

企業整合與數位轉型藍圖

統一 Opcenter APS、UiPath 和 SAS Viya

以實現策略性成長

文件編號：STRATEGY-FY25-001

密等：內部機密

版本控制列表

版本	日期	作者	修訂說明
1.0	2025/07/23	Gemini	初版建立：整合開源軟體策略、IDE 策略、數據網格架構分析及合作契約附件。

I. 執行摘要

本公司目前仰賴 Opcenter APS 進行生產排程、UiPath 執行基礎自動化，以及 SAS Viya Studio 進行數據分析，這三套軟體雖然功能強大，但各自獨立運作，導致營運效率低下、人工流程繁瑣，並產生大量的「搜尋時間成本」。這種碎片化的現狀阻礙了企業的敏捷性與數據驅動決策的能力。

為應對此挑戰，本報告旨在提出一套全面的策略與技術藍圖，將這些獨立系統轉化為一個統一、智慧化的資源、時間與工作管理平台。此轉型不僅能有效降低營運成本，更關鍵的是，它將釋放前所未有的生產力，提升企業敏捷性，並透過數據賦能決策，最終推動營收成長並強化市場競爭力。

本報告的核心建議包括：

- 策略性整合：** 建立強健的整合架構，運用 API 與事件驅動模式，將 Opcenter APS、UiPath 和 SAS Viya 無縫連結，形成一個協同運作的生態系統。

- **AI 驅動自動化：** 將人工智慧（如 GPT/機器學習）融入開發生命週期，從程式碼生成到智慧郵件處理及自動化測試，加速創新並提升營運智慧。
- **數據中心化與治理：** 建立統一的數據層，並實施嚴謹的數據對齊與版本控制實踐，確保所有平台上的數據完整性與可存取性。
- **統一管理平台：** 將現有分散的工具與日曆整合，逐步演進為全面的資源、時間與工作管理平台。
- **績效與問責：** 制定清晰的關鍵績效指標（KPIs），並建立具約束力的問責機制，以確保專案成功執行並達成可衡量的業務成果。

預期效益將包括：顯著提升生產力、改善準時交貨率、優化庫存管理、加速新功能上市時間、提升客戶滿意度，並形成一個正向的數據驅動工作循環。

II. 現況分析與挑戰識別

本節將深入探討 Opcenter APS、UiPath 和 SAS Viya Studio 各自的現有功能與關鍵痛點，並闡明其獨立運作所導致的效率低下問題。

Opcenter APS：生產排程與營運缺口

Opcenter APS 是一款先進的排程與規劃軟體，旨在優化生產排程、縮短停機時間並提升客戶服務¹。它支援長期策略規劃（數月至數年）、中期戰術規劃（數週）及詳細的順序與排程，並提供 Opcenter Scheduling 和 Opcenter Planning 等模組¹。該系統可偵測瓶頸、減少設定時間、提高機器利用率、降低在製品庫存，並能將準時交貨率提升高達 50%¹。Opcenter APS 設計上可與企業資源規劃（ERP）和製造執行系統（MES）無縫整合²，特別是與 MSSQL 資料庫（如 Microsoft Dynamics 365 Business Central）的介接，支援圖形化生產規劃與客製化 KPI³。

然而，儘管具備這些能力，Opcenter APS 在實際應用中仍面臨以下挑戰：

- **外部演算法缺乏版本控制：** 目前用於 Opcenter APS 的五種外部開發演算法缺乏 Git 版本控制，僅能透過套件更新方式進行管理。這種版本控制的缺失，導致了嚴重的技術債務和營運風險。在沒有 Git 等版本控制系統的情況下⁴，對這些客製化演算法的

每一次修改都成為高風險的人工操作。當錯誤發生時，缺乏清晰的變更歷史記錄使得問題追溯和偵錯變得極其困難。這直接增加了問題解決時間⁵和事件發生率⁵，並可能導致系統不穩定或生產排程錯誤。這種狀況也阻礙了新演算法的快速迭代與實驗，因為變更難以管理和追蹤，進而扼殺了核心製造營運的創新與敏捷性。

- **異常通知仰賴人工郵件：** 現行系統仰賴供應商透過電子郵件發送異常通知。這種人工通知流程本質上是被動且低效的。對於關鍵警報，人工監控和處理電子郵件會耗費寶貴的人力工時，直接增加了查詢和處理異常所需的「搜尋時間成本」。這不僅延長了問題解決時間⁵，也影響了整體系統的可用性⁵。製造環節中異常偵測的延遲，將直接導致停機時間成本增加⁵、生產延誤，甚至可能錯過交貨期限¹。
- **自動化範圍受限：** 儘管 Opcenter APS 在製造排程方面表現出色，但將其「相同架構」擴展到公司內部會議通知的需求，揭示了內部流程自動化潛力的未被充分利用。這表示公司的「時間管理」和「工作管理」目前是碎片化的，依賴獨立的、可能是人工或半自動化的內部排程流程，導致存在「四個以上」的現有工作日曆【使用者查詢】。將 Opcenter APS 的排程邏輯，或至少其底層數據，應用於內部營運，是實現「資源管理平台、時間管理平台、工作管理平台」願景的關鍵一步，標誌著從點狀解決方案向整體企業視角的轉變。

UiPath：自動化潛力與溝通脫節

UiPath 目前用於公司內部和外部的郵件與系統通知。它利用機器人流程自動化（RPA）來執行常見的郵件任務，例如回覆、轉發、刪除、處理附件和匯總資訊⁸。此外，它還能根據郵件內容在任務管理應用程式中建立任務⁸。UiPath Studio 與 Outlook 和 Gmail 等主流郵件應用程式相容⁸。

然而，UiPath 的當前應用存在以下限制：

- **郵件智慧化與進階處理不足：** 儘管 UiPath 處理通知，但使用者明確指出需要「清理電子郵件、分析多郵件本文的內容並依照內容分『處理等級』，並回覆處理程序與步驟，如查核資料或資料庫，設定驗證，版本控制，正常與錯誤差異，給相對應的人員」。這表明目前的 UiPath 實施仍停留在基礎 RPA 層面，未能充分利用其進階 AI 能力進行智慧內容理解和自動化決策。人工分析郵件內容、判斷處理等級並撰寫詳細回覆的步驟，抵消了大部分自動化效益。這也是「low code 但其實與 vba 比較不出差異」觀點的來源——它執行的是簡單的、基於規則的自動化，而非智慧流程自動化。人工智慧驅動的郵件分類系統能夠根據意圖和優先級自動分類郵件，甚至草擬回

覆¹⁰。對關鍵事件郵件（例如來自 Opcenter APS 的異常通知）而言，這種對人工干預的依賴造成了顯著的瓶頸，直接影響了問題解決時間⁵，並加劇了「搜尋時間成本」問題。公司因此錯失了將 AI 與自動化結合以把握時機的「超自動化」（hyperautomation）機會¹¹。

- **「Low Code 卻與 VBA 無異」的觀感：**非資訊部門同事認為 UiPath 與 VBA 差異不大，導致「資訊不同不與混淆」。這種觀感問題顯示，UiPath 在展現其超越簡單腳本的價值主張方面存在不足，阻礙了其更廣泛的採用，並導致溝通碎片化。這種認知表示 UiPath 在複雜、整合式工作流程、集中式協調¹²和可擴展性¹²方面的能力未被有效利用或傳達。這也凸顯了對使用者進行更有效培訓和展示更高價值自動化的必要性⁵。儘管「low code」旨在賦予公民開發者能力，但如果其功能未被視為優於 VBA，則這種賦能將無法實現。這種觀感也促使非資訊部門員工偏好使用 Line，導致「資訊不同不與混淆」，進一步加劇了溝通孤島，阻礙了「統一的網路溝通介面」的建立。
- **Salesforce 整合缺口：**現行 UiPath 無法利用自動化的特性將客戶管理系統 Salesforce 的資料進行客戶與場景分類。這表示在數據驅動的客戶關係管理和目標業務策略方面，公司錯失了重要機會。Salesforce 儲存著關鍵的客戶數據，若無法進行有效的分類和歸類，公司將無法有效地細分客戶群體，以制定有針對性的行銷、銷售策略或提供個人化服務。這將直接影響銷售效率和客戶滿意度⁵。值得注意的是，UiPath 實際上可以透過 API 和預建連接器與 Salesforce 整合¹²，因此當前的限制可能源於實施或配置不足。

SAS Viya：數據分析與擴展性挑戰

SAS Viya 是一個完整的數據分析工具。它支援從 Azure 環境中獲取數據，包括 Azure Data Lake Storage Gen2、Azure SQL Database 等，並能與 Azure Machine Learning/Synapse Machine Learning 整合，支援 Python、R 等多種程式語言進行模型建構與部署¹¹。SAS Studio 也支援 Git 版本控制功能¹¹。

然而，SAS Viya 的潛力目前因以下問題而受限：

- **即時與固定時程串流開發重複：**使用者指出，對於即時串流與固定時程串流（5 分鐘與 1 小時）的功能，需要額外多次重複開發。這表明在數據串流處理方面，缺乏可重複使用的組件或標準化的數據管道，導致開發效率低下並增加了開發時間¹⁶。這種重複性工作不僅耗費資源，也延緩了從數據中獲取即時洞察的能力。

