開放標準與無密碼時代

陳佑任

中華民國 112 年 6 月

作者任職於中央銀行資訊處，本文內容純屬個人意見，與服務單位無關，如有錯誤亦概由作者負責。

摘　要

現今資訊服務生態與設備裝置日益多元，各種基於雲端及網路的服務亦深入生活。在各種科技讓日常生活更為便利的另一面，是使用者在使用不同設備、不同服務時，需要記憶不同帳號、不同密碼。當使用者對密碼的依賴逐漸加深，除密碼帶來的不便，也伴隨資訊安全的風險。為解決便利與安全兩大課題，各種開放標準應運而生，嘗試降低不同應用服務場景對密碼的依賴。

本文挑選 4 項推動無密碼驗證與授權生態體系的主流開放標準：OAuth、OpenID、SAML、FIDO，介紹這些開放標準的主要角色、機制與流程。接著透過綜合歸納比較，分析這 4 項開放標準的關係並說明其共同構成的無密碼生態。

對於企業而言，上述開放標準帶來機會的同時，相應的技術快速變化與生態體系的複雜化也帶來挑戰。本文嘗試從企業觀點出發，彙整企業資訊架構與上述開放標準間的關係、分析企業資訊架構各面向對於不同開放標準導入的考量，並對企業身分驗證與授權系統導入開放標準的模式提供建議。

目　次

一、緒論

現今各種資訊服務日益豐富，設備裝置也日益多元，在各種科技讓日常生活更為便利的另一面，是使用者在使用不同設備、不同服務時，需要記憶不同帳號、不同密碼。同時各種網路攻擊與資安事件的頻傳，讓服務提供者與大眾的資安意識不斷提高，使用密碼的不便，及儲存與傳輸密碼帶來的風險乃成為亟需改善的課題。

在此背景下，各種開放標準(Open Standard)應運而生。這些開放標準側重的場景或有不同，但都希望可以透過共通的標準，在便利使用者跨服務存取資訊的同時，透過減少密碼的使用，提高服務存取的資訊安全。

這些標準的開放性質，一方面說明面對多平台、多服務商的資訊服務生態，難以期待有私有、封閉的標準能夠整合所有資訊服務，但同時開放標準的演進、競合關係也對採用者帶來與私有標準不同的挑戰。

本文將以企業的觀點出發，嘗試理解與分析這些開放標準與企業資訊架構間的關係，進而評估與規劃這些標準的導入。希望透過本文的介紹與分析，能夠建立讀者相關領域之概念地圖(Concept Map)，對於相關標準乃至於未來新的標準之理解、評估與決策提供幫助。

本文選擇 4 項與無密碼驗證授權相關的主流開放標準進行介紹及分析比較：OAuth、OpenID、SAML、FIDO。首先在第二章至第五章分別分析這些標準的主要角色(Role)、機制與流程。第六章綜合比較與分析各項標準的關係。第七章嘗試建構易於理解的概略性觀點(High-Level View)，以歸納各項標準與企業資訊架構的關係、分析企業資訊架構各面向對於不同標準導入的考量，以及針對身分驗證與授權系統設計提供建議。第八章為結論。

二、OAuth 標準

OAuth(意指 Open Authorization)社群與標準(最新版為 OAuth 2.0)的出現，源於人們對於安全性和便利性之間的平衡追求。在過去，當一個應用服務需要存取使用者的資源時，通常要求使用者分享其帳號和密碼。這種做法存在風險，因為一旦應用服務獲得使用者的帳號和密碼，它就可以自由存取用戶的資源，甚至可能惡意濫用使用者帳號。

OAuth 的目標是提供一種安全且標準化的方法，允許使用者授權第三方應用服務存取其擁有的資源，同時保護使用者的帳號和密碼不被直接共享，以解決前述問題。這讓使用者可以授權特定的權限給應用服務，而不必將其帳號和密碼交給應用服務。OAuth 標準在網際網路服務和 API 服務中得到廣泛應用，為使用者提供了更好的安全性和控制權。

OAuth 標準的機制是基於授權伺服器(Authorization Server)、資源伺服器(Resource Server)、使用者(User)和客戶端(Client)之間的互動。授權伺服器負責驗證使用者身分並授予存取權限，資源伺服器儲存並管理使用者的資源，使用者是資源的所有者，而客戶端則是希望存取使用者資源的第三方應用程序。

OAuth 2.0 原始標準針對不同場景定義了 4 個不同的授權流程：Authorization Code Grant(最常見，適用於客戶端屬於伺服器端應用程式)、Client Credentials Grant(適用於無使用者，客戶端要取得屬於自身的資源時)、Implicit Grant(適用於客戶端屬於前端應用程式)、Resource Owner Password Credentials Grant(密碼儲存於客戶端的流程) (Hardt, 2012)。其後，因應網站技術的演進及資安的發展，OAuth 發展了許多補充標準，而最新的安全最佳實務 OAuth 2.0 Security Best Current Practice 已經禁用 Implicit Grant 與 Resource Owner Password Credentials Grant。

這裡以一個社群媒體網站的應用場景為例，來說明最常見的 Authorization Code Grant 流程：使用者 Alice 使用一個社群媒體網站(客戶

端)，並希望將其使用的雲端硬碟服務(角色包含授權伺服器及資源伺服器)上的資源(例如照片)分享到社群媒體網站。社群媒體網站希望獲得 Alice 的授權，以存取並顯示她的照片。

