
Java EE7 Web

課程簡介及開發環境建置

鄭安翔

ansel_cheng@hotmail.com

課程大綱

- 1) 課程簡介
- 2) 課程開發及執行環境
- 3) 網路應用程式技術演進

課程對象及目標

- 課程對象：**Web** 應用程式的開發人員。
- 課程目標：
 - 使用**Java** 程式語言來開發**Web** 應用程式。
 - 使用 **Java™ servlets** 開發網站流程
 - 使用**session**、**Filters** 與 **Security** 管理
 - 使用 **Java Server Pages™** 技術撰寫網頁 (**JSP**)
 - 表達語言(**Expression Language**)、**JSP** 標準標籤函式庫(**JSTL**)
 - 資料庫整合
 - 進階網站開發技術
 - **ORM**軟體工程 – **Java Persistent API**
 - **RESTful Web Service**設計與應用
 - **WebSocket** 服務整合應用

課前準備

- 會撰寫Java SE應用程式
- 會使用Java technology API 文件
- 了解基本的UML 的塑模語言來
 - 分析與設計軟體系統
- 基本HTML知識
- 基本資料庫語法

教學內容

■ Slides

- 教師自編教學投影片

■ Examples

- 上課教學範例

■ Exercises

- 上課練習
- 投影片截圖

■ Solutions

- 練習解答

課程學習地圖

第一堂 網路應用程式技術演進

第二堂 Java 網路應用解決方案

第三堂 MVC設計一 – Servlet 處理請求

第四堂 MVC設計二 – JSP 顯示回應

第五堂 Servlet組態

第六堂 網站應用系統共用資源

第七堂 網站應用系統狀態管理

第八堂 JSP EL/JSTL運算語言應用

第九堂 JSP Action Element應用 自訂標籤

第十堂 應用程式與資料庫的整合

第十一堂 應用程式安全機制

第十二堂 Filter 請求攔截機制

第十三堂 非同步處理機制

第十四堂 網站ORM軟體工程

第十五堂 RESTful Web Service設計與應用

課程大綱

- 1) 課程簡介
- 2) 課程開發及執行環境
 - 安裝 **Java SDK**
 - 安裝 **NetBeans IDE**
 - 安裝 **Tomcat**
 - **NetBeans** 管理 **Tomcat**
- 3) 網路應用程式技術演進

課程開發及執行環境

- 開發及執行環境
 - Java 8 SDK 以上
 - NetBeans 11
 - Java EE 7
 - Servlet 4.0 / JSP 2.3 / EL 3.0
 - Servlet Container
 - Apache Tomcat 9.0 以上
 - Apache TomEE

課程開發及執行環境

■ 開發環境 Java EE 7

□ Java SDK

- <https://www.oracle.com/tw/java/technologies/javase-downloads.html> 下載

- jdk-11.0.x_windows-x64_bin.exe

□ NetBeans 11.x 整合開發工具IDE

- <https://netbeans.apache.org/download/index.html> 下載

- Apache-Netbeans-11.x-bin-windows-x64.exe

□ Apache Tomcat 9.0.x

- <https://tomcat.apache.org/download-90.cgi> 下載

- apache-tomcat-9.0.x.exe

安裝 Java SDK

■ Java SDK下載

- <https://www.oracle.com/tw/java/technologies/javase-downloads.html> 下載
- jdk-11.0.x_windows-x64_bin.exe

■ 安裝Java SDK

- 執行 jdk-11.0.7_windows-x64_bin.exe

安裝 Java SDK

■ 環境變數設定

- Windows 【控制台>系統及安全性>系統>進階系統設定】的【環境變數】項目
- 新增 JAVA_HOME = [Java SDK Location]
 - JAVA_HOME = C:\Program Files\Java\jdk-X.Y.Z
- 編輯 PATH 加入
 - %JAVA_HOME%\bin;

■ 測試

- java -version
- javac

安裝 NetBeans IDE

- NetBeans 下載

- <https://netbeans.apache.org/download/index.html>

- Apache-Netbeans-11.x-bin-windows-x64.exe

- 安裝 Netbeans

- 測試

課程開發及執行環境

■ 執行(佈署)環境

□ Apache Tomcat 9.0

<http://tomcat.apache.org/>

□ Tomcat 版本與Servlet / JSP / EL 版本對應

Apache Tomcat Version	Servlet Spec	JSP Spec	EL Spec	Latest Version	Supported Java Version
10.0.x	5.0	3.0	4.0	10.0.8	Java 8 以上
9.0.x	4.0	2.3	3.0	9.0.50	Java 8 以上
8.5.x	3.1	2.3	3.0	8.5.68	Java 7 以上
8.0.x	3.1	2.3	3.0	8.0.53	Java 7 以上
7.0.x	3.0	2.2	2.2	7.0.109	Java 6 以上

