» 何謂雲端？

雲端的定義

雲端是**雲端運算**的簡稱，指**透過網路利用**資訊系統與伺服器、網路等 **IT 資產的型態**。如圖 1-1 所示，雲端服務的組成包含提供雲端服務的業者、利用該服務的企業組織或者個人。

雲端為**雲（cloud）**的意思，這也是使用雲朵符號簡單表達網際網路的由來。

因雲端改變的企業 IT 與個人生活

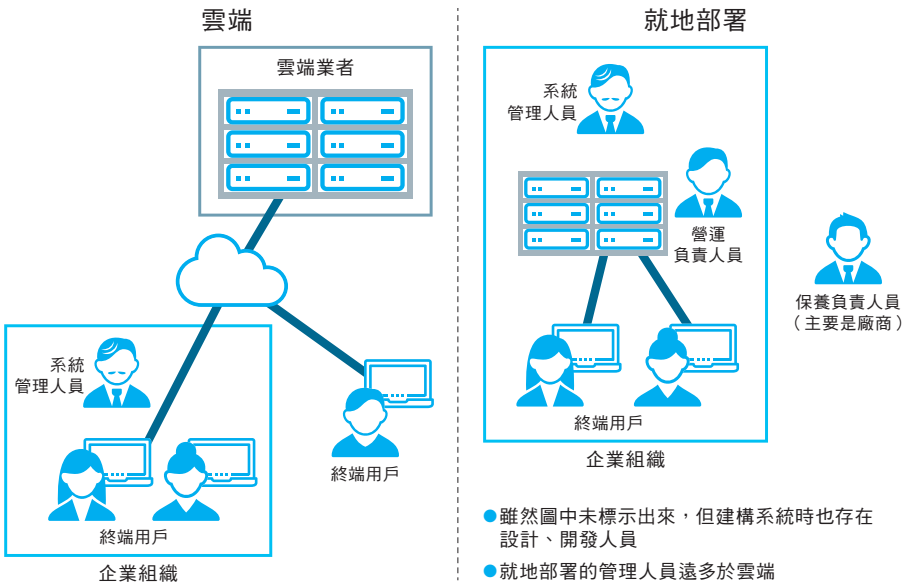
在企業組織中，資訊系統原本用就地部署（On-premises）的方式，將伺服器、網路設備設置於總公司或者資訊系統中心等能夠自行管理的場所（圖 1-1）。除了需要相應的設置空間外，還得管理系統正常運行與發揮功能，以及維護 IT 資產，數量愈多管理起來就愈辛苦。

若使用雲端服務，則只需要支付企業組織、個人用戶的使用費，就可在網際網路上使用這些 IT 資產。當硬體設置場所、管理主體改變後，就能夠將心力專注於使用方法上，管理也會變得相當容易。

如圖 1-2 所示，雲端服務也改變了日常生活，我們可透過隨選視訊（Video On Demand）觀賞以前需將 DVD 置入電腦播放的影片；或經由智慧手機確認行車記錄器的影像。

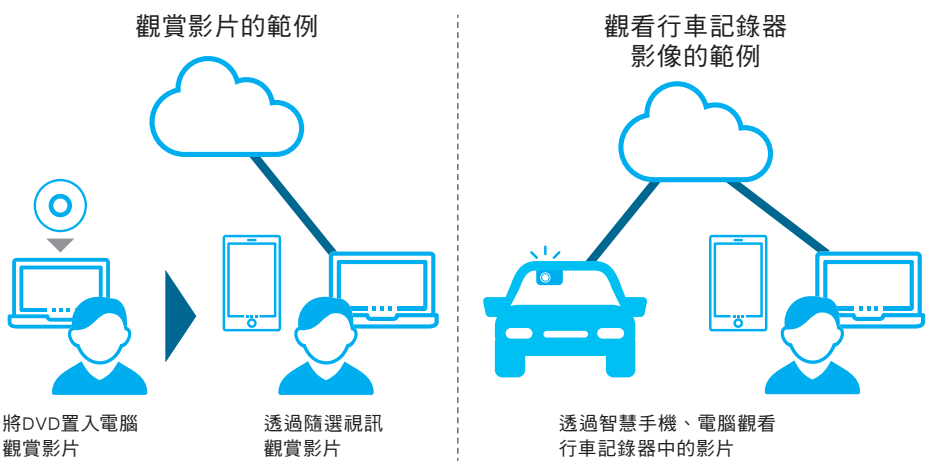
雲端運算大幅度改變了企業組織的 IT 資產、運用型態，對個人生活也帶來深遠的影響。雲端運算也符合共享各種事物的時代潮流。

