

第二章 文獻探討

第一節 Web Services 技術

根據 Curbera[3]等人在文獻中提到關於 Web Services 系列的標準主要包括了 XML、SOAP、WSDL 以及 UDDI 四種開放性的標準。透過這些標準，應用程式可以利用 Web Service 在分散式的網路環境之中進行溝通，改變了以往人與應用程式溝通的模式，轉換成應用程式與應用程式的溝通模式。例如，一個程式師使用最好的程式開發工具，輕易地開發出一個新的入口網站應用程式（Portal Application），然後將此程式視為：現有、可再使用的服務，並將它運用於 ERP 應用程式中，同時也運用到另一個企業自行開發的 J2EE 應用程式中，再進行運用，原因在於不必理解被引用的程式其內部是如何運作。

而根據 W3C 對 Service 的定義[17]，一個 Service 會使用 WSDL 定義根據此定義可以了解 Web Service 有兩大優點：

- Web Services 可以用來建構分散式架構系統，實現分散式架構動態整合、平衡存取負擔、單元升級等優點。
- 採用開放式標準讓 Web Services 具有良好互通性，在不同平台上用不同程式語言建置的系統也透過使用 Web Services 輕易整合，克服目前分散式系統各自使用不同機制造成整合困難的情形。

下面我們將繼續對 Web Services 相關標準分別探討 XML、SOAP、WSDL、UDDI 等四種標準。

2.1.1 XML(Extensible Markup Language)

Doug Tidwell[7]在文獻中對 XML 的描述，可擴充標記語言（XML, Extensible Markup Language）是一種可以由使用者自己建立標籤符號的標記語言。由 World Wide Web Consortium（W3C）建立標準，並克服了 HTML（Hypertext Markup Language）的缺點與侷限。

XML 規格書中對制定設計 XML 文件有著十大目標如下[22]:

- XML 的內容必須簡潔，能夠降低剖析器的負擔，容易應用於網路上。
- XML 與 SGML 必須相容。
- 處理 XML 的程式應該很容易夠被撰寫。
- 必須把選擇性的功能保持在最少，最好是 0。
- 文件很容易被建立。
- 讓人容易被閱讀。
- 標記須嚴格，起始與結束都必須被嚴格標記出來，明確的標記筆簡捷更重要。
- XML 應該被應用程式所接受，作為資料分享的方法，而不僅僅是在 Web 應用。
- 設計必須很正式且正確。
- XML 文件能夠很迅速推出，避免被單一廠商所壟斷。

舉個例子一個簡單的 XML 檔案例子可以如【表 2】所示:BookData 是 root，檔案中包含兩筆資料 Book，第一筆資料的名稱是 Open eyes、價格是 400、作者是 C.K.；第二筆資料的名稱是 Sister、電話 350、作者是 L.L.。

【表 2】XML 範例

```
<?xml version="1.0" encoding="utf-8"?>
<BookData>
  <Book>
    <name> Open eyes </name>
    <charge>400</charge>
    <wright>C.K.</wright>
  </Book>
  <Book>
    <Book>Sister</Book>
    <charge >350</charge>
    <wright>L.L.</wright>
  <person>
  </BookData>
```

2.1.2 SOAP(Simple Object Access Protocol)

在 2001 年 4 月時，W3C 聯盟召開了 Web Service 的專題研討會，其目的是為了探討 Web Service 架構標準化實作的發展與方向。在討論中提出了 SOAP(Simple Object Access Protocol)在 Web Service 架構中所扮演的角色，主要在傳遞訊息功能方面。W3C

所公佈之 SOAP 標準包含下列幾項特性[10]:

- SOAP 為輕量化之通訊協定。
- 應用於程式與平台之間的溝通。
- 不依賴任何程式語言。
- 不依賴任何技術元件。
- 是以 HTTP 當作底層的通訊協定。
- 以 XML 當作基礎。
- 簡單且有彈性。



