

Tecnologie di Sviluppo per il Web

Programmazione Web: Architetture

versione 3.0

Questo lavoro è concesso in uso secondo i termini di una licenza Creative Commons
(vedi ultima pagina)

G. Mecca – mecca@unibas.it – Università della Basilicata



Programmazione Web: Architetture >> Sommario



Sommario

- Introduzione
 - ⇒ Server Applicativo per il Web
- Principali tecnologie
 - ⇒ ASP 3.0
 - ⇒ PHP
- Sun J2EE e Microsoft .NET
- In questo corso

G. Mecca - Tecnologie di Sviluppo per il Web

2



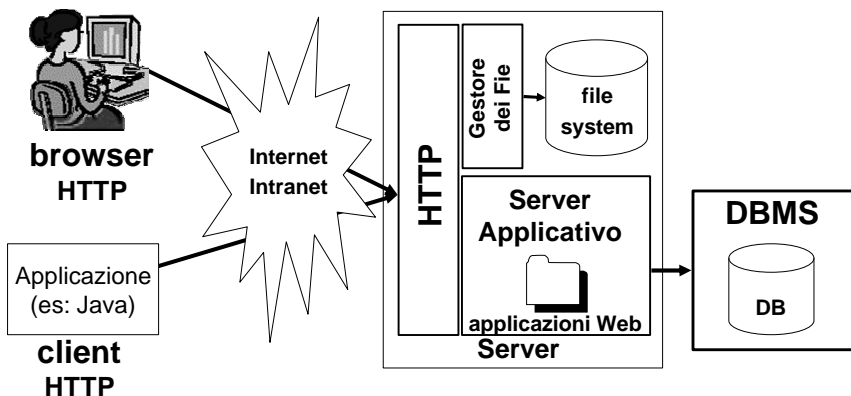
Introduzione

- Nella lezione precedente
 - ⇒ tecniche per lo sviluppo di applicazioni Web
 - ⇒ per semplificare la vita al programmatore, è necessario utilizzare piattaforme che forniscano servizi applicativi
- Server applicativo
 - ⇒ modulo aggiuntivo del server Web che fornisce i servizi richiesti



Introduzione

○ Server Web con Server Applicativo





Server Applicativo per il Web

○ Web Application Server

- ⇒ software che fornisce servizi per lo sviluppo di applicazioni Web
- ⇒ lavora congiuntamente al server HTTP oppure include un server HTTP
- ⇒ eterogeneità nelle tecnologie e nell'offerta di servizi (fascia alta, fascia bassa)



Server Applicativo per il Web

○ Servizi tipici

- ⇒ gestione del ciclo di vita delle applicazioni (es: più richieste con una attivazione)
- ⇒ decodifica delle richieste del client (es: decodifica di valori sottomessi con una form)
- ⇒ supporto alla produzione della risposta HTTP
- ⇒ gestione delle sessioni (es: utilizzando in modo trasparente i cookie)
- ⇒ gestione del contesto dell'applicazione



Server Applicativo per il Web

- Servizi di fascia alta
 - ⇒ replicazione e “clustering”
 - ⇒ bilanciamento del carico
 - ⇒ alta affidabilità (“failover”)
 - ⇒ transazioni distribuite
 - ⇒ gestione trasparente della persistenza
 - ⇒ sicurezza
 - ⇒ servizi Web (“Web services”)



Principali Tecnologie

- Tecnologie non orientate agli oggetti
 - ⇒ Microsoft ASP 3.0 (VBScript)
 - ⇒ PHP (PHP: Hypertext Preprocessor)
 - ⇒ ColdFusion
- Tecnologie orientate agli oggetti
 - ⇒ Sun Java 2 Enterprise Edition (J2EE)
 - ⇒ Microsoft ASP.NET



Microsoft ASP 3.0

- Applicazione Web con ASP 3.0
 - ⇒ collezione di pagine ASP (.asp)
- Pagina ASP
 - ⇒ codice HTML
 - ⇒ codice di script eseguiti dal client (browser)
JScript, VBScript
 - ⇒ codice di script eseguiti dal server prima di
inviare la pagina al browser <% ... %>



Microsoft ASP 3.0

- Esempio di pagina ASP 3.0

```
<HTML>
<!-- primo.asp -->
<HEAD>
<TITLE>Primo Esempio</TITLE>
</HEAD>
<BODY>
  Benvenuti.
  Ecco la data e l'ora <% Response.write Date %>
  <% Response.write Now %>
</BODY>
</HTML>
```