- **額外演算法開發與報告生成：**連結不同資料庫、開發其他額外演算法、生成不同特徵報表與經營管理分析報告，都需要額外開發。這暗示了開箱即用分析能力的不足，或整合客製化模型與數據源的複雜性，導致重複性的開發工作。同時，難以靈活生成不同視角和層次的報告，也阻礙了為不同利害關係人提供數據驅動的經營洞察。
- **Kubernetes 環境與網路控管複雜性：**大量的 Kubernetes 環境與網路控管問題，導致 SAS Viya 「並不能與其他軟體合作」。儘管 SAS Viya 旨在部署於 Kubernetes 環境並支援多種 Kubernetes 發行版¹¹，但複雜的網路配置和管理可能導致其與企業其他系統的互操作性受阻。這種操作複雜性可能阻礙了其部署和擴展，並限制了其與其他系統的數據交換能力。
- **與其他軟體合作受限：**使用者觀察到 SAS Viya 「並不能與其他軟體合作」，儘管坊間有「uipath(ETL)+mlops(model dev)+power BI(報表視覺化)」的整合模式。這種認知與 SAS Viya 宣稱的整合能力（例如與 Azure 服務、Microsoft 365、Power Automate、Git 等）¹¹存在矛盾。這可能源於當前實施的技術障礙、缺乏策略性整合規劃，或對其潛在整合能力的認識不足。這種隔離狀態阻礙了數據分析成果向業務流程的有效傳遞，導致「完成分析報表是一件事，跟如何思考下一步對策並無法有效的正向工作循環」的問題。

跨系統挑戰與策略缺口

除了上述各系統的具體問題外，現況還存在以下跨系統的整合挑戰和策略缺口：

- **數據對齊問題：**「資料庫的內容對齊」是跨不同系統整合的重大挑戰。這導致數據孤島、數據不一致，並阻礙建立「單一事實來源」²⁰。數據不一致性會影響分析報告的準確性，進而影響決策品質。
- **SAP 功能整合：**查詢中提及「SAP 的功能」，但未明確其具體角色。這暗示了公司可能存在現有的 SAP ERP 系統，其與 Opcenter APS、UiPath 和 SAS Viya 的整合將是複雜的企業整合挑戰²。
- **統一網路溝通介面缺失：**對於「統一的網路溝通介面」的需求，以及「現行的工作日曆就有四個以上」的問題，凸顯了公司內部溝通和時間管理的碎片化。這不僅增加了「搜尋時間成本」，也導致了資訊傳遞效率低下和協作困難¹⁸。
- **缺乏正向工作循環：**整體而言，現有環境未能形成「分析報表是一件事，跟如何思考下一步對策並無法有效的正向工作循環」。這表示從數據洞察到實際營運執行之間存在脫節，阻礙了企業從數據中獲取最大價值並進行持續改進。這也反映了對「最小單位、人的工時、半成品、組裝成品的完成時間、工作站內部與外包協調方式」等原

子化數據的缺乏有效整合與利用。

III. 策略願景與三年營收計畫

本節將闡述轉型後的策略願景，並提出未來三年的營收目標拆解、關鍵績效指標（KPIs）及具體的兩個月內里程碑，以確保業務目標與技術實施緊密結合。

營收目標與拆解

本公司的策略願景是透過數位轉型，將現有分散的軟體資產轉化為一個協同運作的智慧平台，從而實現顯著的營收增長。設定三年營收目標需要基於現實的增長預期，並考慮公司的現有增長率、定價策略以及人才招聘影響²⁵。

建議的三年營收目標拆解策略如下：

- **第一年：基礎設施強化與核心流程自動化**
 - **目標：** 營收增長 15%。
 - **拆解：** 主要來自於營運效率提升所帶來的成本節省（例如，減少人工錯誤、縮短排程時間、加速郵件處理），以及部分新客戶獲取（透過優化客戶管理）。
 - **重點：** 完成核心系統（APS、UiPath、SAS Viya）的基礎整合，實現關鍵數據對齊，並將 UiPath 自動化擴展至高頻繁、重複性郵件處理與內部通知。
- **第二年：數據賦能與智慧決策**
 - **目標：** 營收增長 25%。
 - **拆解：** 透過 SAS Viya 強化數據分析能力，實現更精準的生產預測、供應鏈優化，以及基於 Salesforce 數據的客戶洞察。這將提升客戶滿意度，促進交叉銷售與向上銷售。
 - **重點：** 深度整合 SAS Viya，實現即時數據串流與分析，並將 AI/機器學習模型應用於業務決策。
- **第三年：平台化與創新服務**
 - **目標：** 營收增長 35%。
 - **拆解：** 建立統一的資源、時間與工作管理平台，實現跨部門、跨系統的無縫協作。透過自動化和數據洞察，開發新的增值服務或優化現有產品，提升市場競爭

- 力。
- **重點：** 實現全面的平台化願景，探索 AI 驅動的創新應用，並將技術優勢轉化為市場差異化。

關鍵績效指標（KPIs）與問責框架

為確保營收目標的達成，將設定一系列具體的關鍵績效指標（KPIs），並建立嚴格의問責機制。這些 KPI 將涵蓋技術營運、開發效率、業務影響及團隊績效等多個層面⁵。

以下為建議的關鍵 KPI 列表：

KPI 類別	關鍵績效指標 (KPI)	定義與衡量標準	目標 (3 年內)
營運效率	系統可用性與正常運行時間 (System Availability & Uptime)	系統可運行與使用者可存取的時間百分比。高可用性確保服務一致性並最小化中斷 ⁵ 。	≥ 99.9%
	問題解決時間 (Resolution Time)	解決已報告 IT 問題所需時間。更快的解決方案確保及時服務恢復並減少停機時間 ⁵ 。	減少 30%
	錯誤率 (Error Rates)	系統錯誤的頻率，按嚴重性分類，以識別重複的技術問題並提高可靠性 ⁵ 。	降低 50%
	異常事件頻率 (Incident Rates)	特定時期內記錄的 IT 事件總數。反映系統可靠性與需改進領域 ⁵ 。	降低 40%
	技術債務減少率 (Technical Debt)	衡量過時或未維護系統中累積成本與低效	減少 25%

	Reduction)	率的減少情況 ⁷ 。	
開發與創新	開發週期時間 (Development Cycle Time)	技術團隊開發、測試和發布新產品或功能所需時間 ⁷ 。	縮短 20%
	自動化測試覆蓋率 (Automated Test Coverage)	程式碼被自動化測試覆蓋的百分比。確保程式碼品質並加速發布 ⁷ 。	提升至 80%
	新功能採用率 (Feature Adoption Rates)	使用者採用新實施工具的百分比。高採用率表示成功推廣和使用者接受度 ⁵ 。	提升至 75%
	程式碼部署頻率 (Code Deployment Frequency)	程式碼部署到生產環境的頻率 ⁷ 。	增加 50%
數據與分析	數據對齊準確性 (Data Alignment Accuracy)	跨系統數據同步、一致和有意義的程度 ²⁰ 。	達到 98%
	數據查詢效能 (Database Query Performance)	資料庫查詢的平均響應時間，確保對關鍵數據的無縫存取 ⁵ 。	提升 30%
業務影響	專案交付率 (Project Delivery Rates)	按時完成專案的百分比 ³⁰ 。	提升至 90%
	客戶滿意度與留存率 (Customer Satisfaction & Retention)	衡量客戶對產品和服務的滿意度，反映客戶體驗和改進領域 ⁵ 。	提升 15%
	業務影響 (Business Impact)	技術實施對營收、利潤、市場份額等業務成果的直接貢獻 ³⁰ 。	顯著正向影響

問責框架與 KPI 契約：

為確保這些 KPI 的達成，將與總經理簽訂一份正式的 KPI 契約，明確責任機制。此框架將建立在以下原則基礎上：

1. **設定清晰期望：**對於每一項任務、專案或職責，明確定義具體交付成果、明確的時間表、品質標準、溝通要求和成功衡量指標³⁰。例如，對於安全審計，需明確要求：「在 3 月 15 日前完成全面系統覆蓋，每週提供進度更新，以標準格式記錄發現結果，並提出最終建議及實施時間表」³⁰。
2. **基於經驗的監督：**針對技術團隊內不同經驗層級的成員，採取量身定制的監督方法。對於初級開發人員，重點關注具體任務和日常目標，每週進行 2-3 次檢查，衡量任務完成度和程式碼品質，並提供直接指導和輔導。對於中級開發人員，則側重於衝刺目標和技術決策，每週檢查一次，衡量專案里程碑和程式碼架構，並提供定期回饋和指導³⁰。
3. **建立問責文化：**作為技術長，必須以身作則，履行承諾、主動溝通、勇於承擔錯誤並始終如一地執行。對於達成目標者給予肯定，對於未達標者則及時處理，提供建設性回饋並制定改進計畫³⁰。確保有清晰的溝通渠道，包括定期狀態更新、開放的挑戰討論、透明的升級路徑和持續的回饋循環³⁰。
4. **管理問責挑戰：**當承諾未能兌現時，應立即進行基於事實的討論，評估影響，分析根本原因，並制定明確的糾正計畫³⁰。同時，應監控並處理重複性延誤、品質問題、溝通缺口和逃避責任等績效模式³⁰。
5. **KPI 契約條款：**契約將明確規定，若未能達成約定的關鍵績效指標，將被視為未履行職責，其後果將被視為自動離職。這將確保高層級的責任感和對目標的堅定承諾。

