



Università di Bergamo
Facoltà di Ingegneria

Applicazioni Internet B

Paolo Salvaneschi

B2_1 V1.7

Le tecnologie software Internet

Standard aperti / Sun Java

Il contenuto del documento è liberamente utilizzabile dagli studenti, per studio personale e per supporto a lezioni universitarie.
Ogni altro uso è riservato, e deve essere preventivamente autorizzato dall' autore.
Sono graditi commenti o suggerimenti per il miglioramento del materiale

INDICE

- Tecnologie (standard aperti + Sun Java)
 - Pagine Web statiche (HTML,CSS)
 - Scripting lato client, JavaScript (DHTML)
 - Java Applet
 - CGI
 - Java Servlet
 - Java Server Pages
 - Connessione con i Data Base

Tecnologie (standard aperti + Sun Java)

- Tecnologie
- Nomenclatura:

Linguaggio

es. Java

Tecnologia

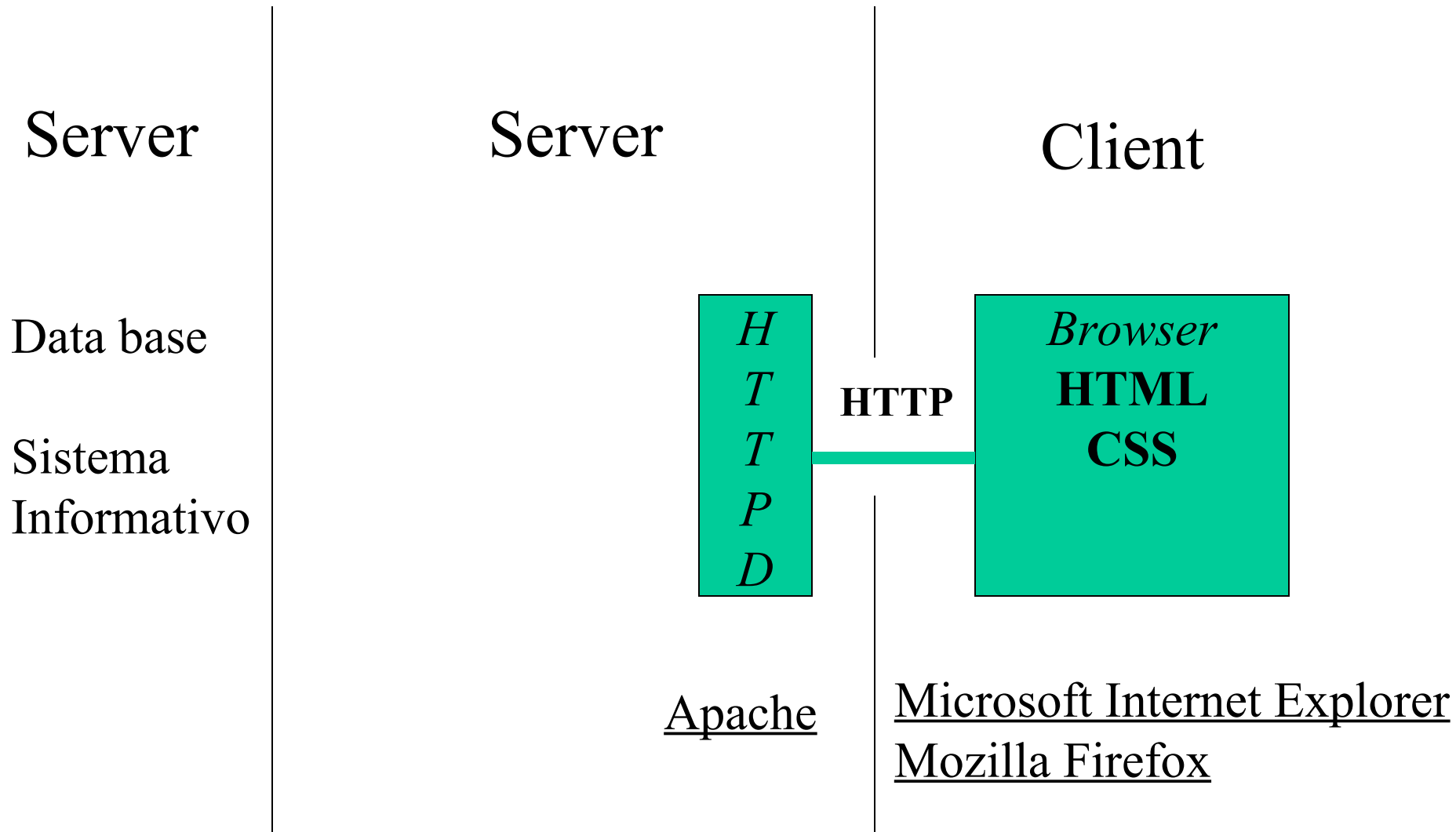
es. JSP

Esempio di Strumento

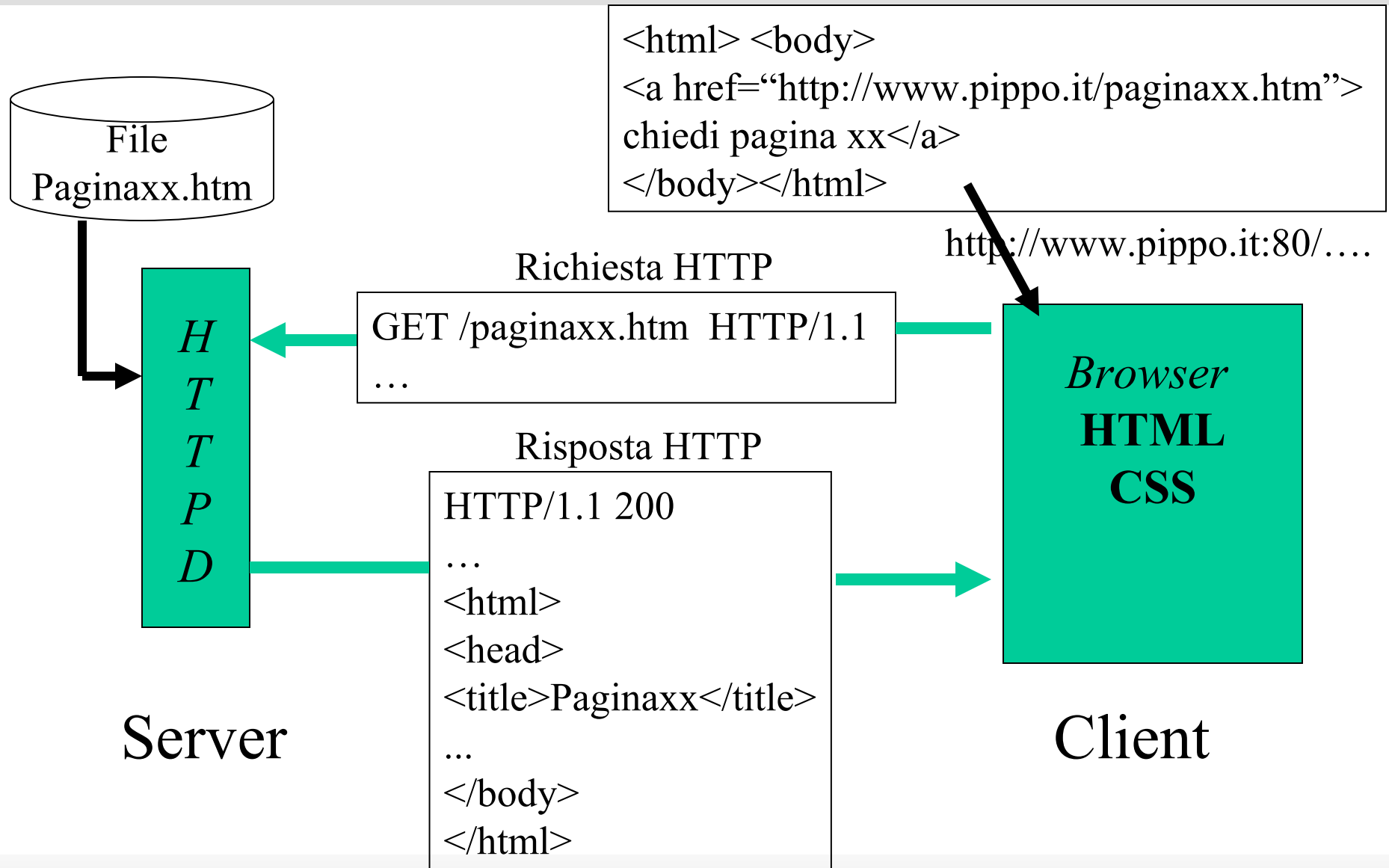
es. Tomcat

- Pagine Web statiche (HTML,CSS)
- Pagine Web dinamiche lato Client
 - Scripting lato client, JavaScript (DHTML)
 - Java Applet
- Pagine Web dinamiche lato Server
 - CGI
 - Servlet, JSP
 - Connessione con i Data Base