安裝 Tomcat

■ 下載Tomcat

- <https://tomcat.apache.org/download-90.cgi>

- apache-tomcat-9.0.x.exe

■ 安裝Tomcat

- 設定Shutdown Port

- 設定管理者帳號密碼

■ 檢視伺服器設定

安裝 Tomcat

■ 環境變數設定

- 新增 CATALINA_HOME = [Tomcat 安裝路徑]

CATALINA_HOME = C:\Program Files\Apache Software Foundation\Tomcat 9.0

- 編輯 PATH

PATH = %CATALINA_HOME%\bin;

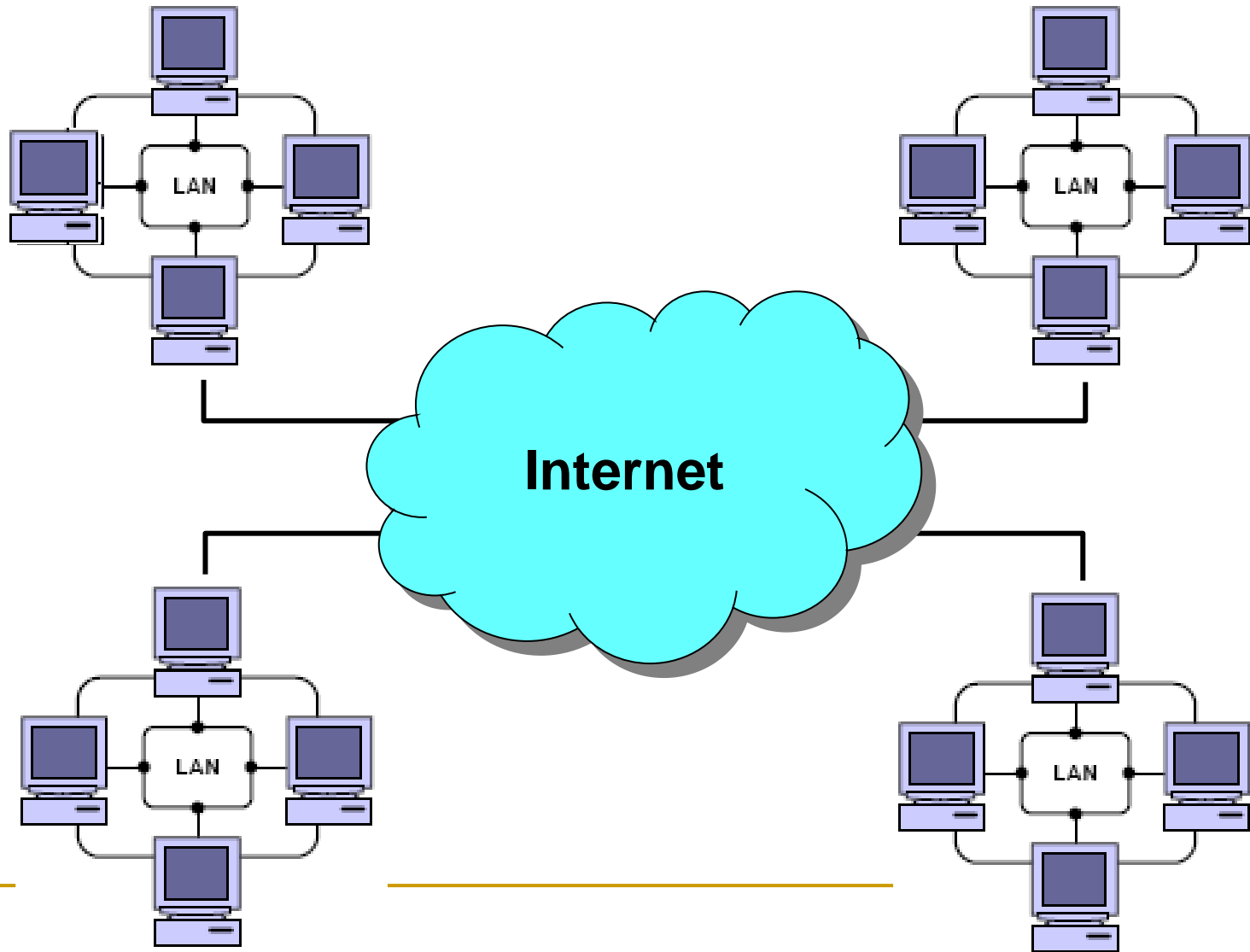
■ 測試

- 啟動關閉 Tomcat
- 瀏覽器檢視內建網站

課程大綱

- 1) 課程簡介
- 2) 課程開發及執行環境
- 3) 網路應用程式技術演進
 - 網路架構
 - 網路應用程式

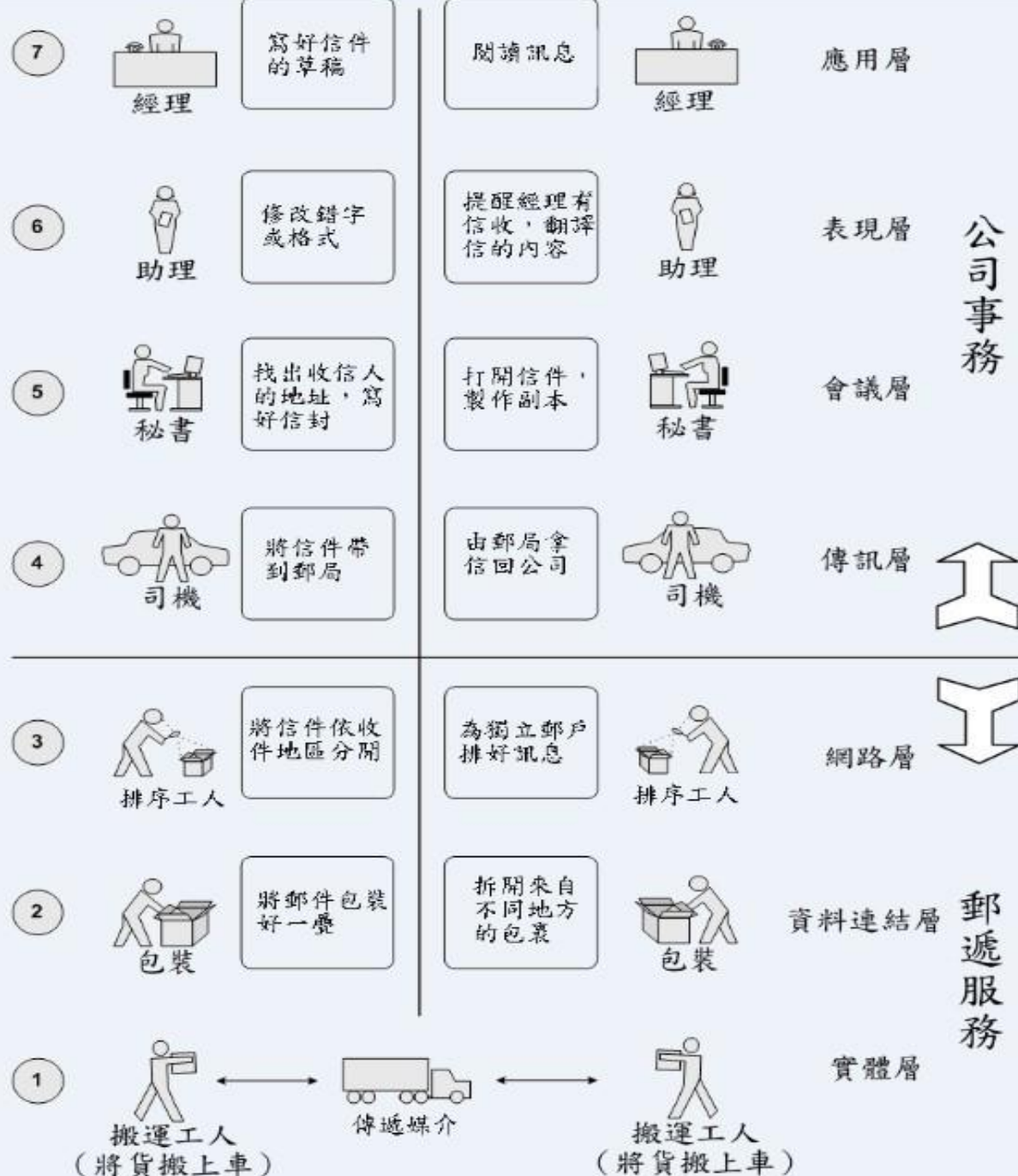
網際網路的發展



網路通訊協定堆疊

應用層 Application Layer	定義供應用程式存取的介面與功能, 常見的應用層協定包括HTTP、HTTPS、FTP、TELNET、SSH、SMTP、POP3等
