

Informatica **Umanistica**

# Il Server web

Laboratorio di Progettazione Web

*AA 2009/2010*

*Chiara Renso*

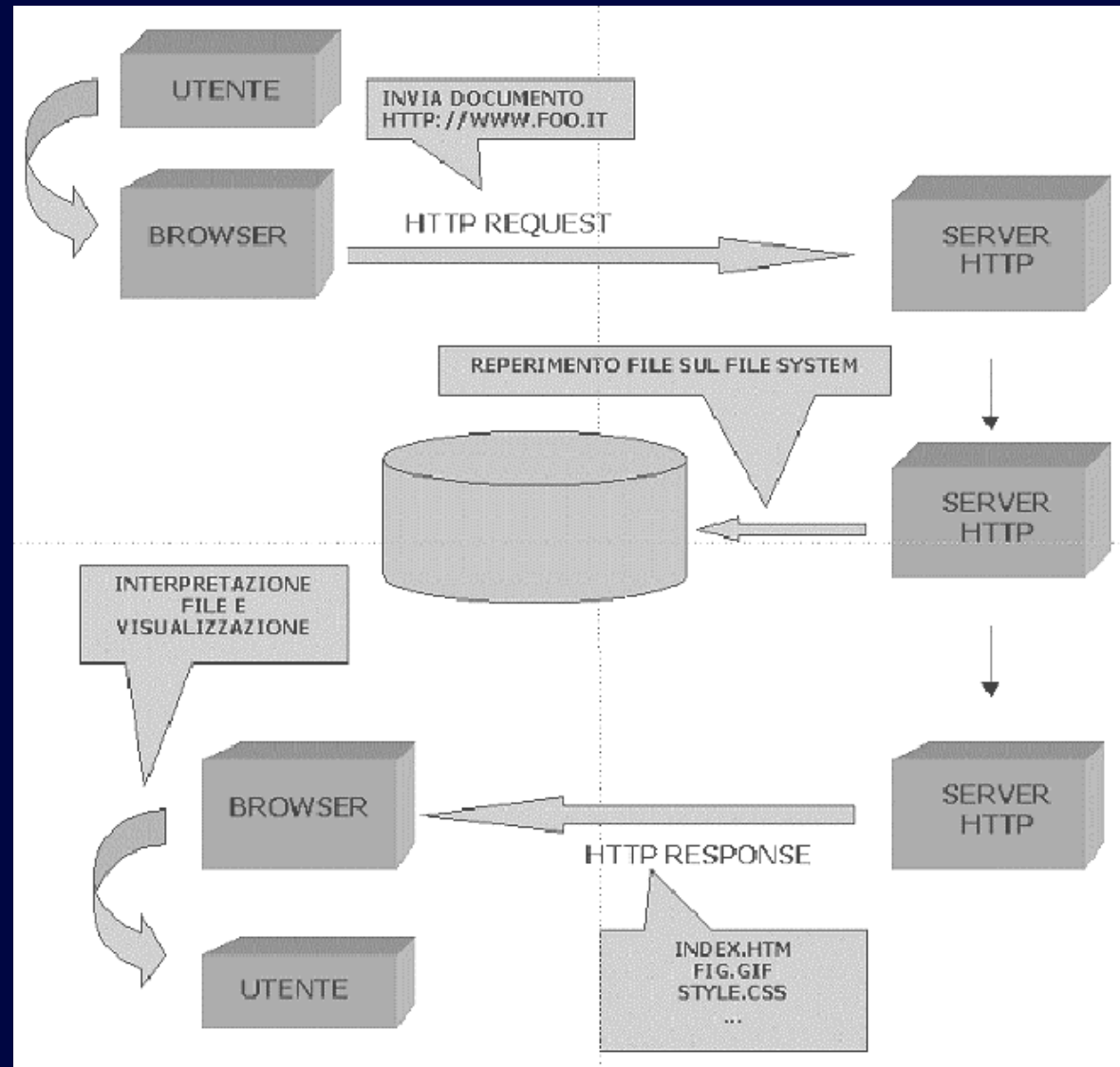
*ISTI- CNR - c.renso@isti.cnr.it*



UNIVERSITÀ DI PISA

# Il Web Server e il protocollo HTTP

E' un programma sempre attivo che ascolta su una porta le richieste HTTP. All'arrivo di una richiesta la esegue e restituisce il risultato al browser, rimettendosi in ascolto



