

Basi di Dati

Introduzione HTML e PHP

Prof. Mauro Conti

Dipartimento di Matematica - Università degli studi di Padova

conti@math.unipd.it - <http://www.math.unipd.it/~conti>



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

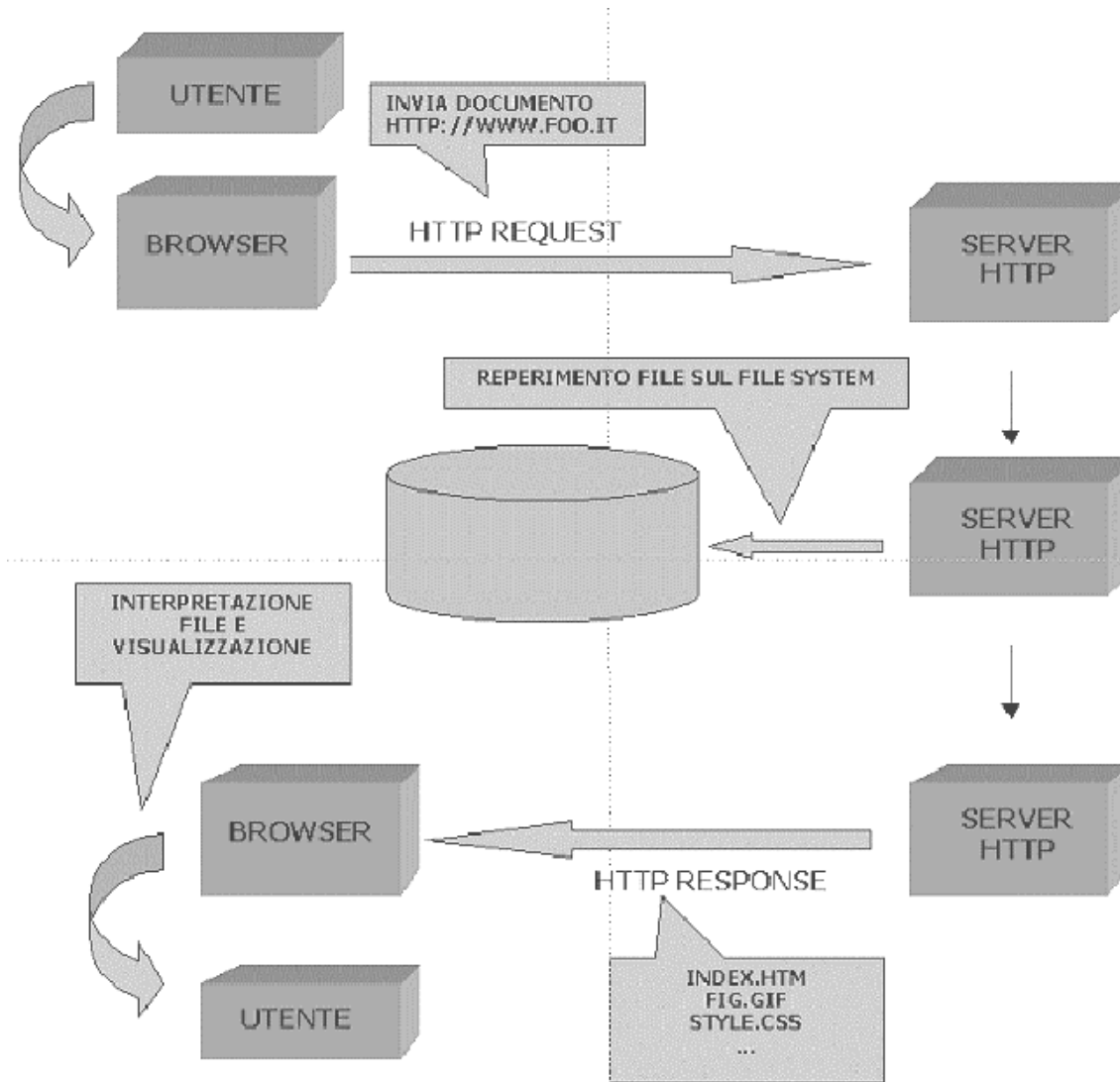
Slides credits to:
Paolo Baldan

- Basi di dati e web: nozioni generali
- HTML
- Basi del linguaggio PHP
- PHP e API MySQL
- Interazione con l'utente: forms e gestori
- Mantenimento dello stato: sessioni e cookies
- Autenticazione

