W3

Introduzione

Un **sistema informativo** è uno strumento tecnologico che ci consente di gestire e condividere informazioni.

Il W3 è un sistema informativo**:**

- **multimediale**: l'informazione non viene memorizzata in un solo formato, ma vengono gestiti tutti i contenuti informativi in un unico contesto.
- **distribuito**: il sistema informatico che gestisce la mia informazione è costituito da più calcolatori caratterizzati da ambienti operativi diversi dislocati in luoghi diversi, ma che operano come se fossero una entità sola nella rete.
- **eterogeneo**: il sistema è costituito da sottosistemi che possono essere diversi a livello di CPU, tipo di programma che eseguono, ecc... Tutti, però, comunicano tramite uno stesso standard di comunicazione (formalmente tramite protocollo HTTP).
- **collaborativo**: s'intende che non esiste una gerarchia di tipo master/slave tra i sistemi, ma c'è un meccanismo di handshacking tra client e server sul formato del dato che viene richiesto dal client. Più precisamente il Client fa la sua richiesta e dice che tipi di formato è in grado di gestire, invece il

Server riceve la richiesta insieme al formato e se può esaudire la richiesta sceglie il formato compatibile tra quelli che ha a disposizione e lo reinvia al client. Il concetto di collaborazione risiede proprio nel fatto che ci si accorda sul dato e sul tipo. Questo concetto si sposa perfettamente al concetto di eterogeneità. Infatti, se ho un nuovo formato di dato, questo viene integrato insieme ai formati attuali, non devo buttare via tutti i client e i server preesistenti e poi aggiornarli (una volta che ho definito il nuovo standard metto a disposizione anche un plugin che si comporta come decodificatore, altrimenti quando il client scarica il mio formato nuovo non è in grado di decodificarlo). Il protocollo HTTP implementa la collaborazione tramite la negoziazione del formato e ottengo un sistema scalabile.

Nel dettaglio possiamo poi dire che il W3 è:

- un **principio**, ovvero un concetto astratto formale.
- un **insieme di protocolli**, ovvero un insieme di tecnologie (in questo caso il termine protocollo è usato in senso ampio cioè come definizione formale di qualche cosa).
- un **insieme di applicazioni,** ovvero le applicazioni che vengono costruite all'interno del mondo web per poter usufruire dei contenuti.
- una **ragnatela di informazioni**.

Principio

Tim Berners-Lee in persona ha definito il principio che sta alla base del W3. Il principio che il web soddisfa è l'**accesso universale all'informazione** . Si intende il superamento del modello precedente dell'informazione presente prima del web. Infatti, prima del W3, per accedere a documenti diversi residenti su computer diversi (dove con diversi intendiamo diverso SO, diverse interfacce e diversi software) occorrevano diversi terminali, collegati a macchine differenti e con software distinti. Invece con l'avvento del W3 e l'accesso universale, se un documento è disponibile lo è da ogni computer in ogni località ed è accessibile da parte di ogni utente autorizzato e attraverso un unico software (denominato *browser*).
La realizzazione di tale principio si basa su
4 pilastri fondamentali:

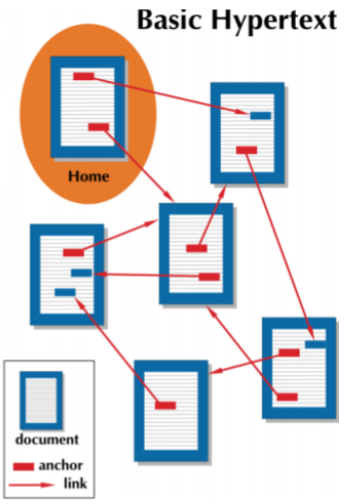
- **Rappresentazione ipertestuale dei documenti**. Un modello unico per la rappresentazione (non codifica) dei documenti (tutti i documenti che il web

veicola sono ipertesti). Il mio documento è un insieme di documenti correlati e l'informazione si compone in maniera incrementale.

