

Informatica **U**manistica

# Laboratorio di Progettazione Web Applicazioni Web

*AA 2010/2011*

*Claudio Lucchese*

*ISTI - CNR – [claudio.lucchese@isti.cnr.it](mailto:claudio.lucchese@isti.cnr.it)*



UNIVERSITÀ DI PISA





L'Alta Velocità rivoluziona il modo di viaggiare. Sali a bordo.

ULTIME NOTIZIE > Roma Termini

FS News

WebRadio

Il Gruppo

Trenitalia

Alta Velocità

RFI

Italferr

Ferservizi

Per chi viaggia

Biglietti | Abbonamenti | Hotel | Offerta A/R

Da dove vuoi partire?

Dove vuoi arrivare?

Quando vuoi partire?

giorno 09 mese 03 anno 2009

ore 11 minuti 46

Invia

Scopri i pass InterRail | Acquista un pass InterRail

Area riservata

User-ID

.....

Registrati ora!

Entra

Servizio Clienti

- I Call Center
- Servizi on line
- Assistenza clienti
- In caso di sciopero



In primo piano

Sali a bordo!

10% di sconto per i tuoi viaggi



35% di sconto

sui viaggi di A/R in giornata

ACQUISTA ORA



La "nuova metropolitana veloce" d'Italia

NAPOLI-MILANO

I Tempi cambiano



IL GESTORE DELL'INFRASTRUTTURA FERROVIARIA

**ViaggiaTreno**

Segui in tempo reale il percorso del tuo treno

ALTA VELOCITA'. UN ALTRO RECORD



ACCANTO AL CLIENTE in ogni momento del viaggio

Mobile Ticketing



Acquista l'abbonamento AV

Viaggi illimitati al prezzo di 10 corse Risparmi fino al 75%

Home > Orari e acquisto

- Cambio prenotazione
- Rimborso on-line
- PostoClick
- Ticketless
- Biglietto regionale on-line
- Abbonamenti integrati
- Guida all'acquisto

SCELTA DEL TRENO | Prezzo | Acquisto

Stazione di partenza	Stazione di arrivo	Data
Pisa ( Tutte Le Stazioni )	Palermo ( Tutte Le Stazioni )	9/3/2009

**PROPOSTE DI VIAGGIO**

Per orario di partenza | Per tempo di percorrenza | Per numero di cambi

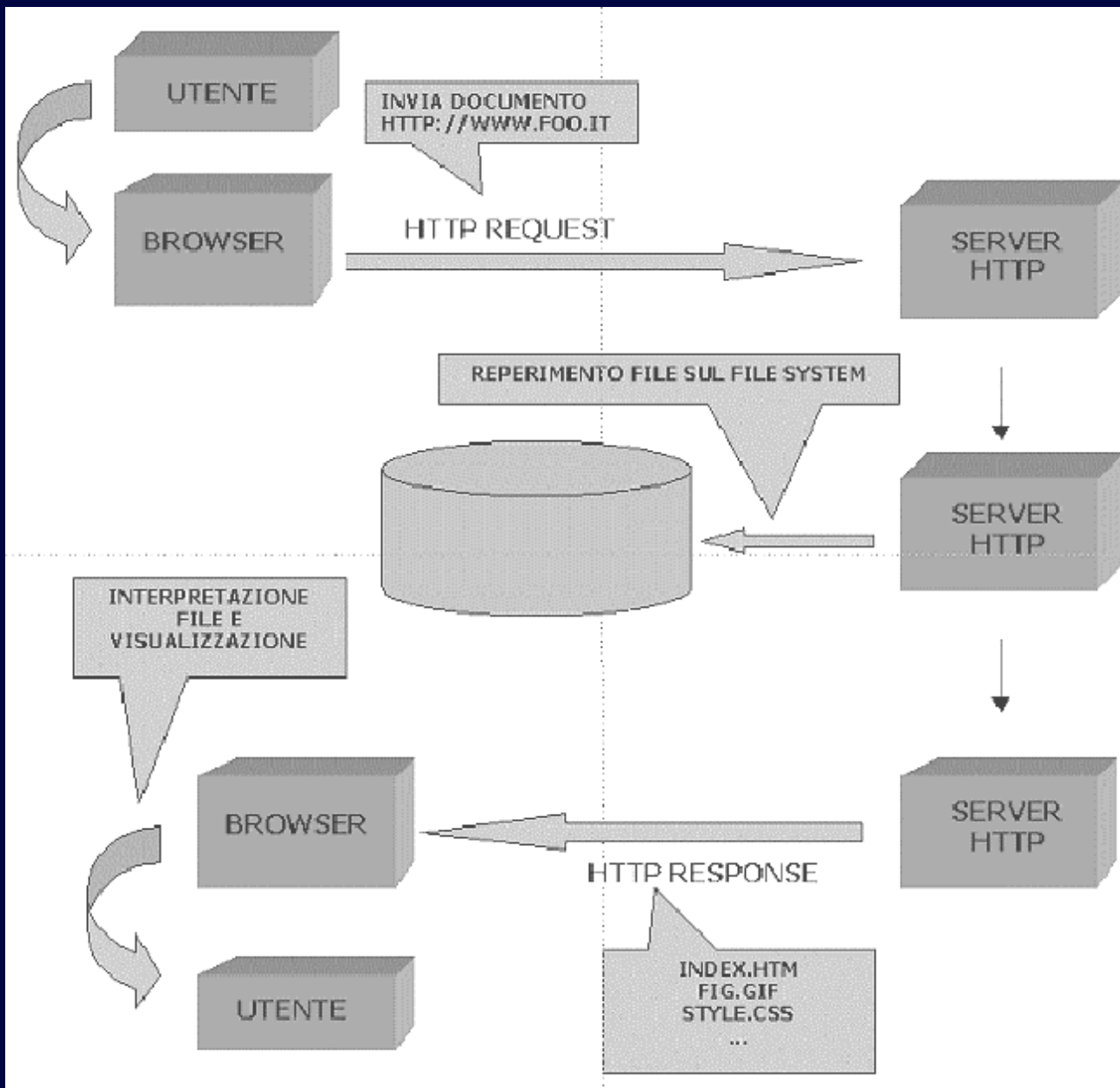
Partenza	Arrivo	Durata	Stazione di Cambio		N° Treno	Categoria Treno	1 classe**	2 classe**	Info	Acquista
			Stazione	Arrivo						
17:00 PISA C.	10:00 PALERMO	17:00			9799	<i>ES<sub>city</sub></i>	ND	ND	<a href="#">Dettagli</a>	
			ROMA TE	20:14	1939 Bellini	E				
			ME C.LE	06:40	3873	R				
20:47 PISA C.	13:13 PALERMO	16:26			1941 Treno del Sole	E	ND	ND	<a href="#">Dettagli</a>	
00:50* PISA C.	17:36 PALERMO	16:46			761 Scilla	<i>ICN</i>	ND	ND	<a href="#">Dettagli</a>	
			V.SGIOV	12:16	29					
			ME MAR.	13:25						
02:19* PISA C.	19:06 PALERMO	16:47			809	E	ND	ND	<a href="#">Dettagli</a>	
			ROMA TE	05:51	723 Peloritano	<i>ICplus</i>				