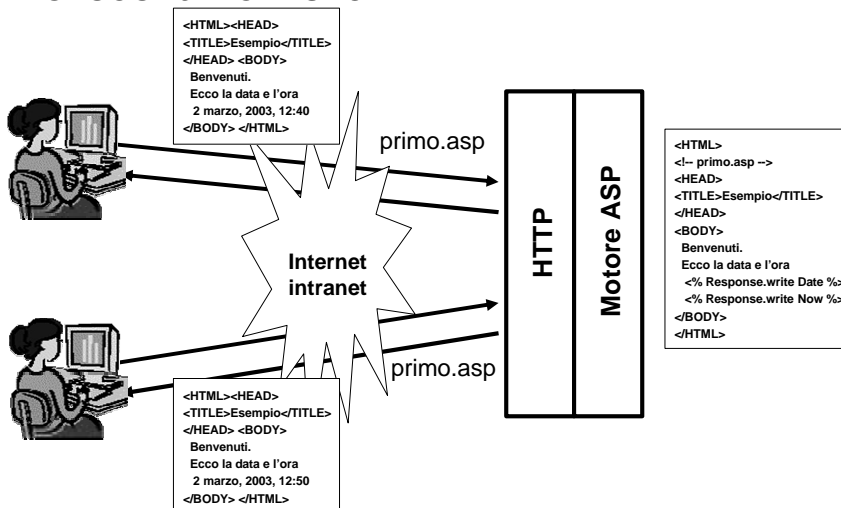
Microsoft ASP 3.0

o Semantica

- ⇒ il server riceve una richiesta ad un URI che corrisponde ad una pagina ASP
- ⇒ il motore ASP preleva il codice ASP
- ⇒ gli script vengono eseguiti dal server prima di inviare la risposta al client
- ⇒ il codice HTML risultante viene inviato come risposta al client



Microsoft ASP 3.0





Microsoft ASP 3.0

○ Il linguaggio VBScript

- ⇒ linguaggio in cui vengono scritti gli script delle pagine ASP 3.0
- ⇒ linguaggio interpretato
- ⇒ sottoinsieme di Visual Basic
- ⇒ debolmente tipato (uso senza dichiarazione)
- ⇒ programmazione “basata su oggetti”



Microsoft ASP 3.0

○ Programmazione “basata su oggetti”

- ⇒ è possibile utilizzare oggetti predefiniti invocandone i metodi
- ⇒ è possibile creare nuovi oggetti di classi predefinite
- ⇒ non è possibile definire nuove classi
- ⇒ il server applicativo (Internet Information Services) mette a disposizione degli oggetti predefiniti per gestire le richieste



Microsoft ASP 3.0

○ Oggetti predefiniti

- ⇒ Request: rappresenta la richiesta HTTP (inclusi i valori delle eventuali form)
- ⇒ Response: rappresenta la risposta HTTP
- ⇒ Session: rappresenta la sessione di lavoro dell'utente
- ⇒ Application: rappresenta il contesto dell'applicazione
- ⇒ Server: rappresenta il server HTTP



Microsoft ASP 3.0

○ Tipicamente, in una pagina ASP 3.0

- ⇒ gli script leggono i dati sottmessi dall'utente accedendo all'oggetto Request
es: nome = Request.QueryString("nome")
- ⇒ salvano dei dati nell'oggetto Session per mantenere lo stato (utilizza i cookie)
- ⇒ scrivono il risultato da inviare al client utilizzando l'oggetto Response
es: Response.Write("Tentativi:" & tentativi)

PHP

- Filosofia molto simile ad ASP 3.0
- Linguaggio di scripting
 - ⇒ interpretato dal server
 - ⇒ eseguibili su tutti i principali server HTTP
 - ⇒ open source (www.php.net)
 - ⇒ debolmente tipato
 - ⇒ modeste funzionalità orientate agli oggetti
 - ⇒ privilegia la semplicità

PHP

○ Esempio

```
<HTML>
<!-- primo.php -->
<HEAD>
<TITLE>Primo Esempio</TITLE>
</HEAD>
<BODY>
  <?php echo "Benvenuti"; ?>
  <?php echo "Ecco la data e l'ora"; ?>
  <?php echo date("F j, Y, g:i a"); ?>
</BODY>
</HTML>
```



Tecnologie non Orientate agli Oggetti

- In sintesi
 - ⇒ applicazione Web come collezione di script (funzioni)
- Vantaggio
 - ⇒ semplicità d'uso (rapidità di sviluppo)
- Limiti
 - ⇒ poco scalabili
 - ⇒ paradigma di programmazione Web diverso da quello tradizionale



Tecnologie Orientate agli Oggetti

- Applicazioni Web
 - ⇒ collezioni di "componenti" (classi e oggetti)
 - ⇒ definiscono la logica applicativa
- Idea fondamentale
 - ⇒ lo sviluppo di un'applicazione Web non dovrebbe essere molto diverso dallo sviluppo di un'applicazione tradizionale
 - ⇒ es: riusabilità della logica applicativa



Tecnologie Orientate agli Oggetti

- Componenti “leggeri”

- ⇒ oggetti di classi ordinarie progettate secondo criteri opportuni per facilitare il riuso

- ⇒ es: JavaBeans

- Componenti “pesanti”

- ⇒ oggetti di classi speciali, che usufruiscono di servizi particolari (es: distribuzione, persist.)

