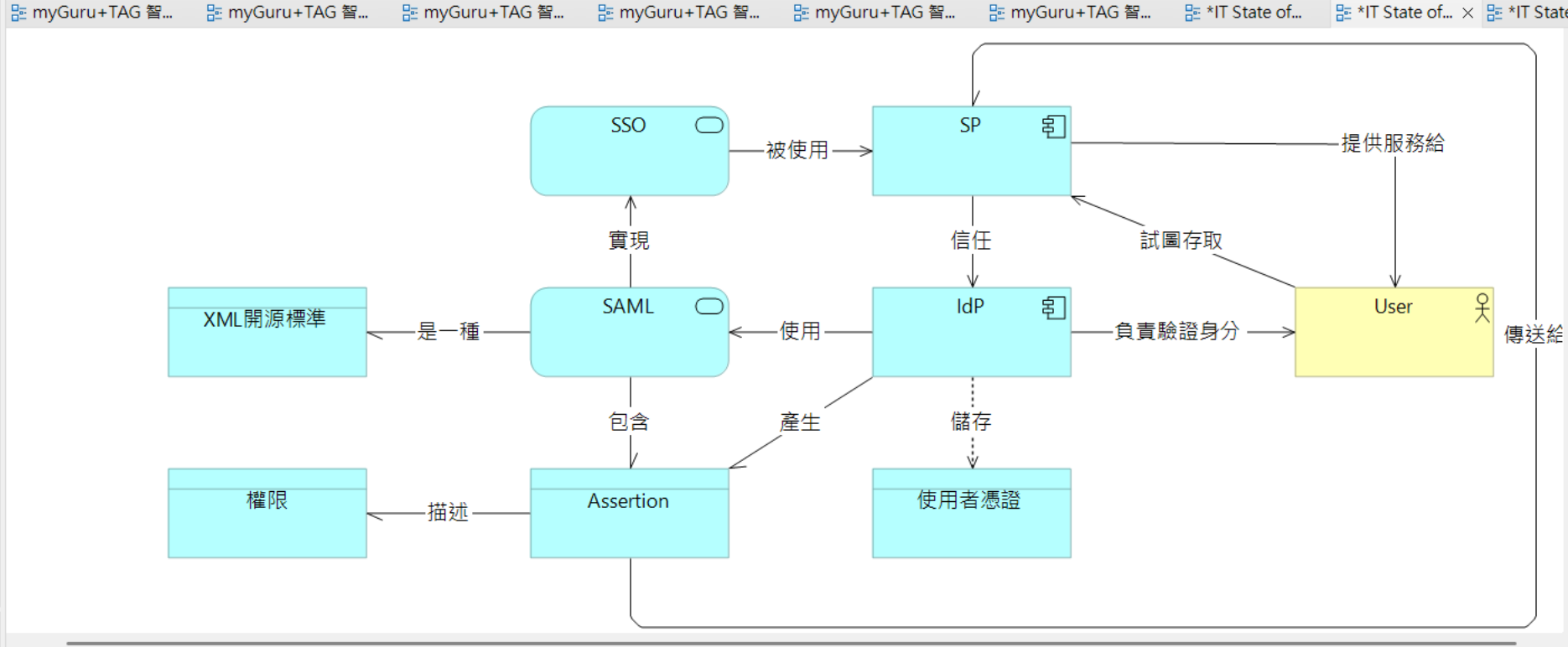
Models x

*IT State of Arts

- Strategy
- Business
- Application
- Technology & Physical
- Motivation
- Implementation & Migration
- Other
- Relations
- Views
- 01_SAML
- 01_SAML_Semantic
- New ArchiMate View

2026_成果導向業務流程梳理與 ABB_SBB

- Strategy
- Business
- Application
- Technology & Physical
- Motivation
- Implementation & Migration
- Other



Palette

- Magic Connector
- Composition relation
- Aggregation relation
- Assignment relation
- Realization relation
- Serving relation
- Access relation
- Influence relation
- Triggering relation
- Flow relation
- Specialization relation
- Association relation
- Junction
- Note
- Group
- Connection
- Location
- Grouping

Properties x Validator Visualiser Change History Branches

01_SAML_Semantic

Main Viewpoint: None

Properties Name: 01_SAML_Semantic

Appearance Documentation:

語意關係三元組

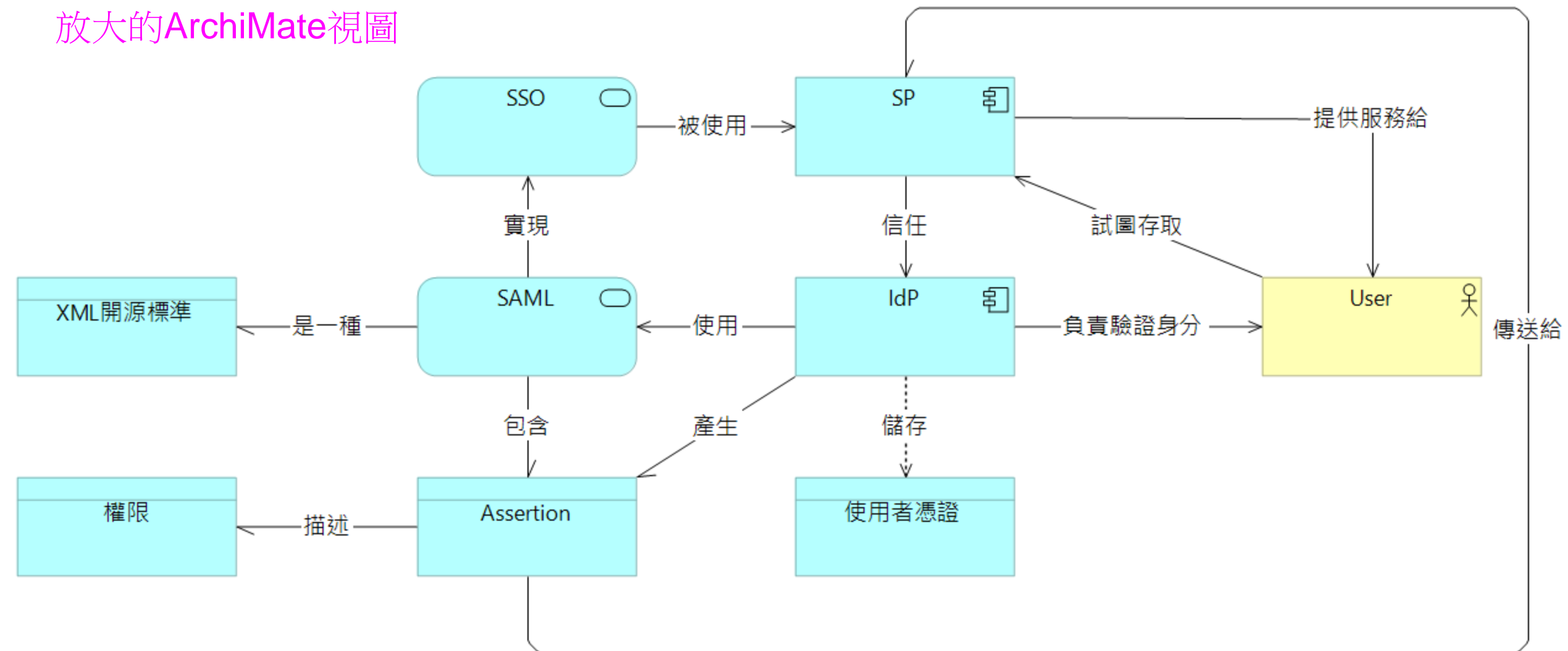
```

text
1. IdP --[負責驗證身分]--> User
2. IdP --[儲存]--> 使用者憑證
3. IdP --[使用]--> SAML
4. SAML --[實現]--> SSO
5. SSO --[被使用]--> SP
6. SP --[提供服務給]--> User
7. User --[嘗試存取]--> SP
  
