

*Concetti base*

## Impianti Informatici

 POLITECNICO DI MILANO



Web application



➤
La diffusione del World Wide Web
2

---

Supporto ai ricercatori


- Organizzazione documentazione
- Condivisione informazioni

Scambio di informazioni di qualsiasi natura

- Chat
- Forum

Intranet

- Stessi standard usati in Internet



Impianti Informatici
POLITECNICO DI MILANO

Il Web ha le sue origini negli anni 90, quando si è definito come sistema multimediale ad ipertesto,

1. nato essenzialmente come supporto ai ricercatori sparsi in tutto il mondo,
2. per permettere di organizzare e
3. condividere la documentazione prodotta. Presto si sono apprese le notevoli potenzialità di comunicazione,
4. e si è cominciato ad utilizzare il Web anche x attività esterne a quelle della ricerca, che si è dimostrato anche un mezzo di aggregazione
5. Sono nate le chat, una forma di comunicazione scritta anziché parlata; sono nati
6. i gruppi di discussione, i forum, una sorta di comunicazione asincrona centrata su uno specifico argomento.
7. I concetti base del Web sono diventati anche lo standard all'interno delle singole aziende, come l'uso del protocollo HTTP come protocollo applicativo e, a livello inferiore, dello stack protocollare TCP/IP.
8. Si sono diffuse quindi le Intranet, reti aziendali immediatamente integrabili con Internet, dato che seguono gli stessi standard protocollari.

**Applicazioni Web** 3

Una qualsiasi applicazione client/server che:

- Usa HTTP come protocollo di comunicazione
  - Effettua richieste Http
  - Riceve risposte Http
  - Usa i browser standard come client (ma non necessariamente)

Solitamente tale applicazione funziona da "interfaccia" per l'accesso ad una base di dati

```
graph LR; Client[Client] -- "HTTP request" --> Server[Server]; Server -- "HTTP reply" --> Client;
```

Il diagramma illustra l'interazione tra un Client e un Server. Il Client invia una richiesta HTTP (HTTP request) al Server, e il Server risponde con una risposta HTTP (HTTP reply).

Impianti Informatici POLITECNICO DI MILANO

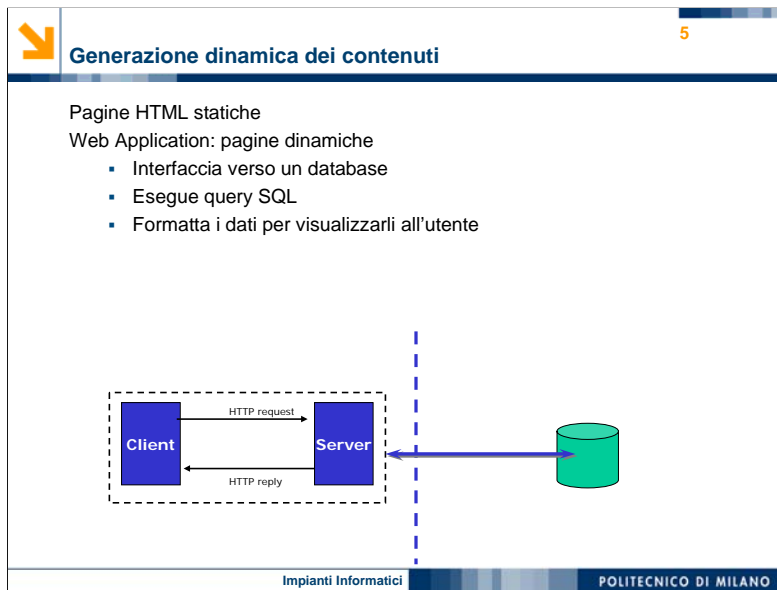
La diffusione del Web ha portato alla nascita di una nuova categoria di applicazioni, denominata appunto Web Application.

1. Una Web application è una qualsiasi applicazione basata sul paradigma client/server, e che utilizzi
2. il protocollo applicativo HTTP per la comunicazione tra client e server. Questo significa che i due componenti si scambiano dati effettuando
3. richieste HTTP, quindi usando metodi come GET o POST, e ricevendo
4. delle risposte HTTP. Il client di una Web Application
5. è quindi generalmente un browser, dato che utilizza nativamente il protocollo HTTP per la comunicazione. In ogni caso è possibile che venga implementato un qualsiasi altro client, che si appoggi però allo stesso standard.
6. Uno dei classici impieghi di una web application è l'accesso ad una base di dati, operando così da interfaccia per l'utente.

Applicazione standard vs Web Application	
Applicazione Standard	Web Application
<ul style="list-style-type: none"> <li>▪ Protocollo proprietario</li> <li>▪ Protocollo non fissato</li> <li>▪ Client ad-hoc <ul style="list-style-type: none"> <li>• Potenzialità di interazione</li> <li>• Potenzialità di visualizzazione</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>▪ HTTP</li> <li>▪ Stack TCP/IP</li> <li>▪ Browser <ul style="list-style-type: none"> <li>• Interazione limitata</li> <li>• Visualizzazione semplice</li> </ul> </li> </ul>
Impianti Informatici	POLITECNICO DI MILANO