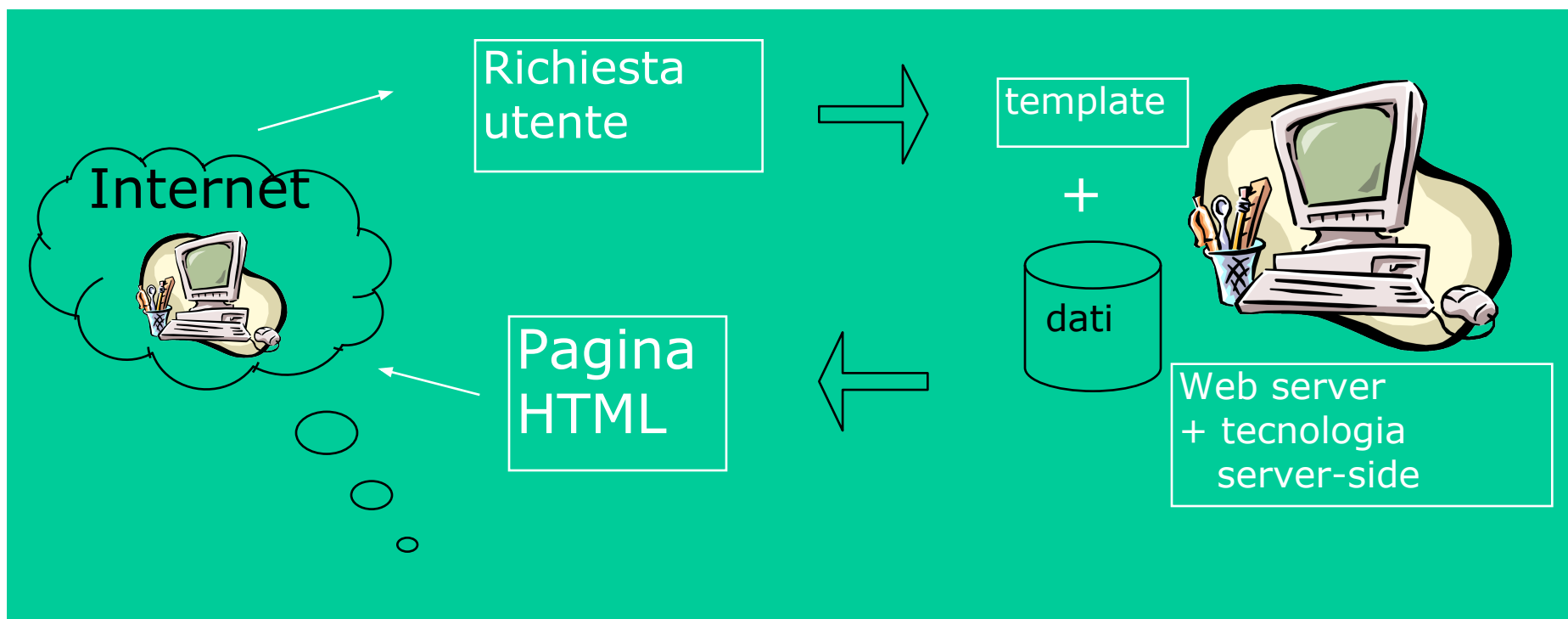
- **Applicazione web**
 - fornisce un servizio all'utente per mezzo dell'**infrastruttura WEB**
 - l'utente interagisce con un **sito web**
 - le pagine sono costruite dinamicamente, sulla base dell'interazione con l'utente (**siti dinamici**)
 - le informazioni rilevanti per l'applicazione memorizzate in un **database**
- Es.: prenotazione on-line, commercio elettronico, home-banking,