在此場景中， Authorization Code Grant 流程包含以下步驟：

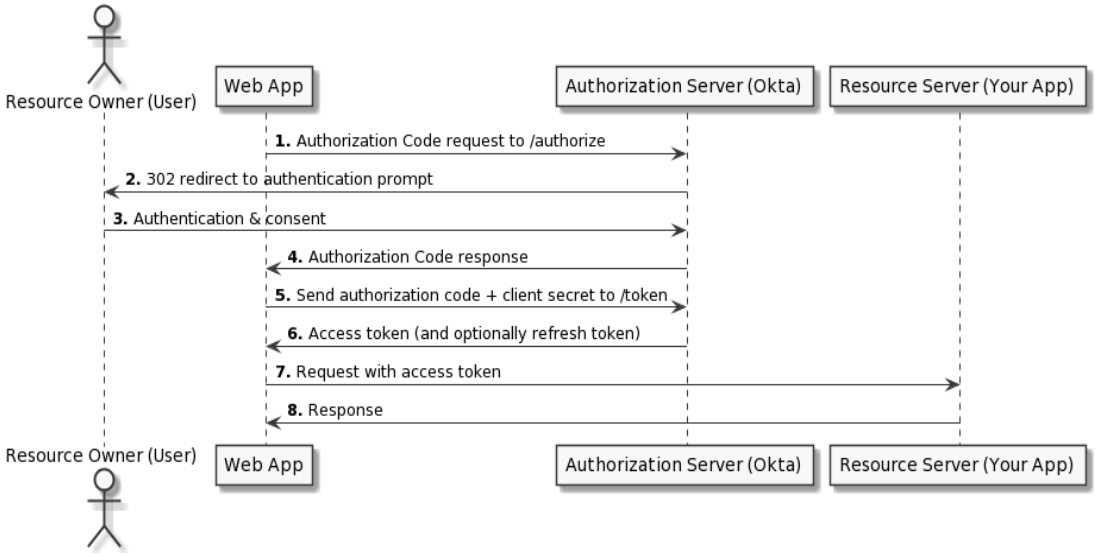

圖 1：OAuth 中的 Authorization Code Grant 流程圖
(來源：Okta Developer Guide)

1. 社群媒體網站(圖 1 中的 Web App)向雲端硬碟服務的授權伺服器(圖 1 中的 Authorization Server)發出授權請求，並指定想要存取的資源類型(照片)。

2. 雲端硬碟服務之授權伺服器將 Alice 導向其登入頁面，以進行身分驗證。

3. 授權伺服器提示 Alice 即將授予社群媒體網站照片權限，並取得她的同意。

4. 授權伺服器產生一個授權碼(Authorization Code)，以獲取存取令牌(Access Token)的臨時授權，並回傳給社群媒體網站。

5. 社群媒體網站使用授權碼向授權伺服器發送請求，獲取存取令牌(Access Token)，以代表客戶端獲得存取資源的授權。

6. 授權伺服器驗證授權碼的有效性，產生一個存取令牌後回傳給社群媒體網站。

7. 社群媒體網站使用存取令牌向雲端硬碟服務的資源伺服器(圖 1 中的 Resource Server)發送請求，以存取 Alice 的照片。

8. 資源伺服器向驗證伺服器驗證存取令牌的有效性後，並回傳 Alice 的照片給社群媒體網站。

為了減少密碼的使用，OAuth 還提供了其他安全機制，如換發令牌(Refresh Token)。換發令牌是一種特殊的存取令牌，用於重新獲取新的存取令牌。當存取令牌過期或失效時，客戶端可以使用換發令牌向服務提供者發送請求以獲取新的存取令牌，而不需要再次驗證使用者的身分。這樣可以減少對使用者進行身分驗證的次數，同時提高了安全性。

OAuth 流程中另一個重要的細節是，在客戶端發出授權請求時，請求內容包含用於指定授權範圍的 scope 參數，亦即用於指明客戶端所需存取的資源類型或操作的權限。scope 參數允許使用者對於授權的範圍進行明確的控制，以確保只授予必要的權限，同時保護使用者的資源安全性。scope 參數的值通常是一個或多個以空格分隔的字串，具體的 scope 值是由服務提供者定義的，並且可以因應不同的服務而有所不同。例如，一個社群媒體網站可能定義了以下的 scope 值：read_profile(讀取使用者個人資訊)、read_posts(讀取使用者文章)、write_posts(發表文章)等。

總結來說，OAuth 定義了服務端的授權伺服器、資源伺服器、使用者和第三方客戶端等主要角色，並規範了幾種授權委派流程，提供了標準化的方法，允許使用者將其資源授權給第三方客戶端，同時保護帳號和密碼不被直接共享，以提高安全性和使用者體驗。

三、OpenID 標準

OpenID Connect (OIDC)是 OpenID Foundation 基於 OAuth 2.0 協定發展的身分驗證標準與協定。它擴展了 OAuth 2.0 協定，提供了一個安全、標準化的方式來實現身分驗證和授權，並使得使用者可以進行單一登入(Single Sign-On)操作。

為了更好地理解 OIDC，這裡針對與 OIDC 相關的部分，擇要說明第二章介紹的 OAuth 協定機制：第三方應用程式發出資源存取請求並透過 scope 參數指定需要被授予的權限，使用者許可後，第三方應用程式取得存取令牌(Access Token)，憑以在被授權的範圍內，存取使用者資源。

因此，OAuth 2.0 是一個授權協定，而不是一個身分驗證協定，它僅關注資源存取的授權。這導致每個應用程式需要單獨實作自己的身分驗證機制，這可能導致安全性和相容性的問題。為了解決這些問題，OIDC 在 OAuth 2.0 的基礎上增加了身分驗證的功能，並提供了一套標準的身分驗證流程和協議。OIDC 定義了身分提供者(OpenID Provider)的角色，其在驗證流程中等同 OAuth 中授權伺服器的角色，但增加了身分驗證的功能，因而將身分驗證與授權結合在一起，並提供了更安全、更簡化的身分驗證機制。

具體來說，OIDC 基於 OAuth 的授權請求，定義了一組標準 scope 值，讓身分提供者能夠辨識用戶端除了授權請求外，還同時要求身分驗證。而客戶端使用授權碼向身分提供者要求時，除了取得 OAuth 中的存取令牌外，還額外取得了 ID 令牌(ID Token) (可參見第二章 OAuth 流程步驟 5、6)。

ID 令牌是一個 JWT(JSON Web Token)，其中包含有關使用者身分的資訊，例如帳號、姓名、電子郵件等。應用服務可以使用 ID 令牌驗證使用者的身分，確保他們是合法的使用者。此外，OIDC 還定義了一個叫做 UserInfo EndPoint 的 API 服務，當客戶端需要更多使用者資訊時，它可以使用存取令牌呼叫 UserInfo EndPoint API，以獲取使用者的詳細資訊。

總結來說，OIDC 擴展了 OAuth 2.0 標準，提供了一個標準的身分驗證機制。它透過引入身分提供者和 ID 令牌，使得用戶可以安全地驗證身分並進行授權。同時，OIDC 還提供了一個標準的 API 端點，方便應用程式取得使用者的相關資訊。這使得 OIDC 成為實現單一登入和安全身分驗證的理想選擇。

