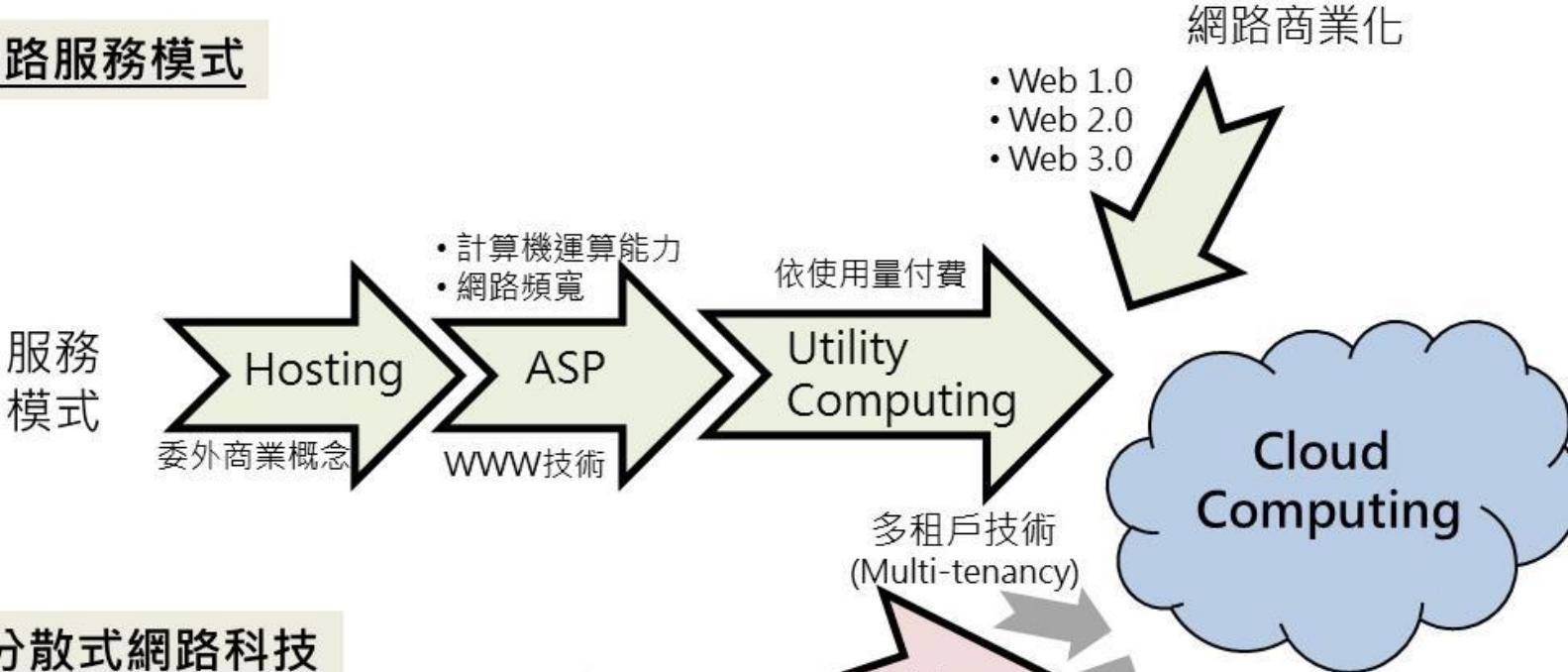


# 雲端運算概論

# 2-1 雲端運算演進

## 1. 網路服務模式



## 2. 分散式網路科技



# 企業端服務模式-1

服務模式	說明	服務方式	計價方式
Hosting	企業電腦、網站代管服務	<ol style="list-style-type: none"><li>1. <u>資源歸屬</u>: 電腦、軟體為企業客戶擁有</li><li>2. <u>資源運算地點</u>: 代管服務提供者資料中心</li><li>3. <u>資源共享方式</u>: 集中共享網路電力、機房設備基礎設施</li></ol>	每月或每年固定收費
ASP	應用服務供應商	<ol style="list-style-type: none"><li>1. <u>資源歸屬</u>: 電腦、軟體為服務提供者擁有</li><li>2. <u>資源運算地點</u>: 服務提供者資料中心</li><li>3. <u>資源共享方式</u>: 共享機房基礎設施、共享主機</li></ol>	每月或每年固定收費

# 企業端服務模式-2

Utility Computing	公用運算	<ol style="list-style-type: none"><li><u>資源歸屬</u>: 公用運算服務提供者 (多為 IT 服務商，如: HP, IBM, Sun Micro )</li><li><u>資源運算地點</u>: 公用運算服務提供者資料中心</li><li><u>資源共享方式</u>: 並無確切說明。有可能是企業各部門軟硬體共享，集中於服務提供者資料中心</li></ol>	如同水電依據使用量計費
Cloud Computing	雲端運算	<ol style="list-style-type: none"><li><u>資源歸屬</u>: 軟硬體設備均為服務提供者擁有</li><li><u>資源運算地點</u>: 服務提供者資料中心</li><li><u>資源共享方式</u>: 可能共享軟體(SaaS)共享平台(PaaS)或共享基礎軟硬體(IaaS)及機房基礎設施</li></ol>	<ol style="list-style-type: none"><li>根據資源使用量計費</li><li>使用者可以彈性地選擇各種計價模式與服務等級</li></ol>

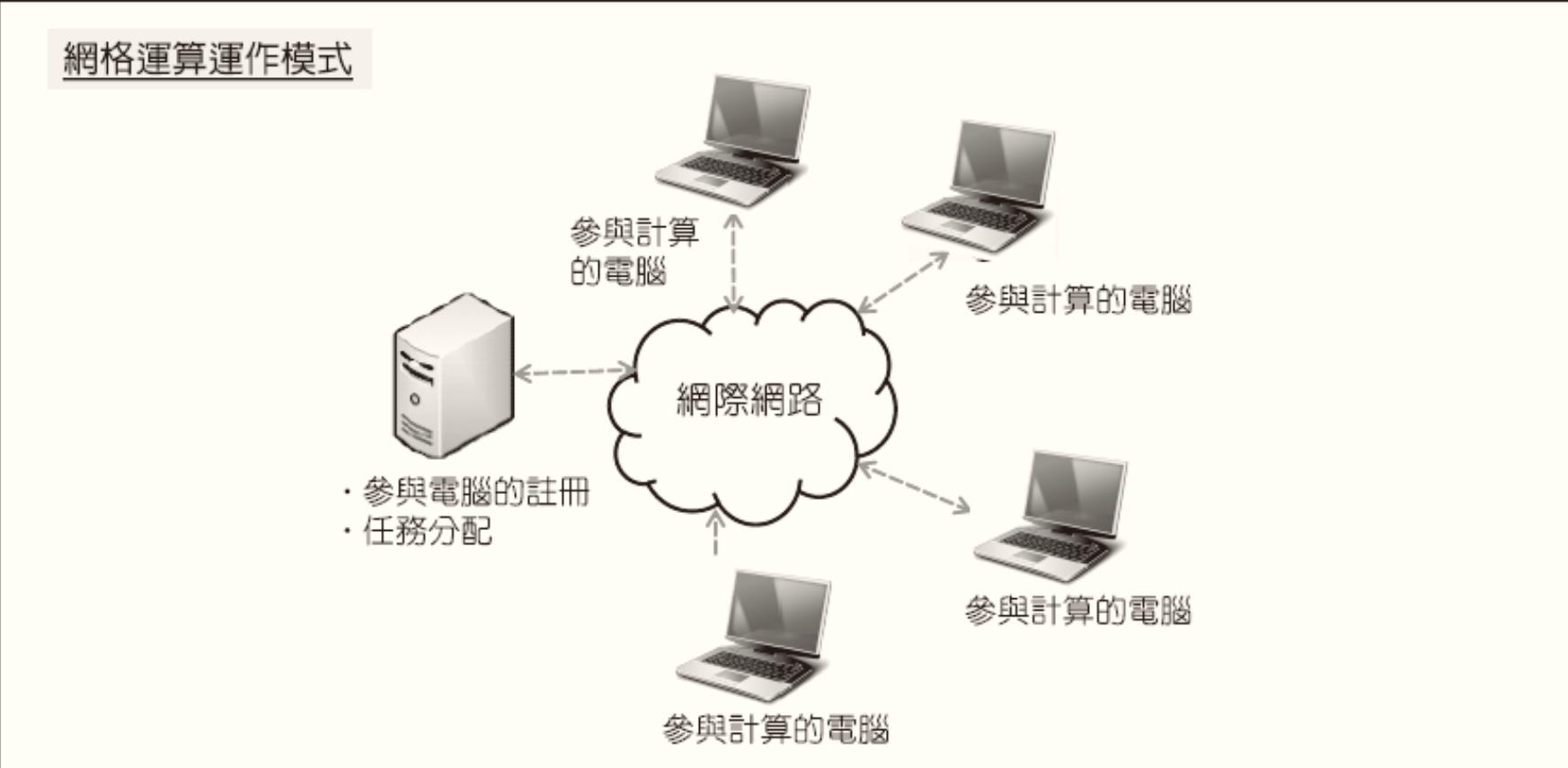
# 消費端服務模式

模式	服務方式	利潤模式
Web 1.0	全球資訊網( <a href="http://www">www</a> )利用超連結的方式，將不同的媒體(聲音、影像、文字)以非線性的方式相連，讓使用者更方便地取得各式豐富資訊，如： <a href="#">新聞網站</a> 、 <a href="#">Yahoo!奇摩入口網站</a>	交易抽成費用、網路服務的購買以及廣告刊登營收
Web 2.0	社交網站，如： <a href="#">MySpace</a> 、 <a href="#">Flicker</a> 、 <a href="#">YouTube</a> 讓使用者間可以互相分享各項資訊	廣告刊登營收、網路服務購買
Web 3.0 (Cloud Computing)	使用者不僅僅透過個人電腦或筆記型電腦上網進一步利用智慧手機、聯網電視等不同載具上網。透過不同載具上網可以獲得不同的資訊、服務、App軟體與社群互動	情境為主的廣告刊登、服務購買、服務交易

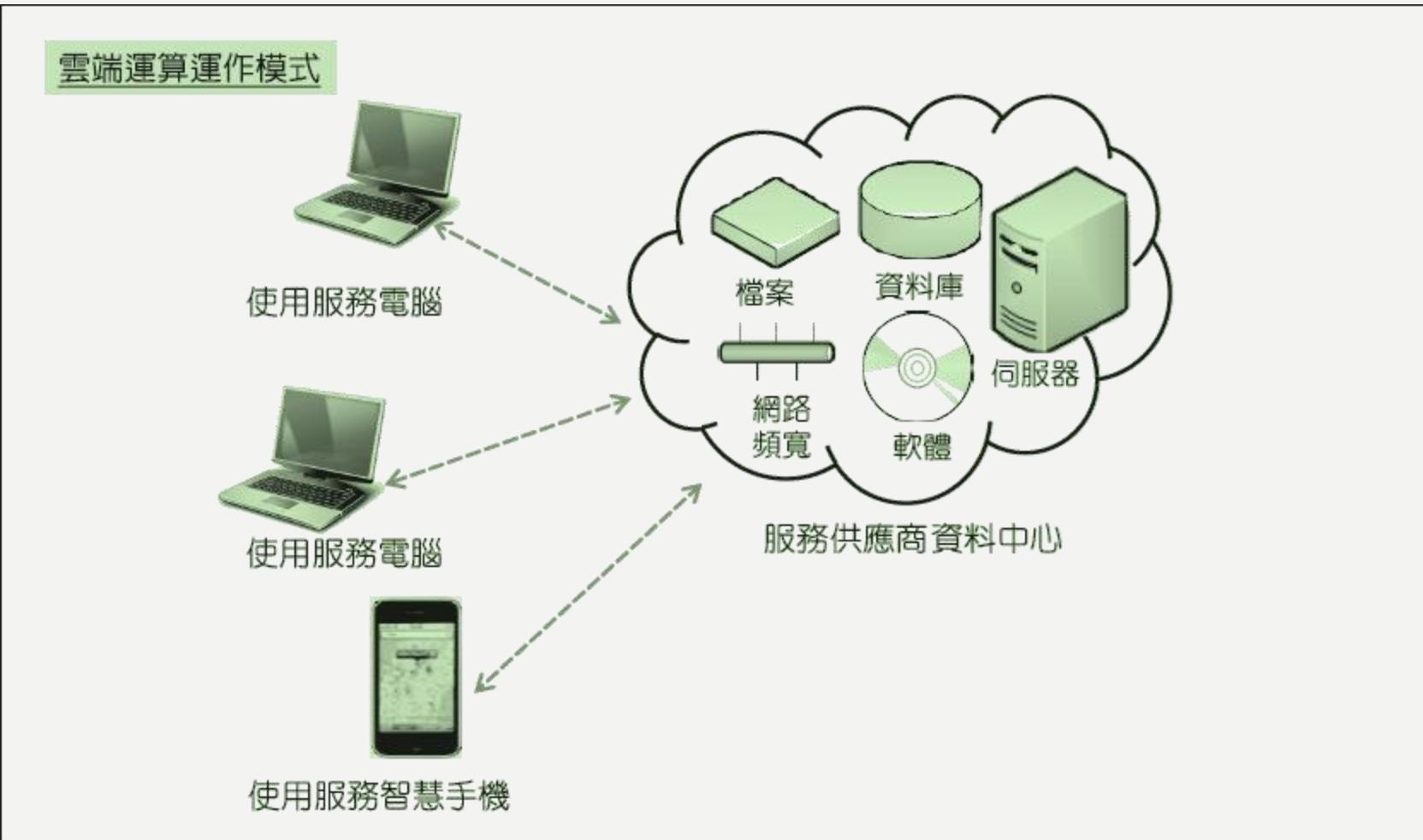
# 網格運算與雲端運算比較-1

特點	網格運算	雲端運算
應用領域	科學研究專案	商業服務營運
使用對象	科學研究人員	企業、一般大眾
解決問題	協同運算以解決複雜問題	分享運算資源以降低成本
運算模式	分散的資源協同	集中的資源池
案例	<b>SETI@Home</b> 集合各個聯網電腦資源，尋找外星人	企業訂用會計軟體即服務，減少購買與維護軟體的成本

# 網格運算與雲端運算比較-2

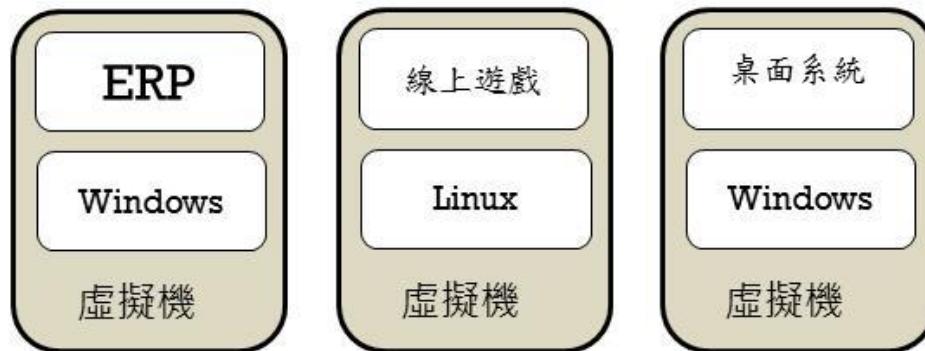


# 網格運算與雲端運算比較-3



# 虛擬化技術-1

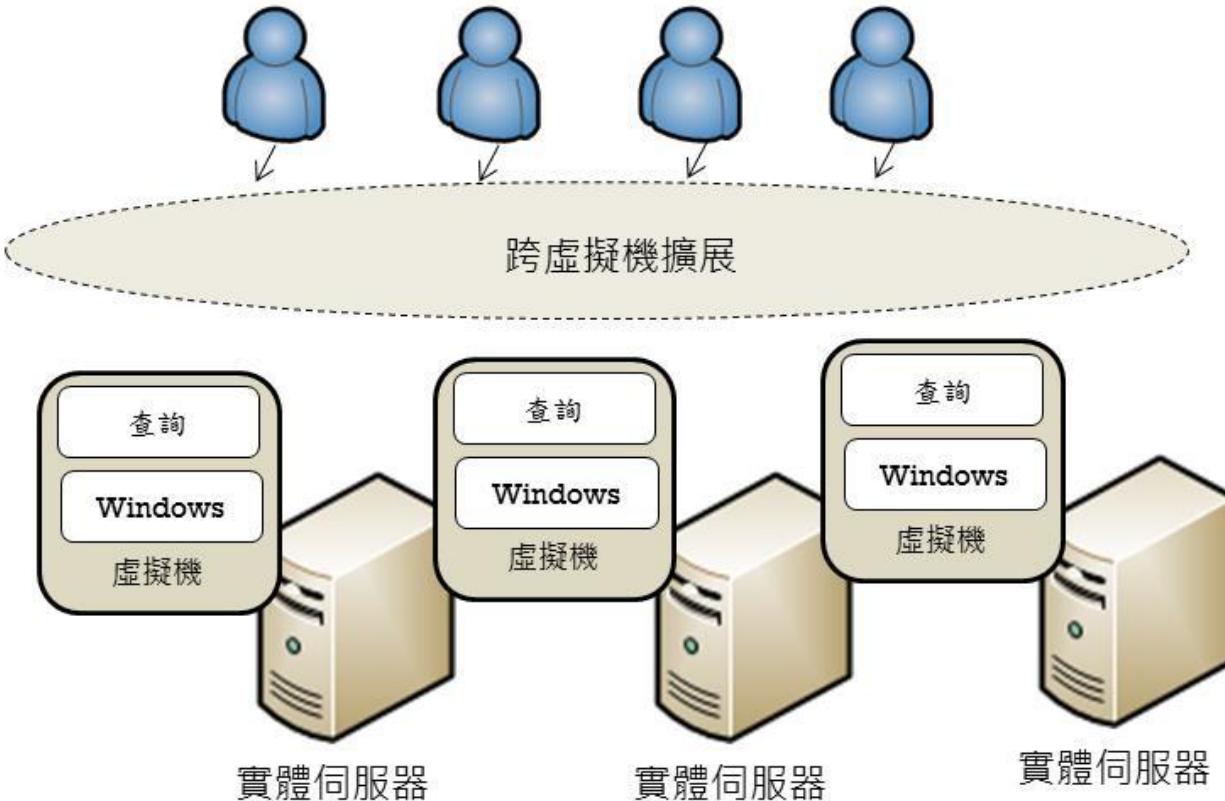
## 向內擴展式虛擬技術應用方式



實體伺服器

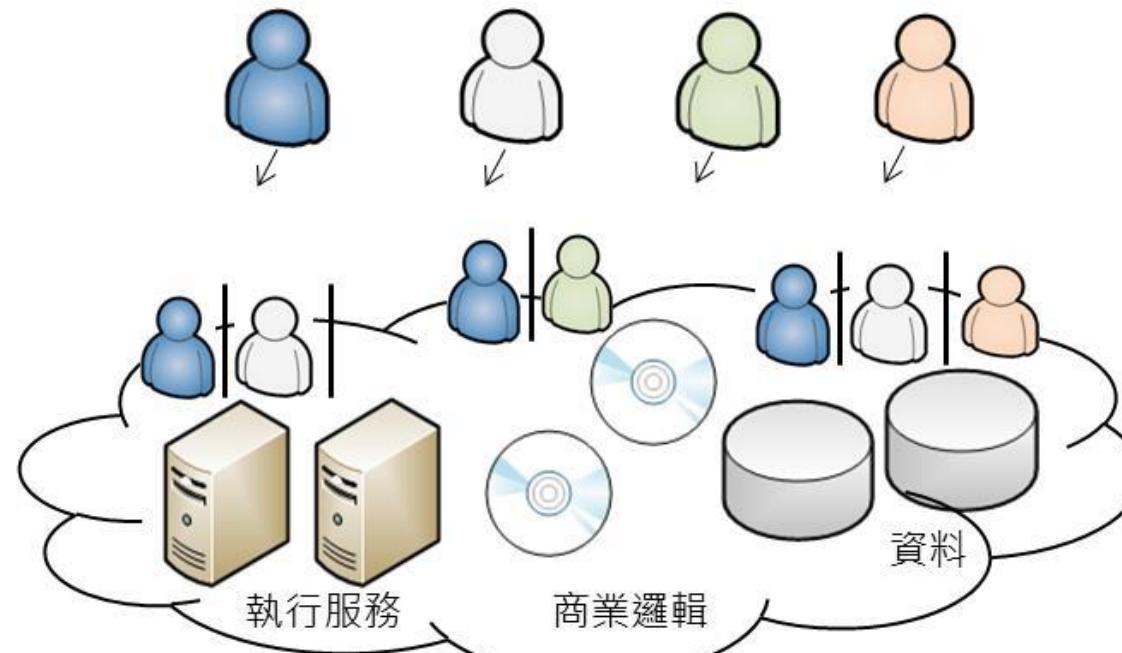
# 虛擬化技術-2

## 向外擴展式虛擬技術應用方式



# 多租戶架構概念

多租戶架構概念



## 2-2 雲端運算意義

- 雲端運算是一種網路服務的商業模式，其關鍵在於企業或消費者隨選所需，使用服務商所提供的運算資源(如：服務、應用程式、網路連線、電腦計算或儲存等)。服務契約方式如同水電般以使用量計價，且使用者可以彈性地調整所需資源的多寡。
- 利用上述商業模式所提供的服務，稱為雲端服務；實現雲端運算服務的軟硬體技術，稱之為雲端運算技術。

# 雲端運算特點

## NIST雲端運算特點

隨選所需

廣泛網路  
存取

快速與  
彈性調整

可衡量  
服務

集中資源

5個  
重要特徵

軟體即服務  
(SaaS)

平台即服務  
(PaaS)

基礎架構即服務  
(IaaS)

3種  
服務模式

公眾雲  
(Public  
Cloud)

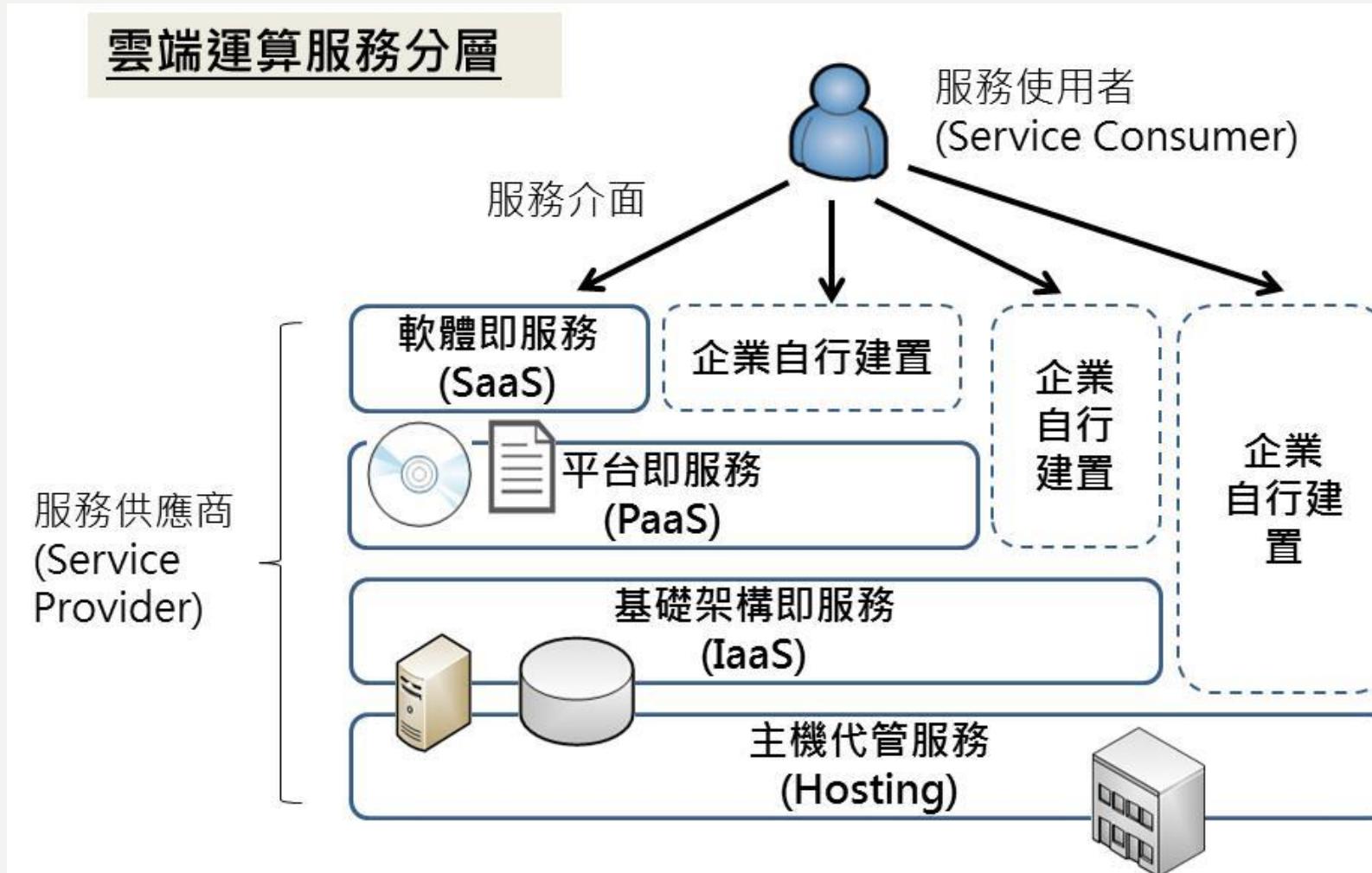
私有雲  
(Private  
Cloud)

混合雲  
(Hybrid  
Cloud)

社群雲  
(Community  
Cloud)

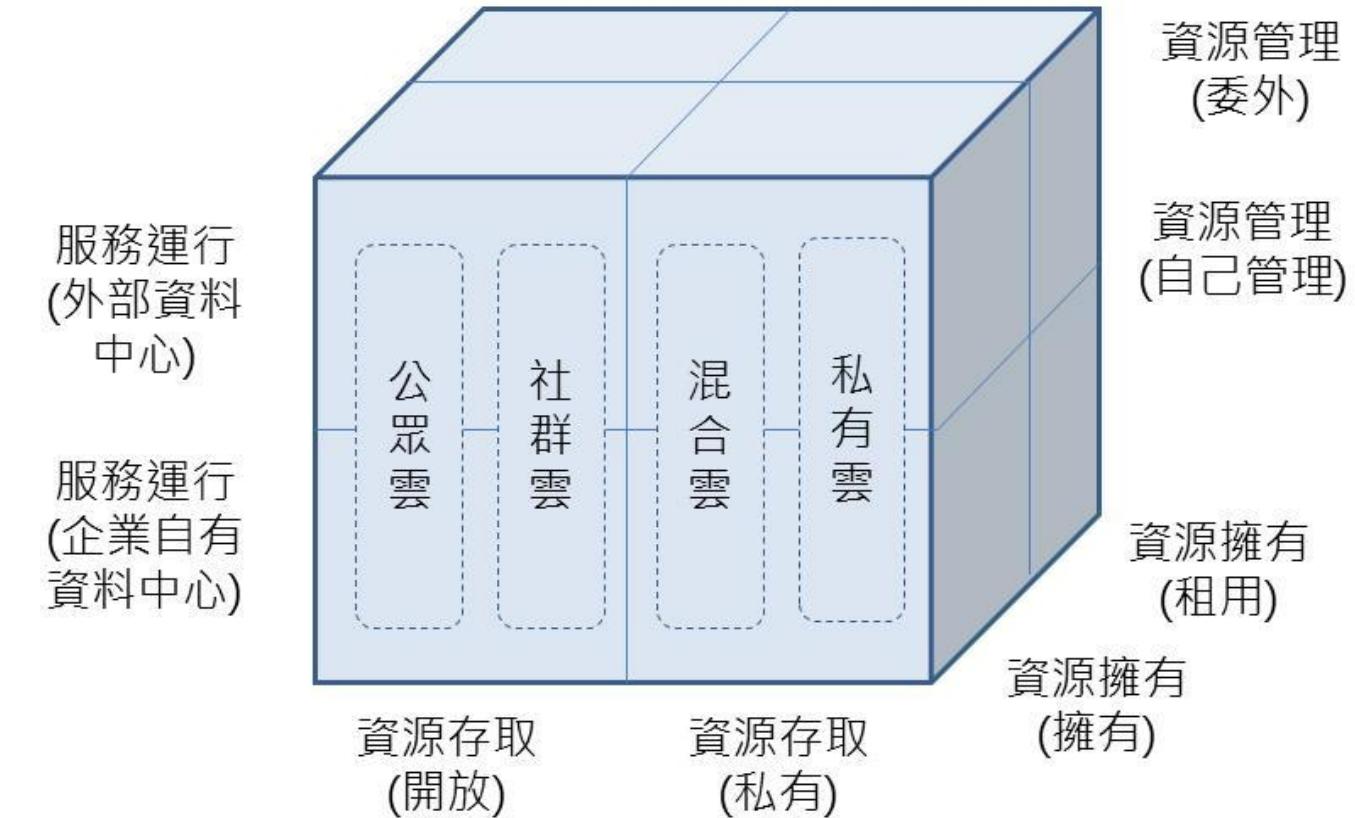
4類  
佈署方法

# 雲端運算服務分層



# 雲端運算佈署模式

## 雲端運算佈署方式選擇



# 雲端運算控制權與選擇

## 雲端運算控制權與選擇

私有雲,  
自有機房

應用軟體

軟體平台

虛擬架構

資訊硬體

機房設施

私有雲,  
代管機房

應用軟體

軟體平台

虛擬架構

資訊硬體

機房設施

公眾雲  
IaaS

應用軟體

軟體平台

虛擬架構

資訊硬體

機房設施

公眾雲  
PaaS

應用軟體

軟體平台

虛擬架構

資訊硬體

機房設施

公眾雲  
SaaS

應用軟體

軟體平台

虛擬架構

資訊硬體

機房設施

服務供應商控制程度

企業(使用者)控制程度

# 3-1 SaaS服務模式與案例-1

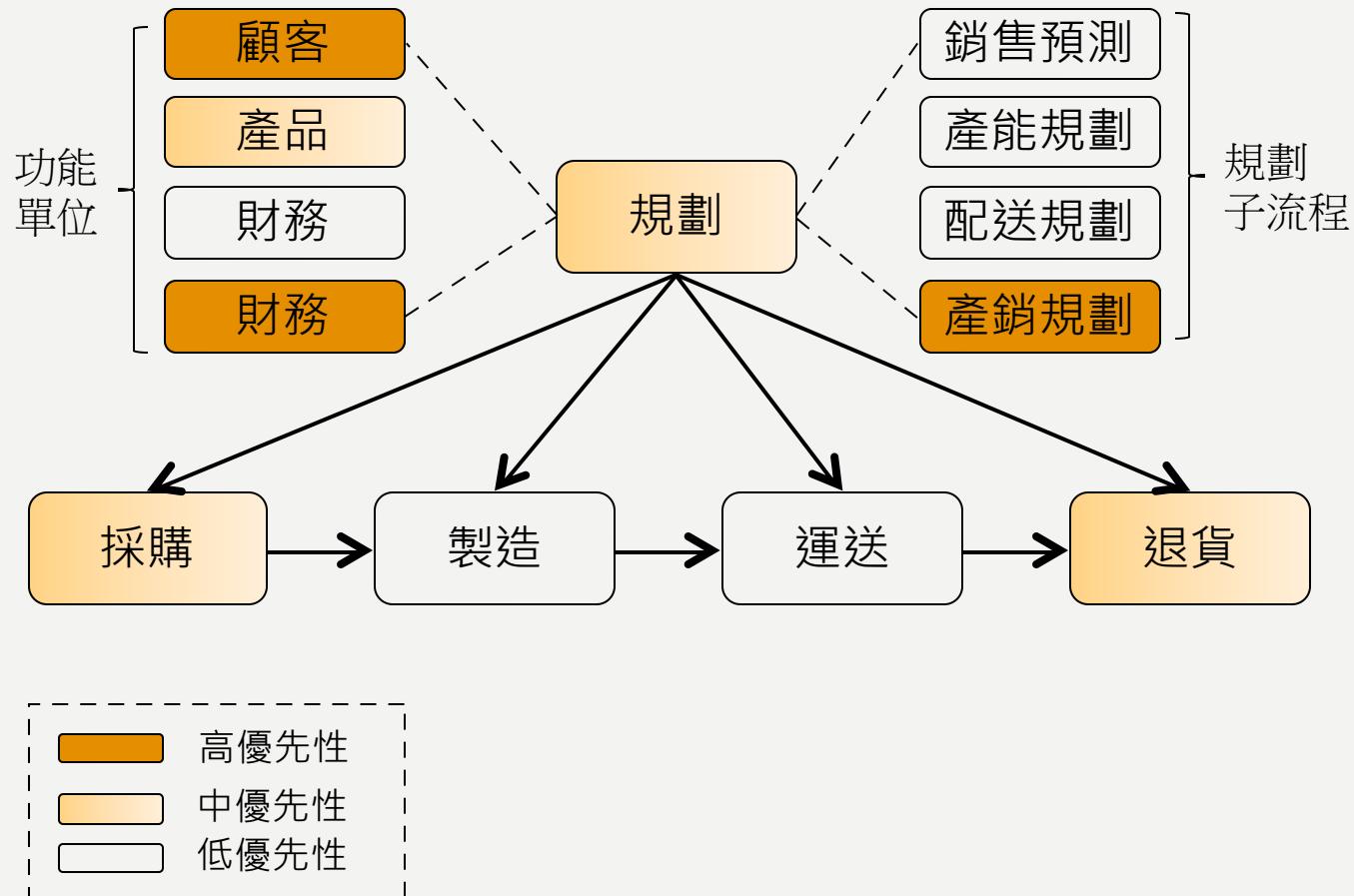
SaaS服務	說明
意義	提供顧客軟體租用或客製的雲端服務模式
特性	<ul style="list-style-type: none"><li>● 網路存取</li><li>● 集中地點管理</li><li>● 「一對多」方式遞送</li><li>● 使用者不須考慮軟體授權與升級</li><li>● 依使用量付費</li><li>● 軟體服務組合</li></ul>
限制	<ul style="list-style-type: none"><li>● 必須透過網路連線</li><li>● 即時資料的取得</li><li>● 敏感性資料的存放</li><li>● 客製化限制</li></ul>
應用方向	<ul style="list-style-type: none"><li>● 與外界接觸較多</li><li>● 常透過網路存取的應用</li><li>● 短暫使用需求</li><li>● 具有週期性使用頻率</li><li>● 與企業流程相關性較低</li></ul>

# 3-1 SAAS服務模式-2

佈署方式	<ul style="list-style-type: none"><li>● 公眾雲</li><li>● 社群雲</li><li>● 混合雲</li></ul>
應用類型	<ul style="list-style-type: none"><li>● CCC：內容、溝通與協同軟體應用</li><li>● CRM：顧客戶關係軟體應用</li><li>● DCC：數位內容軟體應用</li><li>● ERP：企業資源規劃軟體應用</li><li>● Office suites：辦公室生產力軟體應用</li><li>● SCM：供應鏈管理軟體應用</li></ul>