圖1-1 雲端服務的組成



- 雖然圖中未標示出來，但建構系統時也存在設計、開發人員
- 就地部署的管理人員遠多於雲端

圖1-2 雲端服務帶來的新系統



Point

- 雲端是透過網際網路使用資訊系統、IT 資產的型態。
- 雲端也深深影響企業組織的 IT 資產運用型態、個人生活。

» 因雲端改變的系統建構

設計與開發不能缺少的傳統型系統

前面談了雲端系統的特徵與普及的背景，但系統的建構也發生巨大的改變。在尚未利用雲端的舊有系統中，是依據想要以系統實現什麼事情，來檢討業務的應用程式、系統所需的伺服器、網路等設備，同時執行個別的設計與開發（圖 1-25）。

換言之，開發系統的同時，需要檢討並設計易用程式、伺服器、網路等環節。

更詳細來說，也需要檢討伺服器的作業系統、支援中介軟體等應用程式的環境、系統的資安防護等。

利用雲端運算的系統

以利用應用程式、雲端為例，我們可如下思考（圖 1-26）：

- 具有實現雲端服務的應用程式嗎？
- 若是有的話，適合自身的目標用戶、存取狀況嗎？
- 成本花費、資安防護、多樣的通訊方式齊全嗎？

從廣義來說，其檢討的觀點與不利用雲端服務的情況相同，檢討的內容包含服務的有無、適性、服務水準等，業者可進行**服務調查**來「**評價**」，再一面試用服務，一面尋找適合自己的利用方式。

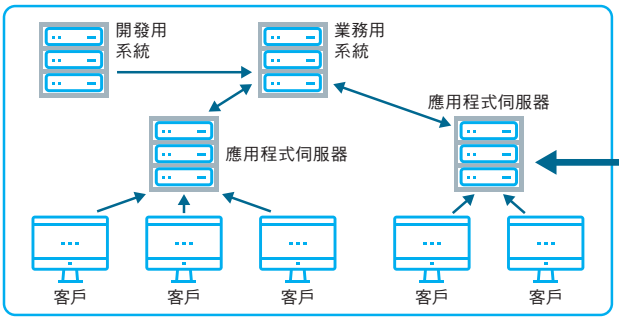
自行建構系統需要著手設計、開發等細瑣作業，但若利用雲端既存的服務，就能輕鬆又快速地完成。

