

PARTE H
HTTP E
WEB DINAMICO

Modulo 1

Cenni di HTTP

Protocollo HTTP

- **HyperText Trasmission Protocol (HTTP) è il protocollo che permette il reperimento delle risorse Web**
- **E' un protocollo applicativo di tipo request/reply basato sulla suite di protocolli TCP/IP**
- **Tutti i client e server Web devono supportare il protocollo HTTP per poter scambiare richieste e risposte. Per questa ragione i client e i server Web sono chiamati anche client HTTP e server HTTP**

- **HTTP è un protocollo stateless (senza stato)**
 - il server non conserva nessuna informazione riguardante le richieste dei client passati
 - I protocolli che conservano lo stato sono complessi!
 - La storia passata (lo stato) deve essere memorizzata
 - Se il server/client subiscono un crash, la vista dello stato può essere inconsistente e deve essere ristabilita

Messaggi HTTP

Due tipi di messaggi:

- messaggi di **richiesta** HTTP
 - messaggi di **risposta** HTTP
- **Messaggio di richiesta HTTP: ASCII**

Linea di richiesta
(comandi GET,
POST, HEAD)

Linee
header

Carriage return,
line feed

indicanti la fine del
messaggio

```
GET /somedir/page.html HTTP/1.1  
Connection: close  
User-agent: Mozilla/4.0  
Accept: text/html, image/gif, image/jpeg  
Accept-language: fr
```

(extra carriage return, line feed)

Richiesta HTTP

- **Una richiesta HTTP comprende**
 - metodo
 - URL
 - identificativo della versione del protocollo HTTP
 - insieme di extension header
- **Il metodo specifica il tipo di operazione che il client richiede al server. Il metodo più comune è GET che serve per acquisire pagine Web.**
- **Gli header contengono informazioni aggiuntive, quali la data e l'ora della comunicazione, il tipo di software utilizzato dal client, i tipi di dato che il browser è in grado di visualizzare, per un totale di circa 50 tipi di header differenti.**

Richiesta HTTP

- **Method: tipo di operazione richiesta dal client**
 - GET: richiesta di un oggetto
 - POST: il client richiede una pagina Web il cui contenuto è specificato dall'utente (es. richiesta ad un motore di ricerca) nel campo entity body
 - HEAD: il client richiede che il server invii soltanto l'header della risposta senza l'oggetto (usato per debugging dei Web server)
 - PUT, DELETE
 - LINK, UNLINK (HTTP 1.0)
 - TRACE, CONNECT, OPTIONS (HTTP 1.1)
- **URL: identificatore dell'oggetto richiesto**
- **version: versione del protocollo HTTP**

Metodi delle richieste

<i>Metodo</i>	<i>Richiesta</i>	<i>Versione protocollo</i>
GET	Ricevere una risorsa dal server	Dalla HTTP/0.9
HEAD	Ricevere il solo header di una risorsa	Dalla HTTP/1.0
POST	Appendere un oggetto ad un altro sul server	Dalla HTTP/1.0
PUT	Inviare un oggetto al server	Dalla HTTP/1.1
DELETE	Cancellare un oggetto dal server	Dalla HTTP/1.1
LINK e UNLINK	Creare o eliminare collegamenti fra oggetti del server	Dalla HTTP/1.1
TRACE	Individuare la catena dei server proxy	Dalla HTTP/1.1

Messaggio di richiesta HTTP (cont.)

Header lines

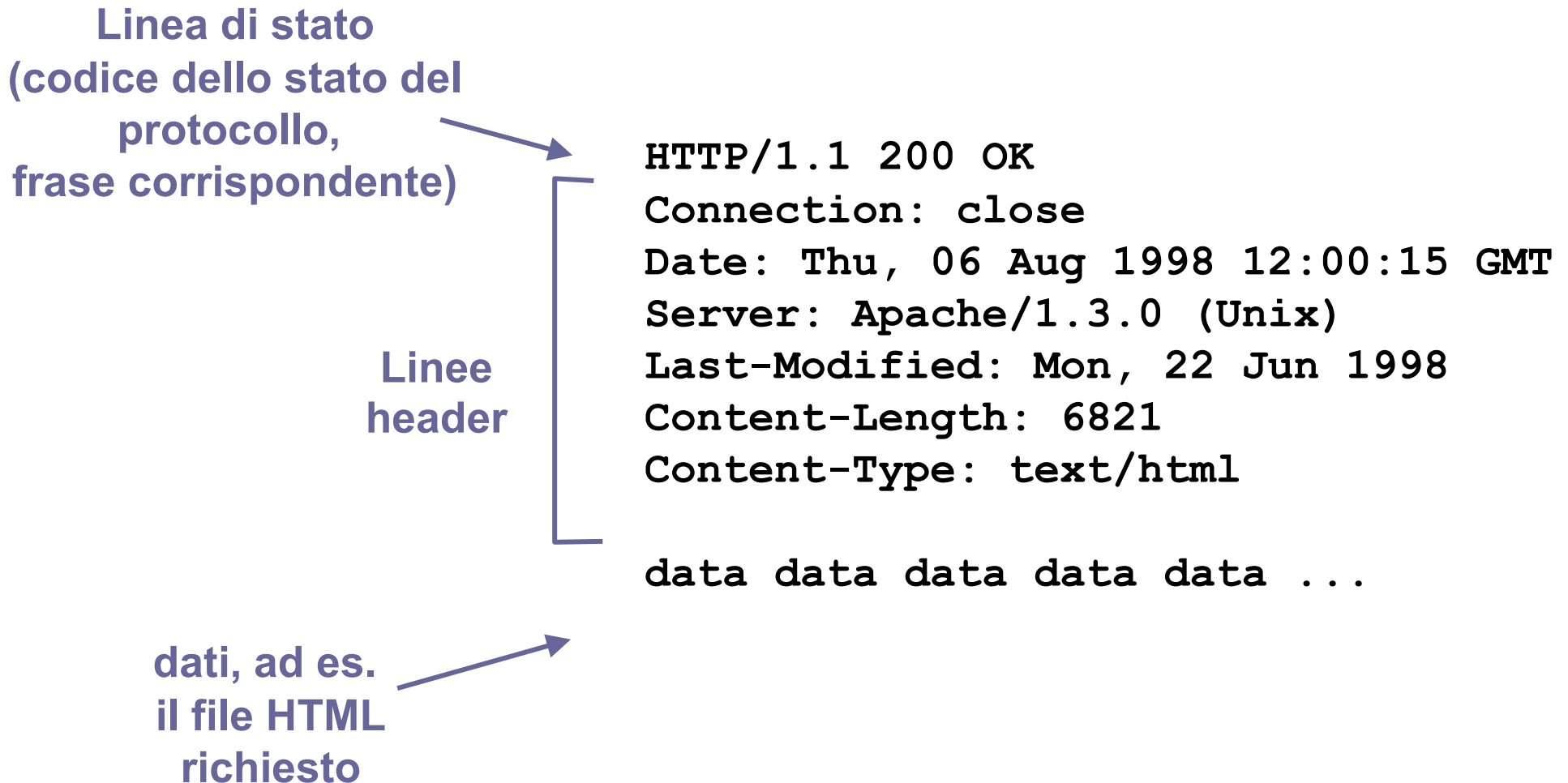
Connection: tipo di connessione richiesta dal client (persistente, non-persistente)

- User-agent: browser utilizzato dall'utente
- Accept: tipo di oggetti che il client accetta
- Accept-language: preferenza della lingua
- Accept-encoding, Accept-charset
- Host: specifica host che ha la risorsa (HTTP 1.1)
- Altri header per garantire la consistenza delle informazioni (es., If-Match o If-Modified-Since)

Risposta HTTP

- **Una risposta HTTP comprende, oltre al contenuto della risorsa richiesta, un header contenente l'identificativo della versione del protocollo HTTP, il codice di stato, l'informazione di stato in forma testuale, ed un insieme di possibili altre informazioni di risposta.**
- **Se la pagina richiesta, oltre al testo HTML, contiene altri oggetti, ciascuno di essi sarà identificato da un URL differente, per cui è necessario che il browser invii un esplicito messaggio di richiesta per ognuno degli elementi collegati alla pagina.**

Messaggio di risposta HTTP



Messaggio di risposta HTTP (cont.)