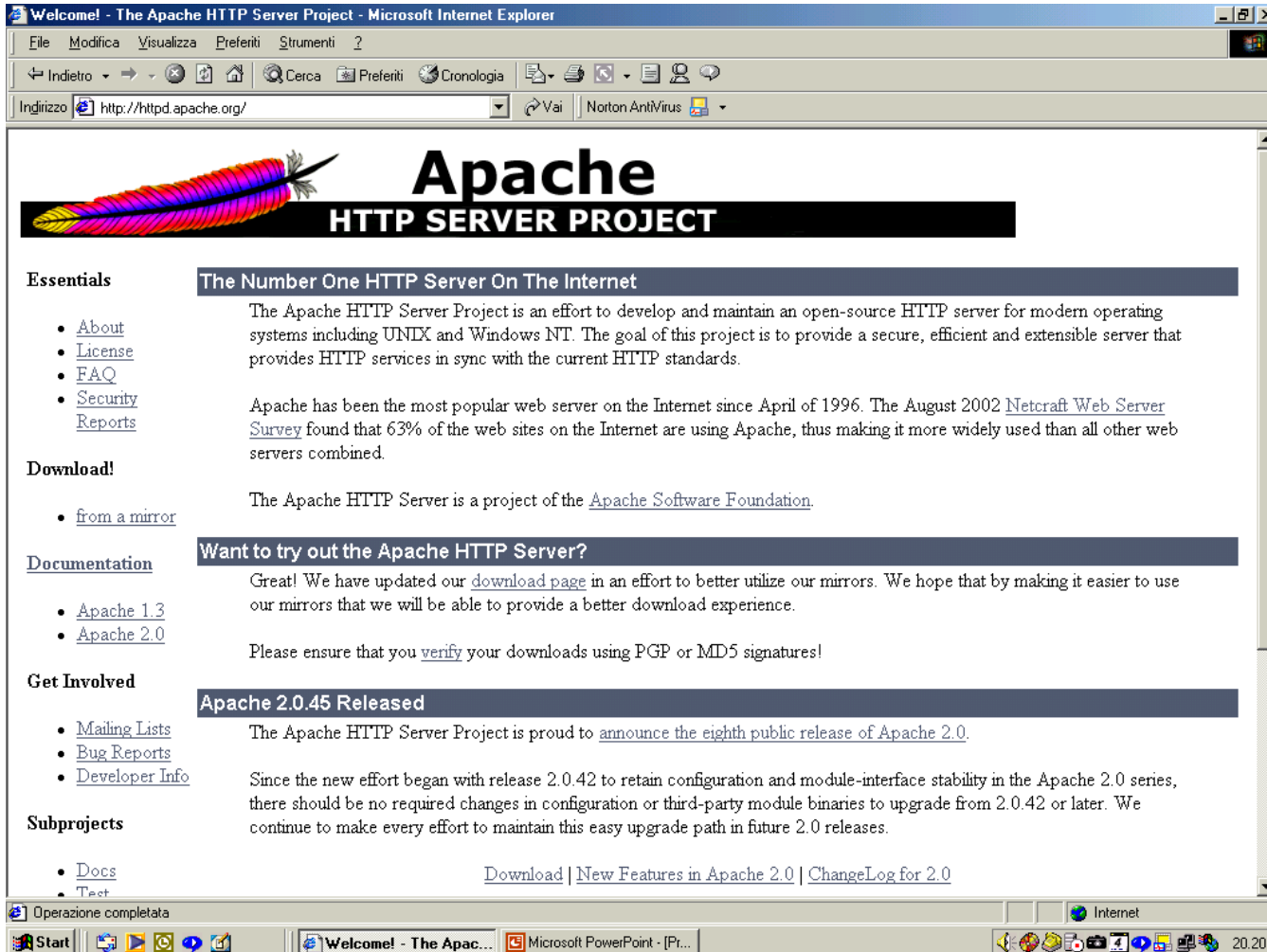
Pagine Web statiche (HTML,CSS)



Pagine Web statiche (HTML,CSS)



Pagine Web statiche (HTML,CSS)



<http://httpd.apache.org>

- Pagine Web statiche
 - Server di documenti ipermediali
 - Distribuzione di documenti codificati in HTML
 - Separazione della struttura dalla presentazione (CSS)

Pagine Web statiche (HTML,CSS)

CSS

```
<html>
```

```
<head>
```

```
<title>La prima pagina Web di Mario Rossi</title>
```

```
<link type="text/css"  
  rel="Stylesheet" href="Es2_css_esterno.css">
```

```
</head>
```

```
<body>
```

```
<h1>Home page di Mario Rossi</h1>
```

```
<p>Questa è la mia prima pagina in HTML.</p>
```

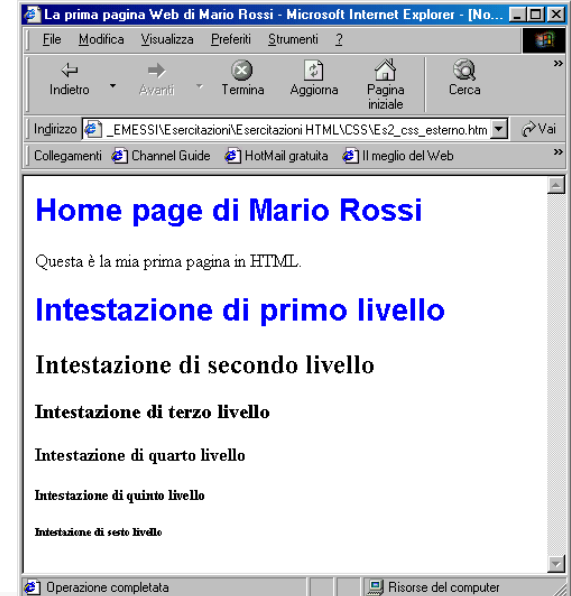
```
<h1>Intestazione di primo livello</h1>
```

```
<h2>Intestazione di secondo livello</h2>
```

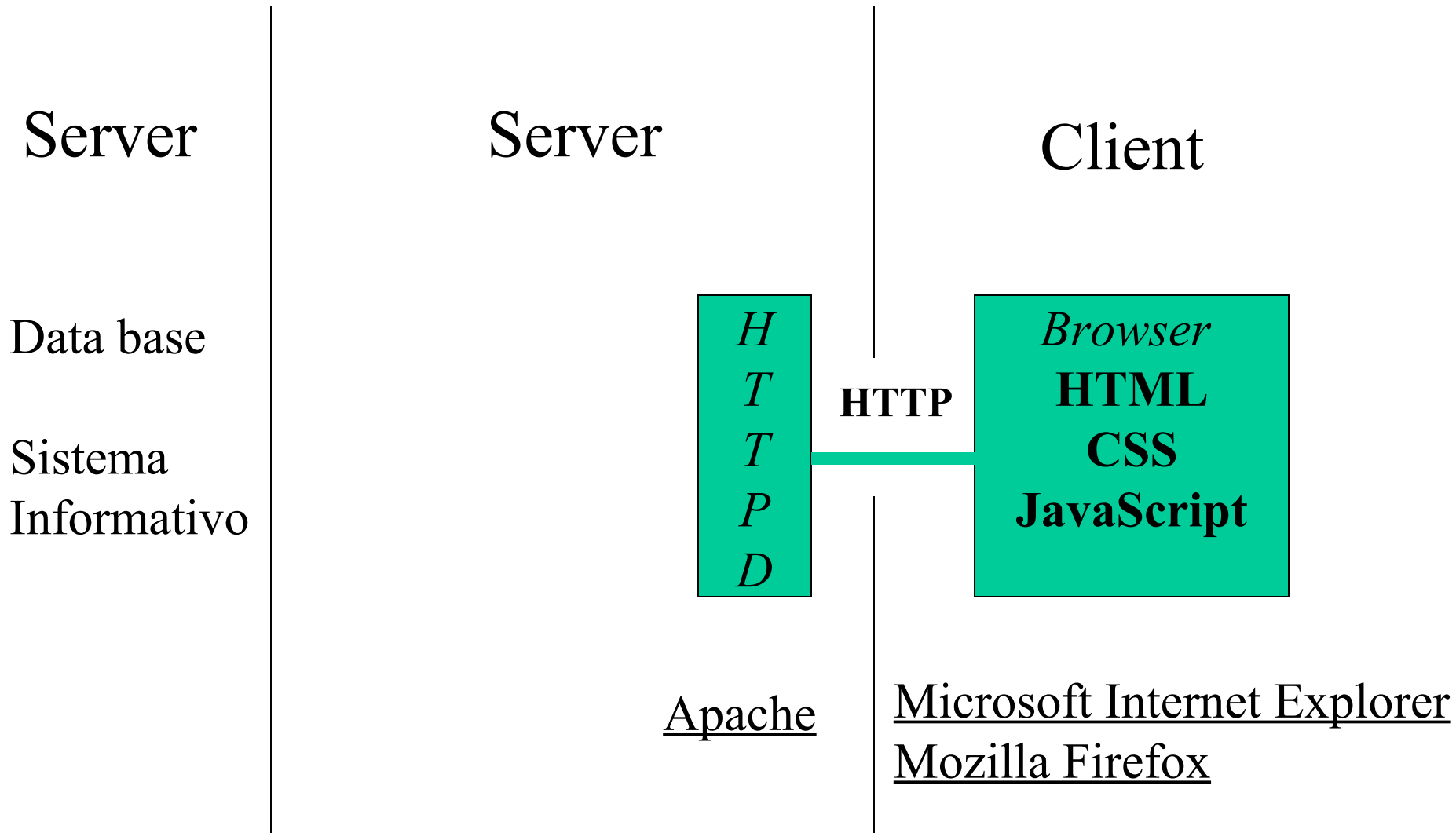
```
<h3>Intestazione di terzo livello</h3>
```

```
...
```

```
h1 {font-family: Arial;  
     font-size: 20pt;  
     color: blue;  
     font-weight: bold;}
```