# HTTP

- ◆ Nel protocollo HTTP il browser invia al server una **richiesta HTTP**, e riceve dal server una **risposta HTTP** contenente l'esito della richiesta.
- ◆ Nella richiesta abbiamo: un **metodo**, la URI e la versione del protocollo

Esempio: dato un link

<http://www.miodominio.it/dir/file.html>

La corrispondente richiesta HTTP sarà

```
GET /dir/file.html HTTP/1.1
```

```
Host: www.miodominio.it
```

# Metodi nella richiesta HTTP

- ◆ **GET *risorsa***: richiede al server l'invio di un file, indicato dal percorso della risorsa
- ◆ **POST *risorsa***: invia al server (in particolare alla risorsa, che tipicamente e' un programma lato server) delle informazioni del tipo *attributo=valore*
- ◆ **HEAD**: simile al GET, ma si usa per la diagnostica

# Risposta HTTP

- ◆ Nel messaggio di risposta troviamo un *codice di stato* e il file richiesto con il metodo GET

## Esempio:

```
HTTP/1.1 200 OK
```

```
Content-type: text/html
```

```
Content-length: 3895
```

```
<HTML>
```

```
<HEAD>
```

# Codici di stato

- ◆ **1XX – informazioni**
- ◆ **2XX – Codici di successo (200 tutto bene)**
- ◆ **3XX – richiesta ricevuta ma e' necessaria una ulteriore azione affinche' la richiesta venga soddisfatta**
- ◆ **4XX – errore, la richiesta non puo' essere soddisfatta**
- ◆ **5XX – errori interni del server, ad esempio errori negli script lato server**

# Quali web server?

- ◆ Essendo il protocollo HTTP uno standard libero, in linea di principio chiunque è in grado di costruirsi un proprio server web, seguendo il protocollo HTTP.
- ◆ I web server sono ormai diventati strumenti estremamente complessi proprio per il supporto alla tecnologia server side, oltre che per ragioni di efficienza nel soddisfare un numero molto alto di richieste contemporanee
- ◆ I web server più conosciuti e usati nei siti di tutto il mondo sono:
  - ◆ Apache (<http://www.apache.org>)
  - ◆ Microsoft Internet Information Server (IIS)

# Gestire un web server

- ◆ **Scelta del server HTTP**
- ◆ **Principali parametri di configurazione:**
  - Avvio, spegnimento
  - Porta di funzionamento
  - Directory di default
  - Documento di default
  - Log degli accessi/errori
  - Alias o cartelle virtuali



# Avvio e spegnimento del server

- ◆ Tipicamente uno server web per sua stessa definizione è un programma *sempre attivo*.
- ◆ Ci sono però dei casi in cui vogliamo disattivare il server (mettere fuori linea il sito per manutenzione, ad esempio). Per questa ragione ogni server web ha comandi per fermare il server e riavviarlo

# Web Server e porte

- ◆ Il web server (o server HTTP) come tutti i servizi basati su TCP/IP, si attiva su una **porta**, che è il numero a cui si fa corrispondere il servizio.
- ◆ In questo modo una macchina può far girare più servizi differenziando le porte
- ◆ Ci sono porte standard per i vari servizi (ftp, telnet, posta elettronica, web)
- ◆ La porta di default del web server è la **80**, ma può essere configurato per funzionare su un'altra porta libera. In questo caso la porta va specificata sia nella configurazione del webserver che nella URL

# Web server e porte

<http://localhost:8080/>

<http://www.miodominio.it:9736/esempi/esercizio.php>

- ◆ Negli esempi di queste URL il web server è stato configurato per funzionare sulla porta 8080 (nel primo caso) e sulla 9736 (nel secondo caso).
- ◆ Nella prima URL la parola chiave **localhost** sta a indicare il server web attivo sulla macchina locale (può ovviamente funzionare senza una connessione internet), la macchina locale fa da server e da client

