

# 企業服務匯流排實作型樣

級別：中級

IBM WebSphere 技術銷售經理 [Victor Grund \(vgrund@us.ibm.com\)](mailto:vgrund@us.ibm.com)

IBM 執行軟體 IT 架構工程師 [Chuck Rexroad \(crexroad@us.ibm.com\)](mailto:crexroad@us.ibm.com)

2007 年 12 月 5 日

本文說明在選擇企業服務匯流排 (ESB) 時應考慮哪些技術準則，以及如何用 IBM 產品來實作 ESB，並考察最常用的 ESB 實作型樣。本文涵蓋 IBM 三大 ESB 產品 (WebSphere Message Broker、WebSphere ESB 及 WebSphere DataPower SOA Appliances) 以及可支援或延伸 ESB 型樣的產品 (WebSphere MQ、WebSphere Service Registry and Repository、WebSphere Transformation Extender、WebSphere Adapters、WebSphere Process Server、WebSphere Business Services Fabric 及 IBM Tivoli Composite Application Management for SOA)。本文並深入說明兩個 ESB 實作個案。

## ESB 在企業架構裡的角色與價值

服務導向架構 (SOA) 以彈性、可延伸、模組化的方式來重覆使用及延伸舊應用程式、建構新應用程式。SOA 最大特點在於，它能將商業程序及基礎 IT 架構的組成元素靈活轉變成安全、標準化的服務，並根據多變的商業需求，加以重覆使用及組裝。利用 SOA 所建構的每一個服務都會有一個定義良好的介面 (該服務接收及傳送的所有訊息是由此介面定義) 及一個介面實作，服務部署好之後，還會有一個連結，將服務連結到指定的服務端點。企業服務匯流排 (ESB) 是一種架構型樣，可將各通訊方之間的服務互動虛擬化並加以管理。ESB 就像 SOA 裡服務提供者與服務要求者之間的服務連線橋樑。這個彈性的連線架構能可靠、安全地整合系統，給系統更少、更精巧、更單純的應用程式介面。

服務不會直接連到另一個服務，而是透過服務連線架構 (即 ESB) 互相連結。此架構的虛擬化及管理功能可實作及延伸 SOA 的核心定義。ESB 型樣能虛擬：

- **位置及身分：**ESB 會鑑別訊息並將訊息從一服務遞送給另一服務。這些服務不會知道通訊對方是誰、在哪裡。例如，服務要求者不必知道一共有多少服務提供者在提供這個服務。您可以在 ESB 上新增服務提供者，將訊息遞送給他們，且不會影響到服務要求者的作業。
- **通訊協定：**ESB 在服務要求者及服務提供者之間傳遞訊息時，不會受限於特定傳輸通訊協定或互動樣式。例如，某服務提供者也許只能接受 JMS 輸入，但他還是可以提供服務來滿足以 SOAP/HTTP 撰寫的服務要求。
- **介面：**服務要求者及服務提供者不需要一致的介面。ESB 可以調解介面差異，將服務要求者的訊息轉換成服務提供者預期的型式。

您可以用各種中介軟體產品 (包括軟體及硬體)、程式設計模型及互動樣式來實作 ESB 型樣。ESB 可以：

- 鑑別應用程式與服務間的訊息並加以遞送。
- 跨越各式傳輸通訊協定及互動樣式，在服務要求者及服務提供者之間傳遞訊息。
- 轉換服務要求者與服務提供者之間的訊息格式。

- 在不同來源間尋找及分送商業事件。
- 提供穩定而安全的通訊。
- 透過外掛元件，建立可延伸的架構。
- 進行智慧型訊息遞送，處理訊息時不受位置限制。
- 透過 meta 資料來管理訊息的描述與定義及訊息格式。
- 整合各類資產，以利企業操作運用。

許多企業都會用 XML 及 Web 服務等標準服務連線來實作 ESB。這些標準儼然已成爲業界對 ESB 的一致定義。但事實上 ESB 並不受限於這些標準。只要加以延伸，ESB 架構型樣也能支援現有 IT 系統，包括大型主機、套裝應用程式、感應器/裝置、檔案，以及未採用常用標準的自訂通訊協定。*ESB 必須能滿足企業的所有連線需求。*

ESB 要能支援各類服務互動，包括：單向訊息與要求/回應；非同步呼叫與同步呼叫；發佈/訂閱模型與複雜事件處理程序（即觀察或處理一系列事件，並根據各事件間的關係，產生出一個事件）。

企業要建構 SOA，ESB 勢不可少。ESB 能協助企業：

- **簡化**：減少介面的數量、規模與複雜度。
- **減少風險與成本**：讓 IT 更有效率地回應多變的商業需求。
- **提高重複使用率**：更充分利用資料及商業邏輯，讓應用程式更輕鬆地提供服務。
- **實現動態、即時、因事件制宜的 SOA**：摒棄僵化、回應力差、批次更新的 IT 系統。

## 使用者角色與任務

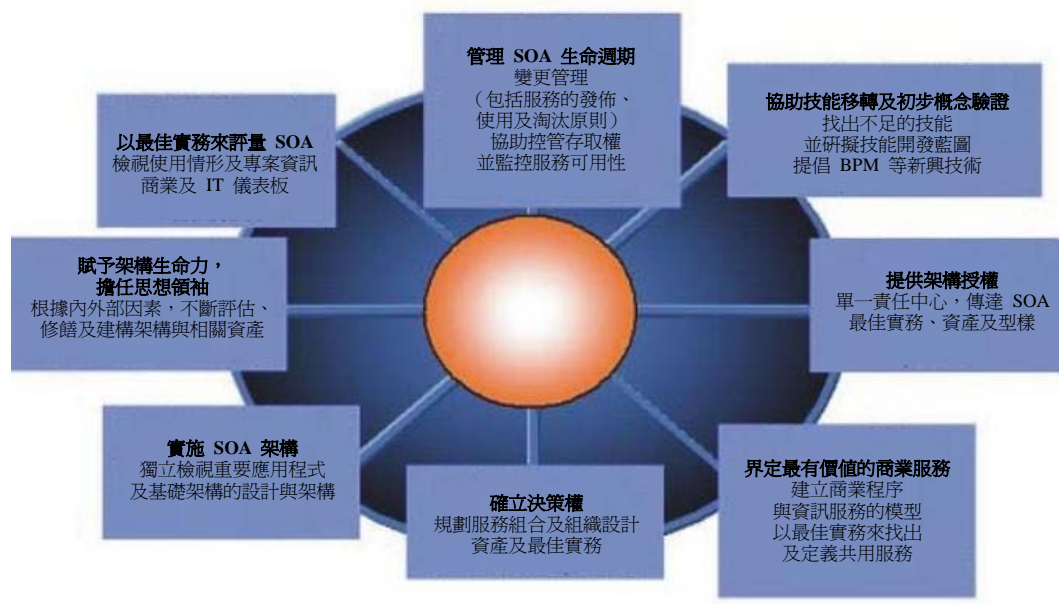
一如所有 IT 專案，SOA 專案成敗亦取決於企業的技術、人力與程序。要成功實作 ESB，企業必須擁有完善的 ESB 技術控管機制。