表現層 Presentation Layer	處理各種資料格式(文字,圖片,影音)轉換,使系統可以正確的解讀數據,也提供資料的加密(ENCRYPTION)與解密(DECRYPTION), 壓縮(COMPRESSION)與還原.
會議層 Session Layer	定義連結建立與結束的對話(DIALOG)關係, 一般網路資料傳遞需先建立網路連線, 等到資料傳輸結束時, 再將連線中斷
傳送層 Transport Layer	確保資料到達順序與正確性,包括重組資料大小, 封包順序檢查,流量控制,超時(TIMEOUT)與重傳, 錯誤偵測與多工(MULTIPLEXING).
網路層 Network Layer	定義網路路由及定址功能。最主要的通訊協定是網際網路協定 (Internet Protocol, IP), IP位址會告訴網路設備這筆資料的來源及目的地。
資料鏈結層 Data-Link Layer	將資料切割並加上來源與目的地位址與資料長度等訊息包裝成頁框(FRAME). 並加上傳輸過程中處理流量控制及錯誤偵測, 以確保資料正確傳遞
實體層 Physical Layer	用來定義網路裝置之間的位元資料傳輸方式(在電線上傳遞的0與1電子訊號)。包含了纜線的規格、傳輸速度, 以及資料傳輸的電壓值, 用來確保訊號可以在多種物理媒介上傳輸。