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



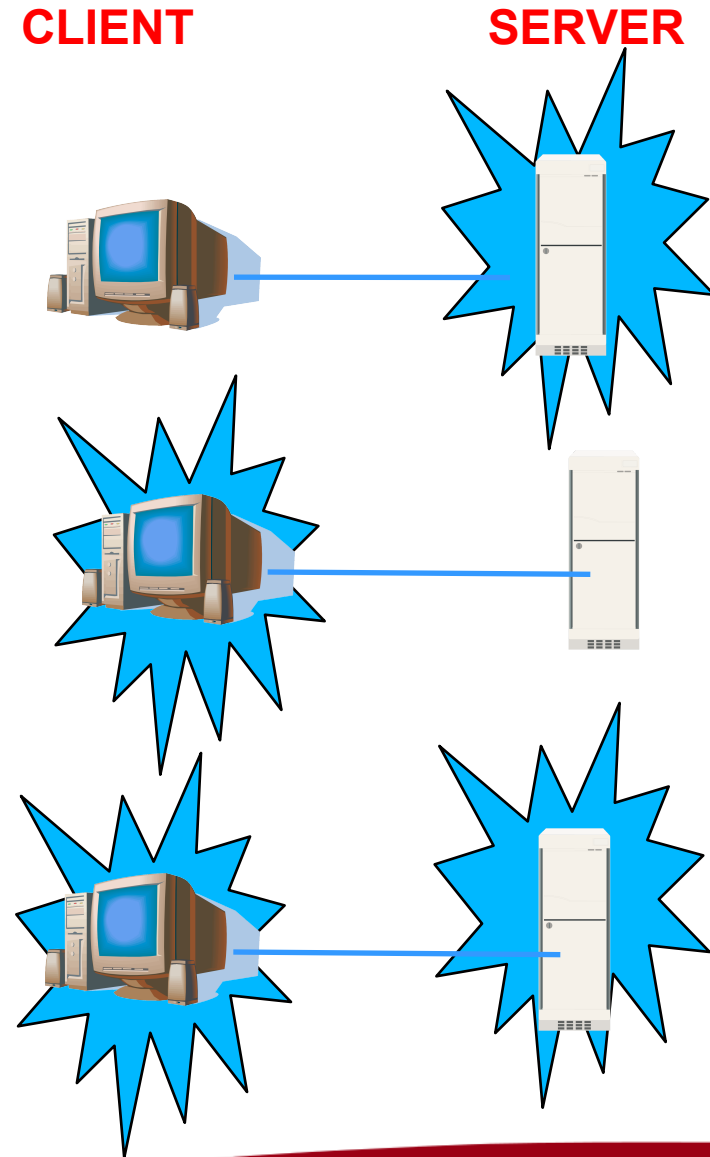
- Protocollo HTTP per il trasferimento dei dati
- Alla struttura della pagina fissa (template) si “uniscono” i dati provenienti da un database
- Necessaria una fase di elaborazione