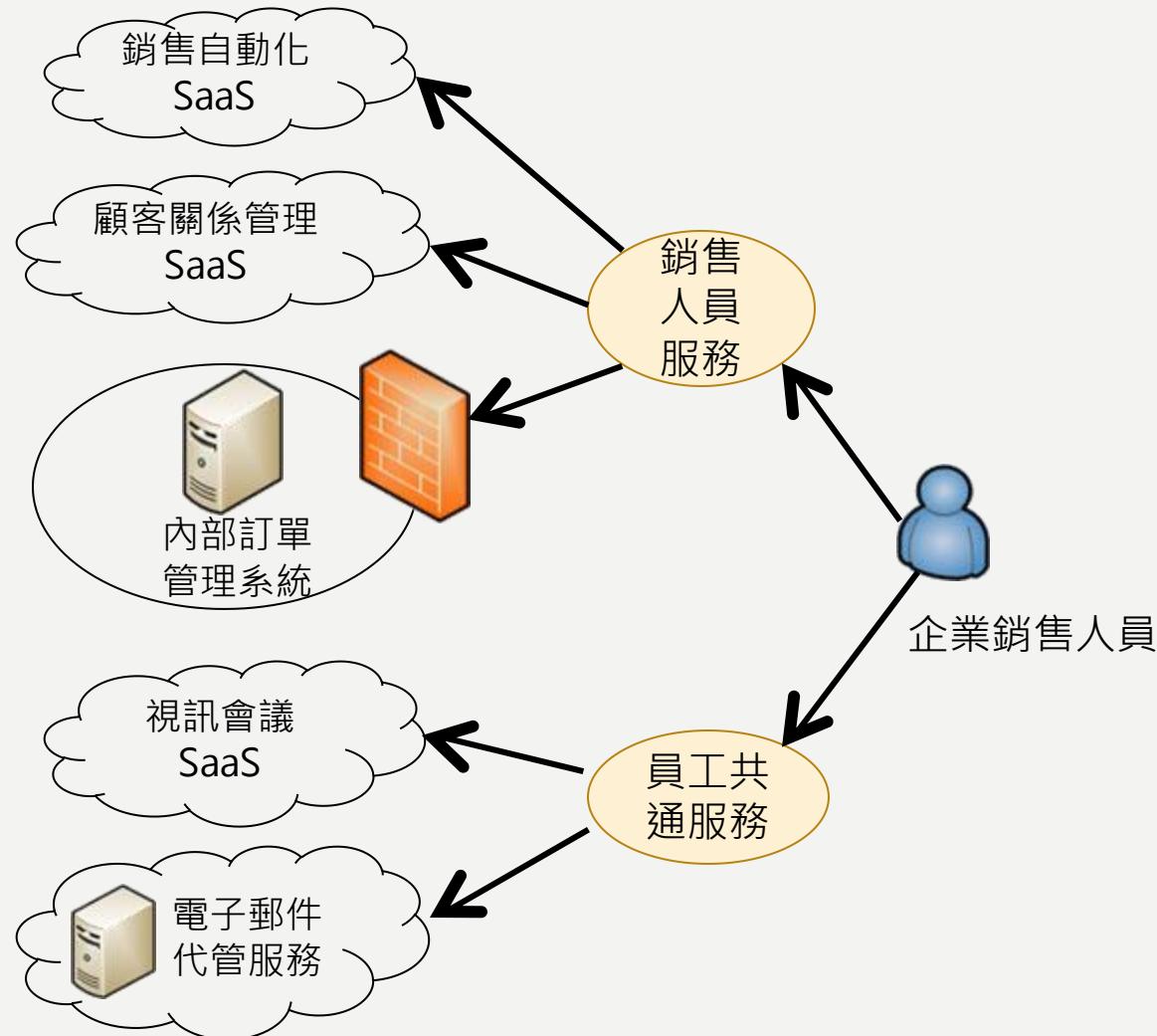
# 各行業SAAS採行需求不同

## 製造業SaaS採行優先順序



# 企業SaaS混合雲佈署情境

企業SaaS混合雲佈署情境



# SALESFORCE.COM案例

- 發展歷史
- 價值主張
- 服務內容
- 營收模式
- 成功關鍵

# 3-2 PaaS服務模式與案例-1

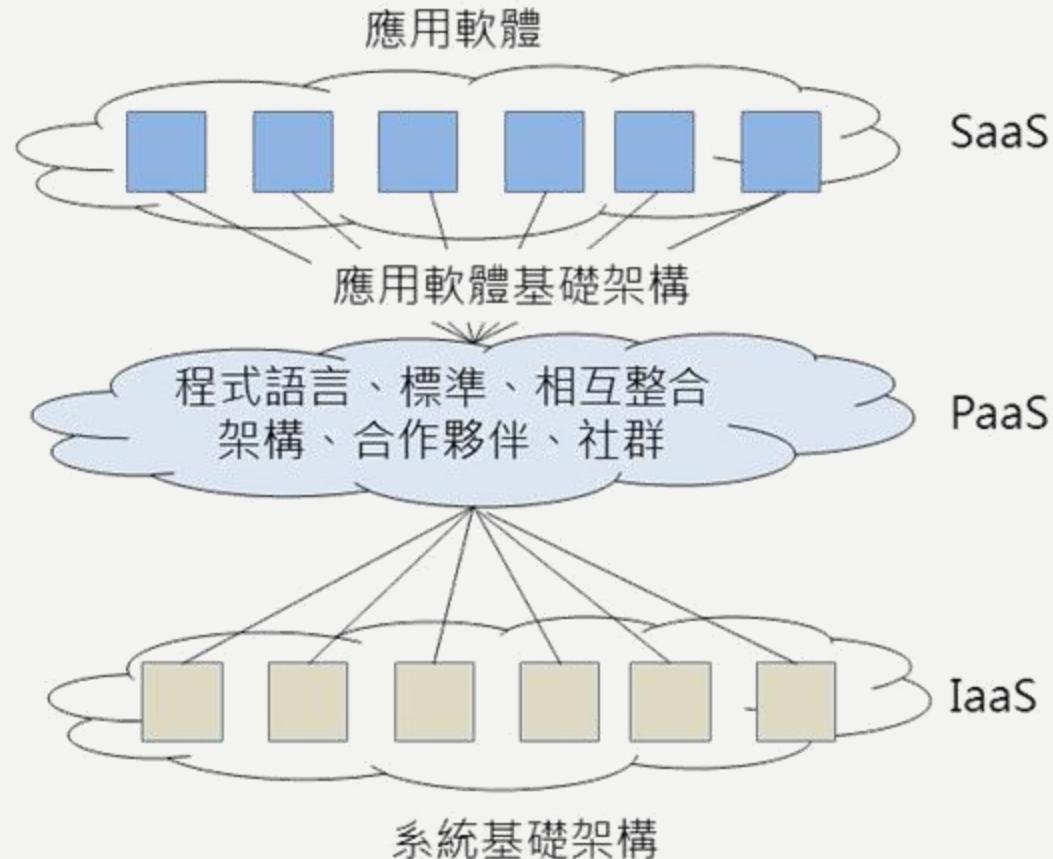
PaaS服務	說明
意義	提供應用程式發展、佈署、整合、中介等平台的雲服務
特性	<ul style="list-style-type: none"><li>● 程式發展生命週期</li><li>● Web介面</li><li>● 多個使用者分享服務</li><li>● 整合能力</li><li>● 使用者協同</li></ul>
限制	<ul style="list-style-type: none"><li>● 必須透過網路連線</li><li>● 移植便利性</li><li>● 語言特殊性</li><li>● 軟硬體資源的彈性</li></ul>
應用方向	<ul style="list-style-type: none"><li>● 多使用者共同開發程式、測試、佈署程式</li><li>● 需進行各項整合</li><li>● 需快速發展程式</li></ul>

# 3-2 PaaS服務模式與案例-2

PaaS服務	說明
佈署方式	<ul style="list-style-type: none"><li>● 公眾雲</li><li>● 社群雲</li><li>● 私有雲</li></ul>
應用類型	<ul style="list-style-type: none"><li>● aPaaS：應用程式的開發、測試、佈署、程式組合、應用市集等</li><li>● iPaaS：使用者介面、流程、訊息、資料、應用程式的中介整合</li><li>● mPaaS：伺服器績效、網路績效、資訊安全管理</li></ul>

# PAAS的中介特色

## PaaS的中介特色



# GOOGLE APE案例

- 發展歷史
- 價值主張
- 服務內容
- 營收模式
- 成功關鍵

# 3-3 IAAS服務模式與案例-1

IaaS服務	說明
意義	提供顧客使用處理程序、儲存、網路或其他基礎運算資源的雲端服務模式
特性	<ul style="list-style-type: none"><li>● 分散式運算資源</li><li>● 動態資源調整</li><li>● 彈性定價</li><li>● 多使用者分享資源</li></ul>
限制	<ul style="list-style-type: none"><li>● 敏感性資料存放</li><li>● 運算績效</li><li>● 無法滿足特殊硬體需求</li></ul>
應用方向	<ul style="list-style-type: none"><li>● 運算資源需求變化性高</li><li>● 短暫運算資源需求</li><li>● 新創組織運算資源需求</li><li>● 組織快速成長資源需求</li><li>● 組織運算資源成本控制</li></ul>

# 3-3 IAAS服務模式與案例-2

IaaS服務	說明
佈署方式	<ul style="list-style-type: none"><li>● 私有雲</li><li>● 公眾雲</li><li>● 社群雲</li></ul>
應用類型	<ul style="list-style-type: none"><li>● CaaS：提供計算服務</li><li>● Storage as a Service：檔案儲存或資料備援的服務</li></ul>

# AMAZON WEB SERVICE案例

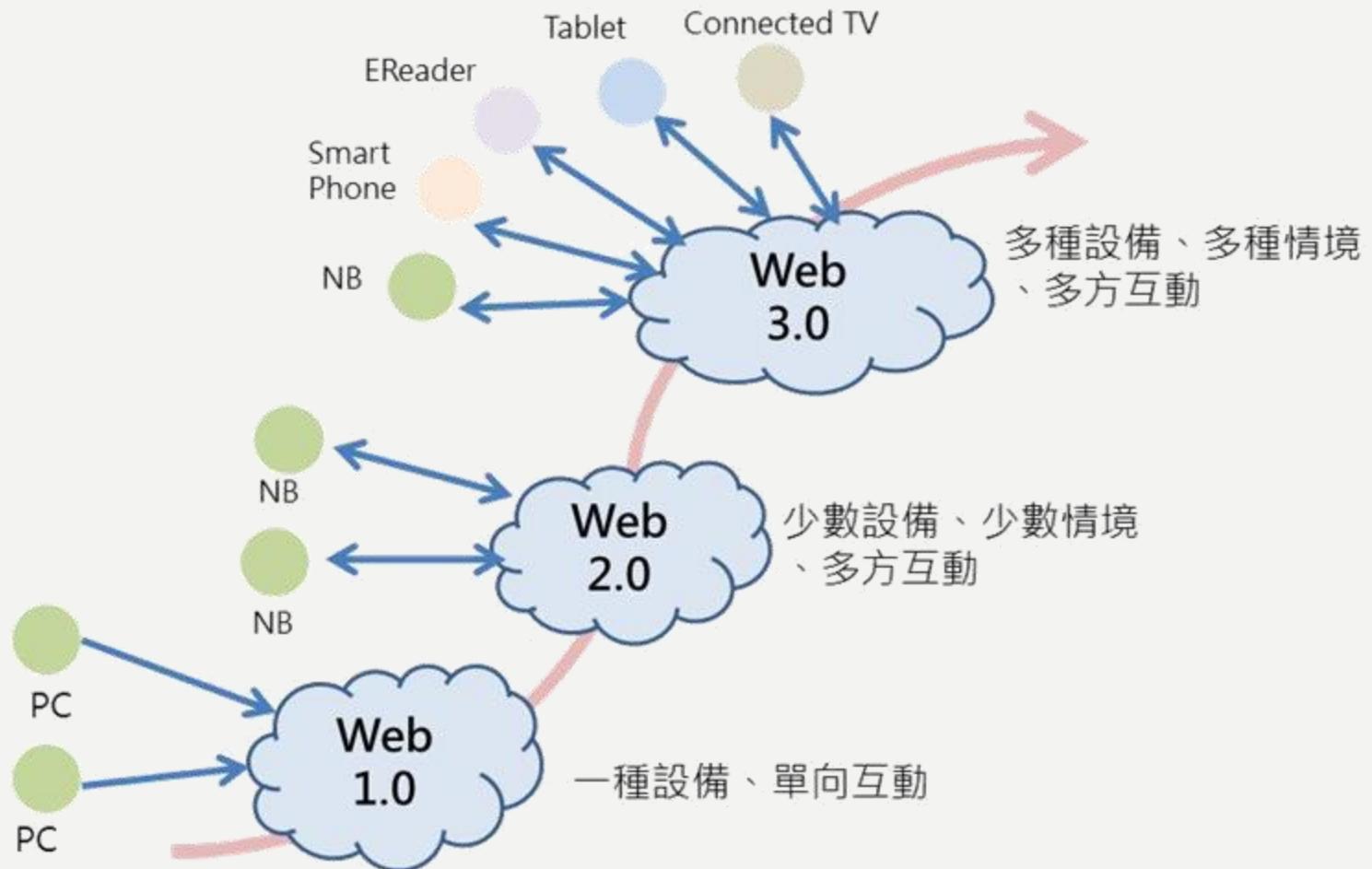
- 發展歷史
- 價值主張
- 服務內容
- 營收模式
- 成功關鍵

# 3-4 WEB 3.0服務模式與案例

Web 3.0 服務	說明
意義	提供使用者可以透過多種載具、在多種情境下與多方使用者進行互動
特性	<p>SoLoMo :</p> <ul style="list-style-type: none"><li>● 多種載具連結</li><li>● 無所不在連結</li><li>● 多種情境</li><li>● 社群網路</li></ul>
應用方向	<ul style="list-style-type: none"><li>● 結合雲端的SoLoMoCo</li><li>● 結合各種情境</li><li>● 結合社群網路</li><li>● 跨載具的連結</li><li>● 結合商務應用</li></ul>
應用類型	<ul style="list-style-type: none"><li>● 消費應用：遊戲、桌面小工具、娛樂、社群、生產力、圖片、音樂、生活型態等等</li><li>● 商務應用：個人生產力、桌面延伸、協同工具、群體協同工具、業務支持、商業智慧、行業別應用等</li></ul>

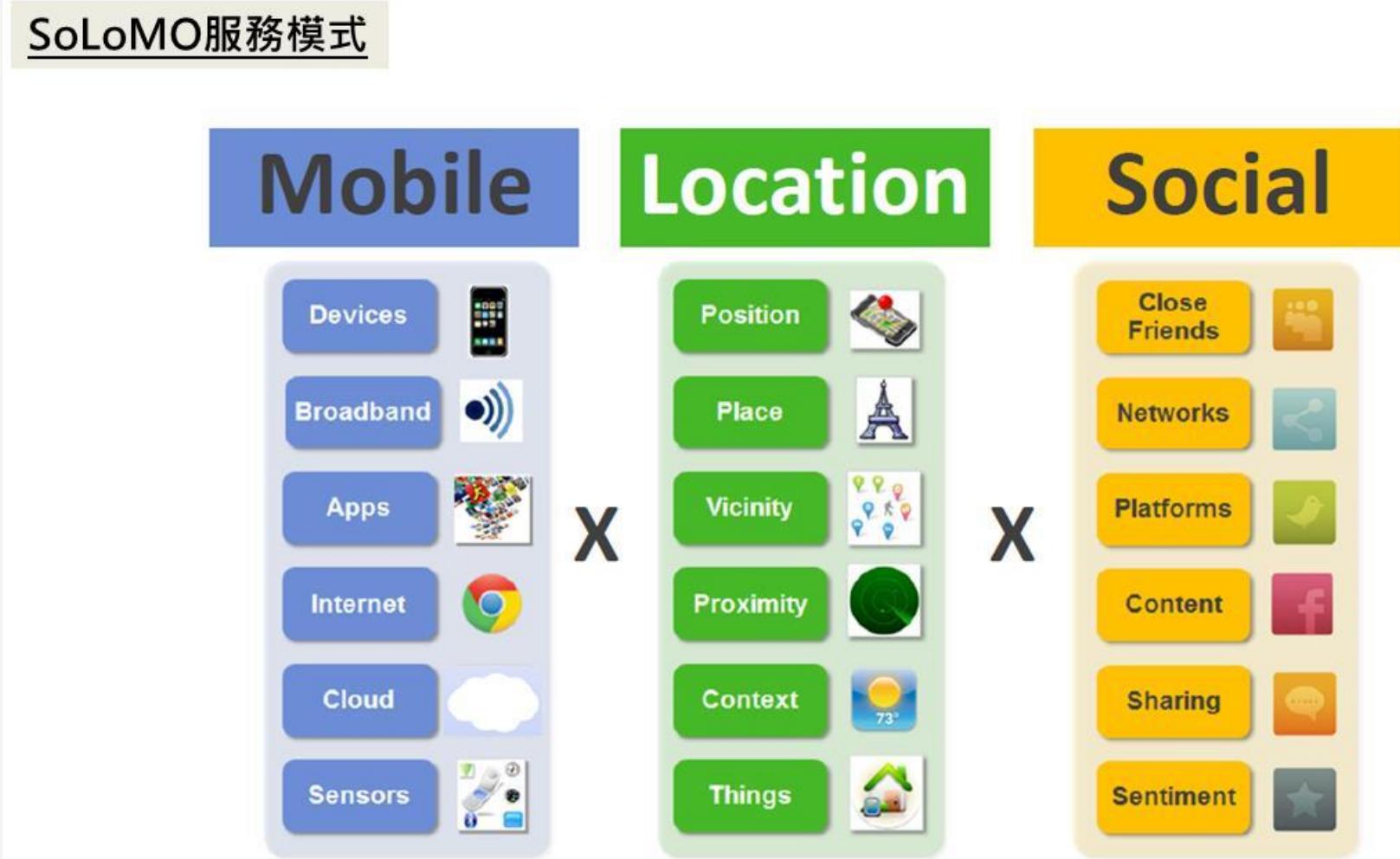
# 消費端網路服務發展

## 消費端網路服務發展

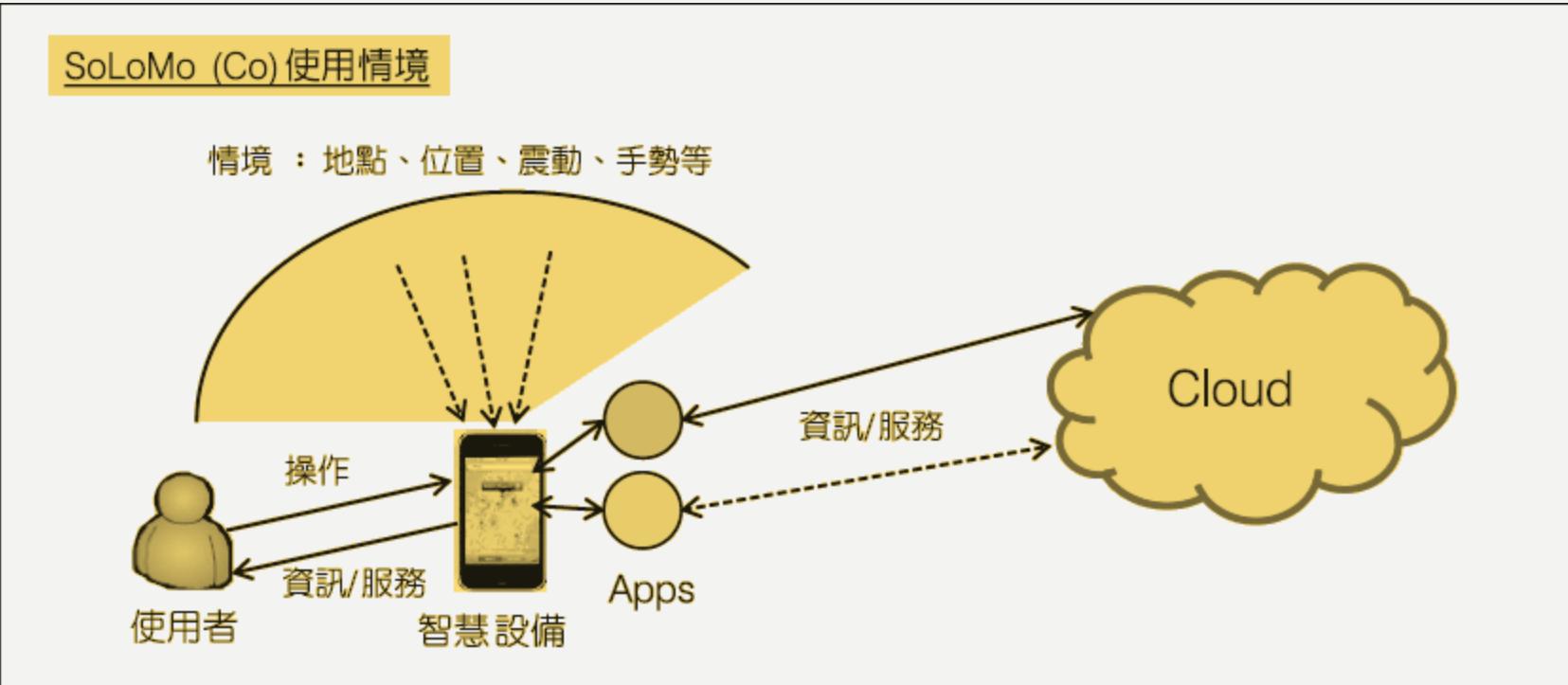


# SOLOMO服務模式

## SoLoMO服務模式



# SOLOMO使用情境



# YAMMER案例

- 發展歷史
- 價值主張
- 服務內容
- 營收模式
- 成功關鍵

## 7.1 SaaS：軟體即服務

- SaaS ( Software as a Service ) 是一種把軟體部署在網路（網際網路、內網或專網）上供使用者（透過瀏覽器）使用的服務模式，服務提供者把應用軟體當成一種服務授權給使用者，使用者或按訂閱（ Subscription ）的方式使用並付費，或按需按次使用並付費。
- 頻寬的增加，以及Web和軟體技術的不斷成熟，加上使用者無須安裝軟體等優勢，越來越受到使用者的歡迎，如圖7-1所示，圖中列出了六大優勢。

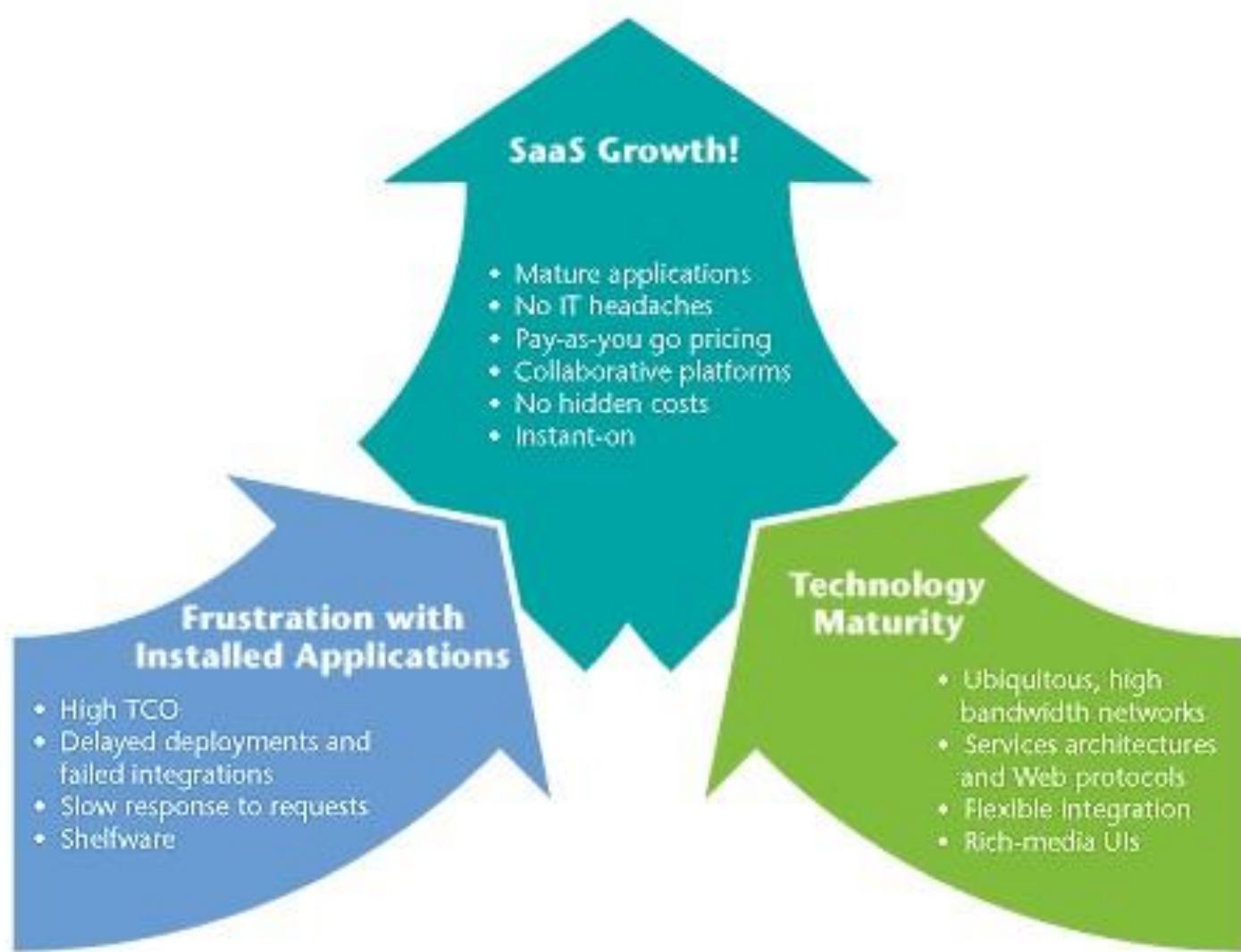


圖7-1

– 如圖7-3所示，這種看法直接把雲端運算作為一種像網格運算那樣的基礎設施來看待，縮小了雲端運算的範圍。

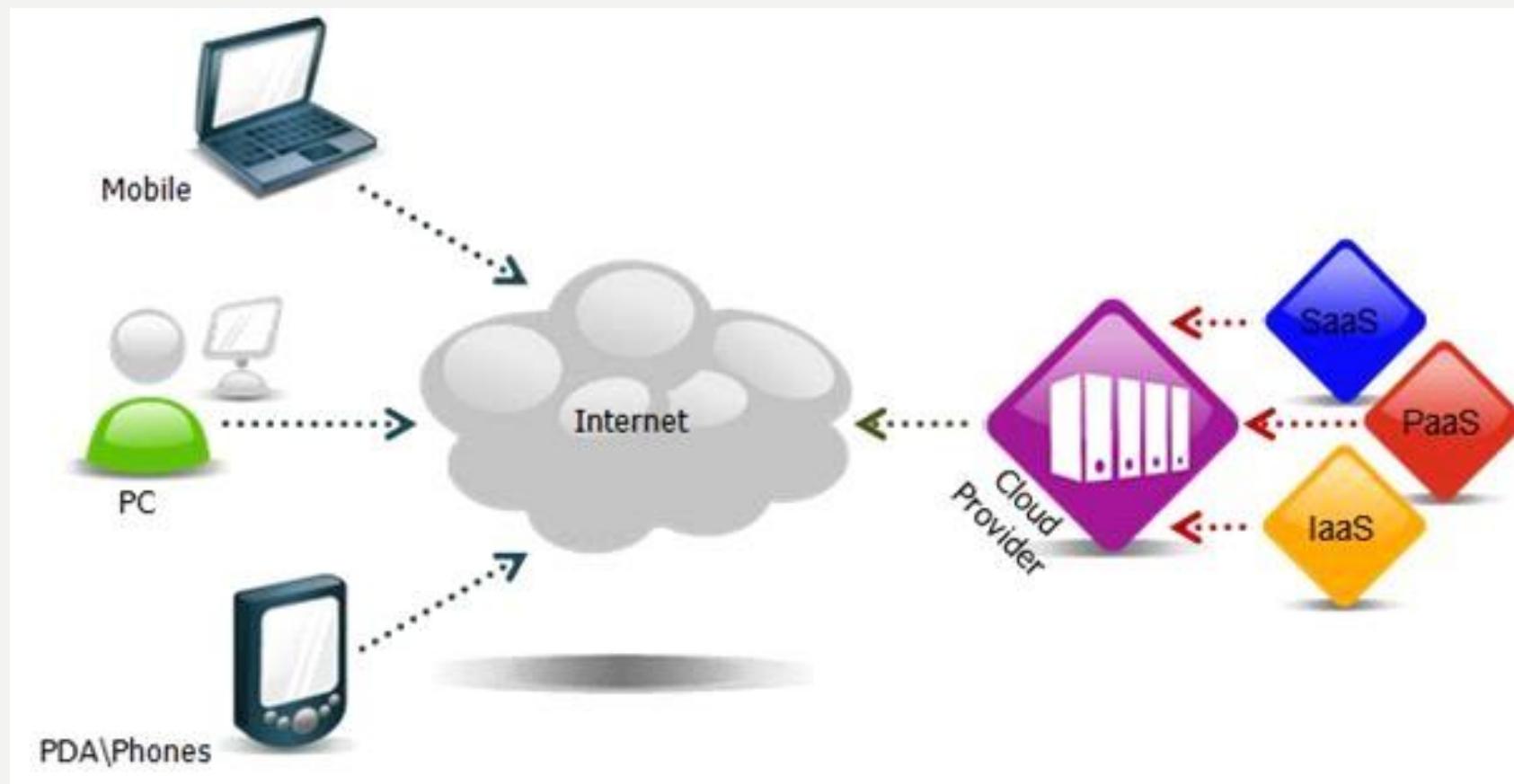


圖7-3

– 也有人把SaaS作為雲端運算的一個分類來考慮（如圖7-4所示）。SaaS後來被NIST劃為雲端運算的一部分，是“事後”的規範劃分，SaaS為一種應用層面的業務模式。這也是為什麼本章要把SaaS單獨列出來介紹的原因。

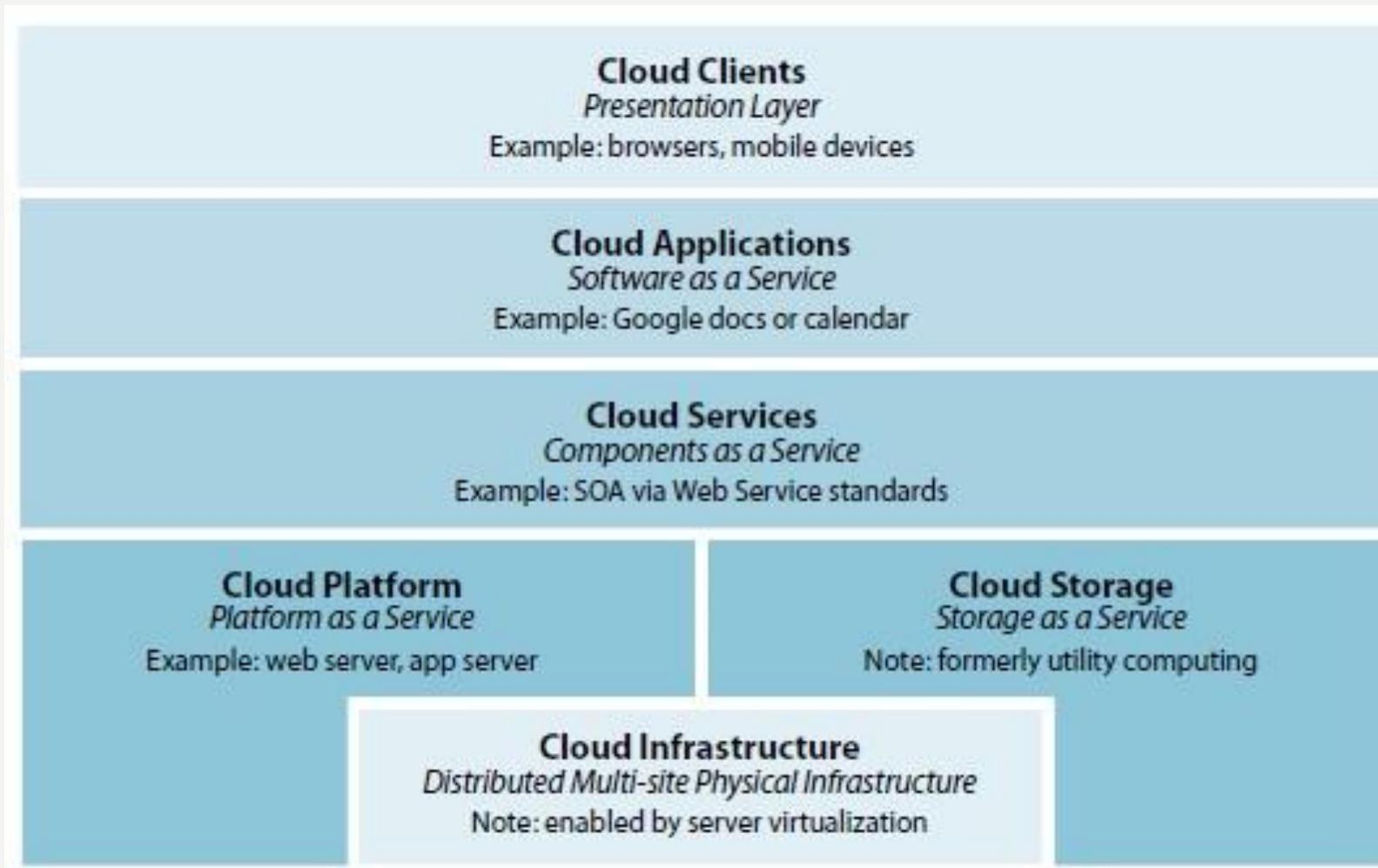


圖7-4

– SaaS是網際網路技術發展和軟體成熟度不斷提高的結果，它是一種透過網路提供軟體的模式。使用者不用再購買軟體，而改用向提供商租用基於Web的軟體，來管理企業經營活動，且無須對軟體進行維護，服務提供商會全權管理和維護軟體，軟體廠商在向客戶提供網際網路應用的同時，也提供軟體的離線操作和本地資料儲存，讓使用者隨時隨地都可以使用其定購的軟體和服務。

- 1995年，網際網路上的瀏覽器的出現改變了整個世界，它也是SaaS的主要致能（Enabler）技術。
- SaaS模式已成為應用軟體市場令人興奮的發展趨勢。在歐美等IT業發達地區，使用者已經開始對SaaS軟體營模式給予了高度的認同。圖7-7展示了客戶感興趣的SaaS服務應用軟體調查結果。

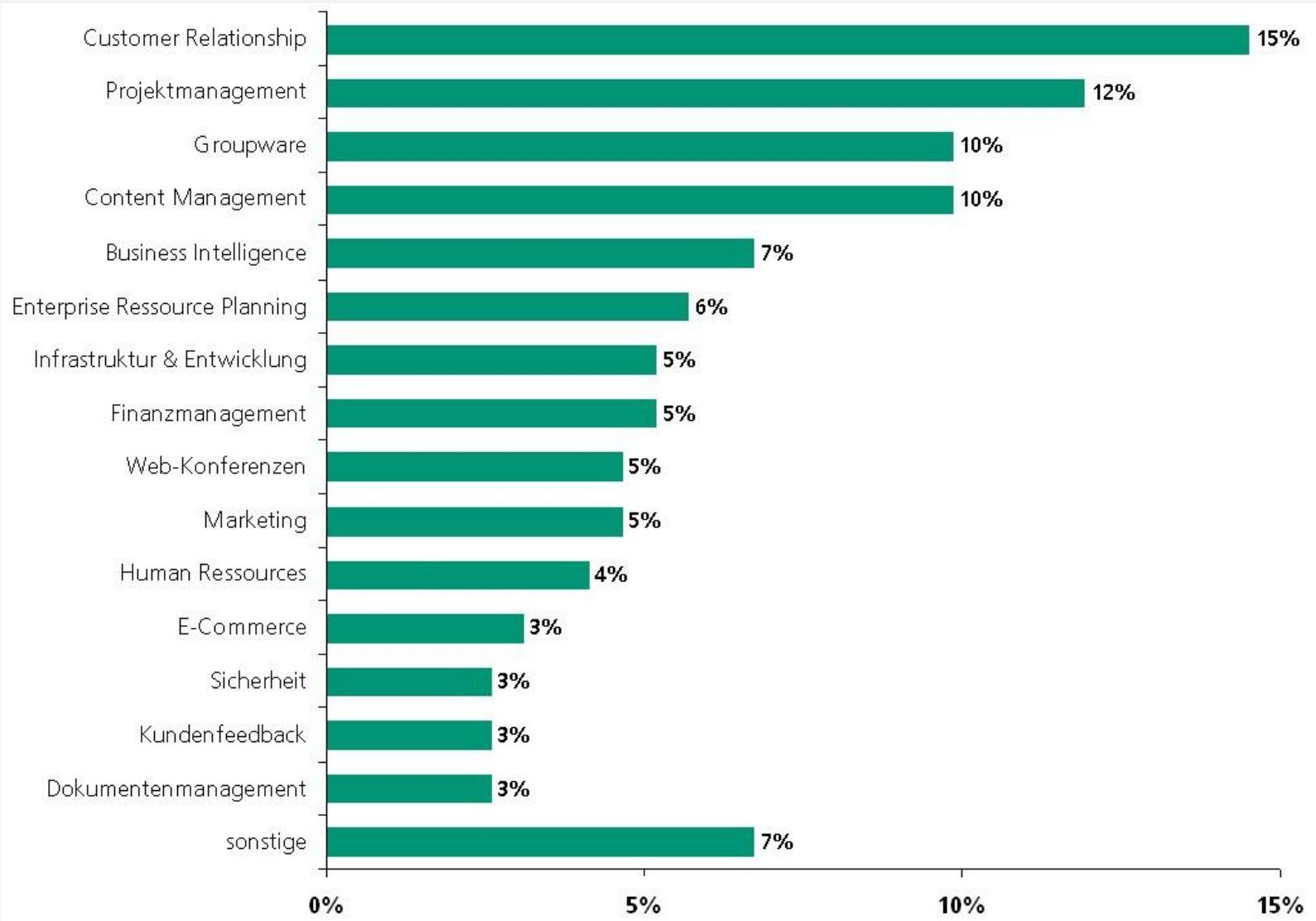


圖7-7

– SaaS使以前那些因為無法承擔軟體許可費用或者是沒有能力配置專業人員的中小（SMB）使用者，都變成了潛在的客戶，圖7-8展示了SMB感興趣的SaaS應用服務。