許多企業會設置 SOA 卓越中心 (COE) 來加速部署 SOA 技術，以便迅速創造價值。COE 必須促進公司所有團隊分享經驗、進行人員培訓、提出程序改進建議，並評估 SOA 完成度。COE 也要協助 SOA 控管中心處理 SOA 相關問題，如版本管理、服務分享、安全議題等。SOA COE 是跨部門 IT 團隊，它必須根據 SOA 願景與策略擬定的策略性 IT 共享解決方案，來引導 IT 投資、設計決策與實作。COE 的職責包括：

- 控管
  - 作爲公司裡 SOA 相關事宜的資訊傳播媒介
  - 協助 SOA 控管中心定義 SOA 控管及管理程序
  - 實作 SOA 控管及管理程序
- 思想領袖及願景研擬：引導 SOA 控管及管理程序的相關思惟
- 貢獻 SOA 專業技能與資源
- 收集及管理知識資產
- 負責 IT 部門的內部溝通與外部聯繫

下圖詳細描繪 COE 的多重角色。

## SOA 卓越中心 (COE)：備受肯定的跨部門 SOA 控管模型



## 技術準則

### 互動模式

完整的 SOA 架構會用到多種互動模式，如要求/回應、發佈/訂閱、事件等等。ESB 必須能以單一基礎架構，支援以下三大企業架構類型，才能充分支援各類互動：

- 服務導向架構 (SOA)。此架構下，應用程式透過可重複使用的服務及明確定義的介面來互相溝通。服務導向式互動採用基礎傳訊及事件通訊模型。
- 訊息導向架構。此架構下，應用程式透過 ESB 將訊息傳送給接收端應用程式。
- 事件導向架構。此架構下，各應用程式獨立產生及接收各自的訊息。

服務使用端呼叫服務的方式有以下幾種：

- **同步**：服務使用端用單一執行緒來呼叫服務；執行緒傳送要求，服務執行時封鎖執行緒，並等待回應。
- **非同步**：服務使用端用一對執行緒來呼叫服務；一執行緒傳送要求，另一執行緒接聽及接收回應。
- **發佈/訂閱**：服務發佈端針對特定主題來發佈訊息。多個服務訂閱端可同時訂閱此主題並接收所發佈的訊息。

ESB 最好可以用幾種不同的呼叫模型來提供同一個服務，以便服務使用端自由選擇偏好的呼叫模型。

選擇 ESB 產品時，要考慮互動模式。架構裡的基本流程參照範例如果是非同步或虛擬非同步，則最適合使用 IBM® WebSphere® Message Broker。

### 維持狀態

有時，訊息在穿越 ESB 時，必須維持原有狀態。例如，如果必須根據前一個相關訊息的環境定義資訊來選擇特定服務端點。或者，如果訊息及事件會隨環境定義而改變（語義環境定義、時間環境定義、時空環境定義），就

要偵測此類複雜情況，並據此延伸 ESB 參照範例。輸入值若來自多重事件來源，則此「維持狀態」功能尤其重要。以安全、金融、銀行及保險等產業最常見的服務層次協定警示及違規檢查為例，來自業務、應用程式或基礎架構的輸入值，其環境定義均各不相同。

ESB 硬體通常無法直接提供「維持狀態」功能，因此必須從軟體著手。WebSphere Message Broker 內建獨一無二的複雜訊息處理流程，最適合需要此功能的企業。

## 端點、標準及通訊協定

端點意指透過 ESB 來互動的服務使用端及服務。

選擇採用標準技術及通訊協定（如 Web 服務）的端點，建構出來的 ESB 最為靈活。但如果企業已經有龐大的舊系統，則一味根據這些標準來選擇端點，不僅不實在，且所費不貲。同理，企業如果能用自家 API 來整合套裝應用程式，那麼將整合大任委託給中介軟體供應商，才是明智之舉。因為適當的配接器可以大大簡化整合工程。但配接器必須有軟體執行時期（不含在 ESB 硬體裡），而且不同的配接器有不同的軟硬體需求，因此在選擇 ESB 產品時，必須先想清楚要不要使用配接器。

## 根據原則來管理互動並動態選取服務

最新的 ESB 技術可以根據代理程式提供的端點統計資料（如同伺服器使用率及性能）來動態選取端點。IBM 有些 ESB 產品已內建此功能，只要搭配 IBM Tivoli Composite Application Manager 即可隨裝隨用。

## 訊息數量、大小及類型

ESB 軟硬體的延伸性各家不同。一般來說，硬體延伸性都不錯，且可充分支援產業標準資料格式，但多元性卻不如軟體。WebSphere Message Broker 效能極佳，ESB 功能齊備，且可透過自訂元件及配接器加以延伸。WebSphere ESB 則可支援各式產業標準格式、Java 及 Web 服務標準，且也可透過自訂元件及配接器加以延伸。

## 可靠性、可用性及服務功能 (RAS)

各家 ESB 產品均有其作業特點，且會透過各種機制及架構（如應用程式伺服器叢集 vs OS 叢集）來提高延伸性及可用性。因此企業在選擇 ESB 產品時，必須考慮上述特點，並根據企業的預算、技能及作業複雜度，設法達成最佳 RAS。

## 調解能力

如果不想自己寫程式碼，則要考慮 ESB 產品的調解能力。WebSphere Message Broker 可充分支援 ACORD、SWIFT 及 COBOL 記錄定義檔等標準資料格式。WebSphere DataPower® SOA 應用程式則可以極高效能支援 XML 及 Web 服務等資料格式。

## IBM ESB 產品

IBM 有多元的企業服務匯流排 (ESB) 產品及選項，企業可根據自身需求，選擇適當技術，來建立 ESB。大型企業由於牽涉的地理區及技術等較多，可能得綜合多種 ESB 技術來建立混合式 ESB。例如，如果兩大部門分別設置了自己的 SOA，之後如果這兩個 SOA 需要交互運作，就會需要混合式 ESB。

IBM 三大 ESB 產品分別為 WebSphere Message Broker、WebSphere ESB 及 WebSphere DataPower SOA Appliances。