兩個月內需完成的里程碑

為確保快速啟動並產生初期影響，以下是兩個月內需達成的關鍵里程碑：

- **第一個月：基礎評估與規劃**
 - **數據資產盤點與對齊初探：**完成所有核心系統（APS、UiPath、SAS Viya、Salesforce、MSSQL）的數據資產全面盤點，識別關鍵數據元素及其在各系統中的現有格式與位置。啟動數據對齊的初步分析，識別主要不一致點²⁰。
 - **APS 外部演算法版本控制導入評估：**針對 Opcenter APS 的 5 種外部演算法，完成 Git 版本控制解決方案的技術評估與選型，並制定詳細的導入計畫。
 - **UiPath 郵件處理需求細化：**與業務部門共同細化郵件清理、內容分析、處理等

級分類及自動回覆的具體需求，並啟動相關 AI/ML 模型選型與數據準備工作。

- **整合架構概念設計：**繪製整體系統架構的初步概念圖，標示各系統的核心功能模組及預期的主要整合點。
- **專案團隊組建與職責定義：**確立跨部門的專案核心團隊，明確各成員在整合專案中的角色與職責³¹。
- **第二個月：原型開發與風險識別**
 - **APS 異常郵件自動化通知原型：**開發一個小型原型，利用 UiPath 監控來自 Opcenter APS 相關數據庫的異常數據（而非等待供應商郵件），並自動發送內部會議通知。這將展示從製造排程延伸到內部通知的潛力。
 - **UiPath 郵件分類 AI 模型 POC：**針對特定類型的供應商異常郵件，訓練一個初步的機器學習模型，實現郵件內容的自動分類和處理等級識別，並驗證其準確性¹⁰。
 - **SAS Viya 數據連接器測試：**驗證 SAS Viya 與 MSSQL 資料庫的高效連接，並測試數據提取的穩定性與效能¹¹。
 - **整合前環境檢查清單初稿：**擬定整合前的環境檢查清單初稿，涵蓋數據對齊、安全性、版本控管、測試計畫等關鍵項目³¹。
 - **潛在技術風險識別：**召集技術專家，識別在整合 APS、UiPath 和 SAS Viya 過程中可能遇到的主要技術風險，並初步提出應對策略⁷。

這些里程碑將為後續的全面整合奠定堅實基礎，並在短期內展示技術轉型對業務的潛在價值。

IV. 整合系統架構與開發藍圖

本節將詳細描述整體系統架構，定義 Opcenter APS、UiPath 和 SAS Viya 各自的功能模組及其串接方式，並闡述如何運用 AI 技術加速開發流程。

整體系統架構圖（概念性描述）

為實現資源、時間與工作管理平台的願景，建議的整體系統架構將採用分層、模組化且鬆散耦合的設計原則。核心思想是透過一個中央整合層，將 Opcenter APS、UiPath 和

SAS Viya 這三個獨立系統連接起來，並引入 AI 能力以提升智慧化水平。

概念架構組件：

1. 操作層 (Operational Layer)：

- **Opcenter APS**：作為核心生產排程系統，負責製造工作排程、資源分配、訂單優先級排序等¹。其數據主要儲存在 MSSQL 資料庫中。
- **Salesforce**：作為客戶關係管理系統，儲存客戶與場景分類等關鍵業務數據。
- **現有工作日曆**：包含多個現有日曆系統。

2. 數據層 (Data Layer)：

- **統一數據儲存**：整合來自 APS (MSSQL)、Salesforce 及其他系統的數據，建立一個統一的數據湖或數據倉儲，作為「單一事實來源」。
- **數據對齊與轉換模組**：負責數據清洗、標準化、映射與轉換，確保數據一致性²⁰。

3. 整合層 (Integration Layer)：

- **API 閘道 (API Gateway)**：作為所有外部和內部服務請求的單一入口點，負責路由、負載平衡、認證和授權，並提供統一的 API 介面³⁶。
- **訊息佇列/事件匯流排 (Message Queue/Event Bus)**：實現系統間的異步通信和鬆散耦合。Opcenter APS 的生產事件（如異常、完成）可作為事件發佈，觸發下游系統的響應；UiPath 處理的郵件事件也可發佈至此¹⁸。
- **整合服務/微服務**：開發專門的整合服務或微服務，處理複雜的業務邏輯、數據轉換和系統間的協調。

4. 自動化與智慧層 (Automation & Intelligence Layer)：

- **UiPath RPA 平台**：執行機器人流程自動化，處理郵件自動化（讀取、發送）、數據提取（如 Salesforce 數據）、內部通知發送等⁸。
- **AI/機器學習模組**：內嵌於 UiPath 或作為獨立服務，用於郵件內容分析、分類、情感分析、智能回覆草擬；以及 Salesforce 客戶與場景分類¹⁰。
- **SAS Viya Studio**：進行數據分析、模型開發與部署 (MLOps)、即時串流處理、特徵報表與經營管理分析報告生成¹¹。

5. 應用層 (Application Layer)：

- **統一管理平台介面**：提供一個整合的儀表板或入口網站，用於資源管理、時間管理和工作管理，取代現有分散的工具和日曆。
- **客製化應用程式**：根據業務需求開發的特定應用程式。

功能模組與串接方式

各系統的功能模組將被明確定義，並透過標準化的介面進行串接：

- **Opcenter APS：**

- **功能模組：** 生產排程核心、資源管理、訂單處理、排程優化演算法。
- **串接方式：**
 - **數據庫介接：** 透過 MSSQL 連接器與底層數據庫進行數據交換，讀取生產訂單、物料清單、工作中心、機器中心、庫存等數據，並寫入排程結果³。
 - **API/事件發佈：** 針對生產異常、排程完成、關鍵瓶頸等事件，透過整合層（訊息佇列/事件匯流排）發佈事件，供 UiPath 或 SAS Viya 消費。
 - **外部演算法介面：** 建立標準化的 API 介面，供外部開發的演算法調用 APS 數據並回傳排程建議，以便於版本控制和部署。

- **UiPath：**

- **功能模組：** RPA 機器人、郵件自動化活動、Salesforce 連接器、AI Center（用於 ML 模型）、Orchestrator（用於流程管理與監控）。
- **串接方式：**
 - **郵件系統整合：** 透過內建活動與 Outlook/Gmail 整合，實現郵件的讀取、發送、移動等操作⁸。
 - **AI/ML 整合：** 結合 UiPath AI Center 或外部 ML 服務，對郵件內容進行分析、分類（處理等級）、情感分析，並根據預定義規則或模型生成回覆草稿¹⁰。
 - **Salesforce API 整合：** 利用 UiPath 預建連接器或直接調用 Salesforce REST/SOAP API，實現客戶數據的讀取、寫入、更新和查詢（SOQL），並進行客戶與場景分類¹²。
 - **內部通知與日曆整合：** 透過 UiPath 與統一日曆 API (如 Meetergo, OneCal) 整合，自動發送內部會議通知，並同步各類工作日曆⁴²。
 - **事件消費：** 訂閱來自 Opcenter APS 的生產異常事件，觸發相應的郵件處理或內部通知流程。

- **SAS Viya Studio：**

- **功能模組：** 數據管理（連接器）、數據探索與建模、模型部署（MLOps）、即時串流分析、報表與視覺化。
- **串接方式：**
 - **數據庫連接：** 透過 SAS/ACCESS 介面和 ODBC 驅動程式連接 MSSQL、Azure Data Lake Storage Gen2、Azure Database for MariaDB/MySQL、PostgreSQL、Azure HDInsight、Azure Synapse、Azure SQL Database Managed Instance、Azure SQL Database 和 Azure SQL Server Big Data

Clusters 等多種數據源，進行數據的提取、處理和分析¹¹。

- **即時串流整合：**透過 Azure IoT Hub/IoT Edge 和 Azure Synapse RT Analytics 等服務，實現即時數據串流的攝取與分析，並將分析結果推送至統一數據層或 UiPath 觸發自動化¹¹。
- **模型部署：**利用 SAS Model Manager 將開發的模型部署到 Microsoft Azure 端點，並監控模型性能，確保決策智慧的持續影響¹¹。
- **報表與視覺化：**透過 Power BI (如使用者提及) 或 SAS Viya 內建的視覺化工具，呈現各類特徵報表與經營管理分析報告。

運用 AI 技術加速開發流程

人工智慧技術將在整個軟體開發生命週期中扮演關鍵角色，從程式碼生成到測試與數據管理，顯著提升效率並加速產品上市時間²⁰。

- **GPT/Codex 於程式碼生成：**

- **加速整合程式碼開發：**GPT 或 Codex 等大型語言模型能夠根據自然語言描述或現有程式碼上下文，自動生成整合邏輯、API 調用腳本、數據轉換函數等樣板程式碼⁴⁷。這對於連接不同系統的 API 和數據格式轉換尤為有用，可大幅減少開發人員編寫重複性程式碼的時間。
- **客製化演算法輔助開發：**針對 Opcenter APS 的外部演算法，AI 可輔助生成演算法的骨架、優化建議或測試用例，加速其開發與迭代。
- **自動化測試程式碼生成：**AI 可根據功能需求和系統行為自動生成單元測試、整合測試的程式碼，確保新開發的整合模組的品質和穩定性⁴⁷。
- **文件與註釋生成：**AI 可自動生成程式碼註釋和 API 文件，提高程式碼的可讀性和可維護性，這對於跨系統協作的整合專案至關重要⁴⁷。
- **挑戰與應對：**儘管 AI 在程式碼生成方面具有巨大潛力，但仍需注意其生成程式碼的準確性、安全性和潛在的抄襲風險⁴⁸。因此，開發人員必須對 AI 生成的程式碼進行嚴格審查、測試和驗證，確保其符合企業級標準和安全要求。AI 應被視為開發人員的「副駕駛」，而非完全取代人工。