# Web server e documenti

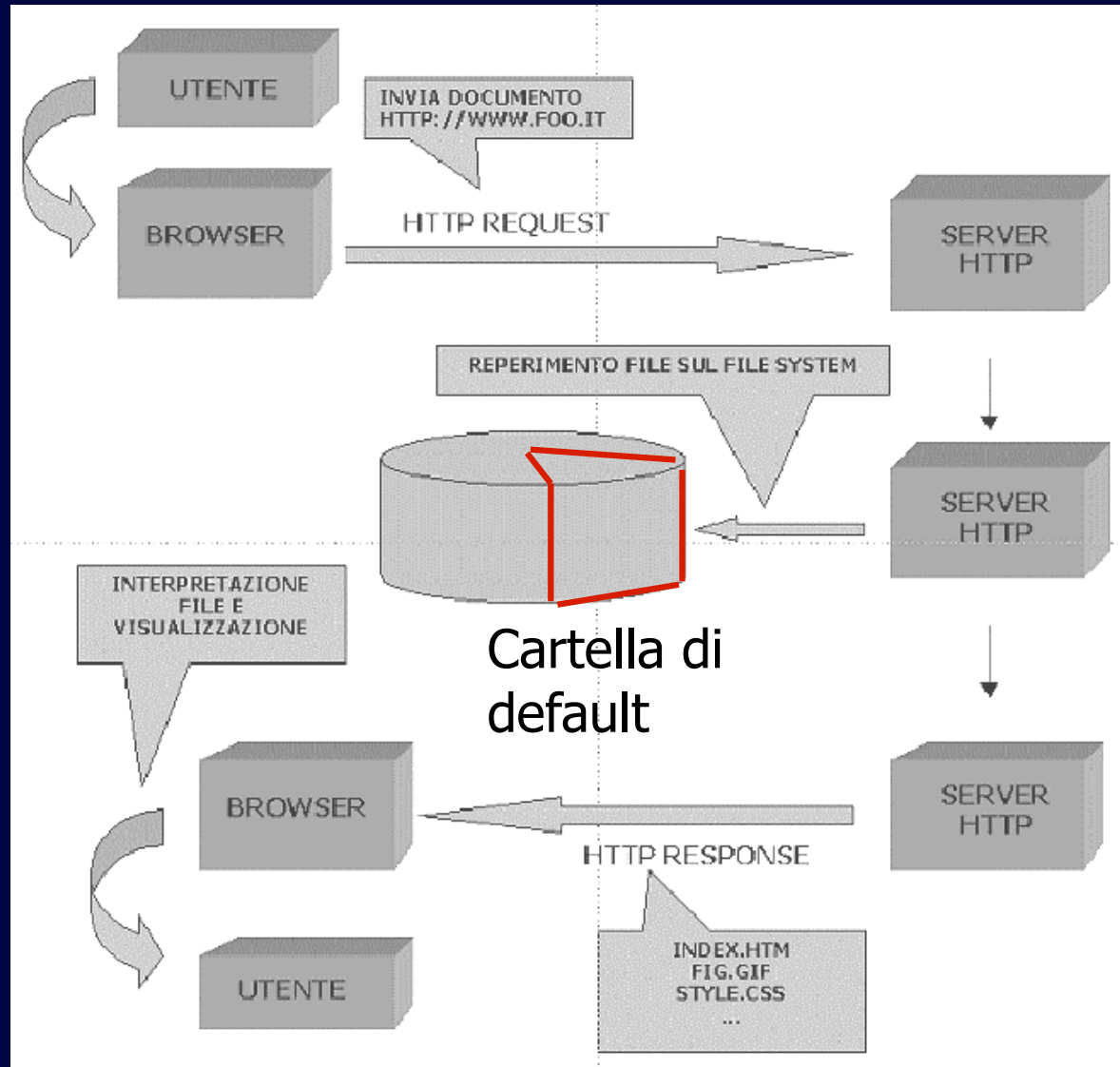
<http://localhost:8080/>

<http://www.miodominio.it:9736/esempi/esercizio.php>

- ◆ Nella prima URL attiviamo una richiesta al server web locale attivo sulla porta 8080. Non specifichiamo quale documento restituirci. Il web server in questo caso decide di ritornarci il **documento di default**
- ◆ Nella seconda URL invece specifichiamo di restituirci la pagina **esercizio.php** nella cartella **esempi**