## SMB - SaaS Application Ranking

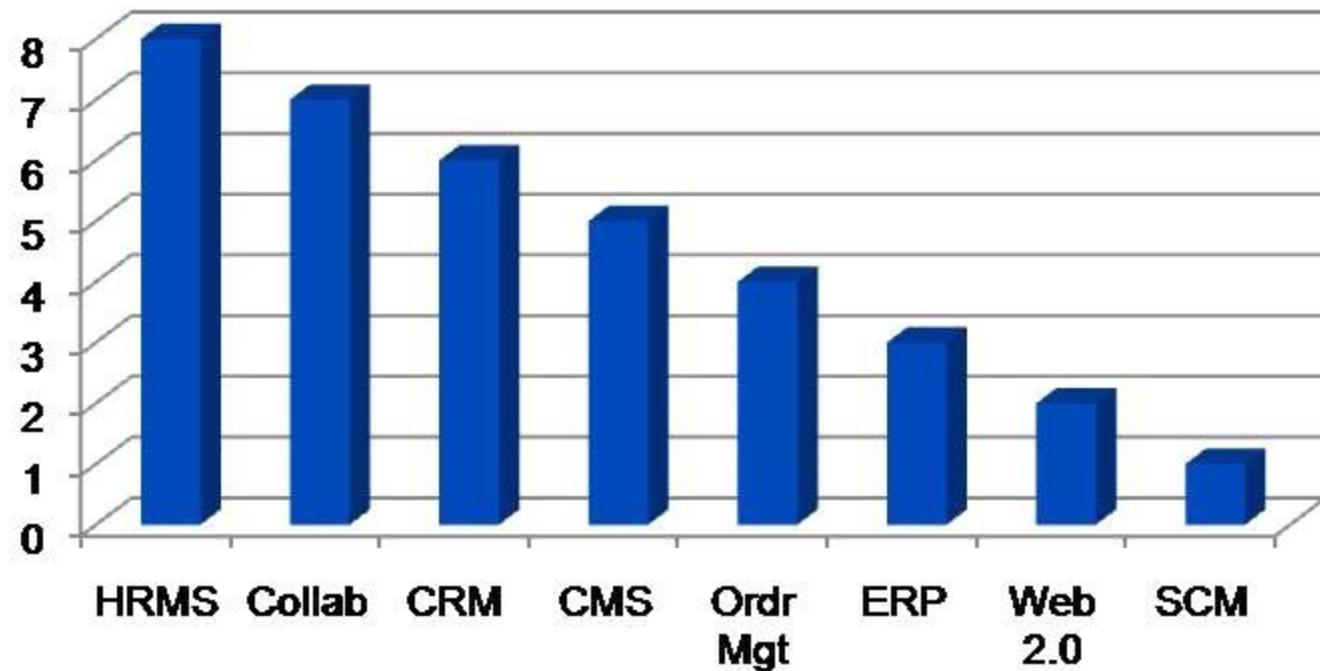


圖7-8

- 7.1.1 Salesforce：軟體終結者
  - 但Salesforce在開展CRM業務的同時提出了“**No Software**（軟體終結者）”的口號，到2001年基於SaaS模式和多租戶（**Multi-Tenants**）技術的門戶推出，它成功地讓人們認同了“**No Software**”口號的價值，Salesforce小雞變鳳凰，成為SaaS的集大成者和最大受益者之一。“終結軟體”的說法雖然有些極端，但指出了一個大趨勢，這就是軟體與網際網路的融合。

- Salesforce可能是SaaS模式最早的使用者，但是SaaS能夠成為IT業界的一大引領IT發展方向的“現象”，Salesforce的確是功不可沒。
- Salesforce的下一個目標是：把平台當做一種服務來提供，即 PaaS ( Platform as a Service )，並推出了force.com門戶，也許我們可以把PaaS和IaaS稱做“**No Hardware**”，即“硬體終結者”。

- 7.1.2 Multi-Tenancy 技術與模式
  - 多租戶技術（Multi-Tenancy或Multi-Tenants）或稱多重租賃技術，是一種用同一個程式實例（Instance）實現對多客戶或多客戶群的服務。最簡單的多租戶技術實作的系統就是電子郵件系統。多租戶技術與一般常用的多實例軟體系統不同，在一般軟體系統中，一個實例只服務一個客戶或客戶群，而且一般被安裝和執行在不同的硬體系統中。

同一個軟體對每一個使用者帳號提供單獨的虛擬執行環境，讓每個使用者感覺他們是在用一個屬於自己的單實例的典型軟體系統。

- 20世紀60年代IBM透過在大型機上設立不同帳號的方式租賃分時共用（Time-Sharing）的CPU時間。
- 20世紀90年代的ASP託管模式。
- 2000年前後，很多基於Web的應用自覺不自覺地採用了多租戶技術和模式。
  - 正是多租戶技術的採用，SaaS才突破了原有的ASP模式的瓶頸。

- 多租戶模式被分為以下4種或4個層次，如圖7-13所示：
  - **Level 1**：把不同的單租戶軟體放到一起在同一個託管平台上執行，ASP屬於這一類。
  - **Level 2**：把相同的單租戶軟體的多個實例分別執行服務不同的使用者，大型機的多租戶服務屬於這一類。

- Level 3：多租戶軟體，一個執行實例，服務多個使用者，郵件系統屬於這一類。
- Level 4：一個多租戶軟體，可自助個性化定制（Self Customizable），可伸縮（Scalable）的多個執行實例，服務大批量使用者。

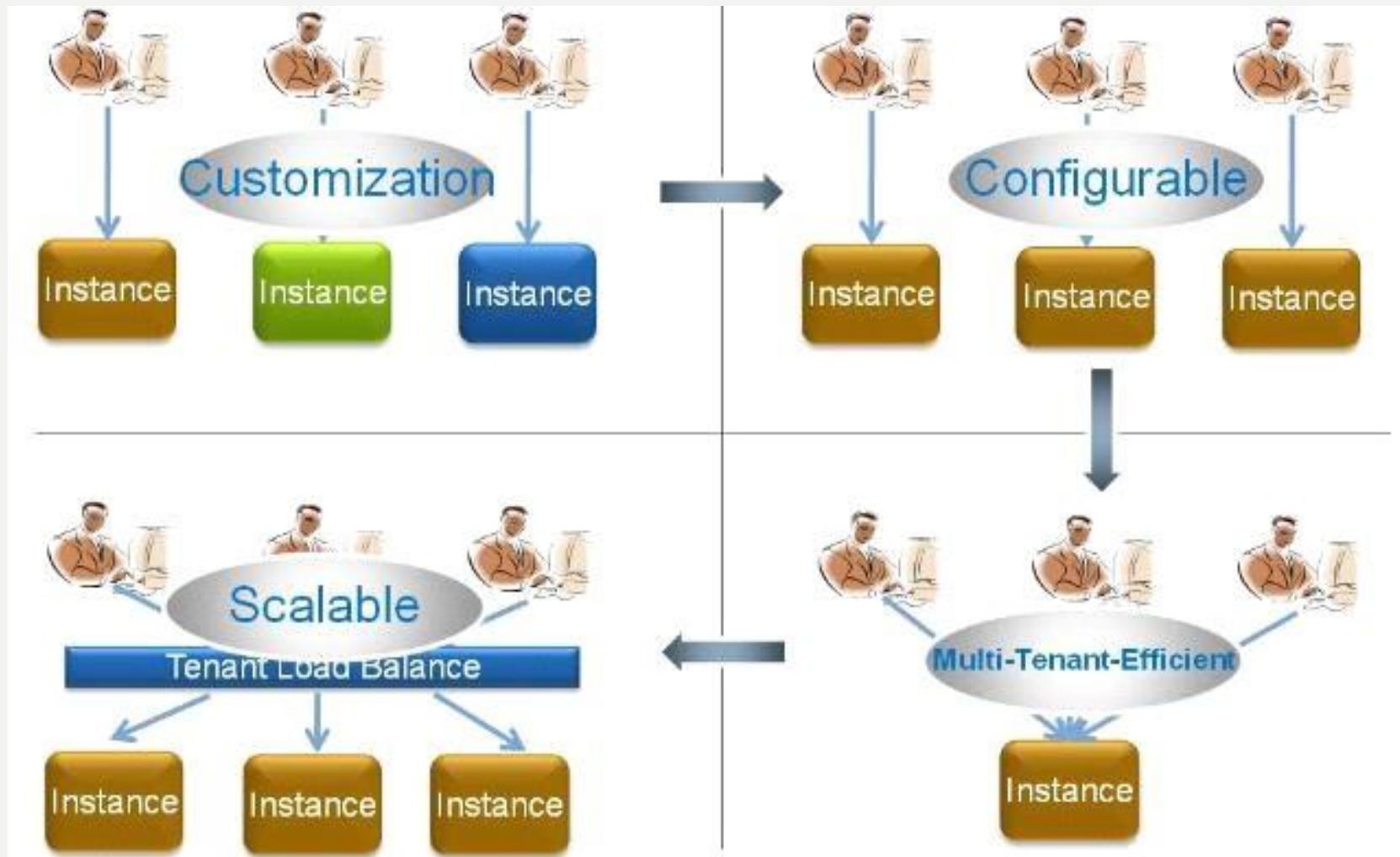


圖7-13

—軟體應用（以下簡稱多租戶應用）和單租戶應用的詳細區別，如圖7-14所示。

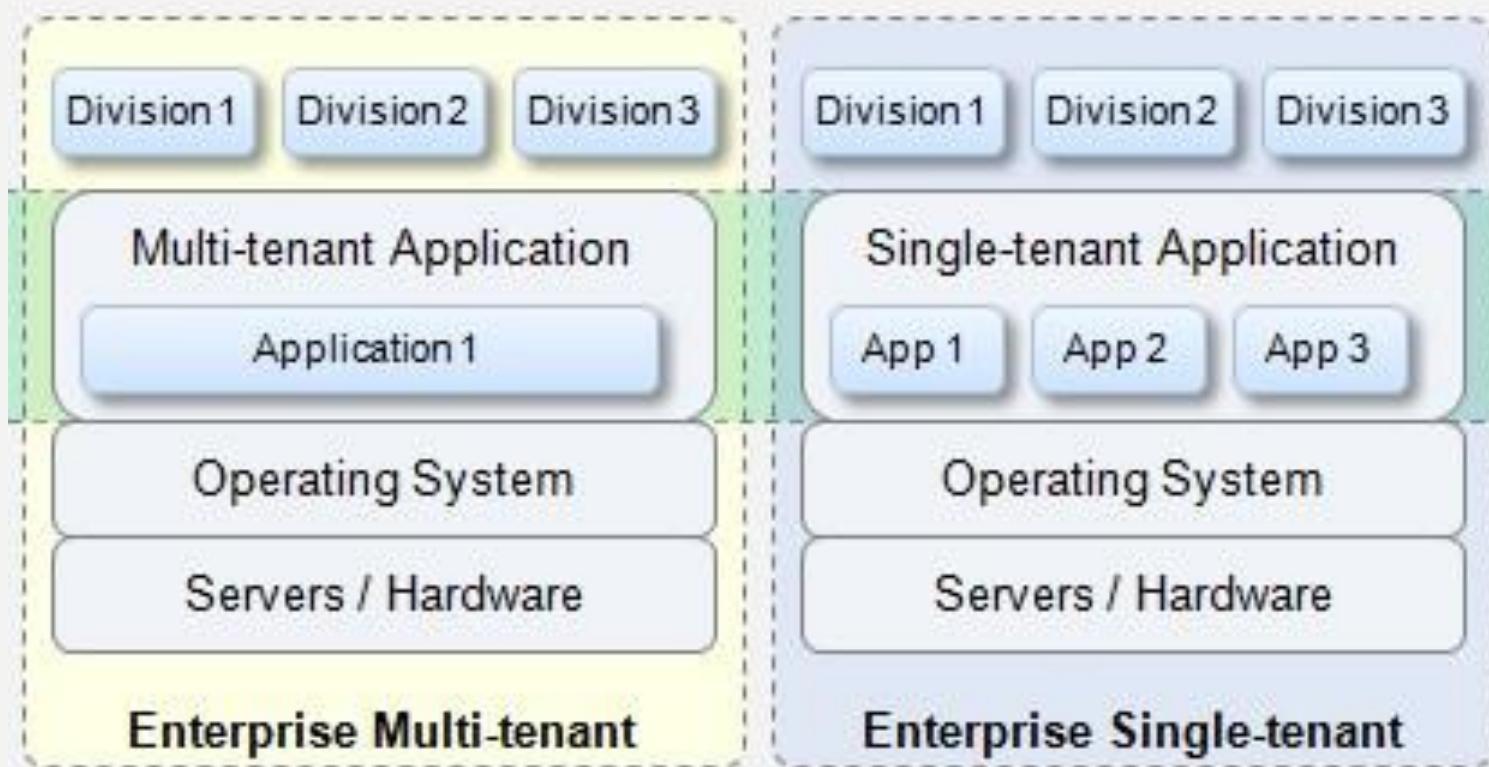


圖7-14

- 多租戶應用因為資料儲存和資料庫使用方式的不同，還分為全多租戶系統和基於“單租戶資料庫”的半多租戶系統等，如圖7-15所示。

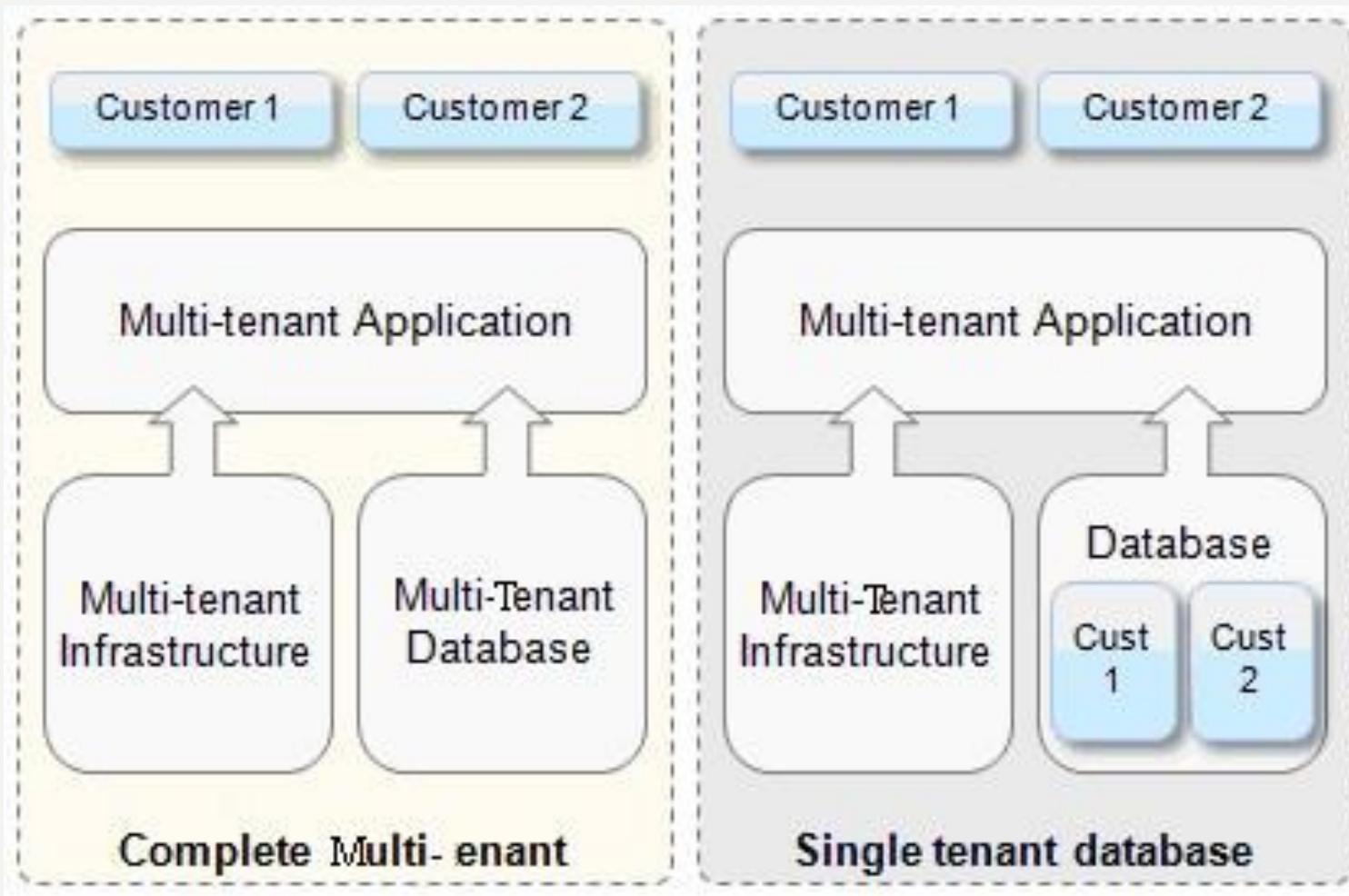


圖7-15

- 按資料庫使用的“共用”和“分離”程度，多租戶應用可分為3種。
  - 獨立的資料庫系統實例（Instance），獨立的資料庫（Schema）。
  - 同一個資料庫系統，獨立的資料庫（Schema）。
  - 同一個資料庫系統，同一個資料庫（Schema），如圖7-16（c）所示。
- 前兩種屬於“單租戶資料庫”式的多租戶應用，只有最後一種才是正真的全多租戶（Complete Multi-Tenancy）應用。

TenantID	CustName	Address		
4	TenantID		ProductID	ProductName
1	4	TenantID	Shipment	Date
6	1	4711	324965	2006-02-21
4	6	132	115468	2006-04-08
	4	680	654109	2006-03-27
		4711	324956	2006-02-23

圖7-16(c)

—顯而易見，全多租戶應用有如下一些優點：

- 減少硬體資源使用量，降低成本。
- 由於資料集中，便於統計、報表、資料探勘等。
- 簡化了軟體版本管理和發佈。
- 全多租戶（Shared Approach）應用的缺點是對軟體設計和開發的要求較高，技術難度較大，需花費更多的研發費用，但從長遠來看會節省總體成本。

- 有人對ERP系統SaaS化前後的使用成本進行了比較或估算。
  - 如果把ERP軟體以（不加修改的原單租戶模式）SaaS方式使用，使用成本可降低到原來的 $1/10$ 。
  - 如果把ERP軟體改寫成支援多租戶的系統，使用成本還可降低到單租戶模式的 $1/5$ 。
- 從節能降耗的角度來說，多租戶技術的普及與推廣無疑會為綠色計算做出巨大的貢獻。

– 多租戶模式的含義和上面介紹的SaaS的多租戶在理念和技術上都不一樣。IaaS的多租戶模式如圖7-20所示，一個（單機）虛擬化的私有雲端或公有雲端，可以按照SLA的要求把虛擬主機池中的虛擬主機劃分組合後分配給多個使用者，實現多租戶運作。

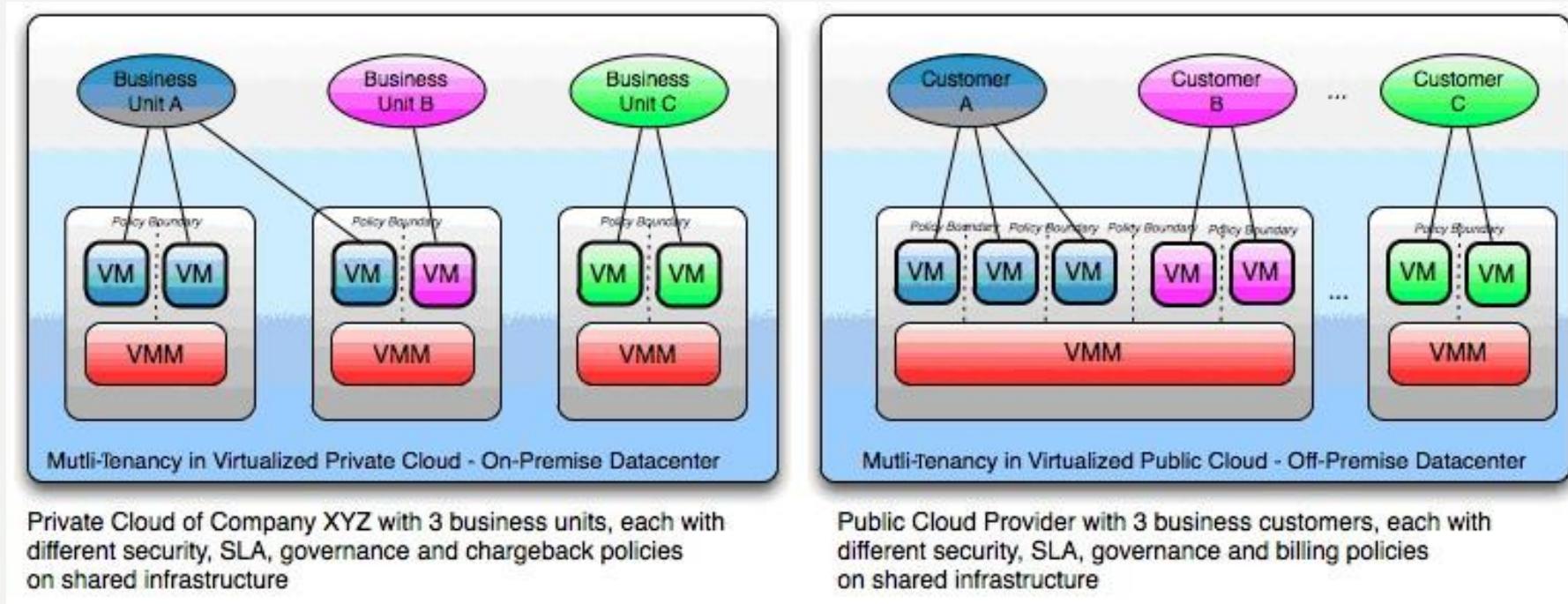


圖7-20

## 7.2 SAAS與套裝軟體

– 從使用者的角度來看，ASP是自己開發軟體，放到ASP那裏執行；SaaS是自己不開發，直接從SaaS服務商那裏開一個帳號，使用軟體；PaaS（Platform as a Service）是到PaaS服務商提供的平台上開發軟體，供自己或別人使用，如圖7-21所示。

# ASP > SaaS > PaaS

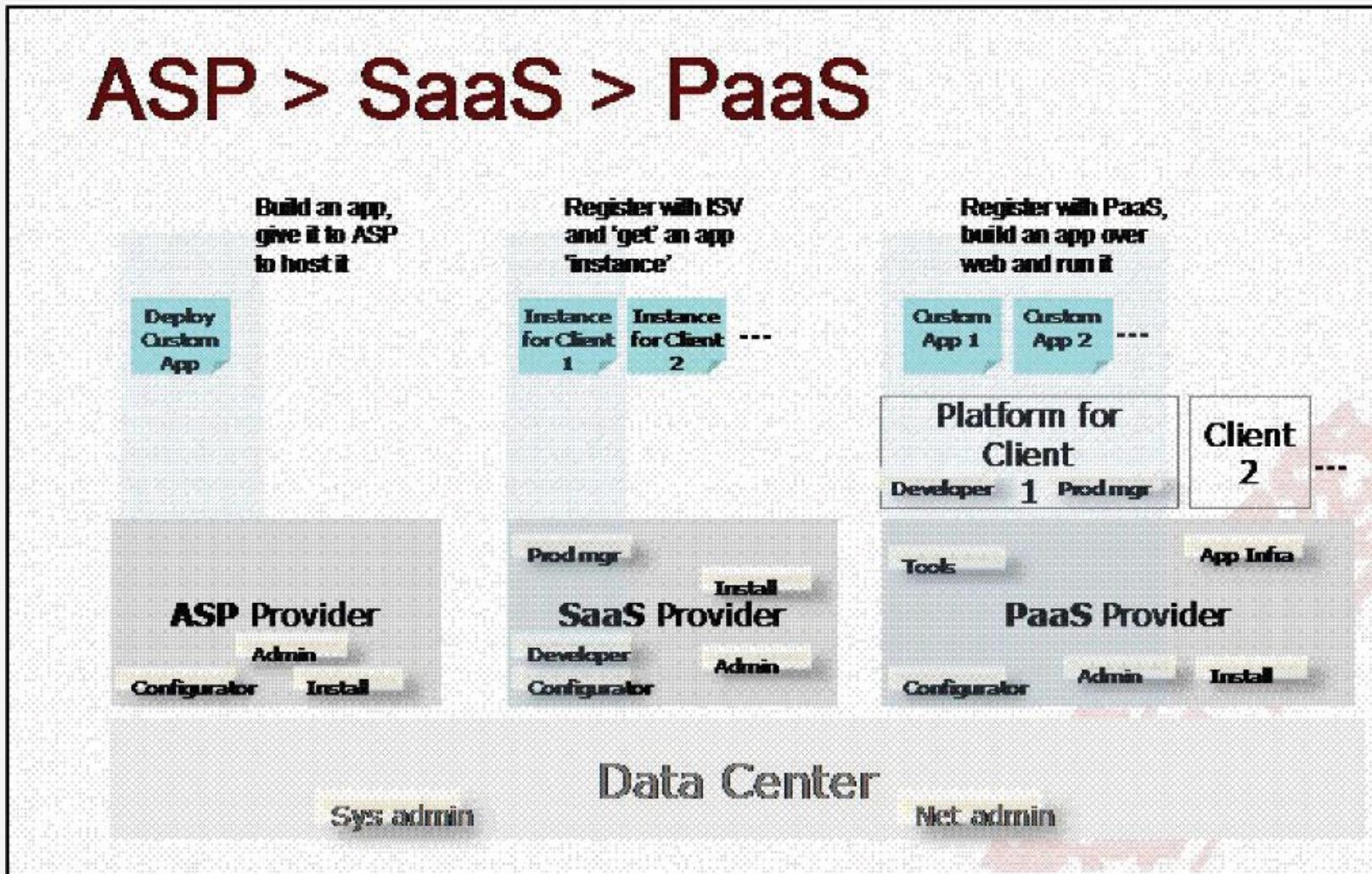


圖7-21

- 理論上說，所有傳統的套裝軟體（ Packaged Application ）都可以被改造成 SaaS 方式來使用，這主要還得感謝網際網路 /Web 技術的發展。
- 不過，出於安全性、使用者習慣等因素的考慮，有一些應用更適合 SaaS ，如 CRM 、電子郵件系統、網路銀行和支付等，如圖 7-23 所示在這些案例中，有許多企業在當時並沒有意識到他們是在做 SaaS 。



圖7-23

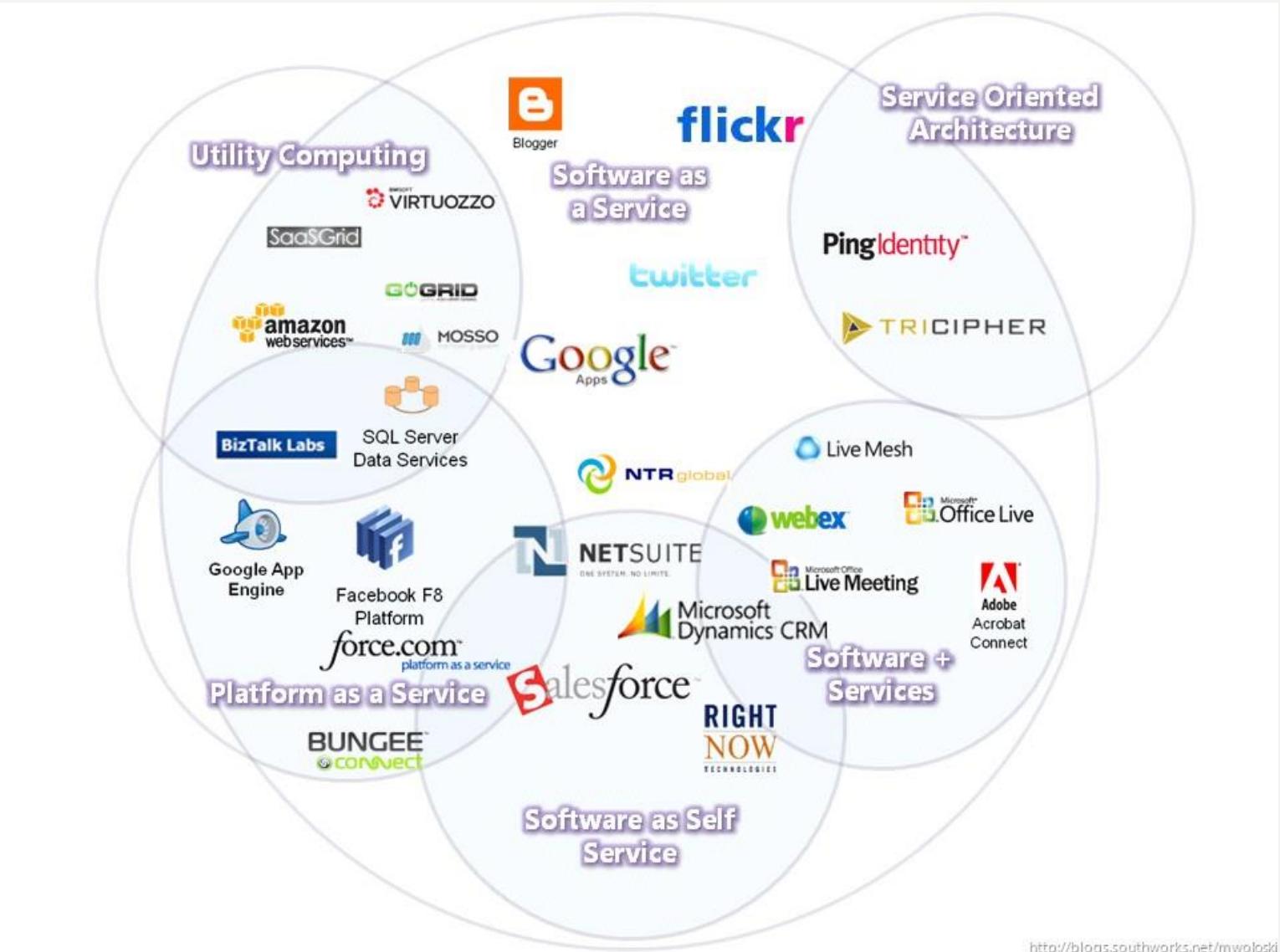
– 經過近20年的發展，把越來越多的套裝軟體變成SaaS，這就是SaaS發展的“長尾理論”：即隨著時間的推移，實作多租戶模式的技術會越來越成熟，越來越多的套裝軟體會被改變成SaaS模式。

- 7.2.1 Google 大戰微軟

- 最關鍵的是Google Docs與Microsoft Office的競爭，這是關係到未來整個產業發展和各自企業生死存亡的焦點。
- 最不希望看到SaaS大行其道的，那一定是非微軟莫屬。
- Google和微軟的博弈，從某種意義上說也是代表了SaaS與套裝軟體的競爭。是SaaS模式在短時間內難以改變的，這也是Google在短期內難以撼動微軟的原因。

– 微軟也在做SaaS（如Office Live Workspace等）和雲端運算（如Windows Azure等），不過所有這些都是微軟“被動回應”的結果。

- 7.2.2 國內外 SaaS 發展現況
  - SaaS企業及它們提供的SaaS、PaaS等服務，最值得關注和借鑒學習的SaaS業務應該還是Google（ Apps和GAE ）和 Salesforce（ salesforce.com和force.com ）。



<http://blogs.southworks.net/mwołoski>

圖7-27

– 發現在中國SaaS採用者中 “安全和合規（Compliance）” 應用的需求最高，其次是網路會議/協同、客戶關係管理和電子郵件等，如圖7-29所示。另外，日本的調查顯示SRM（Supplier Relationship Management）是SaaS模式的首選，如圖7-30所示。

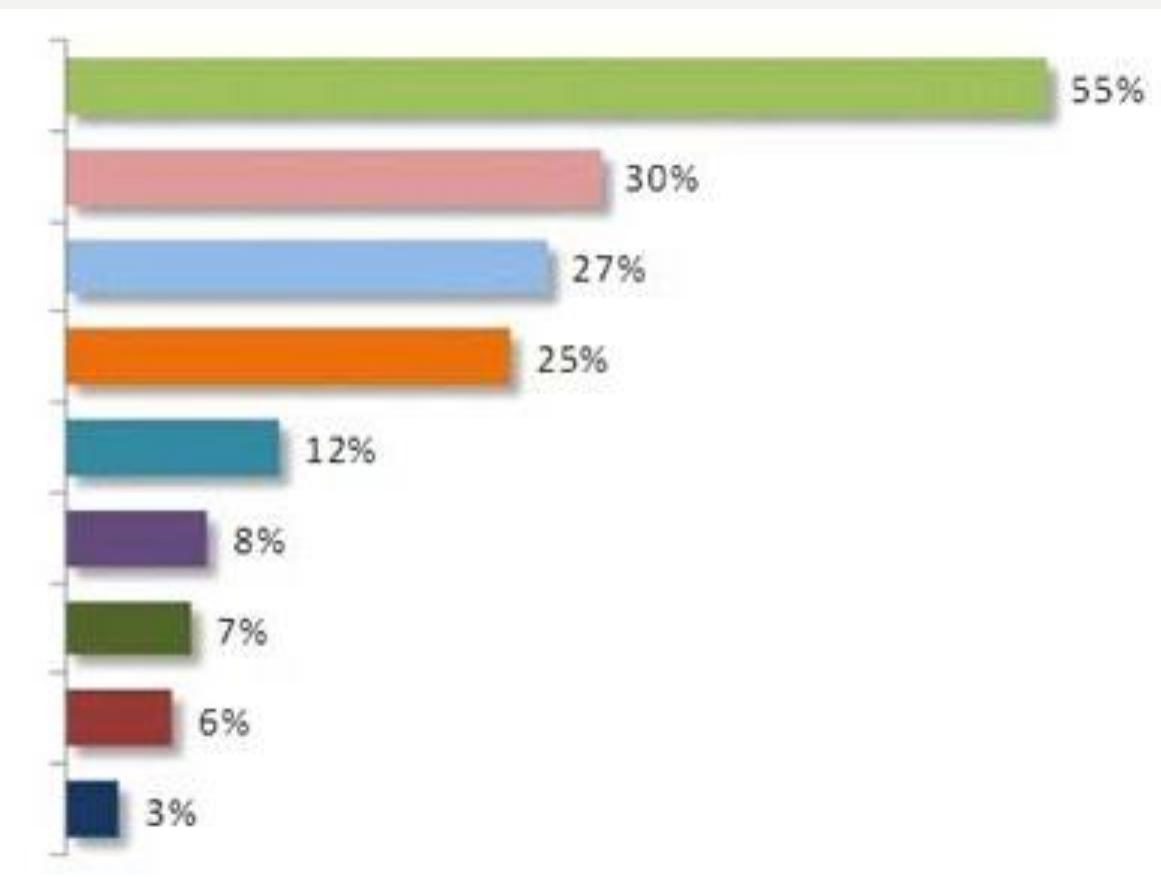
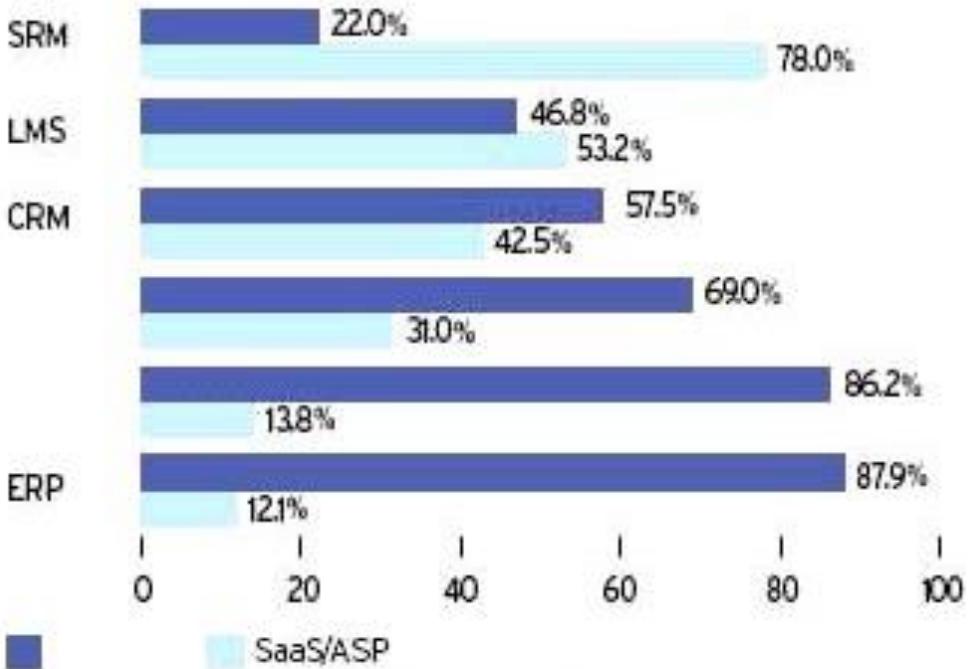