SOAP 協定主要作用是將訊息或資訊以 XML 格式封裝起來，可以經由 HTTP 協定進行傳送。SOAP 格式是加上了定義 HTTP Header 以及 XML 內文的檔案，並利用分散式傳輸，可以很輕易的由本機端的程式可以呼叫另一部電腦上的程式。它也定義了被呼叫的程式要如何回應，使訊息可以跨平台傳輸並穿過防火牆成為可能，SOAP 是利用所謂的「訊息」（Message）為溝通的基本單位。而一個標準的 SOAP 訊息，就是一個標準的 XML 文件，並包含了以下幾個基本元素[9]：

1. 一個 SOAP 封包（Envelop），用來定義整個 SOAP 訊息的內容。
2. 一個 SOAP 訊息標頭紀錄（Header），此部份為選用，包含了所有標頭所應登錄的資料。
3. 一個 SOAP 訊息主體（Body），其中包括了所有的呼叫描述與回應內。結構如

【圖 1】所示：



SOAP 訊息結構圖

【圖 1】SOAP 訊息結構圖

資料來源：(INT Media Group, Incorporated)

2.1.3 WSDL(Web Service Description Language)

WSDL 是以 XML 格式化去說明描述 Web Services 的標準，由 Ariba、Intel、IBM 和微軟等開發商提出，用來定義與描述 Web Services 詳細資訊。它定義了 Web 服務收發的有關操作和消息。

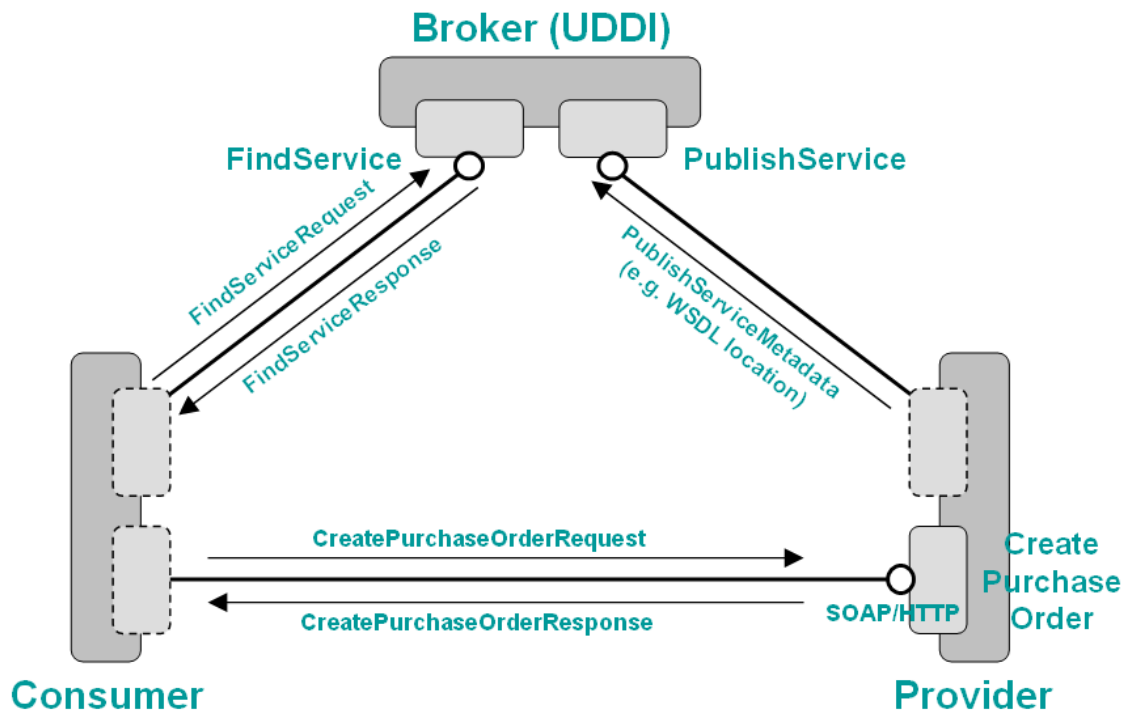
根據 W3C 的規範說明[6]，WSDL 是以 XML 格式用來描述服務以及服務傳遞之訊息資訊，在一個 WSDL 檔中需要包含下列幾個部份：