圖1-25 系統的檢討

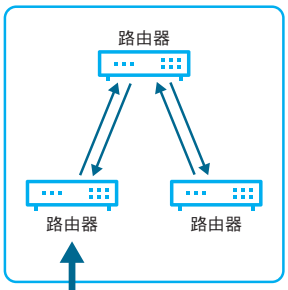
❶業務應用程式的檢討

- 東京總公司與大阪分公司全體職員的訂貨系統
- 由於近年商品增加，預計需要追加系統功能

❷系統配置的檢討

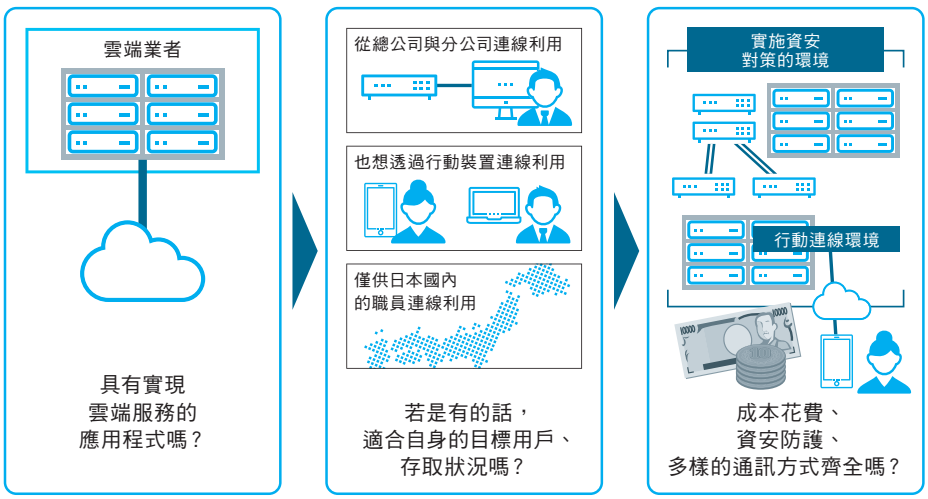


❸網路的檢討



各設備性能、
配置等的檢討

圖1-26 業務應用程式的雲端服務檢討範例



Point

- 以雲端為前提的話，尋找、評價適當的服務很重要。
- 雲端逐漸從「自行建構」轉為要求「尋找並運用自如」的能力。

» 跟伺服器的導入比較來看 IaaS

系統配置的視點

假設已經存在業務與業務應用程式，想想該如何組織建構伺服器（圖 2-13）：

❶ **選擇實體獨占還是虛擬環境？**

首先，思考有沒有需要實體獨占伺服器。關於必須獨占的理由，可舉非公開資訊（例：新產品的設計資料、具有高保密性的個人資訊）、與特殊裝置的實體連線、講求高性能與高回應性等等。

❷ **一台還是複數台？**

接著，無論實體或者虛擬皆用一台實現，還是採用複數台設置備份？此觀點在建構網路時也要考慮。

這是鑒於系統的重要性，即便遭逢災害等災害復原（Disaster Recovery）、重大故障等也能繼續執行業務，而準備複數配套的構想。這也能夠設定、選擇不同的地域。

❸ **有做好業務資料、應用程式的故障對策？**

到上述的❷，即便伺服器端做好系統環境遭逢災害、故障的防護對策，也需要檢討軟體端的因應對策。**就地部署的伺服器導入也是同樣的情況。**

運行維護

除了系統配置外，還有該如何監視系統是否正常運行、是否發生故障等的觀點。

另外，變更、增強系統配置的方法；變更時應用程式的運用、停止步驟等，也需要進行檢討（圖 2-14）。重要的是釐清服務選單，而這基本上可由搭載於 **IaaS** 的**系統重要性來決定**。