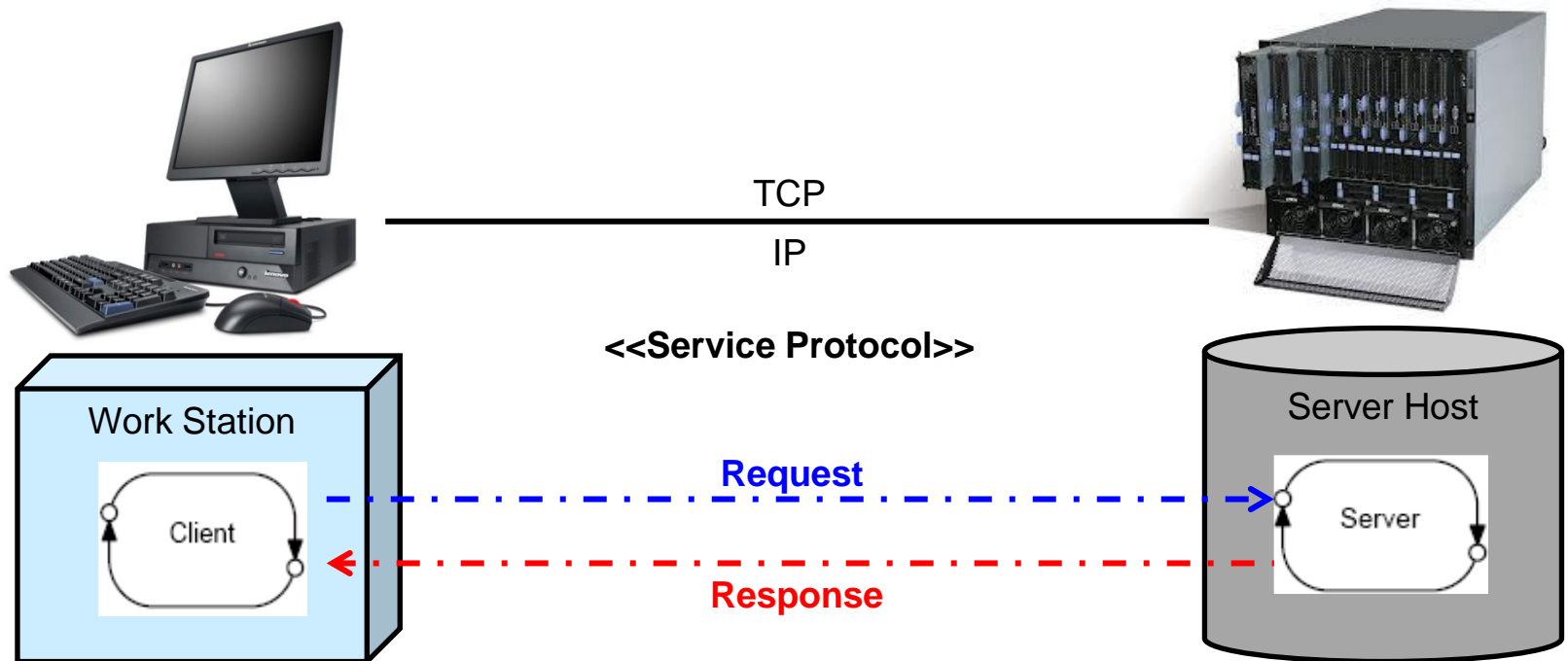
OSI 七層架構



OSI和郵件收發

網路Client Server 架構

- Client 發出Request 給Server,
- Server 接收Request所請求的的資料後進行處理
- 將結果Response 給Client.