- Disponibilità di efficaci **meccanismi di ricerca**
- **Modello Client/Server**. Architettura di sistema per implementare un modello distribuito per l'interazione tra sistemi caratterizzata dalla scalabilità (aggiunta di un nuovo client non varia tutto il sistema)
- **Negoziazione del formato** dei documenti. Vero elemento di novità introdotto. Il protocollo basato sulla negoziazione è il protocollo HTTP.

Ipertesti

I documenti devono essere degli ipertesti. Un **ipertesto** è una tipologia che consiste in un modo di organizzare l'informazione tra più documenti correlati tra di loro tramite link (frecce). Sono testi che, al loro interno, contengono riferimenti ad altri testi.



Non è più un documento monolitico. Questo collegamento lo troviamo in una porzione del documento principale (àncora).

Che vantaggi abbiamo? L'alternativa sarebbe una struttura sequenziale che va dal primo all'ultimo paragrafo in un unico documento. I vantaggi sono:

- **Tecnico**: gli ipertesti consentono di segmentare il contenuto informativo. Queste sotto-pagine possono essere dislocate/memorizzate in luoghi fisicamente diversi ma con la condizione che la macchina su cui risiede

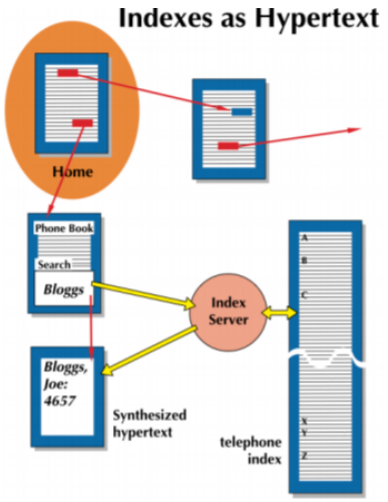
deve essere un nodo della rete (un pc/hard disk connesso al web). Tramite la ridondanza dei file posso irrobustire il mio sistema e renderlo disponibile anche in caso di guasti. Inoltre, un'organizzazione di questo tipo favorisce anche una gestione a livello di team in un'organizzazione (ogni team si prende una sotto-parte dell'informazione che la gestisce e la modifica secondo le necessità senza che debba andare offline l'intero documento).

- **Durante la navigazione:** la rappresentazione ipertestuale del documento consente una **fruizione personalizzata** del documento stesso. Posso saltare di argomento in argomento in funzione delle esigenze informative. Il percorso di acquisizione del contenuto del documento è personalizzabile, più efficace e più efficiente.

Oggi gli ipertesti si sono evoluti in **ipermedia** (contengono elementi di tipo multimediali).

Meccanismi di ricerca

Lee pensava che il web si sarebbe riempito velocemente di contenuti informativi. Si crea il problema di come cercare l'informazione che mi interessa all'interno della mole di informazioni disponibili. Da dove devo partire?
Serve un
meccanismo di *indicizzazione* delle pagine (non andavano bene i modelli di ricerca come quelli per i database). La mia richiesta di ricerca di un file (richiesta sotto forma di testo) arriva ad un server che va a consultare la locazione dove mantiene indicizzati tutti i contenuti e ricerca quello della richiesta. La risposta del server sarà un riferimento ad un documento ipertestuale.



Nota bene: quando inseriamo la stringa da cercare si attiva il processo di ricerca in una base di dati che contiene tutte le singole parole che l'utente può ricercare (si attiva direttamente il processo su index server).

Questo processo fa parte di un processo più grande e continuo che è quello dell'indicizzazione delle pagine web.
Fondamentalmente stiamo guardando un motore di ricerca.
Come funziona a grandi linee il motore di ricerca?

Si parte dal meccanismo di "**crawling**" (to crawl = gattonare), quella fase in cui il motore di ricerca va ad esplorare il mondo del web e lo spazio intorno a lui.