圖2-13 檢討伺服器配置的步驟

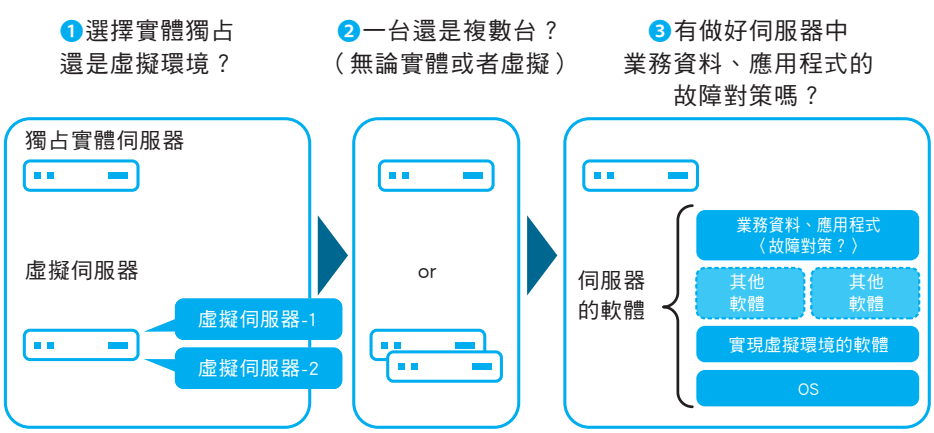
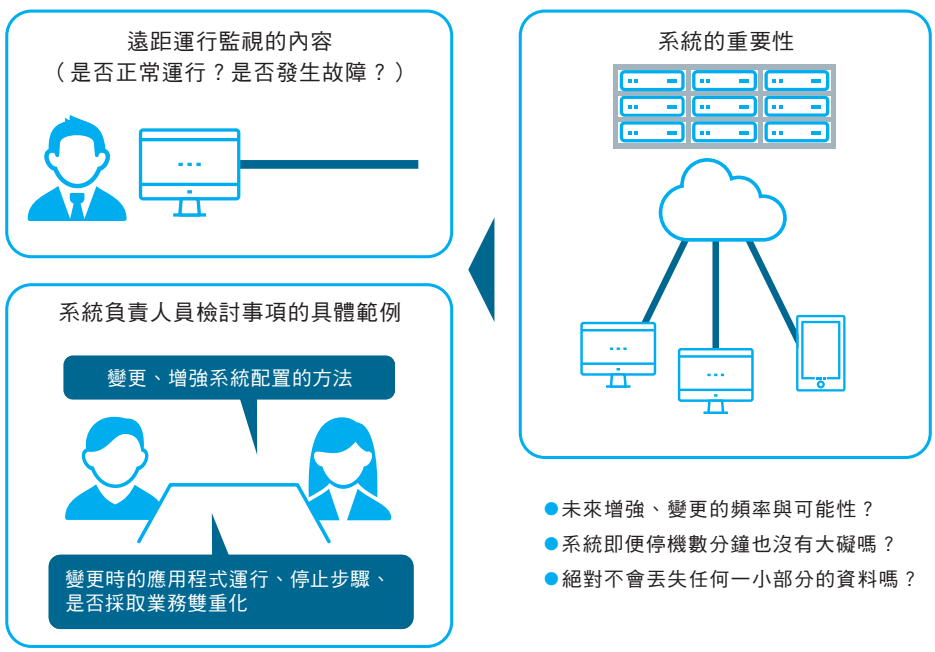


圖2-14 由系統的重要性決定如何運行、維護



Point

- 在檢討 IaaS 的導入時，可採取與就地部署的伺服器系統配置相同的思維。
- 系統配置可由目標系統的重要性來判斷。

» 虛擬化技術①
～ Hypervisor 型態～

Hypervisor 型態

VMWare vSphere Hypervisor、Hyper-V、Xen、Linux 的功能之一 KVM，在虛擬化軟體中被稱為 **Hypervisor 型態**，是**引領現今虛擬化場景的產品**。

Hyper-V 是微軟提供的虛擬化軟體，可在相對較新的 Windows 作業系統勾選使用。Hyper-V 能夠免費立即使用，或許也是虛擬技術廣泛滲透的理由之一。

Hypervisor 型態是目前虛擬化軟體的主流，但作為實體伺服器上的虛擬化軟體，尚需搭載 Linux、Windows 等的**客機作業系統（Guest OS）**來運行。由於客機作業系統與應用程式所構成的虛擬伺服器（虛擬機器），運行上不受**主機作業系統（Host OS）**的影響，所以能夠有效率地運行虛擬伺服器。以前也有 Host OS 型態的虛擬化技術（圖 4-9）。