# Architettura Client-Server



# Protocollo HTTP

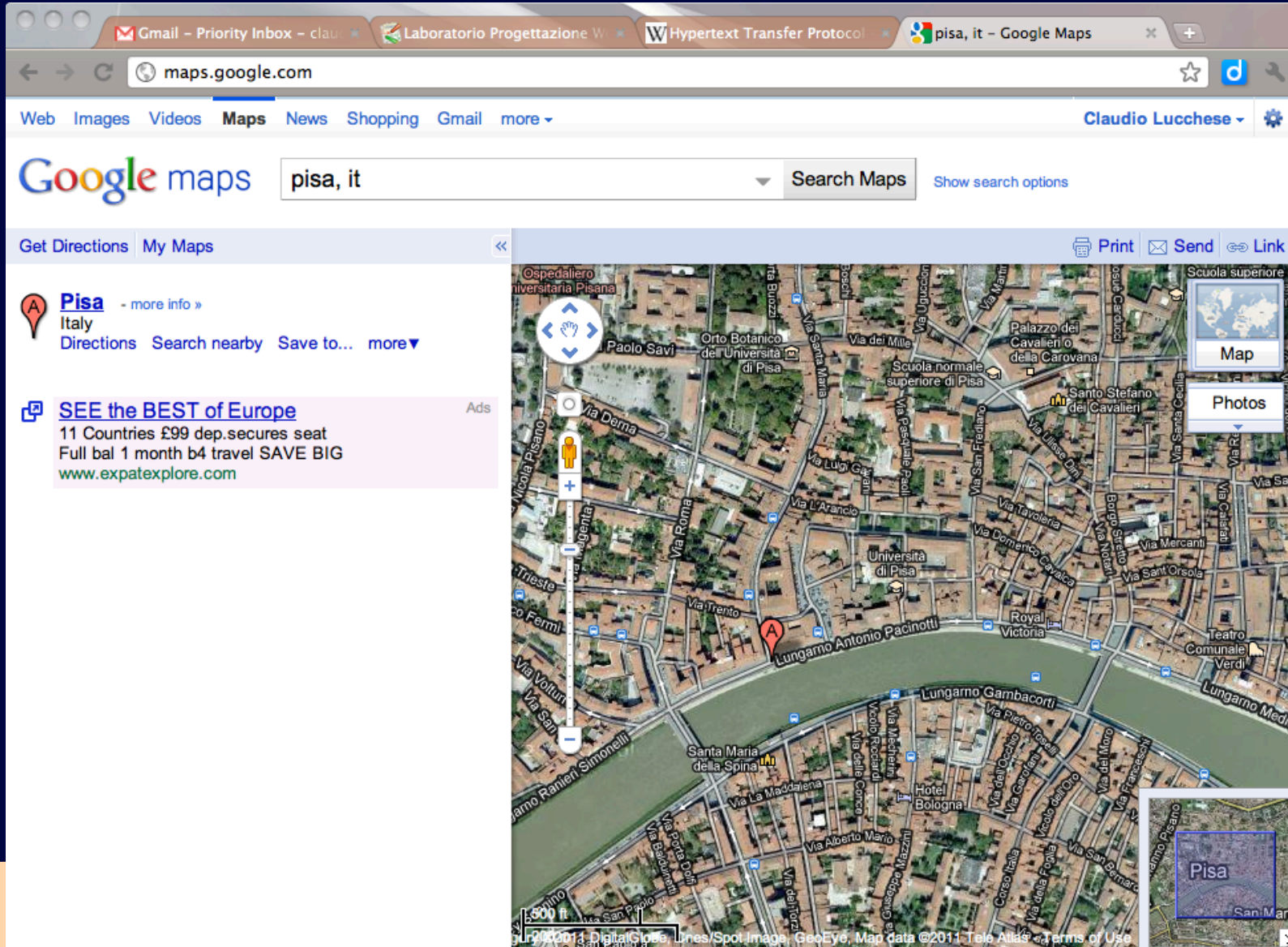
Per sviluppare una applicazione web occorre aggiungere **potere di calcolo** e **memorizzazione dati** a questo modello



# Componenti delle applicazioni web

- L'infrastruttura web si basa sul protocollo HTTP:
  - Hypertext Transfer Protocol
- Ma questo non permette lo sviluppo di *applicazioni complesse* che coinvolgano una fase di elaborazione oltre che di passaggio di dati. Occorre costruire una architettura più complessa di quella client/server basato su HTTP

# Non tutte le pagine rispecchiano strettamente questo modello di interazione





# Non tutte le pagine rispecchiano strettamente questo modello di interazione

The image shows a screenshot of a Google search results page for the query "oscars". The browser's address bar shows the URL: [www.google.com/search?hl=en&tbs=mbl:1&q=oscars&aq=f&aqi=g-z1g1g-z2g6&aql=&oq=](http://www.google.com/search?hl=en&tbs=mbl:1&q=oscars&aq=f&aqi=g-z1g1g-z2g6&aql=&oq=). The search results are displayed in a grid-like format with several columns. On the left, there are navigation tabs for "Everything", "Images", "Videos", "News", "Shopping", "Realtime", and "More". The main content area is divided into sections: "New results will appear below as they become available. Pause", "Timeline" (with a graph showing activity from 12:00 AM to 6:00 PM), "Top links", and "Top updates". The search results include various tweets and news snippets, such as "LivingSocial Check out livingsocial.com for deals in your town to get yourself #Oscars red carpet ready bit.ly/dWIJPs", "StyleWithHeart Livia looked stunning in her eco gown - www.vogue.co.uk/blogs/livia-firth/...", "glaceaunwest oscars fashion alert - cate blanchett, major thumbs up, stunning in givenchy Catwalk Queen: 2011 Oscars: Cate Blanchett in Givenchy Couture - catwalkqueen.tv", "NewsNowBreaking Oscars Red Carpet: The Wonderful, The Wacky, The Whoa ... - newsbreakingonline.com", "DCMVtv Best of 2011 Oscars: Natalie Portman, Jeremy Renner, Scarlett Johansson, Kevin Spacey and more hit the red car... Celebrity Videos | E! Online - eonline.com", and "lisagastineau @ToniFerrara we're laying around in the hotel watching TV! He's has to be up in hour-(actually-".