```

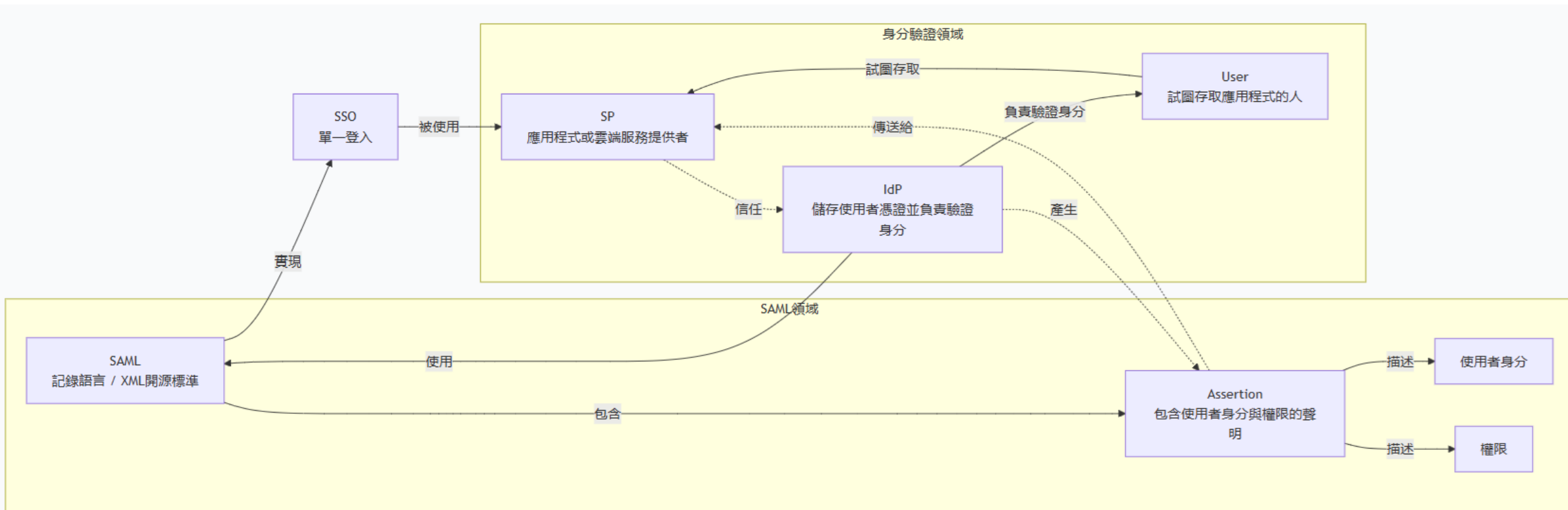
我依它的正規化程式, 重新繪的ArchiMate視圖

SNA Applications by AI Agent DeepSeek

放大的ArchiMate視圖



它依它的正規化程式, 重新繪的語意網路視圖(非ArchiMate視圖)



我要讓它把ArchiMate視圖轉成Skill定義檔, 即Markdown Skill File



請將此圖的語意, 轉成Skill



您希望我將前面那張 **SAML + SSO 語意網路圖**的「語意內容」, 轉換成一個可重複使用的 **Skill 定義檔**。

以下我將該圖中所有語意知識 (節點、關係、層級、對應規則) 封裝為一個完整的 **Markdown Skill 檔案**。

Skill 名稱

它給Skill命名, 以便我們日後識別

```
saml-ss0-semantic-network-analyzer
```

技能描述

此技能專門解析 **SAML (Security Assertion Markup Language)** 與單一登入 (**SSO**) 相關的語意網路圖 (Semantic Network via ArchiMate) , 並能 :

1. 辨識圖中的語意節點與關係
2. 將節點對應到正確的 ArchiMate 元素類型
3. 將語意關係對應到 ArchiMate 標準關係
4. 輸出多種圖形程式碼 (Mermaid / PlantUML / Graphviz)

它說明了Skill的觸發條件

觸發條件

當使用者提到以下關鍵字時觸發：

- SAML、SSO、IdP、SP
- 語意網路、Semantic Network、ArchiMate
- 身分驗證流程、聲明 (Assertion)
- 單一登入架構圖

初學者最困難的在於節點不知要用哪一個
ArchiMate Viewpoint Element表達

一、節點定義與 ArchiMate 對應

原始節點	節點本意	ArchiMate 元素
IdP	儲存使用者憑證並負責驗證身分的系統	Application Component
User	試圖存取應用程式的人	Business Actor
SP	應用程式或雲端服務提供者	Application Component
SAML	記錄語言，XML 開源標準	Application Service + Technology Standard
SSO	單一登入 (Single Sign-On)	Application Service
Assertion	包含使用者身分與權限的聲明	Data Object
使用者身分	身分資訊	Data Object
權限	權限資訊	Data Object
XML 開源標準	XML 技術標準	Technology Standard

二、語意關係與 ArchiMate 對應

初學者最困難的在於節點不知要用哪一個 ArchiMate Viewpoint Relation 表達

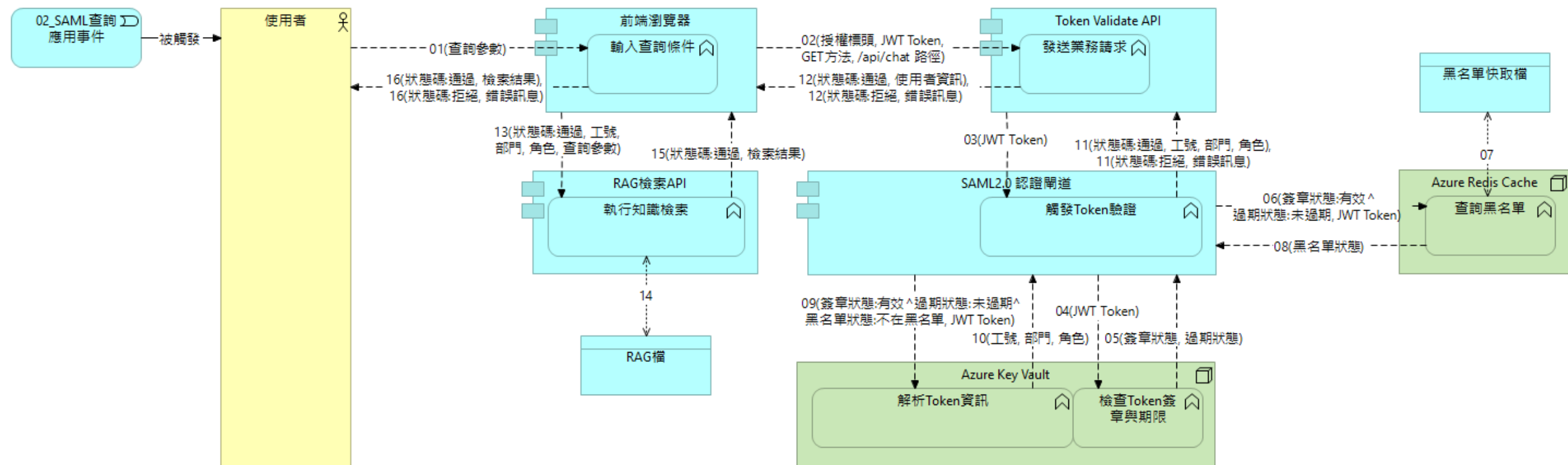
語意關係	ArchiMate 關係	關係符號	方向
負責驗證身分	Assignment	實線實心箭頭	IdP → User
使用	Serving	虛線實心箭頭	IdP → SAML
實現	Realization	實線虛尾箭頭	SAML → SSO
被使用	Serving	虛線實心箭頭	SSO → SP
試圖存取	Assignment	實線實心箭頭	User → SP
包含	Composition	實線空心菱型	Assertion → 使用者身分 / 權限
產生	Realization	實線虛尾箭頭	IdP → Assertion
傳送給	Flow	虛線開放箭頭	Assertion → SP
信任	Assignment	實線實心箭頭	SP → IdP
是一種	Specialization	空心三角箭頭	SAML → XML 開源標準