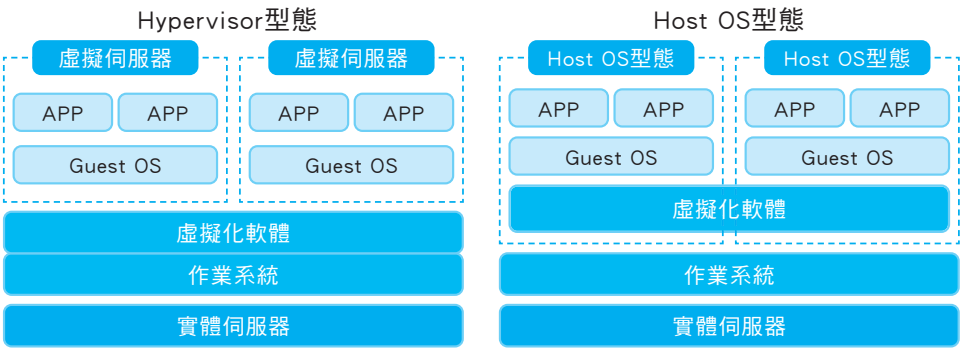
尋求更好的開發環境

雖然虛擬化技術已經透過各種軟體產品普及開來，但軟體的開發人員總是想要利用具備優質高性能伺服器、網路設備的開發環境。

然而，在 Hyper-V 虛擬環境開發的系統無法移轉至 VMWare 的環境，反之亦然，需要在移動目的地的伺服器建構同樣的虛擬環境（圖 4-10）。

雖然前面說過在虛擬環境下容易移轉系統，但其前提是作為基礎的虛擬化軟體、作業系統相同。另外，過去也曾遇到虛擬機器本身容量龐大的問題，但該問題過沒有多久就獲得解決。

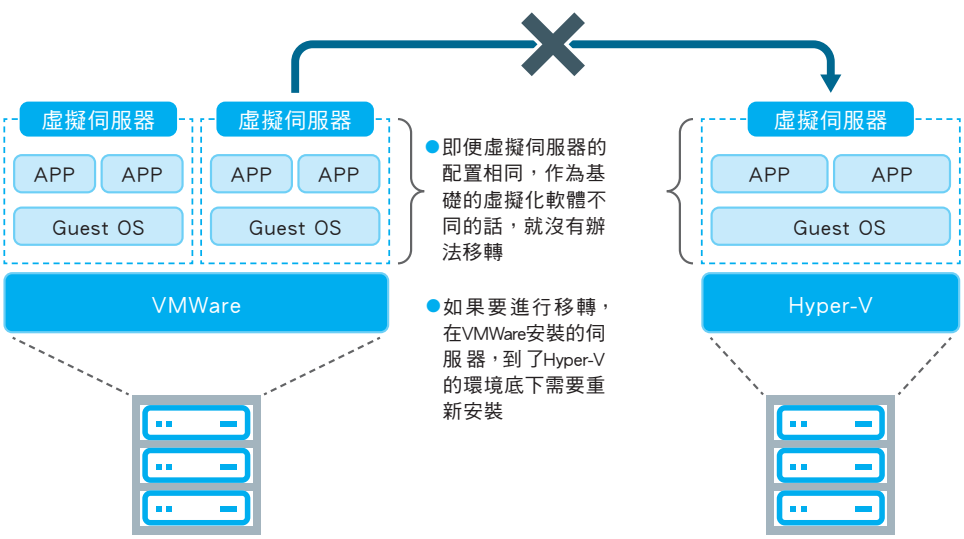
圖4-9 Hypervisor 型態與 Host OS 型態



- 作業系統與虛擬化軟體幾乎融為一體，可提供完整的虛擬環境
- 故障發生時，難以區別是虛擬化軟體還是作業系統出問題
- 大多為相對較新的系統採用

- 從虛擬伺服器存取實體伺服器時是經由主機作業系統，容易發生速度降低的情況
- 故障發生時，比Hypervisor容易區別問題原因
- 在傳統的任務導向系統具有堅實的人氣

圖4-10 虛擬伺服器的移轉問題



- 即便虛擬伺服器的配置相同，作為基礎的虛擬化軟體不同的話，就沒有辦法移轉
- 如果要進行移轉，在VMWare安裝的伺服器，到了Hyper-V的環境底下需要重新安裝

Point

- Hypervisor 是現今虛擬化技術的主流。
- 雖然說虛擬機器容易複製、移轉，但這是在滿足前提條件下的情況。

» 虛擬化技術②～容器型態～

實現高速處理

容器型態將會是今後的虛擬化技術主流。過去的虛擬化軟體是，在虛擬機器上啟動客機作業系統，再呼叫應用程式運行，此過程需要複雜的處理。

在容器型態的配置中，客機作業系統可藉由共用主機作業系統的核心功能達成輕量化。容器內的客機作業系統僅含有最低限度的函式庫，能夠實現減輕 CPU、記憶體負擔的**高速處理**（圖 4-11）。