# I Livelli in un'applicazione web

Un'applicazione Web si sviluppa su tre livelli logici, ovvero regole di buona progettazione

- ◆ presentazione
- ◆ intermedio
- ◆ dati

Non sempre i livelli logici di un'applicazione Web corrispondono a locazioni fisiche sulla rete

# Livello Presentazione

Il livello di presentazione costituisce l'interfaccia utente dell'applicazione Web e corrisponde al client.

Esso è costituito da vari componenti combinati tra loro: browser, documenti HTML, applet Java, controlli ActiveX.

# Livello Dati

Il livello dati fornisce servizi non direttamente disponibili tramite il Server Web; questi servizi sono generalmente forniti da applicazioni indipendenti dall'ambiente Web.

Tipici esempi di applicazioni presenti a questo livello sono:

- Server dati (DBMS).
- Server di mail (POP, SMTP).
- Server di documentazione elettronica.

# Livello intermedio

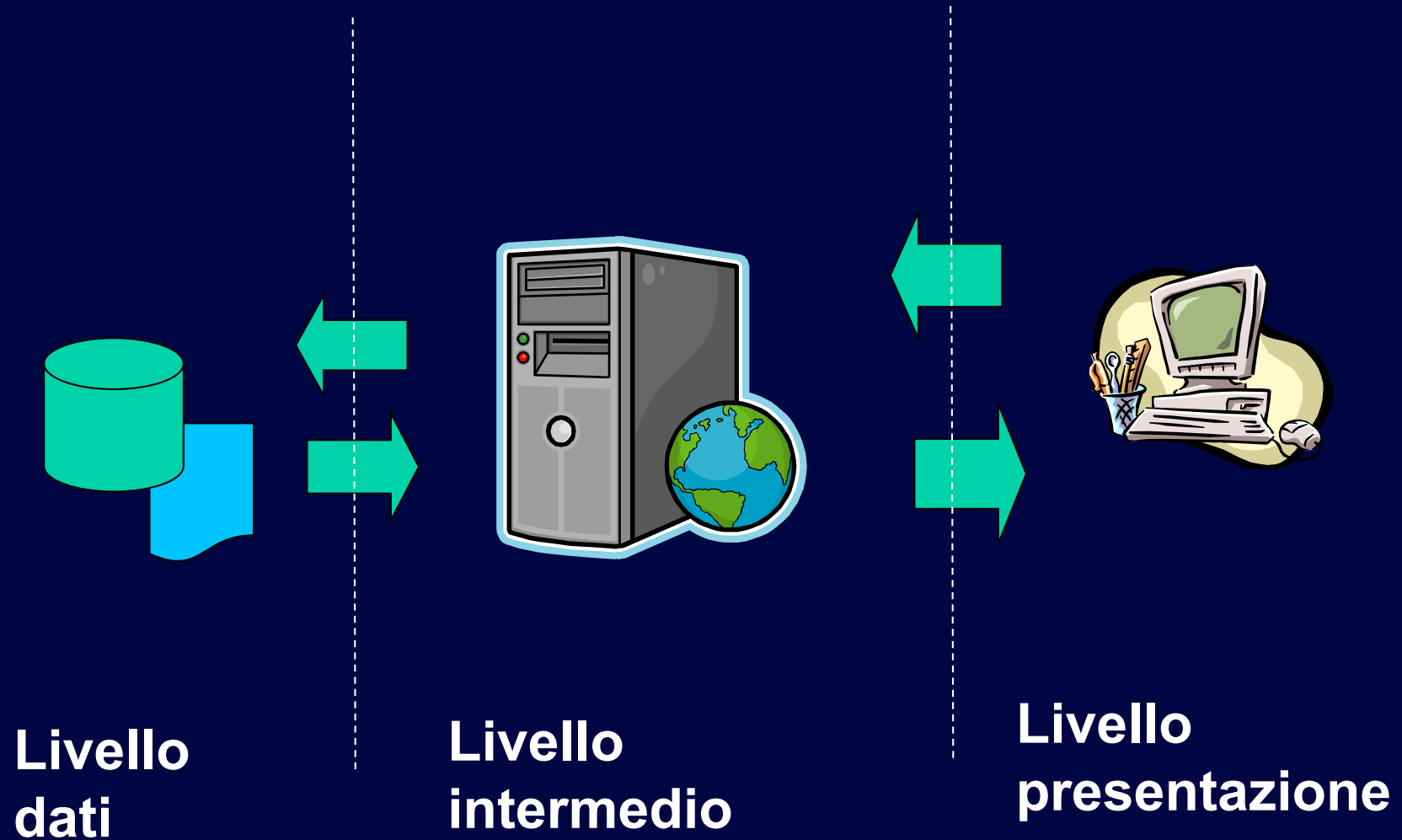
Il livello intermedio di un'applicazione Web corrisponde alla componente elaborativa dell'applicazione:

Le modalità di realizzazione del livello intermedio dipendono dalle caratteristiche e dalle tecnologie supportate dal server Web : può essere costituito da un insieme di script, componenti e programmi interagenti tra di loro e con il server Web.

Alcune delle tecnologie più comuni:

Common Gateway Interface (CGI), Internet Server Application Interface (ISAPI), Active Server Pages (ASP) , PHP, Java Servlet, ....