- **機器學習應用：**

- **智慧郵件分類與響應：**運用機器學習模型對 UiPath 接收的郵件進行深度內容分析，實現自動分類（例如，按「處理等級」分類）、提取關鍵資訊，並根據分類結果自動草擬標準化回覆。這將大幅提升郵件處理效率，減少人工干預，並確保回應的一致性與及時性¹⁰。

- **自動化數據映射：**針對不同系統間的數據對齊問題，利用機器學習自動化數據映射過程。AI 模型可以分析源數據和目標數據的結構、模式和語義，自動推薦或生成數據映射規則，從而簡化 ETL/ELT 流程，減少人工錯誤，並加速數據整合⁵¹。這對於處理來自 APS 的製造數據與 Salesforce 的客戶數據的整合尤為關鍵。
- **自動化測試與缺陷預測：**機器學習可用於分析歷史測試數據和程式碼變更，自動生成測試用例、優先級排序測試執行，甚至預測潛在的缺陷區域⁴⁹。在複雜的系統整合專案中，AI 驅動的自動化整合測試可以顯著提高測試效率和覆蓋率，確保各模組間的無縫協同運作。

1. 可用於獨自開發的免費開源軟體建議

考量到成本效益與開發彈性，除了公司已有的商業軟體外，我們可以導入以下免費開源工具來輔助開發，以應對藍圖中提到的挑戰（如版本控制、系統整合等）：

- **版本控制系統：Git**
 - **用途：**解決目前 Opcenter APS 外部演算法缺乏版本控制的重大問題。透過導入 Git，我們可以追蹤每一次程式碼的變更⁴、促進團隊協作（即使目前只有一人，也應為未來擴張做準備），並在發生問題時快速回溯至穩定版本⁴。
 - **搭配平台：**可以搭配使用 GitLab 或 GitHub 的免費方案，它們都提供了私有儲存庫、專案管理看板 (Issue Tracker) 和 CI/CD (持續整合/持續部署) 功能，能實現演算法的自動化測試與部署。
- **整合與自動化流程 (CI/CD)：Jenkins**
 - **用途：**建立自動化的部署流程，特別是針對客製化演算法和整合腳本。例如，當新的 APS 演算法程式碼推送到 Git 儲存庫後，能自動觸發測試與部署，減少人工操作的風險。
 - **優勢：**擁有龐大的插件生態系，可以輕易地與 Git、Docker 等工具整合。
- **資料庫與訊息傳遞：PostgreSQL & Apache Kafka**
 - **PostgreSQL：**作為一個強大且免費的開源資料庫，可用於建立統一的數據層 (Data Warehouse/Lake)，匯集來自 MSSQL、Salesforce 等不同系統的資料¹¹。
 - **Apache Kafka：**作為事件驅動架構 (EDA) 的核心，用於在 Opcenter APS、UiPath 和 SAS Viya 之間進行即時、非同步的訊息傳遞¹⁸。例如，APS 產生異常事件後，可發布一個訊息到 Kafka，UiPath 再訂閱此訊息並觸發後續的自動化流程¹⁸。

- 容器化技術：Docker
 - 用途：將開發的應用程式（例如：用於串接 API 的微服務）打包成標準化的容器，確保開發、測試和生產環境的一致性，解決「在我的電腦上可以跑」的傳統問題。這對於未來管理複雜的 Kubernetes 環境也至關重要。
- 統一儀表板與視覺化：Grafana
 - 用途：藍圖中提到要建立一個統一的管理平台介面。Grafana 可以連接到多種資料來源（如 PostgreSQL、MSSQL），建立一個整合的監控儀表板，即時呈現來自各系統的 KPI 數據，例如生產排程狀態、自動化流程成功率等。

開源軟體 vs. SAS Viya 功能對應與整合策略

之所以建議導入一些開源軟體，並非要取代 SAS Viya 的功能，而是要將它們作為**「通用黏著劑」**，把公司現有的三大獨立系統（Opcenter APS、UiPath、SAS Viya）以及未來可能加入的系統（如 Salesforce）無縫地黏合起來，建立一個具備彈性、可擴展且廠商中立的整合架構。

以下是開源軟體與 SAS Viya 功能對應清單 (Mapping List)，詳細說明了各自的角色分工與整合策略：

功能領域 (Functional Area)	建議的開源軟體 (Suggested Open Source Software)	SAS Viya 內建功能 (SAS Viya's Built-in Capability)	整合策略與理由 (Integration Strategy & Rationale)
版本控制 (Version Control)	Git / GitLab / GitHub	SAS Studio 支援與 Git 儲存庫的整合，可用於管理 SAS 程式碼和專案 ¹¹ 。	角色分工：SAS Viya 處理「自身」的程式碼版本控制。但我們的核心痛點是**「Opcenter APS 的外部演算法缺乏版本控制」**。Git 將作為一個通用的版本控制平台，統一管理專案中所有的程式碼資產，包括 APS 演算法、

			UiPath 腳本以及串接各系統的整合程式，實現跨平台的變更追蹤與協同作業 ⁴ 。
持續整合與部署 (CI/CD)	Jenkins	SAS Viya 提供 CLI (命令列介面) 和 REST API，可以被外部的 CI/CD 工具調用來執行任務 (如模型訓練、報表生成) ¹¹ 。	協同運作： SAS Viya 負責「執行」分析任務，而 Jenkins 擔任「總指揮」的角色。我們需要一個能跨系統調度的工具來自動化「APS 產生數據 → UiPath 觸發通知 → SAS Viya 進行分析」的完整流程。Jenkins 正好能扮演這個角色，串連起整個端到端的自動化工作流。
數據視覺化與儀表板 (Visualization)	Grafana	SAS Visual Analytics 是一個強大的商業智慧 (BI) 與數據探索工具，用於製作深度分析報表 ¹¹ 。	不同用途： SAS Visual Analytics 主要用於**「商業分析與決策」。而 Grafana 則用於建立「技術維運監控儀表板」**，即時監控整個整合系統的健康狀態，例如：API 的回應時間、Kafka 訊息隊列的長度、UiPath 流程的成功/失敗率等。兩者目標不同，功能互補。
中央數據儲存 (Data Storage)	PostgreSQL	SAS Viya 本身是一個分析平台，它會「連接」到各種數據來源 (如 Azure SQL, Data Lake) ¹¹ ，但它不是一個傳統的關聯式資料庫。	架構定位： 為了徹底解決「資料庫的內容對齊」問題，建議使用 PostgreSQL 建立一個統一的數據倉儲或數據湖。這個中央儲存庫會先匯集、清洗並整合來自

			Opcenter APS (MSSQL) 和 Salesforce 的數據，再提供一個乾淨、一致的「單一事實來源」給 SAS Viya 進行分析 ²¹ 。
事件驅動與訊息傳遞	Apache Kafka	SAS Viya 能夠透過其 Event Stream Processing 引擎連接到 Kafka，即時處理串流數據 ¹¹ 。	核心骨幹： SAS Viya 是 Kafka 的「消費者」或「生產者」之一，但它不是訊息傳遞中介 (Message Broker) 本身。Kafka 將作為整個事件驅動架構 (EDA) 的核心骨幹，負責在 APS、UiPath 和 SAS Viya 之間可靠、非同步地傳遞事件訊息，實現系統間的鬆耦合 (loosely coupled) ¹⁸ ，這對於建立可擴展的即時響應系統至關重要。
應用程式容器化 (Containerization)	Docker	SAS Viya 平台本身就是基於 Kubernetes 部署的，而 Kubernetes 的基礎就是容器技術 (如 Docker) ¹¹ 。	生態系一致性： 我們不會用 Docker 來打包 SAS Viya，而是用它來打包我們為了整合而開發的**「客製化輔助元件」**（例如：一個用 Python 寫的、用於封裝舊版 API 的微服務）。這樣可以確保我們自己開發的元件能以和 SAS Viya 同樣標準化的方式，在同一個 Kubernetes 環境中運行和管理。

2. 兩個月的開發與部署文件清單

這份清單涵蓋了從啟動到交付所需的關鍵文件，確保專案的可追溯性、可維護性與可驗收性。

第一個月 (第 1-4 週)：規劃、設計與基礎建設

- **【文件】專案範疇與目標定義書**：明確本次專案的邊界、核心目標與成功標準³²。
- **【文件】現有系統架構圖**：包含 Opcenter APS、UiPath、SAS Viya 目前的部署架構與資料流。
- **【文件】數據資產盤點與字典**：詳列所有核心系統的關鍵數據欄位、格式與業務意義²¹。
- **【文件】Git 版本控制導入計畫**：包含 Git 流程 (branching strategy)、權限管理與針對 APS 演算法的導入步驟⁴。
- **【文件】整合架構設計草圖**：繪製包含 API Gateway、事件驅動核心 (Kafka) 和各系統模組的初步架構圖³²。
- **【文件】API 與 Atom 元件架構圖**：完成合約要求，定義系統間的 API 介面與資料交換格式。
- **【文件】郵件處理需求規格書**：與業務部門共同細化 UiPath 郵件清理、分類與回覆的具體規則。

第二個月 (第 5-8 週)：開發、測試與交付

- **【文件】三系統串接測試計畫與測試案例**：詳述如何測試 APS -> UiPath -> SAS Viya 的端到端流程，包含正常與異常情境³²。
- **【文件】每日自動異常通知與排程執行紀錄範例**：提供實際產出的通知與紀錄樣本。
- **【文件】部署手冊 (Deployment Guide)**：詳述如何將開發的各個元件部署到伺服器上。
- **【文件】內部 Prompt 可複用範本集**：整理供跨部門每日追蹤進度使用的標準化 Prompt。
- **【文件】完整技術文件**：包含所有程式碼的註解、API 的使用說明以及系統的維運指南。
- **【文件】甘特圖 (Gantt Chart)**：以視覺化方式呈現專案時程、任務依賴關係與目前進度。
- **【報告】《三年業績技術架構報告書》**：完成合約要求的最終報告，闡述此基礎如何支撐未來 NT\$50,000,000 的業績目標。
- **【報告】專案結案報告與 KPI 效益展示**：總結專案成果，並以數據展示自動化流程所