Version: versione del protocollo HTTP

Status code, phrase: esito della richiesta
(codice e frase)

1xx: Informazioni

2xx: Successo

3xx: Redirezione

4xx: Errore del client

5xx: Errore del server

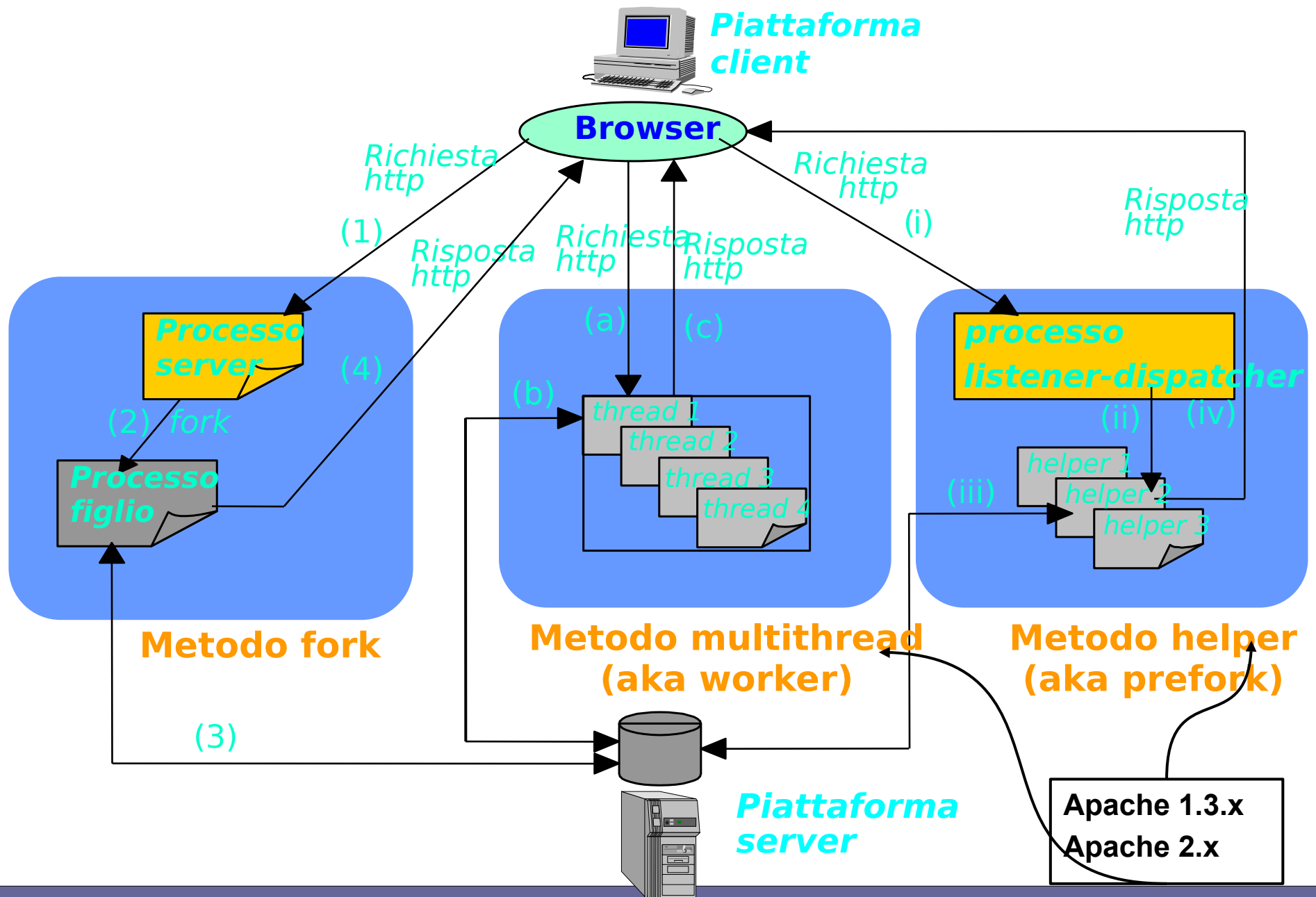
Header line

- Connection: tipo di connessione usata dal server
- Date: data e ora della richiesta
- Server: tipo di Web server e di sistema operativo
- Last-Modified: data e ora creazione o modifica dell'oggetto (caching)
- Content-Length: dimensione in byte dell'oggetto
- Content-Type: tipo di oggetto (es. HTML, GIF, ...)
- **Entity body: oggetto**

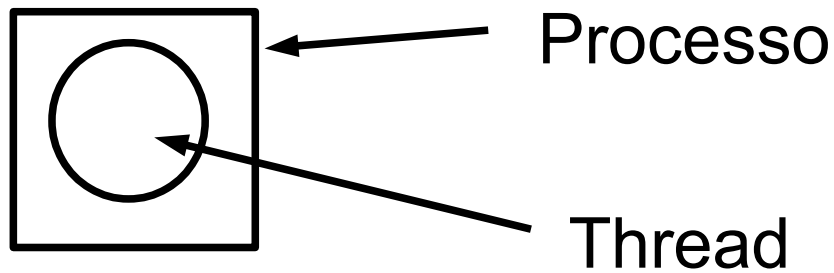
Gestione di richieste HTTP

- **Il Web è basato su un'architettura client-server**
- **Un processo particolare (Web server) si occupa di processare le richieste di risorse Web**
- **Le richieste sono veicolate attraverso il protocollo HTTP (Hyper Text Transfer Protocol)**
- **Ci sono modi diversi di implementare un Web server**
- **Ci sono diversi Web server**

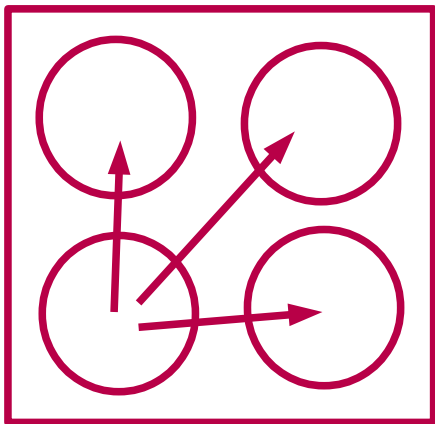
Tre modalità di gestione richieste HTTP



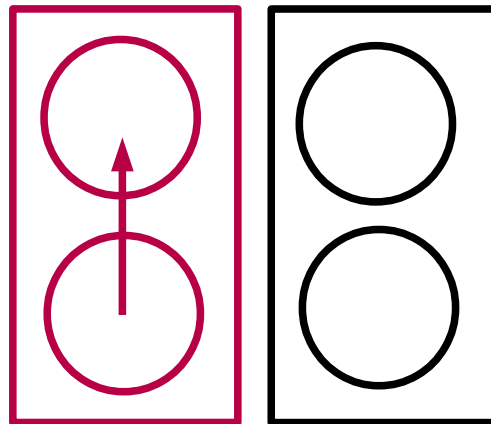
Due parole su sicurezza e prestazioni



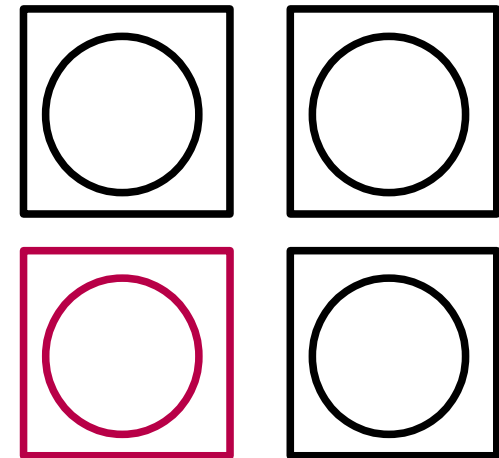
**un processo
molti thread**



**molti processi
più thread
per processo**



**molti processi
un thread
per processo**



Nuova frontiera: gestione event driven

- **Gestione delle richieste event-driven**
- **Un solo processo**
- **Un solo thread (o un thread per ogni core)**
- **Ogni azione necessaria al servizio della richiesta è definita come una funzione**
- **Meccanismo che associa eventi a funzioni (mediante puntatori)**
 - NOTA: completare l'esecuzione di una funzione è a sua volta un evento
- **Sistema che garantisce massime prestazioni**
- **Comunemente usato in**
 - Apache (event-based MPM)
 - Nginx