版本資訊

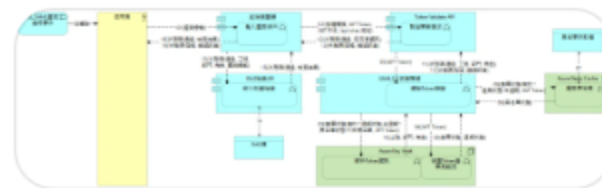
- Skill 名稱：`saml-ss0-semantic-network-analyzer`
- 版本：1.0
- 對應 ArchiMate 版本：3.2
- 建立日期：2026-06-03
- 適用範圍：SAML、SSO、身分驗證、企業架構語意網路

Examples for IFD+SNA: SAML查詢應用事件

這個例子是IFD+SNA



它沒學過, 也能解析



請用繁體中文解析此視圖的語意

這張圖是一個系統流程或API呼叫順序的視覺化表示，主要描述一個具備身分驗證 (SAML 2.0) 與授權 (JWT Token) 機制的RAG檢索API呼叫流程。以下是詳細語意解析：

一、整體目的

使用者透過前端瀏覽器輸入查詢條件，後端經過SAML驗證、JWT解析、黑名單檢查、權限取回後，決定是否允許呼叫 RAG 檢索 API，並回傳檢索結果。

02 SAML查詢 (圖左側區域)

依圖自我解析

- 使用者 → 前端瀏覽器 → 觸發 02_SAML查詢
- 查詢參數 (01) 被送出
- 可能結果：
 - 狀態碼16：通過 → 取得檢索結果
 - 狀態碼16：拒絕 + 錯誤碼0 → 驗證失敗

- 可能結果：
 - 狀態碼12：通過 → 可使用查詢
 - 狀態碼12：拒絕 + 錯誤碼0 → 失敗

步驟 11 (Token Validate API)

- 發送參數請求
- 回傳：
 - 通過 → 取得工單、部門、角色
 - 拒絕 + 錯誤碼0 → 失敗

中間流程：JWT Token 驗證與授權

步驟 03

- 帶有 JWT Token 使用 GET /api/chat

下方驗證與快取機制

04 JWT Token

- 送往 **Azure Key Vault** → 解析Token憑證
- 送往 **Azure Redis Cache** → 查詢黑名單

07、08

- 取得黑名單狀態

10

- 取得：工單、部門、角色

05

- 取得：簽章狀態、過期狀態

右側 RAG 檢索流程

- 前端瀏覽器輸入查詢條件
- 送給 **RAG 檢索 API** (步驟14)
- 執行知識檢索
- 透過 **RAG** 回傳結果

依原圖自我解析語意

總結語意

這是一個企業級安全檢索系統：

使用者透過瀏覽器查詢 → SAML驗證身分 → 取得JWT Token → 經Azure Key Vault解析與Redis

Cache黑名單檢查 → 確認權限 (工單、部門、角色) 與Token有效性 (簽章、過期) → 最後允許呼叫

RAG API執行知識檢索並回傳結果。

圖中的數字 (01, 03, 05, 07, 08, 10, 11, 12, 14, 16) 應為流程順序或狀態碼標記，並標示了通過 / 拒絕的判斷分支。

開始餵它IFD+SNA Markdown Skill File



deepseek_markdown_2...
MD 551B

此為IFD的MD檔, 請用此Skill 解析上面的視圖

根據您提供的 **IFD+SNA 繪圖規則 Skill**，我將把前面那張圖（SAML + JWT + RAG 檢索流程）轉換為 **IFD+SNA 格式**的互動流程圖語意解析。

IFD+SNA 視圖解析 (繁體中文)

圖的名稱

SAML 驗證與 RAG 知識檢索互動流程圖

參與者 (Lifelines) 依IFD+SNA Markdown Skill File的規定精準找出我要的Participants, 做成它所謂的Lifelines(生命線)

角色	說明
使用者	發起查詢的終端使用者
前端瀏覽器	使用者操作介面，發送 HTTP 請求
SAML 查詢模組	處理 SAML 2.0 身分驗證
Token Validate API	驗證 JWT Token 的有效性
Azure Key Vault	解析 Token 憑證 (簽章、過期)
Azure Redis Cache	查詢黑名單狀態
RAG 檢索 API	執行知識檢索的核心 API
RAG 引擎	實際進行檢索的後端服務

流程步驟 (對應圖中數字)

01 使用者輸入查詢條件

- 呼叫者：使用者
- 接收者：前端瀏覽器
- 描述：輸入查詢參數

02 SAML 查詢 (被跟蹤)

- 呼叫者：前端瀏覽器
- 接收者：SAML 查詢模組
- 描述：進行 SAML 2.0 身分驗證
- 分支結果：
 - 狀態碼 16 (通過) → 取得檢索結果
 - 狀態碼 16 (拒絕，錯誤碼 0) → 終止流程

定義每對
Caller, Callee

03 JWT Token 請求

- 呼叫者：前端瀏覽器
- 接收者：RAG 檢索 API
- 路徑/方法：GET /api/chat
- 描述：帶 JWT Token 呼叫 API
- 分支結果：
 - 狀態碼 12 (通過) → 可使用查詢
 - 狀態碼 12 (拒絕，錯誤碼 0) → 終止流程

04 發送 JWT Token 進行解析

- 呼叫者：RAG 檢索 API
- 接收者：Azure Key Vault
- 描述：解析 Token 憑證 (簽章、過期狀態)

05 回傳簽章與過期狀態

- 呼叫者：Azure Key Vault
- 接收者：RAG 檢索 API
- 描述：回傳 Token 有效性判斷

06 查詢黑名單

- 呼叫者：RAG 檢索 API
- 接收者：Azure Redis Cache
- 描述：查詢該 Token 是否在黑名單中

07 回傳黑名單狀態

- 呼叫者：Azure Redis Cache
- 接收者：RAG 檢索 API
- 描述：回傳是否在黑名單

08 判斷黑名單狀態

- 邏輯分支：
 - 不在黑名單 → 繼續流程
 - 在黑名單 → 拒絕請求

09 請求權限資訊 (工單、部門、角色)

- 呼叫者：RAG 檢索 API
- 接收者：Token Validate API
- 描述：傳送參數請求取得使用者屬性

10 回傳工單、部門、角色

- 呼叫者：Token Validate API
- 接收者：RAG 檢索 API
- 描述：回傳授權相關資訊

11 驗證 Token (狀態碼)

- 呼叫者：Token Validate API
- 接收者：RAG 檢索 API
- 描述：回傳驗證結果
- 分支結果：
 - 狀態碼 11 (通過) → 取得工單、部門、角色
 - 狀態碼 11 (拒絕，錯誤碼 0) → 終止流程