## WebSphere Message Broker

WebSphere Message Broker 可協助客戶建立 ESB（或大型解決方案裡的中央 ESB）。WebSphere Message Broker 保有 WebSphere MQ 傳訊空間，最適合會大量使用 MQ 傳訊的環境。它會定義 WSDL 等訊息格式的介面、可調解訊息流程，且能支援 WMX 及 HTTP 等各種通訊格式。WebSphere Message Broker

也可根據內容來進行發佈/訂閱，並提供可管理的主题空間。

WebSphere Message Broker 支援多種服務互動端點的調解型樣，且除了各產業標準訊息集（有些用 XML 編碼，但不是全部都用）外，還可支援其他使用者定義的訊息格式。WebSphere Message Broker 最能滿足客戶以下各類需求：

- 交易式
- 發佈/訂閱
- ACORD、SWIFT 或 COBOL 記錄定義檔標準格式
- XML 格式化資料
- 符合 WS-\* 標準
- WebSphere MQ 傳訊
- 複雜轉換
- 複雜事件處理

WebSphere Message Broker 有多種 ESB 連線選項，且可以在任何資料格式之間轉換，因此能將企業的舊應用程式及不符合產業標準的應用程式，順利連接到 ESB。

## WebSphere ESB

WebSphere ESB 是 ESB 的 J2EE 建立實例。WebSphere ESB 在 IBM WebSphere Application Server 裡的 J2EE 儲存器內執行，最適合採用 J2EE 及 WS-\* 標準的應用程式，尤其是那些只需要跟其他應用程式伺服器通訊的應用程式。本產品也是企業入門 SOA 的絕佳起點，讓企業以最基本的 Web 服務為踏板，逐步建構出完整的 SOA 架構。

WebSphere ESB 提供標準 Web 服務連線功能、JMS 傳訊以及服務導向整合。供點對點傳訊及發佈/訂閱傳訊之用的 JMS 應用程式以及 JAX-RPC 服務導向應用程式可直接連到 WebSphere ESB；或者，訊息也可以透過 WMQ、SOAP/HTTP、SOAP/JMS 等各種傳輸，傳送給 WebSphere ESB。WebSphere ESB 利用 Web 服務開道來調解 SOAP/HTTP 型及 SOAP/JMS 型應用程式。本產品並提供 Universal Description Discovery and Integration (UDDI)。WebSphere ESB 最能滿足客戶以下各類需求：

- J2EE 環境
- Web 服務介面
- SOAP/HTTP

## WebSphere DataPower SOA 應用裝置

WebSphere DataPower SOA 應用裝置兼含硬體與軟體，具備 XML 加速、安全強化、ESB 等重要功能。

DataPower 主要特色如下：

- 將硬體、韌體與內建作業系統調整成最佳狀態
- 保證確實鎖定配置
- 減少安全漏洞
- 將加密金鑰及已上鎖的審核日誌儲存在硬體儲存體
- 無硬碟、光碟或 USB 埠

- 防竄改機箱，一打開機器就不能用
- 名符其實的 SOA 應用裝置，可簡化作業。以最安全的防護措施，帶領企業迅速進入 SOA 殿堂

DataPower 最能滿足客戶以下各類需求：

- 安全開道
- XML 防火牆、剖析及驗證
- 基本遞送
- 內容型遞送
- 橋接通訊協定（HTTP、WebSphere MQ 用戶端、FTP、ODBC 等）

## 輔助技術

除上述 ESB 技術外，企業還可運用以下輔助技術來加速或延伸 ESB 實作：

- [WebSphere MQ](#) 可在各類平台之間傳遞可靠的訊息。許多企業均以 WebSphere MQ 作為傳訊主幹。
- [WebSphere Service Registry and Repository](#) 透過整合式服務 meta 資料儲存庫來控管服務、管理服務生命週期。本產品能深入透視 SOA 裡的服務、提高服務一致性，並減少冗餘服務。
- [WebSphere Transformation Extender](#) 提供通用的轉換功能，可應付極端複雜或迅速變遷的需求（如 EDI）。
- [WebSphere Adapters](#) 是附加程式，可附加於套裝服務導向應用程式或舊系統，將之整合進 SOA。其內的配接器可將舊資訊系統及技術融入 ESB。
- [WebSphere Process Server](#) 具備程序工作流程功能，且內建 WebSphere ESB，能編排複雜的 ESB 互動，來協調不同系統間的服務呼叫與回應。
- [IBM Tivoli Composite Application Management for SOA](#) 係 SOA 環境專用的 Tivoli 監控產品。IBM Tivoli Composite Application Manager for SOA 利用現有系統管理工具來繪製及呈現 SOA 環境的作業全視圖。
- [WebSphere Business Services Fabric](#) 乃端對端 SOA 平台，可建構複合式商業服務的模型，並加以組裝、部署、管理及控管。企業可以利用 WebSphere Business Services Fabric 來建置商業服務，以便日後將這些服務組裝成較大型、全公司通用的商業程序或解決方案，然後根據服務要求的環境定義，動態修改及提供這些商業程序或解決方案。

## WebSphere Service Registry and Repository

WebSphere Service Registry and Repository 除基本 UDDI 服務目錄外，還具備許多其他功能。由於 UDDI 規格尚不明確，因此不同供應商常會用不同方式來實作 UDDI 規格。WebSphere Service Registry and Repository 不僅能記錄已結束服務，還會將服務提供給 ESB，供開發人員搜尋及使用已結束服務。

WebSphere Service Registry and Repository 內含：

- 服務登錄，內有服務資訊（如服務介面、作業及參數）
- meta 資料儲存庫，其架構完善、延伸性佳，可記錄各類服務使用情形

無論公司的 SOA 架構處於哪一階段，均可搭配 WebSphere Service Registry and Repository：

- 在服務開發階段，可與標準資產管理解決方案互動
- 在服務變更及版本管理程序階段，可追蹤服務
- 在執行階段，可高效率存取執行時期設施
- 在管理階段，可與作業系統管理公用程式分享服務資訊及服務 meta 資料

WebSphere Service Registry and Repository 可指出誰在何時用哪個服務做了什麼事。詳情請見以下個案研討。

## 基本 ESB 型樣

本節介紹核心 ESB 調解功能常用的一種語言。這些型樣可清楚呈現 ESB 在環境定義裡的調解方式。您可以根據最佳實務及經驗教訓，用這些型樣來呈現可重覆使用的調解型樣。您也可以組裝這些核心型樣，來呈現 ESB 使用案例。