Modulo 2
Richiami sul
Web dinamico

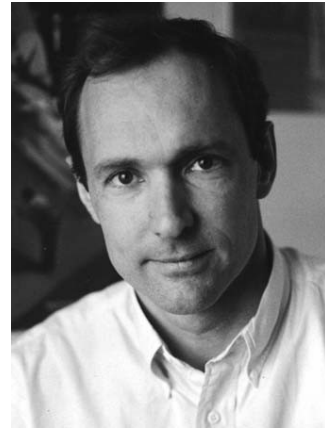
I veri motivi alla base del successo del Web

- **Digitalizzazione dell'informazione**
 - (Qualsiasi informazione come sequenza di 0 e 1)
- **Diffusione di Internet (dagli anni '70)**
 - (Trasporto dell'informazione ovunque, in tempi rapidissimi e a costi bassissimi)
- **Diffusione dei PC (dagli anni '80)**
 - (Accesso, memorizzazione ed elaborazione dell'informazione da parte di chiunque a costi bassissimi)
- **Interfacce utente semplificate (dagli anni '80)**

Quando e perché nasce

Tim Berners-Lee (1989, CERN di Ginevra)

- “The current incompatibilities of the platforms and tools make it impossible to access existing information through a common interface, leading to waste of time, ...”
- “A link is specified as an ASCII string from which the browser can deduce a suitable method of contacting an appropriate server. When a link is followed, the browser addresses the request for the node [document] to the server.”



Ingredienti del Web

- **Meccanismi di comunicazione e naming di Internet**
 - Protocollo TCP/IP
 - Sistema DNS
- **Sistema client-server**
- **“Solo” tre nuovi standard**
 - URL: Sistema di indirizzamento delle risorse
 - HTML: Linguaggio di markup ipertestuale
 - HTTP: Protocollo per le richieste risorse

Motivazione del successo

- **Il Web trascende la sua funzione di tecnologia per**
 - semplificare il request/reply di file statici (get)
 - facilitare il reperimento di file (navigazione point&click)
- **Per diversi motivi, la tecnologia Web è ormai l'interfaccia eletta per l'interazione della maggior parte (tutti?) dei servizi che si possono usufruire via rete:**
 - interrogazioni: a database, a motori di ricerca, ...
 - comunicazioni: posta elettronica, blog, ...
 - download/upload file
 - fruizione di contenuti multimediali
 - svariati applicativi: dai gestionali in su
 - ...

Evoluzione

- **Terminale “stupido” di mainframe**
- **Client/server, dove il client va inteso come software specifico per interagire con quel server. Forte interazione tra i due software.**
- **Il client diventa un software browser-like e il server è in grado di rispondere**
- **Web services**

Dipendenza totale

Forte interazione

Interazione lasca

Completo svincolo

Driver dello sviluppo

- **Ridurre il tempo per rendere disponibili nuovi servizi**
 - Applicazioni mash-up
 - Composizione di servizi
- **Rinnovamento ed evoluzione dei servizi esistenti molto rapido**
 - Sistemi generalisti e facilmente modificabili
 - Sviluppo basato su framework molto flessibili

Modulo 3
Classificazione dei sistemi
per il Web dinamico

Possibili utilizzi

- **Creazione di pagine personalizzate dinamicamente nel momento in cui vengono richieste in base a diversi parametri (es., dati client, ora/giorno, stato del sistema)**
- **Accesso ad informazioni gestite da server non HTTP, come i database ed altre applicazioni**
- **Interrogazioni a motori di ricerca**
- **Interazione personalizzata tra utente e server, che consente all'utente di effettuare accessi riservati, ricerche, acquisti e transazioni**

→ Il Web come interfaccia di servizi sempre più sofisticati

Risorse dinamiche

- **Alcune risorse Web non sono file multimediali, ma richiedono l'esecuzione di (uno o più) programmi**
- **L'aspetto importante è che l'utente non ha bisogno di rendersi conto che l'URL richiesto corrisponde ad un programma né che vi sia l'interazione con due o più server, in quanto il server HTTP trasmette il risultato dell'esecuzione e non il programma (\neq risorse attive)**
 - **RICORDARE L'OBIETTIVO TRASPARENZA**

Livelli logici di un servizio Web-based

- **In un servizio si possono approssimativamente distinguere, a livello logico, le seguenti componenti:**
 - Interfaccia utente, che rappresenta ciò che l'utente percepisce attraverso i propri sensi interagendo con l'applicazione
 - Logica di presentazione, che rappresenta quello che accade quando l'utente interagisce con l'interfaccia

Livelli logici di un servizio Web-based

- **... le seguenti componenti:**
 - Logica dell'applicazione, cioè le operatività cui è preposta l'applicazione (business logic)
 - Logica dei dati, cioè la gestione fisica dei dati (aggiornamenti e ricerche), compresa la loro validazione attraverso verifiche di completezza ed integrità
- **Questi livelli si trovano anche nelle applicazioni più moderne:**
 - Sviluppo con approccio MVC
 - Applicazioni multicanale
 - ...
- **Possiamo riconoscere gli stessi livelli in contesti diversi**
 - Es. dentro a un Web Application Server

Non confondere i “livelli logici” con i “processi” che realizzano i livelli logici e con i “computer” che eseguono i processi.

Quindi,

- Livelli logici
- Processi
- Computer

Alternative per mappare 4 livelli logici sui processi (Visione server side)

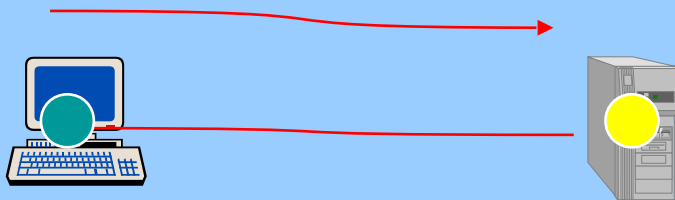
- **Tutto su di un processo (teorico)**
- **Con due processi distinti**
 - Processo client: gestisce il livello “interfaccia utente”
 - Processo server: gestisce i 3 livelli (presentation, application, data)
- **Con tre processi distinti**
 - Processo client: gestisce il livello “interfaccia utente”
 - 1° processo server: gestisce i livelli presentation e application
 - 2° processo server: gestisce il livello data

Alternative per mappare 4 livelli logici sui processi (Visione server side)

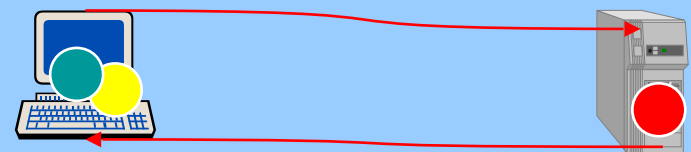
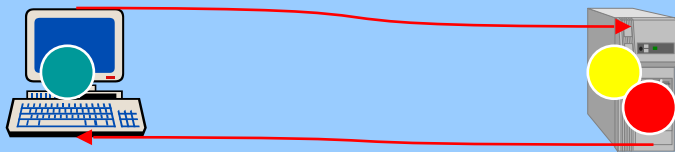
- **Con quattro processi distinti**
 - Processo client: gestisce il livello “interfaccia utente”
 - 1° processo server: gestisce i livelli presentation
 - 2° processo server: gestisce il livello application
 - 3° processo server: gestisce il livello data

Alternative per mappare 2 o più processi su 2 computer

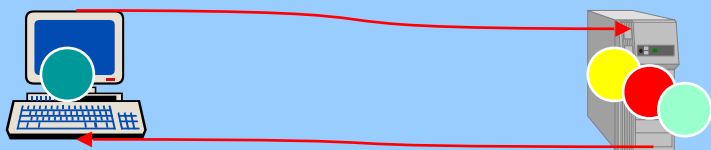
Esempio (2 processi – 2 computer)



Esempio (3 processi – 2 computer)

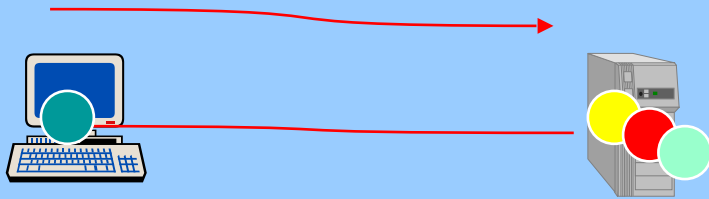


Esempio (4 processi – 2 computer)

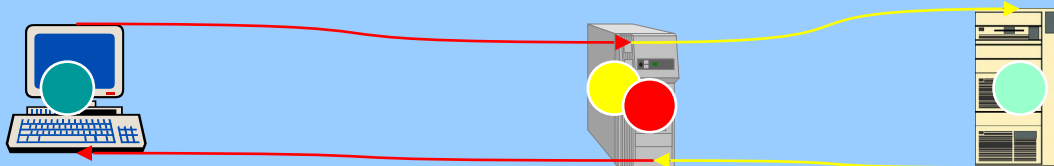


Alternative per mappare 4 processi su 2 o più computer (cont.)