> **Web Crawler (Wikipedia)**
>
> Un **crawler** (detto anche **web crawler**, **spider** o **robot**), è un software che analizza i contenuti di una rete (o di un database) in un modo metodico e automatizzato, in genere per conto di un motore di ricerca. Nello specifico, un crawler è un tipo di bot (programma o script che automatizza delle operazioni), che solitamente acquisisce una copia testuale di tutti i documenti presenti in una o più pagine web creando un indice che ne permetta, successivamente, la ricerca e la visualizzazione. Un uso estremamente comune dei crawler viene effettuato sul Web; esso si basa su una lista di URL da visitare fornita dal motore di ricerca (il quale, inizialmente, si basa a sua volta sugli indirizzi suggeriti dagli utenti o su una lista precompilata dai programmatori stessi). Durante l'analisi di una URL, identifica tutti i collegamenti ipertestuali presenti nel documento e li aggiunge alla lista di URL da visitare. Il processo può essere concluso manualmente o dopo che un determinato numero di collegamenti è stato seguito. Inoltre i

> crawler attivi su Internet hanno la facoltà di essere indirizzati da quanto indicato nel file "robots.txt" posto nella root del sito. All'interno di questo file, è possibile indicare quali pagine non dovrebbero essere analizzate. Il crawler ha la facoltà di seguire i consigli, ma non l'obbligo.

Il software di Google scandisce periodicamente, tramite il crawler, tutte le pagine web accessibili, estrae le informazioni e le parole significative e crea un piccolo riassunto delle pagina (anche tenendo conto delle varie lingue diverse); elimina parole inutili (le "stop words" come articoli o pronomi) mantendendo solo le parole semanticamente significative (informazione che ha un significato che concorre alla definizione del contenuto della pagina).

Il crawler crea una "tabella" nella quale ogni riga corrisponde ad una pagina disponibile nel web, e gli attributi di ogni riga sono le parole significative contenute in quella pagina.

Questa tabella è ordinata secondo un criterio che rispecchia il processo di acquisizione del processo su base pagina, ma non è funzionale al processo di ricerca, perché in realtà il processo di ricerca non parte dalla indicazione della pagina a cui siete interessati e produce le parole contenute su quella pagina, ma viceversa parte dalle parole a cui siete interessati e deve produrre l'elenco delle pagine web che contengono quelle parole.
Sostanzialmente è come se noi dovessimo *invertire* la tabella prodotta dai crawler: passiamo da una tabella nella quale ogni riga rapppresenta una pagina e ogni elemento contenuto nella riga rappresenta una parola significativa contenuta all'interno di quella pagina, a un'altra tabella nella quale ogni riga è associata ad una diversa parola e ogni elemento contenuto in una riga contiene l'indirizzo della pagina che nel web ha al suo interno quella parola. Abbiamo bisogna di una struttura dati nella quale la chiave di accesso all'informazione non sia più l'indirizzo della pagina, ma sia la parola, e un'altra struttura dati che fornisca come risultato (a partire da una parola) tutti gli URL delle pagine nel web che contengono questa parola.

In sintesi quindi un motore di ricerca opera con una coppia di processi: un primo processo che estrae l'informazione dal web e un secondo processo che soddisfa le richieste degli utenti che vanno a ricercare all'interno del web i contenuti informativi a cui sono interessati.

Nel crawling avviene anche il processo di **ranking** per *ordinare le pagine indicizzate per importanza*.
-
Se una pagina contiene molti riferimenti alla parola utilizzata come pattern di ricerca, è molto più probabile che sia più interessante rispetto ad una pagina che contiene pochi riferimenti; quindi essa verrà mostrata prima di altre pagine.
- Se la mia pagina contiene una percentuale di parole inserite nel pattern di ricerca che è maggiore rispetto alla media con cui le stesse parole compaiono nelle altre pagine del web, probabilmente il contenuto della mia pagina è specializzato su quelle parole; quindi la mia pagina è particolarmente significativa per la ricerca che l'utente ha impostato.
- Se una pagina contiene un concetto ed ha al suo interno una serie di link (e quindi di àncore) a pagine che contengono lo stesso concetto, quella pagina è più significativa per quel concetto rispetto alle pagine che invece non hanno questo legame con altre sotto-pagine che riportano lo stesso concetto.