圖7-29



SRM ( Supplier Relationship Management,  
LMS ( Learning Management System,  
CRM( Customer Relationship Management,  
( GroupWare,  
ERP ( Enterprise Resource Planning,

圖7-30

—報告提出：在即將到來的電子商務2.0時代，企業經營管理將更加注意企業內外部的全程管理及上、下游的業務協同，更多高端的企業開始轉向基於SaaS技術的供應鏈電子商務應用。

- 7.2.3 SaaS 與 SPI
  - Salesforce在SaaS模式獲得成功後很快就不再滿足於提供CRM這一項行業業務，於是提出了PaaS ( Platform as a Service )，向使用者提供一個中間層執行平台。
  - 對於一些IT大廠來說，做SaaS或PaaS規模都太小，不過癮，於是以IBM為首又推出了IaaS ( Infrastructure as a Service )、HaaS ( Hardware as a Service ) 等概念，如圖7-34所示，從SaaS到PaaS再到IaaS，使用者要管的事變得越來越多。

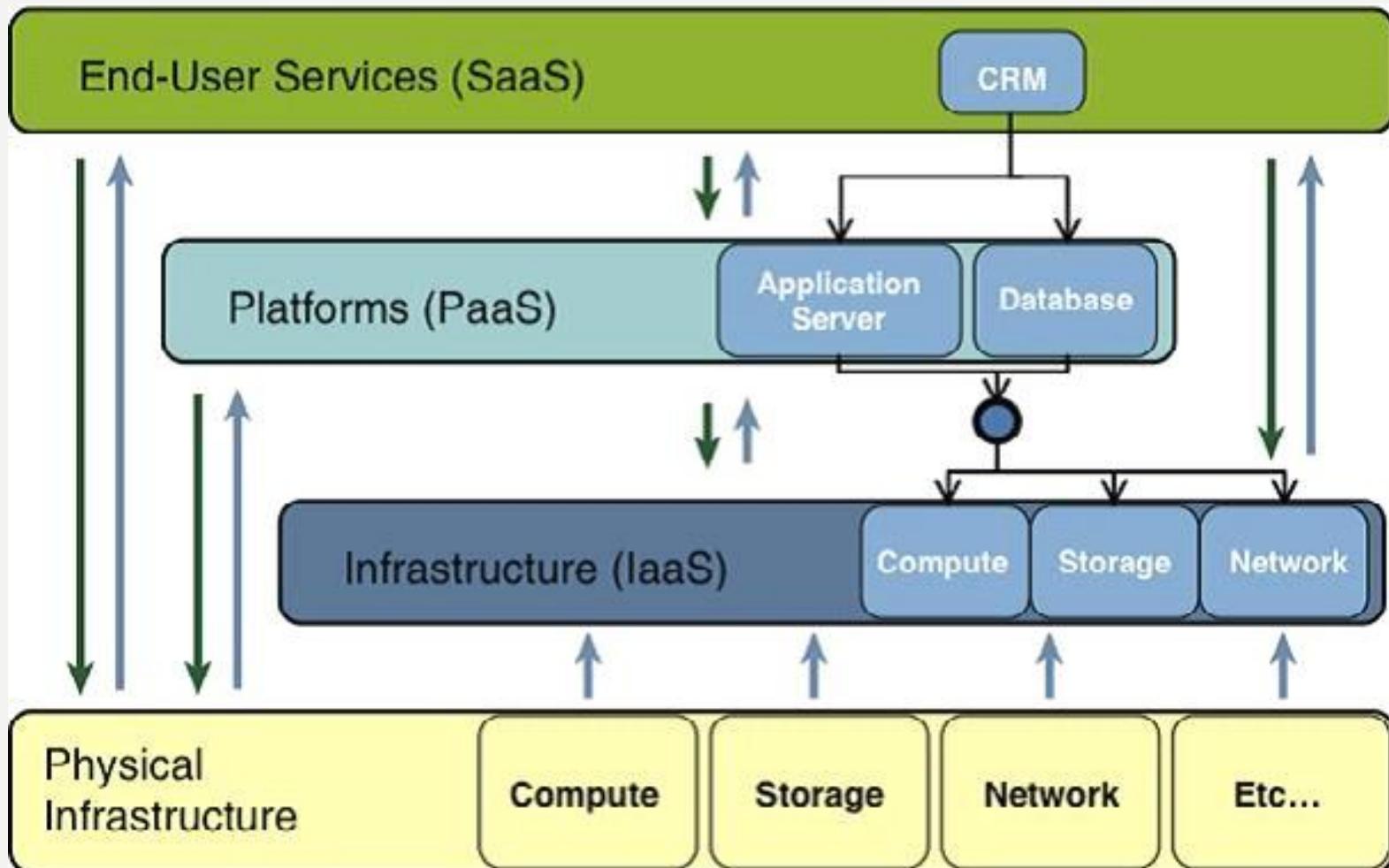


圖7-34

- SaaS模式下的使用者“什麼都不管”，現在從SaaS發展到XaaS (*Everything as a Service*，一切皆服務)，使用者又變成“什麼都管了”，不過這應該是在不同層次上的重複，許多角色的定位與原來已經不一樣了。
- IT服務越來越走向底層，直至雲端運算，也就是使用者不再需要擁有任何設施，從“軟體的終結”走向“硬體的終結”。  
SPI是SaaS、PaaS和IaaS的首字母縮寫，SPI被業界許多人列為雲端運算的三大支柱（這是第8章要詳細介紹的內容），都屬於雲端服務的範圍（如圖7-35所示，範圍及市場規模預測）。

Cloud Service	Service Attributes/Description	Cloud Provider Examples
Software as a Service (SaaS)	<ul style="list-style-type: none"> <li>• Consumer Websites</li> <li>• Mashups</li> <li>• Multi-Tenant Business Focused Web Applications</li> <li>• Collaboration, Email, Office Productivity</li> <li>• CRM, SFA, ERP etc</li> <li>• Management Applications/Interfaces</li> <li>• APIs for Service Integrations</li> </ul>	<ul style="list-style-type: none"> <li>• Flickr.com</li> <li>• Myspace.com</li> <li>• Google Apps</li> <li>• SalesForce.com</li> <li>• Cisco WebEx</li> <li>• PayPal, Amazon FPS, DevPay</li> <li>• Yahoo APIs (Search, Flickr)</li> <li>• Google Apls (Payment, Adsense)</li> </ul>
Platform as a Service (PaaS)	<ul style="list-style-type: none"> <li>• Application Environments – J2EE, RoR, .Net</li> <li>• Mashups of SaaS Services</li> <li>• Configuration Platform and Scripting Engines</li> <li>• Hosted Application Development Environments</li> <li>• Application Infrastructure Capabilities</li> <li>• Database, Data Stores</li> <li>• Message Queues</li> <li>• APIs for PaaS Integrations</li> </ul>	<ul style="list-style-type: none"> <li>• Google App Engine</li> <li>• Microsoft Azure – SQL DS</li> <li>• Amazon Simple DB, S3, SQS</li> <li>• Force.com, VMForce</li> <li>• NoSQL,</li> <li>• Rollbase</li> <li>• Caspio</li> <li>• Hadoop</li> </ul>
Infrastructure as a Service (IaaS)	<ul style="list-style-type: none"> <li>• Virtual Servers</li> <li>• Compute Capacity Enabled by API Based</li> <li>• Provisioning</li> <li>• Logical Disks</li> <li>• Networking</li> <li>• APIs for IaaS Service Integrations</li> </ul>	<ul style="list-style-type: none"> <li>• Amazon EC2</li> <li>• Elastria</li> <li>• Rackspace</li> <li>• Savvis</li> <li>• FlexiScale</li> <li>• RightScale</li> <li>• GoGrid</li> <li>• CloudSwitch</li> <li>• SNIA</li> </ul>

圖7-35

## 7.3 SOA是SaaS和雲端運算的基礎

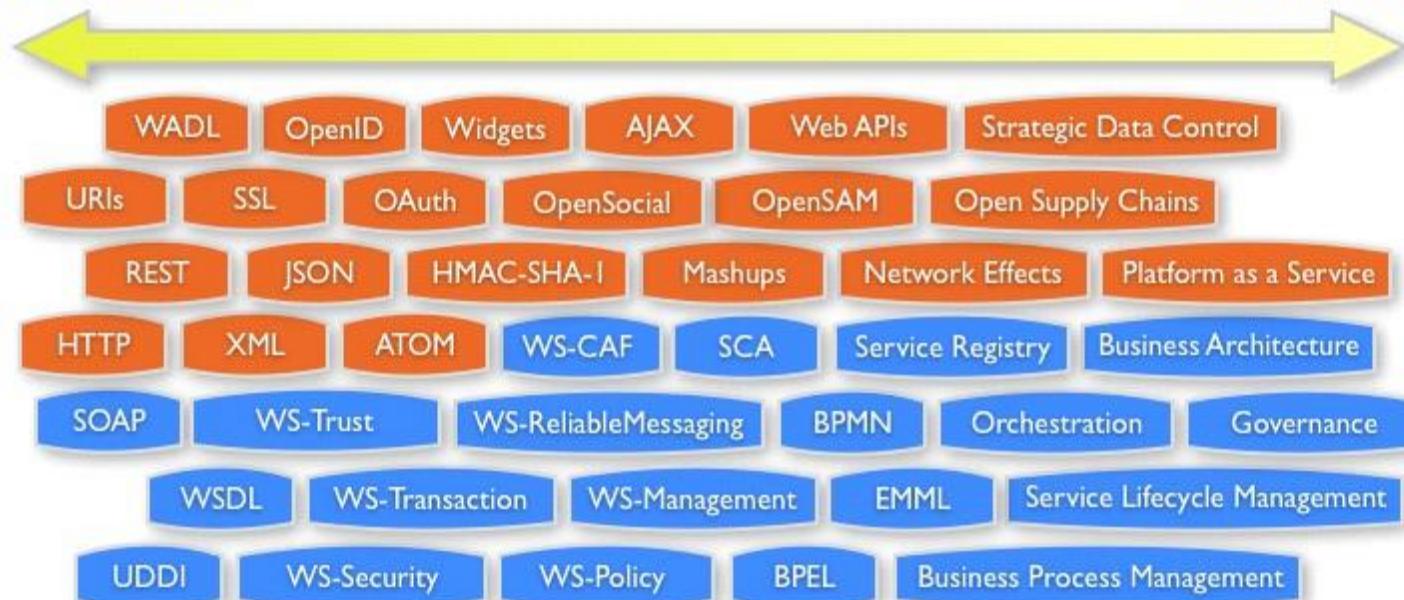
- 有人說SOA和SaaS好比軟體圈內的“大S”和“小S”，它們同根同源，其中的血緣關係就是Web Service這個基因。SOA這個詞出現得較早，其起源和時間難以確定，雲端運算（SaaS 3.0）就是“SaaS的SOA化”，即透過SaaS的模式提供SOA服務。
- SOA--服務導向架構(Service-Oriented Architecture，SOA)

- 7.3.1 SOA 與 EAI
  - SOA主要用於B2B整合，也就是EAI（Enterprise Application Integration），典型工作流程如圖7-36所示，服務提供商用WSDL（Web Services Description Language）描述可提供的服務，並在UDDI服務機構註冊。UDDI在2000年由國際標準組織OASIS提出來後並沒有被業界廣泛接受，到2006年，IBM、SAP等一些提供UDDI服務的設施相繼關閉。
  - **UDDI**是統一描述、發現和集成（Universal Description, Discovery, and Integration）的縮寫。它是一個基於[XML](#)的跨平台的描述規範，可以使世界範圍內的企業在[網際網路上](#)發布自己所提供的服務。

# The Spectrum of Service-Oriented Approaches

Technical

Business



Guided by the Web and  
Online Businesses

**WOA**  
Web-Oriented Architecture



**SOA**  
Service-Oriented Architecture

Driven by Standards  
Bodies and Enterprises

ebiz From <http://www.ebizq.net/blogs/enterprise> by Dion Hinchcliffe

圖7-40

- SOA強調透過Web Service介面（ API ）定義、註冊和發現等方式提供服務呼叫，一般沒有直觀的使用者操作介面，而 SaaS 提供直觀的、可配置、可個性化定制的使用者操作介面。
- 在網格運算系統的支援下，將使 SaaS 走向更加整合化的 B2B 應用，也就是 SaaS 3.0 。

- 7.3.3 SOA 與雲端運算

- SOA是一種標準化的面向服務的IT系統設計規範和軟體系統架構，在一個SOA系統中，每個功能模組或子系統被包裝成一組基於API的服務（主要是Web Services形式），功能模組提供的服務透過互操作實作跨網路、跨業務系統之間的整合和整合，完成對使用者的服務。雲端運算也可以說是SOA、SaaS技術與網格運算技術相融合的產物【71】。

– 圖7-44所示，雲端運算繼承了SOA的許多技術和理念，把SOA從應用層（SaaS和PaaS）擴展到了基礎設施層（IaaS），以Amazon Web Services為核心的IaaS（包括Eucalyptus和Nebula等）就是SOA和網格融合的產物。

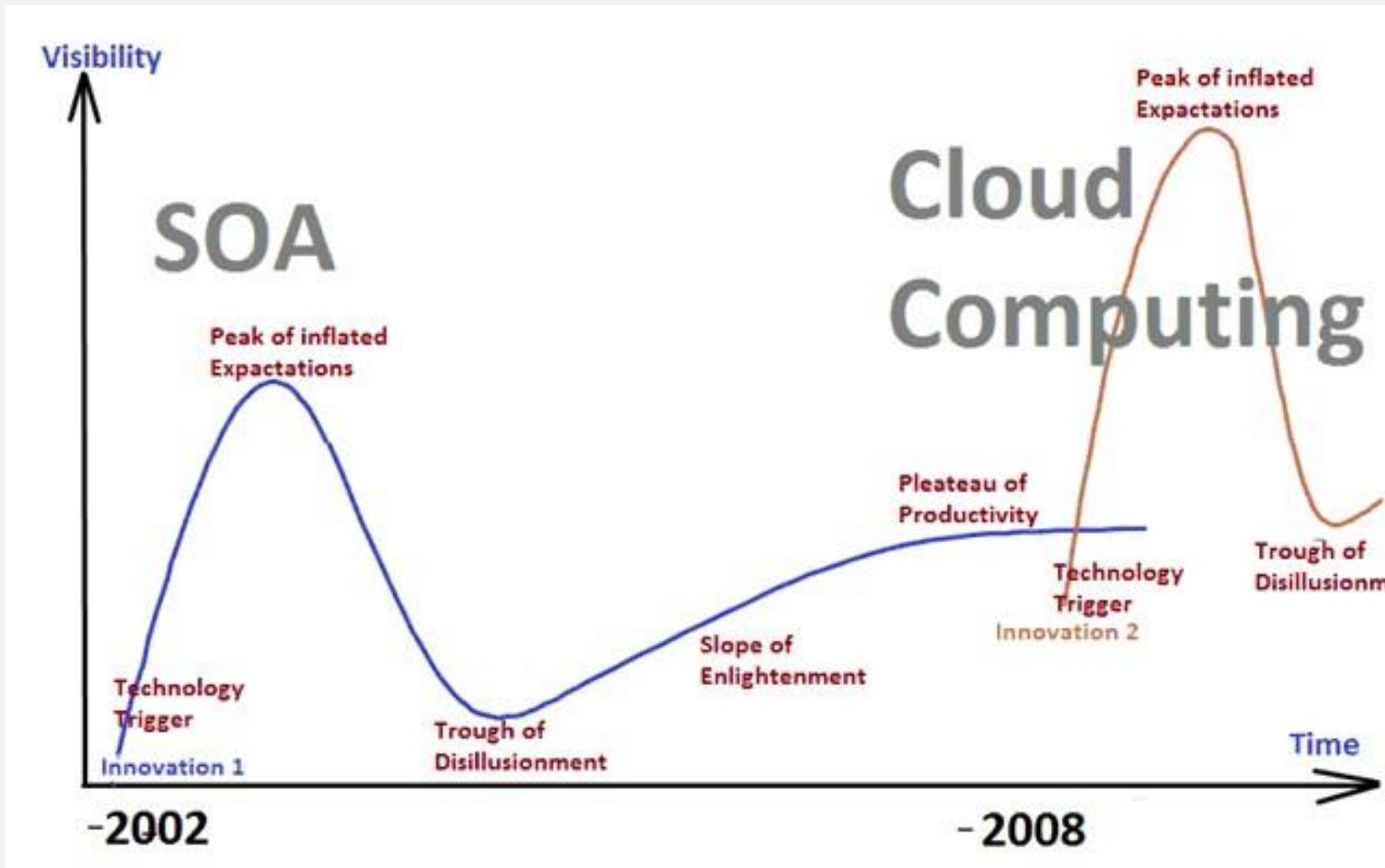


圖7-44

- SOA和雲端運算這兩個理念和技術的區別與共同之處如圖7-45所示。

Cloud Computing	Overlap	SOA via Web Services
<ul style="list-style-type: none"><li>• Software as a Service (SaaS)</li><li>• Utility Computing</li><li>• Terabytes on Demand</li><li>• Data Distributed in a Cloud</li><li>• Platform as a Service</li><li>• Standards Evolving for Different Layers of the Stack</li></ul>	<ul style="list-style-type: none"><li>• Application Layer Components/Services</li><li>• Network Dependence</li><li>• Cloud/IP Wide Area Network (WAN)-supported Service Invocations</li><li>• Leveraging Distributed Software Assets</li><li>• Producer/Consumer Model</li></ul>	<ul style="list-style-type: none"><li>• System of Systems Integration Focus</li><li>• Driving Consistency of Integration</li><li>• Enterprise Application Integration (EAI)</li><li>• Reasonably Mature Implementing Standards (REST, SOAP, WSDL, UDDI, etc.)</li></ul>

圖7-45

## 7.4 SAAS 3.0：雲端服務

– 目前還出現了基於知識處理的語義網（ Semanitc Web ）的 Web 3.0和基於無所不在的設備智慧化連接的Web 4.0的提法，如圖7-47所示。

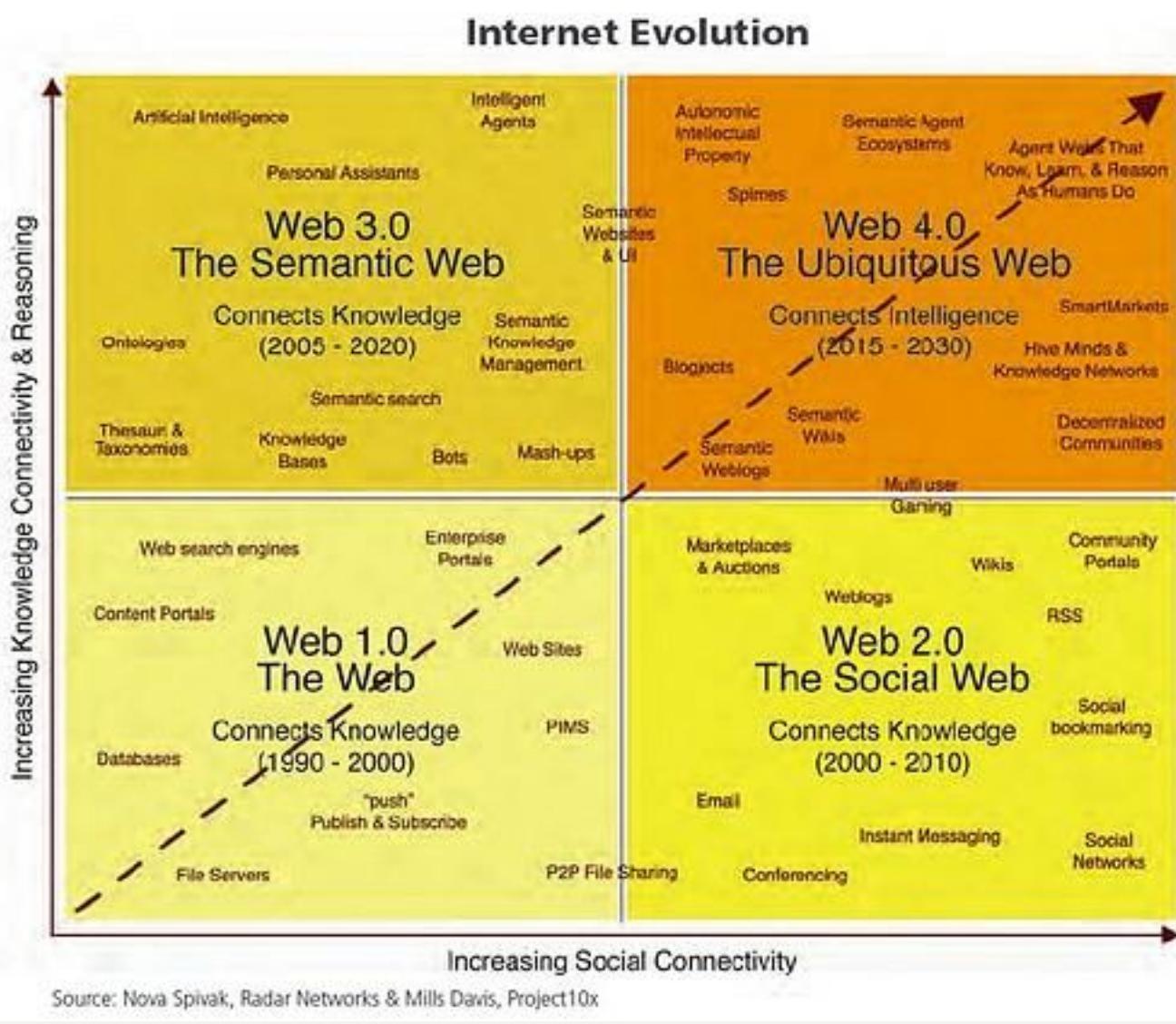
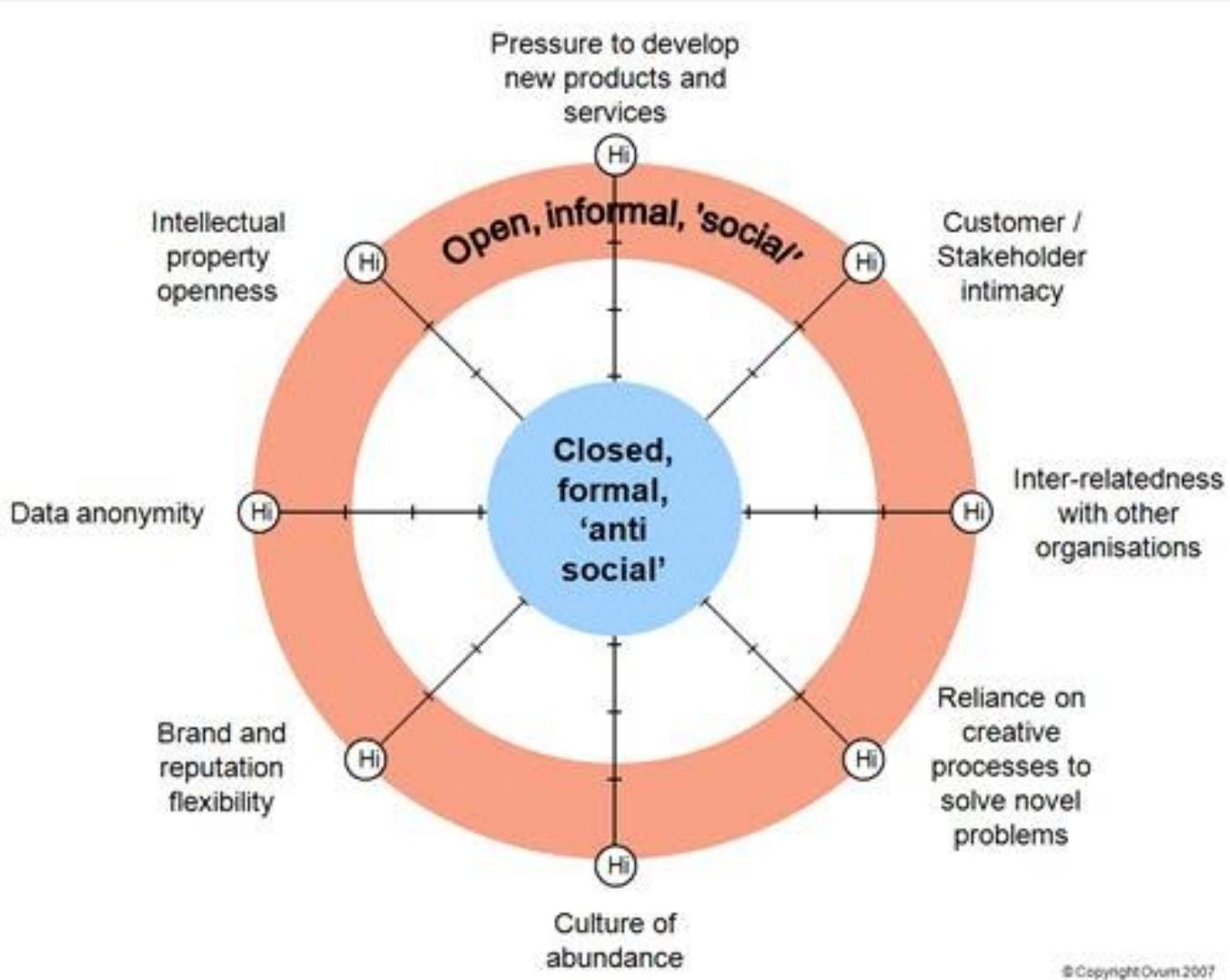


圖7-47

– Enterprise 2.0是在Web 2.0的基礎上，把原來封閉的企業環境擴展到防火牆以外，結合SOA（Service Oriented Architecture），提供開放式、可發現式（Discoverable）的服務，如圖7-48所示。



© Copyright Ovum 2007

圖7-48

– 如果說SaaS 1.0是ASP，那麼SaaS 2.0就是指基於Multi-Tenancy技術的“純SaaS”，如圖7-49所示。

## 1995: SaaS 1.0, Hosted/ASP-based Applications

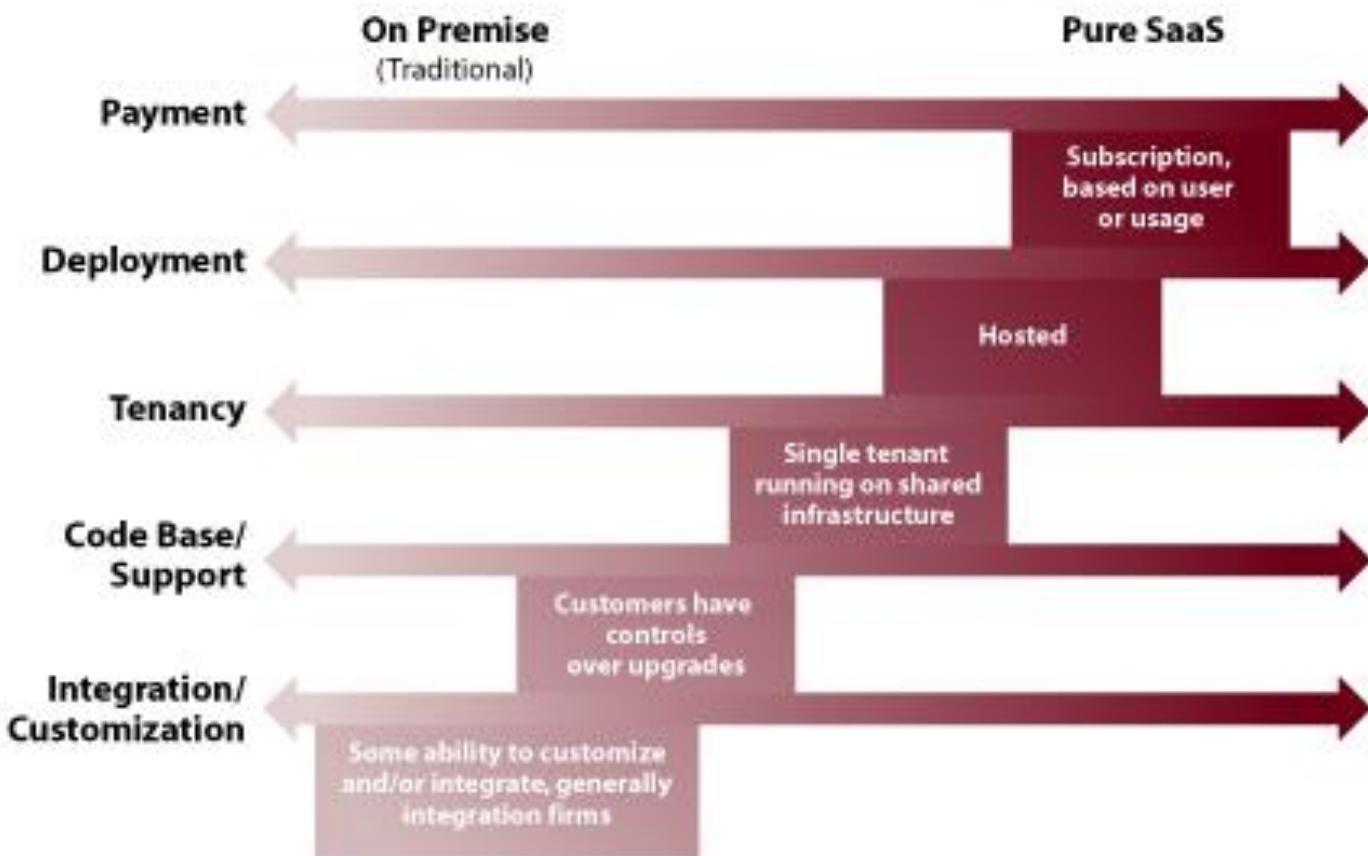


圖7-49

– SaaS 3.0則是SaaS 1.0和SaaS 2.0模式的整合和混合（ Hybrid ）<sup>1</sup>，如圖7-50所示。

## 2005: SaaS 3.0, Hybrid SaaS Applications & Ecosystem

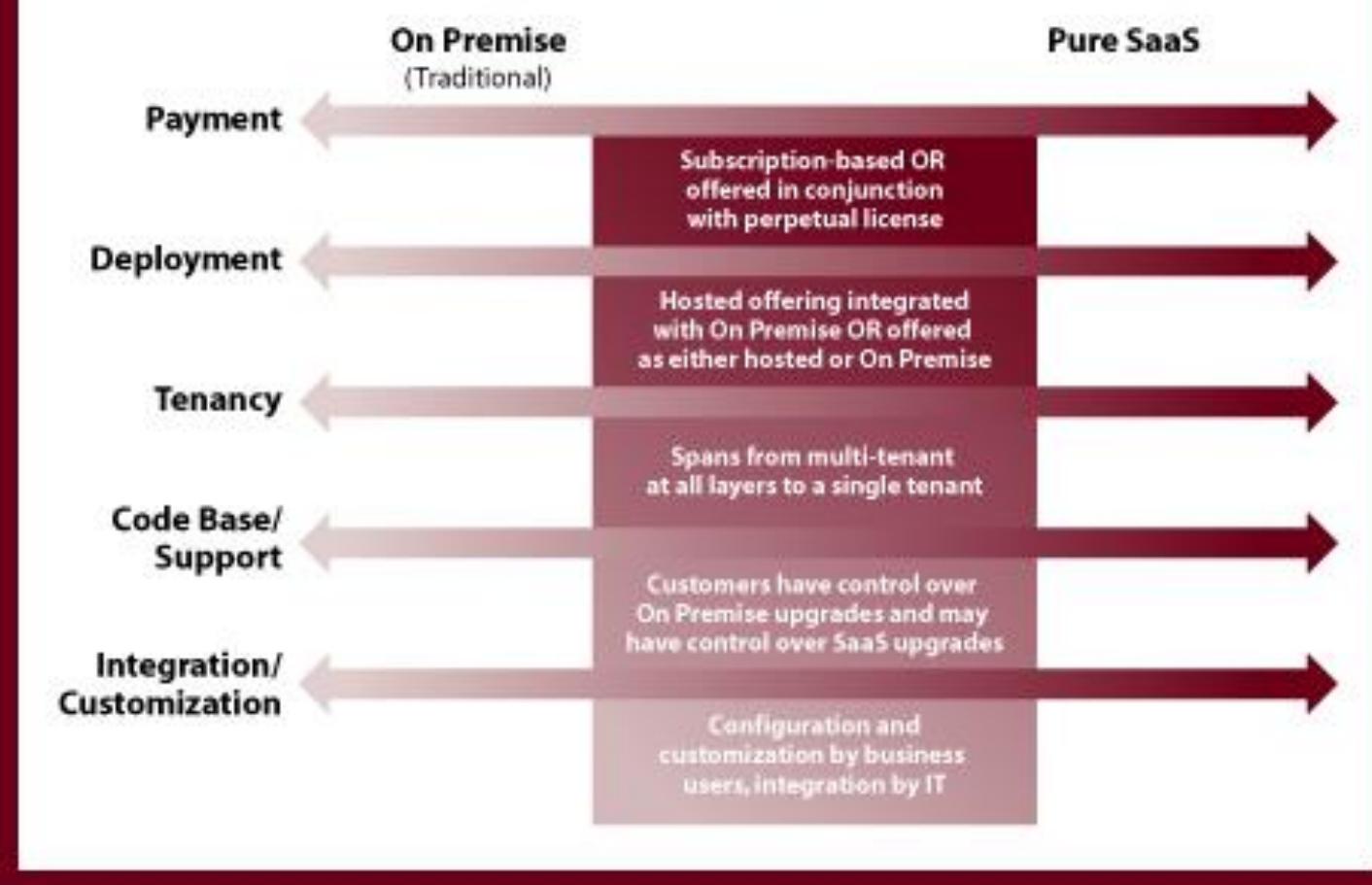


圖7-50

– SaaS 3.0<sup>【49】</sup>也可以說是Enterprise 2.0的補充，強調的是基於SOA技術，將許多託管的SaaS應用與企業內部後臺業務的密切整合，與雲端運算理念深度融合，如圖7-51所示。

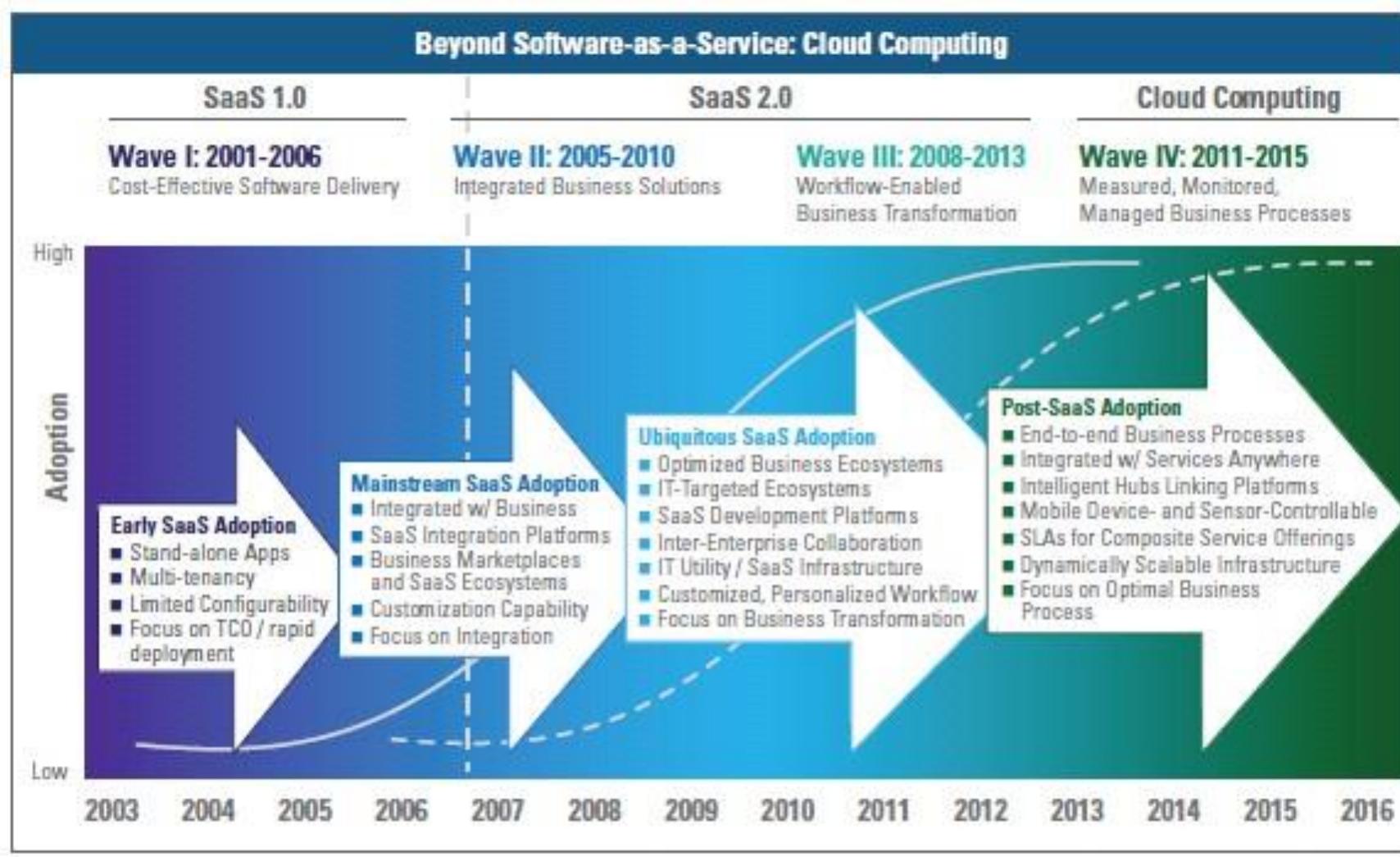


Figure 1 — Saugatuck SaaS Wave Evolution Model

Source: Saugatuck Technology Inc.