- ⇒ es: Enterprise JavaBeans (EJB)



J2EE e .NET

- Due tecnologie OO principali

- ⇒ J2EE

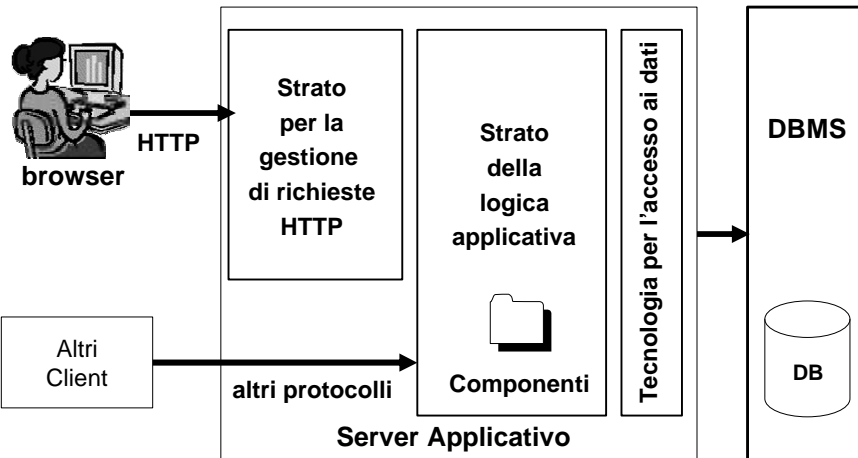
- ⇒ ASP.NET

- Estremamente simili

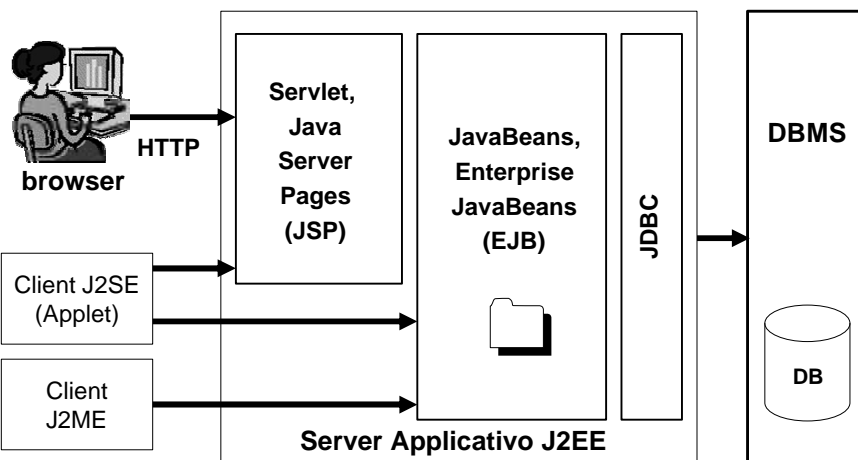
- ⇒ rispondono ad un'architettura di riferimento comune