# I livelli di una applicazione web



# Tecnologie web

**Sia i livelli presentazione che intermedio possono avere una componente elaborativa.**

**La tecnologia utilizzata ha caratteristiche profondamente diverse a seconda che il livello interessato sia presentazione o intermedio.**

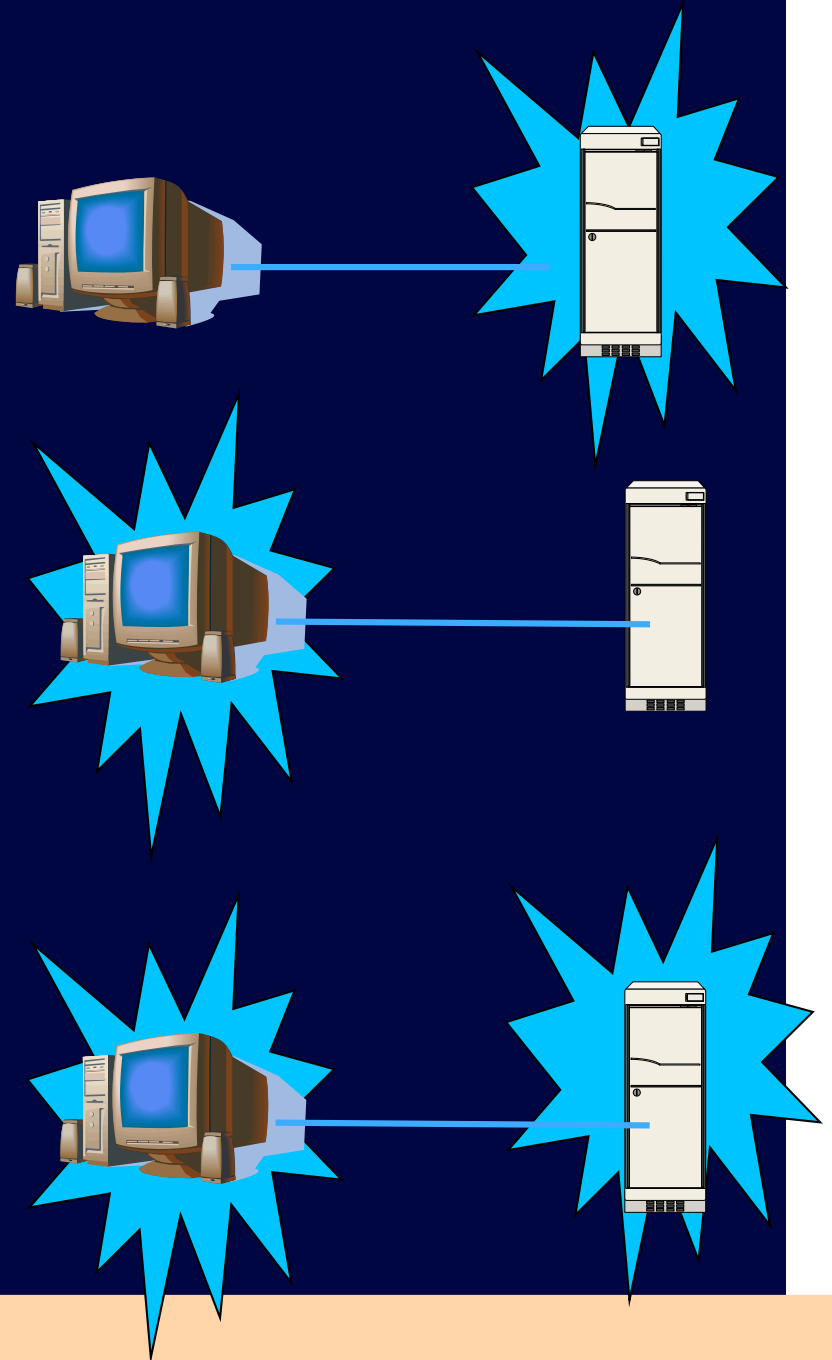
**Le tecnologie web impiegate in un'applicazione web si chiamano lato server (server-side) e lato client (client side)**

# Tecnologie client-side e server-side

**tecnologia server side:** il peso della computazione risiede tutta sul server

**tecnologia client-side** quando la computazione avviene principalmente sul browser.

Generalmente le applicazioni web complesse usano strategia **ibrida**





# Tecnologia Client Side

- ◆ E' la parte di programmazione che coinvolge il browser web ed è quindi **dipendente dal tipo (e versione) del browser** usato. Alcuni linguaggi client-side non sono supportati da alcuni browser – ad es. VBscript è supportato solo da Internet Explorer. Uno standard è Javascript.
- ◆ Uno script può comportarsi in modo diverso nei vari browser web.
- ◆ Tipicamente le operazioni effettuate client side sono di interazione stretta con l'utente – ad es. rollover sui tasti o il controllo di dati inseriti tramite FORM.
- ◆ Generalmente abbiniamo alla programmazione client side compiti semplici e di interazione immediata con l'utente.