Scripting lato client, JavaScript (DHTML)



- Pagine Web dinamiche (DHTML)
- Insieme di tecnologie che permettono di modificare le pagine HTML caricate nel browser senza interagire con il server
 - Es. modifica della formattazione a seguito di interazione con l'utente
 - Es. controllo dell'input

- HTML+CSS+Java Script
 - Integrazione di procedure interpretate nella pagina HTML ; Esecuzione lato client
 - Gestione eventi ed interazione locale con l'utente
 - Definizione oggetti e attributi (DOM Document Object Model) componenti la pagina e modificabilità da programma

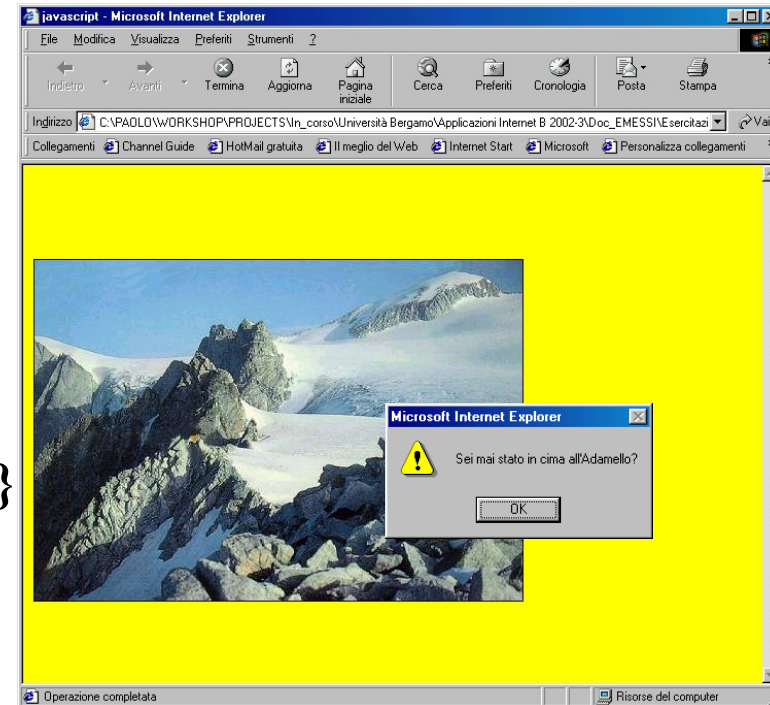
Scripting lato client, JavaScript (DHTML)

Scripting



```
<HTML>
<HEAD>
<TITLE>javascript</TITLE>
  <SCRIPT language="JavaScript">
    function presenta()
    {alert("Sei mai stato in cima all'Adamello?")}
  </SCRIPT>
</HEAD>

<BODY bgcolor="#FFFF00">
  <BR><BR><BR><BR>
  <IMG src="..\immagini\ada_sal.jpeg" onmouseover="presenta()">
</BODY>
</HTML>
```


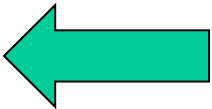


Gestore degli eventi



Scripting lato client, JavaScript (DHTML)

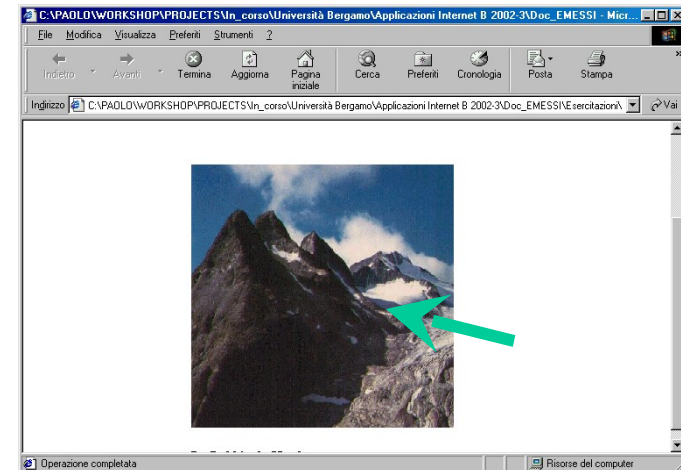
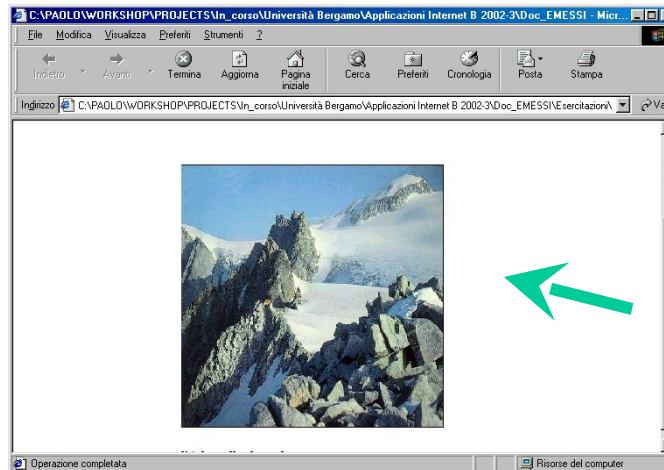
<HEAD>

```
<SCRIPT language="JavaScript">  
function uno()  
{document.all.container1.style.visibility="visible";  
document.all.container2.style.visibility="hidden";}   
function due()  
{document.all.container1.style.visibility="hidden";  
document.all.container2.style.visibility="visible";}   
</SCRIPT>
```

Scripting

DOM

...




Scripting lato client, JavaScript (DHTML)

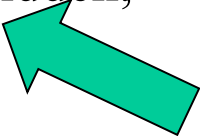
```
...
<STYLE type="text/css">
#container1 {
  position:absolute;
  top: 2in;
  left: 2in;
  width: 2in;
  visibility: visible;
}
#container2 {
  position:absolute;
  top: 2in;
  left: 2in;
  width: 2in;
  visibility: hidden;
}
</STYLE>
</HEAD>

<BODY>
<DIV id="container1">
  <IMG src="..\immagini\ada_sal.jpeg"
    width="300" height="300" onmouseover="due()">
  <P><b>l'Adamello da sud</b></P>
</DIV>
<DIV id="container2">
  <IMG src="..\immagini\ada_lobbie.jpeg"
    width="300" height="300" onmouseout="uno()">
  <P><b>Le Lobbie da Nord</b></P>
</DIV>
</BODY>
</HTML>
```