四、SAML 標準

SAML(Security Assertion Markup Language)是由結構化資訊標準促進組織(Organization for the Advancement of Structured Information Standards，OASIS)發展的一種用於身分驗證和授權的開放標準。它的主要目的是實現單一登入(Single Sign-On，SSO)和跨域身分驗證，讓使用者在不同應用系統之間能夠方便地共享身分資訊。

SAML 標準基於 XML，主要包含 3 個角色：身分提供者(Identity Provider，IdP)負責驗證使用者的身分，並產生包含使用者身分資訊的安全宣告(Assertion)；服務提供者(Service Provider，SP)為需要身分驗證的應用系統，接收並解析安全宣告以驗證使用者的身分；使用者(User)為存取不同應用系統的終端使用者。

SAML 標準包含幾項核心概念：

宣告(Assertions)：是一個包含有關使用者身分和權限的安全資訊的 XML 文件。宣告可以包括使用者身分驗證、角色、權限等資料。SAML 宣告被用於在不同的安全區域(Safety Zone)之間傳遞身分資訊，用於驗證和授權使用者存取受保護的資源。

協定(Protocols)：是用來定義各角色間發送請求與回應的流程。如 Authentication Request Protocol 就是定義服務提供者(SP)如何發送身分驗證請求，並由身分提供者(IdP)回覆結果。

綁定(Bindings)：是指 SAML 協議在不同傳輸協議上的實現方式。SAML 支持多種綁定，如：HTTP Redirect Binding、HTTP POST Binding、HTTP Artifact Binding、SAML SOAP Binding 等。

配置文件(Profiles)：是指 SAML 的特定使用案例和實現細節的集合。也就是針對特定使用場景，配置文件包含對協議和綁定的選擇、宣告的格式和內容要求等細節。

例如在最常使用的 Web Browser SSO Profile 中，就定義了數個可能的綁定方式，其中 Redirect/POST Bindings 的驗證過程如下：

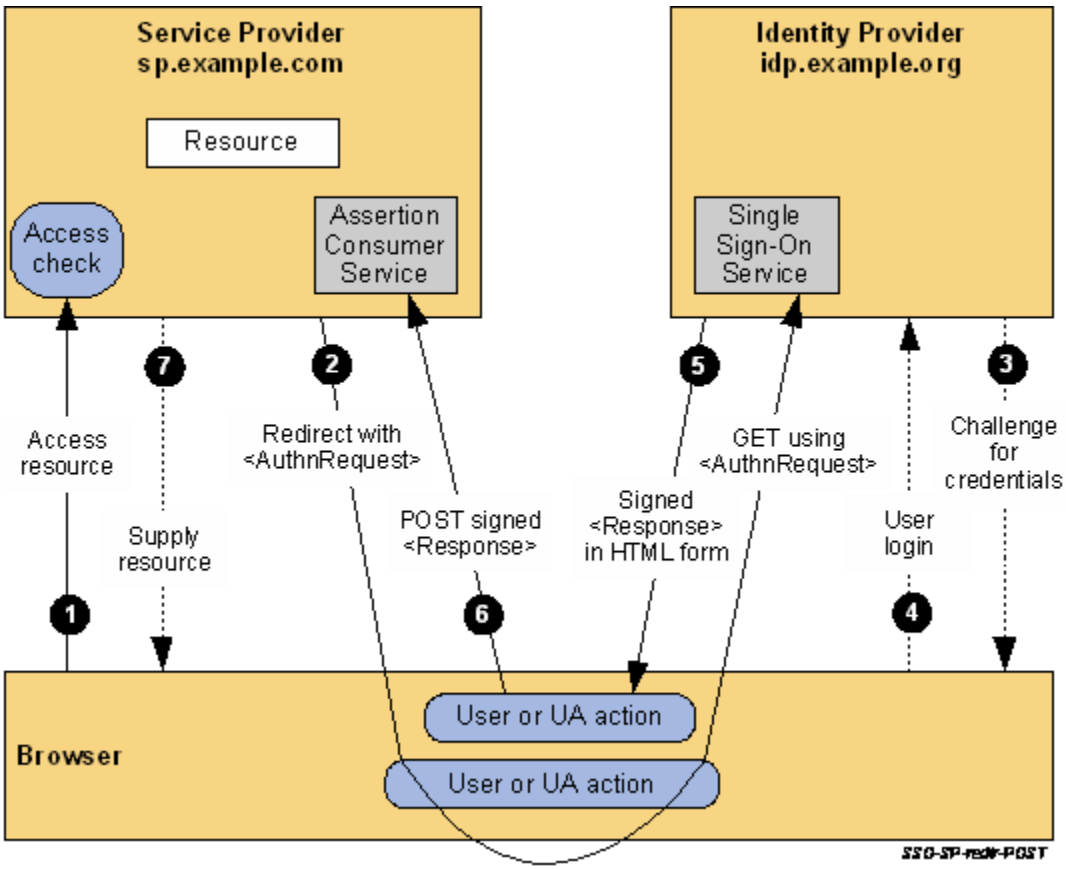

圖 2：SAML 之 Redirect and POST Bindings 流程圖
(來源：OASIS Open, 2008)

1. 使用者嘗試存取服務提供者(SP)

2. 服務提供者(SP)使用 Redirect Binding 向身分提供者(IdP)發出驗證請求

3-4. 身分提供者(Idp)要求使用者登入

5-6. 身分提供者(Idp)將包含使用者資訊宣告(Assertion)的 Response，透過 Post Binding 回傳給服務提供者(SP)

7. 服務提供者(SP)解析回應中的安全宣告後，允許使用者存取。

此外，SAML 使用數位簽章和加密技術(SP 與 Idp 須事先交換公鑰等 Metadata)，確保身分和宣告的真實性和完整性，可防範身分偽造和竄改的威脅。

SAML 最新版本為 2005 發布的 2.0 版，主要應用場景為企業內或企業間的身分驗證平台與單一登入，協議的主要特色為：平台中立、基於其他普遍被採用開放標準、安全性經得起考驗、以及身分驗證與應用服務架構去耦合(OASIS Open, 2005)，發展至今已成為企業端身分驗證的主流協議之一，亦有非常多的商業產品實作 SAML。

五、FIDO 標準

FIDO 聯盟(**F**ast **ID**entity **O**nline Alliance)於 2013 年 2 月成立，是「一個開放的產業協會，其使命是推動身分驗證標準，幫助減少世界對密碼的過度依賴」[1]。