- 7.4.1 行動雲端服務
  - Apple、Google和微軟形成了在“雲端”上的全面對抗，如圖7-53所示，各有優勢，難分難解，但誰要想贏得整個戰爭（War）的全面勝利卻很難說，業界拭目以待。

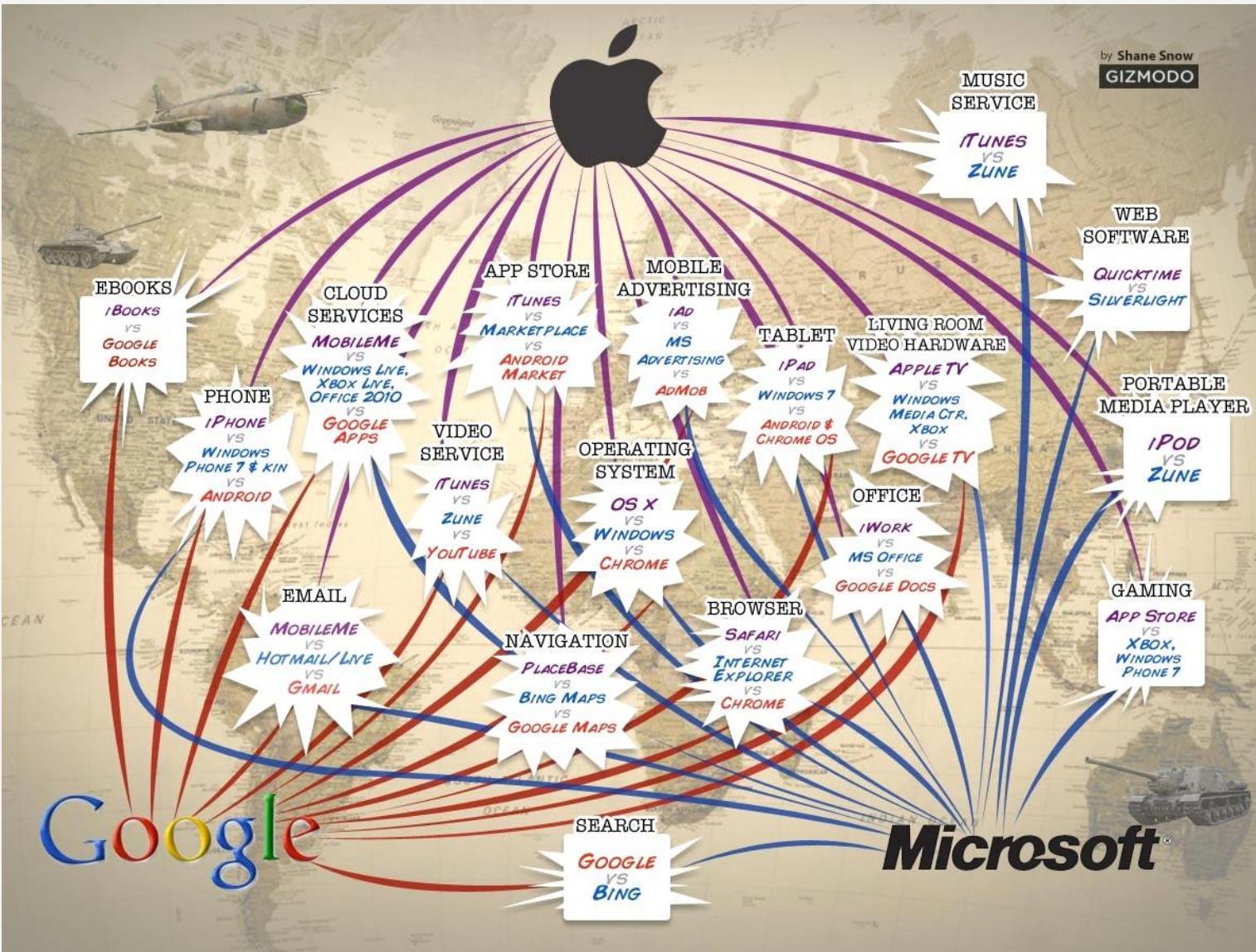


圖7-53

– 使用者端必將使行動終端成為主流，這是一個人人必爭的市場，行動雲端服務（Mobile Cloud Services）或MaaS（Mobility as a Service）也將成為主流。如圖7-54所示展示了Apple和Google在行動雲端服務方面的競爭現狀對比，目前Apple略佔優勢。

# iPhone 4 vs Android: desktop & cloud services

2010 smartphones	iOS iPhone 4 First party offerings from Apple	Android smartphones First party offerings from Google
Desktop PC media and data sync	<a href="#">iTunes</a> (Music, TV shows, movies, rentals, audiobooks podcasts, iTunes U, ringtones, photos, mail, contacts, calendars, notes, bookmarks)	<a href="#">None from Google</a> ; DoubleTwist (Music, photos, movies), Missing Sync (Music, videos, podcasts, ringtones, photos, contacts, calendars, bookmarks), manual device mounting and file copy
Local device backup	<a href="#">iTunes</a> automatic backups	<a href="#">None from Google</a> ; third party apps available
Desktop PC media store	Amazon web store (Music), <a href="#">iTunes Store</a> (Music, TV, movies, rentals, audiobooks podcasts, iTunes U), others. Apple TV integration.	<a href="#">None from Google</a> ; DoubleTwist's Amazon store (Music, podcasts) <a href="#">No options for buying or renting TV, movies</a>
Desktop PC software and ebook store	<a href="#">iTunes App Store</a> , <a href="#">iBooks Store</a> for iPhone & iPad (searchable app, ebook listings), Amazon web store, others (searchable ebook listings)	<a href="#">None from Google</a> ; DoubleTwist's Android Market, other app stores (searchable app listings), Amazon web store, others (searchable ebook listings)
Cloud client apps	<a href="#">Mail</a> (new email alias support), <a href="#">Contacts</a> (new CardDAV support), <a href="#">Calendar</a> (new CalDAV support), <a href="#">Notes</a> , <a href="#">iDisk</a> , <a href="#">Gallery</a> , <a href="#">Find My Phone</a>	<a href="#">Gmail</a> , <a href="#">Google Calendar</a> , <a href="#">Contacts</a> (on Android phones "with Google"), or <a href="#">web apps</a> . <a href="#">Gmail lacks email aliases</a> , <a href="#">Android lacks CardDAV, CalDAV</a>
Push messaging services	<a href="#">MobileMe</a> (Mail, contacts, calendar, notes, bookmarks, remote find/lock/wipe), <a href="#">Yahoo</a> (Mail), Exchange ActiveSync (Exchange Server mail, contacts, calendar, remote wipe/admin; also supports Google Gmail/contacts and Calendar).	<a href="#">Google Gmail/contacts and Google Calendar</a> ; Exchange ActiveSync (Exchange Server mail, contacts, calendar [ <a href="#">required hardware encryption not yet supported on any Android phones; remote wipe/admin features not functional</a> ]).
Cloud media, software and ebooks store services	Apple's <a href="#">iTunes</a> , <a href="#">App Store</a> , <a href="#">iBooks</a> iOS apps (Music, TV, movies, rentals, audiobooks, podcasts, iTunes U, apps, ebooks), Amazon Kindle app (ebooks), Netflix iPhone announced, others.	<a href="#">Android Market</a> (apps), Amazon Kindle app (ebooks), Audible beta app (audiobooks), others. Google plans to offer music sales, Netflix planning an Android app. <a href="#">No TV/movie sales or rentals</a> .

圖7-54

– 不過Google的Android推出後在市場上迅速超過Apple的iOS，  
Google有望後來勝出。如果Apple不改變目前這種OS只與自  
己的機器繫結的模式，歷史將會重演：Google的Android必  
將像當年微軟的Windows打敗當時遙遙領先的Apple  
Macintosh。

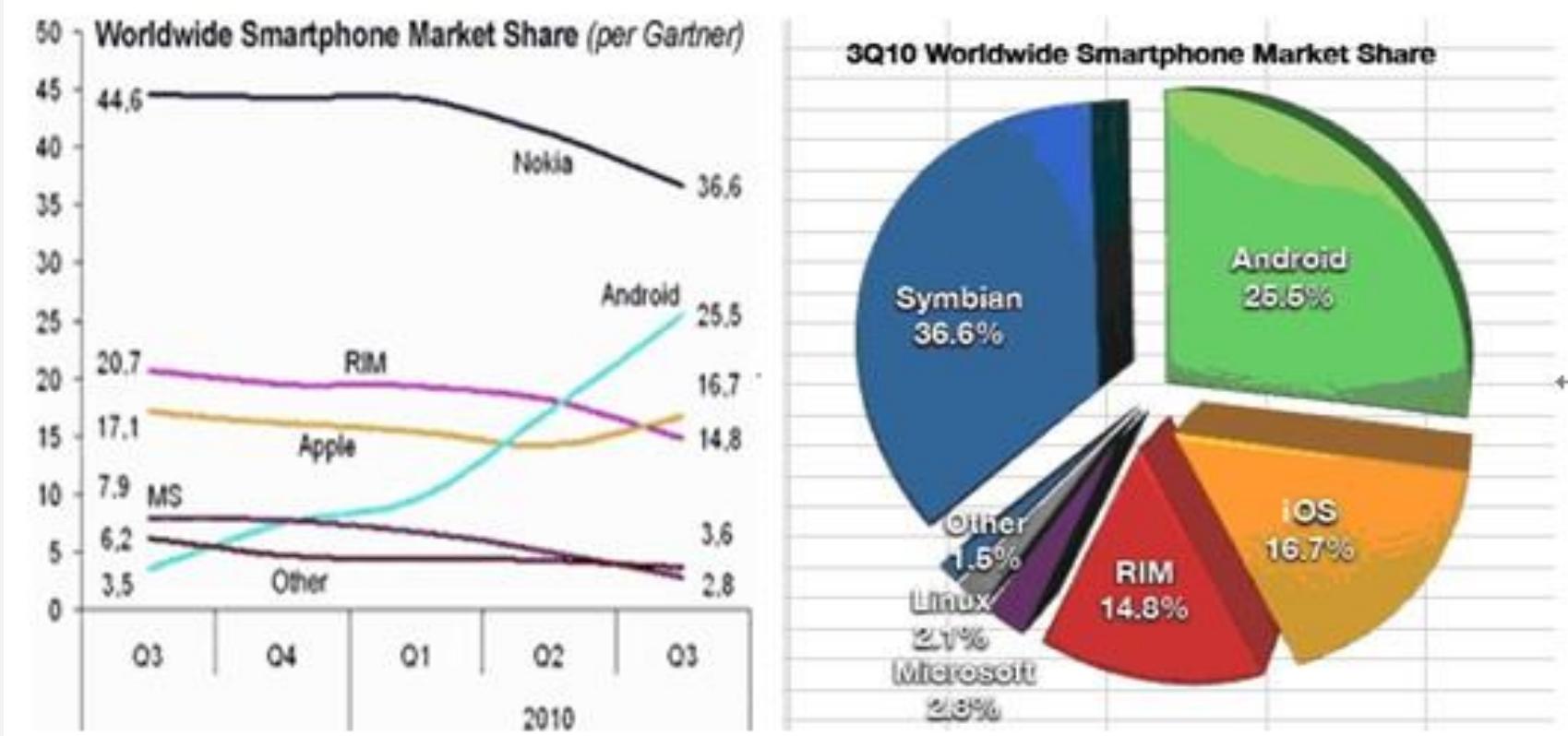


圖7-55

## 7.5 小結

- 儘管目前業界較為普遍認同的一個觀點是雲端運算包括SPI（SaaS、PaaS和IaaS）這3個部分，但SaaS技術及其業務模式出現在雲端運算之前，SaaS 3.0作為雲端運算的另一種表述。

## 9.1 雲端中間層和雲端系統

- 筆者在前面章節中把網格系統歸為**HPC**高性能計算（非單指超級計算）系統的一類，這是從“計算力”的角度來說的，網格運算這個提法被賦予很多新的特性，幾乎相當於雲端運算體系中的IaaS，而HPC則似乎變成了專指超級計算。

– “中間層” 這個詞目前還沒有被廣泛地用來表達和概括所有的可複用軟體，而是直接列出常用的、代表性的軟體，如 Hadoop、MapReduce、Bigtable 等，雲端中間層必將受到越來越多的關注，網格中間層（包括HPC中間層）將會和企業級中間層相融合或組合。

- 9.1.1 IaaS 中間層
  - HPC中間層和網格中間層，主要用於整合計算資源，實作“多機虛擬化”，IaaS中間層具有較高的通用性，之所以被稱為中間層，是因為通用性、可複用是它的重要特徵之一。
  - Eucalyptus和OpenNebula也是基於網格運算的基礎和思路發展而來的，所有基於網格或HPC的IaaS系統都可以算是“由下而上”發展成為雲端運算系統的。

– 基於網格技術和理念的IaaS中間層在本書前文已有介紹，主要是指能支援異構（*Heterogeneous*）的多機系統的平行虛擬主機系統。非網格路線的IaaS中間層如Amazon AWS等也提供和上述MPI、Condor、PBS等中間層的介面和執行支援，如圖9-3所示。

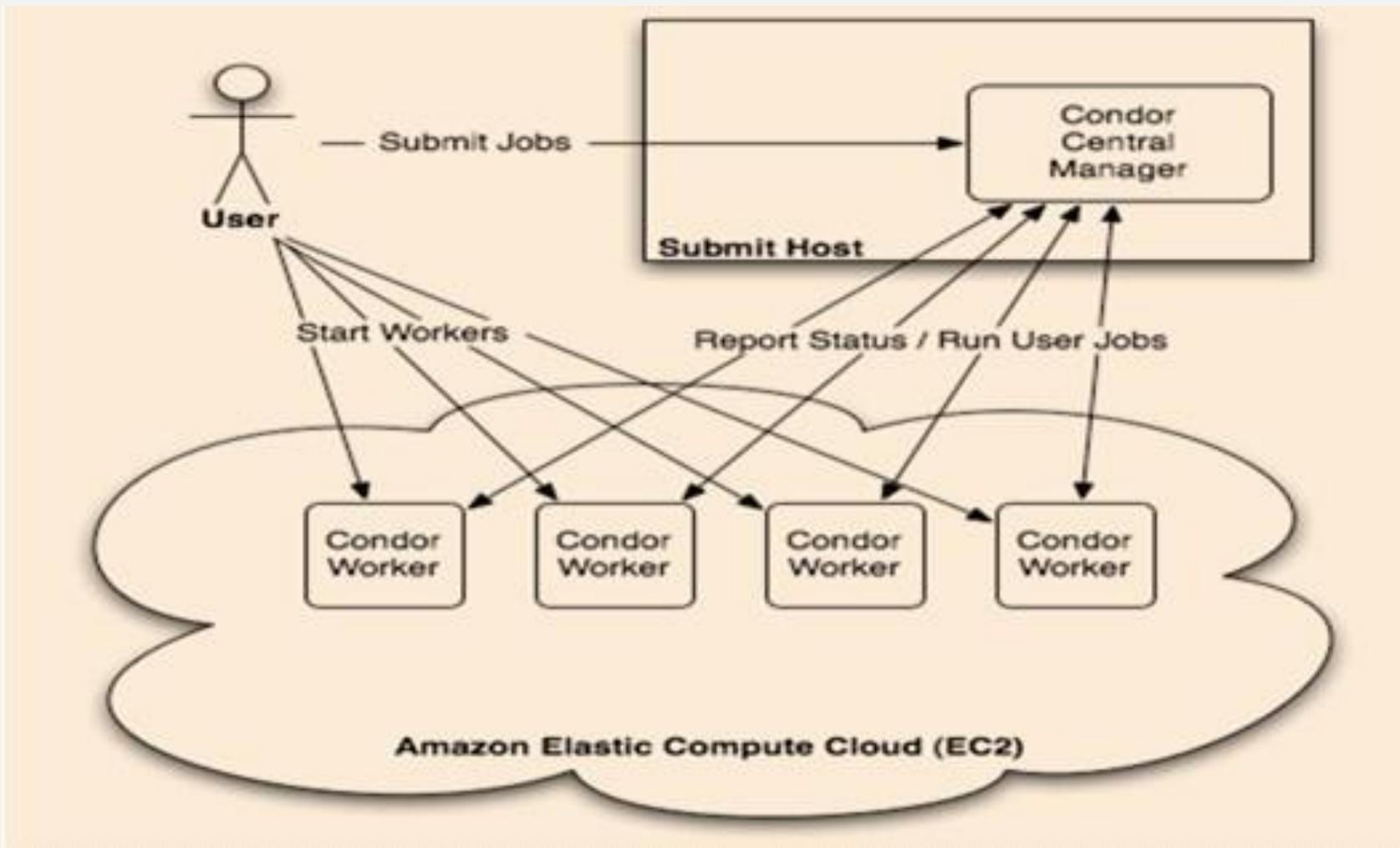


圖9-3

- 9.1.2 PaaS 中間層
  - 隨著以科學計算為主的HPC和網格運算走向以商用為主的雲端運算，企業級中間層技術也必將和雲端運算走向融合，如圖9-4。傳統的中間層技術將成為雲端運算PaaS的基本技術。PaaS中間層仍將延續Java EE與.NET兩大陣營對峙的局面，如圖9-5所示。

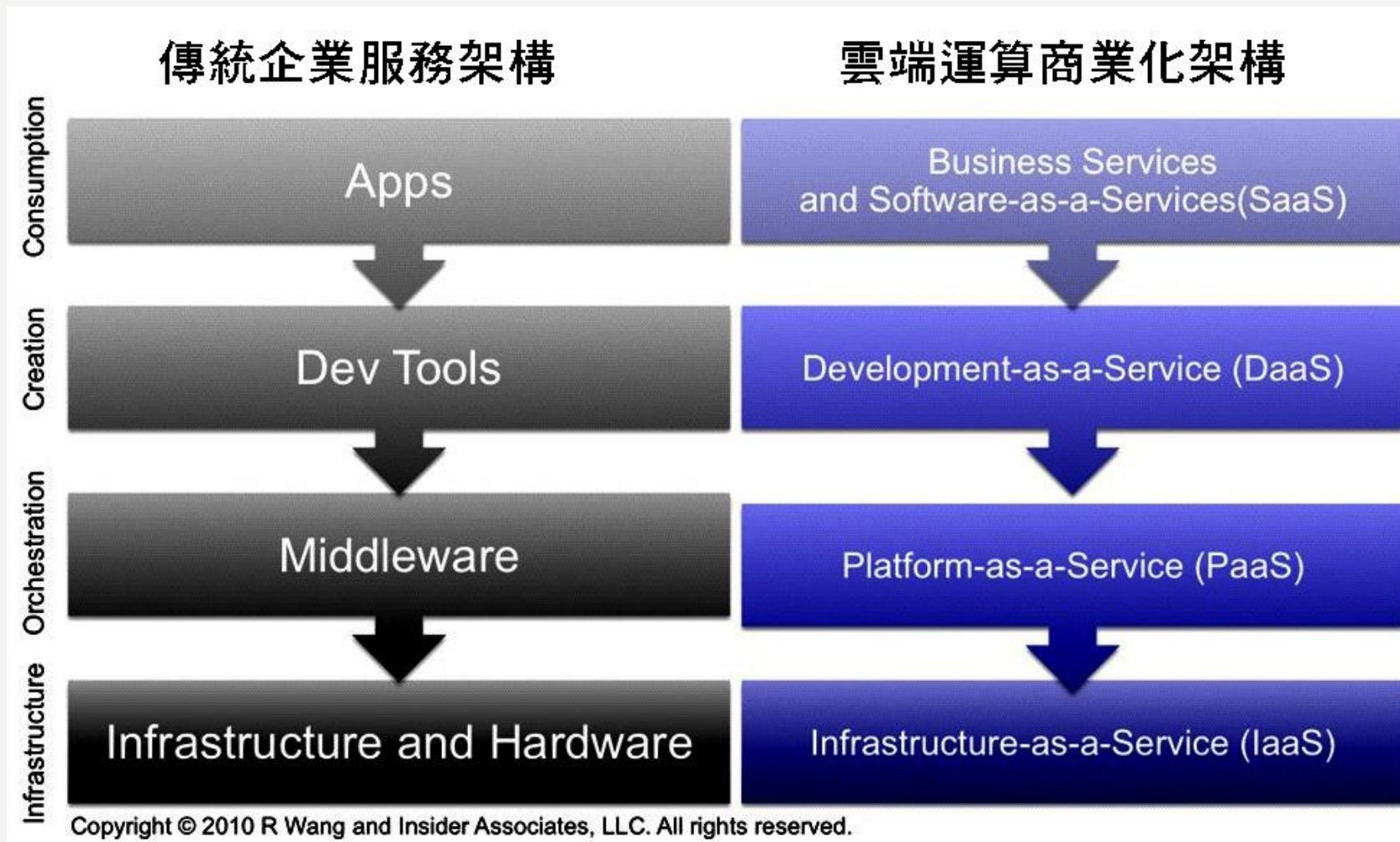
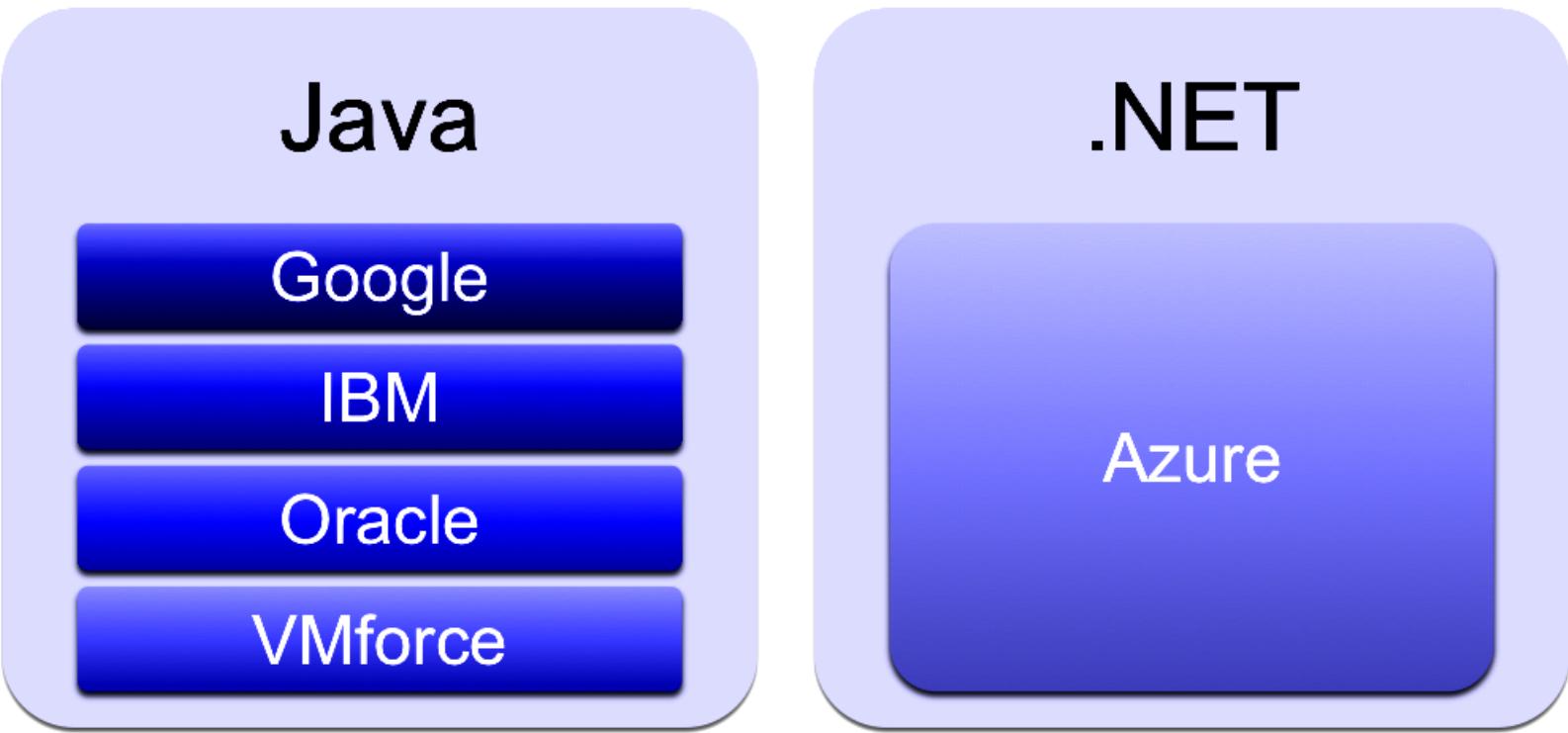


圖9-4

## Creation Tools Layer



Copyright © 2010 R Wang and Insider Associates, LLC. All rights reserved.

圖9-5

- 在企業整合中間層（如App Server）的上面還有一層中間層屬於一種叫“業務基礎軟體平台”的通用中間層，如圖9-6所示。

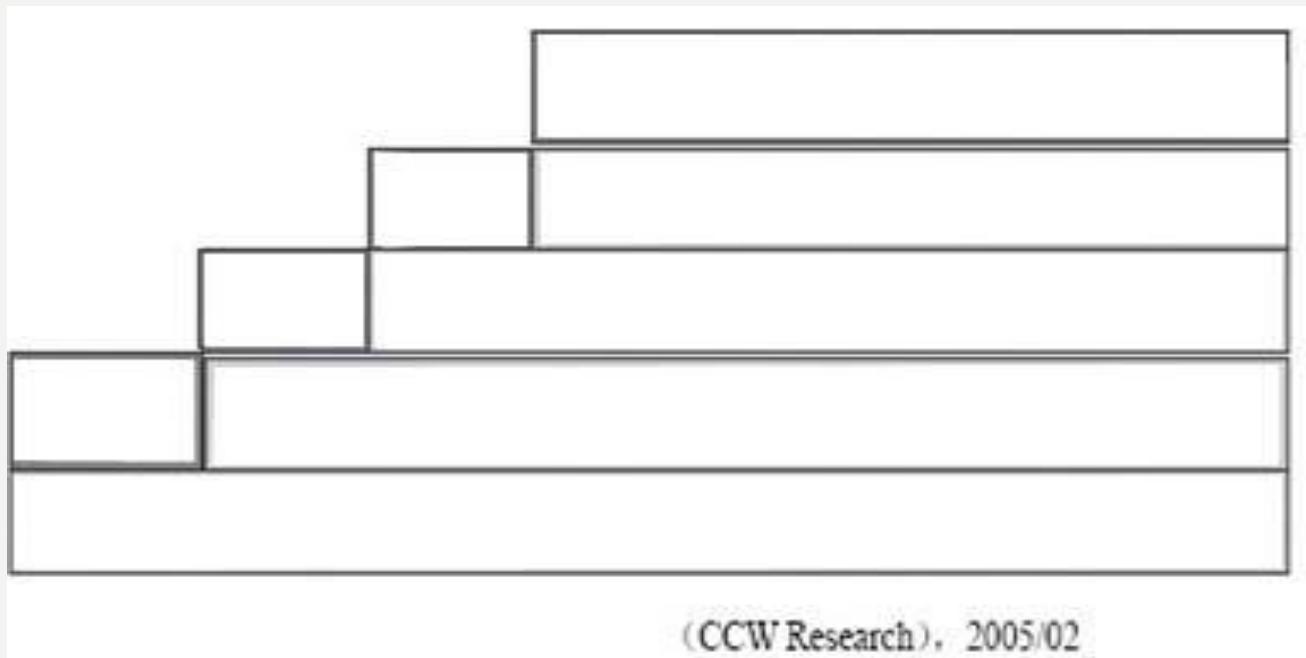
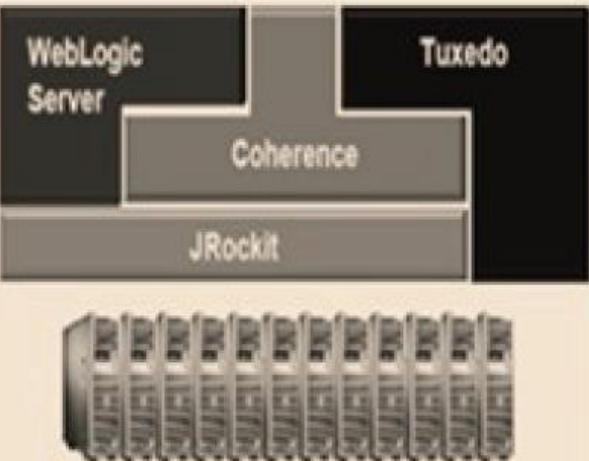


圖9-6

– 如圖9-7所示是Oracle公司基於原有中間層的PaaS平台，具有典型意義，IBM、SAP等企業也有類似方案。

## Fusion Middleware Application Grid

Comprehensive, hot-pluggable, pre-integrated



Enterprise Manager

WebLogic Server

Coherence

JRockit and HotSpot

Exalogic Elastic Cloud Software

Oracle Linux or Solaris

Exalogic Elastic Cloud Hardware

圖9-7

- 圖9-7所示的軟體基礎架構平台（中間層）層面之上增加如下一些面向多租戶架構的功能和工具：

- 多租戶架構的支援，實作多種多租戶模型。
- 快速（Rapid Application Development，RAD）開發和部署工具。
- 工作流（Business Process Management，BPM）實作協同系統功能。
- 內容管理（ECM）、報表和Dashboard功能。

- PaaS中間層基於傳統的企業整合中間層，
- 如App Server、MQ/ESB/SOA等，它的顯著特點之一就是在傳統企業整合中間層和業務基礎中間層（SES）的基礎上實作了不同層次的多租戶支援架構，如圖9-9所示。

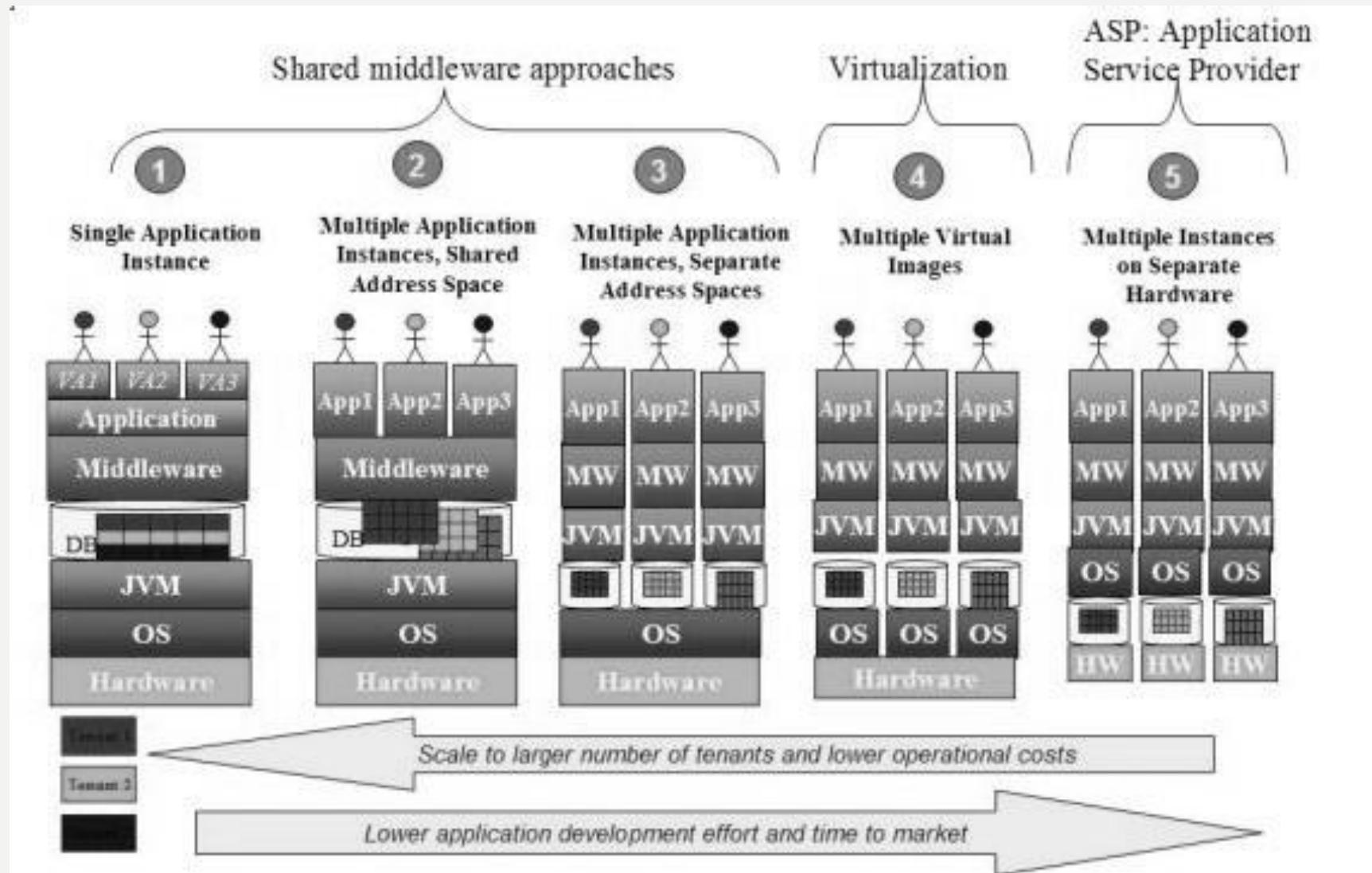
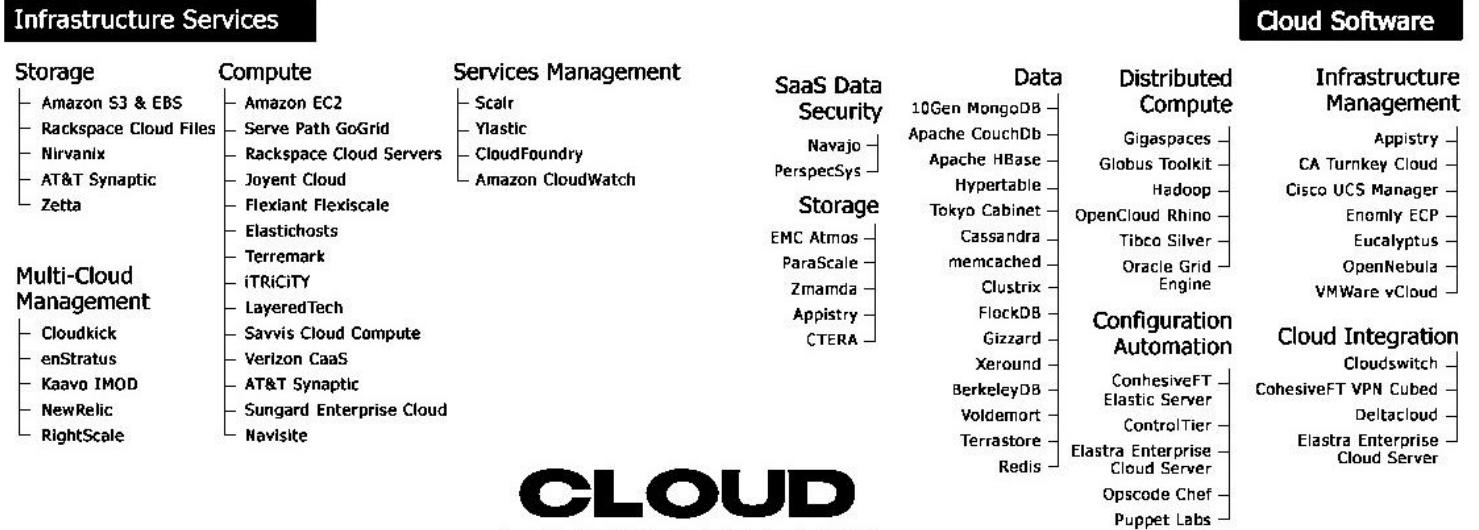