帶來的 KPI 減時效益。

3. 每週工作進度項目清單

這是一個基於合約與藍圖里程碑的詳細週計畫。

- **第一週：專案啟動與需求盤點**
 - 召開專案啟動會議，確認目標與範疇。
 - 完成核心系統 (APS, UiPath, SAS Viya, Salesforce, MSSQL) 的數據資產全面盤點³²。
 - 啟動數據對齊的初步分析，識別主要不一致點²¹。
 - 完成 Git 版本控制方案的技術評估與選型⁴。
- **第二週：架構設計與環境準備**
 - 完成 APS VM 演算法模組、MSSQL 結構清單與更新流程文件。
 - 繪製整體系統整合架構的初步概念圖³²。
 - 與業務部門共同細化 UiPath 郵件清理、分類的具體需求。
 - 完成跨系統整合 API 與 Atom 元件架構圖。
- **第三週：UiPath 原型開發**
 - 建立 UiPath 郵件清理、異常通知分類、會議通知模組的原型流程。
 - 完成 SMTP/IMAP 權限設定。
 - 啟動相關 AI/ML 模型（用於郵件分類）的選型與數據準備工作¹⁰。
- **第四週：SAS Viya 與整合設計**
 - 製作 SAS Viya 分析模組、定時排程流程與命名空間設定文件。
 - 設計 APS -> UiPath -> SAS Viya 三系統串接的詳細流程。
 - 提交第一階段的 Notion 週報與成果。
- **第五週：三系統串接原型開發**
 - 實際開發 APS -> UiPath -> SAS Viya 的三系統串接測試流程。
 - 開發一個小型原型，利用 UiPath 監控 APS 數據庫的異常數據。
 - 驗證 SAS Viya 與 MSSQL 資料庫的高效連接¹¹。
- **第六週：自動化與測試**
 - 完成每日自動異常通知與排程執行紀錄的範例程式。
 - 訓練一個初步的 UiPath 郵件分類 AI 模型 POC¹⁰。
 - 撰寫整合測試案例，並開始執行端到端測試³²。
- **第七週：報告撰寫與文件整理**

- 撰寫並完成《三年業績技術架構報告書》草稿。
- 完成內部 Prompt 可複用範本。
- 整理並提供完整的技術文件與甘特圖。
- **第八週：驗收、展示與總結**
 - 準備並進行專案結束時的成果展示，包含可運行的流程與 KPI 減時效益分析。
 - 完成所有專案文件，並提交管理層驗收。
 - 若交付成果達 80%，準備結案；若未達，則準備檢討會議。

4. 第一個月的討論清單

為了確保第一個月的基礎工作順利，需要與相關部門討論以下議題：

- **數據治理與權限 (第一週)**
 - **議題：** 確認各系統 (特別是 MSSQL、Salesforce) 的數據庫存取權限。
 - **討論對象：** CTO
 - **目標：** 取得必要的讀取權限，以便進行數據盤點與後續的資料整合。
- **APS 演算法版本控制 (第一週)**
 - **議題：** 確認導入 Git 作為 APS 演算法版本控制的方案是否可行？是否有任何潛在的供應商合約限制？
 - **討論對象：** CTO
 - **目標：** 獲得批准，立即開始為 5 種外部演算法建立 Git 儲存庫⁴。
- **UiPath 郵件處理規則 (第二週)**
 - **議題：** 目前郵件處理的痛點是什麼？「處理等級」的具體分類標準為何？希望自動化回覆的郵件範本樣式？
 - **討論對象：** 業務部門、營運部門
 - **目標：** 將模糊的需求（如「清理郵件」）轉化為明確、可執行的程式邏輯。
- **系統整合的安全性考量 (第三週)**
 - **議題：** 各系統間的 API 呼叫是否需要遵循特定的安全規範？例如 API Key 或 OAuth 認證？
 - **討論對象：** CTO
 - **目標：** 在架構設計初期就將資安需求納入考量，避免後期重構⁴。
- **SAS Viya 命名空間與角色權限 (第四週)**
 - **議題：** 根據合約，需確認 SAS Viya 的角色與命名空間安全隔離。我們應如何劃分不同專案或部門的命名空間？

- 討論對象：CTO
- 目標：建立清晰的權限劃分，確保數據分析環境的整潔與安全。

V. 整合前待辦清單與潛在技術風險

在啟動大規模系統整合之前，必須進行全面而細緻的準備工作，以確保專案的順利進行並降低潛在風險。本節將列出整合前的環境檢查項目、數據準備事項、安全考量、版本控制策略及測試計畫，並識別可能面臨的技術風險及其緩解策略。

環境與數據準備

成功的系統整合始於對現有環境和數據的充分準備。

- **數據對齊與標準化：**
 - **數據審計：** 首先，必須對所有數據源進行全面的數據審計，以識別跨系統存在的
不一致性、冗餘和數據缺口²⁰。這包括 Opcenter APS 的 MSSQL 數據、
Salesforce 客戶數據、SAS Viya 相關數據以及其他內部系統數據。
 - **定義數據標準：** 建立清晰的數據輸入、處理和儲存規範，以確保數據在整個企業
範圍內的一致性²⁰。這將涉及定義統一的數據格式、命名約定和業務規則。
 - **自動化數據映射工具：** 考慮利用現代自動化工具來簡化數據映射、模式匹配和標
準化過程²⁰。這將減少人工錯誤，並加速數據整合的效率。
 - **數據治理框架：** 實施強健的數據治理政策和工作流程，以確保數據品質和一致性
能夠長期維持²⁰。
- **系統準備：**
 - **兼容性驗證：** 驗證所有待整合系統（Opcenter APS、UiPath、SAS Viya、
Salesforce）之間的兼容性，包括硬體、軟體、網路基礎設施和數據儲存要求³¹。
 - **必要的升級與修補：** 在整合前，確保所有相關系統都已執行必要的系統升級和安
全修補，以避免潛在的兼容性問題或安全漏洞。
 - **數據清洗與備份：** 在數據遷移或同步之前，對數據進行清洗，移除重複、不完整
或不準確的記錄。同時，執行全面的數據備份，以防在整合過程中發生數據丟失

或損壞³¹。

安全態勢與控制

軟體整合專案中的安全性至關重要，必須實施多項最佳實踐以保護數據和系統。

- **使用安全通訊協定：** 確保所有整合點使用 HTTPS、TLS (傳輸層安全) 或 SSL (安全套接層) 等安全通訊協定，對傳輸中的數據進行加密，防止數據被攔截或篡改⁴⁶。
- **保護 API 和端點：** API 和端點是數據交換的關鍵入口。必須對 API 調用進行身份驗證和授權，並使用安全的存取令牌，如 OAuth 令牌或 API 密鑰，以防止未經授權的存取⁴⁶。
- **定期安全審計與審查：** 定期對整合系統進行安全評估、審計和滲透測試，以識別和解決漏洞。及時應用安全補丁和更新以緩解已識別的風險⁴⁶。
- **限制數據暴露：** 僅暴露整合所需的最低限度數據，顯著降低數據洩露的風險。採用數據最小化、數據匿名化或假名化技術來保護敏感資訊⁴⁶。
- **建立錯誤處理機制：** 確保適當的錯誤處理機制，避免敏感資訊透過錯誤訊息洩露。提供通用但有意義的錯誤響應，以防止攻擊者獲取系統內部運作的線索⁴⁶。
- **最小權限原則：** 遵循最小權限原則，僅向個人或系統授予執行其任務所需的最低存取權限，從而限制因事故或攻擊造成的潛在損害⁵³。
- **員工與終端使用者培訓：** 定期對員工和終端使用者進行安全培訓，提高他們對潛在安全威脅的認識以及遵守最佳實踐的重要性⁴⁶。

版本控制與測試計畫

在複雜的企業整合專案中，嚴謹的版本控制和全面的測試計畫是確保品質和穩定性的基石。

- **版本控制：**
 - **對客製化演算法實施 Git：** 針對 Opcenter APS 的 5 種外部開發演算法，必須立即導入 Git 等分散式版本控制系統⁴。這將允許追蹤每一次程式碼變更、回溯到任何歷史版本、促進團隊協作，並為未來的維護和迭代提供堅實基礎。
 - **統一版本控制策略：** 確保所有客製化開發（包括 UiPath 流程、SAS Viya 腳本、整合服務程式碼）都納入統一的版本控制系統。採用分支策略（例如，功能分

支、整合分支、主分支）來管理開發、測試和生產環境的程式碼流。在部署時，對每個版本進行標籤，確保可追溯性⁴。

- **測試計畫：**
 - **定義測試範圍與目標：** 明確整合測試的範圍，包括哪些系統、模組和數據流將被測試，以及測試的具體目標（例如，數據準確性、系統響應時間、錯誤率）³¹。
 - **測試類型與方法：** 涵蓋單元測試、整合測試、效能測試和使用者的驗收測試。特別強調自動化整合測試，以驗證不同軟體組件之間的無縫互動³¹。自動化測試可以顯著提高效率，並在開發早期發現問題⁵⁰。
 - **測試案例與場景：** 根據系統需求和業務流程，設計詳細的測試案例和場景，包括預期輸入、條件和結果。
 - **資源分配與時間表：** 明確測試團隊成員的角色、所需的工具和測試環境。制定詳細的測試時間表，包括關鍵里程碑和截止日期。
 - **成功標準與報告：** 建立清晰的測試成功標準，例如可接受的通過率或最大缺陷數。定義測試結果的記錄、追蹤和溝通方式。