12 最終 API 授權判斷

- 邏輯分支：
 - 通過 → 執行 RAG 檢索
 - 拒絕 → 回傳錯誤

14 執行知識檢索

- 呼叫者：RAG 檢索 API
- 接收者：RAG 引擎
- 描述：執行知識庫檢索

16 回傳檢索結果

- 呼叫者：RAG 引擎
- 接收者：前端瀏覽器 (經由 RAG 檢索 API)
- 描述：回傳最終查詢結果給使用者

定義觸發順序, 類似SBC中的結構行為合一(SBC)圖

IFD+SNA 結構總結

text

```
使用者 → 前端瀏覽器 → SAML 查詢(02) → [通過/拒絕]
      ↓
      RAG 檢索 API(03) → [通過/拒絕]
      ↓
      Azure Key Vault(04) → 回傳簽章狀態(05)
      ↓
      Azure Redis Cache(06) → 回傳黑名單(07)
      ↓
      Token Validate API(09) → 回傳權限(10)
      ↓
      驗證狀態碼(11) → [通過/拒絕]
      ↓
      RAG 引擎(14) → 回傳結果(16)
```

平行執行與巢狀結構說明

- 步驟 04 與 06：可視為對不同元件的平行呼叫（解析憑證 + 查黑名單）
- 步驟 11 包含 步驟 04、06、09 的綜合判斷結果
- 步驟 03 與 11 屬於巢狀驗證邏輯：先驗 JWT 格式，再驗證內部權限

如果您需要我將此解析轉為 實際的 IFD+SNA 程式碼（例如 Mermaid 或 PlantUML 格式），也可以直接告訴我。

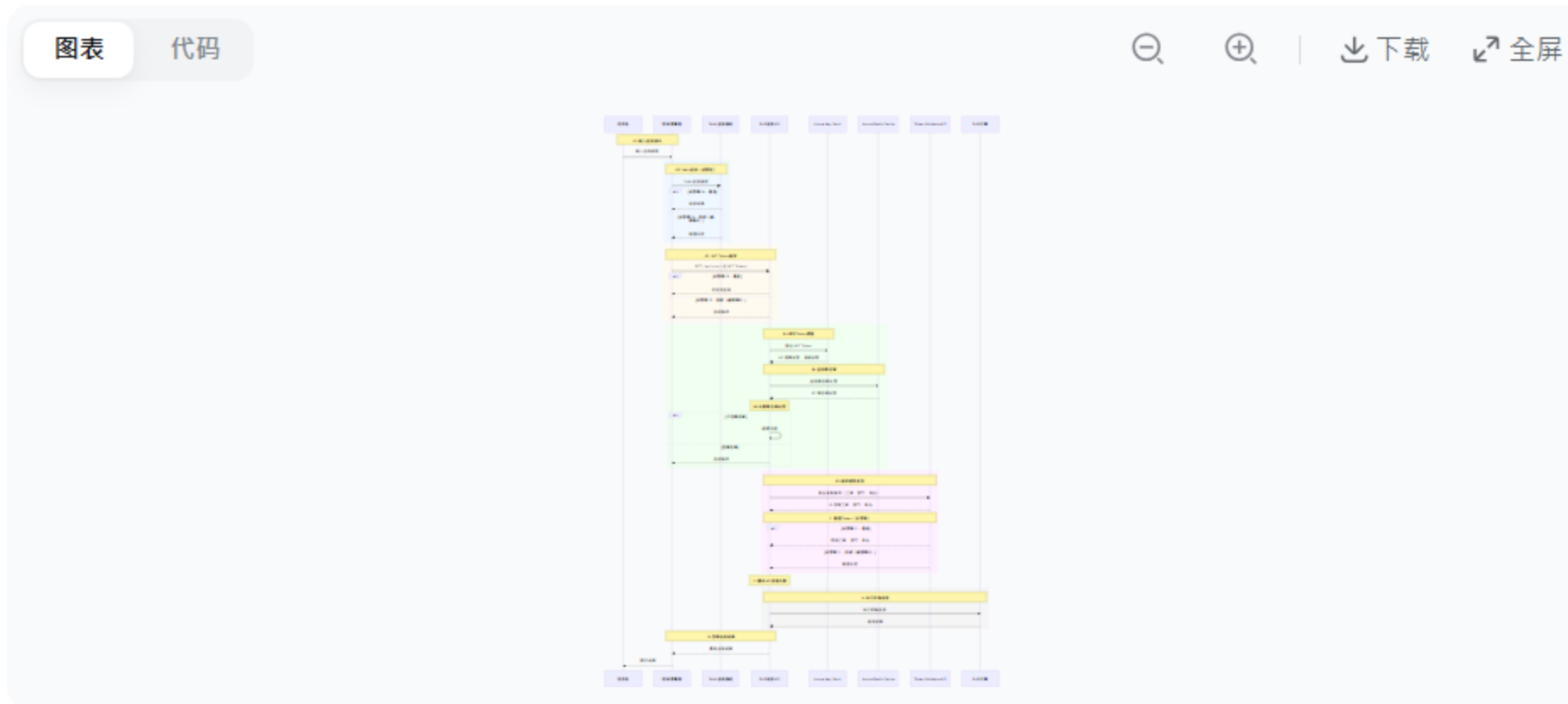
希望反向得到IFD+SNA程式碼, 但它自作主張轉成類似的Mermaid語法, 原因是它不會繪IFD+SNA視圖, 但會繪Mermaid視圖

請解析轉為 實際的 IFD+SNA 程式碼

以下根據您提供的 IFD+SNA 繪圖規則，將前述 SAML + JWT + RAG 流程轉為 Mermaid 語法（最接近 IFD+SNA 互動流程圖的實際程式碼格式）。