L'utente vede solo la fase di ricerca del motore di ricerca e di ottenimento dei risultati.
Un buon motore di ricerca inoltre cerca di selezionare anche sinonimi di parole rilevanti, in modo da non escludere risultati significativi.

Client/Server

L'obiettivo era avere una **struttura scalabile** (capace di adattarsi in maniera semplice ai contenuti da visualizzare) e senza una gestione centralizzata dell'intero sistema web (non ho un computer centrale ma il rapporto tra le macchine è di 1:1); grazie a quest'ultima introduzione si era in grado di ottenere un sistema affidabile e robusto.
Se si vuole aggiungere il proprio sito basta aggiungere un server (nodo) senza necessità di modificare la struttura già esistente.
Inoltre, se un server smette di funzionare solo quel singolo blocco informativo che conteneva diventa indisponibile, tutto il resto è ancora accessibile.

L'infrastruttura che caratterizza il web è un insieme di client, un insieme di server e un'infrastruttura di collegamento (comunicazione) che noi chiamiamo Internet (internet è precedente al web e a Berners-Lee).
La novità sta nel modo in cui concepiamo il mondo dei client e i server:

- **Client**: tutti i client sono di fatto la stessa applicazione: il browser (al massimo declinato in modo diverso i.e. Chrome, Mozilla, Opera, Safari ecc.), ma è un unico punto di accesso comune.

- **Server**: sistemi che parlano la lingua del web. Questa lingua la intendiamo come una serie di protocolli di comunicazione tra client e server che il browser web è in grado di gestire (ftp, mailto, http, https ed altri).
  Contiene tutti i meccanismi di scambio dei dati tra Client e Server comprensibili dal browser.
  Per i protocolli legacy (protocolli antecedenti al W3) utilizziamo una tecnica di tipo "wrapping". Un **wrapper** è un programma che, per utilizzare una metafora, si comporta da *traduttore* dal formato che ho a quello che voglio ottenere (chi sta fuori vede solo l'esterno e mentre la vera struttura interna è nascosta, come se fosse *incartato*), consentendo a questi sistemi di parlare con i browser.

  I server si possono dividere in due sottocategorie: quelli già pronti per interfacciarsi con i protocolli (come HTTP) e quelli che non hanno un protocollo corretto. Questi ultimi necessitano di un **gateway**. I gateway sono dei meccanismi di interfaccia che permettono di realizzare questa comunicazione tra sottosistemi a protocolli diversi (legacy); sono fondamentalmente meccanismi di traduzione. Uno dei primi gateway è Common Gateway Interface (CGI) e consente di interfacciare con il web l'informazione prodotta da un qualunque programma scritto ed eseguibile da una qualsiasi macchina, senza ristrutturalo ma semplicemente incartandolo.

Tra Client e Server abbiamo Internet con le sue tre componenti di base:

- lo schema di indirizzamento (IP)
- protocollo comune per lo scambio di info (TCP)
- meccanismo di negoziazione del formato

Questa struttura è lo scheletro del web.

Negoziazione del formato

Si desidera la coesistenza di più standard, senza limitarsi ad un unico standard. Un ambiente che accetta un solo tipo di file non è accettabile; è necessario costruirne uno in grado di sfruttare varie tipologie di formato in modo tale da non limitare il contenuto informativo fruibile. Inoltre, se si volesse modificare il formato in seguito all'uscita di una nuova versione dello stesso, si dovrebbe eseguire una ristrutturazione del browser.
Soluzione? Lasciamo i

formati aperti, senza un unico standard.