Esempio (4 processi – 2 computer)



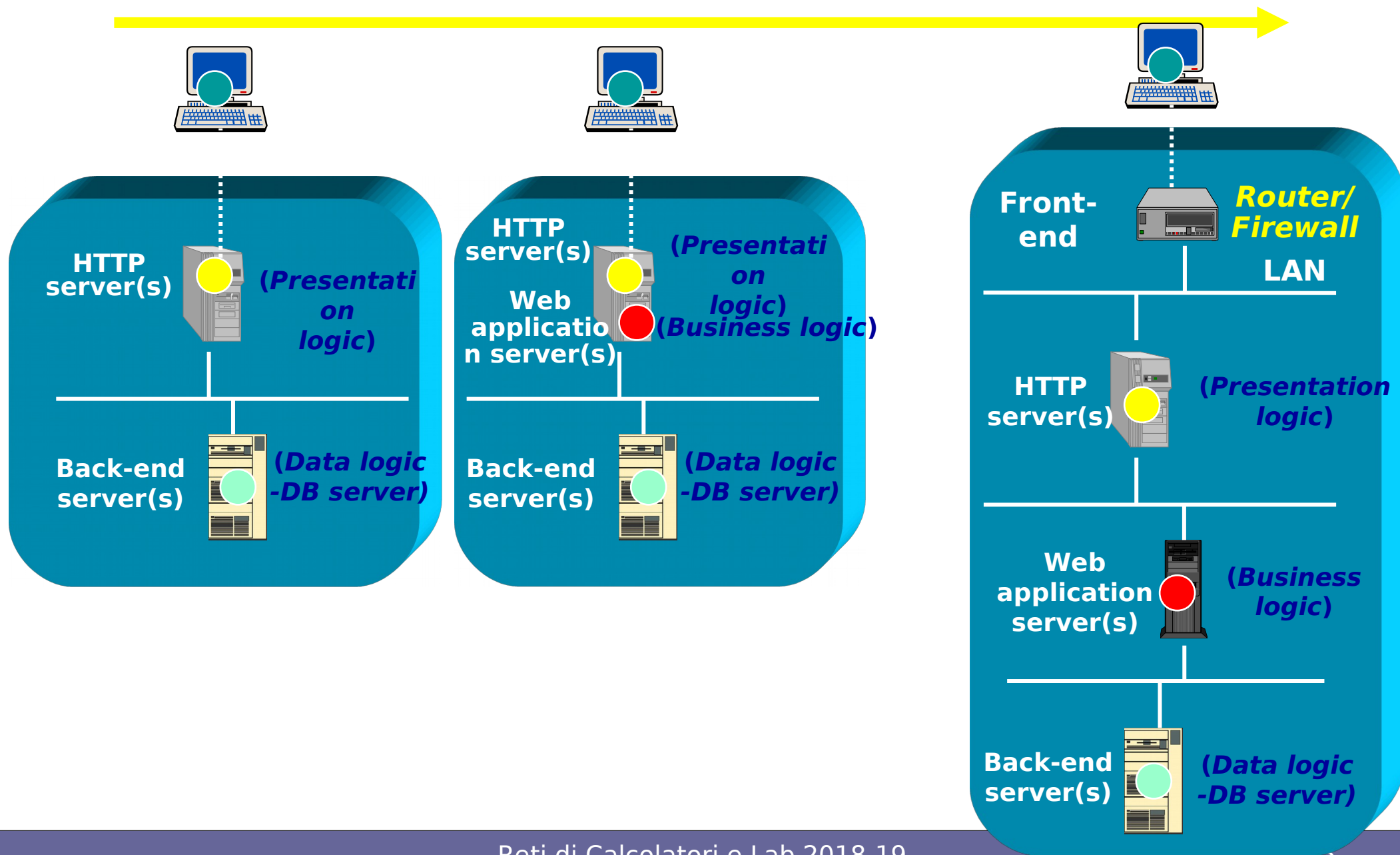
Esempio (4 processi – 3 computer)



Esempio (4 processi – 4 computer)



Evoluzione architetture lato Web server



Una strada alternativa

- **Spostare parte della presentation logic sul client**

Migliora l'interattività

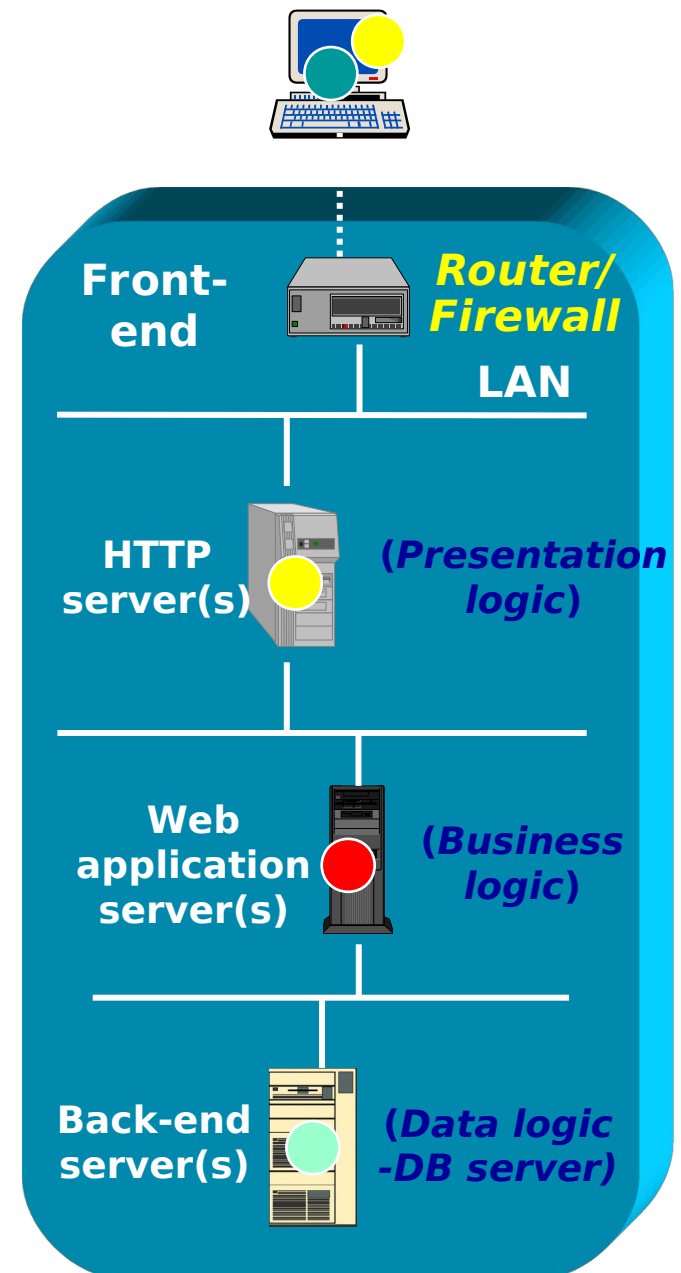
Riduce le latenze di risposta

Consente update asincroni della pagina

- **Approccio molto usato nelle tecnologie con forte dinamicità**

Tipicamente in contesti Web 2.0

Trend anche noto come “La vendetta delle applet”