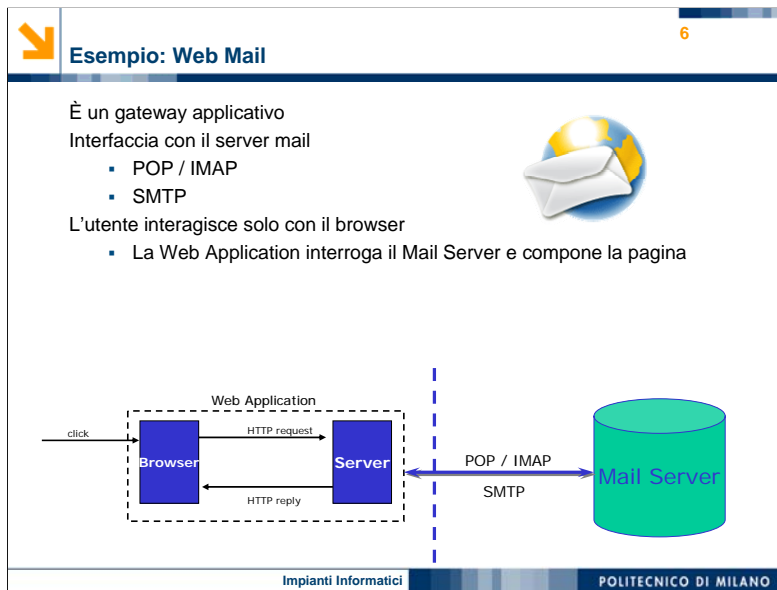
Il principale elemento distintivo tra

1. una applicazione standard di tipo client/server
2. ed una web application, è
3. l'uso del protocollo HTTP per la comunicazione di alto livello, anziché di un protocollo proprietario. Questo ha conseguenze anche nella scelta dei protocolli più di basso livello,
4. Appoggiandosi allo stack TCP/IP. È possibile, come spesso avviene, che anche una applicazione NON web usi lo stack tcp/ip, interagendo con esso, ad esempio, direttamente tramite la socket interface e quindi con l'uso di primitive per l'apertura della connessione Tcp, lo scambio dei dati e la chiusura della stessa.
5. L'uso del protocollo HTTP consente di non dover implementare un apposito client, ma è ipotizzabile l'uso di un comune browser per la navigazione sul Web.
6. Lo svantaggio nell'uso dei browser è la scarsa capacità di interazione che l'utente ha a disposizione; può semplicemente cliccare dei link e inserire dei dati all'interno di form.
7. Anche le capacità di visualizzazione sono tutto sommato limitate; questo complica la realizzazione di alcune operazioni.




Contemporaneamente alla definizione del Web, è nato anche il linguaggio HTML per consentire la visualizzazione delle informazioni.

1. Le pagine HTML sono hanno però un contenuto altamente statico. Nel caso in cui si desiderino modificare i dati contenuti in una pagina, occorre agire direttamente sul suo codice, operazione non sempre agevole.
2. Le Web Application consentono, favorite da una struttura a 3 livelli, che vedremo in dettaglio successivamente, di realizzare pagine dal contenuto dinamico.
3. Esse operano infatti da interfaccia tra l'utente ed il database. In questo modo l'utente può accedere ad una base di dati senza
4. dover conoscere linguaggi di interrogazione, come SQL e in modo trasparente alla struttura del database.
5. È poi sempre la web application che formatta il risultato dell'interrogazione al database presentandolo in maniera intelligibile



Una tra le più classiche Web Application è la Web Mail.

1. Tale applicazione era già stata nominata parlando di gateway, in quanto funge da interfaccia tra protocolli differenti, quelli per il Web e quelli per le e-mail.
2. Anche in questo caso le web application opera da intermediario tra l'utente e una base di dati:
3. in questo caso non si tratta di un comune database, ma di una casella di posta elettronica.
4. Per accedervi occorre utilizzare il protocollo POP o IMAP per l'operazione di lettura dei messaggi,
5. ed il protocollo SMPT per l'invio. L'utente non deve però conoscerne le specifiche,
6. ma interagisce con il proprio browser, cliccando sui link o inserendo dati in una form
7. Il browser, cioè il client della web application, invio al lato server una richiesta http, contenente, ad esempio, la richiesta di accedere ad un dato messaggio della casella di posta.
8. La web application dialoga con il server mail tramite il protocollo POP, compone la pagina contenente il messaggio di posta e
9. La invia al browser, dove viene interpretata e visualizzata all'utente

 **Esempio: mappa stradale** 7

Interazione con una base di dati


- Il DB contiene i possibili percorsi, pesati ad esempio in base a traffico, tipo di strada,...
- Calcola il percorso ottimale (il più breve, il più semplice,...)
  - Algoritmo complesso
- Genera le istruzioni necessarie per seguire l'itinerario
- Formatta una pagina contenente le indicazioni da seguire e la mappa stradale

Impianti Informatici

POLITECNICO DI MILANO

Un altro esempio di applicazione Web è rappresentato

1. dai siti che consentono di visualizzare itinerari stradali. La particolarità di questa tipologia di applicazioni è la complessità presente nella logica del lato server.
2. Infatti il database contiene, ad esempio, il grafo dei possibili percorsi, ciascuno pesato in base a parametri, come tipologia di strada o traffico medio.
3. Di fronte ad una richiesta dell'utente, l'applicazione deve calcolare il percorso ottimale, che può essere quello più breve, oppure quello meno complicato da seguire.
4. Per farlo la logica applicativa risultante è particolarmente complessa, e questa complessità risulta totalmente trasparente all'utente.
5. Il compito dell'applicazione non è finito: dopo aver scelto l'itinerario, occorre, inoltre, generare le indicazioni, che poi vengono presentate all'utente..
6. in una pagina web opportunamente formattata, unitamente alla relativa cartina geografica

 **Applicazioni Web: peculiarità** 8

Di facile utilizzo

- si utilizza un client usato comunemente per la navigazione di pagine web

Integrazione immediata con Internet

- segue lo standard TCP/IP e utilizza l'HTTP come protocollo applicativo

Indipendente dalla piattaforma (TCP/IP)

Interoperabilità tra hw/sw di costruttori vari

Economica

- bassi costi di produzione del relativo sw applicativo

Architettura altamente scalabile

- consente di scrivere anche applicazioni distribuite


Modulare (divisa in livelli logici)

Impianti Informatici POLITECNICO DI MILANO

Una delle caratteristiche principali di una applicazione Web è

1. il fatto di potersi interfacciare all'utente tramite un qualsiasi browser, il client comunemente usato per la navigazione sul Web, dal funzionamento molto semplice ed intuitivo
2. Inoltre, dato che usa il protocollo HTTP a livello applicativo, e segue lo standard TCP/IP, è immediata l'integrazione con Internet
3. Questi standard favoriscono anche l'uso della stessa web application su piattaforme eterogenee,
4. consentendo l'utilizzo di software e hardware multivendor,
5. e con costi di produzione, dell'applicazione stessa, relativamente bassi.
6. L'architettura risulta inoltre fortemente scalabile, con
7. possibilità di scrivere applicazioni distribuite.
8. L'ultimo aspetto, che verrà approfondito in seguito, è la modularità delle web application





Limitazioni del protocollo HTTP

9


Stateless

Transazioni difficili

- Carrello on-line
  - Log-in
  - Selezione prodotti
  - Pagamento
  - Log-off

Possibili soluzioni:

- Cookie
- URL rewriting

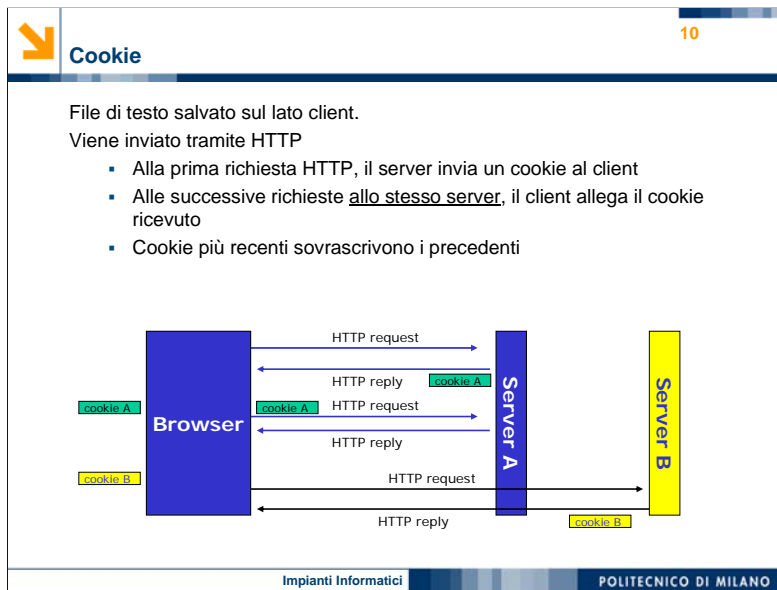


Impianti Informatici

POLITECNICO DI MILANO

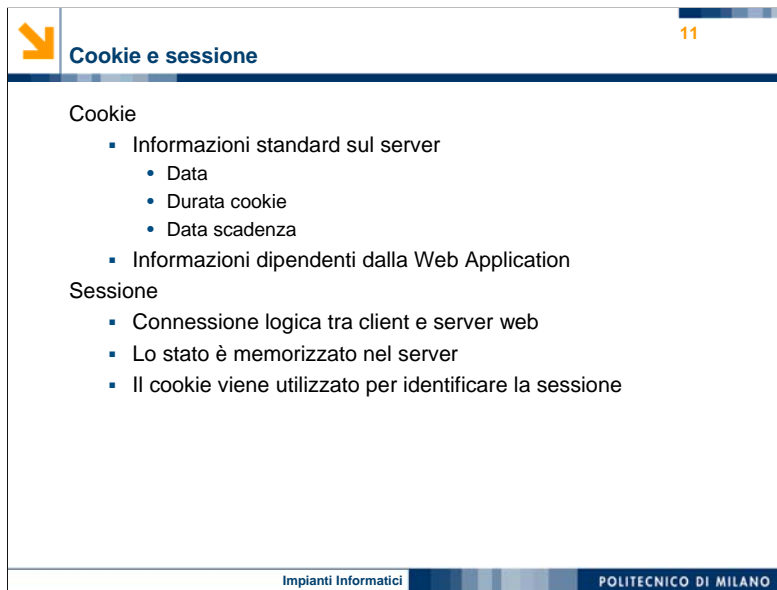
Parlando del protocollo HTTP, sia nella versione 1.0 che nella 1.1, è emersa una sua limitazione fondamentale,

1. ovvero il fatto di essere Stateless. Quindi, ogni richiesta HTTP è indipendente dalle precedenti, in quanto non viene mantenuto uno stato. Lo svantaggio principale
2. è l'impossibilità di eseguire operazioni transazionali.
3. Il classico esempio è quello della gestione di un carrello on-line:
4. l'utente effettua il login,
5. seleziona uno o più prodotti da aggiungere al carrello,
6. quindi inserisce i dati per il pagamento e
7. si disconnette dal sistema. Tutte queste azioni fanno parte di un'unica transazione, che il semplice uso del protocollo HTTP non può gestire.
8. Due possibili tecniche che ovviano a questa limitazione sono:
9. l'uso dei cookie e
10. l'URL rewriting



I cookie sono un sistema specificato nel protocollo http,

1. essi forniscono la possibilità ad una applicazione web di poter salvare delle informazioni sul client nel formato di un file di testo.
2. Quindi un cookie è un file di testo inviato tramite l'http, che ha però delle limitazioni, ad esempio possono essere salvati dal server solo in una determinata cartella del client.
3. La prima volta che il client fa una richiesta http ad un server,
4. quest'ultimo invia un cookie, che viene memorizzato nel client
5. Durante le eventuali connessioni successive allo stesso server, il client invia il cookie ricevuto in precedenza.
6. Da sottolineare il fatto che il client invia i cookie solo a chi glieli aveva mandati in precedenza. In questo modo, con una richiesta HTTP verso il server B, non viene inviato alcun cookie.
7. se un server dovesse mandare un ulteriore cookie più recente, questo va a sovrascrivere il precedente



**Cookie e sessione**

11

Cookie


- Informazioni standard sul server
  - Data
  - Durata cookie
  - Data scadenza
- Informazioni dipendenti dalla Web Application

Sessione

- Connessione logica tra client e server web
- Lo stato è memorizzato nel server
- Il cookie viene utilizzato per identificare la sessione

Impianti Informatici POLITECNICO DI MILANO

1. Un cookie contiene informazioni standard,
2. ad esempio sul server che l'ha inviato,
3. la data,
4. la durata del cookie,
5. la scadenza. Trascorso il tempo della durata e raggiunta la data di scadenza, il client cancella il cookie in quanto non lo ritiene piu valido.
6. Contiene inoltre altre informazioni che dipendono da cosa abbia voluto inserirci il server..
7. Le applicazioni web introducono il concetto di sessione:
8. cioè una connessione logica continuativa tra un client ed un server web. Si può identificare con la navigazione e tipicamente hanno una durata massima
9. Ogni sessione deve utilizzare una particolare entità, l'oggetto di sessione, cioè una sorta di variabile di stato che si trova sul server
10. ed un cookie, contenente l'identificativo della sessione, in modo che una applicazione web possa determinarla esattamente.

 URL rewriting12

URL dinamici

- Con ID della sessione

Vengono riscritti gli URL all'interno delle pagine da inviare al browser


Ogni richiesta HTTP include l'ID della sessione

- Il server separa ID da URL effettivo

Impianti InformaticiPOLITECNICO DI MILANO

Non sempre i browser consentono di memorizzare i cookie inviati dal server. Una tecnica alternativa è denominata URL rewriting.