- types：定義使用的資料型態。
- Message：為抽象型別，定義通訊的資料訊息名稱與資料型態(types)。
- Port Type：為 Operation 的集合，定義此服務提供哪些 Operation。
- Binding：定義將 Port Type 跑在何種通訊協定之上。
- Port：指定 Binding 需要的服務通訊之端點位址。
- Service：為多個 Port 的集合，可指定如何存取服務之資訊。

前四項用來描述 Web Services 的介面，後兩項說明如何執行由服務介面，所提供的服務。目前能夠實作 Web Service 的平台，幾乎都有能夠產生 WSDL 的功能，主要描述在 Web Service 的存取介面，其功能是為了讓需要的後端應用程式可以做連結。

2.1.4 UDDI (Universal Description Discovery and Integration)

UDDI 的標準主要是由 OASIS TC 所主導[11]，其最主要的功能是讓 Web Service 功能的提供者，藉由 UDDI 平台登入所提供的服務，而 Web Service 使用者經 UDDI 平台查詢找到需要的服務在哪邊可以取得，類似我們在生活中可以使用索引找到想要查詢的資訊。其在 Web Service 中所扮演的角色如【圖 2】所示：



【圖2】 UDDI 在 Web Service 扮演之角色

(資料來源：<http://www.w3.org/2004/10/presentations/clus.ppt>)

服務提供者可以將自己提供的服務登錄於一個集中式的 UDDI Server 上，當服務的使用者想要使用網路服務的時候，可以要求 UDDI Server 提出所需要的服務。UDDI Server 收到後回覆服務提供者的 URI，接著服務使用者才向服務提供者要求服務，UDDI Server 讓服務使用者可以找到提供服務的地方。UDDI 版本 1 標明了註冊中心的

基礎，版本 2 則加入了企業關係等功能，版本 3 解決正在進行的 Web 服務開發中的重要性問題，如安全性、國際性等問題，功能目前仍然持續發展中。

第二節 SOCKET 通訊技術

Socket 通訊技術的架構可區分成 JAVA 與 Windows Based 兩大架構，本研究將會採用 Windows Based 的架構。Stardust Technologies 公司在 InterOP 展覽會舉行了一個非正式的會議，為了提供更好的基本網路服務，希望能夠在 Microsoft Windows 上建立一個 TCP/IP 程式的標準介面。並以 Berkeley Sockets 為基礎定義一組 API 來開發網路程式，並整合了 Windows 訊息架構的 API。於 1992，此小組分別發表了 1.0、的 Windows Sockets API 規格[13]。

龔旭陽在文獻中對 Windows Sockets 作了下列定義[28]:

- WinSock 是 Microsoft Windows 下的開放式網路應用程式介面，由許多網路軟體廠商合作發展出來的一套標準規格。
- WinSock 軟體的工作就是提供一個好的使用者介面以及處理資料。
- WinSock 協定層的工作是利用標準傳輸協定、驅動程式、網路媒介來收送資料。

Socket 可以代表一個簡化網路通訊複雜度的機制，程式設計師可以利用此方式來撰寫應用程式，來對網路上的伺服器或是對等的客戶端程式進行連結，它是一個網路通訊中最低階的程式介面。

第三節 Context-Aware 技術

Schilit 與 Theimer 於 1994 年提出 (Context-Aware) 這個概念[2]，在一開始對 Context 的定義認為:地理位置、周圍的人及物件的改變狀態。而在 1997 年時，Brown, P.J.與 Bovey, J.D..等則將 Context 定義為地點、人及物件、時間、溫度狀態的改變[14]。如：