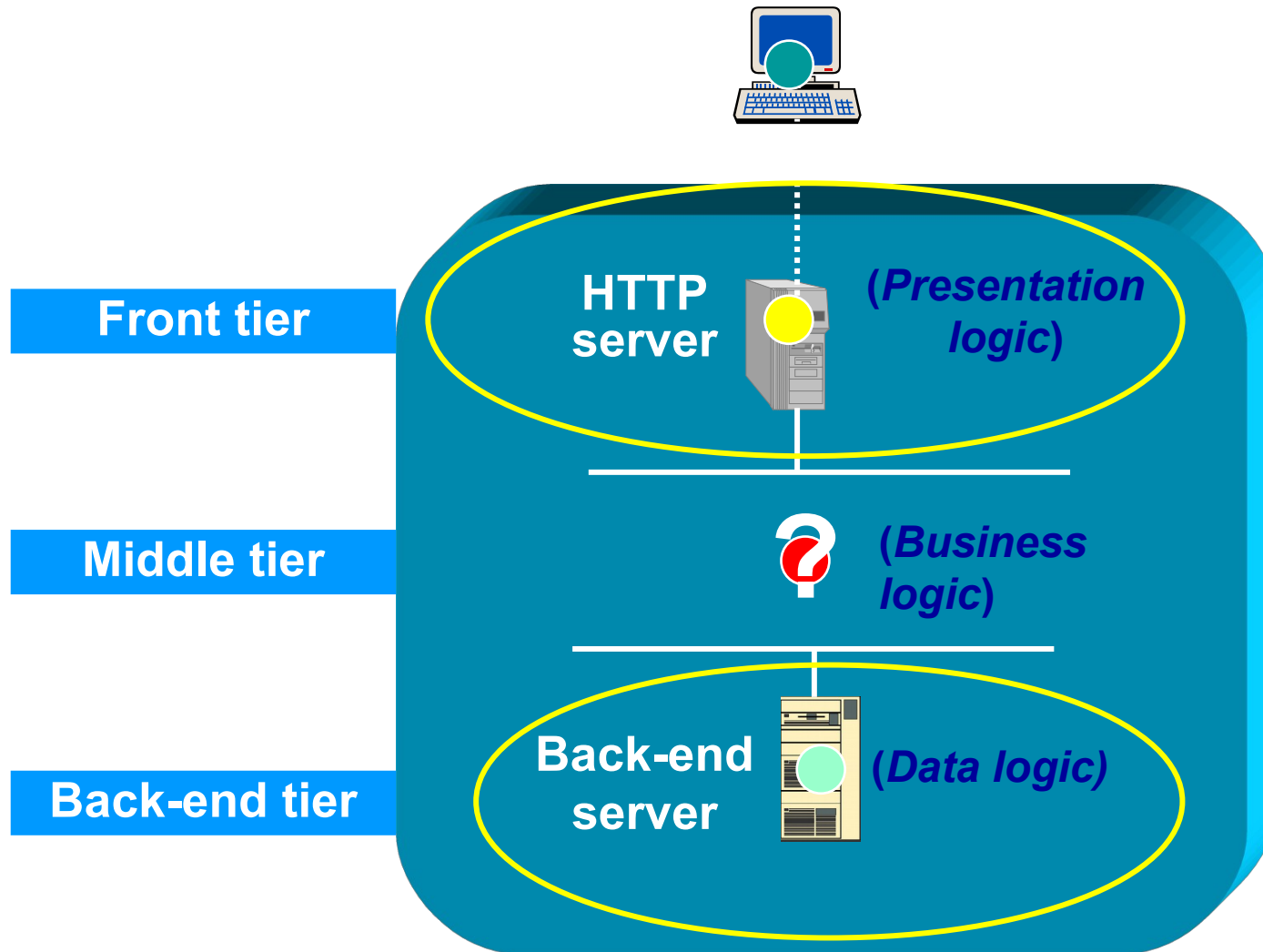
Livello Presentation Logic ***(Visione server side)***

- **Agisce come un'interfaccia tra il livello di interfaccia utente e la business logic**
- **Implementato mediante:**
 - HTTP server
 - Apache
 - Nginx
 - Microsoft IIS
 - (eventuali) Business-logic plugins

Livello Data logic (Visione server side)

- **Gestisce dati strutturati su supporti di memoria permanente**
- **Implementato mediante:**
 - Database Management systems
 - Microsoft Sql server
 - IBM DB2
 - Oracle
 - MySql/MariaDB (open source)
 - Postgres (open source)
 - ...
 - Database NoSQL
 - Sistemi legacy

Livelli definiti: Presentation & Data



- **PROSSIMO OBIETTIVO**
- **Analizzare le tecnologie software che consentono la realizzazione di cosiddetti sistemi Web multi-livello (multi-tier) per realizzare servizi Web-based dinamici**

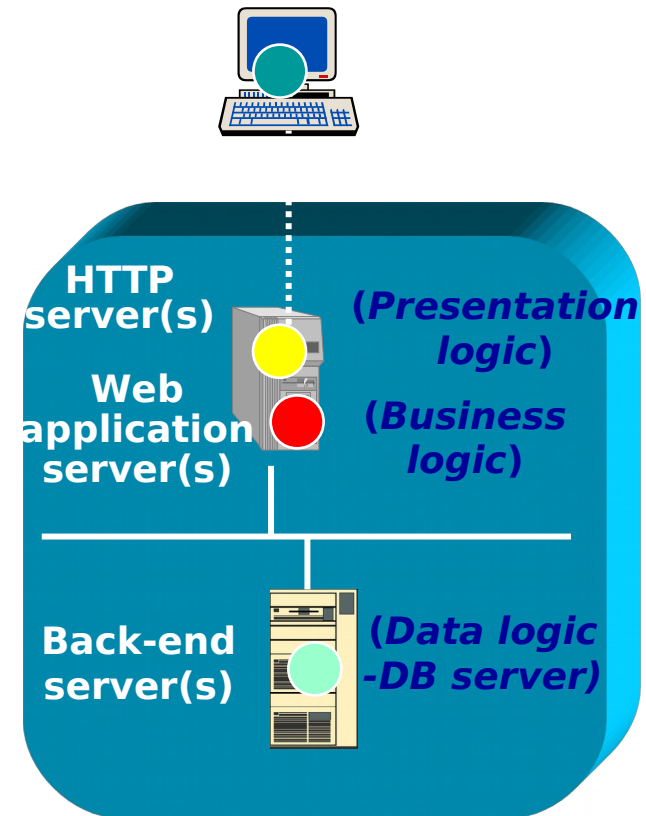
Tecnologie per middle tier

- **Tecnologie che prevedono processi esterni al server HTTP**
- **Linguaggi di scripting**
- **Tecnologie distribuite ad oggetti**

Tecnologie per middle tier (I)

- Tecnologie che prevedono processi esterni al server HTTP
 - basate su Common Gateway Interface (CGI): script in C o Perl (molto popolare, perché?)
 - il server HTTP crea un nuovo processo per il programma CGI che dovrà essere eseguito (copia programma, allocazione memoria, ambiente di variabili separato, ecc.)