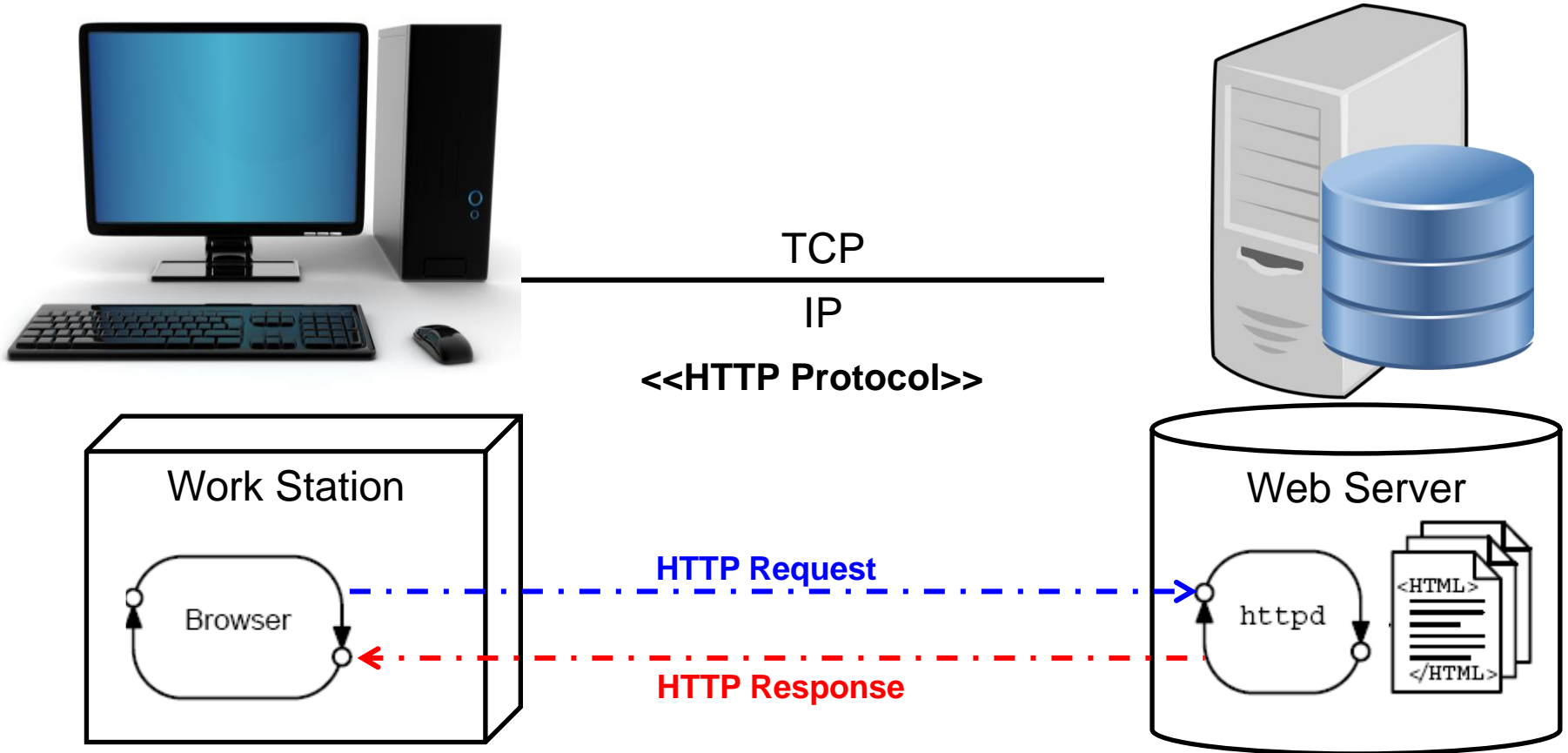


World Wide Web 網際網路簡介

■ World Wide Web

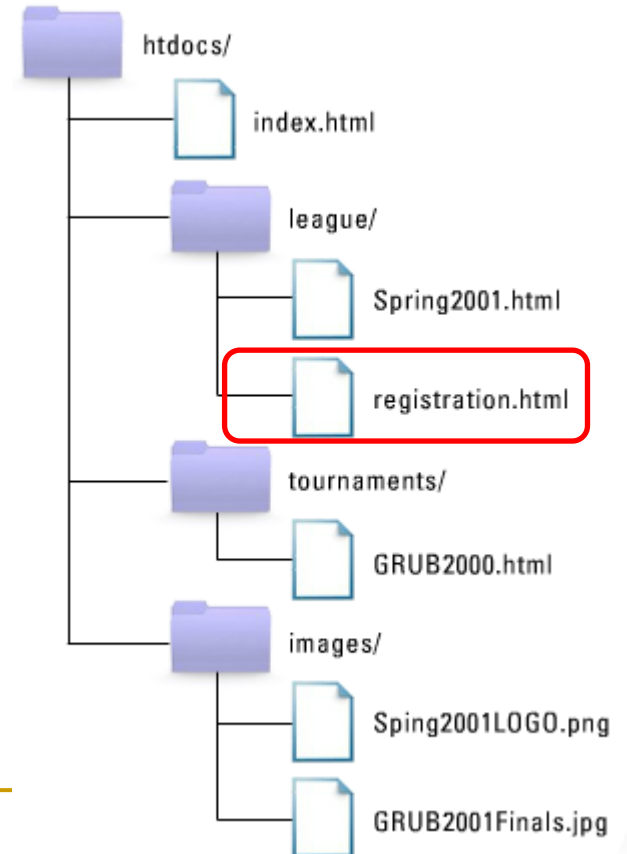
- 簡稱Web、「WWW」、全球資訊網
- 1989年歐洲高能粒子協會一個研究小組開發的Internet服務，Web能夠在網路上傳送圖片、文字、影像和聲音等多媒體資料。
- 使用HTTP通訊協定(HyperText Transfer Protocol)
- 支援HTML (Hypertext Markup Language) 文件語法
- 以URL(Uniform Resource Locator) 定址尋找資源所在