tecnologia server side: peso
della computazione sul server

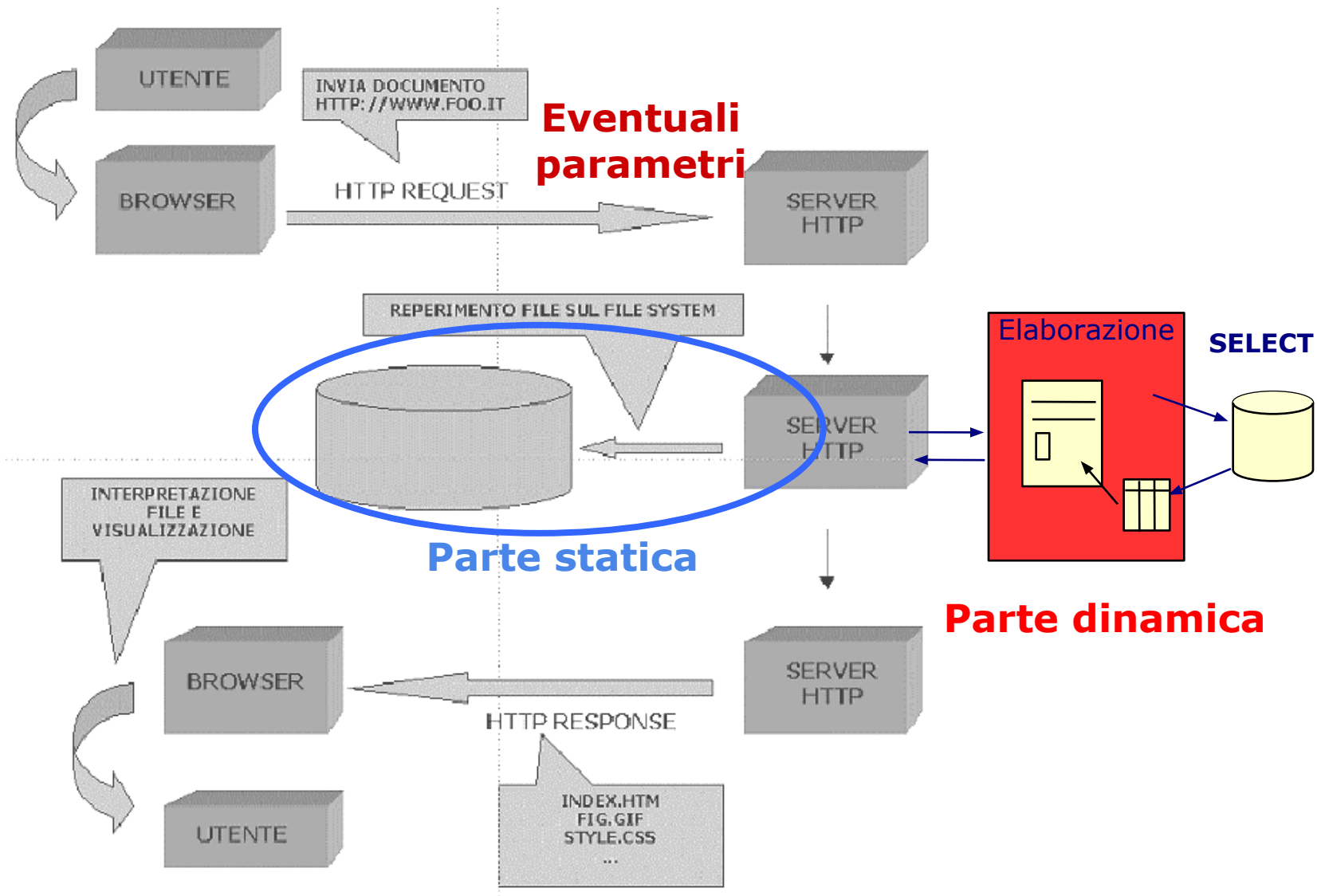
tecnologia client side: peso
della computazione sul client

tecnologia ibrida
(soluzione tipica)

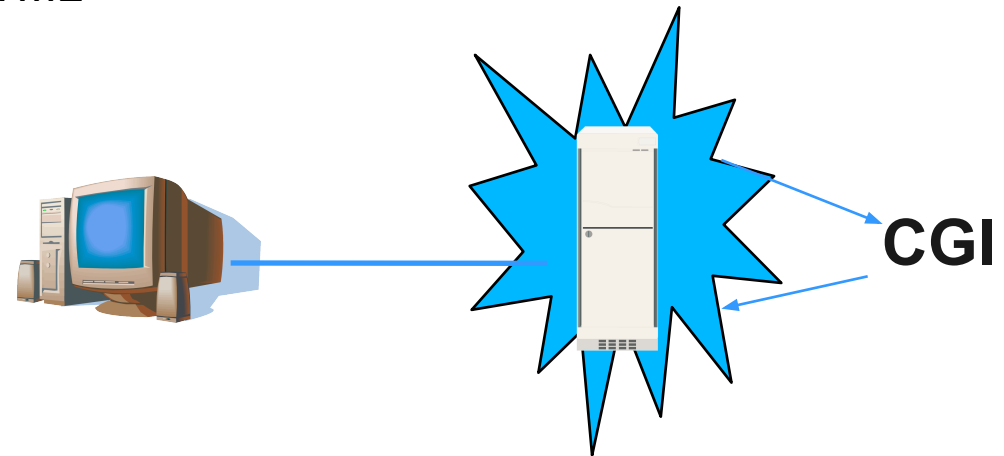


- **Computazione sul client:**
 - **Script interpretati** dal browser, immersi nel codice HTML (es. Javascript)
 - Tipicamente operazioni semplici e in stretta interazione con l'utente
 - controllo delle form
 - gestione eventi (es. rollover)
 - **Eseguibili** scaricati (es. Applet)
 - limitazioni dovute a problemi di sicurezza
 - capacità di accedere a dati sul server (es. DB)
- Criticità:
 - dipendenza dal browser