# Web Server e documenti

- ◆ Il server web ha visibilità di una sola parte del file system della macchina. Per ragioni di sicurezza e di pulizia del sistema
- ◆ La porzione di file system visibile dal server web si chiama **directory** (o **cartella**) di **default** del server web



# Web server e documenti

- ◆ La cartella di default del server web varia a seconda del server considerato e del sistema operativo su cui è installato.
- ◆ Ad esempio:
  - Per Apache su windows tipicamente è la cartella **htdocs** che si trova nella directory di installazione di Apache (ad es. *C:\Programmi\Apache Group\Apache\htdocs*)
  - Per IIS è la cartella **wwwroot** che generalmente si trova in *C:\inetpub\wwwroot*

# Web server e documenti

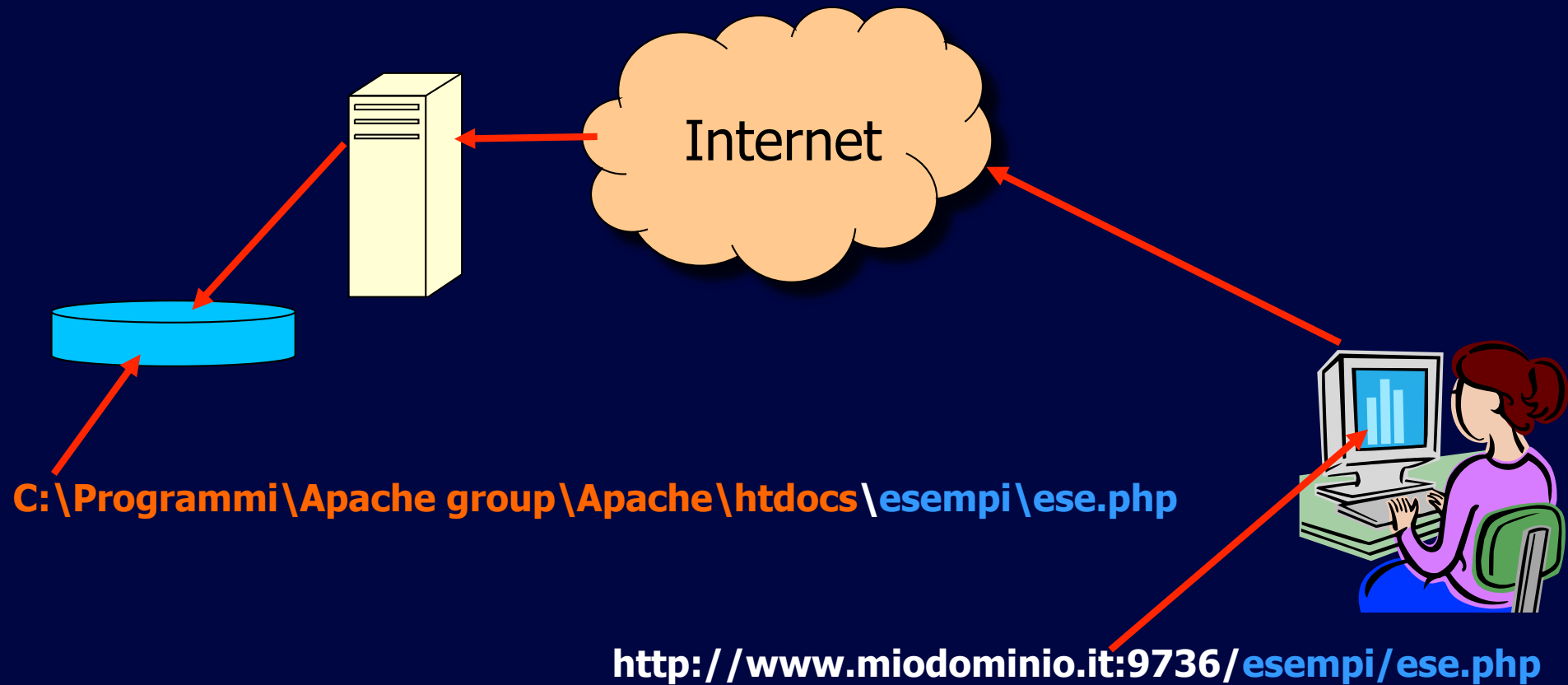
<http://www.miodominio.it:9736/esempi/esercizio.php>