Un formato si definisce "*aperto*" quando ne viene resa pubblica la sintassi, la semantica, il contesto operativo e le modalità di utilizzo.

Obiettivi :

- garantire l'accesso ai dati nel lungo periodo senza incertezza presente e futura riguardo ai diritti legali o le specifiche tecniche (interoperabilità).
- incoraggiare la concorrenza invece di consentire a un solo produttore di mantenere il controllo su di un formato proprietario per inibire l'uso di prodotti concorrenti.

A questo punto devo rendere i sistemi coscienti del formato del file che si stanno scambiando. La negoziazione del formato è quel meccanismo che associa ad ogni dato che viaggia tra client e server un metadato (dato che descrive un altro dato) che rappresenta la codifica del formato del file che si sta scambiando. Questo metadato è alla base del protocollo HTTP.

Come avviene lo scambio di informazioni tra client e server? Una descrizione di massima è la seguente:

- il client fa richiesta al server corrispondente chiedendo l'informazione e comunicando i formati che è in grado di gestire
- il server prende l'informazione che gli è stata richiesta (in accordo con le indicazioni ricevute → HTTP) e la spedisce indietro al client.

Il protocollo HTTP arricchisce questo schema: il protocollo spedisce al server anche tutti i formati che il client sa decodificare. Il server a questo punto ha anche l'informazione sui formati che il client sa gestire e se può risponde nel formato che il client conosce (la risposta è il contenuto richiesto + formato del contenuto richiesto).
(Esempio: clicco su un ancora ed ottengo un popup contenente la frase "stai cercando di aprire la frase MimeType: Mpeg4".
Vuol dire che il mio browser non sa aprire il file di tipo mp4 e mi da due possibilità: scaricarlo oppure indicare con quale applicazione aprirlo, eventualmente da scaricare anch'essa.
Nel caso di aggiunta di un nuovo standard realizzato da me dovrei mettere a disposizione un traduttore per decodificare il nuovo tipo di dato che è stato

appena inserito insieme ai tipi di dato preesistenti senza che io debba buttare l'intero browser (modifico solo una sotto-parte browser stesso, aggiornando un componente).

Un nuovo standard si può diffondere facilmente senza che si debba aggiornare radicalmente tutto il web.
Questo avviene proprio grazie alla negoziazione del formato, la vera novità del W3 che permise di avere un successo così grande.

Insieme di protocolli

URL

Gli indirizzi si esprimono tramite **Uniform Resource Locator** (**URL**).

L'URL è il riferimento alla posizione di un documento all'interno del W3 ed il suo formato è indipendente dalla natura del documento; è l'identificatore univoco di ogni risorsa disponibile sul web.
La struttura è la seguente:

<protocollo>://<nome_dominio><path><nome_file>

http://www.ing.univpm.it/guida/orari.html

dove:

- **protocollo** è inteso come protocollo di comunicazione con il quale io accedo alla risorsa (se devo accedere ad una pagina web il protocollo sarà http, mentre per la posta elettronica può essere usato mailto).
- **nome del dominio** è il riferimento al server in cui risiede la risorsa che sto richiedendo (non è necessariamente una sola macchina fisica ma può essere un'unica entità virtuale che contiene molteplici macchine fisiche distribuite).
- **path** e **nome file**, cioè una volta che ho individuato la macchina (dominio) posso utilizzare il percorso che va dal nome del dominio (cartella predefinita del server web) al punto in cui si trova il file che si vuole utilizzare (che rimane comunque una pagina web).

Alcune di queste parti possono essere omesse. Posso omettere il protocollo (sottointeso http solitamente), oppure il nome del file (in questo caso dipende

da come il webserver viene configurato, ma solitamente si aggiunge una parte standard per reindirizzare ad un contenuto fisso (e.g. univpm.it/index.php)).