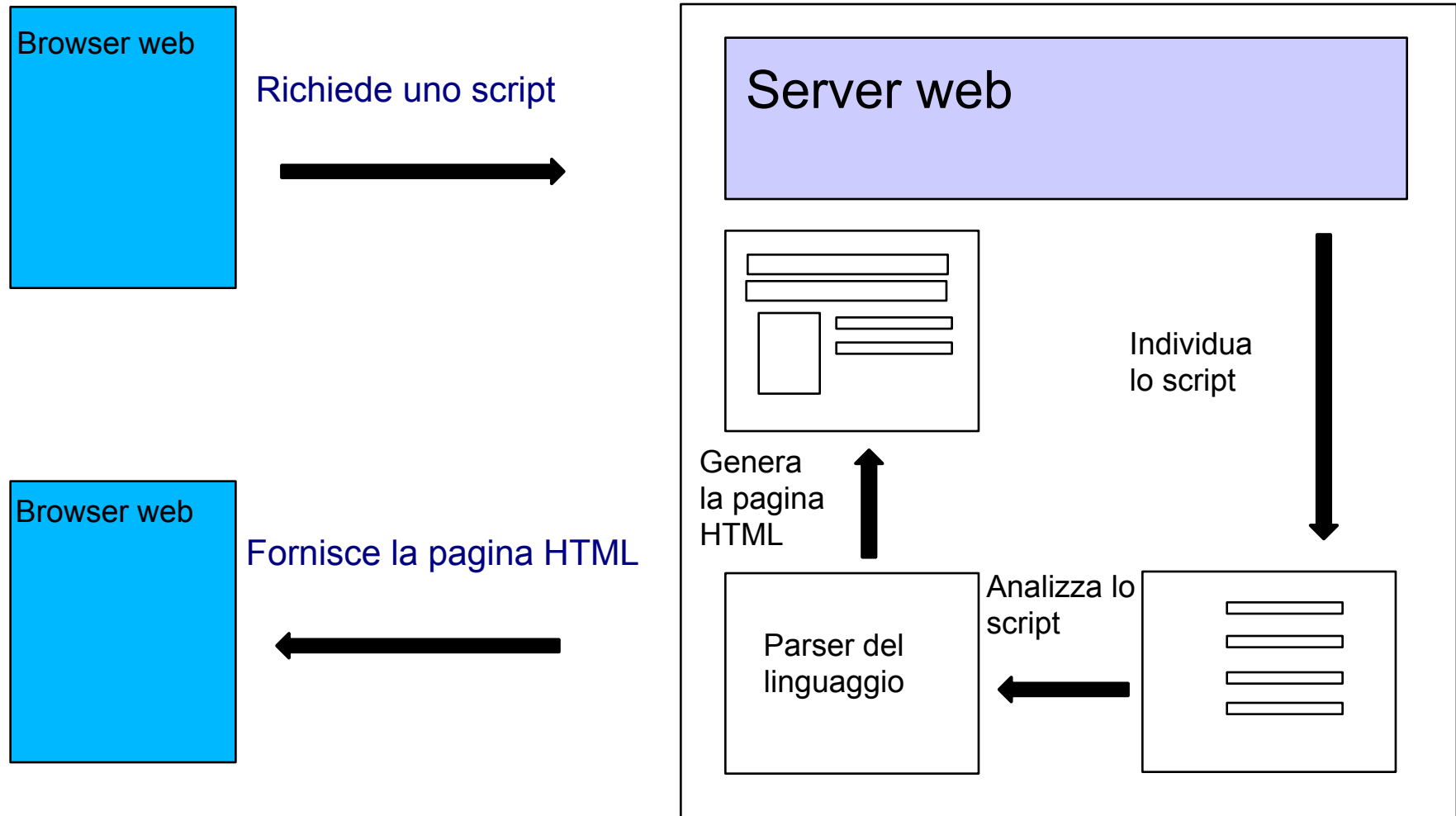
- **Computazione sul server**
 - maggior flessibilità e capacità computazionale
 - accesso e memorizzazione dei dati
 - interattività non immediata (occorre che l'informazione arrivi al server !)
- Il server fa ben più che restituire pagine HTML/script già esistenti!
 - **pagine dinamiche**: create e composte dinamicamente dal server web
 - utente invia dati (es., tramite una form) al server
 - il server li elabora
 - restituisce il risultato come documento HTML:
 - necessità di componenti aggiuntive per l'elaborazione, capaci di costruire pagine su richiesta, sulla base delle richieste e dei dati