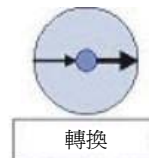
### 通訊協定切換

- 供服務要求者透過各種互動通訊協定或 API（如 SOAP/HTTP、JMS 及 MQ Integrator (MQI)）來分派訊息。
- 將要求轉換成服務提供者預期的格式。
- 可用於要求者端及提供者端或二者之間。
- 各格式的定義及格式間的關係若很清楚，通常會自動建立這種調解型樣。



### 修飾元

- 不變更環境定義資訊，只更新訊息內容。
- 轉換次樣式
  - 將訊息內容轉換成另一種格式（綱目），讓服務要求者及服務提供者的訊息定義相吻合
  - 包括「封套」及「去封套」，意即，為了能將訊息傳輸到其他網路，而將訊息放到格式封套裡的網路格式，或據此移除封套
  - 包括加密



- 補充次樣式
  - 將來自外部資料來源的資訊加進來，如調解或資料庫查詢的自訂參數，以更新訊息內容



補充

## 路由器

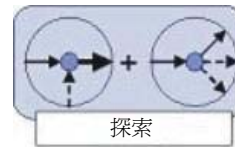
- 變更訊息的預期遞送路徑，選擇某一個與該調解相關的服務提供者。
- 範例：將金卡客戶轉遞給 SLA 更高的服務提供者。



遞送

## 探索

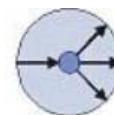
- 查詢 ESB 登錄，以探索有哪些服務提供者符合服務要求者的要求，選定服務提供者，並將訊息遞送過去。
- 遞送型樣加強功能
  - 調解時沒有預先配置的服務提供者群
  - 提供者會根據要求者的訊息格式、需要的 QOS 或可用通訊協定，來比對符合的提供者
- 範例：資料中心失效接手。資料中心連上線、登錄服務、遞送資料流量給資料中心，不必在每次調解時更新配置。



探索

## 分送

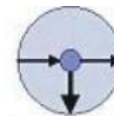
- 將訊息分送給一組相關方，通常是根據訂閱者喜好設定檔。



分送

## 監控

- 觀察通過 ESB 的訊息，但不變更訊息。範例：
  - 監控服務層次
  - 提供資料以判定問題
  - 付費計量
  - 記錄商業問題（交易超出特定收益）
  - 記載訊息以供審核

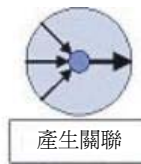


監控



### 產生關聯（複雜事件處理程序）

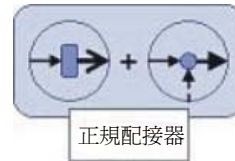
- 從訊息或事件串流中衍生出複雜事件，包括型樣識別規則及探索型樣後的回應規則（例如，根據觸發事件串流的內容，產生一個複雜事件）。



這些基本型樣可加以彙總，創造出複雜型樣。

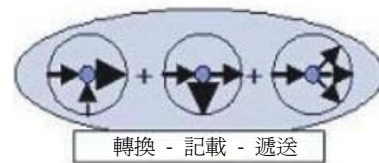
### 正規配接器

- 將所有相關方共用的訊息及商業物件集標準化，形成一個正規格式。正規配接器型樣會將端點的原生匯流排連接通訊協定轉換成標準通訊協定，將內容標準化，再於傳遞後轉換回來。它是「通訊協定切換」與「轉換型樣」的彙總。



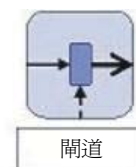
### 轉換、記載、遞送

- 常用的 ESB 彙總型樣。轉換送入訊息（「轉換」）、記載訊息（「監控」）並「遞送」到端點。



### Proxy 或閘道型樣

- 遞送或通訊協定切換型樣的變式，可提供安全功能（授權及存取控制）及記載或審核功能。
- 可結合轉換及監控調解，提供加密及記載或審核功能，也可在一對多關係中彙總及分離訊息。



範例：服務入口網站。服務入口網站是多個服務的單一聯絡點，可隱藏「內部」服務的詳細資料。

## 複雜 ESB 型樣

這些型樣用比較接近真實世界的方式來表示如何用 ESB，將豐富的服務虛擬化。大型企業（或是合作夥伴採用 SOA 的企業）常常需要不只一個 ESB，以達成：

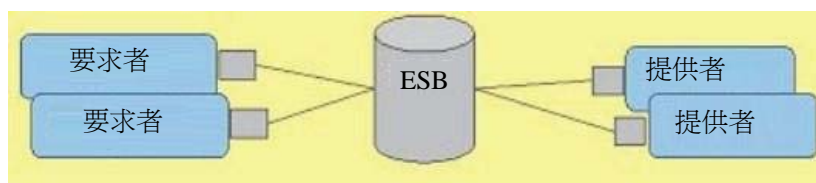
- 安全
- 效能
- 責任性/審核性
- 資源控制

以下型樣不盡完整，僅供參考。您可視特定專案需求，將各型樣加以結合。

## 廣域 ESB

廣域 ESB 將所有服務要求者導向同一個登錄，因此每一個服務提供者可以看到每一個服務。此型樣較適合小型系統測試環境。正式作業環境中，如果只有一個廣域 ESB，沒有其他 ESB（即非混合式 ESB，本節稍後會深入討論），則此型樣可能不盡理想。

- 所有服務均使用同一個名稱空間。每一個服務提供者可以看到這整個異質、由中央管理、分散在各地理區的環境中，每一個服務要求者的要求。
- 適用於個別部門或小型企業，因為整個組織可能都會用到每一個服務。



IBM 為廣域 ESB 提供兩大 ESB 技術選擇：WebSphere Message Broker 及 WebSphere ESB。J2EE 及標準環境最好選擇 WebSphere ESB。如果會跟非 J2EE 系統進行複雜的轉換與互動，則最好使用 Systems WebSphere Message Broker。大型企業可以雙管齊下，以便根據訊息類型及目的地，適當遞送訊息（後面的「聯合 ESB」會詳細說明）。

## ESB 閘道

ESB 閘道型樣可控制內部或外部網域邊界之間的服务互動，並保障其安全。

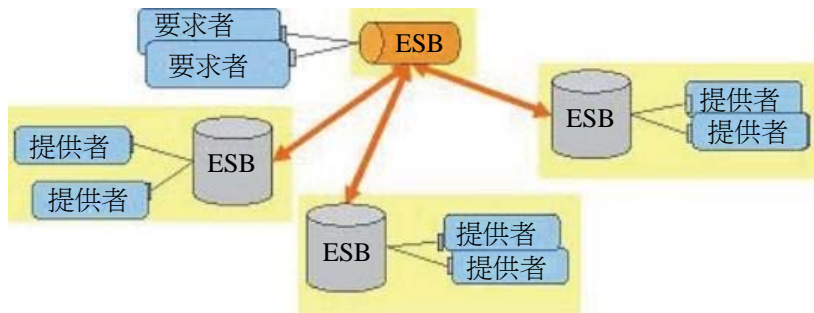


ESB 閘道型樣最常見的實例是 DataPower 應用裝置。該應用裝置除了該有的閘道功能外，還提供 XML 防火牆。DataPower 非常適用於網路邊界，因為它含有內建式作業系統及軟體實作，安裝作業系統時，風險較低。