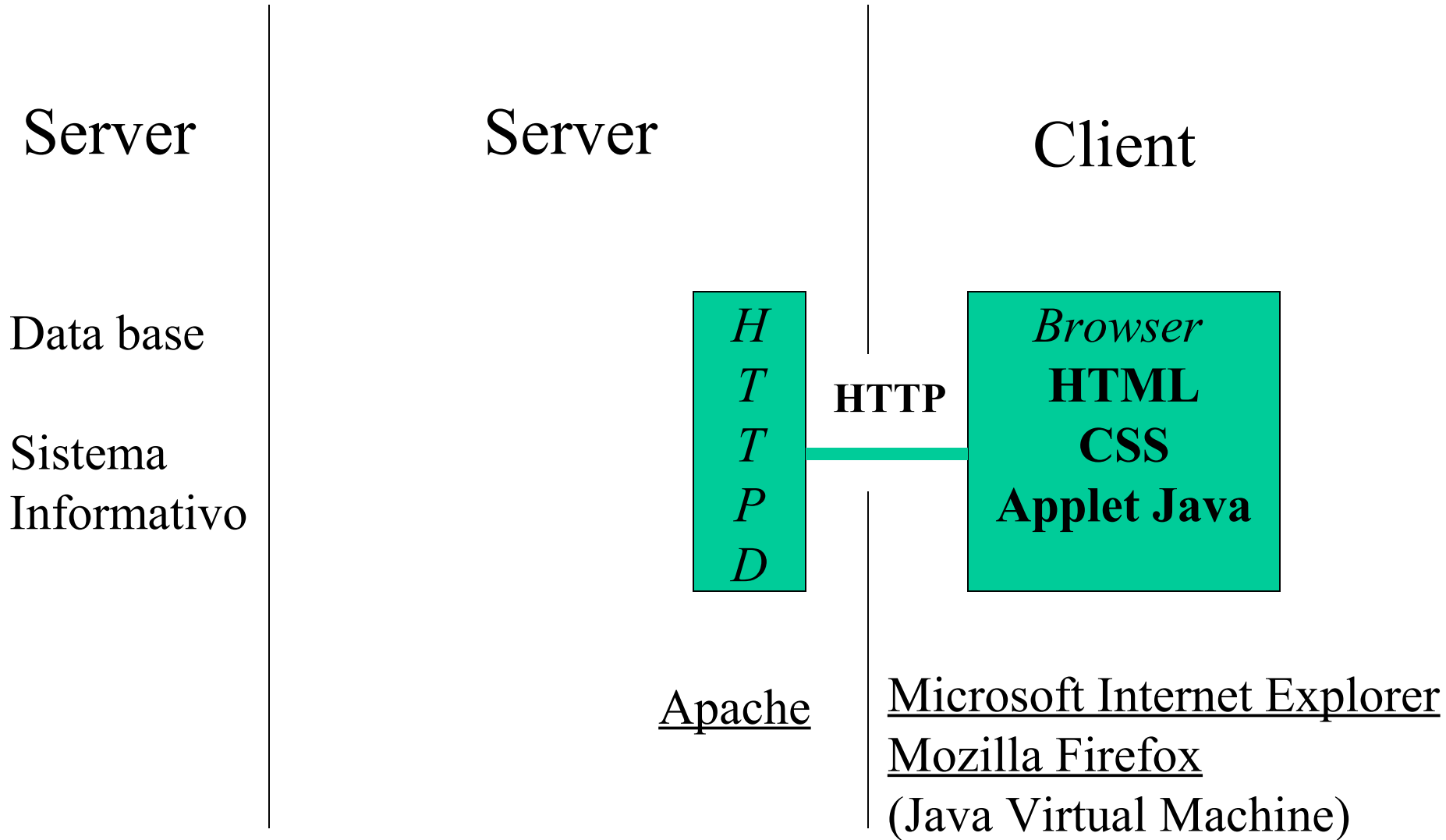
Eventi



CSS

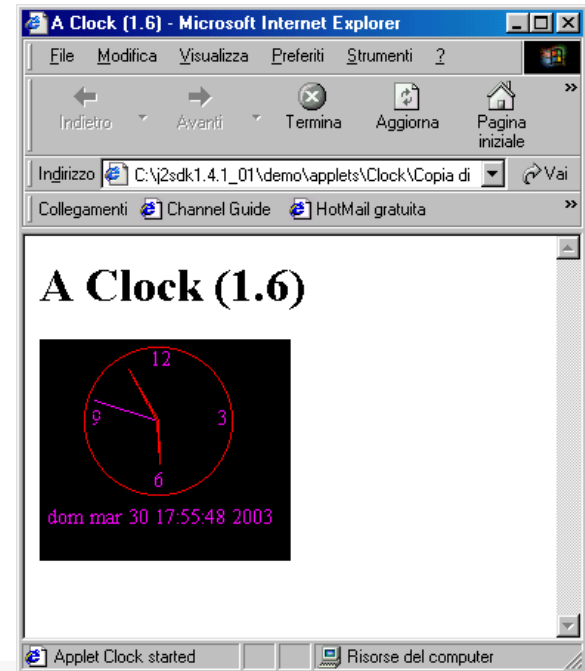


Java Applet

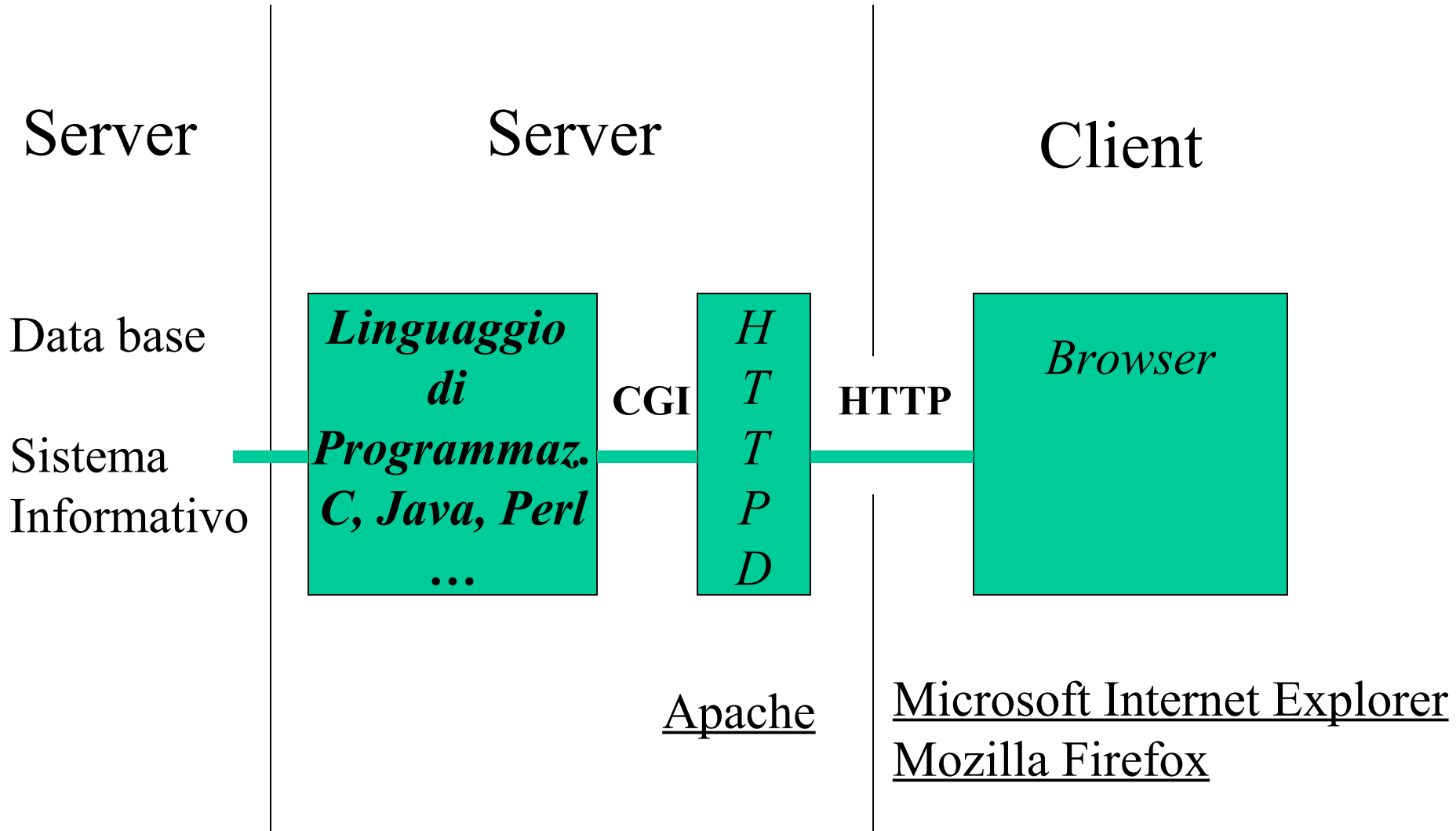


Inserimento oggetto

```
<HTML>
<HEAD>
<TITLE>A Clock (1.6)</TITLE>
</HEAD>
<BODY>
<h1>A Clock (1.6)</h1>
<applet code="Clock.class" width="170" height="150">
<param name="bgcolor" value="000000">
<param name="fgcolor1" value="ff0000">
<param name="fgcolor2" value="ff00ff">
</applet>
</BODY>
</HTML>
```



CGI



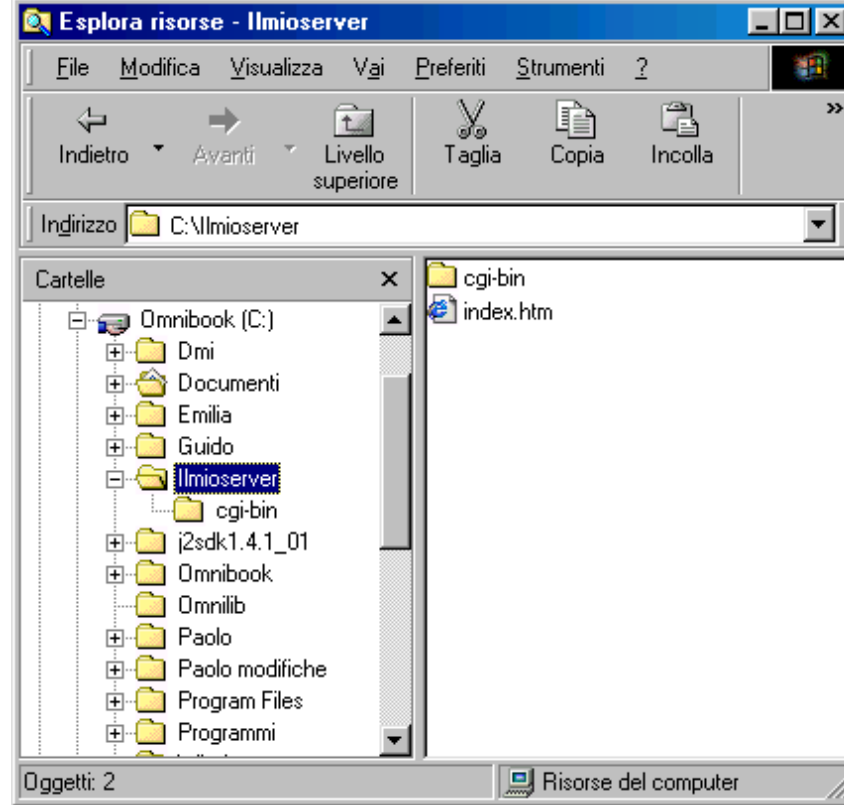
CGI

<http://www.pippo.it>

Setup del Web server:
c:\ilmioserver
è la directory radice

Quando il server riceve una
richiesta corrispondente all'URL
<http://www.pippo.it>
cerca un file di default
nella directory radice
(index.html, index.htm, default.html, default.htm).

Ad ogni richiesta di pagina (<http://www.pippo.it/paginax.htm>)
Il server trasmette la pagina.