- 空間改變：高度、移動速度..等
- 時間改變：時間、季節..等

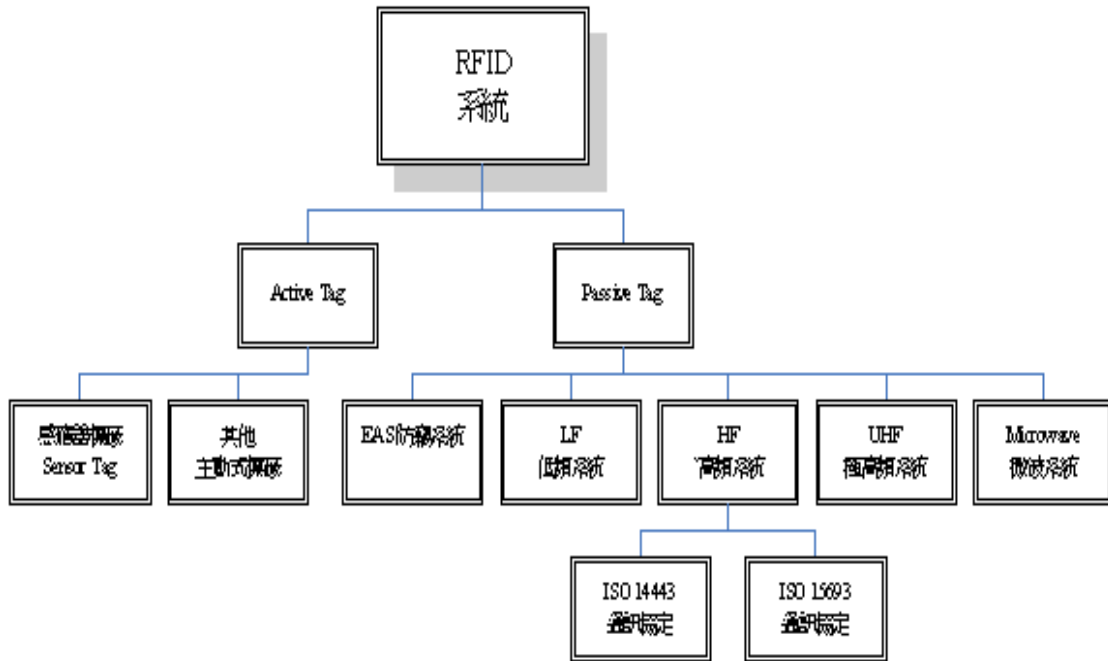
- 人文環境的改變：身份、目前環境噪音、交通..等
- 自然環境的改變：溫度、溼度、壓力資訊..等

Context-Aware System 是一種應用系統，該系統使用 Context 向使用者主動提供與使用者與相關的資訊和服務；如何擷取並且處理所收集到情境的資料如，將是未來一個主要的研究課題。

第四節 RFID(Radio Frequency Identification)無線射頻辨識技術

RFID 是指由微小的基體電路(Integrated Circuit)組件包括了無線通信與天線所構成的通稱；使用具有天線及讀寫器將資料寫入微小的基體電路晶片，或是讀取已有的資料。由於是以無線方式進行通信，因此被稱為 RFID，加上 ID 這個字，是因為所以 RFID 都具有特定的識別碼[29]。

RFID 之 Tag 分成主動式之 Tag 與被動式之 Tag 兩大種類:主動式 Tag 具備內部電力，缺點是體積大必須定期更換電池且價格較昂貴，間隔一段期間就需更換電池，但是可提供較遠的讀取距離。[30]被動式的 Tag 是藉由外部電磁感應產生電力來傳送資料，和主動式優缺點相反，內部不需要電力提供，但是傳輸距離相較於主動 Tag 為短。主要細分如【圖 3】所示:



【圖 3】RFID Tag 分類圖

(資料來源: Global Digital Data Synchronization Network (GSDN)網站)

在工業技術研究院 RFID 中心網站核心技術報告[23]裡提到 RFID 被列為本世紀十大重要技術項目之一，經濟部技術處為國內科技研發推動之火車頭，92 年度起即開始透過工研院系統中心推動高頻 RFID 的研發計畫，研發內容包括 IC 晶片、天線、讀取機(Reader)等重要技術，本年度將完成 IC 晶片開發，預計明年上半年將有國產之高頻 RFID Tag 進軍市場；後續則將研發讀取機以及 RFID 與其他感應器(Sensor)結合之研究計畫，以使 RFID 能多樣化地應用在各方面。

RFID 之使用頻段依照使用程度主要可以分成三類:

- 13.56MHz 的最佳傳輸距離為 1 公尺以下，主要應用於生產管理、會員卡、識別證和貴重物品的進出入管理。
- 860M~960MHz 最遠可達近 10 公尺的傳輸距離，通訊品質佳，適合供應鏈管理。
- 2.45GHz 的最佳傳輸距離為 100 公尺，穿透性較差，適合電子收費系統 (ETC, Electronic Toll Collection)、及時定位系統 (RTLS, Real-Time Locating System)。