1. Consiste nell'uso di URL dinamici,
2. che contengano all'interno un identificativo della sessione.
3. In pratica, dal lato server, durante la generazione della pagina da inviare al browser, vengono riscritti tutti gli indirizzi dei collegamenti presenti, aggiungendo l'id univoco della sessione corrente.
4. Quando l'utente seleziona un link, invia implicitamente anche il valore della sessione, perché contenuto nell'URL della richiesta HTTP
5. Il server effettua quindi l'operazione contraria alla precedente, estraendo dall'URL la parte riservata all'ID della sessione, e quella indicante effettivamente l'indirizzo della risorsa.

 **Thin Client**

13

Il client svolge funzioni limitate  
La logica è tutta sul lato server  
Il client si occupa della visualizzazione e dell'interfaccia utente  
E il Browser Web?

- Non troppo "leggero", dato l'uso di plug-in (ActiveX, JavaScript,...)

Sovraccaricano il lato server


- Occorre connettersi al server per qualsiasi operazione

Impianti Informatici

POLITECNICO DI MILANO

In un'architettura client/server come quella delle web application, è importante distinguere tra thin client e fat client, a seconda delle funzionalità che sono presenti dal lato client.

1. Nel caso di thin client, le funzioni presenti sono molto limitate, e
2. La maggior parte della logica applicativa risiede e viene eseguita da uno o più server
3. Il client è limitato a visualizzare i dati provenienti dal server e interagire con l'utente. L'interazione è generalmente piuttosto limitata, data le limitate capacità dei client thin.
4. Ad esempio il browser web può considerato un thin client,
5. anche se l'integrazione di plug-in, come ad esempio i controlli ActiveX, o codici Javascript, hanno spostato una piccola parte di logica anche sul lato client, cosicché non risulta più così leggero.
6. Dato che la computazione avviene quasi esclusivamente sul server, e considerando che generalmente ad un server corrispondono numerosi client, esso può rappresentare un possibile collo di bottiglia dell'architettura.
7. Anche semplici operazioni, quali la validazione dell'input, richiedono l'interazione con il server

 **Fat (Thick) Client**

14

Sul client vengono eseguite funzioni degli applicativi

- Il client sw è una parte dell'applicazione che va installata su ogni macchina

Più difficili da mantenere

- Molti client
- Cambiamenti nel server hanno effetti sul client

È possibile realizzare interfacce grafiche complesse

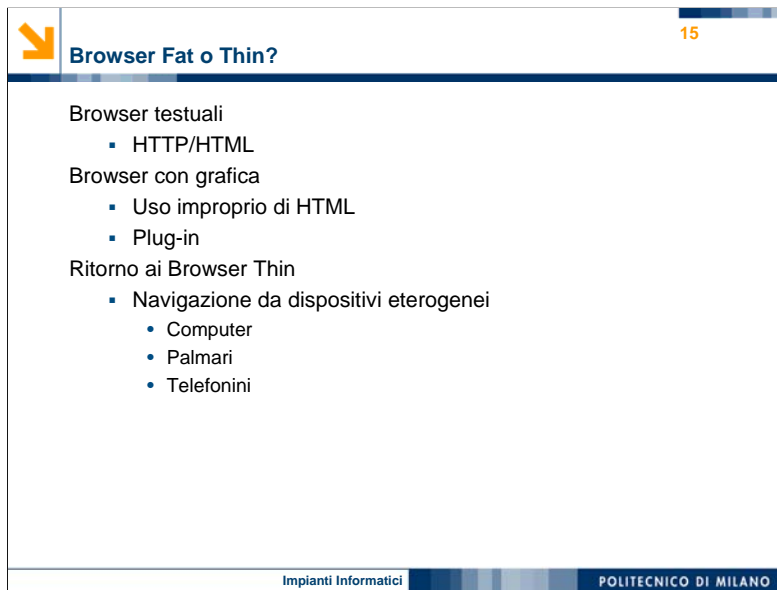
Non richiede un'interazione continua col server

- Riduce il carico sul lato server
- Aumenta il carico sul client

Impianti Informatici

POLITECNICO DI MILANO

1. Su un client di tipo FAT, invece, viene demandata parte della logica
2. Il client è meno disaccoppiato dal server e costituisce una parte dell'applicazione da installare su ogni macchina
3. Risulta così più difficile da mantenere,
4. perché eventuali modifiche vanno apportate su tutti i client, spesso anche
5. cambiamenti nel codice del server.
6. Dato che implementa una parte di logica, il client può proporre interfacce grafiche più complesse, con maggiori possibilità di interazione dell'utente e capacità grafiche più evolute rispetto al client thin.
7. Inoltre, il fatto di non dover continuamente interagire col server,
8. ne riduce il carico computazionale,
9. pur appesantendolo sul lato client.



**Browser Fat o Thin?** 15

- Browser testuali
  - HTTP/HTML
- Browser con grafica
  - Uso improprio di HTML
  - Plug-in
- Ritorno ai Browser Thin
  - Navigazione da dispositivi eterogenei
    - Computer
    - Palmari
    - Telefonini

Impianti Informatici POLITECNICO DI MILANO

La scelta tra client fat e thin merita uno speciale trattamento nel caso in cui il client sia un browser web.

1. I primi browser erano browser puramente testuali, con minimi accenni di grafica.
2. Le prime evoluzioni riguardano l'uso dei tag offerti dal linguaggio html, designati appunto per indicare al client come impaginare il documento. Ogni browser interpreta il linguaggio e lo visualizza all'utente, ad esempio formattando il titolo il un dato modo e il corpo in un altro..
3. La crescita della capacità di elaborazione dei computer e l'esigenza di catturare l'attenzione mediante artifici grafici, ha portato allo sviluppo di browser meno leggeri.
4. Gli effetti grafici venivano inizialmente prodotti mediante un uso improprio di tabelle e frame HTML.
5. Poi non sono stati integrati all'interno dei browser dei plug-in, come ActiveX o JavaScript. Ogni browser, per visualizzare correttamente le informazioni, necessita di implementare questi plug-in. Sono nati così i primi problemi di compatibilità tra browser, come anche oggi si può notare navigando alcune tipologie di siti. Viene così meno l'obiettivo per cui è nato il Web, cioè consentire a tutti di visualizzare le informazioni messe a disposizione.
6. Per questo la tendenza è quella ad un ritorno verso browser di tipo thin, oltretutto considerando
7. la diffusione di dispositivi molto eterogenei tra loro per la navigazione.
8. Oltre ai computer, infatti,
9. Vengono usati sempre più spesso sia palmari
10. che telefonini. Pagine che utilizzano particolari plug-in non possono essere visualizzati da questi terminali, ed è anche poco proponibile realizzare plug in per ogni tipologia di browser presente sul mercato.