## 聯合 ESB

聯合 ESB 使用多個服務登錄及管理網域，可將個別登錄對映到聯合服務集。聯合 ESB 可加快服務與多個 ESB 之間的互動。企業常會利用聯合 ESB，將全企業都會用到的服務集成服務子集。

- 一個主要 ESB，由多個獨立 ESB 聯合而成。
- 服務使用端及服務提供者只要連接到主要 ESB 或其中一個獨立 ESB，即可存取整個網路的服務。
- 主管部門下如果有多個自治部門，則可將這些自治部門聯合起來。

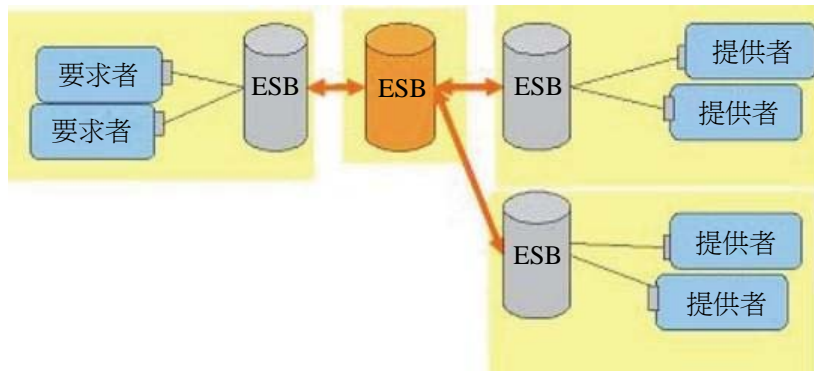


聯合 ESB 的核心產品為 WebSphere Message Broker，次核心產品為 WebSphere ESB，小型遠端辦公室則可考慮 DataPower 應用裝置。本型樣中，各站台有各自的服務登錄。最理想的聯合 ESB 型樣應該是在網路邊界先用 DataPower 應用裝置來加強授權及身分鑑別，並在 XML 進入內部系統前先行剖析與驗證，以此加速交易，並降低 ESB 伺服器裡 CPU 的使用率。

## 分配 ESB

整個組織如果都會用到某一服務子集，但服務並無多個實作，則可用分配 ESB 來支援多個登錄。

- 屬橋接服務，可選擇性地將服務要求者或服務提供者公開給其他 ESB 網域中的合作夥伴。
- 各 ESB 管理各自的名稱空間。
- 由實作橋接服務的共用分配管理系統來實現 ESB 間的服務互動。
- 一部門如果會開發及管理自己的服務，但僅共用少數服務，或只使用部分提供給全公司的服務，即可使用分配 ESB。



分配 ESB 型樣與聯合 ESB 極為雷同，但前者是強制執行點，也是服務路由器。

最常見的分配 ESB 是 JBI 標準 (JSR 208)。IBM 現已揚棄 JBI 規格，因為該規格已無法進一步有效滿足客戶需求。許多技術及開放式規格都能實現更好的互通能力、提供更棒的運作機置。因此 IBM 目前主要著眼於如何將最多的產品、應用程式及現有商業資產充分整合。

## ESB 情境一：ABC 旅館業者

ABC 旅館業者為跨國休閒旅遊公司，旗下有多個高級豪華旅館品牌。該公司擁有、管理或經銷九百多間旅館，房間數總計達 230,000 間。

ABC 展開多年期的大型 IT 專案，想要實作分散式 J2EE 服務，來取代龐大的大型主機型中央訂房系統，希望在淘汰大型主機後，能壓低成本，並讓公司營運更靈活。本專案想採用 SOA 概念，將服務部署到位於兩個不同地點的資料中心，並為個別服務部署多個實例，以提高系統產量及可用性。

這個專案本來並未考慮 ESB。但隨著專案前進，服務整合愈來愈複雜，架構工程師及開發人員開始感到力不從

心，服務運作瓶頸頻出，整個專案幾乎要功虧一簣。因此 ABC 終於決定要架設 ESB。該公司於深入評估後，選出了最適當的應對措施。

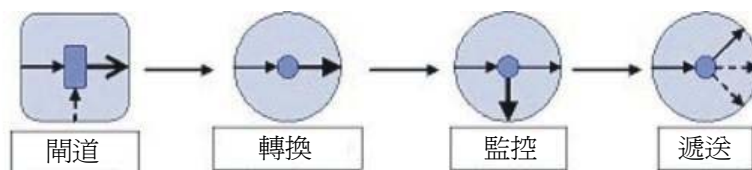
ABC 希望其 ESB 能夠：

- **遞送要求及管理工作量：**ABC 的基礎架構裡，一個服務均部署了多個副本，以確保服務的可靠性、服務性及可用性。ESB 要能將客人的服務要求遞送給適當的服務版本及實例。此外 ESB 還要用一些規定來管理工作量，將要求分送給相同的服務實例。ESB 可以用簡單的工作量分散演算法來管理工作量，但如果能根據端點更主動地管理工作量，當然更好。
- **整合登錄：**ABC 的每一個服務建置時期型錄都有一個 UDDI 服務登錄。沒有更主動的執行時期服務登錄沒關係，但一定要能支援這種登錄。
- **調解：**ABC 希望 ESB 能調解用戶端與服務。服務資料以 WSDL / XML 表示，因此不需要處理太隱密的資料格式，但要加強訊息驗證 (WS-I and XSD) 並加速效能/傳輸。
- **管理交易：**ABC 有時必須償還及回復已確定的服務要求 (即複合服務裡的個別交易)。ABC 希望能根據規則及程序來進行這些小型交易。
- **服務編排及工作流程：**小型服務互動子集一定要能支援複雜的工作流程，BPEL 最為理想。商業狀態機器、互動的排程與計時、產生事件關聯、可與人互動之工作流程等等也不錯。
- **加強與後端系統的整合：**最好能直接整合 Siebel 及其他後端系統。
- **高效能/低延遲：**要儘量減低對整體效能及延遲的負面衝擊。
- **監控、管理、維護：**監控服務層次、收集標準/自訂指標、偵測錯誤、分析根因、產生報告、分析歷程資料等，尤其要能自動向操作員發出警示及通知。
- **安全：**要整合 Tivoli Access Manager、LDAP 及其他協力廠商的安全產品。要考慮認證與金鑰管理功能、標準身份鑑別、存取控制、加密/解密、SSL 加速、沙賓法案 (SOX) 控制、DMZ 部署可行性。
- **SDLC：**ABC 用的是 Eclipse-based Rational Application Developer 及 Rational Software Architect。因此 ESB 部署工具最好能跟現有環境互運。
- **快速部署：**ABC 在決定採用 ESB 之前已投注太多心血，因此太昂貴、太耗時的部署將不予考慮。
- **簡化作業：**ABC 之所以決定採用 ESB，就是為了降低風險及作業複雜度，因此將優先考慮組裝完備、操作簡單的解決方案。