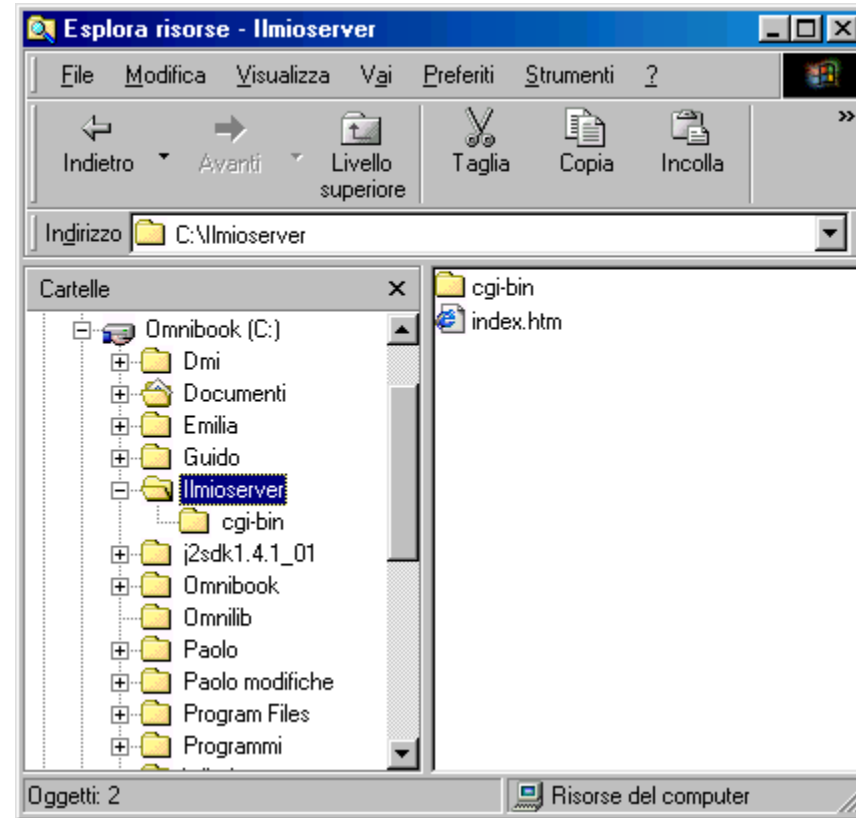


CGI

Nella directory radice è creata una directory **cgi-bin**.

Quando il server riceve una richiesta di un file in cgi-bin

non serve il file ma lo manda in esecuzione e serve l'output dell'esecuzione.

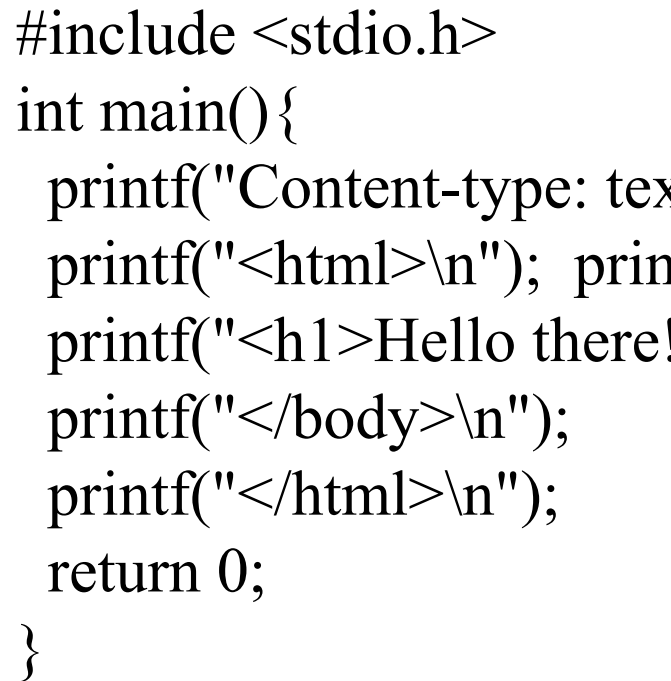


CGI

```
<html> <body>  
<a href="http://www.pippo.it/ilmioserver/cgi-bin/esempio.cgi">  
attiva cgi</a>  
</body></html>
```

```
HTTP/1.1 200  
Content-type: text/html  
...
```

```
<html>  
<body>  
<h1>Hello there!</h1>  
</body>  
</html>
```



```
#include <stdio.h>  
int main() {  
    printf("Content-type: text/html\n\n");  
    printf("<html>\n"); printf("<body>\n");  
    printf("<h1>Hello there!</h1>\n");  
    printf("</body>\n");  
    printf("</html>\n");  
    return 0;  
}
```

CGI

```
<html> <body>  
<a href="http://www.pippo.it/ilmioserver/cgi-bin/esempio.cgi">  
attiva cgi</a>  
</body></html>
```



```
#include <stdio.h>
```

```
int main() {
```

**Il programma può includere
qualsiasi procedura
che accede a banche dati o interagisce
con il sistema informativo esistente**

```
return 0;
```

```
}
```

...



- Invio di dati

- Metodo GET

- Nomi e valori sono appesi alla URL

```
http://nome.server/cgi.bin/programma?nome=valore&nome=valore...
```

- Metodo POST

- Nomi e valori sono nel corpo del messaggio

```
nome=valore&nome=valore...
```

CGI

- Raccolta dati da client via form
- Method (POST | GET) (*case insensitive*): identifica il metodo HTTP utilizzato per trasmettere i dati della form
- Action: indirizza l'applicazione che elabora i dati

...

```
<body>
```

```
<form method="POST"
```

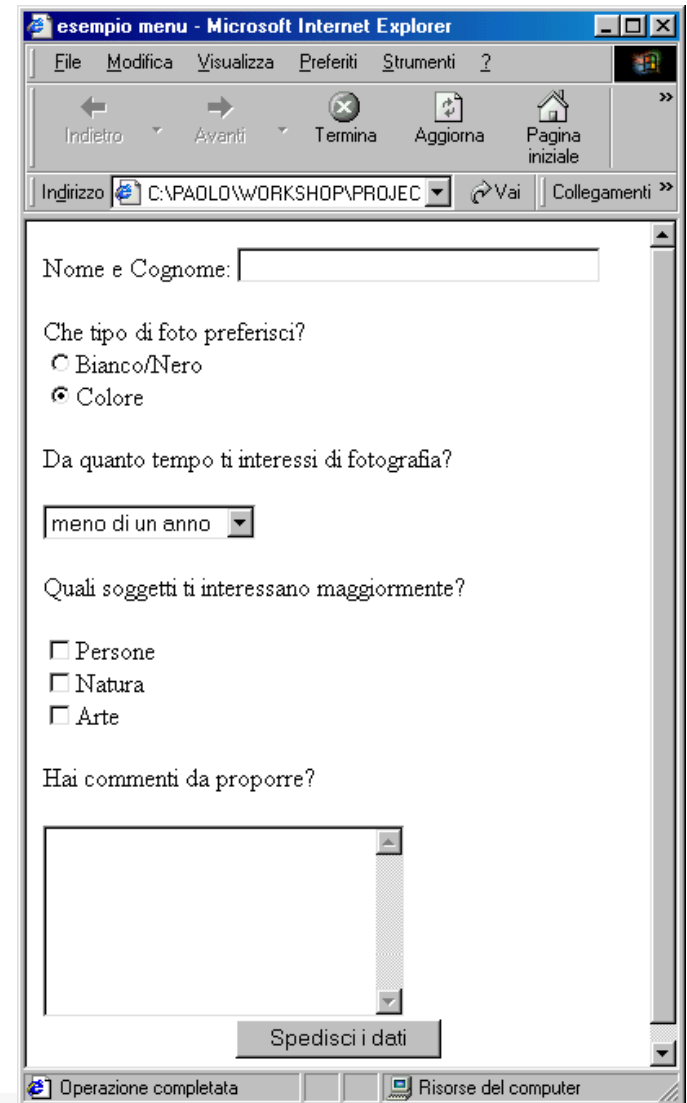
```
action="http://nome.server/cgi.bin/programma">
```

.....

```
</form>
```

```
</body>
```

...