HTTP Client Server 架構

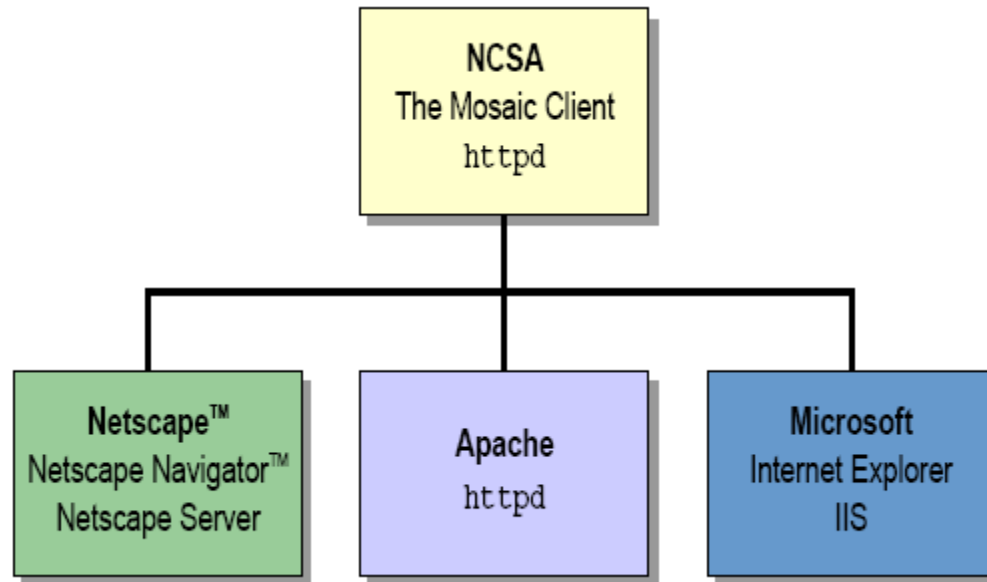


網站存取

<http://www.soccer.org:80/league/registration.html> →



常見網路伺服器及瀏覽器



HTTP通訊協定

- ❑ Client向Web Server 送出 HTTP Request
 - Web Server將開啟一個新連線connection
 - Web Server 解析並處理request
- ❑ Server傳回HTTP Response
 - Web Server 將處理結果包裝成HTTP Response
 - Web Server即關閉此connection
 - Client的執行狀態不會被保存(無狀態 Stateless)
- ❑ 資源更有效運用
 - 提供更多使用者同時上線
- ❑ 需用其他方法解決Stateless的問題
 - Cookies
 - URL Rewriting
 - SSL

HTTP通訊協定

HTTP請求訊息

```
GET /path/file.htm HTTP/1.1
Host: www.host.com
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg....
Accept-Language: zh-tw
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0(compatible; MSIE 6.0; Windows NT 5.1)
Connection: Keep-Alive
```

HTTP回應訊息

```
HTTP/1.1 200 OK
Date: Mon, 12 Oct 2004 19:12:16 GMT
Server: Apache/1.3.12(Unix) Debian/GNU mod_perl/1.24
Connection: Close
Content-Type: text/html
Content-Length: 1345
```

```
<html>
<body>
<h1>.....</h1>
```

```
.....
</body>
</html>
```

有狀態 **Stateful** 通訊協定-ex: **FTP**

- **FTP**為 **stateful** 通訊協定
 - 登入後,**Server**會開啟一個專屬連線給這個**Client**
 - 連線會一直保留,直到連線中斷為止,或閒置時間
Timeout
- **Client**端應用程式執行效能好
 - **Server**配置固定記憶體來服務每個**Client**,並記錄其執行狀態
 - **Client**送出新請求, **Server**不需額外身分驗證即資源配置動作,直接處理請求
- **Server**端資源浪費
 - 限制連線數目
 - 使用者閒置,**Server**不能服務其他使用者

HTML語法

- 標籤（Tag）的方式來定義文件結構
- 瀏覽器依HTML的結構等資訊進行畫面繪製



URL

■ Uniform Resource Locator

- URL 是用來存取網際網路上資源的唯一名稱.