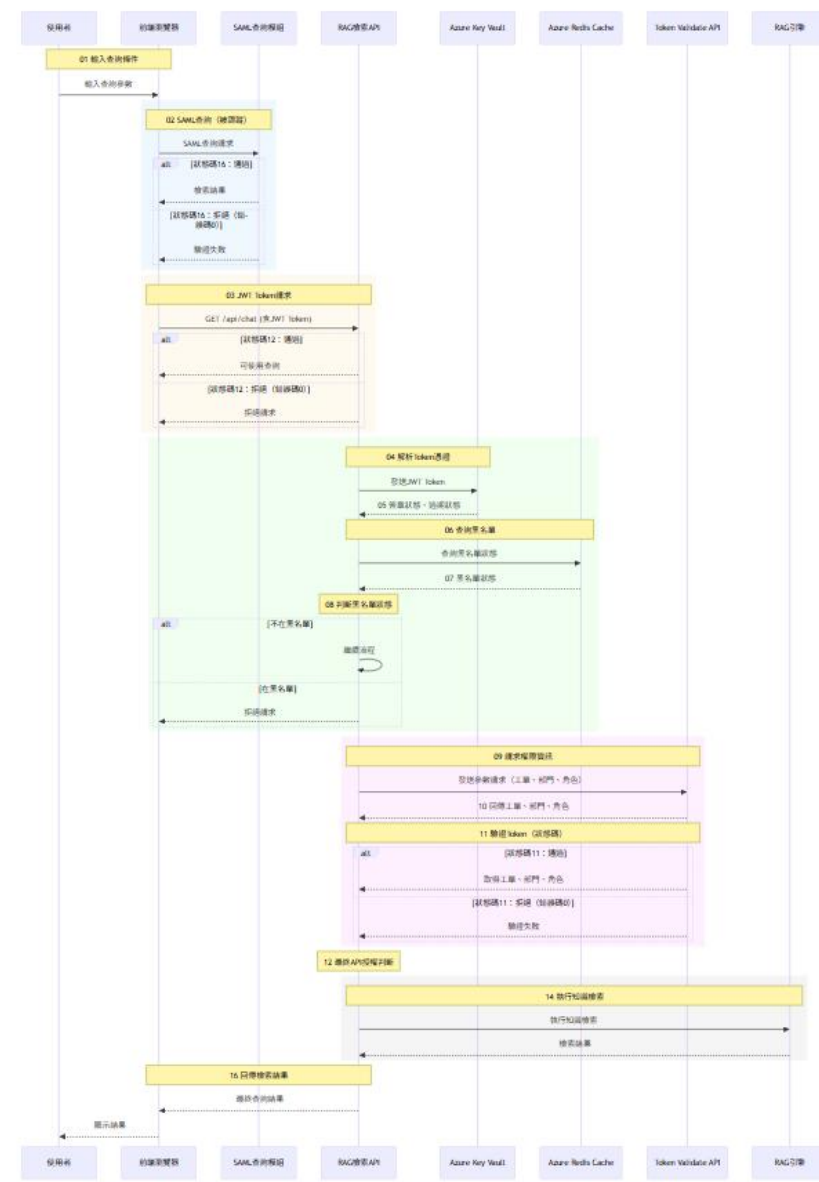
繪出的Mermaid視圖

IFD+SNA 程式碼 (Mermaid 格式)



IFD+SNA Applications by AI Agent DeepSeek

放大的Mermaid視圖



繪出的Mermaid視圖程式碼

```
sequenceDiagram
    participant User as 使用者
    participant Browser as 前端瀏覽器
    participant SAML as SAML查詢模組
    participant RAG_API as RAG檢索API
    participant KeyVault as Azure Key Vault
    participant Redis as Azure Redis Cache
    participant TokenAPI as Token Validate API
    participant RAG_Engine as RAG引擎

    Note over User,Browser: 01 輸入查詢條件
    User->>Browser: 輸入查詢參數
```

每對巢狀呼叫

```
Note over User,Browser: 01 輸入查詢條件
User->>Browser: 輸入查詢參數

rect rgb(240, 248, 255)
    Note over Browser,SAML: 02 SAML查詢 (被跟蹤)
    Browser->>SAML: SAML查詢請求
    alt 狀態碼16: 通過
        SAML-->>Browser: 檢索結果
    else 狀態碼16: 拒絕 (錯誤碼0)
        SAML-->>Browser: 驗證失敗
    end
end
```

每對巢狀呼叫

```
rect rgb(255, 250, 240)
  Note over Browser,RAG_API: 03 JWT Token請求
  Browser->>RAG_API: GET /api/chat (含JWT Token)
  alt 狀態碼12: 通過
    RAG_API-->>Browser: 可使用查詢
  else 狀態碼12: 拒絕 (錯誤碼0)
    RAG_API-->>Browser: 拒絕請求
  end
end
```

```
rect rgb(240, 255, 240)
  Note over RAG_API,KeyVault: 04 解析Token憑證
  RAG_API->>KeyVault: 發送JWT Token
  KeyVault-->>RAG_API: 05 簽章狀態、過期狀態

  Note over RAG_API,Redis: 06 查詢黑名單
  RAG_API->>Redis: 查詢黑名單狀態
  Redis-->>RAG_API: 07 黑名單狀態

  Note over RAG_API: 08 判斷黑名單狀態
  alt 不在黑名單
    RAG_API->>RAG_API: 繼續流程
  else 在黑名單
    RAG_API-->>Browser: 拒絕請求
  end
end
```

每對巢狀呼叫

```
rect rgb(255, 240, 255)
  Note over RAG_API,TokenAPI: 09 請求權限資訊
  RAG_API->>TokenAPI: 發送參數請求 (工單、部門、角色)

  TokenAPI-->>RAG_API: 10 回傳工單、部門、角色

  Note over TokenAPI,RAG_API: 11 驗證Token (狀態碼)
  alt 狀態碼11: 通過
    TokenAPI-->>RAG_API: 取得工單、部門、角色
  else 狀態碼11: 拒絕 (錯誤碼0)
    TokenAPI-->>RAG_API: 驗證失敗
  end
end
```

```
Note over RAG_API: 12 最終API授權判斷

rect rgb(245, 245, 245)
  Note over RAG_API,RAG_Engine: 14 執行知識檢索
  RAG_API->>RAG_Engine: 執行知識檢索
  RAG_Engine-->>RAG_API: 檢索結果
end

Note over RAG_API,Browser: 16 回傳檢索結果
RAG_API-->>Browser: 最終查詢結果
Browser-->>User: 顯示結果
```

程式碼符號說明

程式碼說明

元素	說明
<code>participant</code>	定義流程中的參與者 (生命線)
<code>->></code>	同步呼叫 (請求)
<code>-->></code>	回覆 (回應)
<code>Note over</code>	標註步驟編號與說明
<code>rect rgb(...)</code>	將相關步驟分組 (模組化)
<code>alt</code>	IFD+SNA 中的邏輯分支 (通過/拒絕)

觸發序號說明

對應原始圖中的數字與狀態碼

圖中數字	說明	圖中數字	說明
01	輸入查詢條件	09	請求權限資訊
02	SAML查詢	10	回傳工單、部門、角色
03	JWT Token請求	11	驗證Token狀態碼
04	解析Token憑證	12	最終API授權判斷
05	回傳簽章/過期狀態	14	執行知識檢索
06	查詢黑名單	16	回傳檢索結果
07	回傳黑名單狀態	狀態碼11	通過/拒絕
08	判斷黑名單狀態	狀態碼12、16	通過/拒絕

叫它用我的原始程式碼
修正它的Mermain
格式視圖

這是原始的IFD+SNA程式碼, 請修正您的IFD+SNA 程式碼 (Mermaid 格式) 圖: 01 使用者 -> 前端瀏覽器.輸入查詢條件: 01(查詢參數)

02 前端瀏覽器 -> Token Validate API.發送業務請求: 02(授權標頭, JWT Token, GET方法, /api/chat 路徑)