*Architetture a più livelli*

## Impianti Informatici

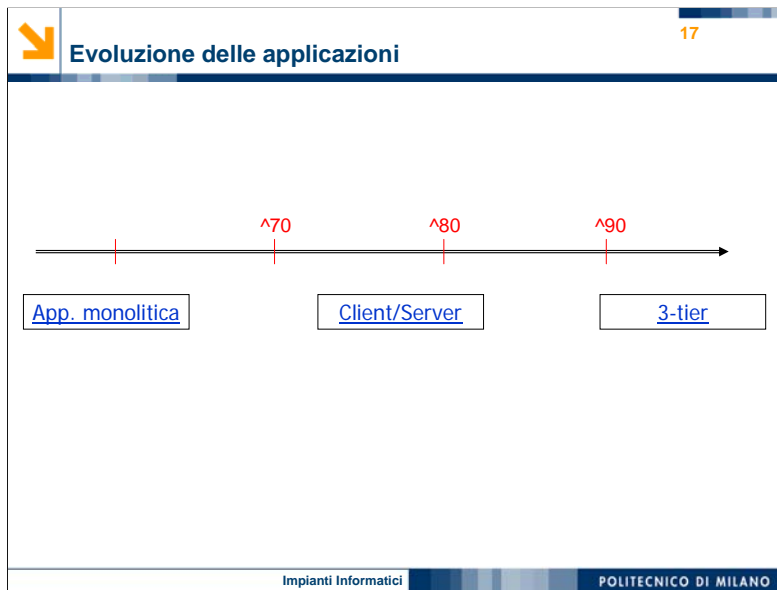
 POLITECNICO DI MILANO



Web application








I paradigmi architetturali su cui si basano le applicazioni hanno subito

1. una evidente evoluzione nel corso del tempo.
2. Le prime applicazioni erano di tipo monolitico, ovvero blocchi di codice indivisibile.
3. Dalla fine degli anni 70, alla metà degli anni 80, si è passati ad una divisione logica dell'applicativo in un modulo client ed un modulo server
4. Infine, all'inizio degli anni 90, sono state introdotte le architetture a tre livelli



Applicazioni Monolitiche

18


Popolare con i mainframe

- Adatti per processi stand-alone

Pezzi di codice indivisibile

Controlla l'intera applicazione

- Logica applicativa
- Gestione e memorizzazione dei dati
- Interfaccia utente




Impianti Informatici

POLITECNICO DI MILANO

L'architettura monolitica è il più antico paradigma di programmazione,

1. molto popolare ai tempi dei mainframe, adatti ad eseguire pochi
2. processi stand-alone, anziché diversi processi comunicanti; tutto ciò quando non esistevano ancora i database.
3. Un'applicazione monolitica è un pezzo di codice indivisibile, in grado di controllare l'intera applicazione,
4. dalla logica applicativa, alla
5. gestione e memorizzazione dei dati, fino
6. all'interfaccia utente.

 Applicazioni Client/Server

19

Passaggio da mainframe a server, pc, workstation:

- Più economici
- Più piccoli

Diffusione Database RDBMS

Architettura monolitica non flessibile

Paradigma client/server


- Server (backend):
  - Gestione di database
  - Elaborazione dei dati
- Client (frontend):
  - Interfaccia utente

Impianti Informatici

POLITECNICO DI MILANO

Dalla fine degli anni '70, alla metà degli anni '80, si è potuta osservare

- la diffusione tanto di server, quanto di pc e workstation, che, rispetto ai mainframe, risultavano
- notevolmente più economici e
- dalle dimensioni inferiori. Più o meno contemporaneamente,
- hanno conosciuto altrettanta notorietà i database di tipo relazionale. Tutto ciò, assieme alla necessità di
- una architettura più flessibile rispetto a quella monolitica, ha favorito lo sviluppo delle
- applicazioni di tipo client/server, suddivise perciò in due componenti logiche con funzionalità diverse:
- il server, detto anche componente di backend, con funzioni come
- la gestione, più o meno complessa, di database,
- e compiti inerenti la manipolazione dei dati;
- il client, anche denominato frontend, essenzialmente
- delegato alla gestione dell'interfaccia utente.



Client/Server: pro e contro

20

Vantaggi:

- Maggiore scalabilità
- Carico computazionale distribuito sui client
- Sviluppo più veloce di applicazioni che accedono agli stessi dati

Svantaggi:

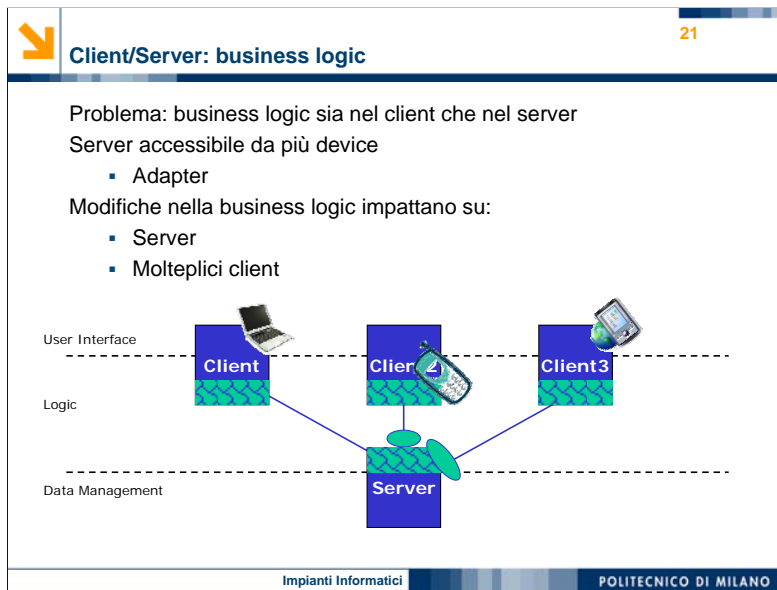
- Traffico di messaggi intenso
- Logica di business inclusa sia nel frontend che nel backend
  - Client e server dipendenti l'uno dall'altro