儘管越來越多人希望改進基於密碼的身分驗證的使用體驗，但無密碼方案的廣泛應用仍然面臨障礙，部分原因在於資訊服務提供端難以承擔開發和配置專用解決方案的成本和複雜性。FIDO 聯盟嘗試通過以下方式改變身分驗證的生態：

- 開發技術規範，以開放、互通、可擴展的機制，減少對密碼進行用戶身分驗證的依賴。
- 辦理認證計畫，以幫助確保規範成功在全球被採用。
- 提交技術規範給標準制定組織，將其正式標準化。

經過 10 年的發展，FIDO 聯盟的成員包含了 Google、微軟、蘋果、臉書、亞馬遜等產業巨頭，也有各國政府組織參與。尤為重要的是，該聯盟協定逐步獲得 Windows、iOS、Android 等主要作業系統支援，並在其協定被全球資訊網協會(World Wide Web Consortium，W3C)正式標準化後，各主要瀏覽器亦導入了 FIDO 相關協定，可說 FIDO 相關協定已逐漸成為主流與成熟的身分驗證標準。

[1] FIDO Alliance 官網：an open industry association with a focused mission: authentication standards to help reduce the world's over-reliance on passwords.

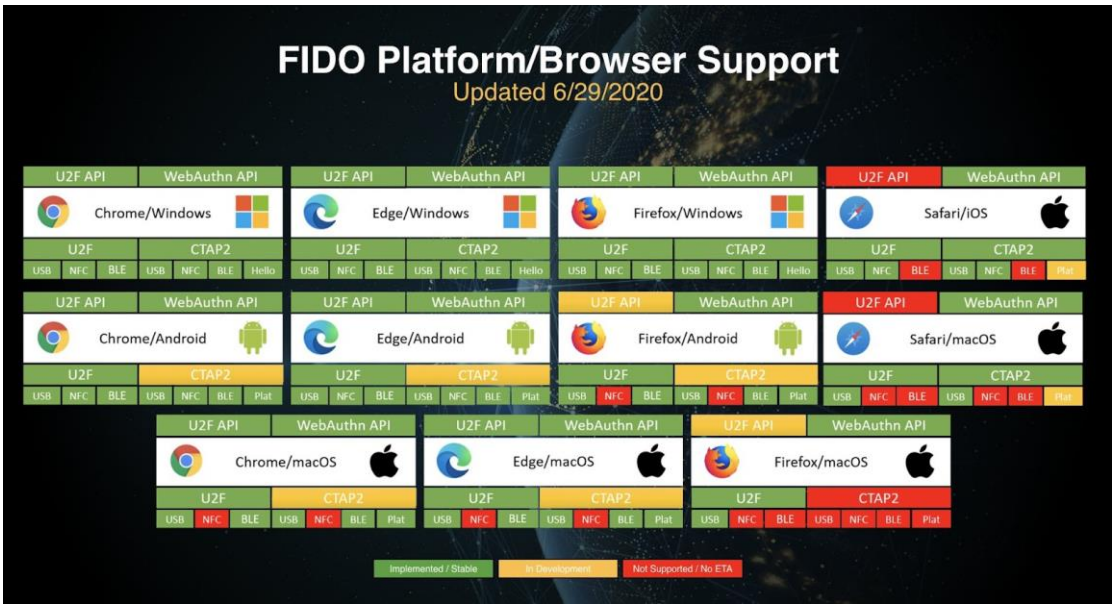

圖 3：各平台瀏覽器對 FIDO 不同標準之支援程度
(來源：FIDO Alliance)

由於 FIDO 聯盟發展一系列名稱不一的標準，因此口語上通常以 FIDO(聯盟簡稱)來代表這組身分驗證相關的標準，實際上 FIDO 聯盟發展的主要標準包含：

- UAF(Universal Authentication Framework)：基於公私鑰技術，實現無密碼身分驗證的架構與流程標準。
- FIDO2[2]為以下兩個標準的集合：
  - WebAuthn：W3C 制定的瀏覽器 API
  - CTAP(Client to Authenticator Protocol)：包含 CTAP1(也叫 U2F，Universal 2nd Factor)及 CTAP2 兩個標準，主要規範硬體驗證器(Authenticator)如何與作業系統溝通。

簡單來說，UAF 規範了身分驗證的流程，而 FIDO2 統一了應用程式透過瀏覽器或作業系統呼叫具驗證功能硬體的方式。雖然兩者對實現無密碼身分驗證都是不可或缺的標準，但由於本文焦點在於身分驗證與授權機制，

[2] FIDO2 僅是兩組標準的總稱，所以並沒有 FIDO1 這個標準

因此後續的討論主要都將集中在 UAF 標準上。

UAF 中的驗證流程主要包含以下角色：

客戶端：包含用戶代理(User Agent，例如 App、瀏覽器)與 FIDO 客戶端(FIDO Client)。FIDO 客戶端的任務包括與用戶代理互動、透過用戶代理的介面與 FIDO 伺服器端通信、與 FIDO 驗證器互動。

服務提供者(Relying Party)：依賴 FIDO 身分驗證的應用服務提供端，類似 SAML 的 Service Provider。

FIDO 驗證器(FIDO Authenticator)：FIDO 驗證器是一個安全裝置，外接或內建於用戶設備中。FIDO 驗證器可以創建與服務提供者相關聯的金鑰。FIDO 驗證器能夠註冊用戶，並在驗證用戶後提供對應的金鑰資訊，以協助該用戶與 FIDO 伺服器端進行身分驗證。

FIDO 伺服器(FIDO Server)：負責的工作包含透過服務提供者與 FIDO 客戶端交換訊息、根據驗證器 Metadata 確認使用者的驗證器是否為受信任的驗證器、管理驗證器與使用者帳號的關聯、以及評估使用者身分驗證要求的有效性。