03 Token Validate API -> SAML2.0 認證閘道.觸發Token驗證: 03(JWT Token)

04 SAML2.0 認證閘道 -> Azure Key Vault.檢查Token簽章與期限: 04(JWT Token)

05 SAML2.0 認證閘道 <- Azure Key Vault.檢查Token簽章與期限: 05(簽章狀態, 過期狀態)

若 :(簽章狀態:有效^過期狀態:未過期), 則繼續流程 06

若 :(簽章狀態:無效v過期狀態:已過期), 則跳至流程 12

06 SAML2.0 認證閘道 -> Azure Redis Cache.查詢黑名單: 06(簽章狀態:有效^過期狀態:未過期, JWT Token)

07 Azure Redis Cache.查詢黑名單 <-> 黑名單快取檔: 07 # 雙向存取

08 SAML2.0 認證閘道 <- Azure Redis Cache.查詢黑名單: 08(黑名單狀態)

若 :(黑名單狀態:不在黑名單), 則繼續流程 09

若 :(黑名單狀態:在黑名單), 則跳至流程 12

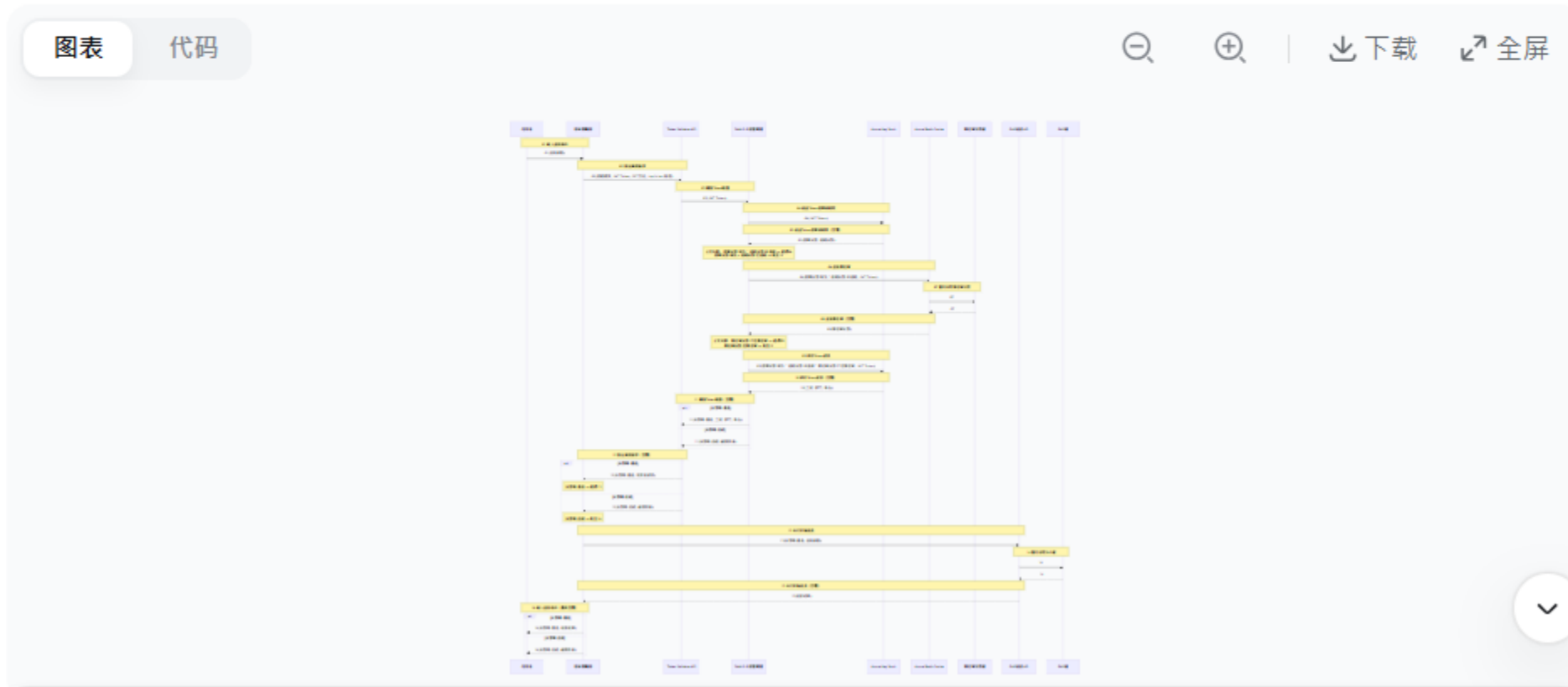
- 09 SAML2.0 認證閘道 -> Azure Key Vault.解析Token資訊: 09(簽章狀態:有效^過期狀態:未過期^黑名單狀態:不在黑名單, JWT Token)
- 10 SAML2.0 認證閘道 <- Azure Key Vault.解析Token資訊: 10(工號, 部門, 角色)
- 11 Token Validate API <- SAML2.0 認證閘道.觸發Token驗證: 11(狀態碼:通過, 工號, 部門, 角色), 11(狀態碼:拒絕, 錯誤訊息)
- 12 前端瀏覽器 <- Token Validate API.發送業務請求: 12(狀態碼:通過, 使用者資訊), 12(狀態碼:拒絕, 錯誤訊息)
- # 若 :(狀態碼:通過), 則繼續流程 13
 - # 若 :(狀態碼:拒絕), 則跳至流程 16
- 13 前端瀏覽器 -> RAG檢索API.執行知識檢索: 13(狀態碼:通過, 查詢參數)
- 14 RAG檢索API.執行知識檢索 <-> RAG檔: 14 # 雙向存取
- 15 前端瀏覽器 <- RAG檢索API.執行知識檢索: 15(檢索結果)
- 16 前端瀏覽器 -> 使用者.輸入查詢條件: 16(狀態碼:通過, 檢索結果), 16(狀態碼:拒絕, 錯誤訊息)

IFD+SNA Applications by AI Agent DeepSeek

感謝您提供完整的原始 IFD+SNA 程式碼。以下根據該原始碼，修正為正確的 **Mermaid** 格式。

修正後，視圖就正確了

修正後的 IFD+SNA 程式碼 (Mermaid 格式)



這是修正後Mermaid格式視圖的程式碼, 也就是說: 用文本, 藉Mermaid, 繪出IFD視圖

```
sequenceDiagram
    participant User as 使用者
    participant Browser as 前端瀏覽器
    participant TokenAPI as Token Validate API
    participant SAML as SAML2.0 認證閘道
    participant KeyVault as Azure Key Vault
    participant Redis as Azure Redis Cache
    participant BlacklistFile as 黑名單快取檔
    participant RAG as RAG檢索API
    participant RAGFile as RAG檔
```

Note over User,Browser: 01 輸入查詢條件

User->>Browser: 01(查詢參數)

Note over Browser,TokenAPI: 02 發送業務請求

Browser->>TokenAPI: 02(授權標頭, JWT Token, GET方法, /api/chat 路徑)

Note over TokenAPI,SAML: 03 觸發Token驗證

TokenAPI->>SAML: 03(JWT Token)

Note over SAML,KeyVault: 04 檢查Token簽章與期限

SAML->>KeyVault: 04(JWT Token)

Note over SAML,KeyVault: 05 檢查Token簽章與期限 (回覆)

KeyVault-->>SAML: 05(簽章狀態, 過期狀態)

Note over SAML: 分支判斷: 簽章狀態:有效 ^ 過期狀態:未過期 → 繼續06
簽章狀態:無效 v 過期狀態:已過期
→ 跳至12

Note over SAML,Redis: 06 查詢黑名單

SAML->>Redis: 06(簽章狀態:有效 ^ 過期狀態:未過期, JWT Token)

Note over Redis,BlacklistFile: 07 雙向存取黑名單快取

Redis->>BlacklistFile: 07

BlacklistFile-->>Redis: 07

Note over SAML,Redis: 08 查詢黑名單 (回覆)

Redis-->>SAML: 08(黑名單狀態)

Note over SAML: 分支判斷: 黑名單狀態:不在黑名單 → 繼續09
黑名單狀態:在黑名單 → 跳至12

Note over SAML,KeyVault: 09 解析Token資訊

SAML->>KeyVault: 09(簽章狀態:有效 ^ 過期狀態:未過期 ^ 黑名單狀態:不在黑名單, JWT Token)

Note over SAML,KeyVault: 10 解析Token資訊 (回覆)

KeyVault-->>SAML: 10(工號, 部門, 角色)

IFD+SNA Applications by AI Agent DeepSeek

Note over SAML,KeyVault: 10 解析Token資訊 (回覆)

KeyVault-->>SAML: 10(工號, 部門, 角色)

Note over TokenAPI,SAML: 11 觸發Token驗證 (回覆)

alt 狀態碼:通過

SAML-->>TokenAPI: 11(狀態碼:通過, 工號, 部門, 角色)

else 狀態碼:拒絕

SAML-->>TokenAPI: 11(狀態碼:拒絕, 錯誤訊息)

end

Note over Browser,TokenAPI: 12 發送業務請求 (回覆)

alt 狀態碼:通過

TokenAPI-->>Browser: 12(狀態碼:通過, 使用者資訊)