HTTP

HyperText Transfer Protocol *→ Protocollo di comunicazione specifico per il W3.*

È tra tutti i protocolli di comunicazione Internet (FTP, Telnet, SSH, ...) quello **specifico** per il W3. Ha i vantaggi di essere *veloce*, *universale* ed *estendibile*.

Implementa la negoziazione tra Client e Server del formato del documento da trasferire.

I messaggi scambiati tra Client e Server, che sono sostanzialmente stream di byte, sono strutturati in 3 blocchi:

- Header: contiene una serie di informazioni di controllo sulla natura del dato che viene scambiato, fra cui il formato del documento.
- Payload: contiene il dato che il messaggio porta con sè.
- Footer: contiente delle infomazioni che consentono, a chi riceve il messaggio, di comprendere se il messaggio è stato correttamente ricevuto (cioè se ad esempio non c'è stato un degrado durante la trasmissione).

Usa un meccanismo di interazione simile al protocollo *Telnet* (HTTP però è senza autenticazione) che sostanzialmente collega il browser come se fosse un terminale remoto del server, cioè il browser si identifica nel server e poi inizia ad inviargli una serie di comandi che sono quelli che consentono il trasferimento di dati fra l'uno e l'altro.

HTML

HyperText Markup Language *→ linguaggio dominante nei documenti W3, ogni client è in grado di gestirlo.*

Dato un testo, ci consente di definire quali sono le componenti di questo testo (titoli, paragrafi, contenuti) in modo tale che, a partire da questa struttura, il client web sia poi in grado di visualizzare il contenuto del documento stesso, in maniera tale che questa visualizzazione sia (in?)dipendendente dalle caratteristiche della macchina sulla quale il client web è implementato.

Storicamente, è una variante dell'SGML (Standard Generalized Markup Language) che consente la strutturazione del testo e l’inserimento di link.

HTML **descrive la struttura** dei documenti, non il loro formato di visualizzazione. Il formato di visualizzazione viene deciso dal browser nel momento in cui il documento viene visualizzato su una particolare finestra.

Consente di ottenere una visualizzazione ottimale dei dati su piattaforme differenti che usano differenti font.

Se la pagina non è codificata in HTML non viene correttamente visualizzata dal browser (anche in funzione della dimensione della finestra, meccanismo adattivo per la visualizzazione della pagina rispetto all'effettiva dimensione della finestra. Per fare questo devo essere capace di identificare ogni parte del mio documento). Non descrive il formato di visualizzazione, che è compito dei CSS.

Insieme di applicazioni

Il web supporta tantissime applicazioni di ogni tipo. Fondamentalmente abbiamo 3 classi:

- **Web Servers** (Apache, IIS, Nginx, ...)
- **Web Browser** [*client*] (Chrome, Firefox, Safari, Opera, ...)
- **Tools**, tra cui troviamo i motori di ricerca, gli strumenti di supporto per lo sviluppo di applicazioni web(HTML Validators, link checker, CSS Validators, ...)

Tutto il resto sono pagine web elaborate in maniera sofisticata. E.g. per Google Docs ho un browser (client) ed un server per elaborare il documento.