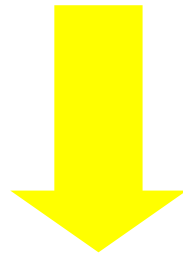
→ TECNOLOGIA NON SCALABILE
(per un server che deve gestire molte richieste, questo è inaccettabile)



CGI-based
**(Presentation+
Business processi
separati)**

Primo passo di evoluzione

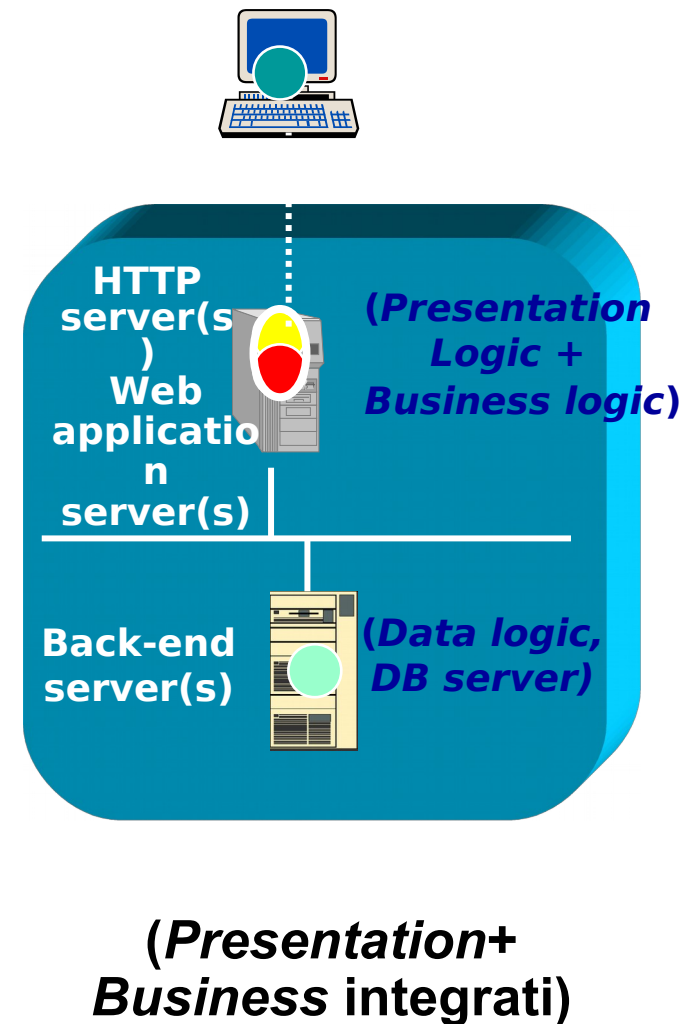
- **Se il problema è la scalabilità limitata**
- **Se il limite alla scalabilità è dovuto principalmente alla necessità di attivazione di un nuovo processo per ogni richiesta dinamica**



- **Si aumentino le prestazioni, mediante l'utilizzo di tecnologie che evitino la creazione di un nuovo processo**
- **→ Tecnologie di scripting**

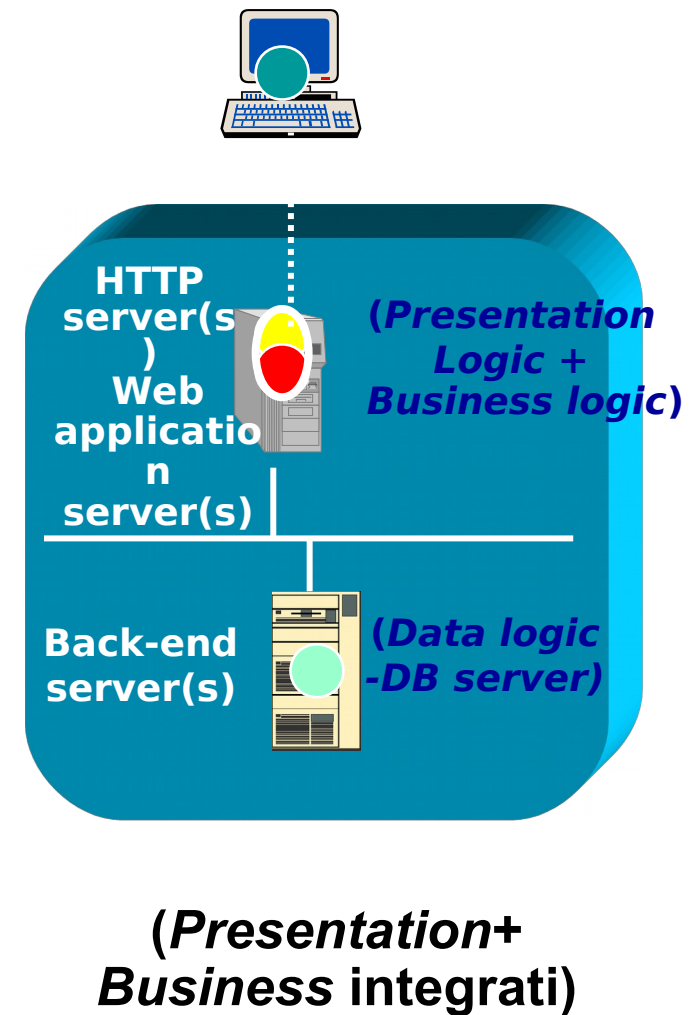
Tecnologie per middle tier (II)

- FastCGI: permette di condividere un'istanza di un programma CGI
- Server API (Netscape NSAPI, MS ISAPI): Librerie condivise caricate nello spazio del server HTTP, in grado di servire richieste multiple senza creare nuovi processi → poca portabilità (legate al server HTTP) e vulnerabili (thread safe)
- mod_perl, mod_python, mod_php: moduli di Apache, in grado di interpretare script all'interno del processo HTTP



Tecnologie per middle tier (II bis)

- Linguaggi di scripting (codice inserito nelle pagine HTML ed interpretato direttamente dal server HTTP)
 - Active Server Pages (ASP, MS): scritte in HTML o VBScript → Soluzione specifica per server HTTP (IIS), problemi di manutenibilità del software HTML/script integrato
 - Java servlets: programmi server side in grado di servire multiple richieste HTTP con un solo processo (come un applet multi-thread eseguito lato server)
 - Java Server Pages
 - PHP “Programming Language of 2004”



Secondo passo di evoluzione

- **Con l'evoluzione tecnologica delle piattaforme hardware (CPU, dischi, RAM, cache), l'aspetto prestazionale delle applicazioni Web-based rimane un fattore importante, ma aumentano anche le aspettative sui servizi che si vogliono poter erogare via Web**
- **La complessità del middle tier aumenta considerevolmente (non più solo accesso a un DB, ma a DB multipli, Insiemi di file XML, directory service, ecc.)**
 - **si sviluppa una vera e complessa business logic**
- **Applicazioni sempre più complesse**
 - **necessità di modularità, portabilità (spaziale e temporale), manutenibilità e riutilizzo del software**

Secondo passo di evoluzione (2)

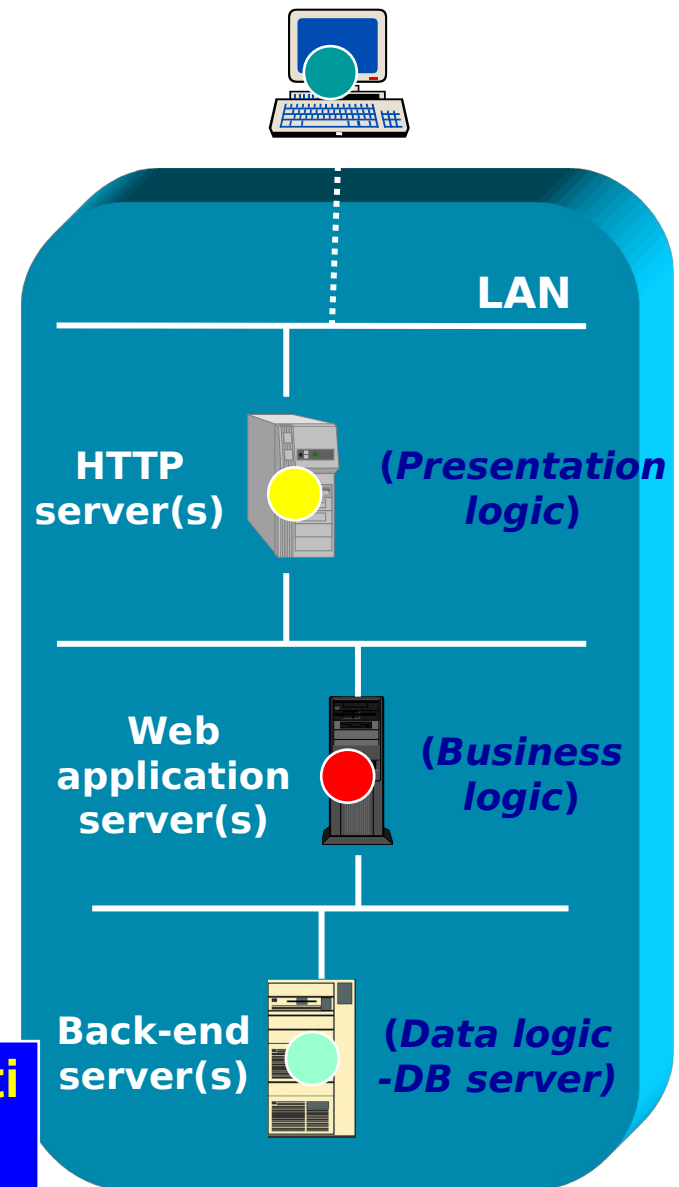
- **Le tecnologie di scripting, che mirano principalmente all'aumento delle prestazioni, non soddisfano i nuovi requisiti richiesti dalle nuove applicazioni software**
- **La business logic complessa rende necessario una separazione dal processo server HTTP**
- **Attenzione: Scopi diversi. Non è un ritorno al CGI!**
- **Qual è la tecnologia che più risponde ai requisiti di:**
 - Modularità
 - Portabilità
 - Manutenibilità