潛在技術風險與緩解策略

在整合複雜的企業軟體系統時，將面臨多種技術風險。預先識別並制定緩解策略至關重要。

潛在技術風險	具體描述	緩解策略		
兼容性與整合挑戰	來自不同供應商的軟體（APS、UiPath、SAS Viya）可能存在數據格式不匹配、API 兼容性問題、協議差異等，導致整合困難 ⁷ 。	綜合規劃與策略開發： 制定詳細的整合藍圖，包括明確的目標、時間表、里程碑和依賴關係 ³⁴ 。	徹底的兼容性測試：在實施前，對所有工具和系統與現有應用程式和基礎設施的兼容性進行測試，及早識別並解決問題 ³⁴ 。	標準化介面： 優先使用標準化介面（如 RESTful API、ODBC）和通用數據格式（如 JSON, XML）。
數據不一致性與	整合過程中可能導致數據重複、	數據對齊策略： 實施數據清洗、	數據治理： 建立	

重複	數據遺失或數據不一致，影響數據完整性和決策準確性 ³⁴ 。	標準化和數據映射技術，確保數據在源系統和目標系統之間的一致性 ²⁰ 。	數據所有權、數據品質標準和數據驗證流程。	
網路與 Kubernetes 控管複雜性	大規模 Kubernetes 環境的部署、管理和網路配置複雜，可能導致 SAS Viya 與其他軟體隔離，或影響系統穩定性和可擴展性 ¹⁹ 。	專業知識與培訓：投資於 Kubernetes 專業知識的內部團隊，或尋求外部專家協助 ¹⁹ 。	雲端託管服務：考慮利用雲端供應商提供的託管 Kubernetes 服務（如 Azure AKS），以抽象化部分底層複雜性 ¹¹ 。	網路分段與監控：設計清晰的網路分段策略，並實施全面的網路監控工具，及早發現並解決網路瓶頸或配置錯誤。
客製化演算法管理困難	Opcenter APS 的外部演算法缺乏版本控制，更新困難，且難以追蹤變更，導致維護成本高、風險大【使用者查詢】。	導入版本控制系統：立即為所有客製化演算法導入 Git 等版本控制系統，並建立標準的程式碼管理流程（分支、合併、程式碼審查） ⁴ 。	自動化部署流程：建立 CI/CD 管道，實現演算法的自動化測試和部署，減少人工錯誤。	
RPA 機器人運作風險	UiPath 機器人可能因底層流程變更或意外錯誤而故障，導致「中斷-修復」循環，限制 RPA 潛力 ³⁵ 。	健壯的錯誤處理：在 UiPath 流程中實施強健的錯誤處理機制，確保機器人能夠從意外問題中恢復，而無需人工干預 ¹⁵ 。	持續監控與警報：實施實時監控系統，追蹤機器人性能，並在出現異常時觸發自動警報 ¹⁵ 。	流程變更管理：建立嚴格的流程變更管理協議，確保任何底層業務流程的變更都能及時反映在 RPA 流程中，並進行充分測試。
網路安全風險與數據暴露	整合系統增加了數據洩露、未經授權存取或憑證盜竊的風險，尤其是在 RPA 機器人具有特權存取	強化安全政策：更新現有安全政策以符合新的 IT 基礎設施，並實施加密演算法、存取控制和數據	特權存取管理 (PAM)：嚴格管理 RPA 機器人的特權存取，並定期審計其活動。 安全審計：定期	

	權限的情況下 ³⁴ 。	丟失預防措施 ³⁴ 。	進行安全審計和滲透測試，以識別和解決漏洞。
員工抗拒與技能差距	員工可能抗拒新技術的導入，或缺乏操作新系統所需的技能，導致採用率低和效率低下 ³⁴ 。	有效的變革管理：透過清晰溝通、展示效益和早期成功案例來促進員工接受度 ³⁴ 。	充分的培訓與技能發展：提供全面的培訓計畫和技能發展課程，確保員工能夠有效協作並充分利用自動化能力 ³⁴ 。

VI. 原子化串接策略與統一平台

為實現各系統的無縫協同運作，本報告建議採用原子化串接策略，將各系統的最小可單獨運作單元（atom）連接起來。這將透過事件驅動架構、API 優先方法和集中式數據管理來實現，最終建立統一的資源、時間和工作管理平台。

事件驅動架構（EDA）實現即時協調

事件驅動架構（EDA）是一種軟體設計模式，允許系統偵測、處理、管理並即時響應事件¹⁸。其核心優勢在於促進系統組件之間的鬆散耦合，從而實現獨立開發與部署、提高可擴展性和容錯能力，並與外部系統無縫整合¹⁸。

- 應用場景：
 - **Opcenter APS 事件**：當 Opcenter APS 偵測到生產異常（例如，機器故障、物料短缺、排程延誤）或完成關鍵排程步驟時，將發佈相應的「生產事件」。這些事件將被推送到中央事件匯流排（如 Apache Kafka），作為最小可運作單元。
 - **UiPath 觸發**：UiPath 流程將訂閱這些「生產事件」。例如，當接收到「生產異常事件」時，UiPath 可自動啟動郵件處理流程，分析異常郵件內容，並根據預定義的處理等級自動發送內部會議通知給相關人員，取代目前的人工郵件通知模式。同時，UiPath 也可以將處理結果發佈為事件，供 SAS Viya 進行後續分析。

- **Salesforce 數據同步：** Salesforce 中的客戶數據更新（如客戶類別變更、新場景創建）可觸發事件，由 UiPath 捕獲並同步至統一數據層，供 SAS Viya 進行即時分析和分類。
- **SAS Viya 數據流：** SAS Viya 可訂閱來自 Opcenter APS 或其他操作系統的即時數據流事件，進行即時分析和模型評分，並將分析結果（如預測性維護警報、優化建議）發佈為事件，供 UiPath 或其他操作系統消費。
- **技術組件：** 採用事件代理（如 Apache Kafka）作為核心，結合微服務架構和事件驅動程式設計。微服務將作為獨立部署的服務，透過標準協議進行通信，確保靈活性和可擴展性¹⁸。

API 優先方法與微服務整合

API 優先的開發方法將確保各系統間的互操作性和數據交換的靈活性。

- **Opcenter APS 功能暴露：** 儘管 Opcenter APS 的外部演算法目前缺乏 Git 版本控制，但應探索並建立標準化的 API 介面，以暴露其核心功能（如排程更新、異常警報）和數據。如果現有版本不直接支持 API，則可透過數據庫層的 API 封裝或輕量級微服務來實現。
- **UiPath 作為整合協調者：** UiPath 不僅執行 RPA，還將作為關鍵的整合協調者。它將：
 - **消費 API：** 調用 Opcenter APS 提供的 API 以獲取排程狀態，或調用 Salesforce API 進行數據讀取和寫入¹²。
 - **暴露 API：** 將其智慧郵件處理、內部通知發送等能力包裝成 API，供其他系統（如統一管理平台）調用。
- **SAS Viya 的 API 能力：** SAS Viya 能夠透過其 API 介面消費實時數據流，並暴露其分析模型和數據洞察。例如，SAS Viya 的預測模型可透過 API 供 Opcenter APS 或統一管理平台調用，以獲取優化建議。
- **API 閘道模式：** 考慮採用 API 閘道模式，作為所有 API 請求的單一入口點³⁶。這有助於集中管理 API、實施安全策略、進行流量控制和監控，並簡化客戶端與後端服務的互動。可選擇集中式邊緣閘道、兩層式閘道或微閘道模式，具體取決於安全與可擴展性需求³⁶。

集中式數據管理與同步

建立一個統一且一致的數據視圖是實現「資源管理平台、時間管理平台、工作管理平台」的基礎。

- **主數據管理 (MDM) 策略：** 制定 MDM 策略，識別並管理企業中的關鍵主數據（如產品、客戶、員工、工作站、供應商），確保其在所有系統中的唯一性、準確性和一致性。
- **自動化數據映射與轉換：** 利用 AI 驅動的自動化數據映射工具，簡化並加速數據從源系統（如 Opcenter APS 的 MSSQL 資料庫、Salesforce）到統一數據層的提取、轉換和載入（ETL/ELT）過程⁵¹。這將解決「資料庫的內容對齊」問題，確保數據在不同系統間的語義一致性。
- **數據同步機制：** 根據業務需求，選擇合適的數據同步機制。對於高頻變更的數據，採用事件驅動的實時同步；對於低頻變更或大批量數據，採用批次同步或定時同步。這將確保所有系統始終使用最新且一致的數據。

統一通訊與協作

解決非資訊部門同事偏好 Line 導致的「資訊不同不與混淆」問題，並統一多個工作日曆，是提升協作效率的關鍵。

- **UiPath 整合通訊平台：** 探索 UiPath 與企業級通訊平台（如 Microsoft Teams、Slack，或考慮 Line 的企業級 API 介面）的整合⁷。透過 UiPath 發送自動化通知、警報和報告到這些平台，將資訊直接推送給非資訊部門同事，減少資訊孤島。
- **統一工作日曆整合：** 採用統一日曆解決方案（如 Meetergo、OneCal 或 Cronofy）⁴²。這些解決方案能夠同步來自 Google Calendar、Outlook、Apple Calendar 等多個日曆平台的數據，並提供單一的整合視圖。UiPath 可利用其 API 介面，自動將 Opcenter APS 相關的內部會議通知（例如，異常處理會議、排程審查會議）直接排入統一工作日曆，並發送提醒。這將消除「現行的工作日曆就有四個以上」的混亂局面，顯著提升時間管理效率。