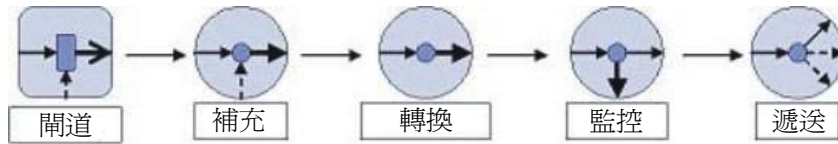
## 使用案例樣本

### 事業夥伴呼叫後端服務

ABC 常要跟事業夥伴交流空房情況及訂房要求。事業夥伴必須能透過 DMZ 部署型樣來虛擬呼叫後端服務。EBS 必須驗證用戶端身分、給予授權，並監控及記載所有活動。

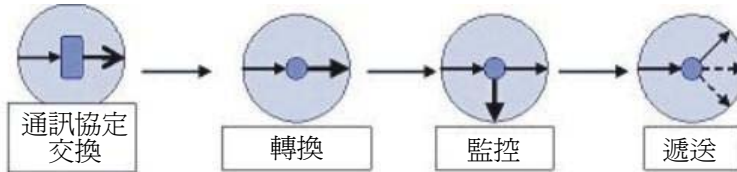


未來可能還得根據用戶端、要求類型或其他資訊將資料流量分段。

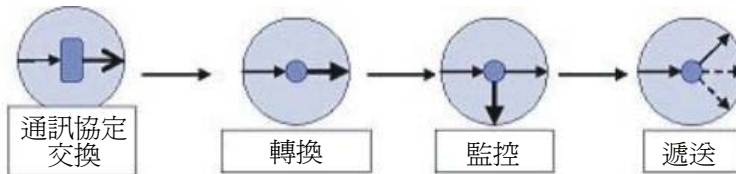


### 個別旅館呼叫後端服務

各旅館的訂房系統必須與中央訂房系統相連接。例如，某旅館剛剛預約出最後一間空房，必須向中央訂房系統發出「無空房」通知。假定中央訂房系統的服務是用 SOAP/HTTP（或其他通訊協定）部署的，個別旅館的訂房系統也要能透過 MQ，用 XML 來通知中央訂房系統。



此型樣可加以擴大，變成可根據端點使用率統計資料來遞送。



上述兩個使用案例可處理掉絕大多數的 EBS 資料流量。

### 錯誤及複雜的回復

假設在履行某旅館的某個用戶端要求時，複合服務裡有某個互動出了錯，而且似乎一定要透過某個包含人為操作的工作流程，才能回復這個錯誤。此時可以在該履行架構裡新增一些元件來實現此功能，但核心 ESB 型樣不會改變。

### 複雜工作流程及後端整合

假設要將一「建立客戶」要求輸入偏好的客人追蹤系統，且業務員想將輸入 Siebel 的部分資訊當成調解的一部分，ESB 可針對現有服務來履行標準調解，並包含一 ISV 後端系統更新。此時可以用轉換-監控-遞送 ESB 彙總型樣，而履行架構可以用多個 IT 元件來履行服務要求。

## 解決方案架構概觀

IBM 建議的混合式 ESB 架構包含軟體及硬體，各元件提供不同的 ESB 功能。

### 硬體應用裝置

- 在效能與延遲之間達成最佳平衡
- 配置、部署及操作極為簡單
- 安全、防竄改
- 不用一般平台處理昂貴的程序

### 軟體

- 支援應用程式主機作業、工作流程及複式調解
- 用配接器整合應用程式

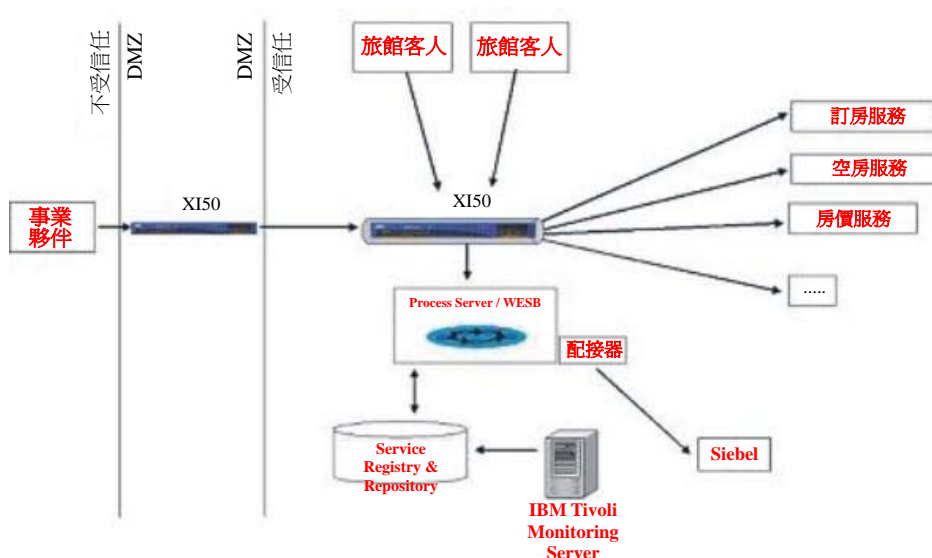
- 可監控商業活動

IBM 相信很多客戶都會部署混合式架構，以兼享二者長處。混合式架構多了 SOA 服務登錄及管理功能。

## 解決方案元件

- DataPower XI50 (硬體 ESB)
- WebSphere Process Server (內含 WebSphere ESB)
- WebSphere Business Integration Adapter (整合後端系統)
- IBM Tivoli Composite Application Management for SOA (監控及管理服務)
- WebSphere Service Registry and Repository (登錄服務)

以下系統環境定義圖說明各元件間的互動。



DataPower XI50 在此架構中擔任兩個要角：DMZ 裡的服務閘道及受信任區裡的 ESB 硬體。涉及 ESB 軟體的任務（如使用 Siebel 配接器或需要人為回復的複雜錯誤）則交由 WebSphere ESB 隨附的核心元件 WebSphere Process Server 來處理。