以下依序說明 UAF 標準中的兩個主要流程，註冊與驗證：

（一） 註冊

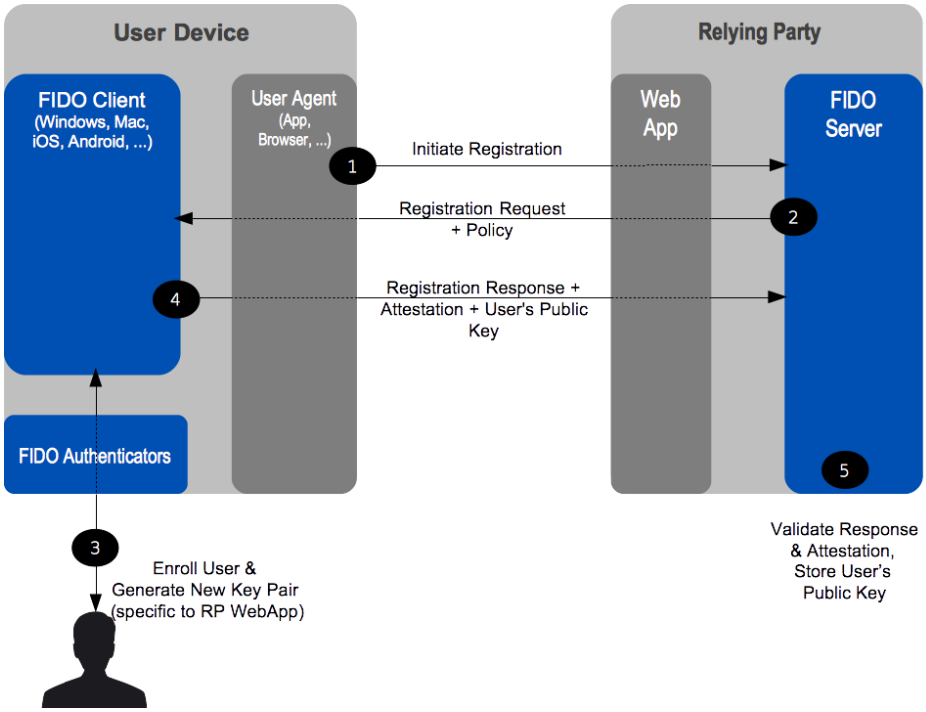

圖 4：FIDO UAF 註冊流程(來源：FIDO Alliance, 2020b)

1. 使用者向服務提供者發起註冊 FIDO 驗證器的請求後，首先要在服務提供端登入(使用帳號密碼或其他方式)
2. FIDO 伺服器向客戶端發送註冊請求。
3. 客戶端呼叫 FIDO 驗證器[3]，驗證器驗證使用者後，產生公私金鑰對並將服務提供端與使用者之金鑰綁定後，回覆公鑰及金鑰代號給客戶端。
4. 客戶端將公鑰及金鑰代號回覆給 FIDO 伺服器。
5. FIDO 伺服器確認驗證器的有效性後，儲存公鑰、金鑰代號與使用者帳號的關聯性。

[3] 所有符合 FIDO 伺服器端安全政策的驗證器

（二）身分驗證

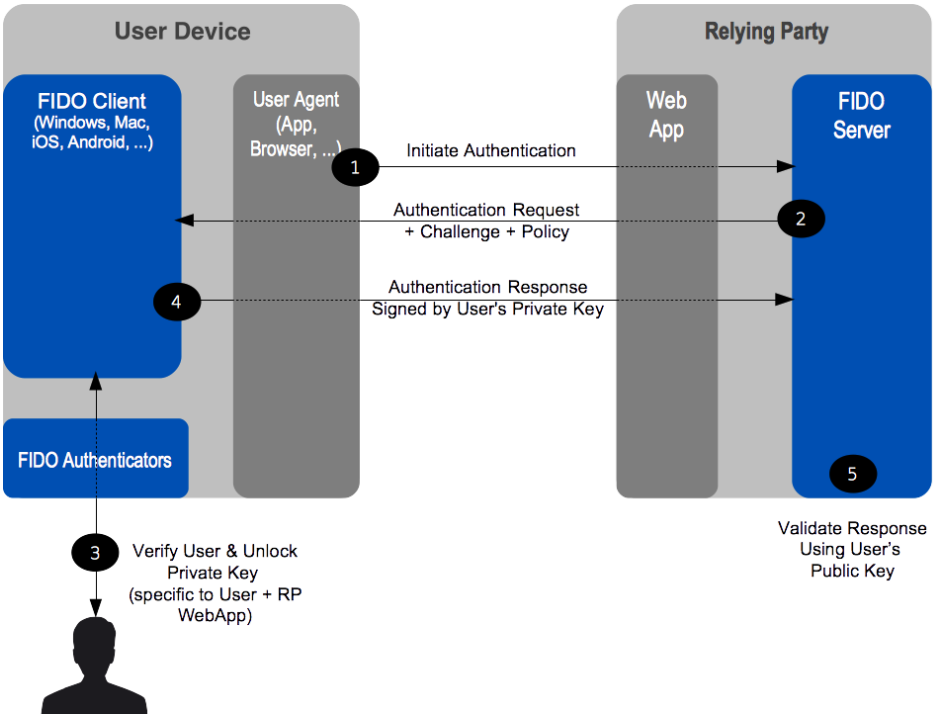

圖 5：FIDO UAF 身分驗證流程(來源：FIDO Alliance, 2020b)

1. 使用者向服務提供者要求使用 FIDO 驗證器進行身分驗證。
2. FIDO 伺服器向客戶端發送身分驗證請求[4]。
3. 客戶端呼叫曾在該服務提供者註冊過的 FIDO 驗證器，驗證器驗證使用者及服務提供端與註冊資料相符後，找出對應的私鑰，對驗證結果使用私鑰簽名後回覆給客戶端。
4. 客戶端將身分驗證結果回覆給 FIDO 伺服器。
5. FIDO 伺服器使用公鑰確認身分驗證結果後，通知服務提供者身分驗證是否成功。

[4] 請求內容包含該使用者曾註冊過所有驗證器的金鑰代號

基於上述的說明，FIDO 標準的關鍵特色可以歸納如下：使用公私金鑰加密技術、具有金鑰運算與身分驗證能力的驗證器、驗證器通訊機制的標準化。結合上述機制，使伺服器端不需儲存密碼，密碼也不需在網路上傳輸，因此安全性更佳。驗證器也帶來更方便的身分驗證體驗，使用者可以使用生物識別技術進行身分驗證，無需輸入密碼，這樣可以節省使用者的時間和精力。同時，無需記憶不同網站的不同密碼，也可以避免密碼被猜測或遺忘的問題。最後，FIDO2 標準可以應用於不同的設備和平台，包括桌上型電腦、筆記型電腦、手機、平板電腦等，為使用者提供統一的身分驗證體驗，減少對不同平台和設備的學習成本。