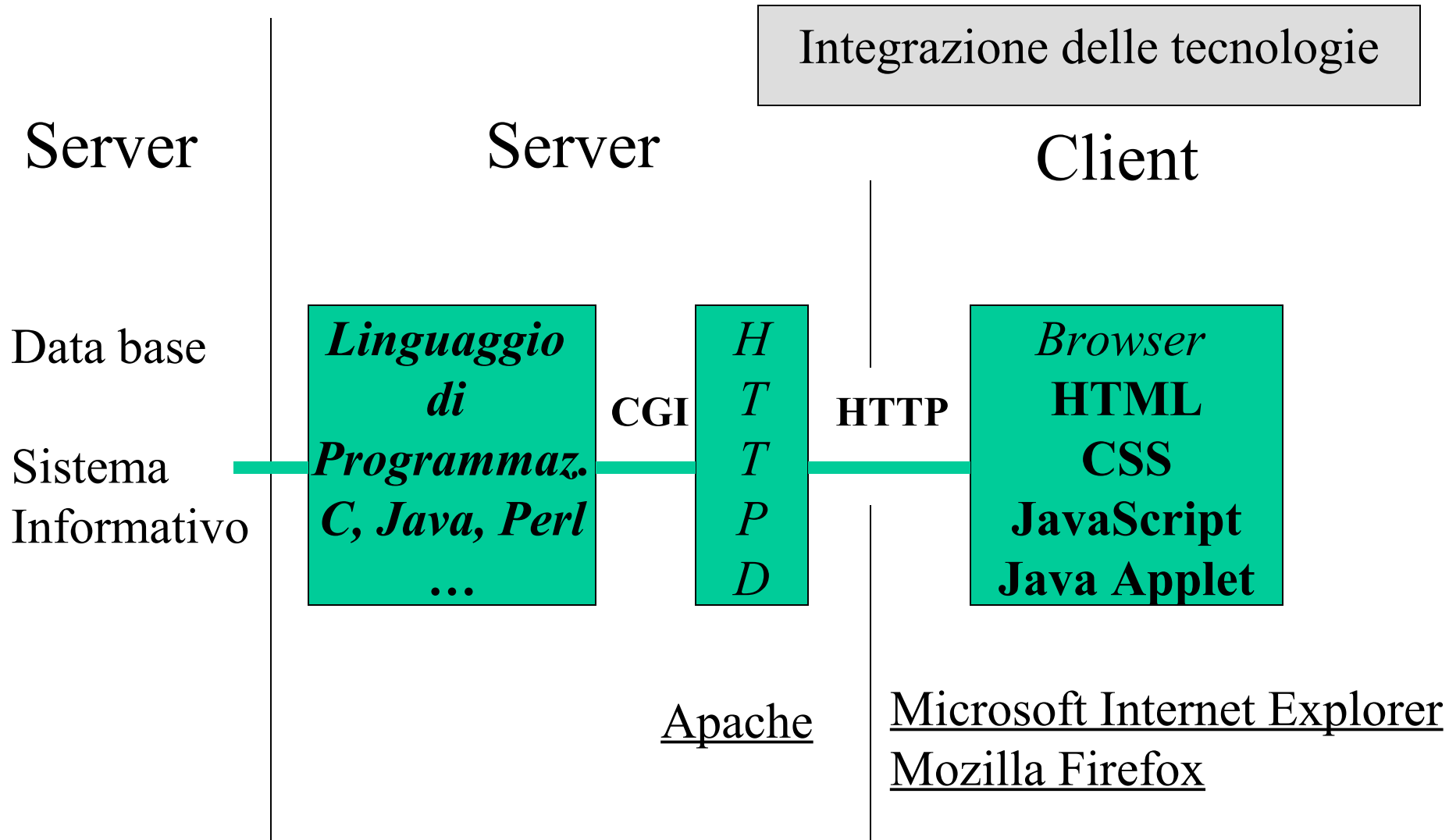


CGI

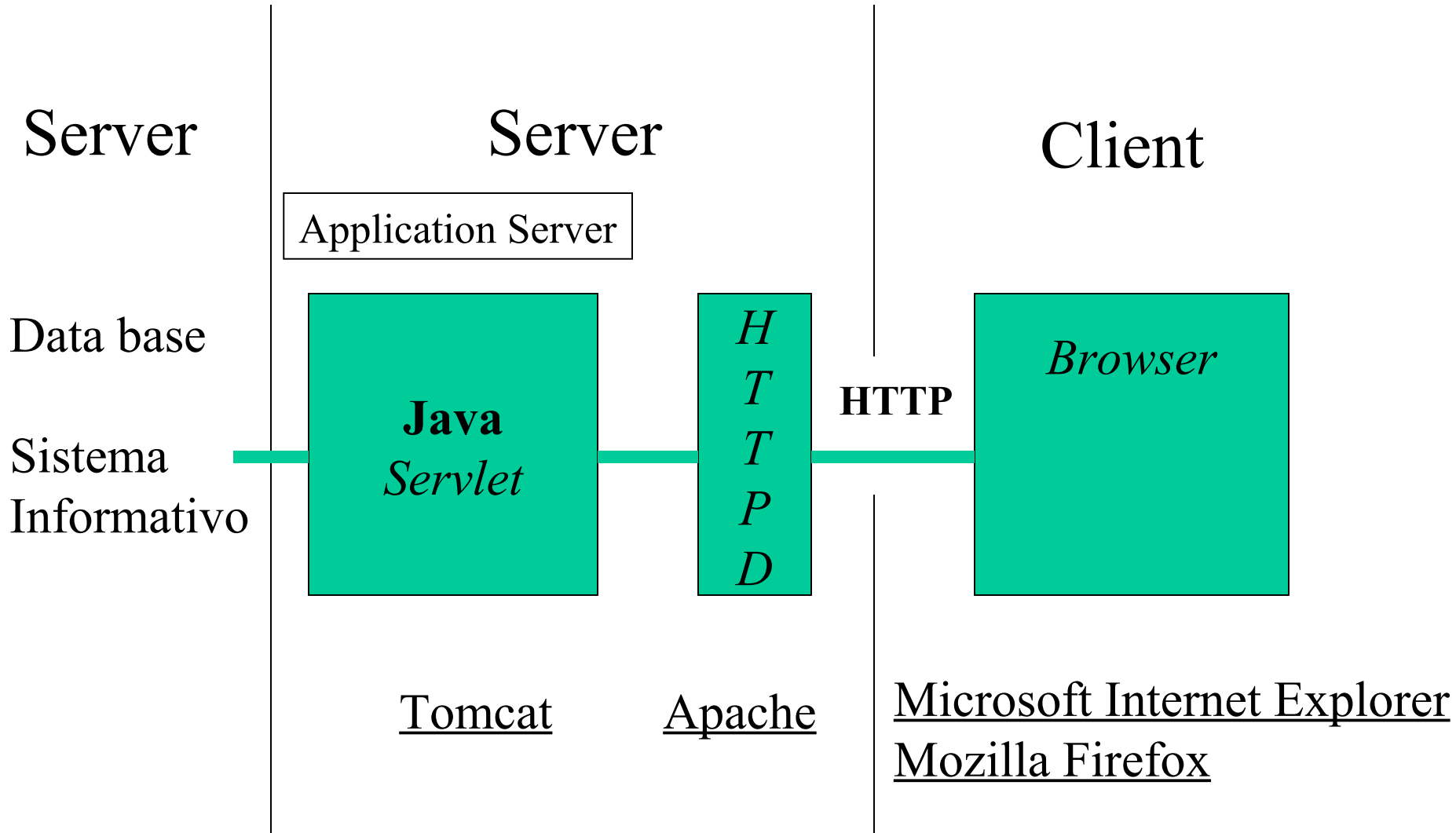
- ...
 - CONTENT_LENGTH
 - CONTENT_TYPE
 - GATEWAY_INTERFACE
 - HTTP_ACCEPT
 - HTTP_USER_AGENT
 - PATH_INFO
 - PATH_TRANSLATED
 - QUERY_STRING
 - REMOTE_ADDR
 - REMOTE_HOST
 - REMOTE_IDENT
 - REMOTE_USER
 - REQUEST_METHOD
 - SCRIPT_NAME
 - SERVER_NAME
 - SERVER_PORT
 - ...
- Variabili di ambiente
 - Il programma attivato può accedere ad un insieme di variabili di ambiente
- Contiene i dati passati dal client con il metodo GET
- Contiene il metodo usato: GET o POST

- Limiti
 - Ogni richiesta al server via CGI genera un processo che termina a fine elaborazione
 - Se il programma attivato via CGI contiene un accesso a DataBase, ogni volta sarà aperta e chiusa una connessione al DBMS
 - Utilizzo di risorse elevato → problemi di scalabilità (es. applicazione B2C)

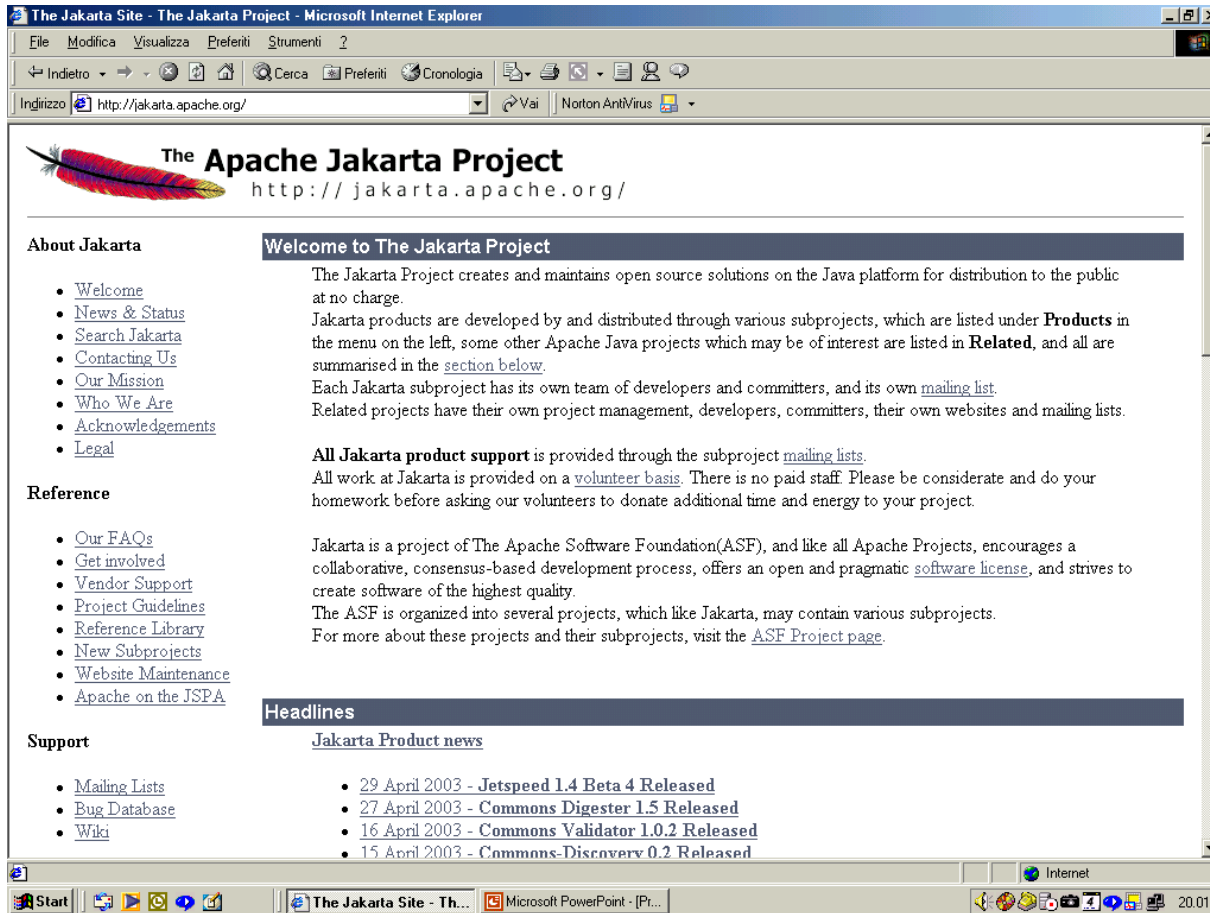
Tecnologie (standard aperti + Sun Java)