Tecnologie per middle tier (III)

- **Tecnologie distribuite ad oggetti**
 - Ogni oggetto ha un'interfaccia (insieme di metodi pubblici)
 - Il codice può essere eseguito dove serve (portabilità)
 - Promuove la generalità: molte applicazioni possono usare business object comuni
 - Migliora la manutenibilità: con buona interfaccia, cambiamenti influenzano solo l'oggetto

Distribuite a oggetti
(Presentation+
Business separati)

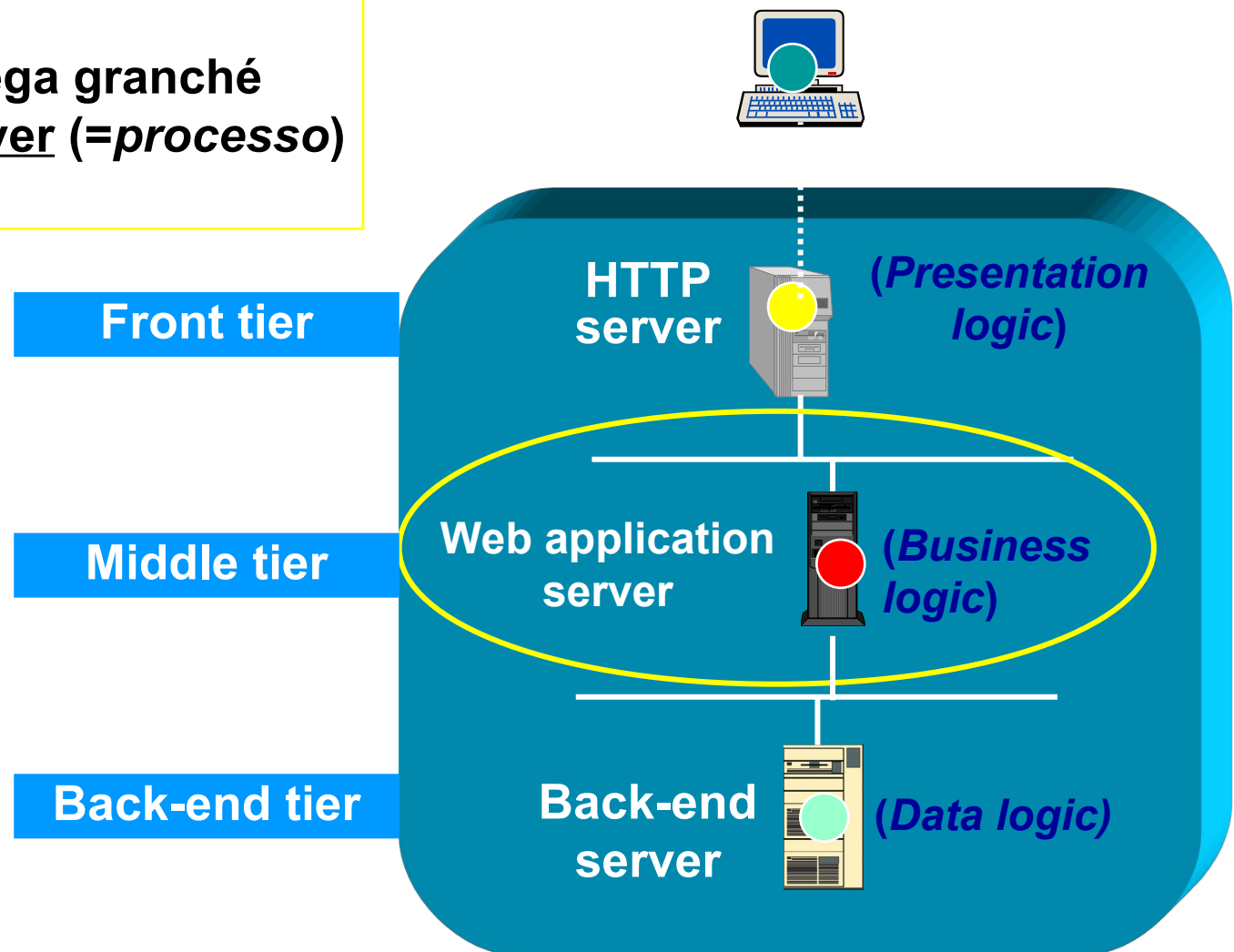


Nasce il Web Application Server

Web Application Server: “un Web server che ha una business logic”

Due note:

la definizione non spiega granché
si parla sempre di server (=processo)
non di computer!



Web Application Server (WAS)

- **WAS**
 - Implementa la business logic del sito Web, ovvero tutte le possibili funzioni che sottendono le operazioni “dinamiche”
 - Traduce le richieste utente in operazioni che interagiscono con applicativi e/o con il livello data logic
- **A dir la verità, si esce dal concetto di tecnologia per il solo “sito Web”**
 - l’insieme delle tecnologie Web viene utilizzata per interfacciarsi all’intero SISTEMA INFORMATICO
- **Implementato mediante:**
 - Tecnologie middleware (una miriade)

Sintesi tecnologie principali

- **Tecnologie che prevedono processi esterni al server HTTP**
 - Tecnologia Common Gateway Interface (CGI)
 - Servlet Java
- **Linguaggi di scripting (codice inserito nelle pagine HTML ed interpretato direttamente dal server HTTP)**
 - Active Server Pages (ASP, Microsoft) Altre
 - PHP FastCGI
 - Java Server Pages (JSP, Oracle) Tecnologia server API
- **Tecnologie distribuite ad oggetti** Es., Netscape (NSAPI)
 - Tecnologia distribuita ad oggetti (CORBA)
 - Java2 Enterprise Edition (J2EE)
 - Microsoft .NET ColdFusion

Altra possibile tassonomia

- **Classificazione sulla base del focus nella fase di sviluppo**
 - Orientato alla pagina (es. PHP, JSP naive)
 - Object oriented (EJB naive)
 - MVC-oriented (framework specifici)
- **Classificazione sua base dell'implementazione**
 - Integrato nel Web server (es. PHP)
 - Processo esterno (es. CGI)
 - Server esterno (es. Fast CGI)

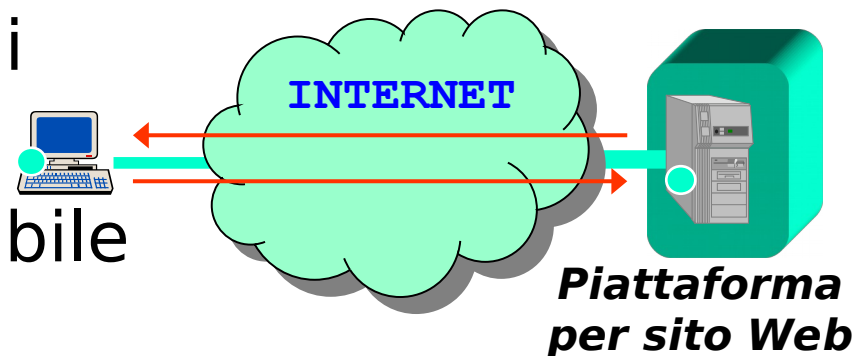
Modulo 4

Le generazioni del Web

Le generazioni del Web (<1998)