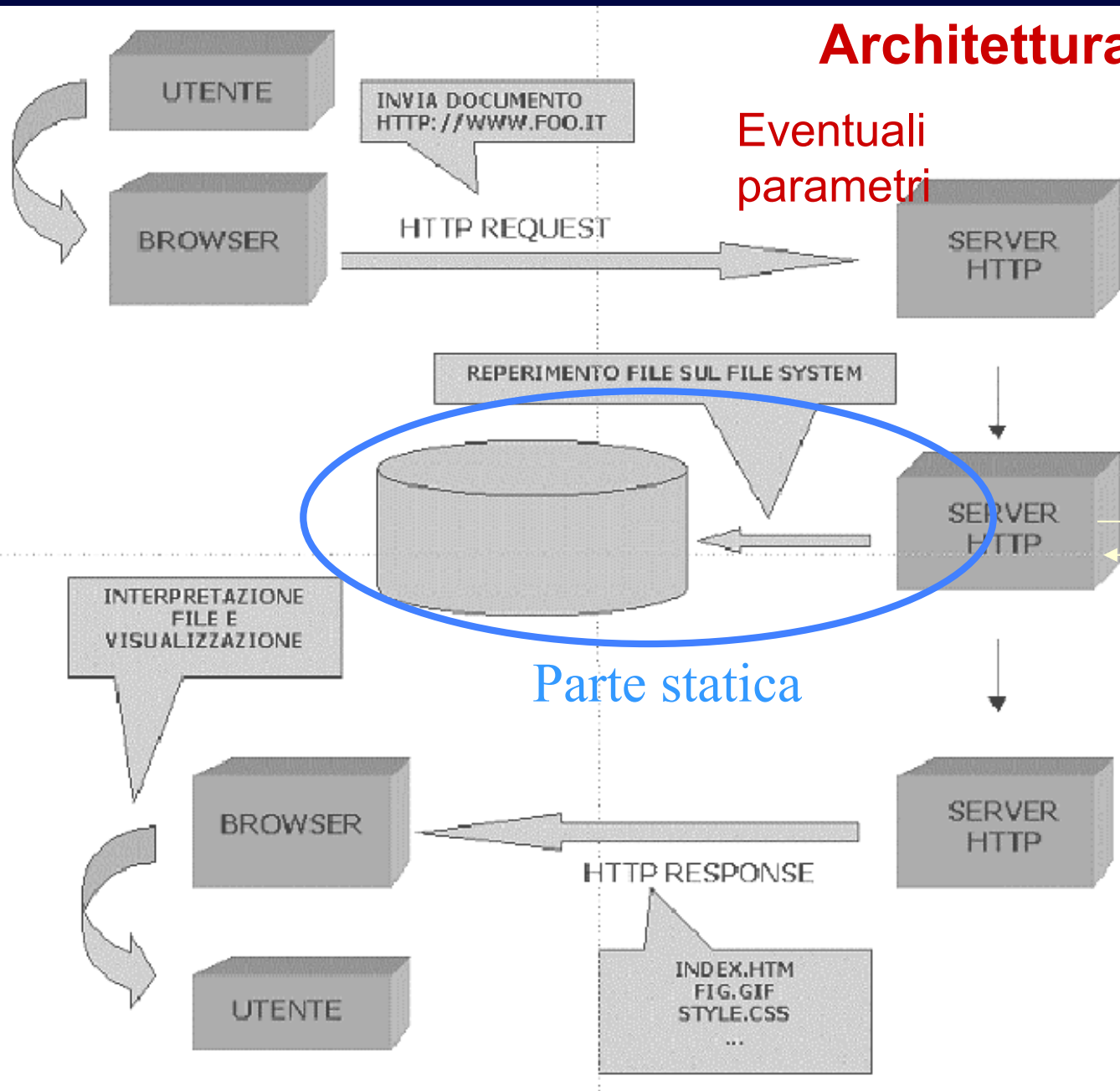
# Tecnologia Server Side

- ◆ La programmazione server side si presta a maggiori possibilità e a **complessità di computazione maggiore, nonche' all'immagazzinamento dei dati.**
- ◆ L'interattività non è immediata in quanto è il server che esegue la parte di programma.
- ◆ Avere la possibilità di **programmare la parte server** ci apre la possibilità per far fare al server qualcosa in più del semplice restituire pagine HTML che già risiedono sul disco.
- ◆ La maggior parte dei siti web che consultiamo ogni giorno sono di fatto applicazioni server-side con parte meno rilevante client-side: portali, commercio elettronico, blog.....

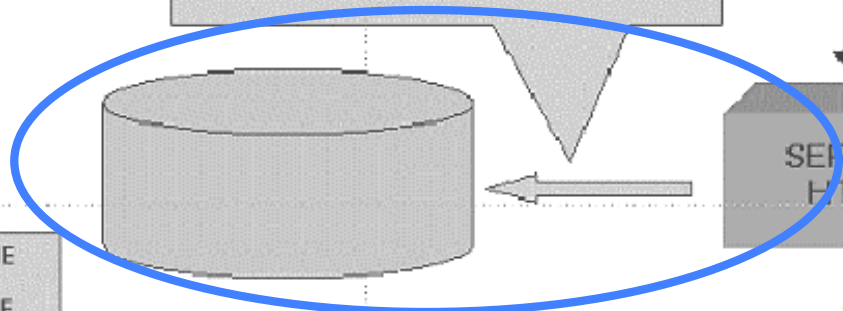
# Tecnologia Server Side

- ◆ La tecnologia server-side permette di creare pagine **dinamiche**, ovvero pagine che non sono presenti effettivamente sul server al momento della richiesta da parte del browser, ma che vengono **create e composte dinamicamente dal server web**, eventualmente servendosi del livello dati, dietro richiesta esplicita da parte del browser.
- ◆ Chiaramente il server web deve essere dotato di tecnologia in grado di “costruire” su richiesta le pagine, assemblando insieme i dati e spedirle al browser.