圖9-9

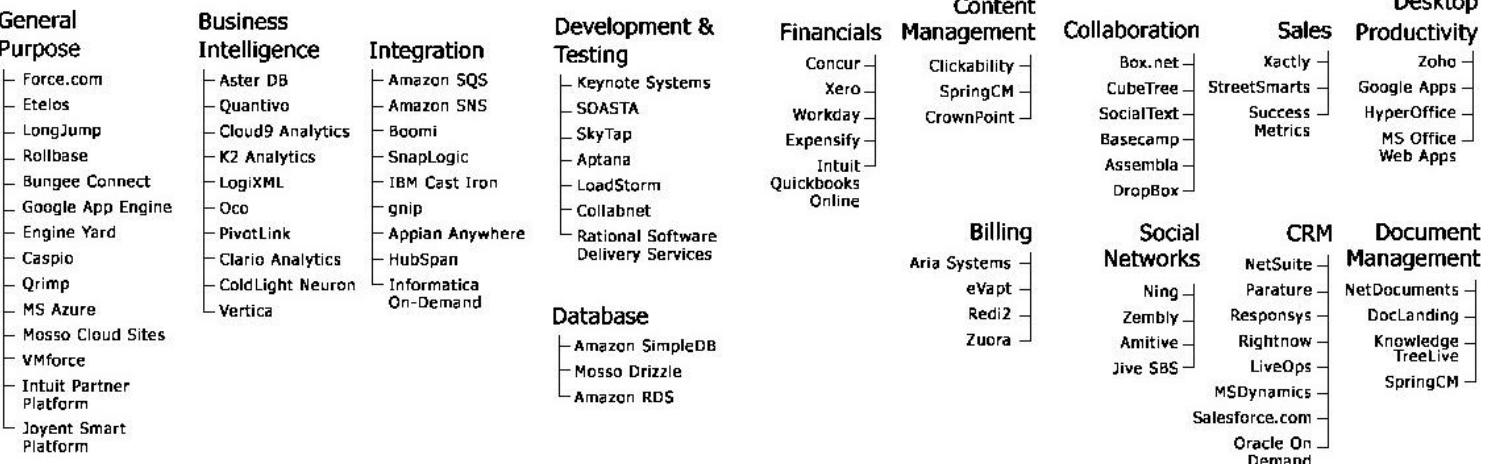
– 如圖9-9，第I種多租戶模式是最高效率、最徹底的多租戶架構，需要對底層中間層和SES應用層中間層做程式碼上的修改才能實現。這也是Salesforce最早提出多租戶模式時實作的SaaS和PaaS架構。

- 9.1.3 雲端系統
  - 這些平台或系統所採用的技術和產品大部分都可以算是雲端中間層。



# CLOUD TAXONOMY

## Platform Services



—雲端運算中間層除了包括基於企業級中間層的PaaS內容以外，還包括前文所述的基於網格中間層的IaaS中間層，如圖9-11所示。

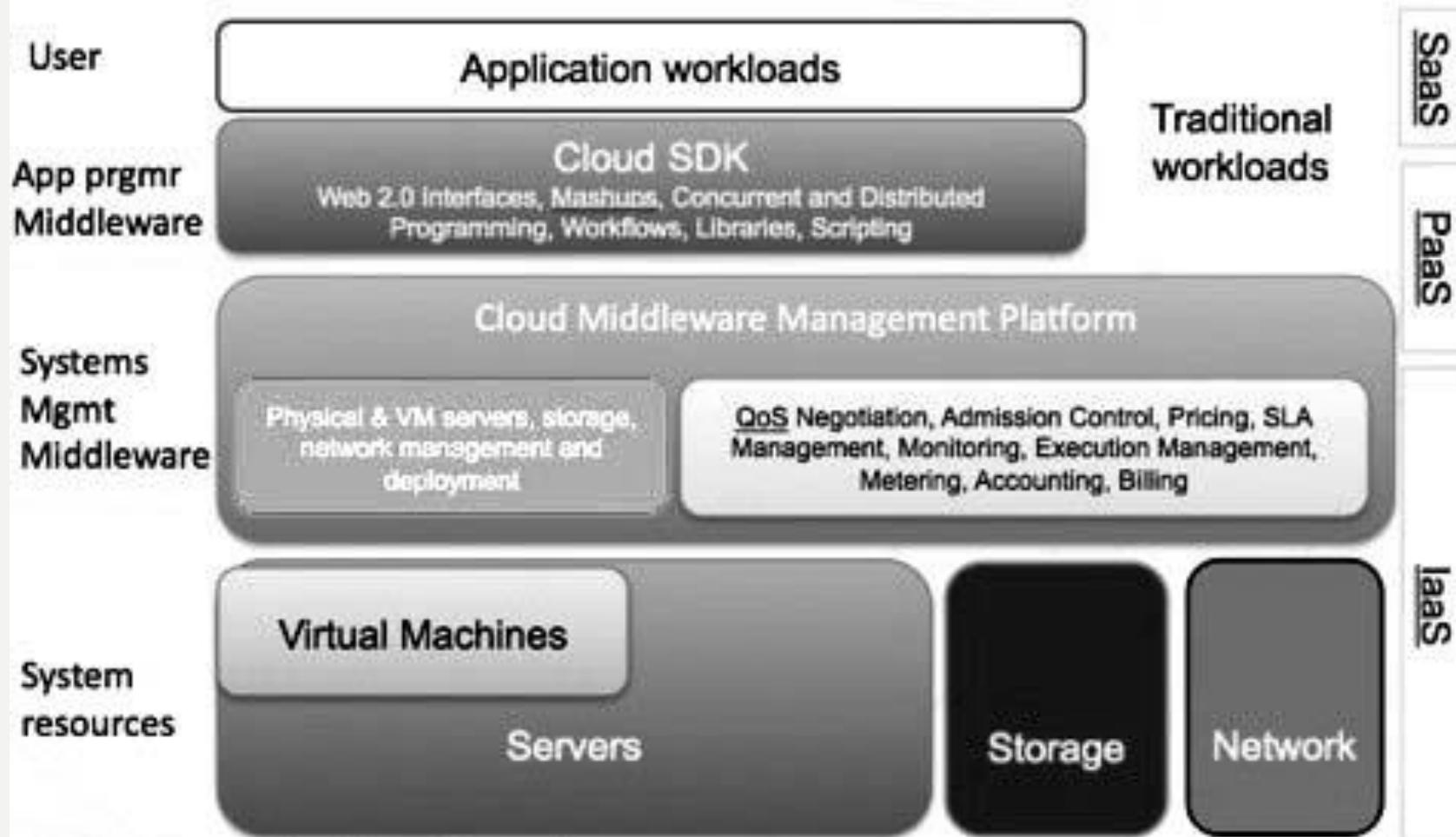


圖9-11

– 包括IaaS和PaaS中間層的雲端中間層是雲端運算系統的核心，但一個完整的雲端運算系統還需要其他業務構件的支援，一般包括圖9-12的6個部分。

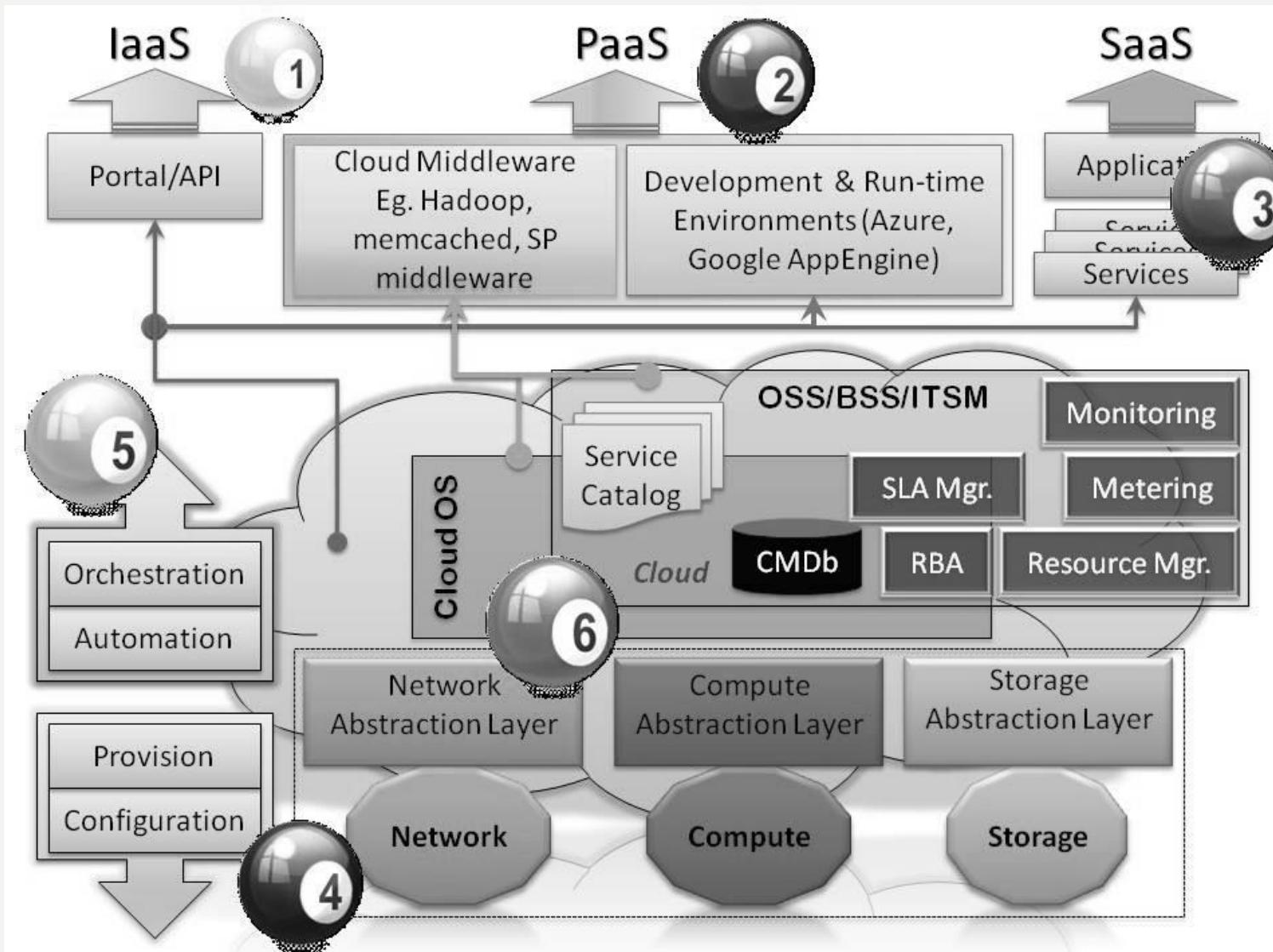


圖9-12

– 圖9-13是一個雲端運算平台的另一種表述，即開發工具、安全框架、系統管理和SPI相結合形成完整的雲端系統。

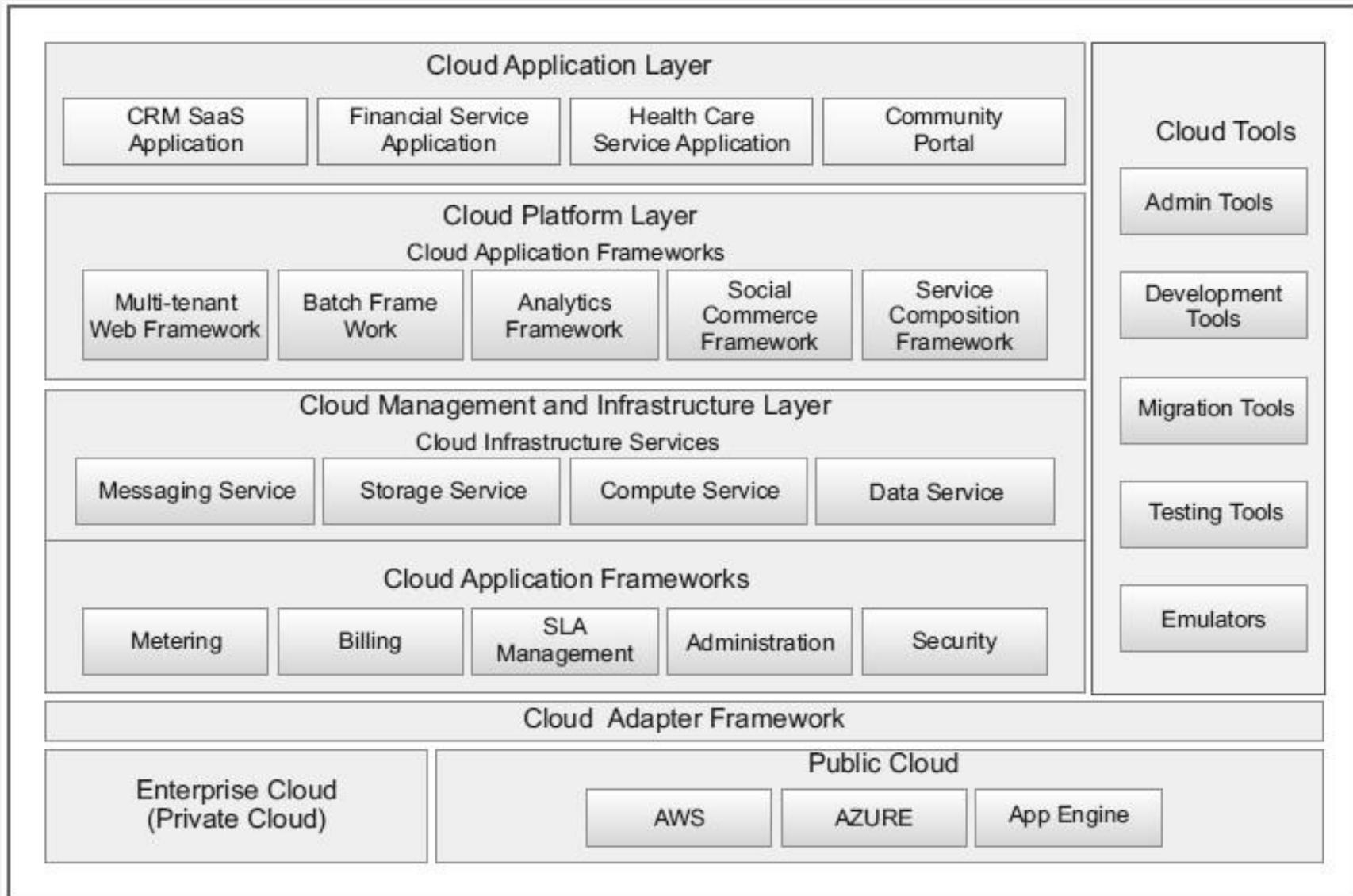


圖9-13

## 9.2 典型**IAAS**技術和系統

– Gartner於2010年底公佈了主要IaaS廠商的Magic Quadrant（魔力象限）。Amazon AWS仍然被列為最有遠見（Visionary）的企業，而Savvis被列為最具執行力的企業。

– Savvis ( NASDAQ : SVVS ) 之前是一個託管 ( Hosting ) 提供商，在2002年，Intel決定退出託管業務，Savvis被Intel選擇為其使用者提供託管和網路服務，後來Savvis收購了Cable & Wireless USA，包括3000個企業使用者、15個資料中心和一條TI鏈路。Savvis現在已經擁有超過2000名員工和28個資料中心。不過，Savvis在雲端運算的SPI幾個領域在技術和業務上似乎沒有什麼大的作為。

- Rackspace也是一家全球領先的託管服務提供商，後來轉向雲端運算業務，其提供的雲端服務叫Mosso。
- Rackspace於1998年成立，目前在全球擁有10個資料中心，管理超過50000台伺服器。2008年在紐交所上市，自2001年起每年均保持盈利。
- Rackspace以Apache 2.0許可證將其雲端運算平台程式碼開源，這是OpenStack專案的一部分，OpenStack是美國NASA和Rackspace合作的產物。

- Joyent在雲端運算風行之前的2004年即開始提供 “基礎建設即服務（IaaS）” ，並且被稱為最大的雲端運算提供商之一。
- Joyent曾宣稱，它每個月為十億計的最終使用者提供數以千計的Web、行動和伺服器應用服務，因此有研究報告也把它列為PaaS服務商。
- 目前Joyent已經為約25%的Facebook的應用提供了平台支援，其客戶之一Linkedin單月的存取量也已經突破10億，
- 另一個和Twitter類似的流行微博客服務提供商Identi.ca也選擇了Joyent，
- Joyent目前的員工總數僅為22人，還包括管理和銷售人員。

- Nebula計算平台，是美國NASA位於矽谷地區的Ames研究中心的一個內部專案，目的是集中計算資源，採用開源的免費軟體，為NASA科學家和工程師提供低價的雲端運算服務。
- 由於必須滿足FISMA ( Federal Information Security Management Act ) 法案，商業雲端運算系統很難做到，於是就出現了Nebula這個內部專案。
- Nebula有一個開源的版本OpenNebula，而且Nebula也是OpenStack系統的貢獻者之一。
- Nebula遵循Open Virtualization Format、AMQP、SQLAlchemy等標準。

- 9.2.1 Amazon Web Services ( AWS )
  - 提供一組雲端運算IaaS服務，使用者可透過應用程式存取Amazon的計算基礎設施，
  - 其中最主要的兩個服務是
    - Amazon EC2 ( Elastic Compute Cloud )
    - Amazon S3 ( Simple Storage Service ) 。
  - Amazon AWS從2002年7月開始向使用者提供各種IaaS服務，主要包括：
    - 儲存 ( S3 )
    - 計算 ( EC2 )
    - 訊息傳遞 ( SQS )
    - 資料集 ( SimpleDB )

– 透過在Amazon提供的可靠且經濟的服務上構建Web應用，可以實作複雜的企業應用程式，這些應用駐留在公有雲端中，因為硬體由Amazon維護和服務，所以使用者不需要承擔維護費用，只需根據使用的資源付費。

- 多數雲端運算系統都提供與AWS的擴充介面（如圖9-19所示是RedHat的MRG（ Messaging 、 Realtime 和 Grid ）系統的EC2整合和基於Condor的資源管理和作業排程），
- **AWS已成為一個業界事實標準**，
- 下面是幾個典型的AWS成功應用案例。
  - SmugMug。
  - 《紐約時報》。
  - Animoto。

## How to Schedule From MRG to Amazon EC2

- MRG Grid will manage the starting, monitoring, and cleaning up of EC2 resources
- Condor jobs on Amazon EC2 are Amazon Machine Images (AMI's)
- Amazon EC2 jobs can run as sub-jobs
  - Automatically add capacity at EC2 when local grid is fully utilized
  - Chain jobs together
- To Schedule Condor jobs on EC2:
  - You:
    - Configure Condor to use EC2
    - Create an AMI, upload it to S3, register it with EC2
    - Create a job description and `condor_submit` it
  - Condor will:
    - Start AMI in EC2
    - retrieve user data
    - transfer input
    - run job
    - transfer output
    - shutdown AMI
    - Cleanup EC2 resources
    - Remove job from schedd

圖9-19

- AWS的硬體資源架構如圖9-20所示，一個物理節點（Physical Node），從普通的單核x86伺服器直到8核的64位元伺服器）上透過開源的Xen虛擬化系統可以虛擬出一個或多個工作節點（Worker Node），
- 透過AWS動態地按需分配給使用者請求。
- AWS還可以透過執行SGE、Condor等資源管理與排程系統實作更細顆粒度的動態分配，
- 使用者也可以自己執行適合自己需求的作業排程系統。

- 每個AWS工作節點上執行一個基於各種版本的Linux（目前AWS只支援Linux，逐步會擴充到支援其他OS）的應用實例（Instance），實例上可以執行任何應用程式或軟體。
- AWS還提供可選的機器映射（Amazon Machine Image，AMI）作為實例的範本，還可以透過指定許可權控制對實例的存取，為使用者提供更多的方便和簡化使用難度。

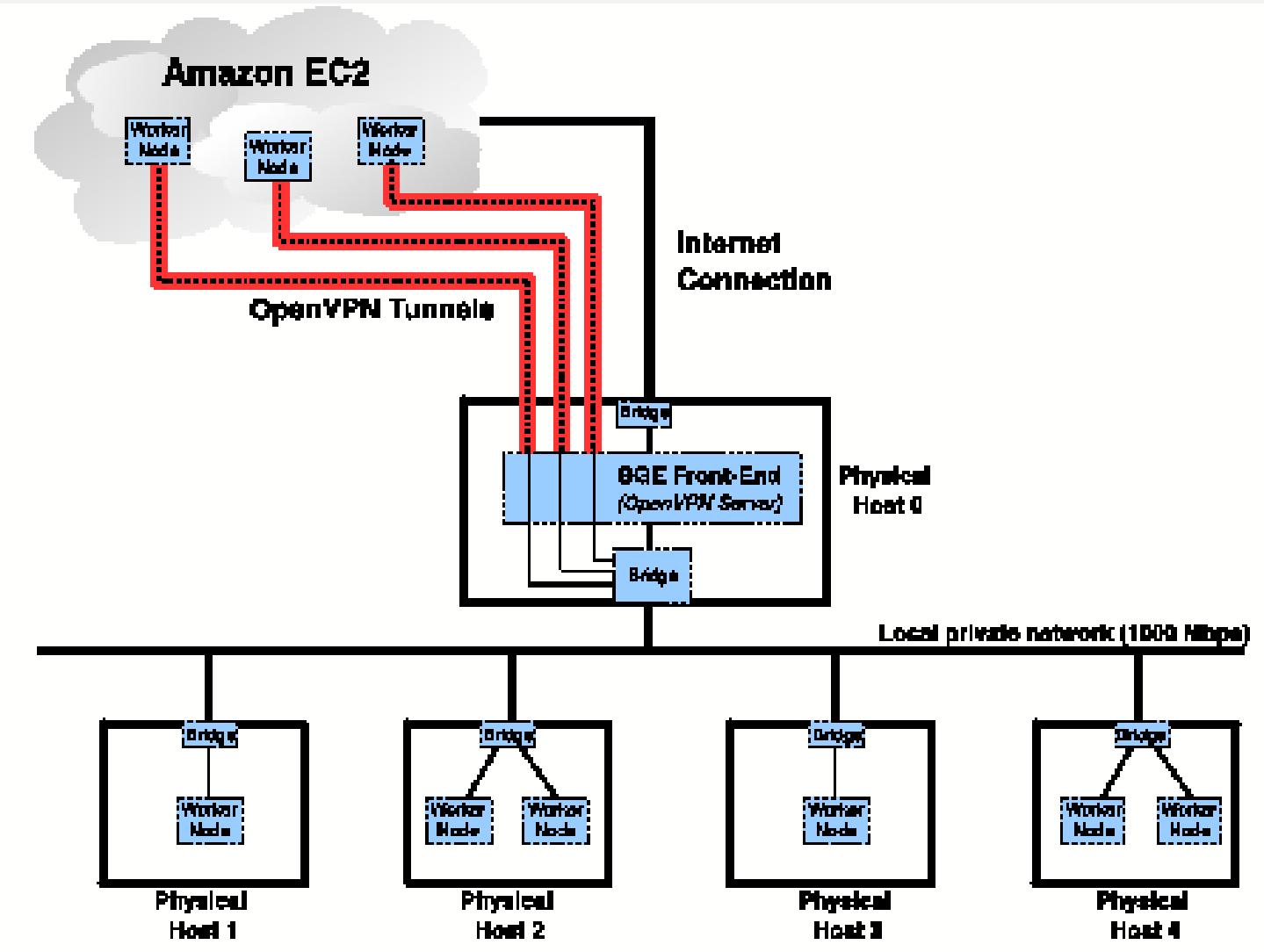


圖9-20

## Eucalyptus Architecture: WS-Cloud

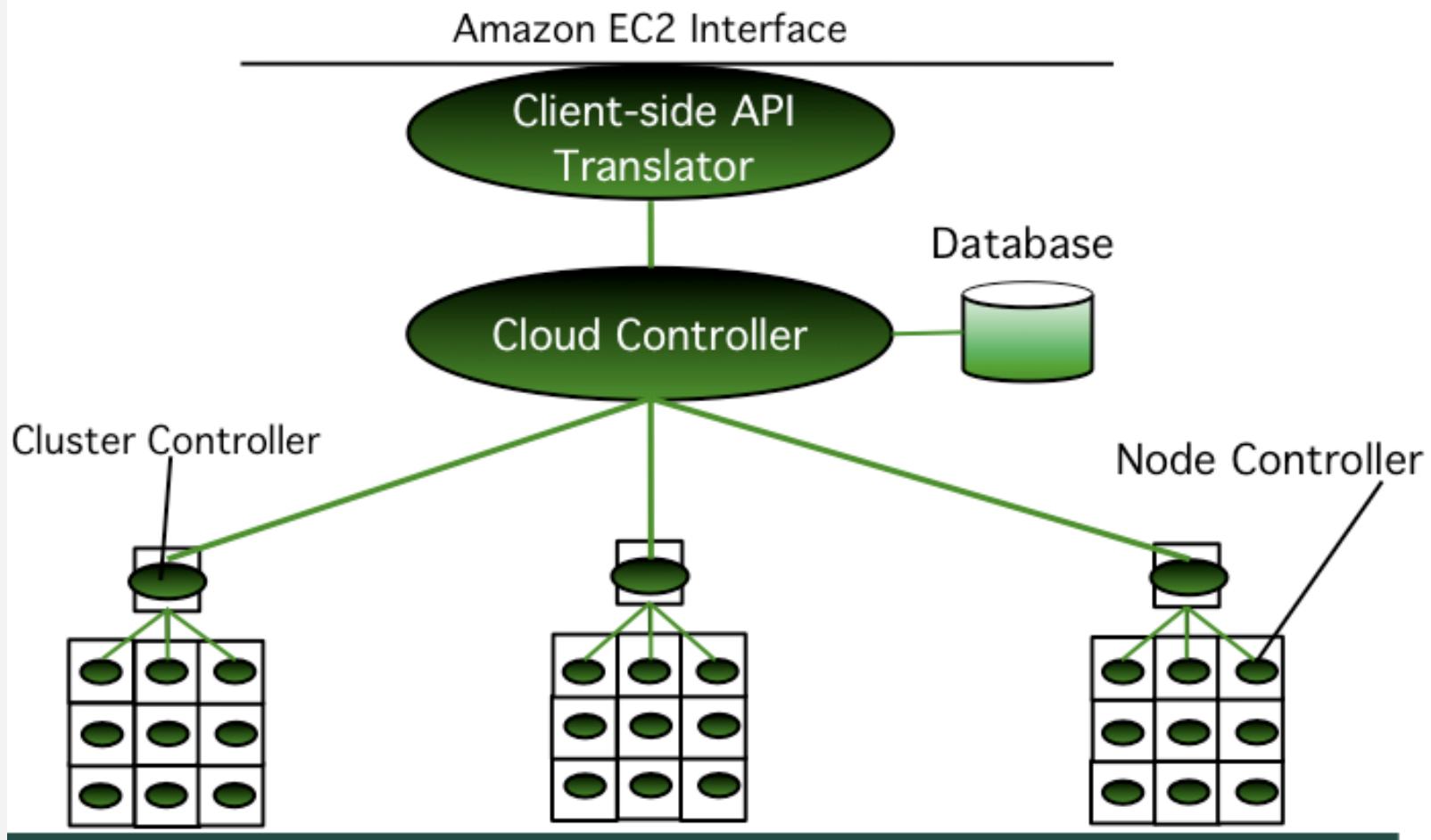


圖9-20

- AWS為使用者提供了標準的SOAP和REST介面，
- 並提供負載均衡（ Load Balance ）和自動伸縮（ Auto Scaling ，即自動獲取和釋放計算資源）功能，如圖9-21所示。

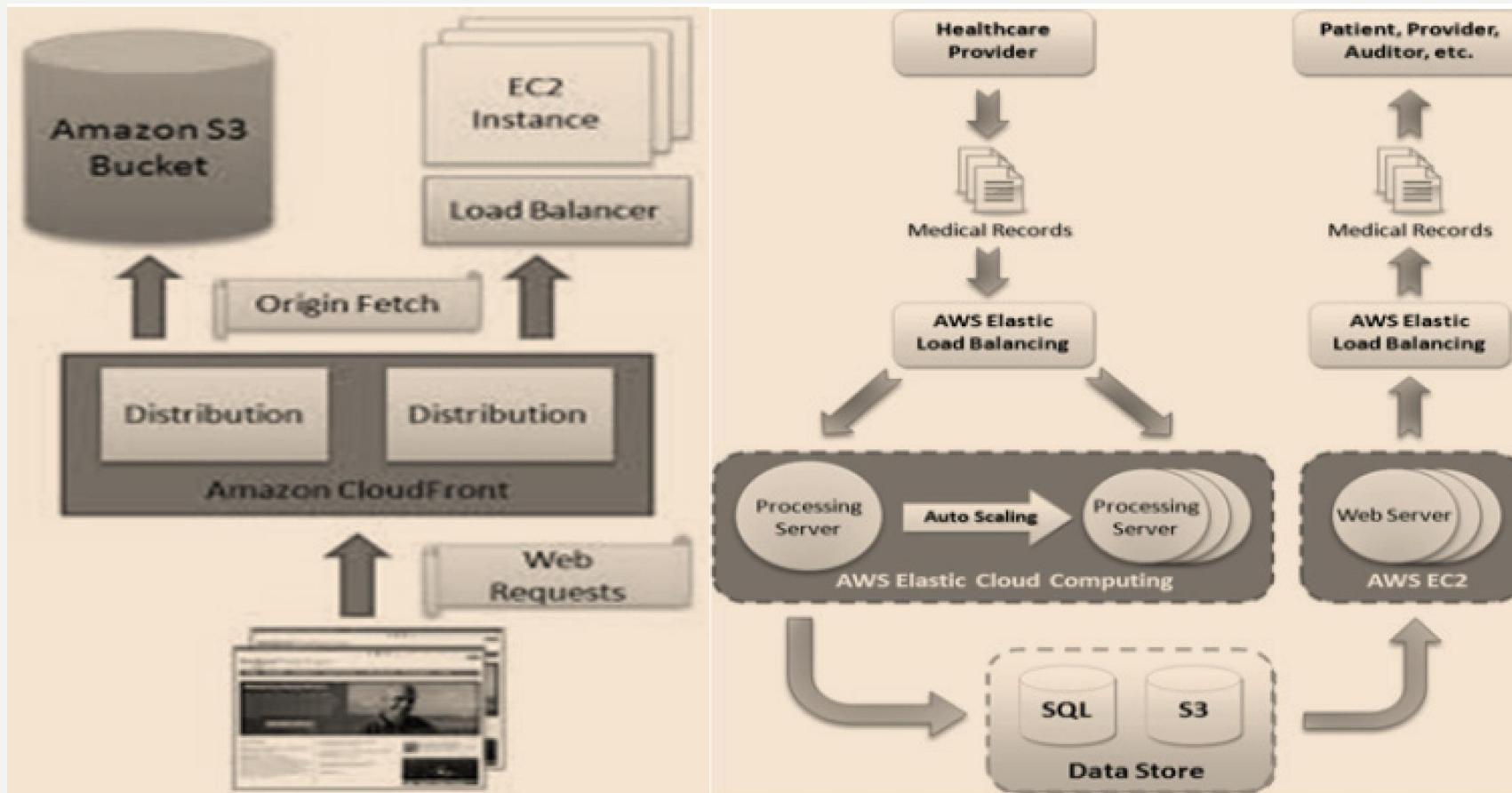


圖9-21

- AWS以IaaS服務為主，只提供一些簡單的PaaS服務，
- 如使用其**Simple Queue Service**可實作使用者應用之間的訊息傳遞，
- 使用**SimpleDB**實作應用程式的結構化和非結構化資料儲存和查詢服務等。
- Google提供SPI三個層面的全方位服務，
- Amazon基本上只提供IaaS服務，但Amazon的IaaS服務比Google的IaaS服務規模和影響力都要大，可見專注是成功的關鍵。

### • 9.2.3 開源 IaaS 與 OpenStack

- 開源社區對ICT產業經濟發展與創新都有著深遠的影響，
- 在資料庫方面，有開源產品MySQL、PostgreSQL等；
- 在Web伺服器方面，有Apache；
- 在應用伺服器方面，有Tomcat、JonAS、JBoss、Jetty、GlassFish等；
- 在指令碼語言（Script language）方面，有Java、Perl、PHP等；
- GNU Linux更是對商業化作業系統及手機作業系統做出了眾多貢獻。

- 開源軟體讓雲端運算服務提供商和最終使用者降低成本的同時，也提供了許多靈活性和可選擇性，避免被廠商 “鎖定 ( Lock-in ) ” 。
- 下面列出了目前主要的與開源雲端運算業務相關的企業或產品。
  - Eucalyptus Systems公司。
  - Redhat公司。
  - Traffic Server。
  - Cloudera。

- Enormaly °
- Joyent °
- Zoho °
- EyeOS °
- Globus Nimbus °
- Reservoir °
- OpenNebula °

- NoSQL理念的類似Bigtable的開源雲端資料庫，
- 如
  - Cassandra
  - Hadoop/Hbase
  - CouchDB
  - MongoDB
  - Membase
  - Memcachedb
  - Neo4j
  - FlockDB 等等。
- 有一些專用的雲端運算系統，如Hadoop、TPlatform等並不需要虛擬化技術的支援。

- 網路託管公司Rackspace和美國國家航空暨太空總署（NASA）於2010年末聯手推出的雲端運算OpenStack在開源軟體與雲端運算業界引起極大關注，被譽為目前最重要的雲端運算產品。
- RackSpace將開源其用於提供雲端儲存和雲端運算服務的軟體。
- NASA也將把其Nebula雲端運算平台的技術整合至OpenStack。
- OpenStack還將致力於增強不同雲端運算服務之間的互操作性，避免不同廠商相互孤立的局面。

- 雲端運算和雲端儲存兩個專案，OpenStack雲端運算主要作用在於為虛擬伺服器提供自動創立和管理，
- OpenStack雲端儲存可以建立大量的、可擴充的物件儲存軟體。

## 9.3 典型**PaaS**技術和系統

- 不同的PaaS提供不同組合的服務，綜合性的PaaS是一個集：
  - 開發
  - 測試
  - 部署
  - 託管
  - 應用
  - 維護
- 為一體的整合運維環境，如圖9-27所示。