- ⇒ con alcune differenze (marginali) nelle tecnologie componenti

Architettura di Riferimento



Architettura J2EE



J2EE

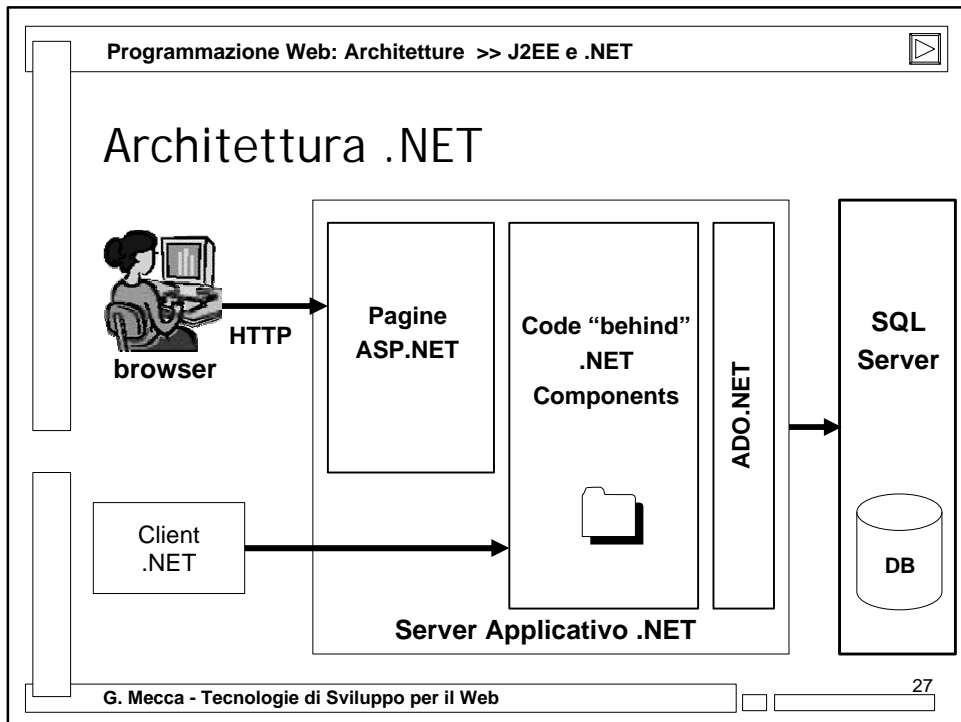
- Servlet
 - ⇒ classi Java orientate a gestire richieste e risposte (in particolare HTTP)
- Java Server Pages
 - ⇒ pagine in cui il codice HTML può essere mischiato a codice Java
 - ⇒ compilate in servlet
- JavaBeans e Enterprise JavaBeans
 - ⇒ componenti “leggeri” e “pesanti”

JSP

- Esempio di pagina JSP

```

<HTML>
<!-- primo.jsp -->
<HEAD>
<TITLE>Primo Esempio</TITLE>
</HEAD>
<BODY>
  Benvenuti.
  Ecco la data di oggi
  <% out.println(new java.util.Date()); %>
</BODY>
</HTML>
    
```



- Programmazione Web: Architetture >> J2EE e .NET
- ## .NET
- **Pagine ASP.NET**
 - ⇒ pagine in cui il codice HTML può essere mischiato con il codice C#, VB.NET ecc.
 - ⇒ compilate e non interpretate
 - ⇒ non esiste un corrispondente dei servlet
 - **ADO.NET**
 - ⇒ tecnologia per l'accesso a DBMS relazionali
- G. Mecca - Tecnologie di Sviluppo per il Web 28



Confronto tra le Architetture

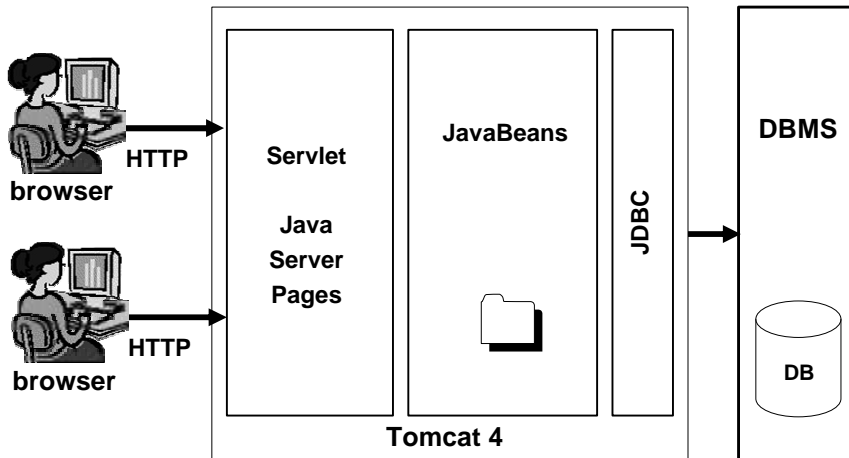
- Vantaggi dell'architettura J2EE
 - ⇒ standard indipendente dalla piattaforma
- Vantaggi dell'architettura .NET
 - ⇒ altamente ottimizzata per Windows
- In effetti
 - ⇒ oramai le due architetture sono intercambiabili
- Il prossimo obiettivo: "Web services"



In Questo Corso

- Architettura complessiva complessa
 - ⇒ server applicativi costosi
 - ⇒ si giustifica solo per applicazioni di fascia alta
- Forma semplificata
 - ⇒ applicazioni di fascia media
 - ⇒ componenti "leggeri"

In Questo Corso



Sommario

- Architetture per Applicazioni Web
 - ⇒ Server Applicativo per il Web
- Principali tecnologie
 - ⇒ ASP 3.0
 - ⇒ PHP
- Sun J2EE e Microsoft .NET
- In questo corso



Termini della Licenza

- This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/1.0/> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.
- Questo lavoro viene concesso in uso secondo i termini della licenza "Attribution-ShareAlike" di Creative Commons. Per ottenere una copia della licenza, è possibile visitare <http://creativecommons.org/licenses/by-sa/1.0/> oppure inviare una lettera all'indirizzo Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.