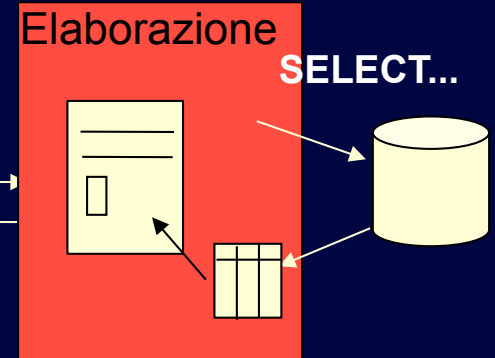
# Architettura Server-Side



Eventuali parametri

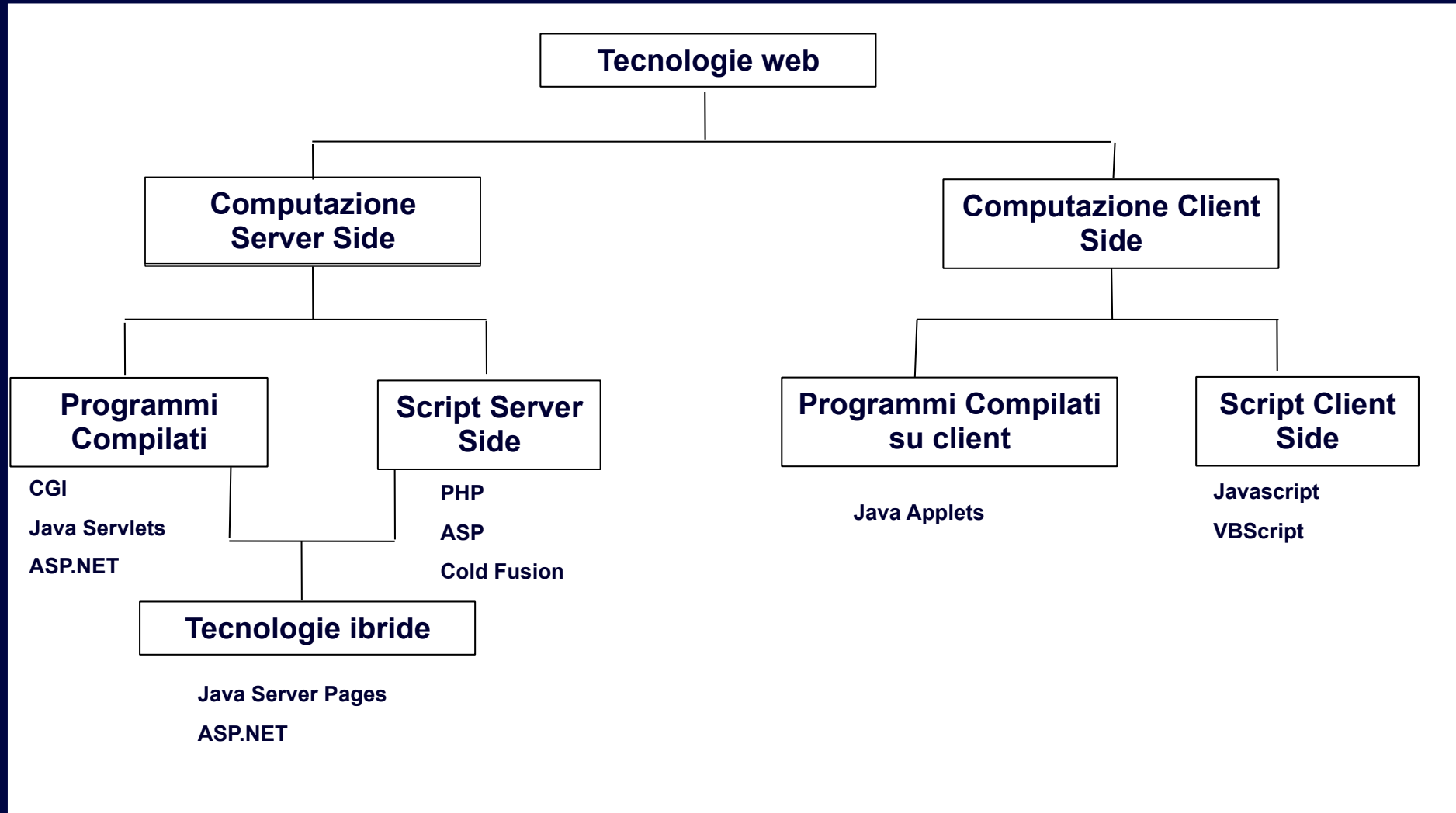


Parte statica



Parte dinamica

# Tecnologie client/server side



# Tecnologie Server side

- L'utente può interagire con il server ad esempio sottomettendo dati che il server elabora per restituire poi una risposta sotto forma di pagina HTML.
- Un tipico caso e' l'accesso da parte dell'utente ad un database che risiede sul server. In questo caso la pagina web funziona come una interfaccia per accedere ai dati che risiedono sul server.
- Ci sono varie tecnologie server side utilizzate tra cui Common Gateway Interface (CGI), PHP, ASP.NET, Servlet, JSP.

# Tecnologia Server Side

- ◆ La computazione avviene lato server e può avvenire tramite **programmi compilati** o **script interpretati**
- ◆ Nel caso di programmi compilati il web server si limita ad invocare, su richiesta del client, un eseguibile. L'eseguibile può essere scritto in un qualsiasi linguaggio che supporti l'interazione con il web server – tipicamente Java, Visual Basic, C++.
- ◆ Storicamente il primo approccio è stato il **Common Gateway Interface (CGI)**

## Tecnologia Server Side – programmi compilati

- ◆ La computazione server side con programmi compilati ha il vantaggio di un tempo di esecuzione molto veloce rispetto agli script, ma lo svantaggio di mandare in esecuzione una istanza del processo per ogni chiamata e quindi caricare molto il server in caso di molti accessi simultanei.
- ◆ Alcuni approcci tentano di rimediare a questo svantaggio – Servlet e ASP.NET – mandando un'esecuzione una sola istanza di processo che gestisce chiamate multiple

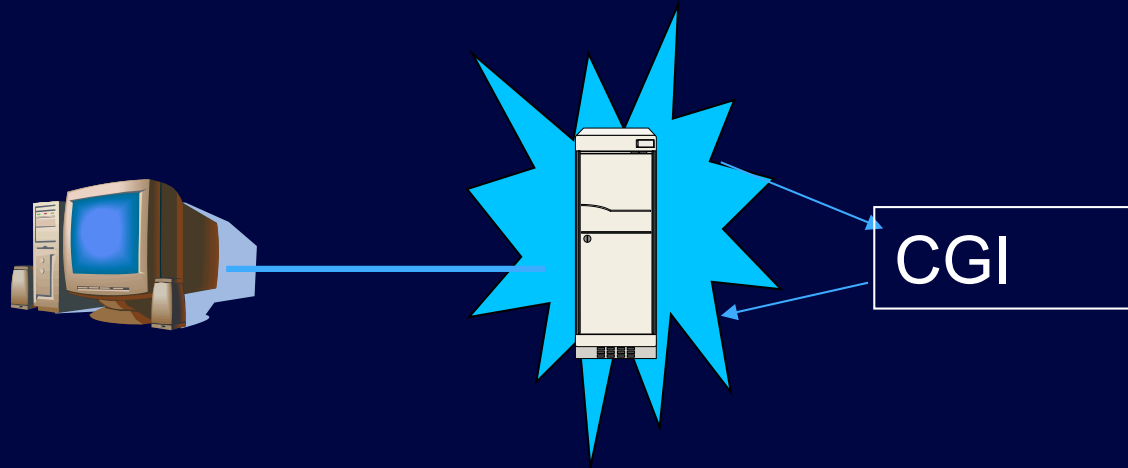


# Tecnologia Server Side - script

- ◆ Nel caso di esecuzione di script, il web server ha al suo interno **un motore** in grado di interpretare il linguaggio di script usato.
- ◆ Si perde in velocità di esecuzione, ma si guadagna in facilità d'uso
- ◆ In alcuni approcci rimane il problema dell'esecuzione simultanea di più processi (Perl), in altri viene eseguito un solo processo che gestisce tutte le chiamate (Cold Fusion, ASP)
- ◆ E' possibile gestire più tecnologie serverside da uno stesso server web

# Common Gateway Interface

I programmi CGI sono scritti in un linguaggio di programmazione e generalmente compilati in codice eseguibile sulla piattaforma del server (o interpretati da un interprete che risiede sul server)