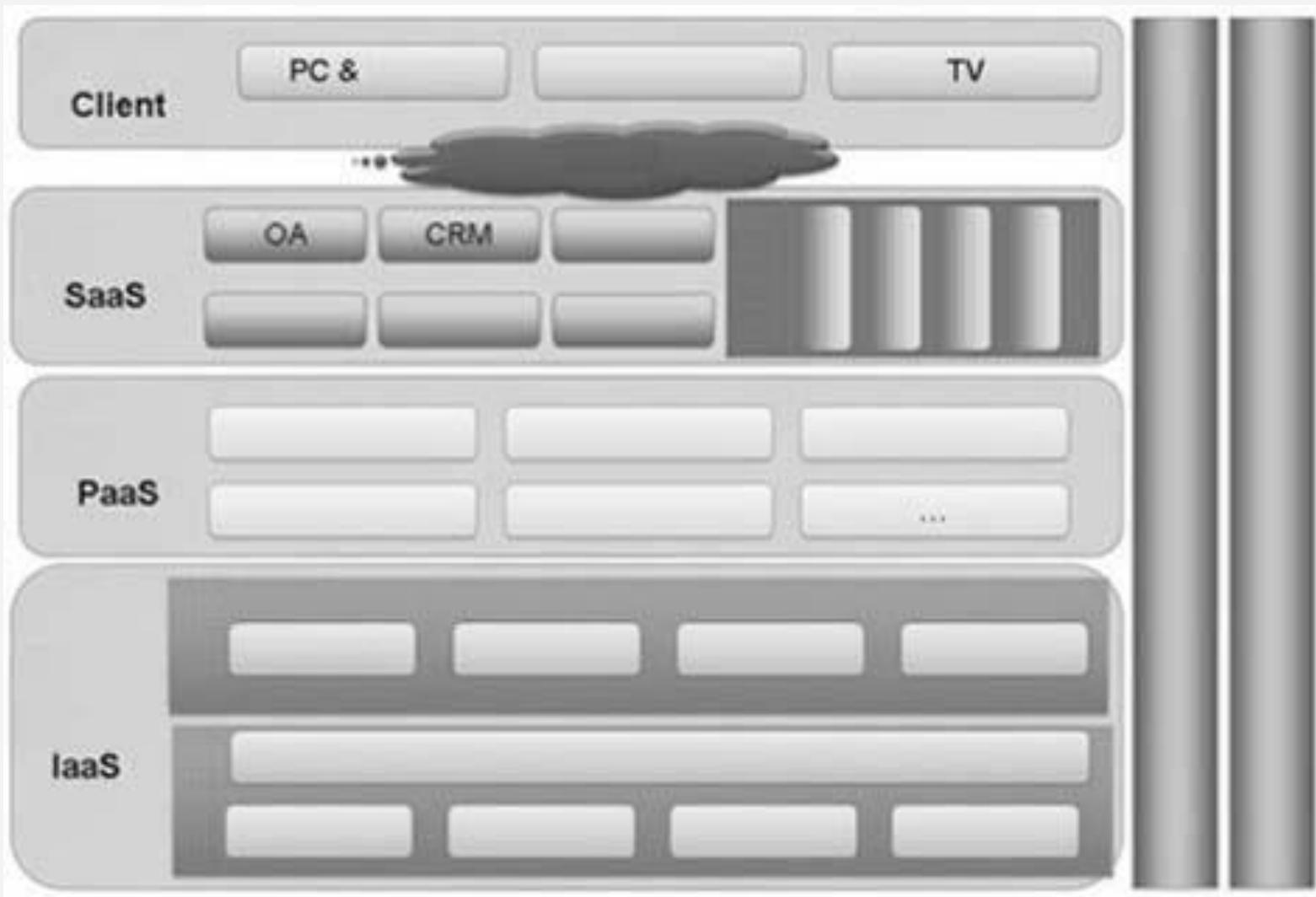


圖9-27

– 如圖9-28所示是幾個最常用的PaaS平台。

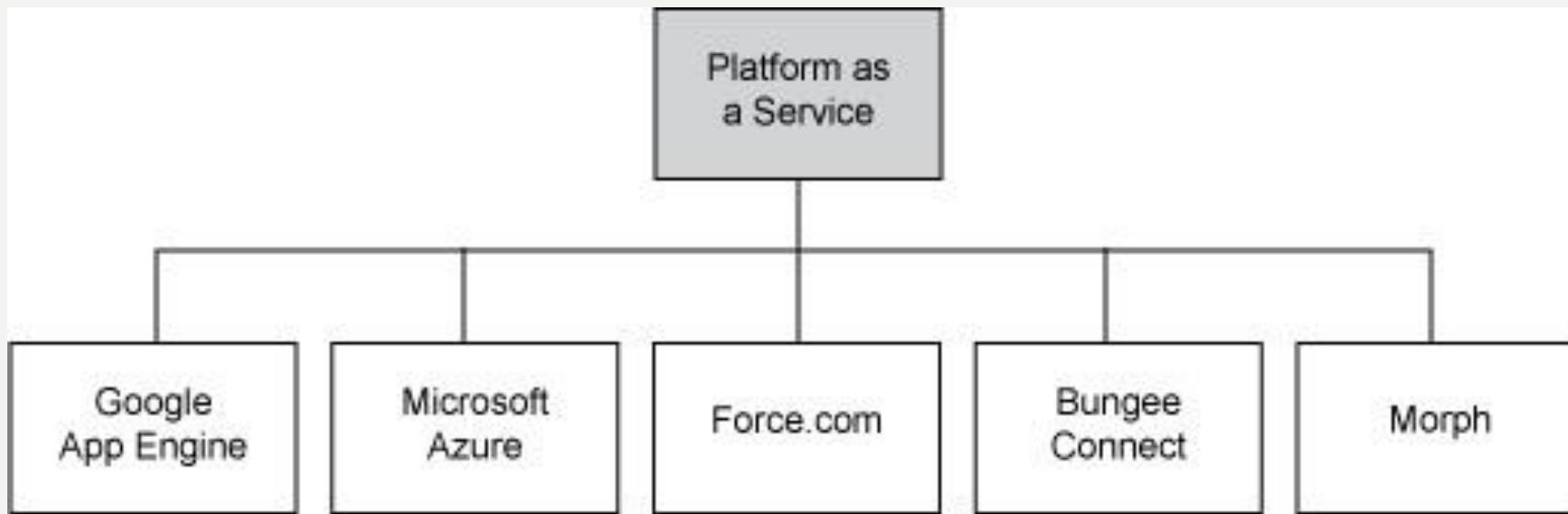


圖9-28

- 透過PaaS模式，使用者可以在一個包括中間層執行Runtime、SDK、文件和測試環境等在內的開發平台上非常方便地編寫應用，
- 而且不論是在部署或在執行的時候，使用者都無須為伺服器、作業系統、網路和儲存等資源的管理操心，這些煩瑣的工作都由PaaS供應商負責處理，
- 而且PaaS在資源整合上面效率很高，例如，一台執行Google App Engine的伺服器能夠支援成千上萬的應用，
- 也就是說，PaaS是非常經濟的，同時可實作節能降耗。
- PaaS主要的使用者是軟體發展、部署和系統管理人員。

- 典型的PaaS系統分為如下幾類。
  - 依附於SaaS應用的訂制、擴充和整合PaaS。
  - 獨立的PaaS環境。
  - 託管PaaS執行環境。
  - 開放的PaaS平台。

- 將PaaS和IaaS進行如下比較：
  - **開發環境**：PaaS一般會給開發者提供一整套包括IDE在內的開發和測試環境，而IaaS一般提供類似AWS和網格的開發和部署環境。

- 支援的應用：因為 IaaS 主要是提供虛擬主機，而且普通的虛擬主機能支援多種作業系統，所以 IaaS 支援的應用的範圍是非常廣泛的。
- 但如果要讓一個應用能執行在某個 PaaS 平台不是一件輕鬆的事，因為不僅需要確保這個應用是基於這個平台所支援的語言，而且還要確保這個應用只能呼叫這個平台所支援的 API，
- 如果這個應用呼叫了平台所不支援的 API，那麼就需要對這個應用進行修改。

- **開放標準**：雖然很多IaaS平台都存在一定的私有功能，但是由於OVF等協定的存在，使得IaaS在跨平台和避免被供應商鎖定這兩方面是穩步前進的。而PaaS平台的情況則不容樂觀，因為不論是Google的App Engine，還是Salesforce的force.com都存在一定的私有API。
- **可伸縮性**：PaaS平台會自動調整資源來幫助執行於其上的應用更好地應對突發流量。而IaaS平台則需要開發人員手動對資源進行調整才能應對。

- 整合率和經濟性：PaaS平台整合率是非常高的，
  - 例如，PaaS的代表Google App Engine能在一台伺服器上承載成千上萬的應用，
  - 而普通的IaaS平台的整合率最多也不會超過100，而且普遍在10左右，使得IaaS的經濟性不如PaaS。
- 多租戶模式和多租戶軟體架構：大部分PaaS系統提供基於多租戶軟體架構的多租戶營運模式，有的PaaS不提供或只支援非完全的多租戶軟體架構，參見前文對多租戶軟體架構的細分。

- **計費和監管**：因為PaaS平台在**計費和監管**這兩方面不僅達到了IaaS平台所能企及的作業系統層面，例如，CPU和記憶體的使用量等，而且還能做到應用層面，
  - 例如，應用的**反應時間**（Response Time）或者應用**所消耗的事務多少**等，這將提高計費和管理的精確性。
- **學習難度**：因為在IaaS上開發和管理應用同現有的方式比較接近，而在**PaaS上開發則有可能需要學一門新的語言或者新的框架**，所以IaaS學習難度更低。

- 9.3.2 App Engine 和 AppScale

- PageRank技術是Google的看家技術，其高效率的實作由7大與雲端運算分為3大類。

- 基礎設施：GFS、Chubby和Protocol Buffer。
    - Google搜尋引擎。
    - Chubby是一種分散式鎖服務，透過Chubby，系統中的上千個Client都能夠對於資源進行“加鎖”或“解鎖”實作同步，常用於BigTable的協同工作。

- Protocol Buffer是一種串列化、結構化資料的方式，採用二進位的格式，其速度是使用XML進行資料交換時的10倍左右。
- 大規模資料處理：MapReduce和Sawzall。

- MapReduce主要透過“**Map**（映射）”和“**Reduce**（化簡）”這兩個步驟來實作第5章中曾介紹過的並行處理術語中所說的所謂**Data-Parallel**資料處理。
- Google宣稱MapReduce是源自早期的函數式（Functional Programming）語言，第5章中筆者曾提到過，其實在MPI/PVM中早已實作了類似操作（如圖9-34所示）。

# MPI\_Bcast

Broadcasts a message to all other processes of that group

count = 1;

source = 1;

broadcast originates in task 1

`MPI_Bcast(msg, count, MPI_INT, source, MPI_COMM_WORLD);`

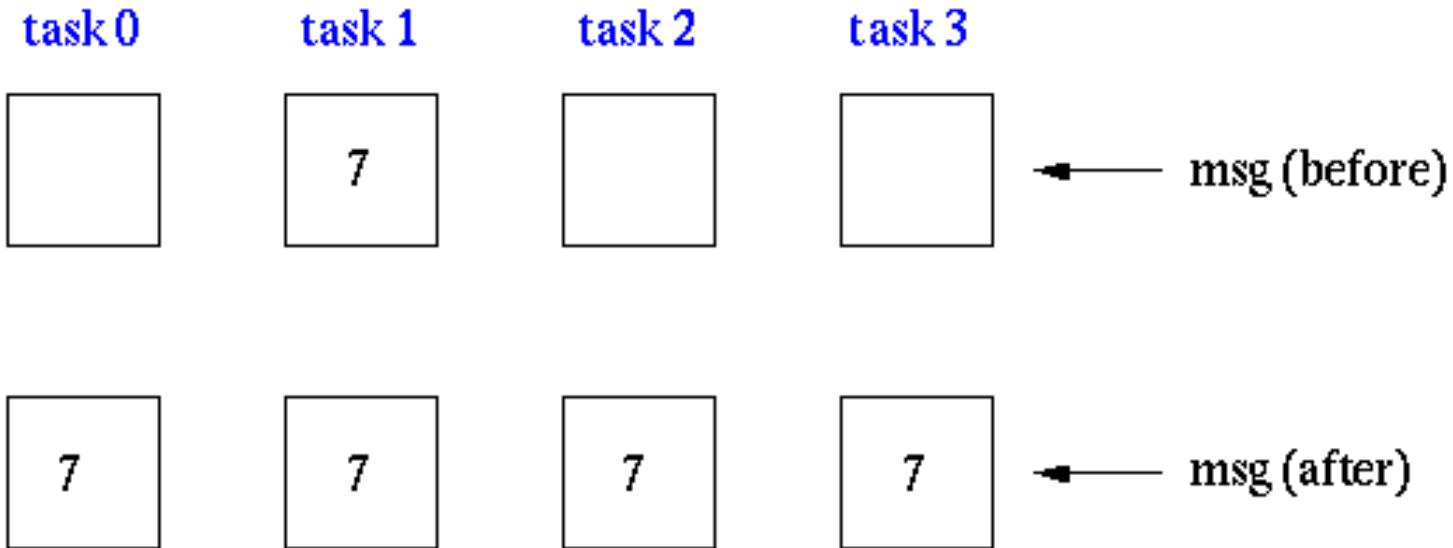


圖9-34

# MPI\_Reduce

Perform and associate reduction operation across all tasks in the group and place the result in one task

count = 1;

dest = 1;

result will be placed in task 1

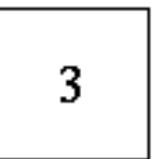
`MPI_Reduce(sendbuf, recvbuf, count, MPI_INT, MPI_SUM,  
 dest, MPI_COMM_WORLD);`

task 0

task 1

task 2

task 3



← sendbuf (before)



← recvbuf (after)

圖9-34

– 圖9-35展示了MapReduce的工作流程和Google的實作架構圖。

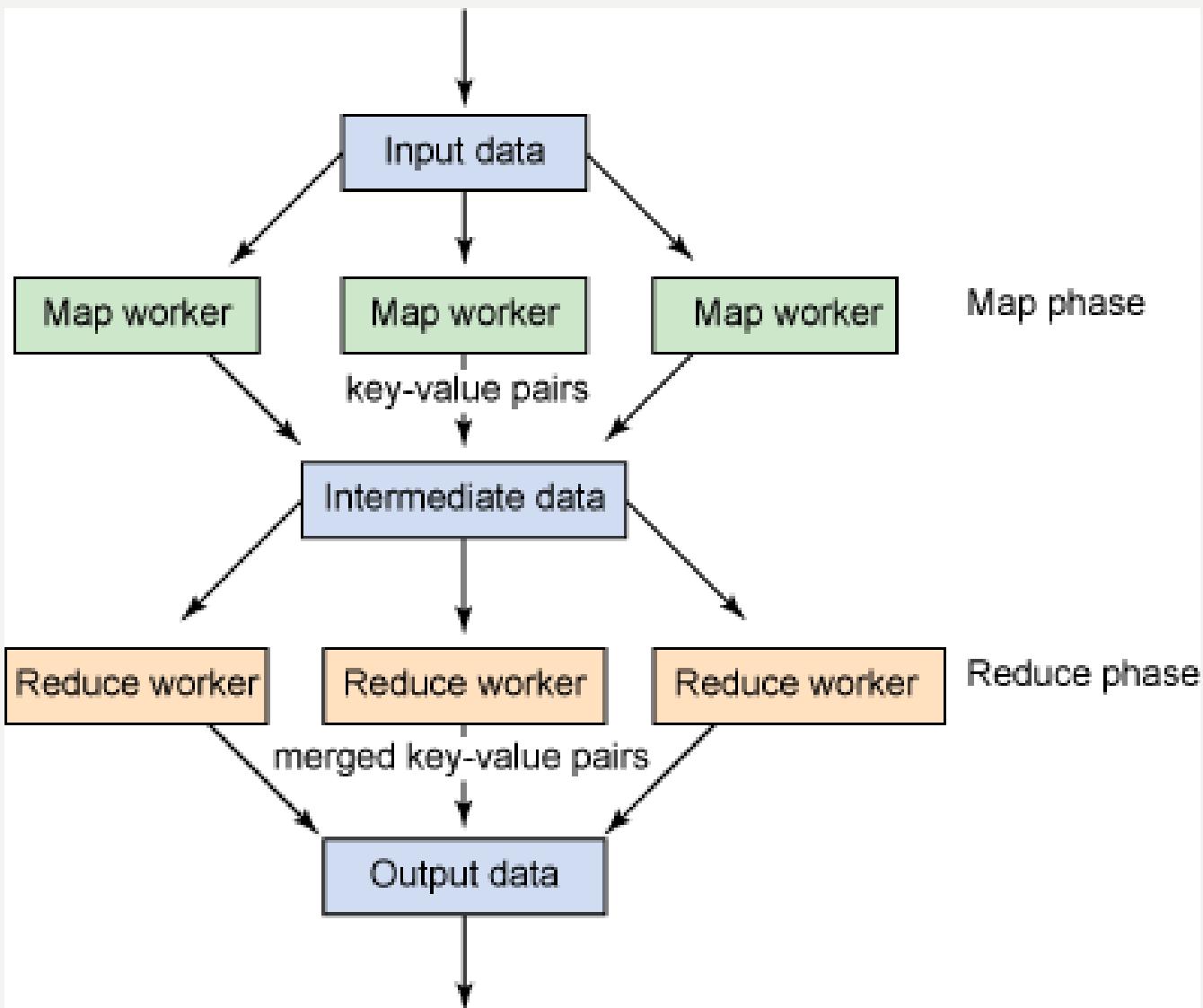


圖9-35

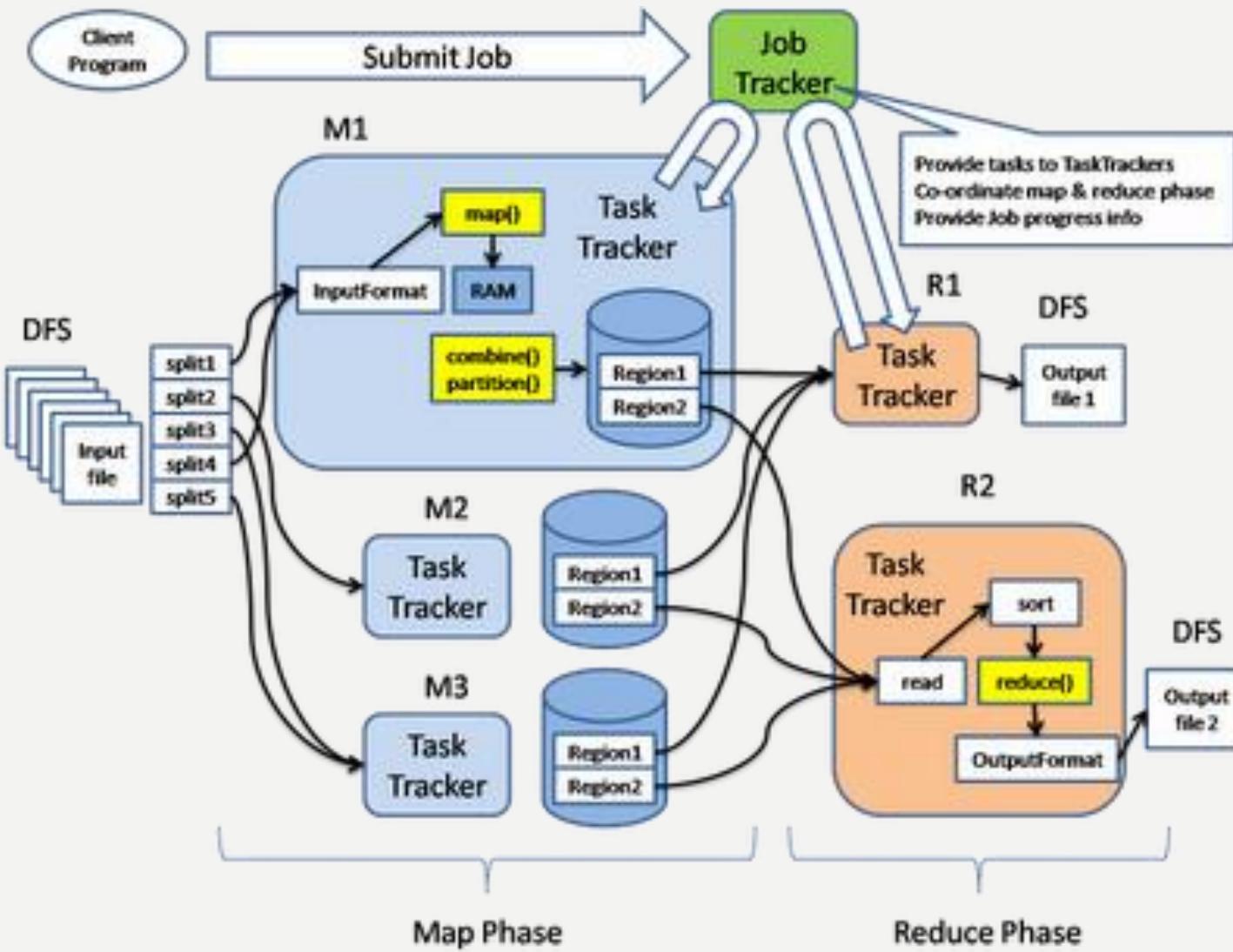


圖9-35

- Sawzall是一種實作MapReduce功能的解釋執行的Data-Parallel指令碼語言。由於Sawzall是類型安全的語言，本身處理了很多問題，所以完成相同功能的程式碼與MapReduce的C++程式碼相比簡化了大約10倍多。
- 資料庫技術：BigTable和Sharding。
- Google開發了一套資料庫系統，名為“**BigTable**”。

- BigTable也不可能是一個傳統的關係型資料庫，而是一種多級映射的資料結構，如圖9-36所示，具備高容錯性和自我管理等特性，使用結構化的檔來儲存資料，實作每秒可處理數百萬量級的讀 / 寫操作。
- BigTable乃基於GFS和Chubby。

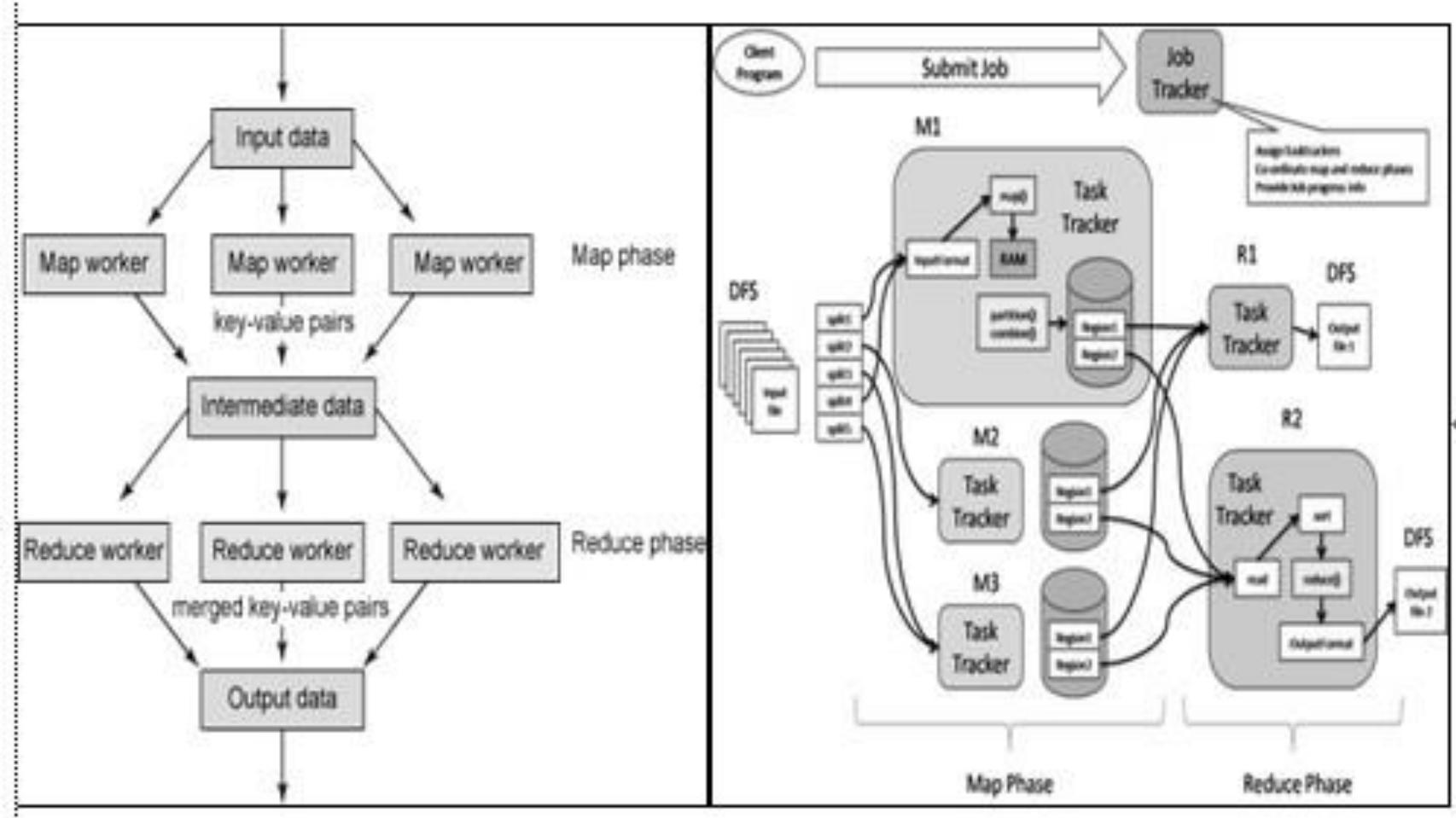


圖9-36

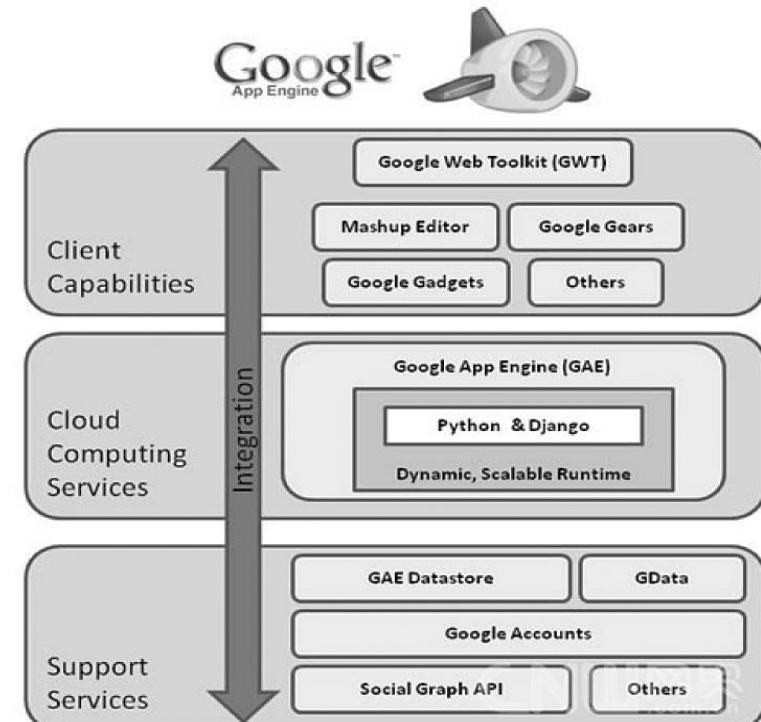
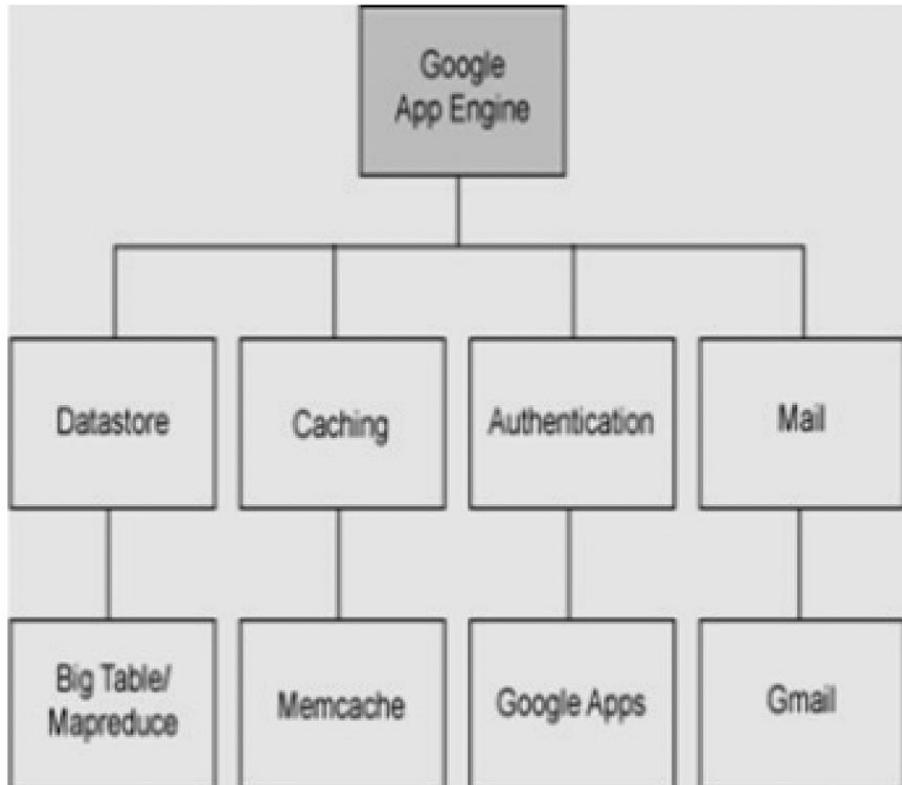


圖9-38

– App Engine是Google的PaaS，基於Python與Java技術，如圖9-39所示，執行在Google自己的IaaS平台之上。App Engine透過在JVM內執行Java版的其他語言來建立應用程式。

## Java standards

JSR-154

JSR-220 JSR-243

Java SE

JSR-913

JSR-107

Java Servlet

JDO / JPA

java.net.URL

javax.mail

javax.cache

Web App  
Container

Datastore  
API

URL Fetch

Mail API

Memcache

## Low level App Engine APIs

圖9-39

– App Engine支援如圖9-40所示的3種應用模式。對於App Engine開發人員而言，有免費帳號和付費帳號兩種不同的選擇。

## Announcing: Google App Engine for Business

Buy from Others



Google Apps  
Marketplace

Buy from Google



Google Apps  
for Business

Build your Own!



Google App Engine  
for Business

Google Apps Admin Platform



Enterprise Firewall



Enterprise Data Authentication



Services User Management



圖9-40

- **AppScale**是Google App Engine的一種開源實作，AppScale透過構建和擴充來自GAE的SDK並實作其開放API，它帶來了App Engine的強大功能，並補充了GAE的功能。
- 使用它可以簡化基於GAE應用程式在非Google的其他IaaS（如Amazon的EC2或Eucalyptus）上的執行。
- Appscale不僅可以在GAE應用程式部署到Google專有資源上之前預先測試和除錯這些應用程式，還能讓使用者用自己的機群系統執行App Engine應用程式。

- 9.3.3 Apache Hadoop 和開源 PaaS

- Apache Hadoop技術目前在雲端運算的實務中扮演著越來越重要的角色，Hadoop幾乎成了雲端運算的代名詞。利用Hadoop，對底層，可以實作對群集的控制和管理；對上層，可以更加便捷地構建企業級的應用。

- Hadoop起源於Nutch，由Lucene和Nutch的發明者Doug Cutting領銜的Yahoo團隊開發。
- 其目標是構建一個開源的網路搜尋引擎，在Google發佈GFS和MapReduce的論文後【33、72、73】，Cutting團隊發現GFS和MapReduce的設計準確地解決了Nutch系統中碰到的問題，於是這兩個系統在Nutch中被克隆（clone）函數重建，基於Java語言，形成了Hadoop。
- Hadoop包括基於GFS的檔案系統HDFS（Hadoop Distributed File System）、計算架構MapReduce及基於Bigtable原理的結構化資料處理HBase系統等，如圖9-44所示。

- 在Yahoo的幫助下，Hadoop很快成為確實可擴充應用於Web資料處理的技術，後來Hadoop成為一個Apache開源專案，出現了其他的一些開源專案（如Pig、Zookeeper、HIVE等【62】）
- 和很多使用者（包括Yahoo、Amazon、Facebook、Twitter、LinkedIn、eBay、AOL、IBM等20多家以網際網路業務為主的使用者），
- 但目前Yahoo仍然是Hadoop的最大使用者。Hadoop這個詞來自Cutting兒子的大象布玩具的名字。