DataPower 可履行 ABC 絕大多數的 ESB 資料流量要求，達成高效能，而在 WebSphere Process Server 及 WebSphere ESB 的輔助下，DataPower 更能在架構裡靈活運作。

IBM Tivoli Composite Application Management 產品能監控並管理 ESB 及與之互動的元件。監控代理程式會收集資訊，依此實現較複雜的 ESB 補充及遞送型樣（像是動態選取服務），也就是根據監控代理程式收集到的資訊（如主機作業伺服器工作量）來選擇目標服務。

WebSphere Service Registry and Repository 可管控服務的整個生命週期，並輔助 ESB 動態選取服務。

如前所述，光靠 UDDI 絕對無法完成所有工作。

本架構的宗旨，是要在效能及彈性之間達成最佳平衡。

## ESB 情境二：XYZ 保險公司

本例結合了現實世界裡的多個保險公司專案案例。醫療保險公司 XYZ 擁有數百名保戶，常需要跟保戶、供應商、保險代理商等互動。



XYZ 想引進 SOA，將公司的資訊技術架構現代化。該公司投注了大筆心血來開發用 COBOL、Assembler 及 PL/I 撰寫的大型主機 (z/OS) 應用程式。這些系統採用 DB2 及 IMS 資料庫，並透過 CICS (客戶資訊控制系統) 與一般使用者互動，其內部及外部應用程式的傳訊主幹係以 IBM 的 WebSphere MQ 為主。該公司有一個主資料中心，連接到多個衛星資料中心及協力廠商系統。

XYZ 打算開發許多新應用程式並進行一系列變更。但在此我們僅探討 XYZ 想用服務導向架構 (SOA) 建置的「呼叫中心環境」(CCE) 及「理賠處理應用程式」(CPA)。

根據 IBM 及 XYZ 的定義，本專案要建構的架構如下：

- 要充分利用舊應用程式，但不要改寫程式碼
- 要充分利用大型主機系統，且要在應用程式適用的平台上建立新服務
- 要支援多種通訊協定 (SOAP/HTTM、WebSphere MQ 等)
- 要能安全地跟外部組織的 SOA 互動
- 要在單一環境裡跟多個分散各處的資料中心互動
- 要能根據希望的服務層次，即時監控及遞送交易

XYZ 想要的 ESB 跟 ABC 很類似，但需要的調解更為複雜。此外，由於涉及的法令繁多，XYZ 需要更安全、更容易審核的架構。

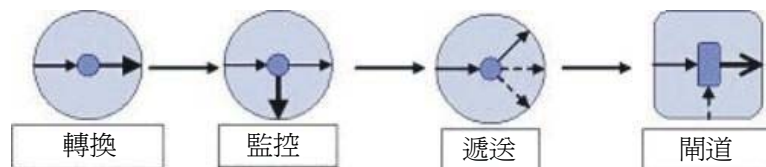
## 使用案例樣本

### 遠端資料中心呼叫主資料中心服務

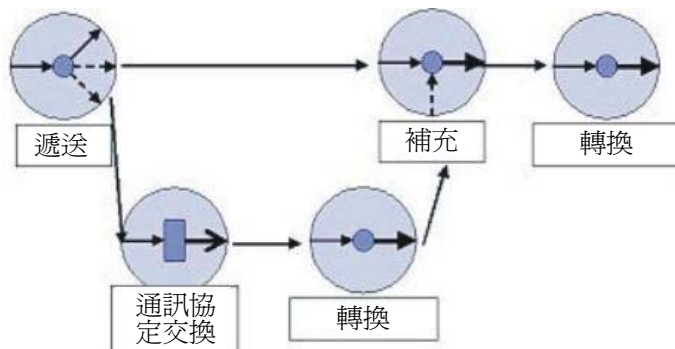
遠端資料中心常需要呼叫主資料中心來支援「呼叫中心環境」(CCE)。雖然兩個資料中心都隸屬公司內部，但為安全 (及方便審核) 起見，這些呼叫還是要透過閘道。

因此當單一服務呼叫主資料中心時，舊應用程式裡可能會引發多個服務呼叫。此時就要編排服務及工作流程元件，以便將收到的資訊組合成單一服務回應。

本服務互動中，遠端資料中心的動作如下：



主資料中心的動作如下。請注意，下圖並未清楚畫出閘道及監控元件，但這兩者都是本解決方案的主要元件：

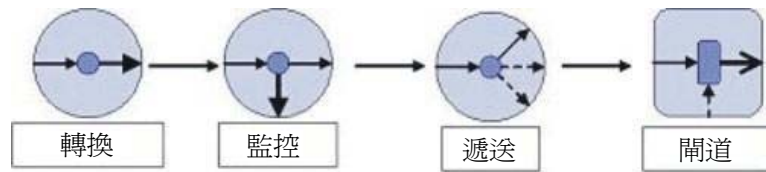


### 主資料中心呼叫遠端資料中心服務

主資料中心常需要呼叫遠端資料中心，以驗證前一個「呼叫中心環境」(CCE) 活動。因此遠端資料中心必須有

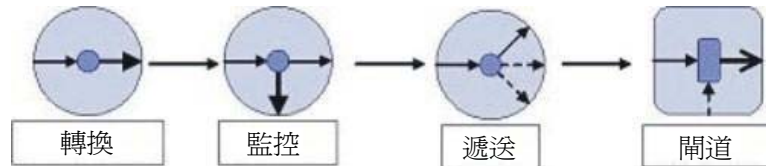
完整的 SOA 環境，且必須抄寫主資料中心的安全及審核功能。

本服務互動中，主資料中心的動作如下：



#### 主資料中心呼叫協力廠商服務以驗證地址

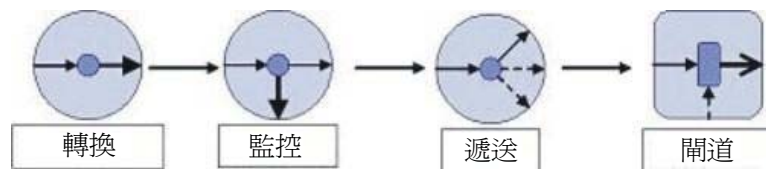
主資料中心將地址存進 XYZ 的資料庫之前，會先呼叫協力廠商服務，來確認地址是否正確。