Java Servlet



Java Servlet



Tomcat

<http://jakarta.apache.org>

- Servlet
 - Oggetti Java particolari che sono caricati ed eseguiti dal web server che li utilizza come proprie estensioni.
 - Il web server mette a disposizione delle Servlet un “container” che si occupa della gestione dell’ambiente all’interno delle quali le servlet girano.
 - Il “container” passa i dati dal client verso le servlet e viceversa.

- Package `javax.servlet`
 - Package di base delle Servlet API
 - Contiene le classi per definire Servlet standard indipendenti dal protocollo.
 - Una Servlet generica è una classe definita a partire dall'interfaccia Servlet contenuta all'interno del package `javax.servlet`.

- Package `javax.servlet.http`
 - Le classi del package estendono le funzionalità di base di una servlet supportando tutte le caratteristiche della trasmissione di dati con protocollo http compresi cookies

- **Ciclo di vita di una Servlet**
 - Una servlet è caricata e inizializzata dal web server al momento della prima richiesta http da parte di un client o al momento dell'avvio del servizio.
 - All'inizializzazione la servlet può, se il caso, aprire una connessione con un DataBase (connessione che resta aperta per servire più richieste del client).

- Ciclo di vita di una Servlet
 - La servlet esegue le richieste del client.
Per ogni richiesta è creato un thread.
 - La servlet rilascia le risorse ed è distrutta

Java Servlet

HelloWorldServlet.java

....

```
import javax.servlet.* ;
```

```
import javax.servlet.http.* ;
```

```
public class HelloWorldServlet extends HttpServlet {
```

```
    public void service (HttpServletRequest req, HttpServletResponse res)
```

```
    throws ServletException, IOException {
```

```
        res.setContentType("text/html");
```

```
        ServletOutputStream out = res.getOutputStream();
```

```
        out.println("<html>");
```

```
        out.println("<head><title>Hello World</title></head>");
```

```
        out.println("<body>");
```

```
        out.println("<h1>Hello World</h1>");
```

```
        out.println("</body></html>");
```

```
    }
```

```
}
```

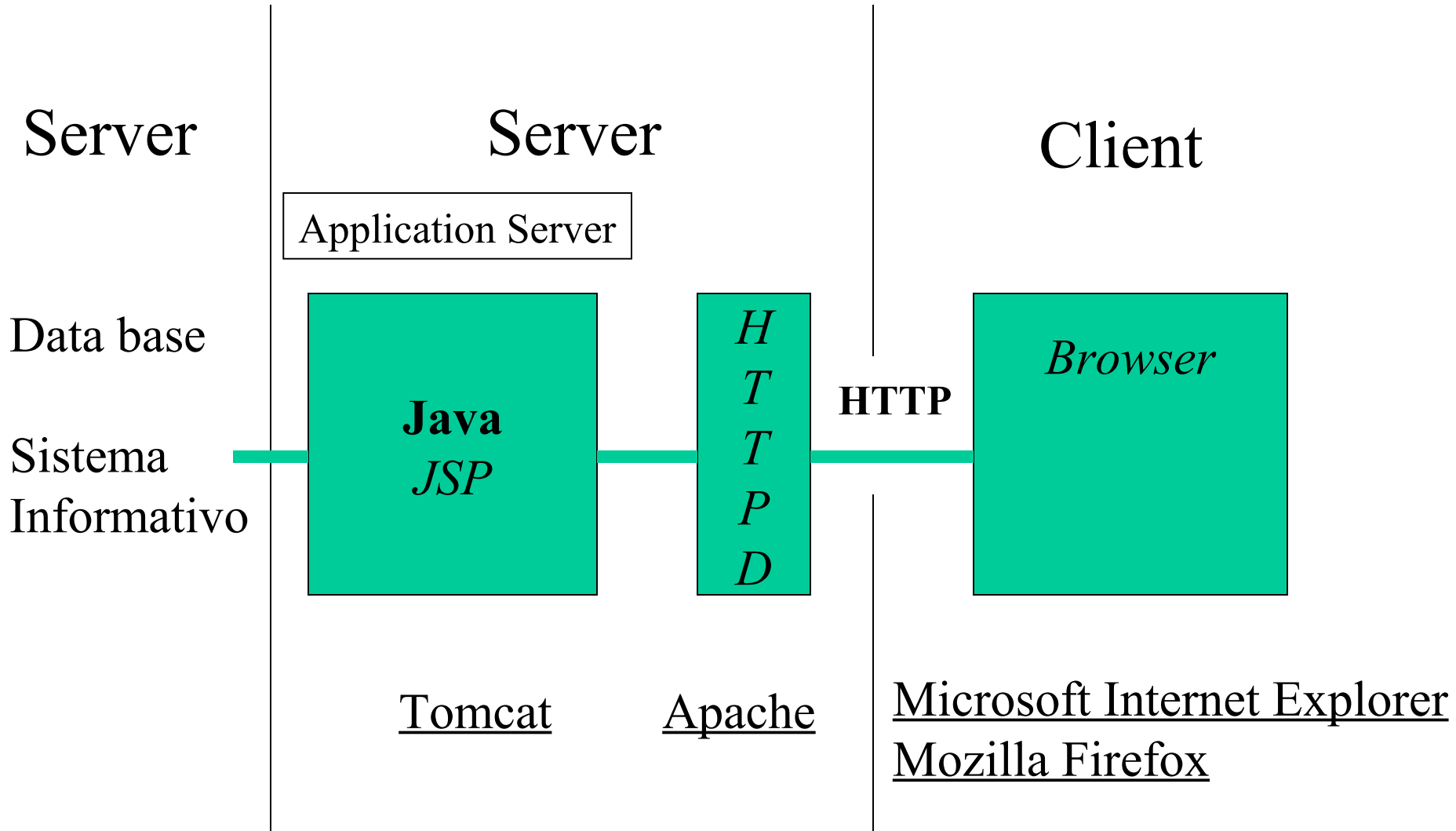
Metodo invocato al momento dell' arrivo di una richiesta da parte di un client

Java Servlet

- Application server

Tomcat	Open Source
WebLogic	BEA
Websphere	IBM
Sun ONE	Sun Microsystems
.....	

Java Server Pages



- Java Server Pages (JSP)
 - Pagina di testo HTML che include codice java
 - Pagina dinamica che è eseguita lato server e restituisce HTML al client

- Scripting lato client
 - Scopo: rendere dinamica l'interfaccia utente
 - Cambiare a seguito di eventi la presentazione
 - Gestire l'input dell'utente
 - ...
- Scripting lato server
 - Scopo: connettersi a programmi e basi dati e modificare di conseguenza la pagina inviata

- Java Server Pages (JSP)
 - Il server utilizza la pagina JSP come una servlet
 - Al momento del primo accesso la pagina JSP è trasformata in un file Java e compilata ottenendo una classe Java che rappresenta una servlet di tipo HttpServlet che crea una pagina html e la invia al client.

- Java Server Pages (JSP)
 - Possibile integrazione tra servlet e JSP: il client invia una richiesta ad una servlet che invia gli oggetti prodotti ad una JSP la quale genera l'HTML e lo invia
 - JSP: View
 - Servlet: Controller
 - Componenti Java: Model

Java Server Pages

esempio1.jsp

```
<html>
```

```
<body>
```

```
<h1> Informazioni sulla richiesta http </h1>
```

```
<br>
```

```
Metodo richiesto : <%= request.getMethod() %>
```

```
<br>
```

```
URI : <%= request.getRequestURI() %>
```

```
<br>
```

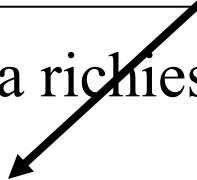
```
Protocollo : <%= request.getProtocol() %>
```

```
<br>
```

```
<body>
```

```
</html>
```

Espressioni :iniziano con la sequenza di caratteri “<%=” e terminano con la sequenza “%>”
Utilizzate per aggiungere alla risposta http il risultato dell'esecuzione di un'espressione di scripting



Java Server Pages

Esempio2.jsp

```
<html>
```

```
<body>
```

```
<% for(int i=0; i<10; i++) {
```

```
    if(i%2==0) {
```

```
        %>
```

```
        <h1>Pari</h1>
```

```
    <% } else { %>
```

```
        <h2>Dispari</h2>
```

```
    <%
```

```
        }
```

```
    }
```

```
    %>
```

```
</body>
```

```
</html>
```

Scriptlet:

iniziano con la sequenza “<%”,
terminano con la sequenza “%>”

Possibilità si spezzare il codice Java contenuto all' interno delle scriptlet per dar modo al programmatore di non dover fondere tag html all'interno del sorgente Java.