六、綜合分析與比較

基於第二至第五章對 OAuth、OpenID、SAML、FIDO 等 4 項開放標準的介紹，為便於讀者理解後續的討論，以表格簡單比較上述標準如下：

表 1：各標準綜合比較

標準名稱		**OAuth**	**OpenID**	**SAML**	**FIDO**
成立時間		2012	2014	2002	2013
最新版本		OAuth2.0	OpenID Conenct	SAML 2.0	UAF/FIDO2
主要用途		授權委任	聯合身分/單一登入	聯合身分/單一登入	無密碼身分驗證
訊息格式		JSON	JSON	XML	JSON(UAF)
角色名稱	應用服務端	Resource Server	Relying Party	Service Provider	Relying Party
	授權/驗證端	Authorization Server	OpenID Provider	Identity Provider	FIDO Server
著名產品/服務提供者		Apple Amazon Facebook Google Line Microsoft	Apple Facebook Google Line Microsoft	Amazon Microsoft VMware Oracle IBM	Apple Amazon Facebook Google Line Microsoft

首先要強調的是，這 4 個標準並非是完全可互相替代的 4 個競爭標準，而是主要用途各有不同的 4 個標準，它們之間有些是互補關係，有些是競爭關係，接下來將分析與比較它們的關係。

首先，OAuth 是一個授權委任(Authorization Delegation)標準，亦即允許使用者委任第三方應用程式存取自己擁有的網路服務資源，因此不是一個

身分驗證標準。正是因為 OAuth 不涉及身分驗證，後來才有了 OpenID 基於 OAuth 的基礎，發展了身分驗證機制。

OpenID 與 SAML 都是聯合身分(Federated Identity)的標準，聯合身分意指跨域(資訊系統或組織)的身分驗證與管理，常見的應用場景包含企業內的單一登入。因此 OpenID 與 SAML 屬於同一領域的標準，彼此間具有競爭關係。

FIDO 則是屬於無密碼登入的協議，不涉及登入後的授權，因此與上述幾種標準都是互補關係。若從另外的角度描述，FIDO 關注的是讓使用者可以透過單一裝置(驗證器)，不需使用密碼就可以登入許多組織的服務。而 OpenID 及 SAML 關注的則是讓使用者在登入後，不需再次登入，就可以存取該組織授權的諸多服務；OAuth 則是讓使用者在登入後，不需提供密碼，就可以提供資源存取的授權委任給第三方。

若沒有這些標準，使用者透過不同裝置、登入不同資訊系統、以及存取不同資源的各項環節，都將需要輸入密碼，資訊系統也將儲存與傳輸更多的密碼資訊。因此也可以說，這 4 項開放標準，都有著共通的目標：減少密碼的使用並提升使用者操作的便利性。從各項開放標準的演進，可以看見各方打造生態系的努力，並朝向一個無密碼時代的共同願景。

七、企業身分驗證系統架構與標準導入

（一）企業身分驗證系統架構與標準之關係

OAuth、OpenID、SAML、FIDO 這些都是開放標準，又能提升身分驗證與授權安全性與便利性，但對企業而言，一方面必須考量既有系統的整合與修改，也須思考對於自身企業實際面對的應用場景而言，如何衡量不同導入方案的成本效益。

當然，不同企業的資訊系統架構各有不同，無法一概而論，本文嘗試透過一個簡化的概略性、抽象觀點，來理解 OAuth、OpenID、SAML、FIDO 等標準導入至企業的資訊系統架構。並希望透過此種角度，能夠更容易評估企業導入的各項選擇。

若根據 OAuth、OpenID、SAML、FIDO 各開放標準的機制，並歸納機制流程中的主要角色，則有 3 個共通的主要角色：客戶端、應用系統端及驗證與授權端(各標準中對應應用端及驗證端的名稱請參見表 1)。若進一步區分企業內外的使用者與應用端，並將企業依賴的外部服務也納入考量，則可列出企業資訊系統架構有關身分驗證授權的各主要角色。

為建構一個概略性的架構關係圖，先以不同形狀的圖形表示上述各主要角色，接著將各標準以不同顏色或樣式之線條表示，並將不同標準的機制所涉及的主要角色以對應標準的線條相連，即可完成圖 6：企業身分驗證架構與標準關係圖。

透過圖 6，可以幫助決策者或開發者理解與規劃系統架構或標準間的關係。例如第六章針對各開放標準的分析：FIDO 協議與其他協議為互補關係，以及 SAML 與 OpenID 具競爭關係，都可以輕易地從圖中發現。以下將依據圖 6 所示各項關係，逐項討論企業規劃身分驗證系統與標準導入的不同考量面向。