Note over Browser: 狀態碼:通過 → 繼續13

else 狀態碼:拒絕

TokenAPI-->>Browser: 12(狀態碼:拒絕, 錯誤訊息)

Note over Browser: 狀態碼:拒絕 → 跳至16

end

Note over Browser,RAG: 13 執行知識檢索

Browser-->>RAG: 13(狀態碼:通過, 查詢參數)

Note over RAG,RAGFile: 14 雙向存取RAG檔

RAG-->>RAGFile: 14

RAGFile-->>RAG: 14

Note over Browser,RAG: 15 執行知識檢索 (回覆)

RAG-->>Browser: 15(檢索結果)

Note over User,Browser: 16 輸入查詢條件 (最終回覆)

alt 狀態碼:通過

Browser-->>User: 16(狀態碼:通過, 檢索結果)

else 狀態碼:拒絕

Browser-->>User: 16(狀態碼:拒絕, 錯誤訊息)

end

IFD+SNA Applications by AI Agent DeepSeek

原始的IFD, 加上了觸發序號, 參數列

原始編號	說明	Mermaid 表現
01	使用者輸入查詢條件	User->>Browser: 01(查詢參數)
02	前端瀏覽器發送業務請求	Browser->>TokenAPI: 02(...)
03	觸發Token驗證	TokenAPI->>SAML: 03(JWT Token)
04	檢查Token簽章與期限 (請求)	SAML->>KeyVault: 04(JWT Token)
05	檢查Token簽章與期限 (回覆)	KeyVault-->>SAML: 05(簽章狀態, 過期狀態)
06	查詢黑名單 (請求)	SAML->>Redis: 06(...)
07	雙向存取黑名單快取檔	Redis<->BlacklistFile
08	查詢黑名單 (回覆)	Redis-->>SAML: 08(黑名單狀態)

IFD+SNA Applications by AI Agent DeepSeek

09	解析Token資訊 (請求)	SAML->>KeyVault: 09(...)
10	解析Token資訊 (回覆)	KeyVault-->>SAML: 10(工號, 部門, 角色)
11	觸發Token驗證 (回覆)	alt 區塊 (通過/拒絕)
12	發送業務請求 (回覆)	alt 區塊 (通過/拒絕) + 分支判斷
13	執行知識檢索 (請求)	Browser->>RAG: 13(...)
14	雙向存取RAG檔	RAG<->RAGFile
15	執行知識檢索 (回覆)	RAG-->>Browser: 15(檢索結果)
16	最終回覆使用者	alt 區塊 (通過/拒絕)

關鍵修正說明

修正前後的比較的說明

原始碼特性

Mermaid 實作方式

巢狀縮排 (02→03→04)

無法縮排，但透過參與者順序與 `Note` 說明層級關係

雙向存取 (07、14)

使用 `->>` 與 `-->>` 一來一回表示

邏輯分支 (若...則...)

使用 `Note over` 說明條件，並用 `alt` 區塊表示不同結果

跳至流程 (12、16)

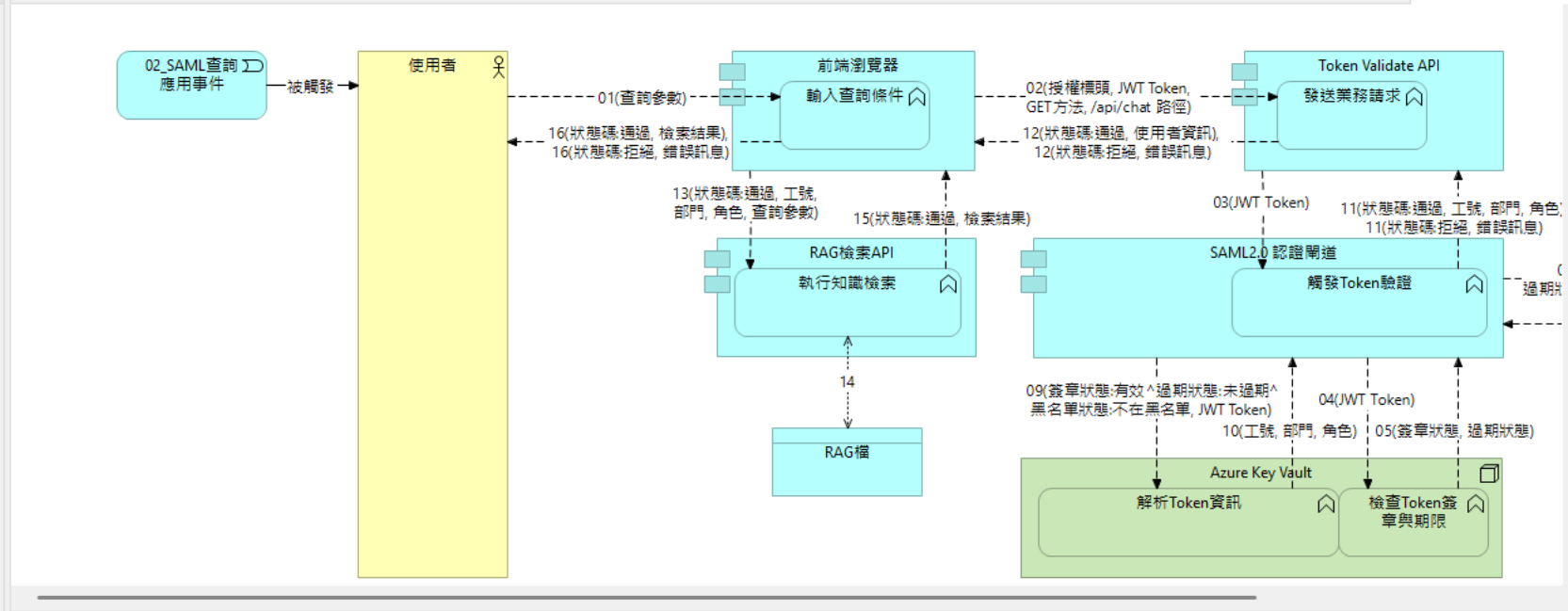
以 `alt` 區塊中的不同路徑表達

`^` (且) 與 `v` (或)

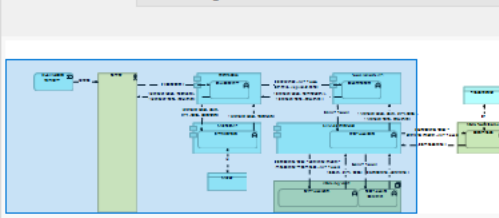
在 `Note over` 中以文字說明

如果您需要 PlantUML 版本或其他調整，請告訴我。

- Application
- Technology & Physical
- Motivation
- Implementation & Migration
- Other
- Relations
- Views
 - 01_ABB_SBB_Continuum
 - 01_Relationship between ABB_SBB
 - 02_SBB 的規格內容
 - 03_ABB and SBB View Content
 - 04_Examples: ABB and SBB
 - 05_Application Event Example: SAML
 - 06_IFD Examples: 01_SAML登入應用專
 - 07_IFD Examples: 02_SAML查詢應用專
 - 02_Generative AI Applications
 - 01_myGuru+TAG 智能護欄
 - 02_業務員智能對練 AI
 - 03_Call Center Smart System(CCSS)
 - 04_業務員 AgentPortal 2.0 知識模塊
 - 05_業務員教育訓練暨考評量系統
 - 06_南山黃金眼 AI 防詐
 - 07 AI 交互中斷系統



- Palette
- Magic Connector
 - Composition relation
 - Aggregation relation
 - Assignment relation
 - Realization relation
 - Serving relation
 - Access relation
 - Influence relation
 - Triggering relation
 - Flow relation
 - Specialization relation
 - Association relation
 - Junction
 - Note
 - Group
 - Connection