- **CGI (common gateway interface)**
 - protocollo per interfacciare un web server con un'applicazione esterna
 - il WS invoca, su richiesta del client, un programma (che risiede in opportune cartelle)
 - compilato
 - interpretato da un interprete che risiede sul server
 - il programma ritorna una pagina HTML
- Perl, PHP, ma anche Java, C++, ...



- **Script (embedded)**
 - il web server include come modulo un motore capace di interpretare il linguaggio di scripting
 - es. PHP (PHP Hypertext Preprocessor), ASP (Active server pages), Java Server Pages (JSP), ...



- 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">
```


- In questo caso il “programma” indicato dalla action è un programma PERL (ad es. mioperl.pl)

```
#!/usr/bin/perl
use CGI;$q = new CGI;$color = $q->param
('color');$username = $q->param('username');
print $q->header();print $q->start_html(-title
=> 'CGI');print "Il nome è: $username<br>\n";
print "Il radio button selezionato è:
$color<br>";print $q->end_html;
```

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

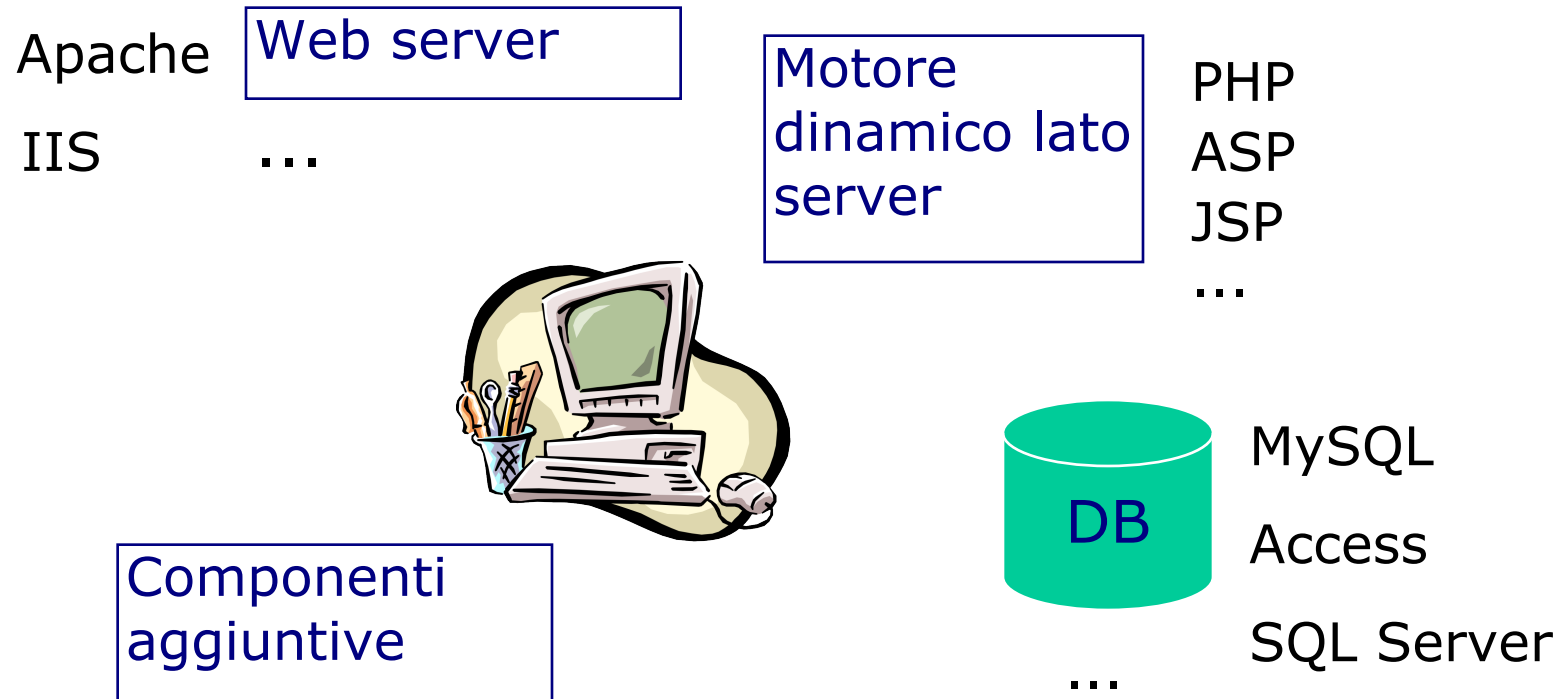
```
<html>
<head><title>PHP</title></head>
<body>
Il nome è: <?php $_GET["username"]; ?>Il radio
button selez. è: <?php $_GET["color"]; ?></body>