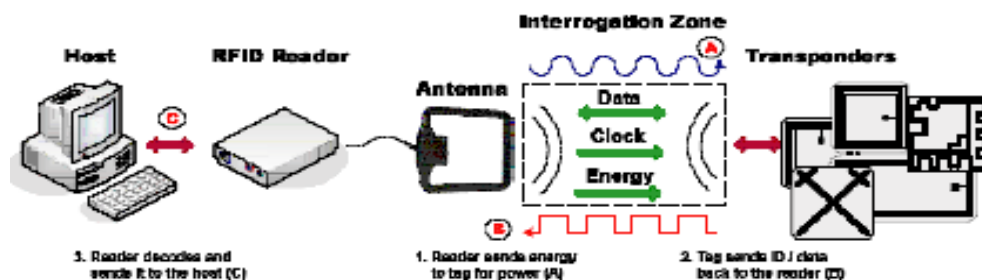
2.4.1 RFID 應用與發展

RFID 原始的發展目的是以軍事用途為主，目前已開放作為加速商業流程以及辨識功能使用。尤其在物流上的應用，將使物流的追蹤更即時，對產業供應鏈產生巨大的影響。RFID 未來可能取代條碼，作為物品辨識的新科技，將能更細緻、更迅速達到物品辨識的要求。RFID 之應用層面包括[24]:

- 門禁管制：人員出入門禁監控。
- 貨物管理：存貨、物流運輸管理。
- 醫療應用：醫院的病歷系統、危險或管制之生化學品管理。
- 交通運輸：高速公路的收費系統，貨物監控。
- 防盜應用：商店之防盜管理。
- 動物監控：動物生態的識別、追蹤。
- 自動控制：電子業之組裝生產。
- 聯合票證：聯合多種用途的智能型卡片。

2.4.2 RFID 資訊系統

Yi-Shyuan WU(2006)在 LANDMARC : Indoor Location Sensing Using Active RFID 文章[18]中提到 RFID 資訊系統的組成包括:Reader、Tag(Transponder)、Host (主電腦應用系統)、antenna。如【圖 4】所示:



【圖 4】RFID 資訊系統圖

RFID 資訊系統的運作模式如下:由 RFID tag 將資訊傳送給 RFID reader，RFID reader 將收集到資訊透過有線或無線的方式對 Host 端的電腦做傳送，以便進行各種不同的資訊應用。

第五節 全球衛星定位系統(Global Position System, GPS)

全球衛星定位系統(Global Position System, GPS)[5]是由美國國防部(Depart of Defense, DOD)自 1970 年所開發的系統，它利用三點定位技術三角測量原理去提供使用者包括時間、速度、高度以及經緯度等精確的定位資訊，目前廣泛應用於我們的日常生活中。GPS 原本使用於軍事定位、定時等目的，目前運用 24 顆衛星平均分佈在 6 個軌道面，可確保在世界上任何時間任何地點皆可同時觀測到 4 至 7 顆衛星。目前已被廣泛應用於基本控制點量測及其他各種導航、地形圖量測等作業。

2.5.1 GPS 訊息

我們可以從 GPS35LP Series Technical Specification[7]得知我們可以從 GPS 訊號接收器取得資訊如【表 3】所示：

【表 3】GPS 所擷取到之訊息實例

\$GPGSA,A,3,01,,06,14,,,,,22,25,,,,,2.0,2.0,1.0*36
\$GPGSV,3,1,10,01,46,335,33,05,15,040,,06,32,112,37,14,68,060,43*72
\$GPGSV,3,2,10,16,30,218,29,18,03,160,,20,02,323,,22,36,184,38*71
\$GPGSV,3,3,10,25,42,312,35,30,35,039,,,,,,*70
\$GPRMC,141134,A,2501.6573,N,12131.7406,E,001.8,006.4,010406,003.4, W*6D
\$GPGGA,141134,2501.6573,N,12131.7406,E,6,05,2.0,31.6,M,17.0,M,,*76