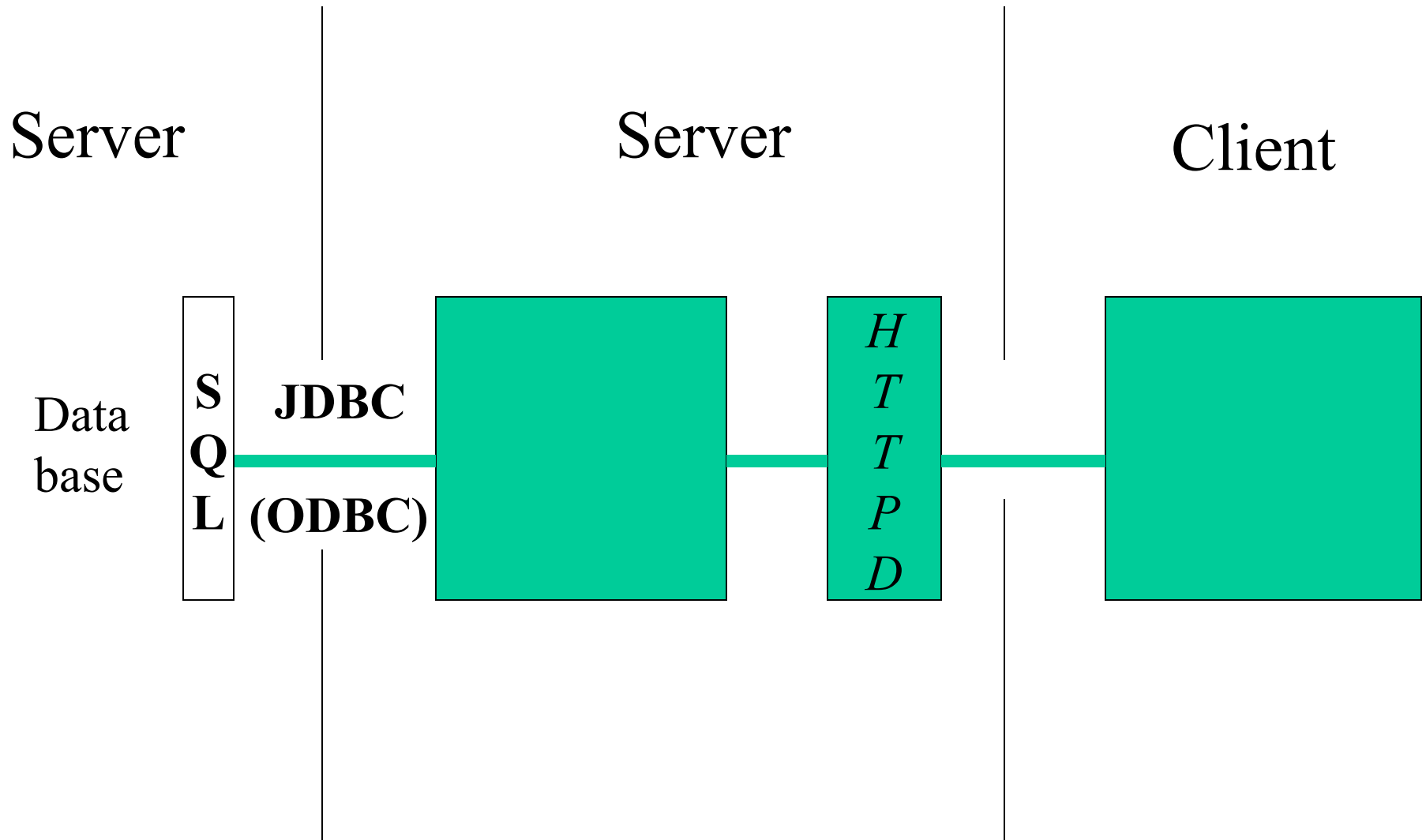
Server Side Include (.shtml)

- Apache SSI (Server Side Includes)
 - Pagina di testo HTML che include direttive valutate dal server
 - Lo script è eseguito sul server alla richiesta della pagina e modifica dinamicamente la pagina che è trasmessa al client
 - Utile per inserire parti dinamiche limitate (altrimenti si utilizza CGI o JSP)

Server Side Include (.shtml)

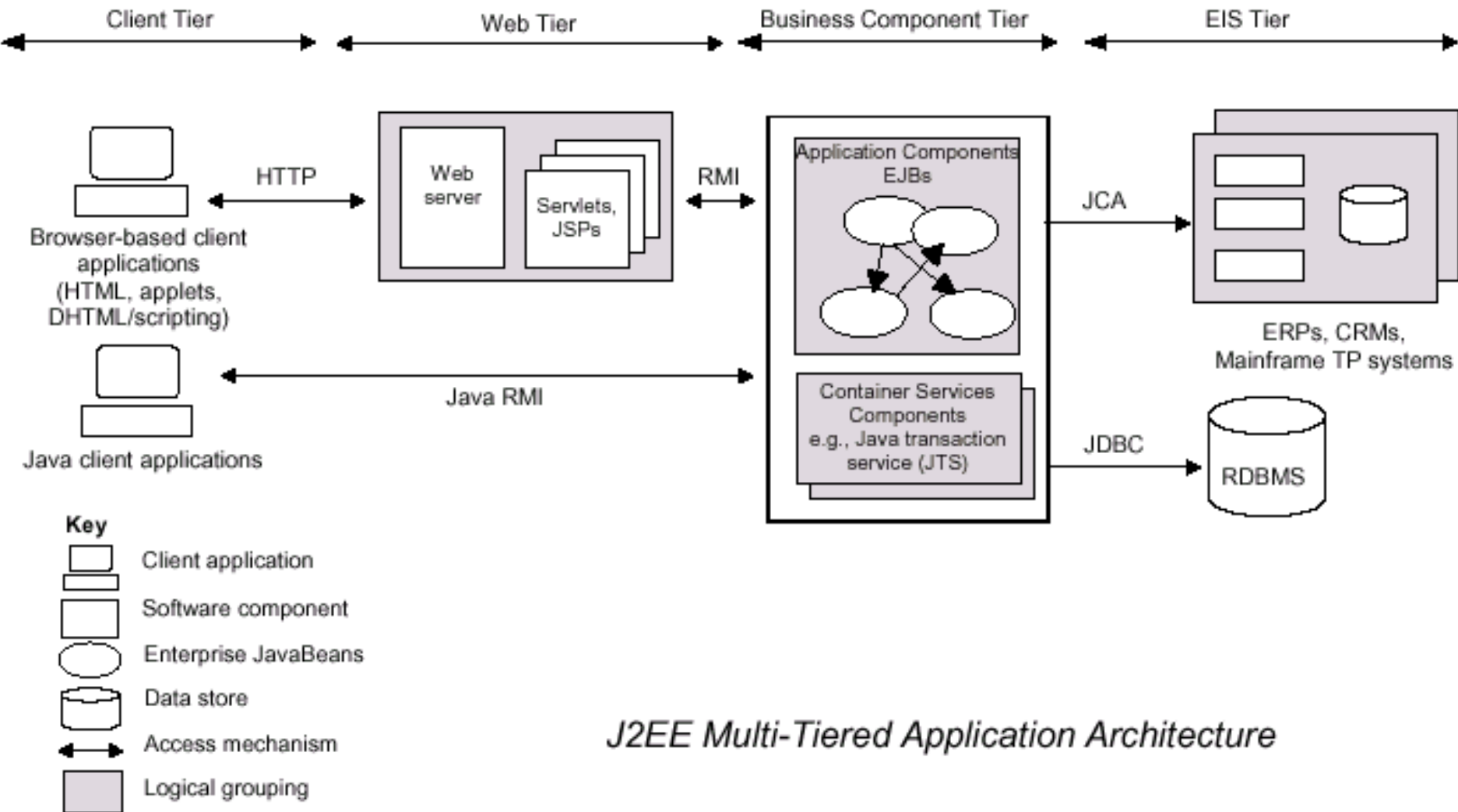
- Apache SSI (Server Side Includes)
 - SINTASSI DELLE DIRETTIVE:
`<!--#element attribute=value attribute=value ...-->`
 - Formattato come commento HTML; se il server non è abilitato non lo elabora ma è visibile nel sorgente HTML
 - Il file che include le direttive deve avere estensione .shtml
 - Esempio: produce la data corrente
`<!--#echo var="DATE_LOCAL" -->`

Connessione con i Data Base



- ODBC (Open DataBase Connectivity)
 - Standard internazionale di accesso a Database Relazionali
- JDBC
 - Interfaccia ad oggetti java di accesso a sorgenti di dati
 - Connessa ai singoli driver di DataBase attraverso una interfaccia nativa o attraverso un ponte verso ODBC

J2EE



J2EE Multi-Tiered Application Architecture