</HTML>
```

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

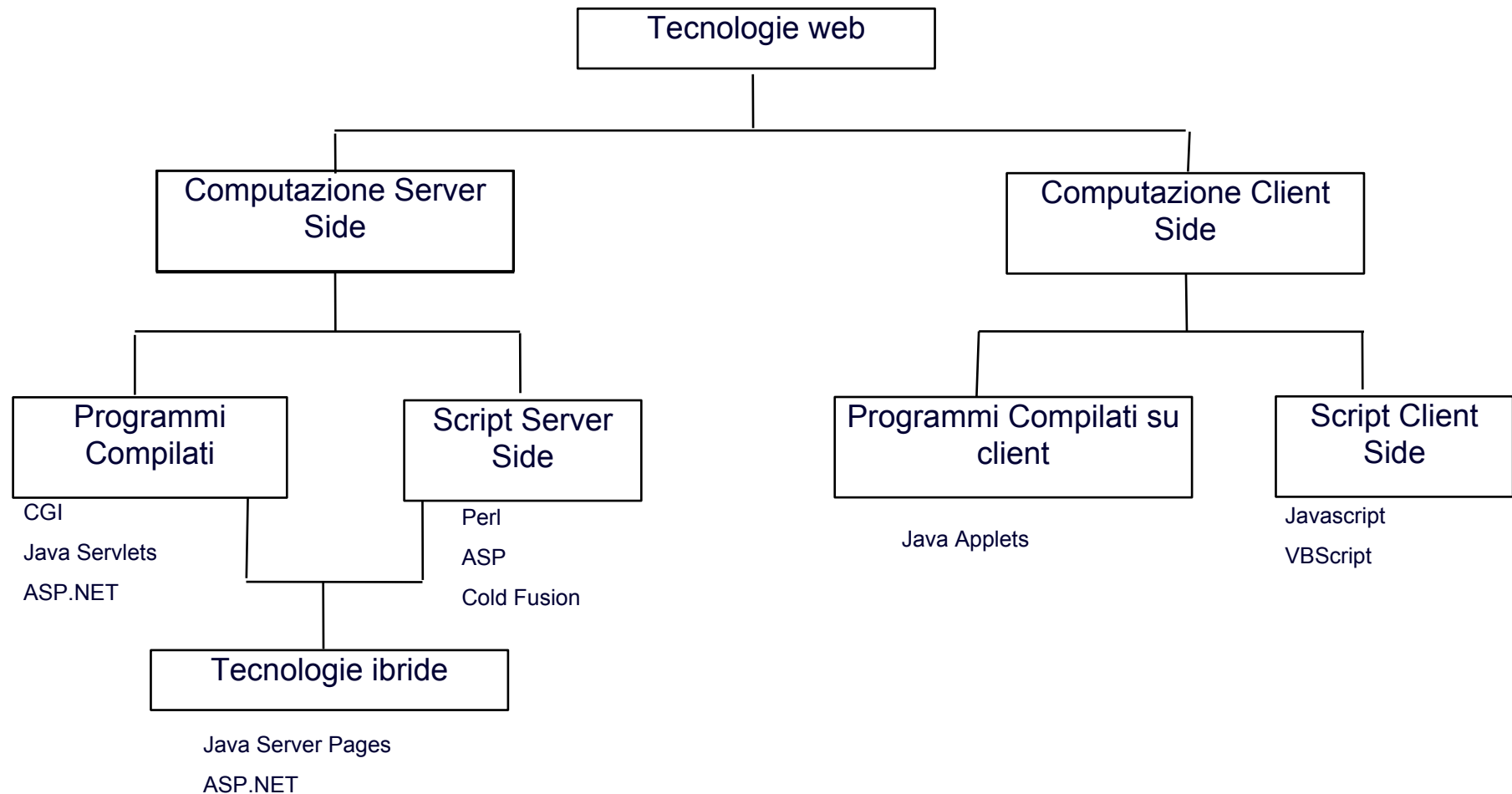
```
<html>
  <head><title>ASP</title></head><body>Il nome
  è: <% request.querystring('username') %>Il radio
  button selezionato è:<% request.querystring
  ('color') %></body>
</html>
```

- Non solo un HTTP server, integra moduli relativi a componenti server-side
- Ambiente per costruire applicazioni web



- 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)

- Noi considereremo
 - Apache
 - MySQL
 - PHP (modulo di Apache)
- L'installazione integrata delle tre componenti non è sempre agevole
- Così comune che in rete si trovano vari kit di installazione integrati (consigliati)
 - LAMP (Linux, Apache, MySQL, PHP)
 - WAMP, MAMP, XAMPP, ...



HTTP in a nut-shell

- Protocollo di livello applicativo, relativo alla suite di protocolli TCP/IP
- Base per lo scambio di informazioni su internet
- Concetto essenziale: i file possono contenere **riferimenti** ad altri file, selezionando i quali si avviano ulteriori richieste di trasferimento
- Un documento è univocamente identificato da una URL
- <http://www.math.unipd.it/~conti/teaching/DB1415/index.html>
 - protocollo
 - indirizzo del web server
 - documento richiesto al server

- URL può essere dinamico (denota un programma che genera la pagina e dei parametri per questo)
 - <http://orario.trenitalia.com/TravelSolutions.do?&stazin=VE&stazout=PD>