整合資源、時間與工作管理平台

最終目標是將所有系統整合為一個全面的管理平台，實現對「人的工時、半成品、組裝成品的完成時間、工作站內部與外包協調方式」的統一管理。

- **單一管理介面：** 開發一個中央儀表板或入口網站，作為資源、時間和工作管理的單一介面。這個介面將匯總來自 **Opcenter APS**（生產排程、資源利用率）、**UiPath**（自動化流程狀態、郵件處理進度）、**SAS Viya**（業務分析報告、預測洞察）和 **Salesforce**（客戶分類、銷售場景）的關鍵數據。
- **數據驅動的決策循環：** 透過整合數據，實現從「觀察」到「思考下一步對策」的閉環正向工作循環。例如，**SAS Viya** 分析的生產效率數據（基於 **Opcenter APS** 的工時、半成品、成品時間）可以直接在統一管理平台中呈現，並觸發 **UiPath** 自動化流程，調整排程或發送協調通知給內部或外包團隊。
- **智慧化協調：** 結合 **AI/ML** 進行預測性分析，例如預測潛在的生產瓶頸、資源衝突或郵件處理延遲，並自動建議優化方案或觸發預防性措施。這將使平台從被動反應轉變為主動管理。

VII. 結論與建議

本報告闡述了將 **Opcenter APS**、**UiPath** 和 **SAS Viya Studio** 這三個獨立的企業軟體系統整合為一個統一、智慧化平台的策略與技術藍圖。現有的碎片化操作導致了顯著的營運效率低下和「搜尋時間成本」浪費，阻礙了企業的敏捷性和數據驅動決策能力。

透過實施本報告所提出的整合策略，本公司將能夠：

- **提升營運效率：** 透過自動化郵件處理、統一排程和數據流，顯著減少人工干預和重複性工作，從而降低營運成本並加速業務流程。
- **強化數據決策：** 建立統一的數據層和強大的分析能力，將分散的數據轉化為可操作的業務洞察，實現更精準的生產預測、客戶管理和經營分析。
- **加速創新與市場響應：** 運用 **AI** 輔助開發、自動化測試和模型部署，加快新功能和客製化演算法的開發速度，使公司能夠更迅速地響應市場變化和客戶需求。
- **建立協同工作環境：** 統一溝通介面、工作日曆和管理平台，打破部門間的資訊孤島，促進跨職能團隊的無縫協作，提升整體組織的生產力。
- **實現策略性成長：** 從單純的成本節省轉向價值創造，透過技術賦能業務，實現營收增長目標，並在競爭激烈的市場中建立差異化優勢。

行動建議：

1. **啟動跨職能整合專案：**立即組建由技術、營運、銷售和數據分析部門組成的核心專案團隊，並指派專案經理，推動本藍圖的實施。
2. **簽訂 KPI 契約：**按照本報告提出的 KPI 框架，與總經理簽訂具約束力的 KPI 契約，明確各項指標的責任人與達成目標。
3. **優先處理基礎整合與數據對齊：**聚焦於前兩個月的里程碑，優先解決 Opcenter APS 外部演算法的版本控制問題，並啟動數據對齊和郵件分類自動化的原型開發。
4. **投資技術人才與培訓：**針對 Kubernetes 管理、AI/ML 開發與應用、以及整合架構設計等關鍵領域，加強內部技術團隊的培訓與能力建設。
5. **建立持續改進機制：**實施持續監控、定期審查和回饋循環，確保整合專案能夠適應不斷變化的業務需求和技術環境。

此轉型藍圖不僅是技術上的升級，更是公司營運模式和策略思維的全面革新。透過堅定不移的執行與持續投入，本公司將能夠建立一個真正智慧化、高效協同的數位化企業，為未來的持續成長奠定堅實基礎。

兩個月工作契約

一、契約目的

整合公司代理之 Opcenter APS、UiPath、SAS Viya Studio，建立跨系統自動化運行原型流程，驗證落地可行性，作為未來三年內達成 NT\$50,000,000 業績之核心技術基礎。未完成時視同辦理自動離職。

二、契約期間

簽核日起 60 天內執行，預計完成日期：YYYY/MM/DD。

三、執行目標與交付項目

第一階段（第 1-4 週）

- 完成 APS VM 演算法模組、MSSQL 結構清單與更新流程文件。
- 建立 UiPath 郵件清理、異常通知分類、會議通知模組原型流程。
- 製作 SAS Viya 分析模組、定時排程流程與命名空間設定文件。
- 完成跨系統整合 API 與 Atom 元件架構圖。

第二階段（第 5-8 週）

- 實際完成 APS -> UiPath -> SAS Viya 的三系統串接測試流程。
- 完成每日自動異常通知與排程執行紀錄範例。
- 撰寫並完成《三年業績技術架構報告書》。
- 完成內部 Prompt 可複用範本，供跨部門每日追蹤進度。
- 提供完整技術文件與甘特圖供管理層驗收。

四、檢核與驗收

- 每週固定進度會議，提交 Notion 週報。
- 專案結束時展示可運行流程與自動化 KPI 減時效益。
- 交付成果達 80% 可視為完成，未達時 CTO 負責提出檢討與調整或啟動離職辦理。

五、注意事項

- **技術準備：** APS 無 Git 版本控管需以檔案編號與 Changelog 管理，UiPath 郵件串接需先完成 SMTP/IMAP 權限設定，SAS Viya 須確認角色與命名空間安全隔離。
- **流程驗證：** 每日測試至少 1 次跨系統流程串接，收集錯誤紀錄。
- **營收驗證：** 量化 APS 排程與 UiPath 自動通知節省工時，建立作為未來向製造業客戶導入之商業模式依據。
- **AI 協作：** 每日使用 GPT 協助生成自動回覆、架構紀錄、範本重構，確保加速開發。

六、合約條款

- 本契約自簽核日起生效，執行 60 天。
- 完成交付後，提交總經理進行 KPI 檢核與展示。
- 若未達標準，雙方於 5 日內完成結案或啟動離職程序。

七、簽署

總經理簽章

CTO 簽章

日期：

日期：

預計與實際使用的 GPT Token 數量：

- 預計 Token 數量：10000-15000 tokens
- 實際使用 Token 數量：（此處將於生成後填寫）

引用的著作

1. Opcenter Advanced Planning and Scheduling (APS) software - Siemens PLM, 檢
索日期：7 月 23, 2025 , <https://plm.sw.siemens.com/en-US/opcenter/advanced-planning-scheduling-aps/>
2. snicsolutions.com, 檢索日期：7 月 23, 2025 ,
<https://snicsolutions.com/platform/opcenter/opcenter-aps#:~:text=Can%20Opcenter%20APS%20integrate%20with,and%20shop%20floor%20control%20systems.>
3. GAC Siemens Opcenter APS Interface - Find the right app | Microsoft AppSource, 檢索日期：7 月 23, 2025 , <https://appsource.microsoft.com/en-us/product/dynamics-365-business-central/pubid.gac%7Caid.gacopcenterinterface%7Cpappid.157f73a9-d163-48ca-828f-5b902c626a8b?tab=overview>
4. Version Control for ML Models: What It Is and How To Implement It, 檢索日期：7 月 23, 2025 , <https://neptune.ai/blog/version-control-for-ml-models>
5. IT Metrics: 21 KPIs to Track IT Strategy Performance - Lumos, 檢索日期：7 月 23, 2025 , <https://www.lumos.com/topic/it-strategy-it-metrics>
6. Key KPIs for CTO Success in Tech Rollouts - MoldStud, 檢索日期：7 月 23, 2025 , <https://moldstud.com/articles/p-key-kpis-for-cto-success-in-tech-rollouts>
7. 20 KPIs Every Chief Technology Officer(CTO) Should Monitor [2025 ..., 檢索日期：7 月 23, 2025 , <https://digitaldefynd.com/IQ/cto-kpis-every-chief-technology-officer-should-monitor/>
8. What is Email Automation | UiPath, 檢索日期：7 月 23, 2025 ,
<https://www.uipath.com/automation/email-automation>
9. Outlook Email Automation - UiPath, 檢索日期：7 月 23, 2025 ,
<https://www.uipath.com/kb-articles/outlook-email-automation>
10. AI E-Mail Classification - Mailytica, 檢索日期：7 月 23, 2025 ,
<https://mailytica.com/en/ai-e-mail-classification/>
11. SAS Viya on Azure Integrations | SAS, 檢索日期：7 月 23, 2025 ,
https://www.sas.com/en_us/solutions/cloud/microsoft-azure/viya-integration.html
12. UiPath Architecture - Process, Communication, Allocation - Ether Solutions, 檢索日期：7 月 23, 2025 , <https://www.ether-solutions.co.uk/uipath-architecture/>
13. Automation Suite - Deployment architecture - UiPath Documentation Portal, 檢索日期：7 月 23, 2025 , <https://docs.uipath.com/automation-suite/automation-suite/2023.4/installation-guide/deployment-architecture>
14. Integration Patterns and Practices - Salesforce, 檢索日期：7 月 23, 2025 ,
https://resources.docs.salesforce.com/latest/latest/en-us/sfdc/pdf/integration_patterns_and_practices.pdf
15. Key UiPath Features for Smooth Salesforce Integration | MoldStud, 檢索日期：7 月 23, 2025 , <https://moldstud.com/articles/p-essential-uipath-features-for->