表中各標籤代表意義如下述，GPGGA(Global Positioning System Fix Data)傳遞修正位置座標的資訊，GPGSA(GPS DOP and Active Satellites)可得知 GPS 系統是否有正常與衛星通訊以及位置定位之精準度，GPGSV(GPS Satellites in View)中描述目前定位使用到運作中之衛星，GPRMC(Recommended Minimum Specific GPS/TRANSIT Data)中描述目前 GPS 所在之經緯度，物體移動速度等資訊。在本研究中所使用之主要資訊為 GPGGA 格式與 GPGGA 格式兩種訊息來源，接下來我們更詳細的解讀該資訊。如【表 4】與【表 5】所示：

【表 4】GPGGA 訊息所表之意義

hhmmss.dd	UTC time of the fix. 此次定位的時刻（世界協調時） hh = hours. 時 mm = minutes. 分 ss = seconds. 秒 dd = decimal part of seconds. 秒的小數點
xxmm.dddd	Latitude coordinate. 緯度座標 xx = degrees. 度 mm = minutes. 分 dddd = decimal part of minutes. 分的小數位
<N/S>	Character denoting either N = North or S = South. 緯度符號 N = 北緯，S = 南緯
yyymm.dddd	Longitude coordinate. 經度座標 yyy = degrees. 度 mm = minutes. 分 dddd = decimal part of minutes. 分的小數位
<E/W>	Character denoting either E = East or W = West. 經度符號 E = 東經，W = 西經
v	Fix valid indicator 定位指示 0 = Fix not valid 無效定位 1 = Fix is valid 有效定位
ss	Number of satellites used in position fix, 00-12. Notice: Fixed length field of two letters. 使用於定位的衛星數
d.d	HDOP – Horizontal Dilution Of Precision. 水平稀釋精度
h.h	Altitude (mean-sea-level, geoid) 高度
M	Letter M. 單位 公尺
g.g	Difference between the WGS-84 reference ellipsoid surface and the mean-sea-level altitude. 高度與橢圓曲面之差
M	Letter M. 單位 公尺
a.a	NULL (not implemented) 保留
xxxx	NULL (not implemented) 保留
*hh	Check sum 檢查碼
<CR><LF>	End 結束符號

範例：\$GPGGA,084053.39,6016.3051,N,02458.3735,E,0,00,0.0,46.6,M,18.2,M,,*5D

【表 5】GPRMC 訊息所表之意義

hhmmss.dd	UTC time of the fix. 此次定位的時刻（世界協調時） hh = hours. 時 mm = minutes. 分 ss = seconds. 秒 dd = decimal part of seconds. 秒的小數點
S	Status indicator 狀態指示 A = valid 資訊可用 V = invalid 資訊不可用
xxmm.dddd	Latitude coordinate. 緯度座標 xx = degrees. 度 mm = minutes. 分 dddd = decimal part of minutes. 分的小數位
<N/S>	Character denoting either N = North or S = South. 緯度符號 N = 北緯，S = 南緯
yyymm.dddd	Longitude coordinate. 經度座標 yyy = degrees. 度 mm = minutes. 分 dddd = decimal part of minutes. 分的小數位
<E/W>	Character denoting either E = East or W = West. 經度符號 E = 東經，W = 西經
s.s	Speed in knots. 速度是幾節
h.h	Heading. 艏向
ddmmyy	UTC Date of the fix. 此次定位的日期（世界協調時） dd = day of month 日 mm = month 月 yy = year 年
d.d	Magnetic variation in degrees, i.e. difference between geometrical and magnetic north direction. 磁偏差
<E W>	Letter denoting direction of magnetic variation. Either E = East or W = West. 磁偏差單位 E：偏東 W：偏西
M	Mode indicator 模式指示 A=autonomous 自動 N=data not valid 資料不可用
*hh	Check sum 檢查碼
<CR><LF>	End 結束符號