- Ogni web server esegue un HTTP server
 - ascolta su di una porta (80)
- Il browser è un client HTTP che manda richieste al server
 - il documento richiesto è indicato con la sua URL
 - il browser traduce le URL in richieste HTTP
 - effettua il parsing e visualizza il documento HTML

- **protocollo state-less**
 - connessione aperta solo per il tempo del trasferimento del file
 - non si mantengono informazioni relative alla connessione
- I passi che vengono effettuati ad ogni richiesta ad un server HTTP sono:
 - Il client (tipicam. un browser) inizia una conness.ione HTTP verso un host
 - Il server accetta la connessione
 - Il client manda una richiesta per un docutmento: GET path from URL
 - Il server reperisce il documento richiesto
 - Il server spedisce il documento trovato
 - Il client accetta il documento

- Quando il trasferimento e' completo, il server chiude la connessione
- Il client termina la connessione HTTP
- Le richieste HTTP possono essere
 - GET richiesta di un documento
 - POST spedizione parametri di una form
 - PUT spedizione dati
 - HEAD informazioni di header (last modified...)
- Il server web restituisce al browser oltre all'eventuale documento, un codice numerico che indica l'esito dell'operazione. Ad esempio il codice 200 indica documento trovato e restituito, mentre 404 indica document not found, oppure 401 unauthorized.