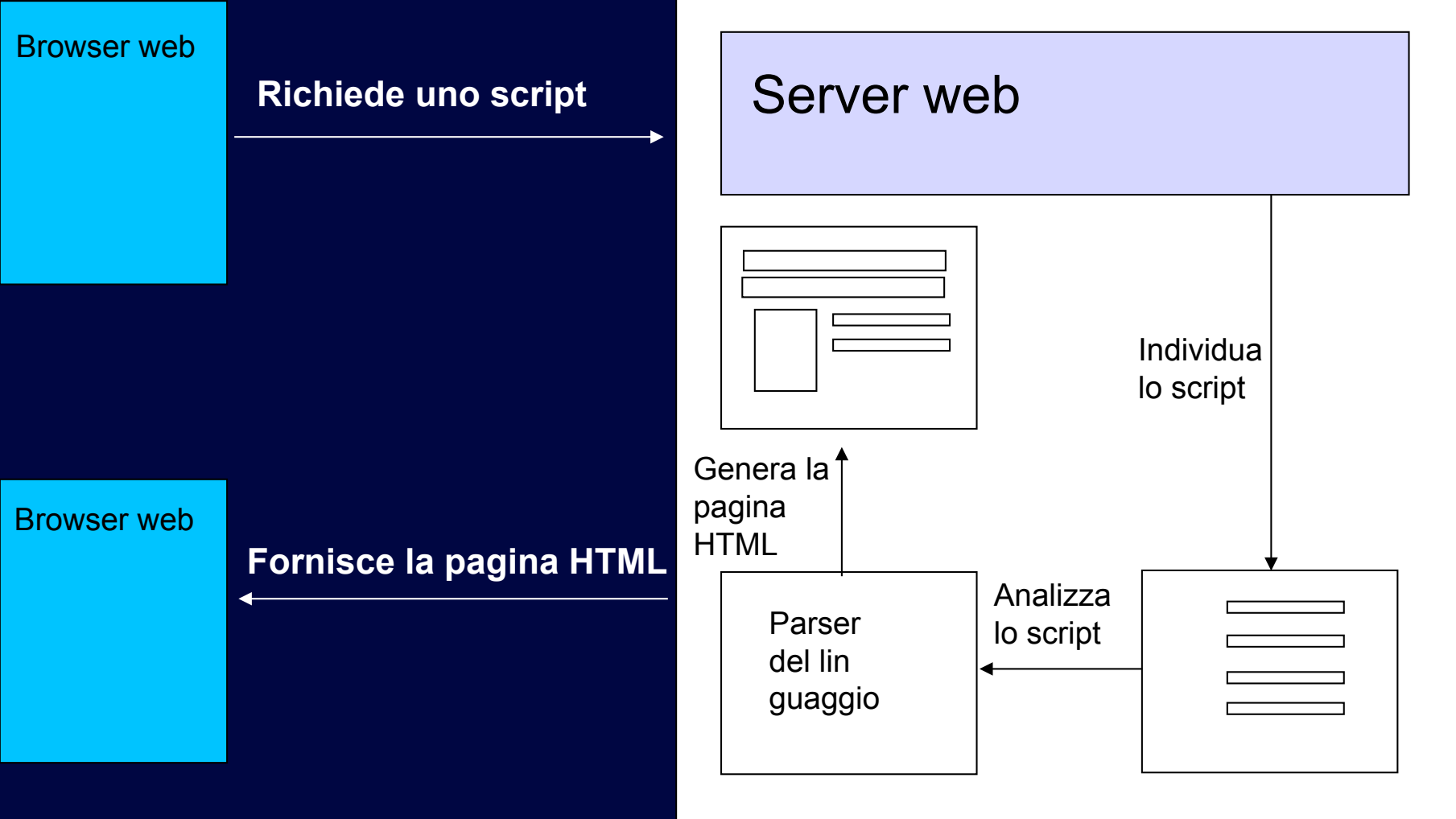


Il programma CGI genera come output una pagina HTML (dinamica)

# Linguaggi di Scripting

- ◆ Per superare i limiti dei CGI (inefficienza, complessità di programmazione) è nata la tecnologia dei **linguaggi di scripting** integrati con il server web.
- ◆ Tra questi approcci ricordiamo Active Server Pages (ASP), Java Server Pages (JSP), PHP, Cold Fusion
- ◆ **In questo corso ci concentreremo su PHP.** Da notare che i principi di funzionamento di PHP sono gli stessi che ritroviamo in altri linguaggi di scripting come ASP o JSP.

# Linguaggi di scripting



# Esempi di approcci Server Side

- ◆ Tipico esempio di applicazione web con interazione con l'utente sono le FORM HTML. Vediamo un esempio di form e come viene implementato usando varie tecnologie.
- ◆ Si realizza una comunicazione tra la pagina contenente la form e componente lato server che gestisce il contenuto della form spedito dall'utente. Questa componente e' diversa a seconda della tecnologia usata

## Parametri

Esempio di Form HTML

Nome:

Rosso  Verde

```
<FORM name=esempio action="programma"  
METHOD=GET|POST>  
<INPUT type=text name=username>  
<INPUT type=radio name=color  
value="Rosso"> .....  
<INPUT type=submit>
```

## Esempio FORM - Codice Perl

In questo caso il “programma” indicato dalla action è un programma perl (ed es. mioperl.pl)

```
Use CGI;
$q= new CGI;
$Color = $q -> param('color');
$Username = $q-> param('username');
Print $!$q ->(header);
`<html><head><title>CGI</title></head>',
`<body>',
`Il nome è: `, $Username, `<br>'
`il radio button selezionato è: `, $color, `<br>,
`</body></html>';
```

# Esempio FORM – Cold Fusion

In questo caso il “programma” indicato dalla action è una pagina Cold Fusion (ed es. miapagina.cfm)

```
<HTML><HEAD><TITLE>Cold Fusion </TITLE></HEAD>
```

```
<BODY>
```

Il nome è: `#form.username#`

Il radio button selezionato è: `#form.color#`

```
</BODY></HTML>
```

# Esempio FORM - ASP

In questo caso il “programma” indicato dalla action è una pagina ASP (ed es. miapagina.asp)

```
<HTML><HEAD><TITLE>ASP </TITLE></HEAD>
```

```
<BODY>
```

```
Il nome è: <% request.querystring('username') %>
```

```
Il radio button selezionato è:<% request.querystring('color') %>
```

```
</BODY></HTML>
```



# Esempio FORM - PHP

In questo caso il “programma” indicato dalla action è una pagina PHP (ed es. miapagina.php)

```
<HTML><HEAD><TITLE>PHP</TITLE></HEAD>
```

```
<BODY>
```

```
Il nome è: <?php $username; ?>
```

```
Il radio button selezionato è:<?php $color; ?>
```