範例：\$GPRMC,095035.91,A,6016.3066,N,02458.3832,E,1.08,210.6,131204,6.1,E,A*0A

本研究將會採用GPGGA的Geoidal海拔高度資訊與GPRMC中的標準定位時間、經緯度資訊、相對位移速度以及相對位移方向五種GPS訊息，當作GPS的資訊來源。



第六節 開發所需之工具

本研究系統實作的過程中，需要之開發環境將會在此小節進行說明，主要開發環境使用 Visual Basic 2005 程式語言，運用 ASP 以及 Visual Studio.NET 伺服器發佈服務或是網站。另外 .NET Framework 在本研究中扮演了相當重要的角色，內含連結功能強大函式庫提供來操作 XML Web Services 之 API、或是利用 DOM 方法來對 XML 文件當作物件進行操作等重要功能，到執行當作 ASP 網頁伺服器、執行 Web Services 的伺服器等重要功能。

2.6.1 .NET Framework

.NET Framework 是微軟所提出的一新的應用程式開發環境平台，其發展方向是要建構一個使用網際網路開放標準的應用程式開發與執行的環境，使任何人從任何地方，在任何時間使用任何裝置都能存取網際網路上得資源與服務。

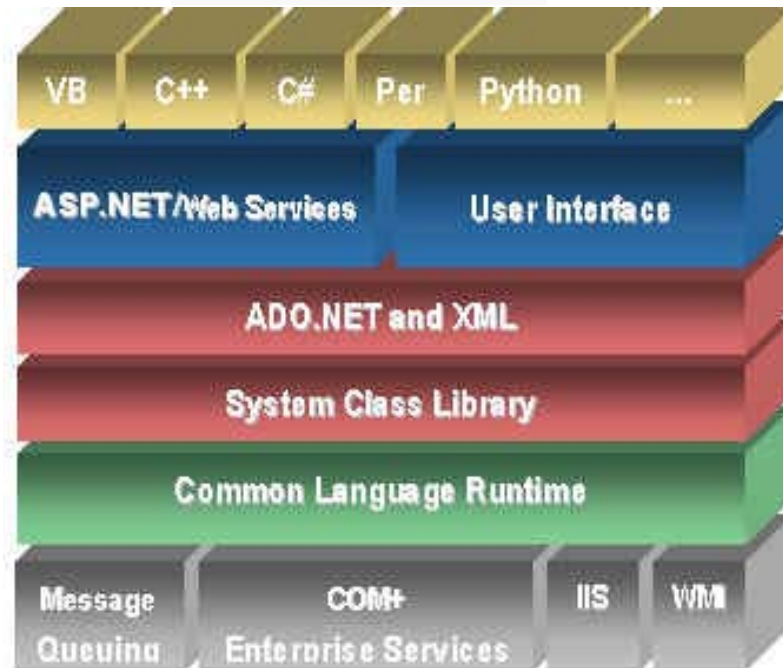
董大偉在 ASP.NET 徹底研究[25]書中提到過去利用 ASP 撰寫程式最容易發生的幾個缺點，包括執行效率、程式結構化、物件導向概念、系統的不穩定性和安全性疑慮等..這些問題在 .NET Framework 中一一被正視並且解決，並且配合 Visual Studio.NET 整合開發環境使得開發過程的過程，便利了許多。它具備了底下幾個特點改善了上述所提到的問題:

- 提供完整的類別程式庫(Class Library):讓程式設計師，有更便利的物件可用，過去要透過其他廠商所提供的物件才能完成的功能，現在只要幾行程式就可以完成，而物件導向的概念，完整的落實到 Web 程式設計中，以往的 ASP 與 VB 程式幾乎沒有物件導向特性。
- 改良的 ADO.NET、ASP、VB 與 C#等程式語言:.NET Framework 徹底的改善

了 WEB 應用程式的開發流程，利用 VB.NET 來開發 WEB 應用程式會比過去的 ASP 開發流程簡單許多。

- 與 Web 的整合:.NET Framework 讓應用程式與網際網路做完整的結合，像是一種新的運算平台，可簡化在網際網路上，在高度分散式環境中應用程式的開發。

.NET Framework 的環境架構圖如【圖 5】所示：



【圖 5】.NET Framework 環境架構圖

(資料來源: EEP ASP .NET 深入探討 [HTTP://www.infolight.com.tw](http://www.infolight.com.tw))