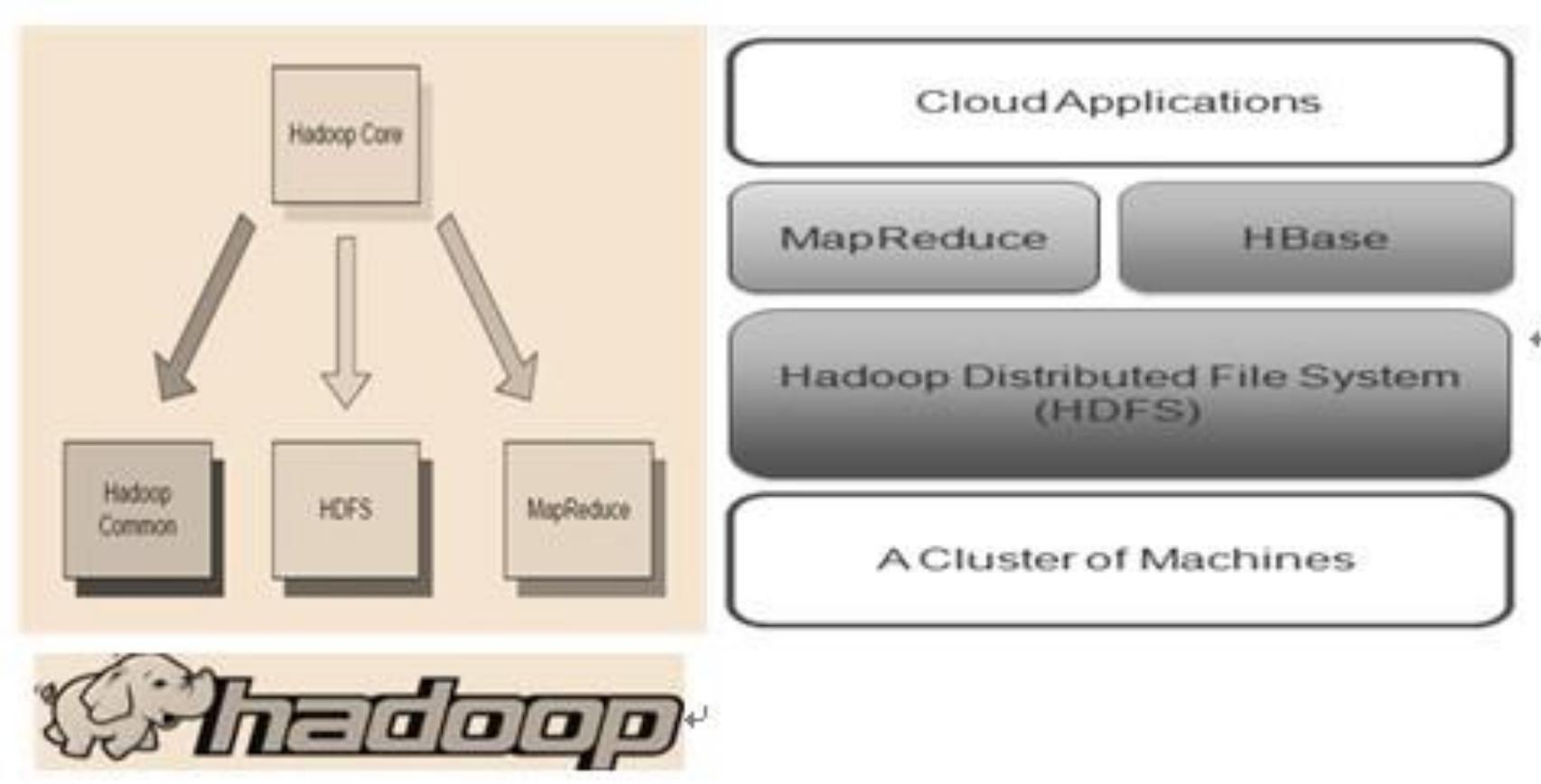


圖9-44

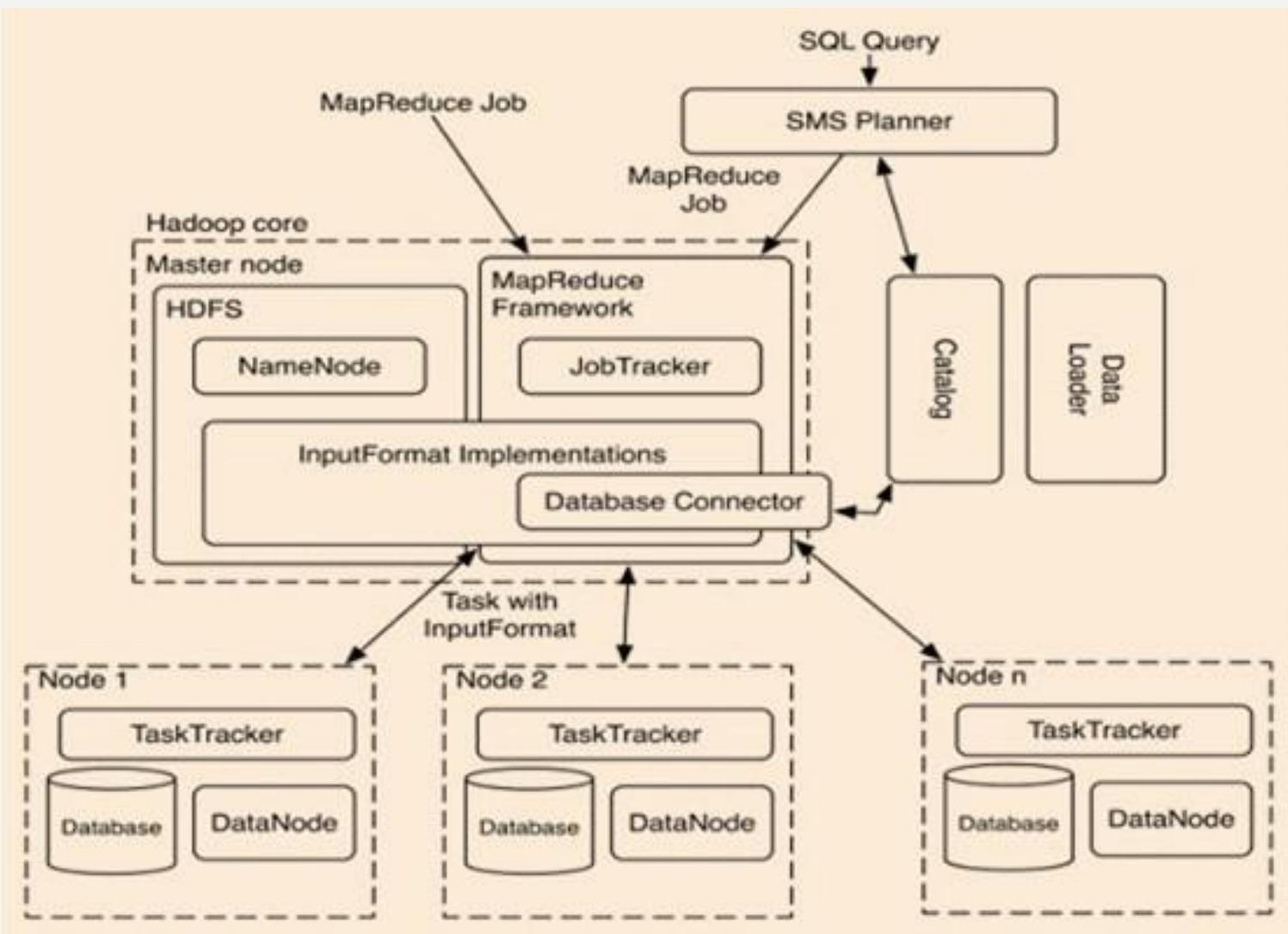


圖9-45

– Hadoop系統的處理節點還可以基於虛擬化技術實作節點的同構化，如圖9-46所示，便於普及應用，這在Google系統中是沒有的。

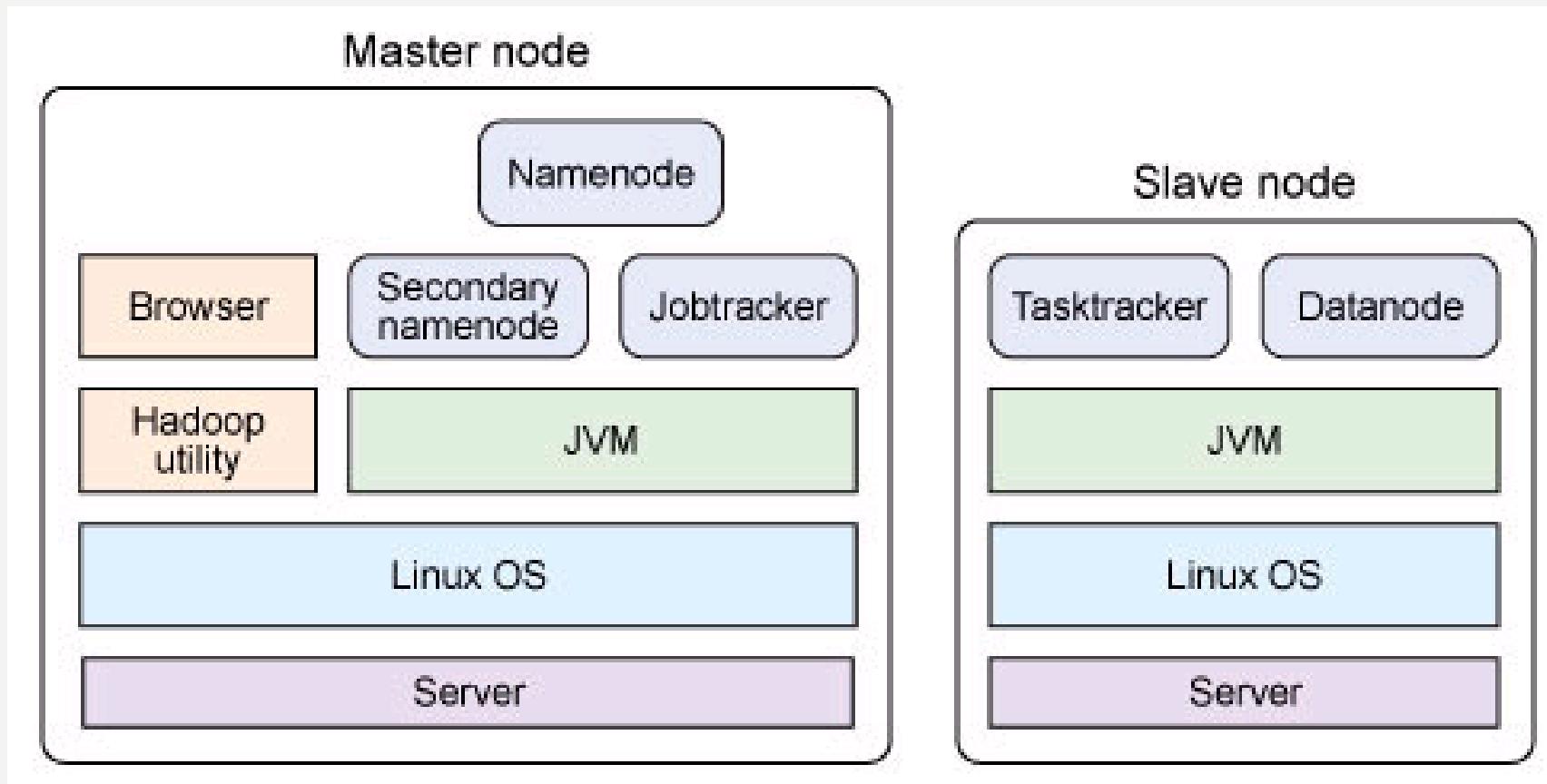


圖9-46

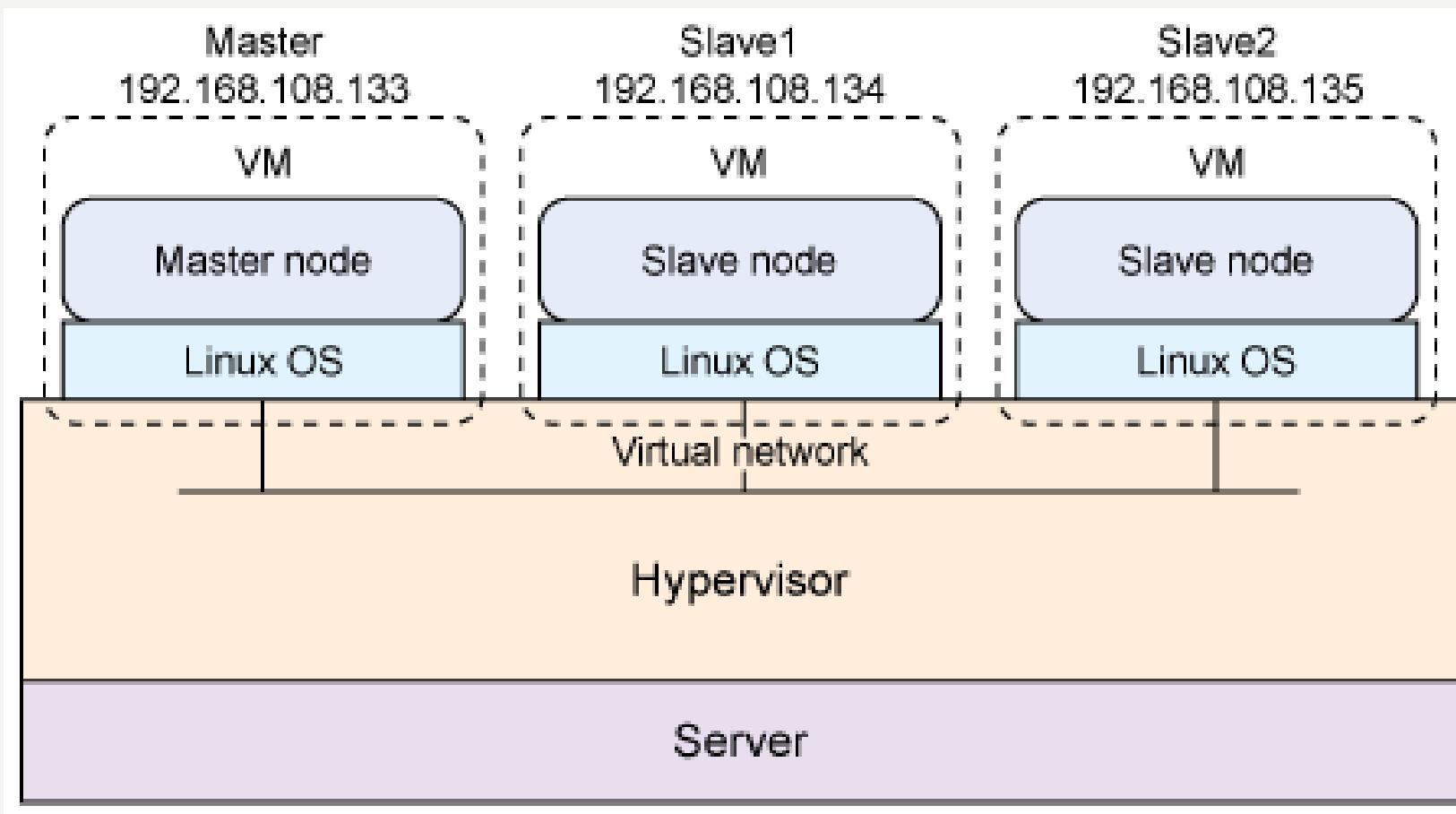


圖9-46(續)

- 提供相關API實作與其他技術的相容，例如，Hadoop可以和HDFS以外的許多檔案系統相容，包括提供新的檔案系統，如Facebook的Cassandra，一個相對全功能型的資料庫；可整合其他並行處理技術，如Condor、MPI等。這也屬於提高其通用性的手段之一。

- 事實上Hadoop技術就是MPI/Condor/HPFS等HPC技術的翻版，
- Hadoop技術體系是HPC技術體系的簡化版，
- 專門針對關聯度很小的大量資料的並行處理。
- 在Hadoop技術被應用到更廣泛的領域時，有可能會需要增加更多更複雜的功能，這時就需要和原有的HPC相結合。

圖9-48展示了包括Hadoop在內的適用於大資料量並行處理的功能模組的各種組合形成的雲端運算系統。

# Data-Parallel Computation

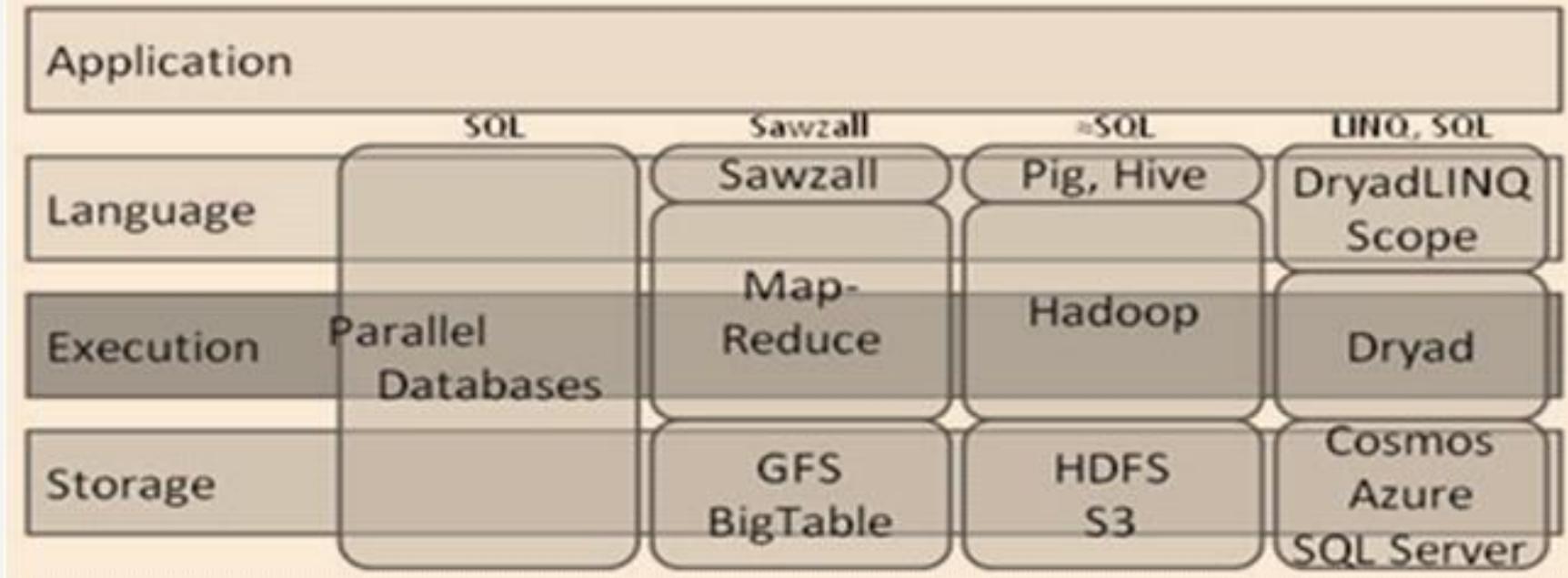
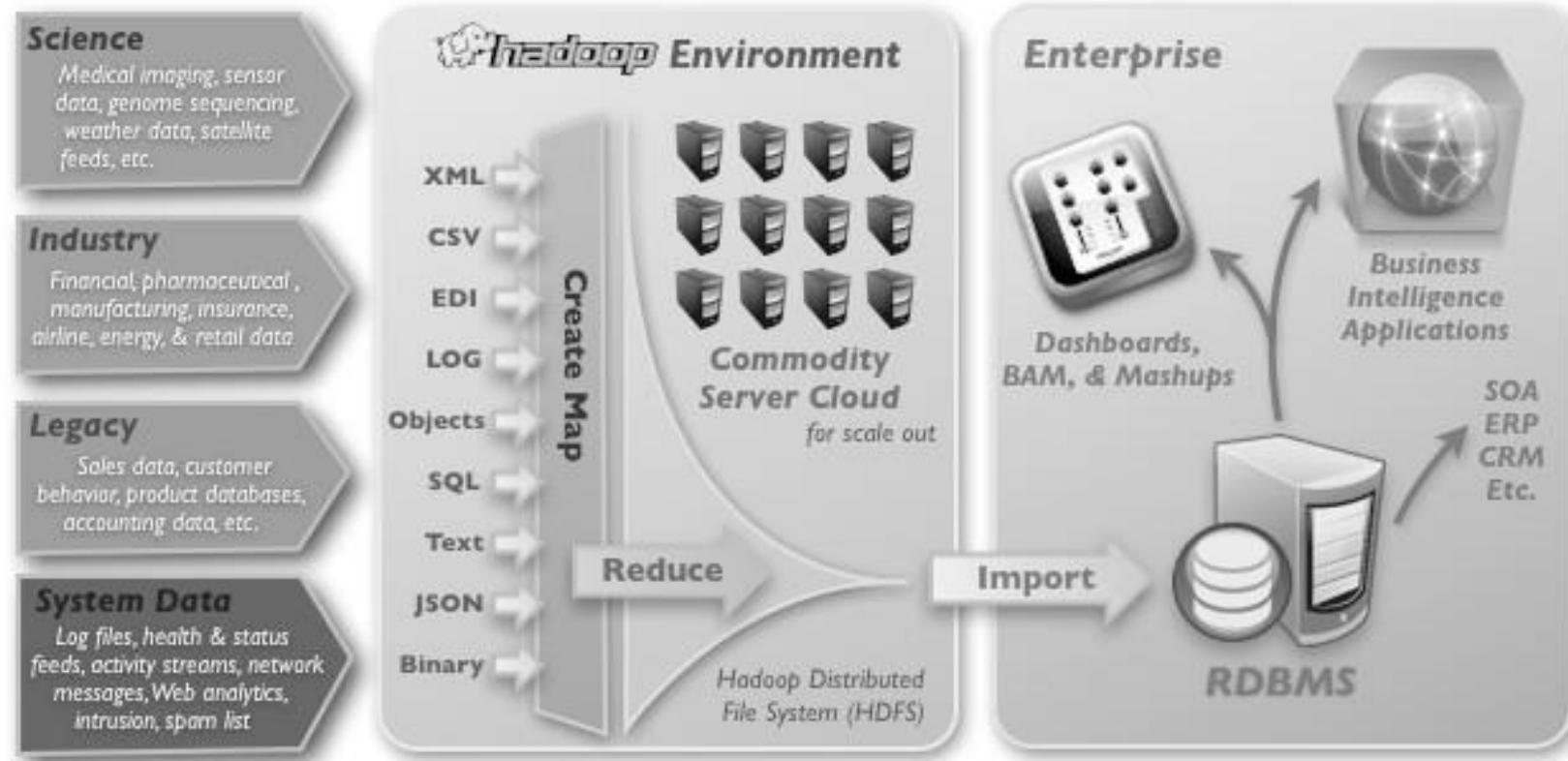


圖9-48

– 由於在Google、Yahoo、Facebook等業務中的成功使用，使得Hadoop技術受到了極大的關注，如圖9-49所示，其中有多人很可能不是很瞭解Hadoop技術的來龍去脈。

# Using Hadoop in the Enterprise



1 High Volume  
Data Flows

2 MapReduce Process

3 Consume Results

From <http://www.ebizq.net/blogs/enterprise>

- 其實Hadoop是一種應用面很窄的技術。
- VMWare收購的開源SpringSource中間層也是在為開展PaaS業務做準備。Redhat PaaS平台在收購了Makara以後變得更加完善。

- 在PaaS技術選型中，人們最關心的問題是可攜性（基於標準化），不希望被提供商鎖定（Lock-in），這就是如圖9-54所示的“開放的PaaS”的問題（開源並不完全等於開放），在這一點上Redhat PaaS比競爭對手VMWare的基於SpringSource的PaaS平台更加開放，有更大的優勢，VMWare的技術主要支援Spring框架。
- VMWare和Redhat，以及Hadoop等PaaS技術都是以Java技術為核心（Java代表開放）。

## 9.4 雲端運算系統全景圖

– Gartner依照SPI劃分對主流雲端運算企業做了如圖9-56所示的比較。

## Summary of Vendor Emphasis

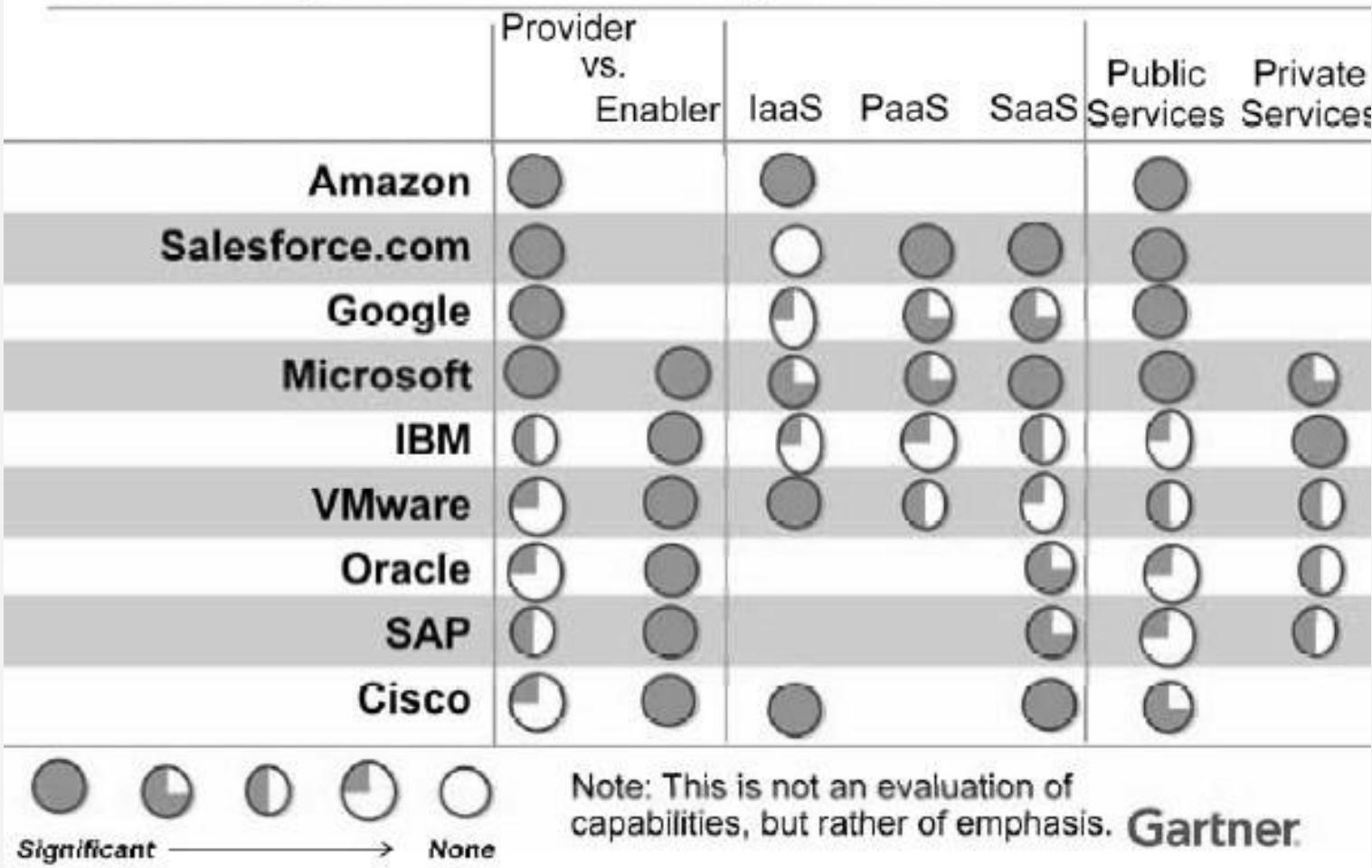


圖9-56

–也有一些劃分採取了更細的方式，包括雲端運算系統監控和管理，開發工具，把雲儲存單獨從IaaS裏劃分出來等等，如表9-1所示。

**表9-1**

科技分類	子分類	範例
1.基礎建設即服務(IaaS)平台		Amazon Web Services EC2, The Backspace Cloud, GoGrid
2.軟體即服務(SaaS)平台		Windows Azure, Google App Engine, Force.com

表9-1 (續)

科技分類	子分類	範例
3.雲端基礎建設服務		從雲端提供基礎IT服務
	• 儲存即服務	Nirvanix, Amazon S3
	• 災難復原即服務	SunGard Virtual Server Replication
	• 備份即服務	Iron Mountain Live Vault, i365 Evault, IBM Business Continuity and Resiliency Services

表9-1 (續)

科技分類	子分類	範例
4. 雲端應用服務		從雲端提供的應用程式服務
	• 資料庫即服務	Google Big Table, Amazon Simple DB, MS SQL Data Services
	• 雲端付費服務	Google Paymenet, Amazon DevPay, Zuora Zcommerce

表9-1 (續)

科技分類	子分類	範例
4. 雲端應用服務		從雲端提供的應用程式服務
	• 整合即服務	Amazon Simple Queuing Service, Boomi, CastIron, Informatica, Linxster, Online MQ, OpSource Connect, Pervasive
	• 商業流程管理即服務	Appian Anywhere, Intensil, Skemma

**表9-1 (續)**

科技分類	子分類	範例
5.雲端管理軟體		Appistry, CloudSwitch, Elastra, RightScale
6.雲端實驗室		Citrix C3 Lab, Electric Cloud, SkyTap, Surgient Cloud
7.桌面即服務		Desktone, MokaFive, Simtone

表9-5

技術類別	代表廠商 名稱	主要產品	功 能	說 明
SaaS，雲 端上的應 用軟體	Salesforce	Salesforce CRM	可用於銷售、服 務、營銷和呼叫 中心營運等。	幾乎是所有人都已 用過的（但是由 Salesforce首先提出 的）一種軟體使用 形式，如Gmail、 雅虎郵件、 wordpress.com、 Google Apps、不同 種類的搜尋引擎、 維基百科、百科全 書等。
	八百客	800APP CRM	以CRM為核心， 集進銷存、財務、 簡易OA於一體， 包括：合作夥伴 關係管理、線上 通用財務系統、 線上標準進銷存 系統等標準化模 組。	

– 表9-5所示為大陸的不完全總結，僅供參考。

表9-5 (續)

技術類別	代表廠商 名稱	主要產品	功 能	說 明
SaaS，雲 端上的應 用軟體	Netsuite	NetSuite OneWorld (線上ERP)	提供了多級水 準的綜合報表、 全球定單管理、 即時管理和自 動財務功能， 可降低成本， 簡化稅務管理。	一些企業將應用軟 體託管在網際網路上，使用者可以租 賃使用而無須考慮 這些資訊的出處、 計算週期和儲存位 元組的來源等因素。 售出的服務就是一 種完整的最終使用 者應用軟體。
	中企開源	企業管理、 電子商務、 行業門戶 等	對中小企業 “按需服務， 隨需而變”， 可以極大地降 低中小企業資 訊化的成本與 風險。	

表9-5 (續)

技術類別	代表廠商 名稱	主要產品	功 能	說 明
PaaS，雲 端上的平 台	Salesforce	Force.com	可構建和執行 任意應用程式， 按需提供企業 級Web應用程 式，而無須付 出部署基礎結 構的成本。	由Salesforce首創的 一種新興的形式， 即將應用軟體平台 提供給雲端上的研 發人員使用。研發 人員可以在雲端運 算平台上對應用軟 體進行編譯，然後 將程式碼再上傳到 雲端上。例如 Mosso、Google App Engine和 Force.com都是這種 雲端運算平台。
	Google	Google App Engine	可在Google 的 基礎架構上構 建並執行自己 的網路應用程 式，無須維護 伺服器。	

**表9-5 (續)**

技術類別	代表廠商 名稱	主要產品	功 能	說 明
PaaS，雲 端上的平 台	八百客	800APP	使用此開發平台，使 用者不需程式設計即 可開發包括CRM、 OA、HR、SCM、進 銷存管理等任何企業 管理軟體，而且無須 使用其他軟體發展工 具並立即線上執行。	

表9-5 (續)

技術類別	代表廠商名稱	主要產品	功 能	說 明
IaaS，雲端上的基礎架構	Amazon	EC(Elastic Compute Cloud)、S3(Simple Storage Service)	為企業提供計算和儲存服務，收費的服務專案包括儲存伺服器、頻寬、CPU資源及月租費。 案例：華盛頓郵報處理柯林頓夫人8年白宮生活資料。	由Amazon首創並由Right Scale提供管理平台的產品，也是最為全面的雲端運算產品。研發人員和系統管理員可從中取得全面的計算、儲存、陣列及其他資源，以最大的開放性來執行他們的應用軟體。

## 表9-5 (續)

技術類別	代表廠商名稱	主要產品	功 能	說 明
IaaS，雲端上的基礎架構	IBM公司	藍雲端 (Blue Cloud)	使計算不僅侷限在本地，透過架構一個分佈的、可全球存取的資源結構，使資料中心在網際網路環境下執行計算。	這也是功能最為強大的雲端運算類型，事實上任何適合放在網路上的應用軟體和配置都適用於這種服務。當然它也需要買家付出更多。
	微軟公司	Azure	包括了Windows Azure、Microsoft SQL資料庫服務，Microsoft .Net服務等主要元件，幫助開發可執行在雲端伺服器、資料中心、Web和PC上的應用程式。	

– 按硬體、虛擬主機（這兩者也屬於IaaS）和NIST的IaaS、PaaS和SaaS共5個層次，畫了一個國內外雲端運算產品和企業及其關係的“全景圖”，如圖9-59所示，這應該是目前為止最全的一個圖。

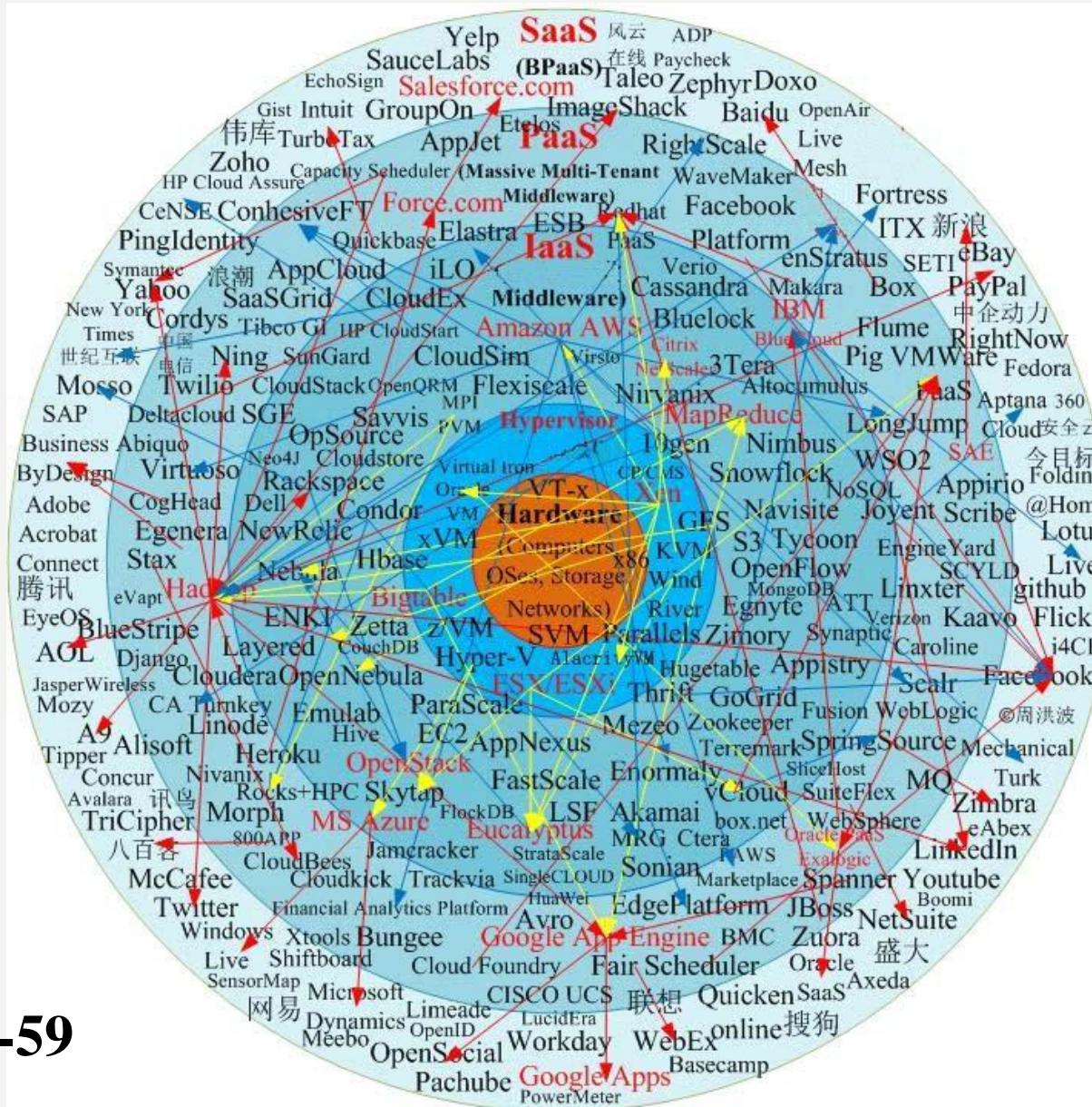


圖9-59

## 9.5 小結

– 本章依據NIST的雲端運算SPI劃分介紹了典型的雲端運算技術及其代表性廠商，介紹了貫穿IaaS和PaaS的基於SOA的中間層理念，也介紹了開源技術在雲端運算發展過程中的重要性及主要開源雲端運算系統。指出了Hadoop系統的侷限性及深度瞭解雲端運算的Grid/HPC淵源的重要性，和它在企業雲端運算系統選型過程中的指導作用。

# 12. 進入雲端運算的大門

- 本書前面提到三個主要的分類
  - IaaS
  - PaaS
  - SaaS
- 本章節藉由實際使用這三大類的主要服務廠商，讀者可更了解什麼是雲端運算以及優勢  
避免讀完本書後，仍然還在雲霧之中，  
先從最底層的IaaS開始介紹。

## 12.1 使用最著名的基礎建設即服務的**AMAZON**網路服務

- IaaS的主要概念為將豐富的硬體資源當成服務來提供。
- 遠端登入使用廠商所提供的伺服器，如Windows Server或是Linux等。
- 登入後可進行你的工作，因此為最彈性與自由的架構。
- 可自行決定與建構自己所要提供的網路服務，如網頁伺服器、電子郵件伺服器、資料庫伺服器等。

– 缺點就是大多需從架設與設定都需要自己動手來，除非是使用已經打包好的映像檔減少建立與設定的過程。

- 可先在瀏覽器中輸入IP位置，應該可連線至網頁伺服器。
- 當然在FTP或是其他網路服務依然使用此IP。
- 最後，指定你所申請的網域名稱對應此IP。
- 使用與管理上會更為便利。

- 12.1.4 Amazon 網路服務小結

- 從本章內容可以了解到Amazon EC2提供的彈性是無與倫比的。
- 自由的選擇所需要的平台、數量與計算能力。
- 並可自訂自己的需求架設站台，例如網頁服務、FTP、IP、資料庫等服務。
- 當然，在不同的平台也需要伺服器的管理技巧與設定。
- 與下一個階段介紹的PaaS差異點就在於此。

## 12.3 一定要會用的**SAAS**雲端服務：**DROPBOX**雲端儲存

- SaaS指的就是軟體以服務的方式來提供。
- 過去軟體採購要花費一筆固定金額，但SaaS依使用次數或人數來計費。
- 較有名的例子如Salesforce.com以租賃方式提供客戶使用顧客關係管理(Customer Relationship Management, CRM)的系統。
- 本章節介紹Dropbox在雲端與多台電腦間同時同步化檔案並進行備份。

- Dropbox好處是依照使用量付費，前2GB容量免費。
- Dropbox會是最易於上手的雲端儲存服務。
- 除了更能了解SaaS的運作，對於儲存重要檔案（如碩博士生的論文）有相當大的助益！

## 12.4 本章總結

- 可了解基礎建設即服務提供相當大的彈性。
- 平台即服務能提供特定語言在該平台上運作。
- 軟體即服務讓使用者可以用很便宜的成本使用雲端服務。

- 雲端運算這個術語中用了“雲端”這個字，比Mainframe和Web這兩個詞更抽象和“虛無縹緲”，
- 一般人很難把它與電腦或IT聯繫起來，業內人士聽起來也有些讓人“不知所云”。這個名字的好處是，由於它的通俗化和人文化，可以讓大眾更容易接受。
- 其實，網格電腦已經是一個很好地能夠表達雲端運算所有含義的辭彙了，網格這個詞既有網路，也有格局，乃至所有種類的網路的意思。

## -雲端運算六大要素的總結：

- 一個金字塔形架構：從硬體基礎設施，到SaaS服務介面及無所不在的行動雲端終端，全都實作標準化的無縫整合和整合，構成ICT的通天塔。
- 二方面的融合：單機虛擬化與多機虛擬化技術打造的HPC和網格運算設施與面向服務的Web Services、SOA、SaaS、效用計算等技術建立的無所不在的服務設施之間的融合。

- 三層SPI架構：SaaS面對最終使用者提供服務，PaaS中間層承上啟下整合軟體資源，IaaS面對底層設施整合硬體資源；三層按需動態服務模式提供多種業務模式，是以盈利為目的的企業商業模式的基礎和支援。

- 四種部署方式：私有雲端一般存在於內網和專網中，為單一機構服務；社區雲端一般存在雲端內網和專網以及VPN網路中，為多家關聯機構服務；共有雲端面向網際網路，服務於大眾；混合雲端是上述兩種或三種雲端的組合。
- 五大關鍵功能：按需自助服務；普及的網路存取方式；動態的資源池；快速可伸縮性；可計量的服務；五大功能的組合是實作雲端運算ICT資源“自來水”式服務理想的基礎。

- 六大基本特性：可自修復能力、透過單機虛擬化實作的同質性、廣泛的地理分佈、服務導向的SOA架構、低費用和先進的安全措施；六大特徵是雲端運算贏得使用者廣泛的認同和接受的優勢所在。