Impianti Informatici

POLITECNICO DI MILANO

Questo tipo di architettura,

1. confrontata con quella monolita, risulta
2. notevolmente più scalabile, data la suddivisione del
3. carico computazionale tra diversi client. Una eventuale necessità di maggior potenza di elaborazione, può essere affrontata aggiungendo nuovi client di supporto.
4. Inoltre, applicazioni che accedono agli stessi dati possono essere sviluppate più velocemente andando a riusare parte del codice già esistente
5. Il paradigma presenta anche una serie di svantaggi:
6. C'è un aumento traffico di messaggi per la comunicazione tra client e server, il che occupa anche risorse di elaborazione. Inoltre,
7. Le pratiche di programmazione tendono ad includere parte della business logic sia nel frontend che nel backend dell'applicativo, cosicché
8. Client e server risultano strettamente dipendenti l'uno dall'altro



La principale problematica del paradigma client/server è il mancato riconoscimento dell'importanza della business logic.

1. Essa risulta confusa tra la componente client e la componente server.
2. In questo modo accade che l'accesso allo stesso server, da parte di client eterogenei,
3. può richiedere un ulteriore strato software, detto adapter.
4. Inoltre eventuali modifiche nella business logic provocano ingenti cambiamenti
5. sia dal lato server,
6. sia tra i molteplici client che utilizzano il servizio.

➔ Applicazioni 3-tier
22

Tre livelli:

- Data Layer
  - Gestione dei dati
  - DBMS
  - File XML
- Business Layer
  - Business Logic
  - Processamento dati
- Presentation
  - Interfaccia utente
  - Presentazione Dati
  - Servizi




Indipendenza dei layer

- Ogni livello ha obiettivi e vincoli propri
- Nessuna assunzione sugli altri livelli

Impianti Informatici
POLITECNICO DI MILANO

1. Le architetture di tipo 3-tier sono state introdotte all'inizio degli anni 90, come evoluzione di quelle client/server.
2. Essi sono composti da tre livelli distinti tra loro:
3. il data layer,
4. il business layer ed
5. il presentation layer
6. Il livello più basso si occupa della gestione dei dati, come database o file XML.
7. Il secondo livello riguarda la business logic ed il processamento dei dati,
8. il terzo è invece inerente l'interfaccia utente, quindi la presentazione dei dati e l'interazione con l'utente..
9. Ogni livello è indipendente dagli altri e
10. ha obiettivi e vincoli di design propri:
11. Quindi nessun livello fa assunzioni sulla struttura o implementazione degli altri


 **Web Application: Presentation Layer** 23

*Web browser*

- Visualizzare gli aspetti dell'applicazione
- Interfaccia utente
- Invia richieste HTTP e interpreta le risposte

*Web server: (?)*


- riceve le richieste HTTP inviate dal browser
- inoltra la richiesta ai livelli successivi
- impagina la risposta e la invia al browser



Impianti Informatici POLITECNICO DI MILANO

Solitamente anche le Web application si basano sulla più recente architettura a 3 livelli; ogni un raggruppamento logico delle componenti è in base alla funzionalità che essi forniscono agli utenti o agli altri livelli applicativi

1. Il livello di presentazione si riferisce a quei componenti responsabili x la creazione e gestione dell'interfaccia verso gli utenti.
2. Include le applicazioni lato client quali i web browser.
3. L'obiettivo di questo layer è quello di visualizzare in maniera differenziata, ai diversi utenti, differenti aspetti dell'applicazione, sulla base dei loro diritti di accesso, delle capacità del client che stanno utilizzando, o di altri fattori...
4. Solitamente anche i web server vengono fatti rientrare in questo layer; è tuttavia più preciso porli come interfaccia tra il presentation e l'application layer: le loro funzionalità sono infatti
5. quelle di ricevere richieste http,
6. inoltrarle ai livelli successivi,
7. impaginare la risposta in un messaggio http e inviarla al browser,
8. dove viene interpretata.

 **Web Application: Application Layer** 24


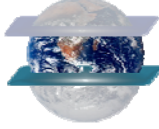
L'insieme dei programmi che eseguono le azioni richieste (J2EE, Enterprise Java Beans,...)

Gestisce

- Dati dell'applicazione
- Stato corrente
- Assicura integrità dati
- Gestisce errori o eccezioni

Esempi di operazioni:

- Query
- Elaborazione dei dati
- Report




Impianti Informatici POLITECNICO DI MILANO

L'application layer, detto anche business logic,


1. si riferisce ai componenti responsabili di implementare le funzionalità dell'applicazione.
2. Questi componenti devono gestire i dati dell'applicazione
3. ed il suo stato corrente, eseguendo le operazioni necessarie.
4. A questo livello occorre assicurare un corretto comportamento del sistema, l'integrità dei dati,
5. la gestione di eventuali errori o eccezioni.
6. Tipiche operazioni sono la
7. creazione di query sql,
8. l'analisi dei risultati, mediante algoritmi più o meno complessi,
9. e la formazione di un report da inviare al presentation layer.