07_IFD Examples: 02_SAML查詢應用事件

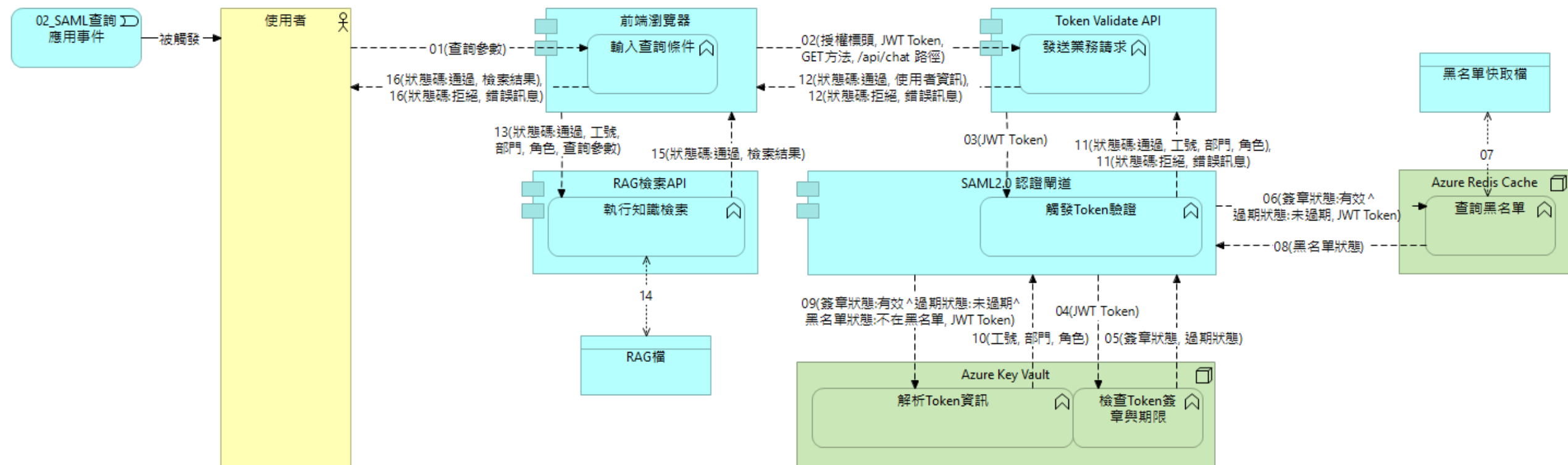
```

Main Viewpoint: None
Properties Name: 07_IFD Examples: 02_SAML查詢應用事件
Appearance Documentation:
01 使用者 -> 前端瀏覽器.輸入查詢條件: 01(查詢參數)
02 前端瀏覽器 -> Token Validate API.發送業務請求: 02(授權標頭, JWT Token, GET方法, /api/chat 路徑)
03 Token Validate API -> SAML2.0 認證閘道.觸發Token驗證: 03(JWT Token)
04 SAML2.0 認證閘道 -> Azure Key Vault.檢查Token簽章與期限: 04(JWT Token)
05 SAML2.0 認證閘道 <- Azure Key Vault.檢查Token簽章與期限: 05(簽章狀態, 過期狀態)
# 若:(簽章狀態:有效^過期狀態:未過期),則繼續流程 06
# 若:(簽章狀態:無效v過期狀態:已過期),則跳至流程 12
06 SAML2.0 認證閘道 -> Azure Redis Cache.查詢黑名單: 06(簽章狀態:有效^過期狀態:未過期, JWT Token)
07 Azure Redis Cache.查詢黑名單 <-> 黑名單快取檔: 07 # 雙向存取
08 SAML2.0 認證閘道 <- Azure Redis Cache.查詢黑名單: 08(黑名單狀態)
# 若:(黑名單狀態:不在黑名單),則繼續流程 09
  
```

此為IFD+SNA的程式碼, 可用此碼, 以人工方式, 繪製 IFD+SNA視圖

IFD+SNA Applications by AI Agent DeepSeek

放大後的IFD+SNA視圖



IFD+SNA程式碼

產生IFD+SNA程式碼的Skill Markdown File要先餵給它, 它就會產生如下標準的巢狀格式程式碼

- 01 使用者 -> 前端瀏覽器.輸入查詢條件: 01(查詢參數)
- 02 前端瀏覽器 -> Token Validate API.發送業務請求: 02(授權標頭, JWT Token, GET方法, /api/chat 路徑)
- 03 Token Validate API -> SAML2.0 認證閘道.觸發Token驗證: 03(JWT Token)
- 04 SAML2.0 認證閘道 -> Azure Key Vault.檢查Token簽章與期限: 04(JWT Token)
- 05 SAML2.0 認證閘道 <- Azure Key Vault.檢查Token簽章與期限: 05(簽章狀態, 過期狀態)
若 :(簽章狀態:有效^過期狀態:未過期), 則繼續流程 06
若 :(簽章狀態:無效v過期狀態:已過期), 則跳至流程 12
- 06 SAML2.0 認證閘道 -> Azure Redis Cache.查詢黑名單: 06(簽章狀態:有效^過期狀態:未過期, JWT Token)
- 07 Azure Redis Cache.查詢黑名單 <-> 黑名單快取檔: 07 # 雙向存取
- 08 SAML2.0 認證閘道 <- Azure Redis Cache.查詢黑名單: 08(黑名單狀態)
若 :(黑名單狀態:不在黑名單), 則繼續流程 09
若 :(黑名單狀態:在黑名單), 則跳至流程 12 # 為註解

- 09 SAML2.0 認證閘道 -> Azure Key Vault.解析Token資訊: 09(簽章狀態:有效^過期狀態:未過期^黑名單狀態:不在黑名單, JWT Token)
- 10 SAML2.0 認證閘道 <- Azure Key Vault.解析Token資訊: 10(工號, 部門, 角色)
- 11 Token Validate API <- SAML2.0 認證閘道.觸發Token驗證: 11(狀態碼:通過, 工號, 部門, 角色), 11(狀態碼:拒絕, 錯誤訊息)
- 12 前端瀏覽器 <- Token Validate API.發送業務請求: 12(狀態碼:通過, 使用者資訊), 12(狀態碼:拒絕, 錯誤訊息)
 - # 若:(狀態碼:通過), 則繼續流程 13
 - # 若:(狀態碼:拒絕), 則跳至流程 16
- 13 前端瀏覽器 -> RAG檢索API.執行知識檢索: 13(狀態碼:通過, 查詢參數)
- 14 RAG檢索API.執行知識檢索 <-> RAG檔: 14 # 雙向存取
- 15 前端瀏覽器 <- RAG檢索API.執行知識檢索: 15(檢索結果)
- 16 前端瀏覽器 -> 使用者.輸入查詢條件: 16(狀態碼:通過, 檢索結果), 16(狀態碼:拒絕, 錯誤訊息)

結論: 以後, 只要告訴AI Agent我們要繪的內容, 它就會產生這種程式碼, 我們再根據它, 來繪IFD+SNA視圖, 會節省很多時間, 而且精準到位

More information and articles on a|CEA at:
<https://www.aeataiwan.org/>

Thank you

Dr. Meng-chyi Harn 韓孟麒

0919-297-147

harn@takming.edu.tw

LINE ID: 0919297147

Ph.D in Computer Science

U.S. Naval Postgraduate School



LINE ID 個人帳號
QR Code 掃碼



中華企業架構師學會
QR Code 掃碼