■ Syntax

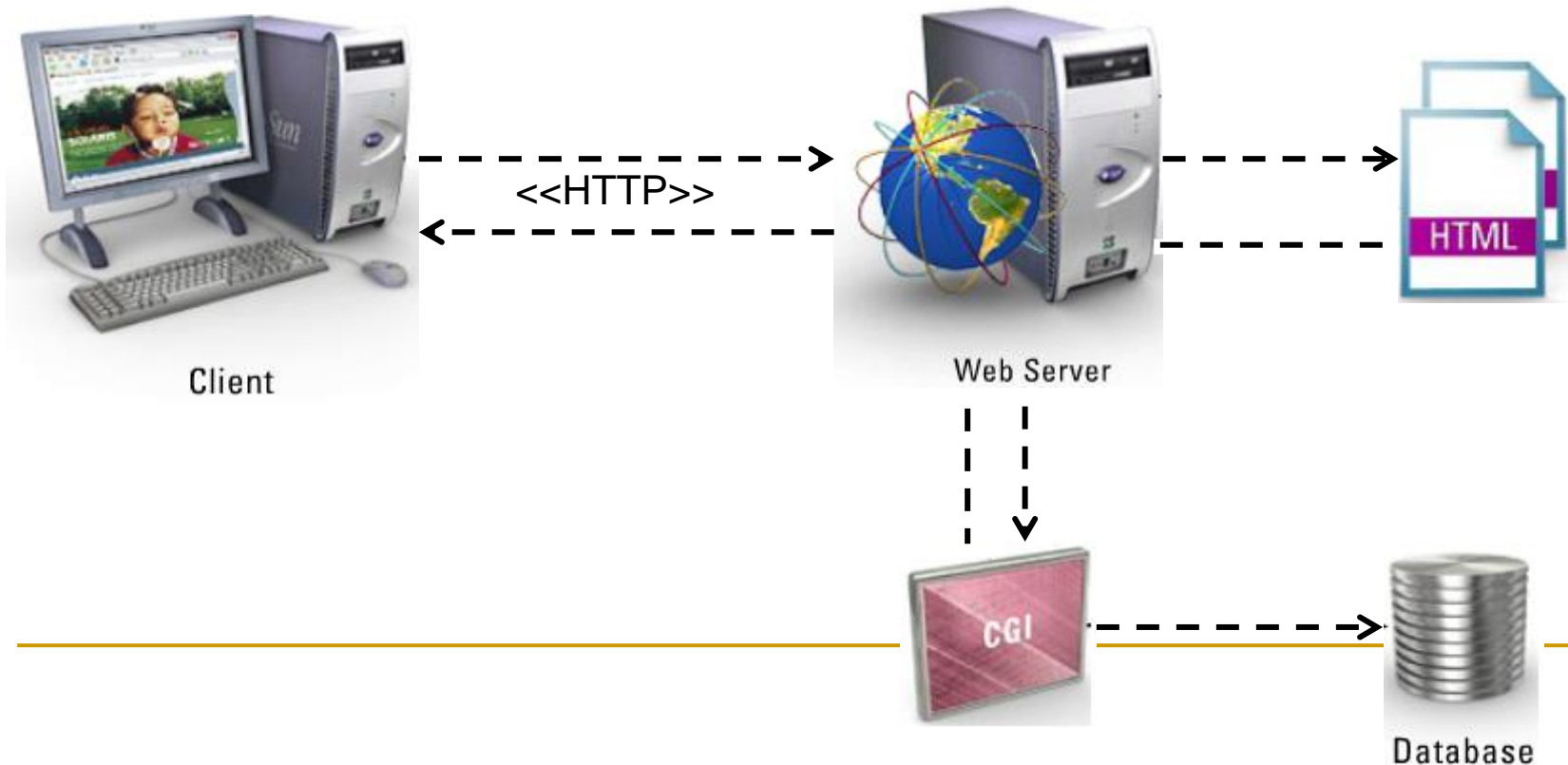
`protocol://host:port/path/file`

- `protocol` : 所使用的通訊協定.
- `host` : 遠端伺服器位置.
- `port` : 提供**Service** 的通訊埠.
- `path` : 完整的目錄結構.
- `file` : 資源**Resource** 名稱.