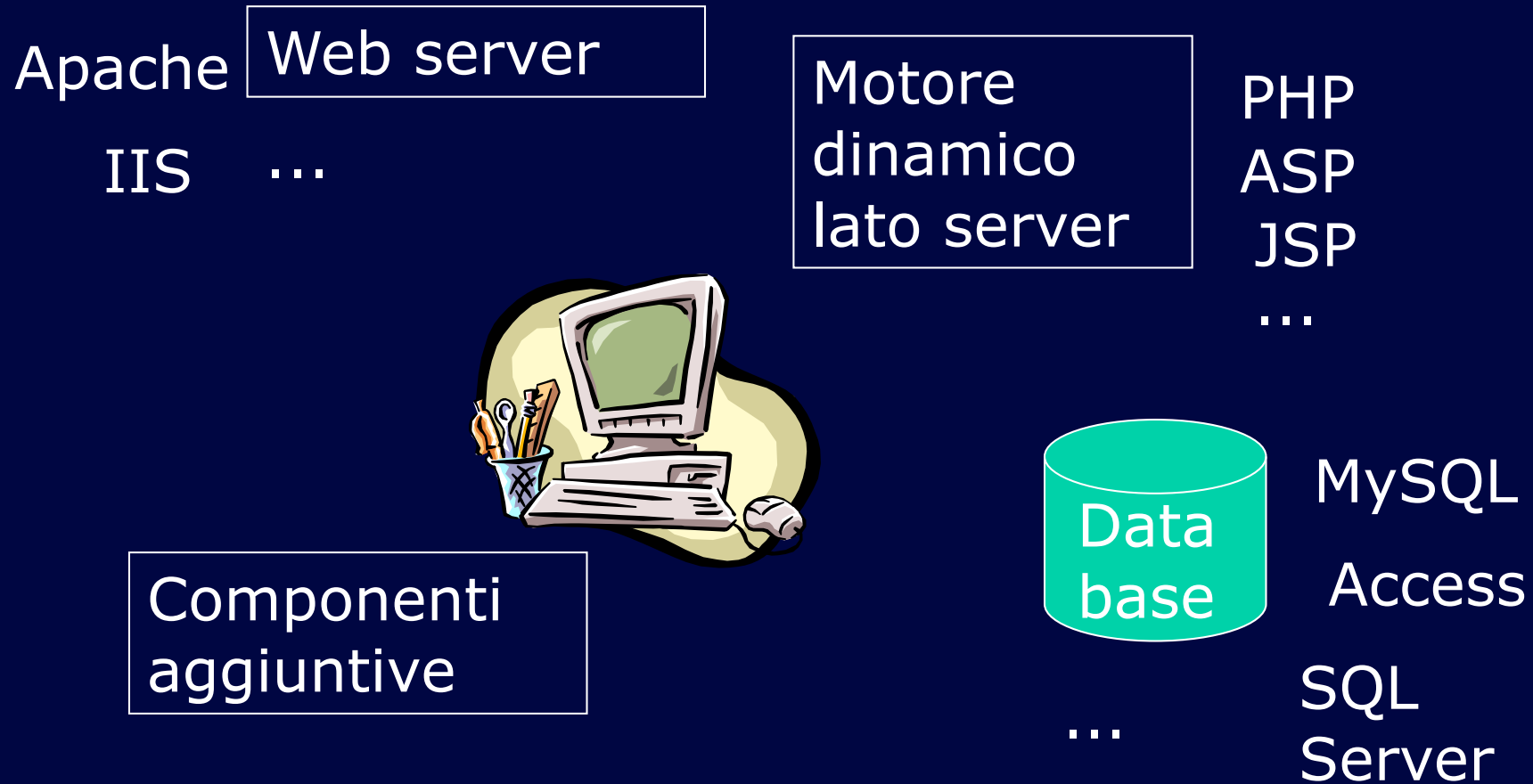
```
</BODY></HTML>
```

# Tecnologie Client Side

- ◆ **La computazione lato client può avvenire con programmi scaricati sul browser (applet, plug-in) oppure tramite script (Javascript, VBscript)**
- ◆ **Gli Applet sono programmi scritti in Java che vengono scaricati dal server al browser ed eseguiti sulla macchina client. Hanno delle limitazioni di funzionamento dovute a problemi di sicurezza. E' possibile comunque accedere a dati che risiedono sul server (database)**
- ◆ **Gli script lato client sono generalmente usati per validare dati di input dell'utente e sono supportati dal browser (Javascript/VBScript)**
- ◆ **Recentemente si sta diffondendo AJAX, un mix di tecnologie clientside per implementare una connessione asincrona tra client e server (google maps, gmail)**

# Application server

Ambiente per costruire applicazioni web



# Application Server

## *AS commerciali (proprietary)*

- ◆ Cold Fusion di Adobe.
- ◆ ASP. NET di Microsoft.

## *AS open source*

- ◆ PHP (P Hypertext Preprocessor), ideato da Rasmus Lerdorf.
- ◆ ZOPE (Z Object Publishing Environment), ideato da Jim Fulton.
- ◆ TOMCAT, è un servlet container per applicazioni JSP.
- ◆ (Esistono anche server J2EE proprietari come IBM WebSphere)

# Conclusioni

- ◆ La maggior parte dei siti web di oggi sono **Applicazioni web**, ovvero abbinano alla parte statica del protocollo HTTP una componente di programmazione e, tipicamente, una base di dati.
- ◆ La componente di programmazione può essere delegata al server e/o al client (tecnologia server-side vs client-side). Server side permette applicazioni più complesse e la connessione ad un database.
- ◆ In questo corso tratteremo il lato server e in particolare vedremo un'introduzione al linguaggio di script PHP e la connessione al database MySQL

# Da fare Adesso !

## Installare un Web Server e testare una pagine di prova

### ◆ Mac OS X:

- MAMP: <http://www.mamp.info/en/index.html>
- XAMPP: <http://www.apachefriends.org/en/xampp.html>

### ◆ Windows:

- XAMPP: <http://www.apachefriends.org/en/xampp.html>
- EasyPhp: <http://www.easyphp.org/>
- Lighty2Go: <http://www.lighty2go.com/get-lighty2go/>
- Wamp Server: <http://www.wampserver.com/en/>

### ◆ Linux

- XAMPP: <http://www.apachefriends.org/en/xampp.html>
- Bitnamp: <http://bitnami.org/stack/lampstack>

### ◆ Otherwise:

- Google: “LAMP” oppure “MAMP” oppure “LAMP”