Ragnatela di informazioni

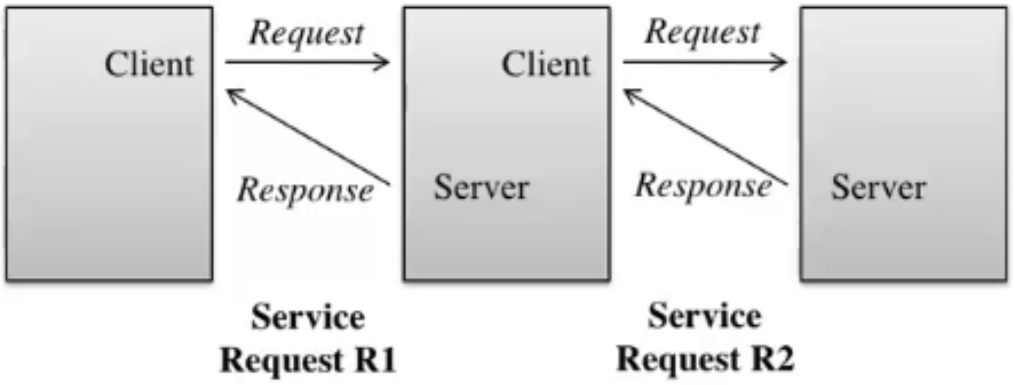
Modello Client/Server nel W3

È un modello di interazione tra una coppia di processi elaborativi che interagiscono attraverso (ad esempio) una connessione di rete, per sviluppare un processo di elaborazione condiviso in un'ottica distribuita.

- Processo: esecuzione temporale di una componente del programma
- Programma: concetto temporale. Composto da più processi.

Un processo client è un processo che richiede informazioni ad un altro processo, mentre il server è quello a cui la richiesta viene inoltrata e provvede a fornire l'informazione.
Nell'ambito dell'interazione, cioè il meccanismo di richiesta/risposta, i ruoli

rimangono costanti, quindi il client rimane client e il server rimane server (non succede che il client fa una richiesta al server e questo prima di rispondere fa anche una richiesta al client che l'ha appena interrogato).
Invece è possibile che i processi assumano ruoli diversi in funzione delle diverse interazioni.



Vediamo il client A che effettua una richiesta al server B. Può succedere che B per rispondere alla richiesta di A deve andare ad interrogare anche C. In questo secondo contesto B diventa client e C server, quindi B è sia Client che Server ma in contesti diversi (ed è quindi ammesso). A questo punto B ottiene tutte le informazioni che gli servivano, risponde ad A e l'interazione complessiva viene completata.
I ruoli dei processi che interagiscono sono dinamici (possono cambiare nel tempo) e dipendono dal tipo di interazione che avviene tra i processi; i ruoli sono definiti all'interno della coppia dei processi.

Il modello C/S presenta 5 caratteristiche:

1. **I Client ed i Server sono moduli funzionali con interfacce ben definite.**

   Il modello si basa sul concetto di interfaccia ed è il concetto per cui un processo espone al mondo esterno una serie di funzionalità che sono quelle con il quale il mondo esterno può interagire con quel programma (l'implementazione delle funzioni elaborative interne è nascosta, abbiamo solo l'accesso ai metodi esposti → anche detto *incapsulamento*).

   Un server web e un client web sono sostanzialmente delle black box che possono essere utilizzati a partire dalle funzioni che espongono.
   Come queste funzioni vengono implementate non è visibile all'utilizzatore e questo consente a chi gestisce il web server di poter modificare

l’implementazione interna senza che il client se ne accorga, dato che le interfacce rimangono invariate
.
Queste funzioni di interfaccia possono essere implementate da:
– sistemi software.
– strutture hardware (più veloce rispetto alla soluzione software).
– sistemi misti/ibridi (in parte software e in parte hardware).
Ho massima flessibilità sotto questo punto di vista.
Infine,
i Client e/o i Server possono risiedere o su computer dedicati o su computer *general purpose*.

2. **L'interazione Client/Server si attiva tra due moduli funzionali quando uno di essi (Client) richiede un servizio e un altro (Server) decide di rispondere alla richiesta.**

   Non esistono macchine intrinsecamente client ed intrinsecamente server.
   Ad un certo punto un processo A effettua una richiesta ad un processo B.
   Da quel momento e fino alla fine dello scambio messaggi A è client e B è server.
   I ruoli dei processi rimangono costanti durante una stessa interazione, mentre possono variare per interazioni diverse.