- **1^a generazione = Web publishing**

- Un ulteriore canale per informazione non critica
- 95% dell'informazione costituita da testo più eventuali immagini
- Siti Web prevalentemente statici con alcune tecnologie (es., CGI) per funzionalità dinamiche
- Manutenzione e aggiornamenti occasionali
- Canale economico
- Prestazioni molto variabili
- Affidabilità non garantita
- Sicurezza non indispensabile



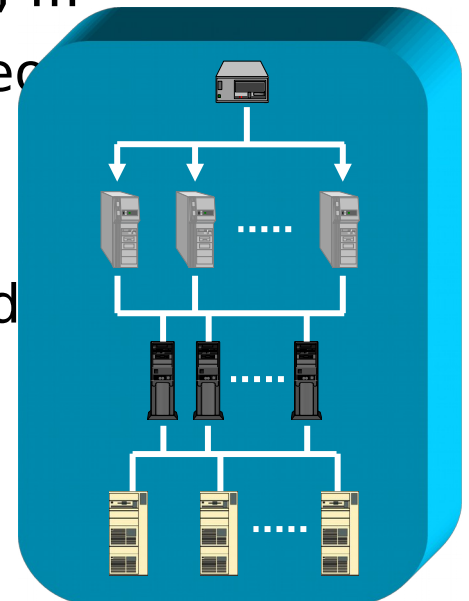
Le generazioni del Web (1998-2005)

- **2ª generazione = Web-based Information Systems**

- Canale di informazione critica, che sta diventando un mezzo di comunicazione privilegiato per molti utenti
- “Vetrina” importante per industrie e organizzazioni
- Contenuti dinamici ed attivi in continuo aumento
- Interfaccia per molti altri servizi informatici usufruiti via rete (anche se non propriamente di rete)
 - Posta elettronica, trasferimento file, connessioni, ...
 - Accesso a archivi, basi di dati, banche dati, ...
- Introduzione di servizi di streaming audio e video
- Servizi personalizzati, servizi a pagamento, ...

Quindi: necessità di qualità del servizio per

- Prestazioni garantite (regola degli 8 secondi)
- Affidabilità (capacità di tollerare guasti)
- Sicurezza



Quanto è affidabile il Web?

- **Chiunque può pubblicare sul Web ciò che vuole**
- **Molta dell'informazione disponibile è falsa, fuorviante, ingannevole, faziosa, diffamatoria o disgustosa**
- **Come facciamo a sapere se le pagine che troviamo sono affidabili?**
- **I nomi dei domini registrati possono essere ingannevoli o intenzionalmente truffaldini**
- **Occorre verificare l'identità della persona o dell'organizzazione che pubblica la pagina Web**

Caratteristiche dei siti affidabili

- **Esistenza reale nel mondo fisico.** Il sito fornisce un indirizzo, numero di telefono e indirizzo di posta elettronica
- **Certificazione.** Il sito include riferimenti, citazioni o credenziali, nonché collegamenti a siti altrettanto certificati
- **Chiarezza.** Il sito è ben organizzato, facile da navigare e fornisce servizi come una ricerca interna
- **Aggiornamento.** Il sito è stato aggiornato di recente
- **Professionalità.** La grammatica, l'ortografia, la punteggiatura e l'aspetto grafico sono corretti; tutti i link funzionano
- **Tuttavia, anche un sito che esibisce tutte queste qualità potrebbe non essere affidabile!**
- **NUOVA CULTURA:** saper distinguere in mezzo a tantissime informazioni quelle giuste, corrette, affidabili

Impatti “sociali” del Web

- **La distribuzione dell'informazione si amplia notevolmente: non c'è più bisogno di un editore per pubblicare e distribuire qualcosa (rivoluzione paragonabile all'invenzione della stampa e forse anche di maggiore impatto in quanto non trova una società di analfabeti)**
- **Chiunque riesce a creare e gestire un sito Web contribuisce alla riorganizzazione dell'informazione: non più gerarchica, non più classificata, ma reticolare (ogni link aggiunto modifica la modalità di accesso all'informazione)**
- **Ma non finisce qui ...**

Le rivoluzioni “sociali” del Web

(>2005)

Web 1.0 (“Apertura”)

- Enorme apertura verso l'esterno
- Si infrange il modello tradizionale di redazione , selezione e pubblicazione delle informazioni
- Si amplia moltissimo la platea dei possibili “autori” rispetto al mondo cartaceo, ma la barriera di ingresso rimane abbastanza elevata

Web 2.0 (“Partecipazione”)

- Grazie a nuovi strumenti software, si abbatte ogni barriera di ingresso di tipo tecnologico: “chiunque può essere un creativo (scrittore, cantante, regista, attore, ...) e avere successo mondiale”
- Nasce una platea enorme in grado di ricevere ed offrire informazioni, e di alimentare il Web di nuovi contenuti

Dal Web “read only” al Web “read-write”

- **Verso “piattaforme di collaborazione”**
- **Utenti: da lettori passivi a partecipanti attivi (paradigma utente-centrico)**
- **Contenuti: da risorse standardizzate generate da pochi provider a contenuti eterogenei generati dagli utenti stessi (user generated content)**

Quattro tipologie significative del Web 2.0

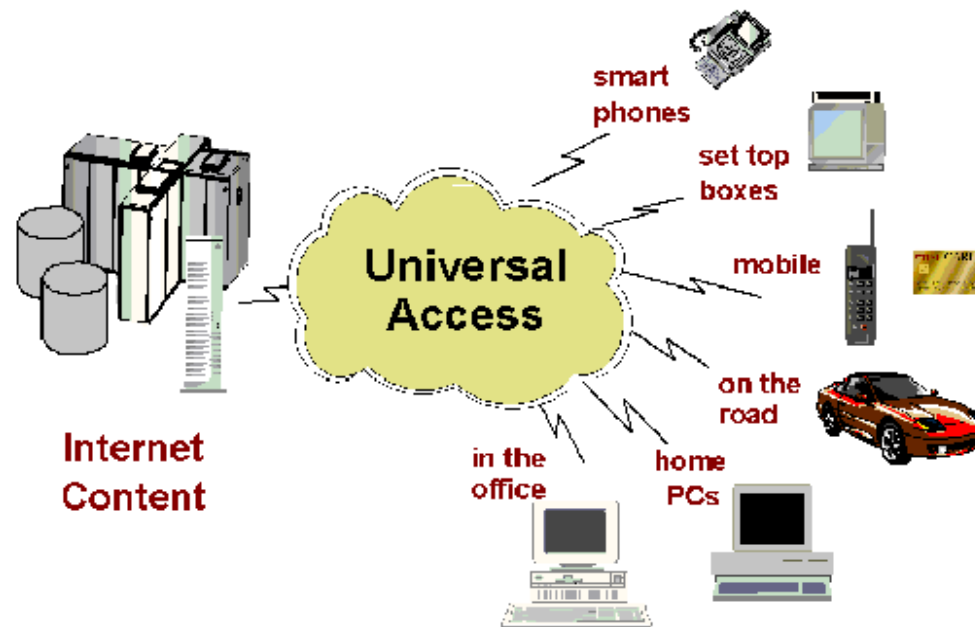
- **Blog**: uno spazio personale, un diario online aggiornato di continuo che tutti possono leggere e commentare
- **Wiki**: veri e propri dizionari collaborativi a cui chiunque è in grado di contribuire aggiungendo o modificando le informazioni presenti
- **Social networking**: comunità di persone accomunate da interessi e legami sociali di vario genere che si scambiano informazioni sotto forma di testo, immagini, audio, video (Facebook, YouTube)
- **Gaming**: si creano delle comunità enormi e vitali sui giochi di tutti i tipi

Le generazioni del Web (oggi/domani)

Stato dell'arte = Ubiquitous Web

Possibilità di usufruire di tutti i servizi Web di 2a generazione:

- anytime - sempre (24/7)
- anywhere - ovunque
- anyway - da qualunque dispositivo



Le generazioni del Web (oggi/domani)

- **Ulteriori elementi di complessità**
 - Uso di piattaforme distribuite (cloud, CDN, ...)
 - Interazione con dispositivi (IoT)
- **Ogni evoluzione porta nuove esigenze**
- **Questo costringe a ripensare le soluzioni tecnologiche**