- ◆ In questo caso la cartella **esempi** è una sotto cartella della directory di default quindi su apache sarà in:

`C:\Programmi\Apache group\Apache\htdocs\esempi`

Affinchè il file `esercizio.php` sia visibile alla URL di esempio sopra dovrà quindi essere salvato in questa cartella

# URL e directory





# Documenti di default

- ◆ E' uso comune configurare il server web affinché restituisca un documento predefinito qualora non ne venga specificato uno nella URL
- ◆ Questo permette ai siti web di rendere visibile una "home page" digitando il solo indirizzo del web server
- ◆ Ad es: [www.facebook.com](http://www.facebook.com) o [www.google.com](http://www.google.com)
- ◆ La pagina visualizzata prende il nome di homepage e tipicamente ha uno dei nomi: **index.html, index.htm, index.php, default.html, default.htm, default.asp** etc...
- ◆ Queste pagine di default e l'ordine in cui vengono cercate **possono essere configurate sul server web**

## Logs di accesso e errori

- ◆ **Ogni richiesta arrivata al server web viene registrata in appositi file chiamati Logs di accesso. Possono avere formati diversi, ma essenzialmente contengono informazioni su:**
  - Indirizzo IP del client
  - Data e ora
  - Tipo di richiesta (GET o POST)
  - Documento richiesto
  - Risposta del server (tutto bene – 200 - o codice di errore)

## Logs di accesso e errori

- ◆ Analogamente vengono registrati tutti gli errori che genera il server web ad una data richiesta
- ◆ Ad es. Pagina non trovata (codice 404) o **errori specifici dello script se la pagina è dinamica** (ad es PHP)
- ◆ Per i siti web è importante controllare periodicamente i logs per verificare che non ci siano troppi messaggi di errore (es esempio qualche link errato o errore in qualche script), ma anche per tenere sotto controllo eventuali tentativi di accesso non autorizzato (virus, hacker etc)

# Log Files - esempio

IP address client

Risposta del server

Comando HTTP

```
127.0.0.1 - - [21/Apr/2005:15:47:36 +0200] "GET /favicon.ico
HTTP/1.1" 404 293
127.0.0.1 - - [21/Apr/2005:15:47:47 +0200] "GET /my_public_html/
HTTP/1.1" 200 1653
127.0.0.1 - - [21/Apr/2005:15:47:47 +0200] "GET /icons/folder.gif
HTTP/1.1" 200 225
127.0.0.1 - - [21/Apr/2005:15:47:47 +0200] "GET /icons/blank.gif
HTTP/1.1" 200 148
127.0.0.1 - - [21/Apr/2005:15:47:47 +0200] "GET /icons/
unknown.gif HTTP/1.1" 200 245
127.0.0.1 - - [21/Apr/2005:15:47:47 +0200] "GET /favicon.ico
HTTP/1.1" 404 293
127.0.0.1 - - [21/Apr/2005:15:47:51 +0200] "GET /my_public_html/
orario_saluto.php HTTP/1.1" 200 174
```

Documento richiesto

# Log Files – strumenti di analisi

- ◆ I file di log sono praticamente impossibili da analizzare “a mano”, registrano ogni azione del server e quindi tendono ad essere enormi
- ◆ Esistono strumenti per facilitare l’analisi, forniscono funzionalità statistiche in forma grafica, alcuni sono free, altri a pagamento.
- ◆ Sono strumenti preziosissimi per il webmaster