3. I **moduli funzionali possono risiedere sullo stesso calcolatore o su macchine diverse, interconnesse attraverso una rete.**

   I moduli client/server sono dei software che possiamo installare sulla stessa macchina o su macchine diverse. Ad esempio, ogni calcolatore che noi usiamo ha una coppia di client/server che viene gestito dal SO.
   La GUI è una componente che si configura come la componente client di un’altra componente che si presenta come il gestore dei comandi (server) del SO che prende il comando e lo trasforma (ogni volta che clicco un’icona genera un comando ed invia il comando al server/gestore dei comandi che elabora ed esegue quel comando).
   Noi studieremo interazioni C/S tra moduli che risiedono su macchine diverse.

4. **Lo scambio di informazioni tra i moduli avviene attraverso messaggi**.
   Un messaggio è una "
   capsula" (ovvero una struttura dati) all’interno della quale vengono inseriti i

dati oggetto della comunicazione e che viene inviato al modulo corrispondente. Questi messaggi sono incapsulati in una struttura che è sempre la stessa, indipendentemente dal suo contenuto (se viaggia una immagine piuttosto che un video non importa perché dall'esterno vedo sempre è solo il messaggio). Il concetto di messaggio rende il modello estremamente flessibile.
Esempio: posta pneumatica. Prendo il contenuto che devo inviare qualunque esso sia, lo inserisco nel tubo e lo invio. Tutti i tubi hanno la stessa destinazione. Una volta che ho inserito il tubo, questo viene inviato verso la destinazione finale. Con il tubo posso spedire qualunque tipo di contenuto attraverso lo stesso sistema di scambio di informazioni.

5. **I messaggi scambiati sono tipicamente interattivi e generano l'attivazione di processi all'interno dei moduli**.

   Il trasferimento di un messaggio dal client al server innesca in chi lo riceve un processo di elaborazione che serve per gestire il contenuto del messaggio, e un analogo processo si attiva nel momento in cui il server invia la sua risposta al client.
   Ad ogni capsula è associato un processo elaborativo che serve per manipolare il contenuto della capsula e che viene attivato nel momento in cui il messaggio stesso arriva nel dispositivo di destinazione.

   Questo meccanismo di interattività può avvenire anche in modalità differita, ovvero
   in un primo momento il messaggio viene ricevuto e in una fase successiva viene processato.
   Il protocollo di comunicazione che viene utilizzato nell'ambito di questi sistemi è un protocollo che non trasferisce immediatamente il messaggio quando viene ricevuto, ma questi viene temporaneamente memorizzato in una coda nel quale vengono sequenzializzati tutti i messaggi ricevuti dal server intermedio, e poi in un momento successivo da questa coda viene recuperato il messaggio desiderato per essere inviato all'elemento successivo della catena di interconnessione fra noi ed il nostro corrispondente. Il ritardo è dovuto alla scelta di come sono organizzate le diverse fasi dell'elaborazione; in ogni caso, quando il messaggio arriva innesca l'elaborazione (rimane l'interattività).

C/S e distributed computing

La radice dei concetti che stiamo esprimendo è l'insieme dei **modelli computazionali**, cioè i modelli che definiscono come un certo programma viene eseguito all'interno di un sistema informatico.

I modelli computazionali si dividono in 2 grandi famiglie:

- modelli terminali host : il programma viene mandato in esecuzione in un host (macchina di potenza di calcolo notevole localizzata e associata a un insieme di terminali attraverso i quali gli utenti possono accedere) e quindi tutta l'elaborazione viene concentrata in un unico sistema di elaborazione.
- modelli di elaborazione distribuita : l'elaborazione viene condivisa tra diversi sistemi interconnessi che concorronno insieme alla produzione del risultato dell'elaborazione stessa.
  A loro volta sono divisi in:
  1. P2P (peer-to-peer) : modelli client/server che non hanno una funzionalità definita in modo rigido e gerarchico, possono al tempo stesso rispondere a richieste o fare richieste. Non hanno forte caratterizzazione del ruolo che svolgono
  2. C/S. (client/server)
  3. modelli per il trasferimento di file : l'elaborazione avviene fra processi distribuiti, i quali utilizzano dei file per scambiarsi l'informazione.