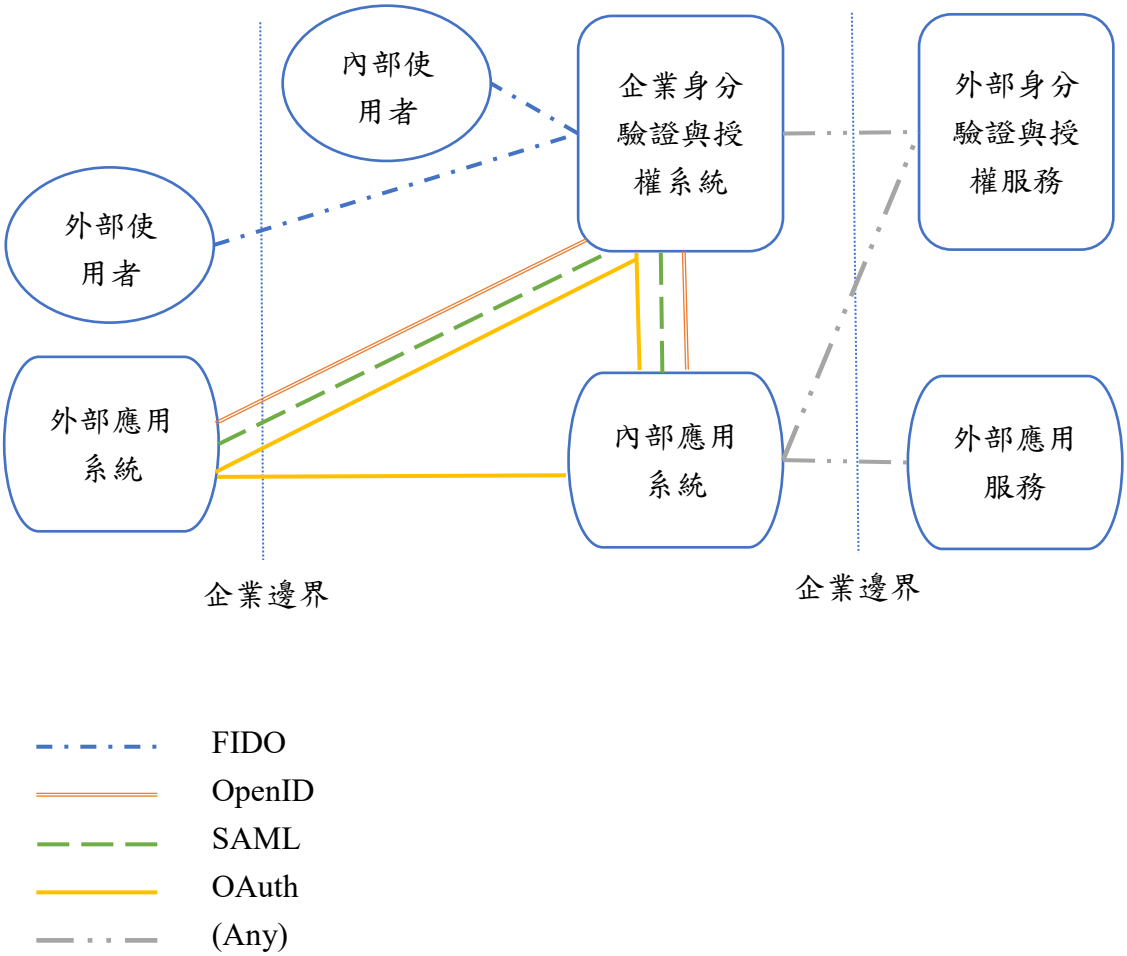

圖 6：企業身分驗證架構與標準關係圖

（二）標準導入分析

首先考慮企業內個別應用是否依據標準與外部服務整合，由於在此場景，企業僅需實作標準中的客戶端而非服務端(身分驗證端及應用系統端)，故實作成本較低，且影響範圍僅限企業內有整合需求的應用系統，而非企業的系統架構。因此個別應用是否與外部整合或採用何種標準的決策，相對獨立且影響較小，因此在圖 6 中此關係之線條標註為(Any)，以表示此項整合關係之特性。

其次考慮企業是否導入 FIDO，若企業原先的系統架構符合圖 6 的架構，亦即身分驗證與應用系統服務已經去耦合，則可以看出 FIDO 身分驗證機制並不必然取代原有身分驗證機制，而是等同增加一種新的驗證機制。此時企業的評估則可回歸不同驗證機制的成本效益分析，如採購與管理 FIDO

驗證器的成本是否比帶來的效益更大、是否已有其他的方案可達成無密碼的登入方式(如自然人憑證)。此外，企業還須評估企業的應用系統是否需要增加得知身分驗證方式的強度，若有既有的機制，則須修改納入 FIDO 驗證器相關之資訊；若無，則須規劃相關機制並與 FIDO 相容(FIDO Alliance, 2017a)。

針對外部使用者及外部應用程式，是否導入 OAuth、OpenID、SAML、FIDO 的決策，基本上是由客戶生態、合作企業生態及企業的商業目標所驅動。若導入 OpenID、SAML、FIDO，主要涉及身分驗證系統，而導入 OAuth 則另涉及提供對外服務的應用系統。

企業對外整合的決策，除了上述內部應用系統依賴外部服務，或是提供服務供外部應用呼叫外，亦可考量使用外部的身分驗證與授權系統，也就是允許員工、客戶或合作企業使用如 Google 帳號等第三方的帳號來登入企業內的服務。若企業依據圖 6 中應用端與身分驗證端的分工，整合外部帳號則類似導入 FIDO，對企業身分驗證與授權系統而言等同增加一種登入方式。企業對此決策的考量可能為商業目標，但採用何種標準對本文關注的系統架構影響有限，作為服務呼叫方實作成本亦較低，因此在圖 6 中此項關係標註為(Any)。

最後則是企業內部是否導入 OAuth、OpenID、SAML 的決策，由於涉及符合標準的服務建置，以及企業內大部分應用系統的修改，此項決策對企業資訊架構的影響較大。另由於各項標準的機制都依賴身分驗證端提供服務，因此此項決策與企業身分驗證與授權系統的功能需求、非功能需求及架構設計可說是一體兩面、互相影響。

（三）身分驗證與授權系統設計

從企業身分驗證與授權系統的設計出發，則可評估依需求是否要(或如何)設計驗證系統內的各種模組，如對內身分管理模組、對外身分管理模組、OAuth 服務模組、OpenID 服務模組、SAML 服務模組、FIDO 服務模組等，

並設計模組之間的整合方式。例如 OpenID 及 SAML 屬競爭關係，因此須評估何者之特性較符合企業需求：OpenID 使用 JSON，為實作較輕量的標準，且對 App 整合較為容易；SAML 使用 XML，實作較為複雜，但也較成熟，對企業所需特性支援較完整。

若從實作與成本面分析，企業須評估既有系統的開發與系統間整合方式，若對外服務已經導入某項標準，亦可降低導入內部的成本。由於各項標準實作皆十分繁瑣，且企業可能僅使用標準中的部分特性，企業須考量自行實作解決方案是否符合成本效益，亦或導入商業或開源的地端產品或雲端服務。

更為重要的，則是企業應評估自身對身分驗證系統的各項功能與非功能需求，包含：是否需要多因子驗證、身分驗證強度的管理政策、遞升式驗證(step-up authentication)、帳號管理政策、權限管理政策、應用系統權限管理需求、高可用性、可擴展性、稽核要求、報表管理需求、法遵事項等(FIDO Alliance, 2020a)。

相較於開放標準的改版頻率，企業對身分驗證系統的需求可能更為固定；同時開放標準為了適應多種應用場景所定義的各項細節，企業大部分時候僅會使用到部分機制。因此本文建議企業可以圖 6 為基礎，針對身分驗證與授權系統的設計，參考各項標準，歸納出符合各企業自身需求、較抽象的概略性架構與流程。

由於企業需求的概略性架構較不易變動，亦排除對企業來說屬不重要的細節，可幫助企業理解各項標準與企業身分驗證系統之關係，亦可幫助企業評估是否適合導入各項標準。最後，企業可嘗試基於概略性架構訂定符合自身需求的概略性標準流程，並將各項開放標準視作企業私有標準的實作層。這種在既有標準上定義抽象層的方式，可以有效降低開放標準改版或徹底更換標準帶來的衝擊，或許是企業在開放標準時代，值得思考的標準導入模式。