# Alias e cartelle virtuali

- ◆ Quando un server web è attivo su una macchina la sua cartella di default è automaticamente resa pubblica.
- ◆ Se vogliamo **pubblicare** un documento dobbiamo quindi copiarlo in tale cartella
- ◆ In alcuni casi può essere utile rendere pubblica anche un'altra porzione del file system
- ◆ Si parla in questo caso di **alias** o **cartelle virtuali**
- ◆ Il meccanismo sta nel definire una nuova URL che punta ad una specifica cartella del disco, non necessariamente dentro la cartella di default

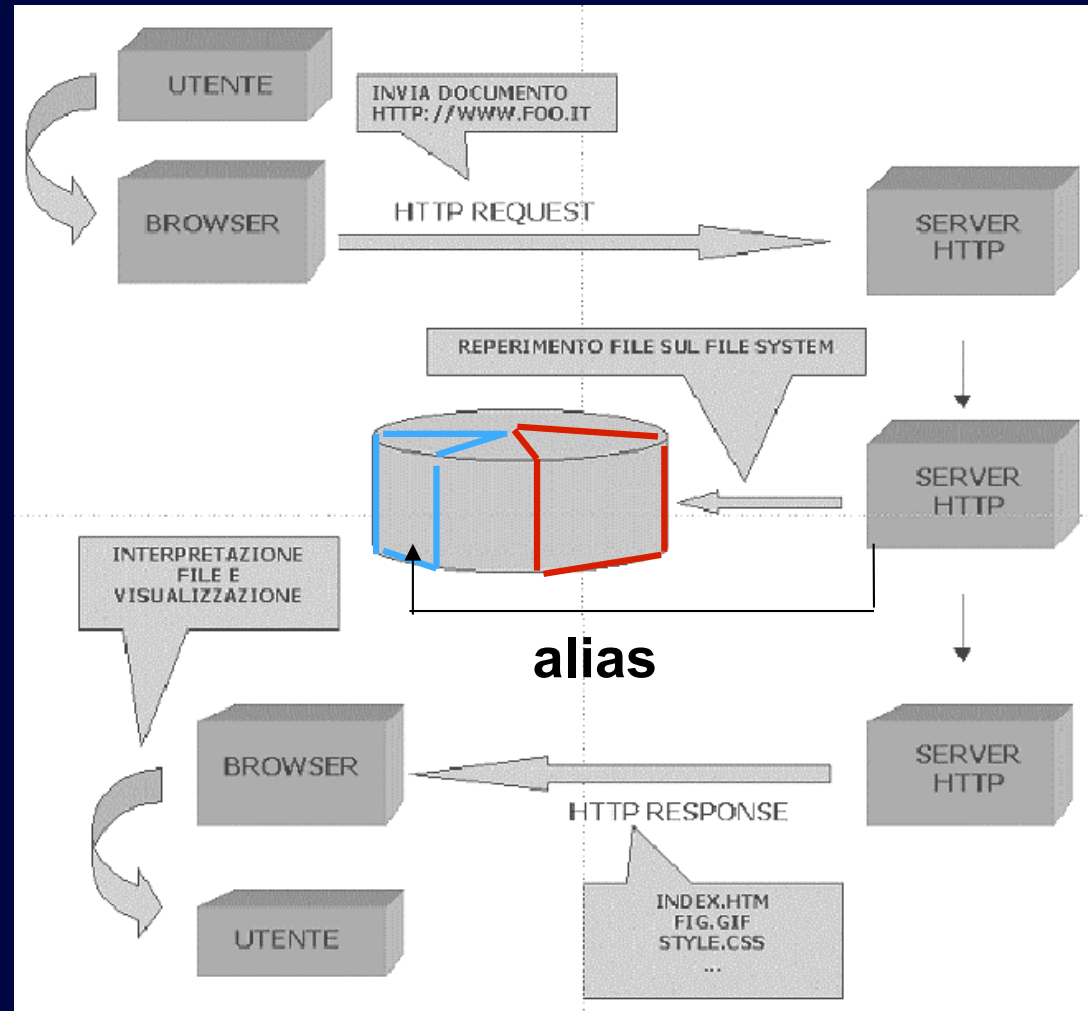
# Alias e cartelle virtuali

Creare un alias vuol dire creare una corrispondenza tra una URL e una cartella del disco

<http://localhost/esalias/>

Corrisponde alla cartella:

C:\esempioalias



# Apache

- ◆ Il web server Apache è free opensource creato e mantenuto dalla Apache foundation ed è liberamente scaricabile da <http://www.apache.org> dove si può trovare tutta la documentazione
- ◆ Può essere installato sia su Windows che su Unix e MacOS anche se è stato realizzato specificatamente per Unix.
- ◆ Supporta varie tecnologie serverside, occorre però installare il corrispondente modulo. Alcuni moduli sono già installati nella configurazione di default, altri vanno appositamente installati (ad es PHP).
- ◆ Esistono **installazioni integrate di Apache + PHP + MySQL**
- ◆ La versione più recente è la 2.2



# Configurare Apache su Windows

- ◆ L'installazione su Windows è abbastanza agevole
- ◆ Può funzionare correttamente anche senza specifiche configurazioni (a parte installare il modulo per la tecnologia serverside che si intende usare)
- ◆ Per farlo partire(fermare):
  - Selezionare Avvia(Ferma) dal menu Apache nel menu Windows\programmi\
  - Dal prompt dei comandi digitando:  
Net start(stop) apache
  - Dal pannello di controllo\servizi\apache

# Configurare Apache su Windows

- ◆ Si può configurare dal file **httpd.conf**, generalmente reperibile alla voce Apache del menu *Programmi* di windows, oppure nella cartella di installazione di Apache o del kit di installazione
- ◆ E' un file di testo editabile con blocco note
- ◆ E' suddiviso in varie sezioni:
  - Global: parametri globali del server, si modificano solo per esigenze particolari
  - Main server: funzionamento del server web
  - Virtual hosts: eventuali altri host virtuali gestiti dal server

# httpd.conf – sezione Main Server

In questa sezione troviamo i parametri di configurazione principali:

`Port 80` (`Listen 80` dalla versione 2.0 in poi)

Questo parametro setta la porta di funzionamento, di default la 80

`ServerAdmin admin@localhost`

Indirizzo email dell'amministratore per eventuali segnalazioni

# Main Server

**Il nome del server**

```
ServerName kddport.isti.cnr.it
```

**La cartella di default del server web.**

```
DocumentRoot "/Applications/MAMP/htdocs "
```

**Quando si cambia questa direttiva va cambiata anche la riga**

```
<Directory "/Applications/MAMP/htdocs">
```

**File di configurazione per directory**

```
AccessFileName .htaccess
```

# Main Server - Alias

```
<IfModule mod_alias.c>
```

```
Alias /icons/ "C:/Program Files/Apache Group/  
Apache/icons/"
```

```
<Directory "C:/Program Files/Apache Group/  
Apache/icons">
```

```
Options Indexes MultiViews
```

```
AllowOverride None
```

```
Order allow,deny
```

```
Allow from all
```

```
</Directory>
```

# Configurare Apache

- ◆ Editare il file **httpd.conf** per le modifiche alla configurazione
- ◆ Riavviare apache ad ogni modifica
- ◆ Se attivate apache o un qualunque server web su un PC continuamente collegato ad internet, considerare i problemi di sicurezza!
  - Installare un personal firewall e tenere il server web costantemente aggiornato con opportune *patches*, controllare periodicamente i *logs*
  - Generalmente IIS è più soggetto a buchi di sicurezza rispetto ad Apache
  - **Nessun server è totalmente sicuro, una volta che è online!!**

# Mettere in linea una applicazione web

- ◆ **Nella fase di progettazione della applicazione web una fase importante e' la scelta dell'hosting del sito.**
- ◆ **La scelta può variare tra tenere un proprio server in rete oppure usare lo spazio fornito da un provider (gratuito o a pagamento)**
- ◆ **Queste scelte hanno vantaggi e svantaggi complementari:**
  - Il server proprio permette maggiori possibilità di personalizzazioni e flessibilità nella configurazione, ma maggiori costi di mantenimento
  - Il server in hosting scarica il webmaster dal peso dell'installazione e il mantenimento del software e dell'hardware, ma non permette flessibilità e personalizzazione della configurazione
- ◆ **Molti provider forniscono gratuitamente o a basso costo il supporto per PHP/MySQL**