 **Web Application: Data Store Layer** 25

Dati dell'applicazione

- Database
- Sistemi di altro tipo
  - Mainframe
  - Sistemi legacy
  - File XML

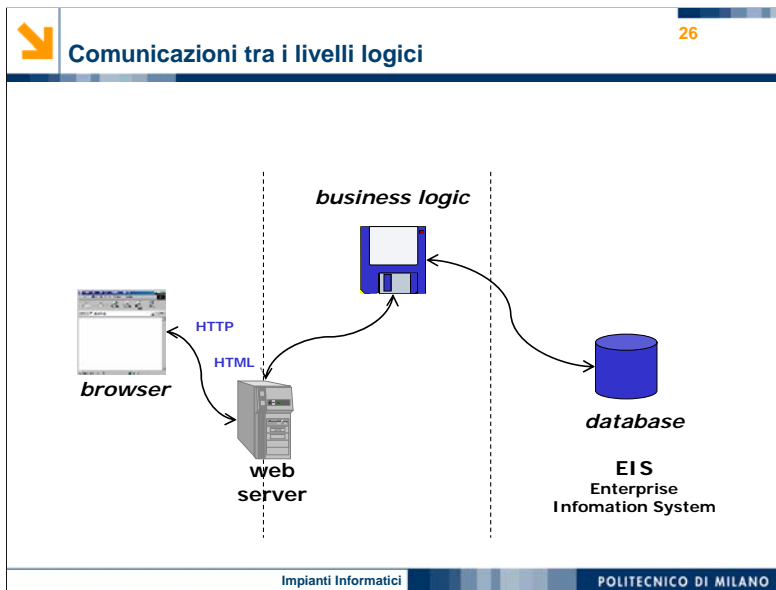


The diagram illustrates the Data Store Layer of a web application. It features a globe at the top right, representing the application's data. Below the globe, there are three icons: an XML file icon, a mainframe icon, and a server rack icon. These icons are arranged in a triangular pattern, suggesting different types of data storage or processing systems.

Impianti Informatici POLITECNICO DI MILANO

Il livello inferiore riguarda i componenti che

1. gestiscono i dati interni all'applicazione,
2. tipicamente dei Database relazionali,
3. oppure mainframe,
4. vecchi sistemi legacy o anche
5. File xml



Riassumendo, in una web application,

1. il livello di presentation include il browser web, o comunque un client che usa http come protocollo di comunicazione.
2. Tale client invia richieste http, che vengono
3. intercettate da un server web, il quale richiama l'opportuno
4. applicativo della business logic. Qui vengono eseguiti gli algoritmi necessari
5. all'estrazione dei dati dal livello data store, che vengono quindi
6. manipolati e impaginati in un opportuno formato,
7. ad esempio HTML, e
8. passati al server web che li invia tramite http al
9. browser, dove vengono interpretati e visualizzati



Applicazione come collezione di componenti

27

Non c'è comunicazione diretta tra *presentation* e *data layer*  
Applicazioni viste come collezione di componenti cooperanti.  
Ogni componente può essere contemporaneamente parte di applicazioni diverse

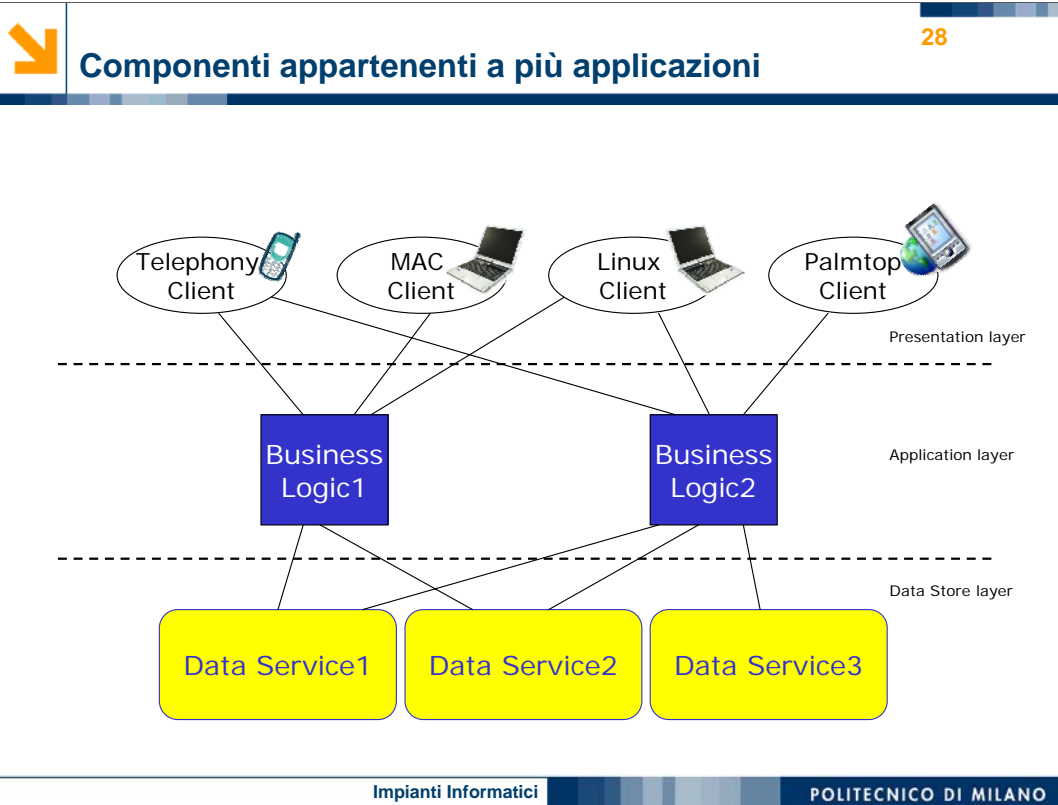


Impianti Informatici

POLITECNICO DI MILANO

In una architettura 3-tier,

1. non c'è comunicazione diretta tra il livello di presentazione e quello dati: è la business logic che filtra i passaggi di informazione nei due sensi.
2. Se i 3 livelli vengono implementati correttamente, essi operano senza assumere di essere parte di una specifica applicazione:
3. le applicazioni sono così viste come una collezione di componenti cooperanti.
4. In questo modo ogni componente può essere contemporaneamente parte di applicazioni diverse.



Viene ora mostrato, mediante uno schema, uno scenario di possibile interazione tra componenti differenti di applicazioni differenti. Viene così a cadere il concetto di applicazione singola .

1. Ad esempio è possibile che un dispositivo mobile usi la
2. business logic 1 per elaborare i dati presenti nei
3. data service 1 e 2, ad esempio per produrre un report dei risultati aziendali;
4. lo stesso dispositivo client può usare un'altra
5. business logic, per effettuare computazioni differenti,
6. magari sugli stessi dati aziendali, questa volta generando dei grafici di previsione. Senza necessità di modificare il codice esistente,
7. si possono usare altri dispositivi per accedere ai servizi offerti dalle
8. business logic presenti nell'application layer. Inoltre, eventuali modifiche nella business logic non comportano cambiamenti né
9. nel livello data store, né in quello di presentation.