輕量化是其特徵

容器型態能夠順暢地啟動應用程式、改善資源的使用效率，但優點不僅止如此。容器型態有時也被稱為**輕量虛擬化的基礎**，能夠輕量化縮小由客機作業系統、應用程式所構成的虛擬機器封包。

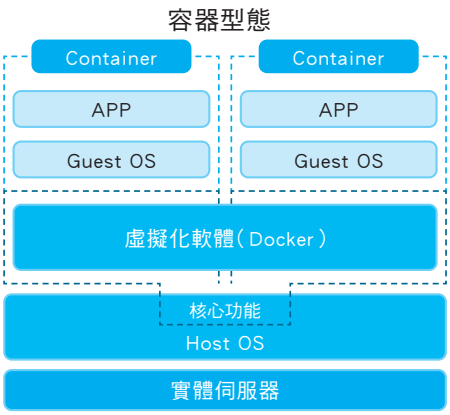
封包縮小有什麼好處呢？這有助於應用程式系統的移轉。

容器的作成是使用名為 **Docker** 的軟體。如圖 4-12 在容器基礎上作成的虛擬機器（容器），能夠以「容器單位」移轉至持有其他容器環境的伺服器（安裝了輕量虛擬化基礎環境 Docker 的客機作業系統）。

容器環境本身也相對容易建構。

當然，由於跟過去的虛擬環境設計不同，需要具備容器型態的專業知識技術。

圖4-11 容器型態的概要

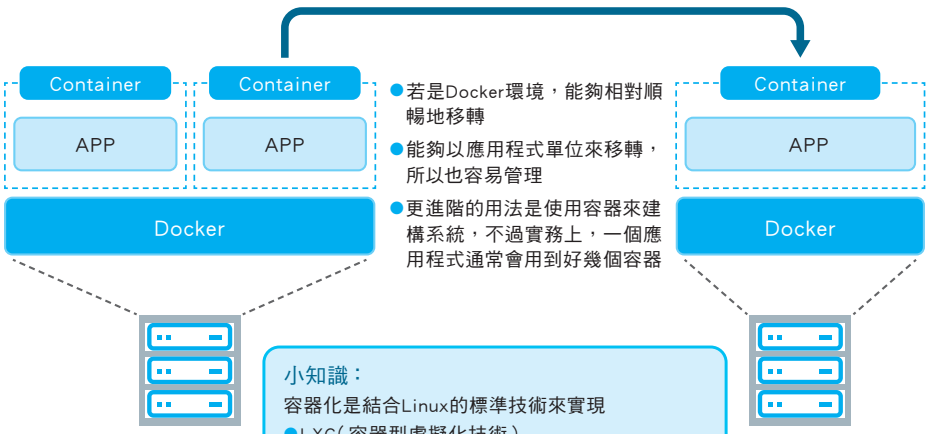


小知識：

- 容器化原本是Docker公司提供作為PaaS基礎功能而開發的技術
- 2013年3月登場後大受歡迎，確立了容器管理技術在實務上的地位
- 是以谷歌開發的Go語言所寫成

- 虛擬化軟體(Docker)將單一作業系統分割成名為容器的用戶箱子
- 每個箱子能夠獨自使用實體伺服器的資源
- 容器的客機作業系統可共用主機作業系統的核心功能

圖4-12 Docker 環境下的移轉



- 若是Docker環境，能夠相對順暢地移轉
- 能夠以應用程式單位來移轉，所以也容易管理
- 更進階的用法是使用容器來建構系統，不過實務上，一個應用程式通常會用到好幾個容器

小知識：

容器化是結合Linux的標準技術來實現

- LXC(容器型虛擬化技術)
- namespaces(程序的分離)
- cgroups(CPU等的資源控制)
- AUFS(檔案、目錄的差分管理)