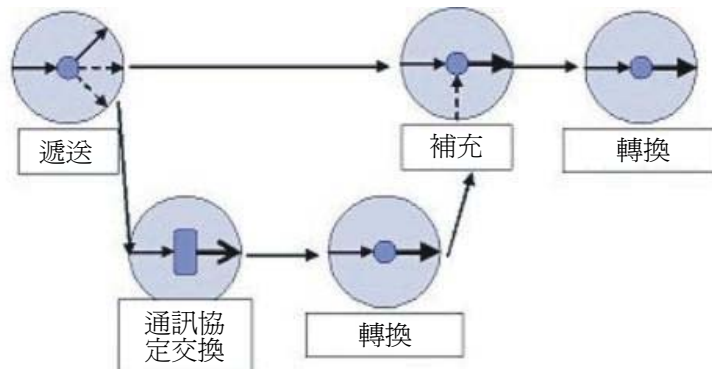
#### 外部代理商呼叫舊應用程式服務

由於外部協力廠商應用程式並不知道其所呼叫的服務是來自舊應用程式，因此這個使用案例的架構很特別。

本服務互動中，遠端資料中心的動作如下：



主資料中心的動作如下。請注意，下圖並未清楚畫出開道及監控元件，但這兩者都是本解決方案的主要元件：



#### 根據服務呼叫型樣來偵測詐騙

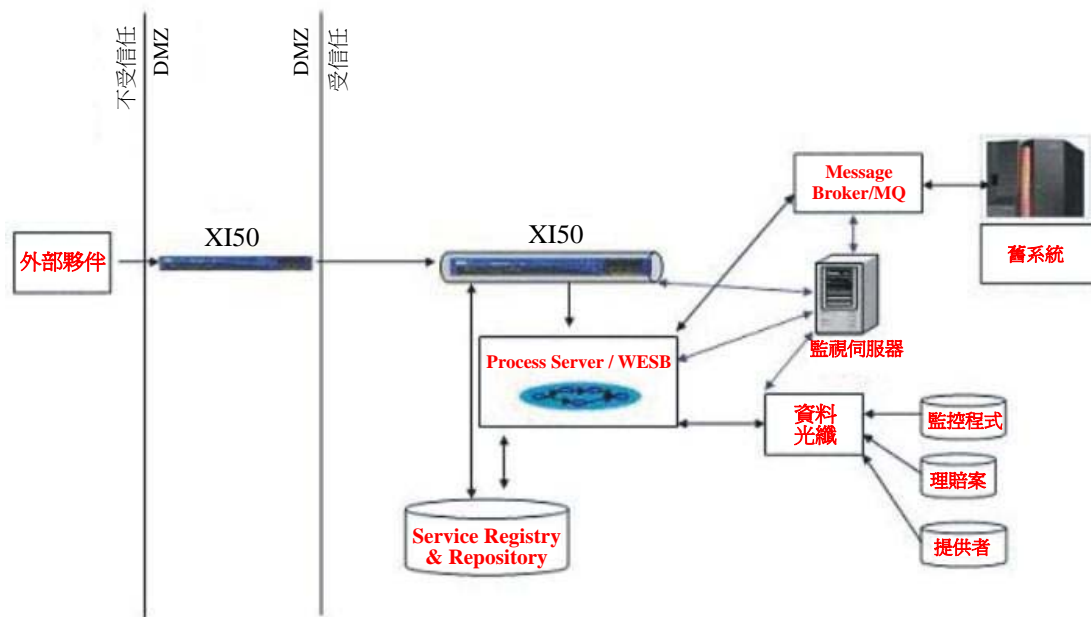
本使用案例較為複雜，不僅要監控呼叫次數，還得監控是誰在呼叫？在何時呼叫？當某位客戶或供應商呼叫次數過於頻繁時，ESB 會暫時中止該使用者的存取權，並通知 XYZ 調查此異常行為。此系統必須整合監控、記載、產生關聯及安全等功能，難度頗高，但現有 ESB 及 SOA 技術已能實現這一點。

#### 解決方案架構概觀

IBM 與 XYZ 合作建置一混合式 ESB 架構，囊括軟體與硬體。IBM 的三大 ESB 技術全都派上用場，XYZ 根據其對舊系統整合、服務編排及基本工作流程的需求，決定了架構要採用的技術。

以下系統環境定義圖說明各元件間的互動。





一如 ABC 專案，本例中，DataPower XI50 仍扮演兩大要角：DMZ 裡的防火牆及受信任區裡的 ESB 硬體。WebSphere Process Server/ESB 負責編排複雜的服務互動，擷取資料光纖裡的資料並將之匯總成較複雜的商業服務回應。WebSphere Message Broker 負責處理複雜的訊息轉換，以便與舊系統互動。三大 IBM ESB 解決方案三管齊下，各展所長，相輔相成，成效斐然。

IBM 深知，許多企業都跟 XYZ 一樣，需要部署複雜的 ESB 架構，需要多項技術多管齊下，才能實現必要功能，並在可用性、效能及安全之間達成最佳平衡。

## 解決方案元件

XYZ 採用的解決方案包括：

- DataPower XI50 (硬體 ESB、XML 安全閘道)
- WebSphere Process Server (內含 WebSphere ESB，且可編排複雜服務互動的程序)
- WebSphere Message Broker (軟體 ESB，支援複雜訊息轉換及非標準傳訊)

## 結論

本文說明 ESB 的概念及其在真實世界中的使用範例。本文並探討在建置 ESB 時有哪些技術可供選擇、哪些情況下應使用哪類產品？如需深入技術領域，請與有經驗的專家共閱本文，以便根據您的特殊情況，設計最適當的解決方案。

## 致謝

感謝 Marc-Thomas Schmidt、Kyle Brown 及 Rachel Reinitz 為 IBM SOA 專案打下根基，並感謝他們在 ESB 領域的貢獻。

## 作者簡介

**Victor Grund** 是 IBM 的 WebSphere 技術銷售經理，負責新英格蘭及紐約州北部的 WebSphere 產品。Grund 獻身企業 IT 領域達 17 年，最近七年待在 IBM，擔任過經理、軟體架構工程師、SOA 架構主工程師、IT 專家及顧問。Grund 在設計及實作企業應用程式與整合解決方案方面擁有豐富的經驗。電子郵件：

[vgrund@us.ibm.com](mailto:vgrund@us.ibm.com)。

**Chuck Rexroad** 是軟體 IT 主架構工程師，負責 IBM 在新英格蘭及紐約州北部地區的業務。Rexroad 於 1998 年加入 IBM，擔任 Tivoli 品牌技術專家，他在系統管理各方面均擁有豐富的經驗。Rexroad 投身 SOA 專案與技術領域多年，曾協助中大型企業客戶實作 SOA。電子郵件：[crexroad@us.ibm.com](mailto:crexroad@us.ibm.com)。