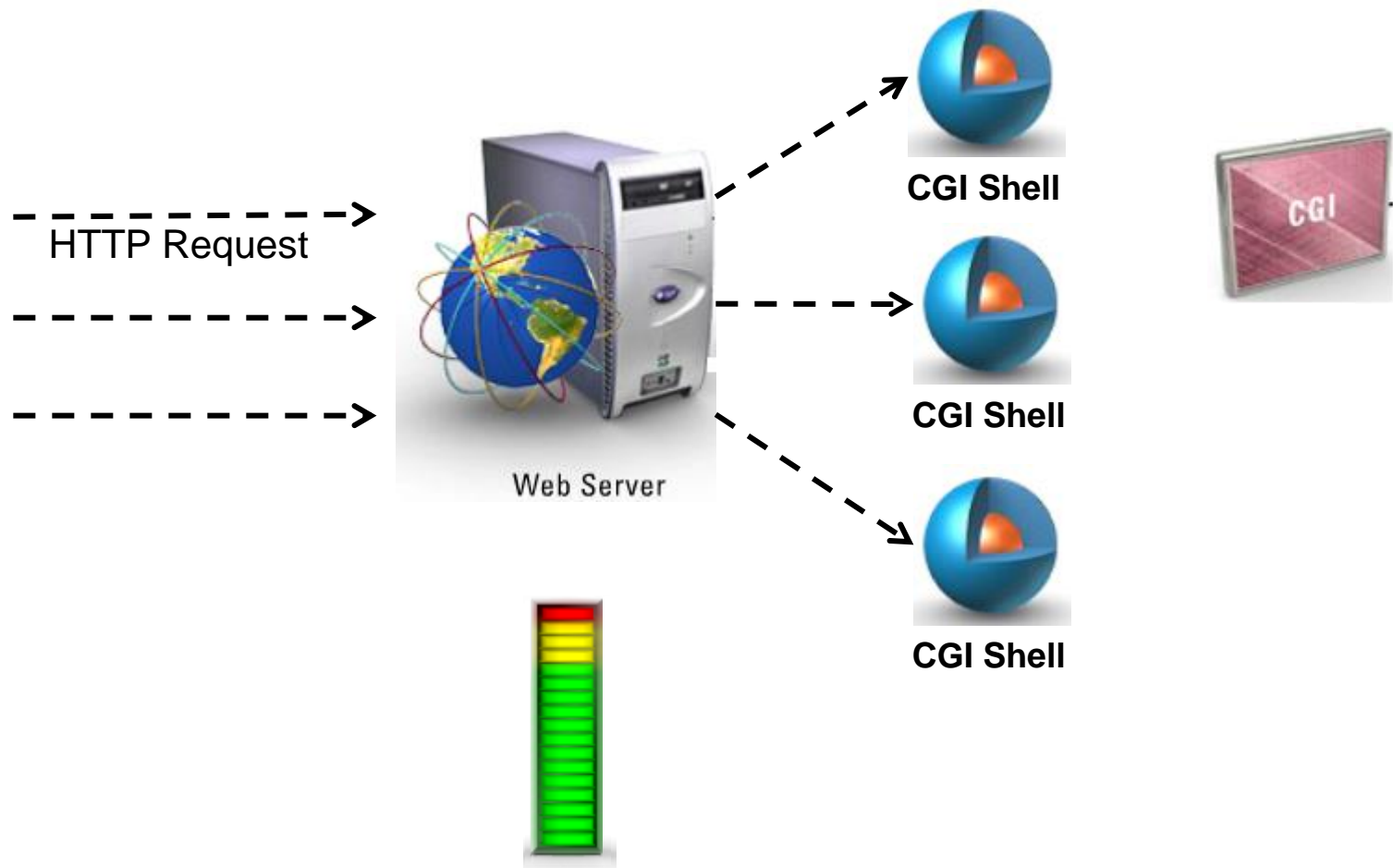
`http://www.oracle.com:80/technetwork/java/index.html`

靜態網頁 vs. 動態網頁

- 使用**CGI** 技術產生動態網頁
 - 使用者透過**HTML**的表單(**Form**)傳送數據至伺服器端
 - 伺服器端使用**CGI**技術處理使用者動態請求



CGI Common Gateway Interface



CGI 技術產生動態網頁

■ CGI 技術產生動態網頁優點

- 可以多種不同語言開發
- 為獨立程序,有問題時不影響網路伺服器運作
- 處理資料庫資料時是獨立並行的

■ 缺點

- 程式回應時間較長
- 擴充性有限
- 安全性較低
- 非物件導向,程式重複使用性低
- 程式語言受限於作業系統平台提供的功能
- 同時處理商業邏輯與展示層,不易維護

NetBeans 管理 Tomcat

- NetBeans 設定管理 Tomcat
 - Service – Server 管理
 - 新增 Apache Tomcat or TomEE Server
 - 管理者帳號密碼
- 測試
 - NetBeans 中啟動Tomcat
 - Tomcat 伺服器首頁
 - WebApp Manager (Optional)
 - Tomcat User設定管理者帳號，新增manager-gui權限