- [seamless-salesforce-integration-what-you-need-to-know](#)
16. Best Practices for Component Reusability in SaaS | Midday | Dedicated Fullstack Design and Development as a Subscription, 檢索日期：7 月 23, 2025 ,
<https://www.midday.io/blog/best-practices-for-component-reusability-in-saas>
 17. Best Practices for Creating Reusable Components - PTC Support Portal, 檢索日期：7 月 23, 2025 ,
[https://support.ptc.com/help/thingworx/platform/r9/en/ThingWorx/Help/Best Practices for Developing Applications/reusable component general guidelines.html](https://support.ptc.com/help/thingworx/platform/r9/en/ThingWorx/Help/Best_Practices_for_Developing_Applications/reusable_component_general_guidelines.html)
 18. Event-Driven Architecture (EDA): A Complete Introduction - Confluent, 檢索日期：7 月 23, 2025 , <https://www.confluent.io/learn/event-driven-architecture/>
 19. SAS Viya Platform and Kubernetes, 檢索日期：7 月 23, 2025 ,
<https://documentation.sas.com/doc/en/calintro/latest/n0frl6yr75mda7n1peedf5nkqh1s.htm>
 20. Exploring Inspiring Case Studies of Successful AI Implementations in the World of Enterprise Software Development - MoldStud, 檢索日期：7 月 23, 2025 ,
<https://moldstud.com/articles/p-exploring-inspiring-case-studies-of-successful-ai-implementations-in-the-world-of-enterprise-software-development>
 21. Create a Data Integration Strategy in 10 Steps - Datamation, 檢索日期：7 月 23, 2025 , <https://www.datamation.com/big-data/how-to-create-a-data-integration-strategy/>
 22. Data Alignment—Crack the Code for Data Consistency - Acceldata, 檢索日期：7 月 23, 2025 , <https://www.acceldata.io/blog/data-alignment-consistent-data-better-decisions>
 23. What Are the UCaaS Best Practices? - Symplicity Communications, 檢索日期：7 月 23, 2025 , <https://symplicitycom.com/ucaas-best-practices/>
 24. Enterprise Unified Communications: What It Is + Benefits | SmartChoice, 檢索日期：7 月 23, 2025 , <https://smartchoiceus.com/blog/enterprise-unified-communications/>
 25. Growth Analyzer - Agency Revolution, 檢索日期：7 月 23, 2025 ,
<https://agencyrevolution.com/growth-analyzer/>
 26. Agency growth strategy: How to actually hit your revenue goals next ..., 檢索日期：7 月 23, 2025 , <https://sakasandcompany.com/hit-your-revenue-goals/>
 27. Top 20 IT KPIs To Evaluate IT Project Performance in Minutes - AssessTEAM, 檢索日期：7 月 23, 2025 , <https://www.assessteam.com/it-kpi-list/>
 28. 8 Contract Management KPIs Legal and Procurement Teams Should Track - Gatekeeper, 檢索日期：7 月 23, 2025 ,
<https://www.gatekeeperhq.com/blog/contract-management-kpis-measuring-what-matters>
 29. A Complete Guide to Contract Management KPIs - DataBrain, 檢索日期：7 月 23, 2025 , <https://www.usedatabrain.com/blog/contract-management-kpis>

30. The Power of Accountability: The Fifth Rule of an Extraordinary CTO ..., 檢索日期：7 月 23, 2025, <https://medium.com/@ruben.alapont/the-power-of-accountability-the-fifth-rule-of-an-extraordinary-cto-210004102c2a>
31. Unlock SAS Viya Workbench Multi-Language Architecture Capabilities - YouTube, 檢索日期：7 月 23, 2025, <https://www.youtube.com/watch?v=wlnkrzMvaRY>
32. Integration Planning Checklist | Process Street, 檢索日期：7 月 23, 2025, <https://www.process.st/templates/integration-planning-checklist/>
33. Solved: SAS ACCESS to ODBC Setup on SAS Viya 4 (Kubernetes) - SAS Communities, 檢索日期：7 月 23, 2025, <https://communities.sas.com/t5/Administration-and-Deployment/SAS-ACCESS-to-ODBC-Setup-on-SAS-Viya-4-Kubernetes/td-p/969013>
34. Risks In Implementing RPA & How To Mitigate Them? - Auxiliobits, 檢索日期：7 月 23, 2025, <https://www.auxiliobits.com/blog/risks-in-implementing-rpa-how-to-mitigate-them/>
35. Top 5 RPA Risks for Enterprises and How to Mitigate Them | by Kanerika Inc | Medium, 檢索日期：7 月 23, 2025, <https://medium.com/@kanerika/top-5-rpa-risks-for-enterprises-and-how-to-mitigate-them-889d78646451>
36. API Gateway Pattern: 5 Design Options and How to Choose - Solo.io, 檢索日期：7 月 23, 2025, <https://www.solo.io/topics/api-gateway/api-gateway-pattern>
37. API gateway pattern - AWS Prescriptive Guidance, 檢索日期：7 月 23, 2025, <https://docs.aws.amazon.com/prescriptive-guidance/latest/modernization-integrating-microservices/api-gateway-pattern.html>
38. Real-Time Data Streaming Architecture: Benefits, Challenges, and ..., 檢索日期：7 月 23, 2025, <https://estuary.dev/blog/real-time-data-streaming-architecture/>
39. Messaging Patterns Overview - Enterprise Integration Patterns, 檢索日期：7 月 23, 2025, <https://www.enterpriseintegrationpatterns.com/patterns/messaging/>
40. Supervised methods of machine learning for email classification: a literature survey, 檢索日期：7 月 23, 2025, https://www.researchgate.net/publication/391778011_Supervised_methods_of_machine_learning_for_email_classification_a_literature_survey
41. Accelerate Real-Time Insights With SAS and SingleStore - YouTube, 檢索日期：7 月 23, 2025, <https://www.youtube.com/watch?v=9M5EPanrLEg>
42. Unified Calendar - All-in-One Scheduling Solution - meetergo, 檢索日期：7 月 23, 2025, <https://meetergo.com/en/features/unified-calendar>
43. OneCal: Sync your calendars, avoid scheduling conflicts, 檢索日期：7 月 23, 2025, <https://www.onecal.io/>
44. Calendar API for integrated scheduling applications - Cronofy, 檢索日期：7 月 23, 2025, <https://www.cronofy.com/developer/calendar-api>
45. Unified Calendar API - OneCal, 檢索日期：7 月 23, 2025, <https://www.onecal.io/unified-calendar-api>
46. How an AI-enabled software product development life cycle will fuel innovation -

- McKinsey, 檢索日期：7 月 23, 2025 ,
<https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/how-an-ai-enabled-software-product-development-life-cycle-will-fuel-innovation>
47. Top 7 AI Code Generation Platforms You Must Know - Zencoder, 檢索日期：7 月 23, 2025 , <https://zencoder.ai/blog/top-7-ai-code-generation-platforms>
 48. GPT code generation: AI-assisted programming future - WeAreBrain, 檢索日期：7 月 23, 2025 , <https://wearebrain.com/blog/gpt-code-generation-ai-assisted-programming/>
 49. AI in Software Testing | What it is & How to use AI in Testing, 檢索日期：7 月 23, 2025 , <https://testsigma.com/blog/is-ai-really-important-in-software-test-automation/>
 50. Automated Integration Testing: Elevate Your Software Quality - Functionize, 檢索日期：7 月 23, 2025 , <https://www.functionize.com/automated-testing/automated-integration-testing-guide>
 51. Data Mapping Automation - Securiti, 檢索日期：7 月 23, 2025 , <https://securiti.ai/products/data-mapping/>
 52. Automated Data Mapping: Key Concepts & Use Cases | Nexla, 檢索日期：7 月 23, 2025 , <https://nexla.com/data-integration-101/automated-data-mapping/>
 53. Securing Integrations: Best Practices with Getint, 檢索日期：7 月 23, 2025 , <https://www.getint.io/blog/secure-integrations-with-getint>
 54. Codex Questions : r/OpenAI - Reddit, 檢索日期：7 月 23, 2025 , https://www.reddit.com/r/OpenAI/comments/1m6x9kz/codex_questions/
 55. Free Test Plan Template | Confluence - Atlassian, 檢索日期：7 月 23, 2025 , <https://www.atlassian.com/software/confluence/resources/guides/how-to/test-plan>
 56. Integration testing Collection Template | Postman, 檢索日期：7 月 23, 2025 , <https://www.postman.com/templates/collections/integration-testing/>
 57. Integration Service - Introduction - UiPath Documentation Portal, 檢索日期：7 月 23, 2025 , <https://docs.uipath.com/integration-service/automation-cloud-public-sector/latest/user-guide/introduction>
 58. UiPath automation platform: drive AI transformation with agentic automation | UiPath, 檢索日期：7 月 23, 2025 , <https://www.uipath.com/>
 59. Integration Service - Building your first connector - UiPath Documentation Portal, 檢索日期：7 月 23, 2025 , <https://docs.uipath.com/integration-service/automation-cloud-public-sector/latest/user-guide/building-your-custom-connector>
 60. Effortless custom connectivity with UiPath Connector Builder | Community blog, 檢索日期：7 月 23, 2025 , <https://www.uipath.com/community-blog/tutorials/custom-connectivity-with-uipath-connector-builder>