Point

- 容器型虛擬機器實現了比過往更快速的處理。
- 容器化是以 Docker 作為實行環境。

» 何謂微服務？

微服務的特徵

2-2 曾經提到容器化與微服務是極具代表性的雲端技術。**微服務整合許多的小型服務以提供大型服務的一種架構**，因此，即使修正個別的小服務，也不會對其他服務造成影響。

具體而言，微服務是透過 **API**（Application Interface）呼叫個別應用程式來使用。

舉例來說，過去的系統不是藉由 API 呼叫個別的應用程式，進行用戶認證、資料輸入與存放、資料分析等，而是以應用程式內的原始碼來定義。因此，變更特定的應用程式時，也得變更其他的應用程式。而微服務是以 API 連結應用程式，即便變更某項程序，其他程序不需要跟著改變（圖 4-20）。

微服務與容器化的效果

將以 API 連結的應用程式掛載至個別的容器後，不但容易變更應用程式，也可輕鬆進行移轉。

若能夠同時活用前面解說的容器化、Kubernetes 與微服務等技術，應用程式就可**尋求當下的最佳環境，移轉至適合自己的虛擬伺服器、實體伺服器**。

整備好容器化、編配以及微服務等的環境後，就有可能穿梭在不同的雲端業者之間。

圖4-19 微服務的思維

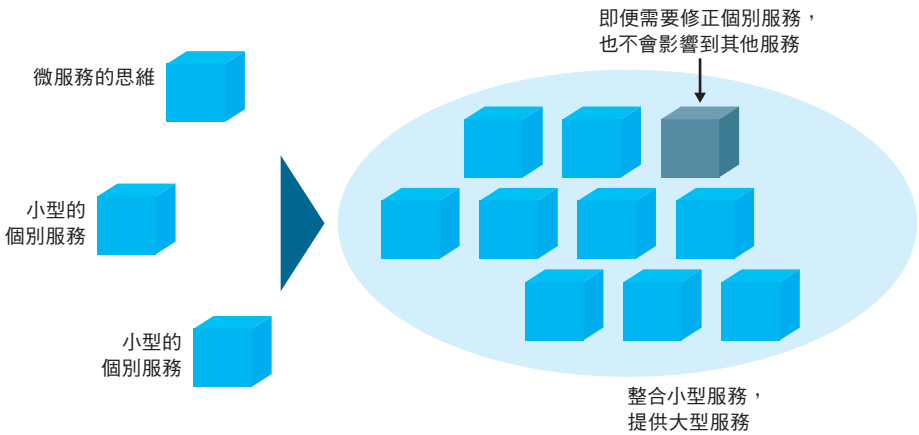
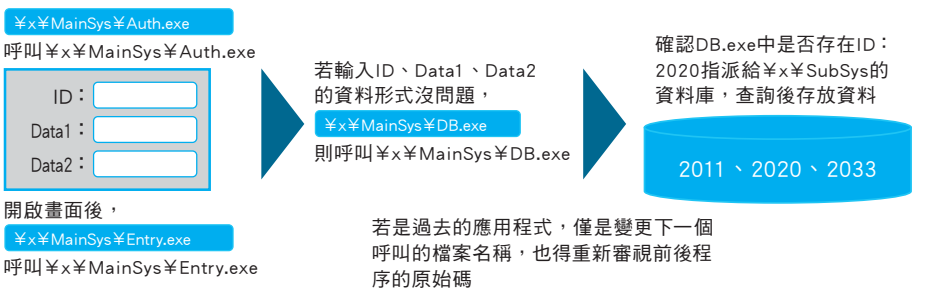


圖4-20 API 連結的便利性範例

〈透過原始碼呼叫的過往系統〉



〈透過API連結的微服務〉

例如，透過API定義下一個呼叫的應用程式，
就不需要返回原始碼進行變更

Point

- 微服務是指，整合小型服務來提供大型服務的思維。
- 微服務與容器化提升了雲端環境的自由度。