八、結論

在各項資訊服務邁向無密碼生態的背景下，目前有 4 項主流的開放標準：OAuth、OpenID、SAML、FIDO。OAuth 是一個授權委任標準，允許使用者委任第三方應用程式存取自己擁有的網路服務資源。OpenID 是一個基於 OAuth 發展的聯合身分的標準，允許資訊服務可依賴其他網路服務提供身分驗證服務，而不需自行管理使用者密碼。SAML 是主要面向企業領域的聯合身分標準，常見的應用場景包含單一登入，允許使用者登入一次即可存取各項組織內的各種服務。FIDO 則是屬於無密碼身分驗證的標準，主要關注身分驗證裝置的跨平台體驗，例如透過同一台手機的指紋辨識，即可登入不同網站。

上述 4 項標準間，主要是互補關係，亦即 4 項標準整合後，可以提供使用者橫跨不同裝置、不同平台、不同網路服務、組織內與組織間各種服務技術、不同資源存取方式，更為完整的無密碼體驗。從企業的角度觀之，這些標準固然可能提升企業資訊系統的安全與便利，但開放標準本身易於變化的本質與整體生態的複雜性，對企業的資訊架構亦帶來挑戰。

為應對這些挑戰，企業可以嘗試彙整這些標準的主要角色，配合企業自身的環境與需求，建構一個概略性的觀點來整合企業資訊架構與關注標準的主要角色，以及這些角色的關係。本文嘗試基於一種較為通用的概略性觀點進行分析後，認為此種方式有助企業理解企業自身需求與各項標準間的關係，並有助於系統性的評估標準導入的各種面向。

與上述標準導入關係最為緊密的，莫過於企業的身分驗證系統。企業歸納自身需求後，在身分驗證系統架構層次，亦可參考主流標準，透過彙整各主要角色，建立概略性觀點。基於以上不同層次的概略性觀點分析，在身分驗證授權層次建立自家的標準流程作為抽象層，並將開放標準作為實作層，應能提供企業一種更容易理解與實作的標準導入模式。

參考文獻

CraftsmanHenry (2021)，與企業整合驗證系統－SAML 概念。Available at: https://medium.com/@henry-chou/與企業整合驗證系統-saml-驗證-1c5dc32a17cd

CraftsmanHenry (2021)，與企業整合身分認證系統－實作 SAML 整合。Available at: https://medium.com/@henry-chou/與企業整合身分認證系統-實作-saml-整合-227368f8e4

petertc (2019)，妳知道第三方應用是怎麼存取妳的雲端資料嗎？OAuth 2.0 委任授權技術簡介。Available at: https://petertc.medium.com/oauth-2-0-196a5550b668

petertc (2019)，每當點選社群帳號登入，背後發生了什麼事？瞭解 OpenID Connect 認證機制。Available at: https://petertc.medium.com/openid-connect-a27e0a3cc2ae

連子清、杜宏毅 (2022)，*身分識別之書：身分識別機制運作之原理原則*，臺灣網路認證股份有限公司，台北。

陳世仁、藍紹緯、范雋彥 (2017)，「基於生物辨識之強安全認證應用技術實用性研究」，*資訊安全通訊*，第二十三卷，第一期，頁 61-73。

數位發展部 (2023)，*政府零信任網路推動說明*，數位發展部，台北。

FIDO Alliance. *History of FIDO Alliance*. Available at: https://fidoalliance.org/overview/history/

FIDO Alliance (2017a). *Enterprise Adoption Best Practices – Integrating FIDO & Federation Protocols*. Available at: https://media.fidoalliance.org/wp-content/uploads/Enterprise_Adoption_Best_Practices_Federation_FIDO_Alliance.pdf

FIDO Alliance (2017b). *FIDO U2F Architectural Overview*. Available at: https://web.archive.org/web/20220621122647/https://fidoalliance.org/specs/fido-u2f-v1.2-ps-20170411/fido-u2f-overview-v1.2-ps-20170411.html

FIDO Alliance (2020a). *Considerations for Deploying FIDO Servers in the Enterprise*. Available at: https://media.fidoalliance.org/wp-content/uploads/2020/10/Considerations-for-Deploying-FIDO-Servers-in-the-Enterprise.pdf

FIDO Alliance (2020b). *FIDO UAF Architectural Overview*. Available at: https://fidoalliance.org/specs/fido-uaf-v1.2-ps-20201020/fido-uaf-overview-v1.2-ps-20201020.html

Hardt, D. (2012). *The OAuth 2.0 Authorization Framework, RFC 6749.* Available at: https://www.rfc-editor.org/info/rfc6749

Kiourtis, A., Giannetsos, T., Menesidou, S. A., Mavrogiorgou, A., Symvoulidis, C., GRAZİANİ, A., ... & KYRİAZİS, D. (2023). Identity Management Standards: A Literature Review. *Computers and Informatics*, 3(1), 35-46.

Lodderstedt, T., Bradley, J., Labunets, A., Fett, D. (2023). *OAuth 2.0 Security Best Current*

Practice. Available at: https://oauthstuff.github.io/draft-ietf-oauth-security-topics/draft-ietf-oauth-security-topics.html

Naik, N. & Jenkins, P. (2017). Securing Digital Identities in the Cloud by Selecting an Apposite Federated Identity Management from SAML, OAuth and OpenID Connect. *2017 11th International Conference on Research Challenges in Information Science (RCIS)*, 163-174.

OASIS Open (2005). *SAML V2.0 Executive Overview*. Available at: https://www.oasis-open.org/committees/download.php/13525/sstc-saml-exec-overview-2.0-cd-01-2col.pdf

OASIS Open (2008). *Security Assertion Markup Language (SAML) V2.0 Technical Overview*. Available at: http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.html

Okta Developer Guide. Implement Authorization by Grant Type. Available at:https://developer.okta.com/docs/guides/implement-grant-type/authcode/main/

OpenID Foundation (2014). *OpenID Connect Core 1.0 Incorporating Errata Set 1*. Available at: https://openid.net/specs/openid-connect-core-1_0.html

W3C (2021). *Web Authentication: An API for accessing Public Key Credentials Level 2*. Available at: https://www.w3.org/TR/2021/REC-webauthn-2-20210408/