➔
**3-tier: vantaggi**
29

---

**Flessibilità e modificabilità**

- Componenti usabili in sistemi diversi
- Modifiche indipendenti
- Estensioni dell'applicazione facilitate
- Ricerca di bug agevolata


**Interconnettività**

- Non sono necessari gli *adapter*
- Accesso a dati comuni da parte di applicazioni diverse facilitato

Impianti Informatici
POLITECNICO DI MILANO

I principali vantaggi dell'architettura 3-tier sono:

1. La flessibilità e modificabilità dei sistemi
2. I componenti possono essere riutilizzati in differenti sistemi, rendendo più rapida la realizzazione delle applicazioni
3. se l'interfaccia viene mantenuta inalterata, le modifiche apportate ad una componente non impattano sul resto del sistema,
4. così come eventuali estensioni dell'applicazione, sia implementando nuove funzionalità alle componenti correnti, sia aggiungendo delle componenti.
5. Grazie alla separazione e isolamento delle diverse funzionalità, la ricerca dei bug risulta più focalizzata.
6. Le API delle componenti superano il problema degli adapter del modello client/server, permettendo a molteplici interfacce diverse di connettersi allo stesso servizio,
7. e consentendo l'uso dello stesso gestore dei dati da parte di business logic diverse.

 Web Application 3-tier: vantaggi

30

Gestione di sistemi distribuiti

- Business logic aggiornabile senza interventi sui client

Preleva direttamente i dati aggiornati (*hot data*) operando on-line sul database

- Consente di realizzare pagine HTML dinamiche
- Elimina la necessità di aggiornare le pagine HTML (statiche)

Impianti Informatici

POLITECNICO DI MILANO

Nel caso specifico di una web application costruita secondo il modello 3-tier,

1. viene favorita la gestione applicazioni web distribuite, ad esempio sistemi con alcuni server replicati e client remoti,
2. in cui la business logic può essere aggiornata senza richiedere un aggiornamento dei client, che sarebbe molto oneroso da sostenere.
3. La struttura a 3 livelli si presta molto bene all'interazione con basi di dati,
4. Consentendo di realizzare pagine web dinamiche, e di
5. Eliminare l'uso di pagine statiche, difficilmente aggiornabili.

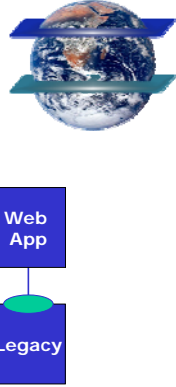
➤
**3-tier: svantaggi**
31

Dimensione ed efficienza

- Comunicazione tra componenti onerosa
- Uso di apposite librerie software

Software Legacy


- Difficile applicare il modello 3-tier
- Uso di *adapter*



Impianti Informatici
POLITECNICO DI MILANO

Analizzando le problematiche del

1. modello 3-tier, emerge
2. La maggiore dimensione ed efficienza del codice, dovute essenzialmente al
3. pesante scambio di messaggi attraverso le rete, con i relativi tempi di latenza dovuti all'overhead della comunicazione.
4. Essa richiede inoltre l'uso di librerie software per lo scambio di informazioni, aumentando la voluminosità del codice.
5. Molte imprese usano tutt'oggi del software vecchio per gestire i propri dati, basato sul modello monolitico, il che
6. rende difficile l'applicazione del modello 3-tier e necessita
7. l'uso di adapter per interfacciarsi con tale software



Web Application 3-tier: svantaggi


32

Web server e Db server sono spesso i colli di bottiglia del sistema

Elevati problemi di *sicurezza*

- Virus
- Worm
- Trojan horses

Rapida evoluzione delle tecnologie



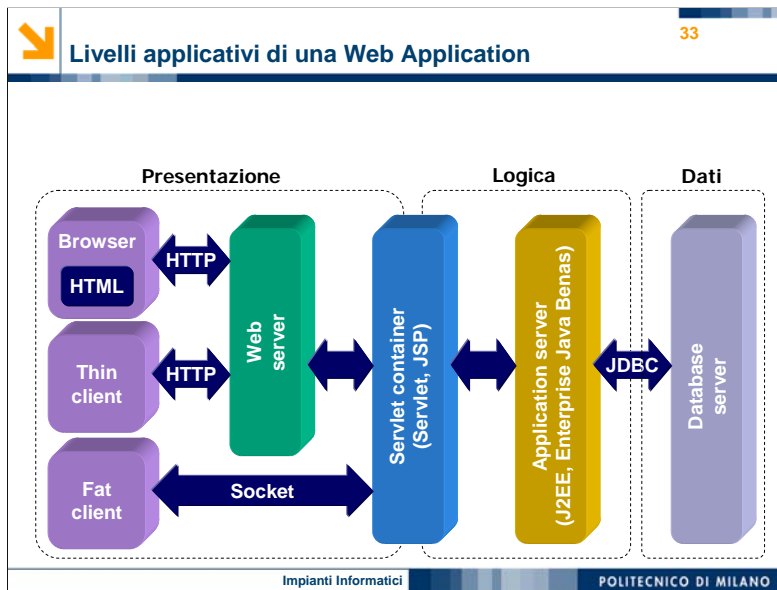
Impianti Informatici

POLITECNICO DI MILANO

Nel particolare caso di una applicazione web,

1. risulta che il Web Server e il Database Server tendono a diventare i colli di bottiglia del sistema, richiedendo un opportuno dimensionamento dell'impianto.
2. Emergono inoltre delicate questioni di sicurezza, dovute essenzialmente al dominio in cui le web application vengono impiegate, e cioè quello di Internet, esponendole a possibili attacchi da parte di malintenzionati,
3. di virus, worm o cavalli di troia, con elevati problemi di confidenzialità, integrità e disponibilità delle informazioni.
4. Come ultima considerazione, le tecnologie su cui sono basate le web application sono in continua e rapida evoluzione, così da rendere difficoltose le scelte progettuali.





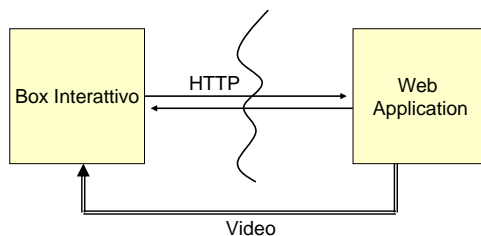
Osservando una applicazione web reale,

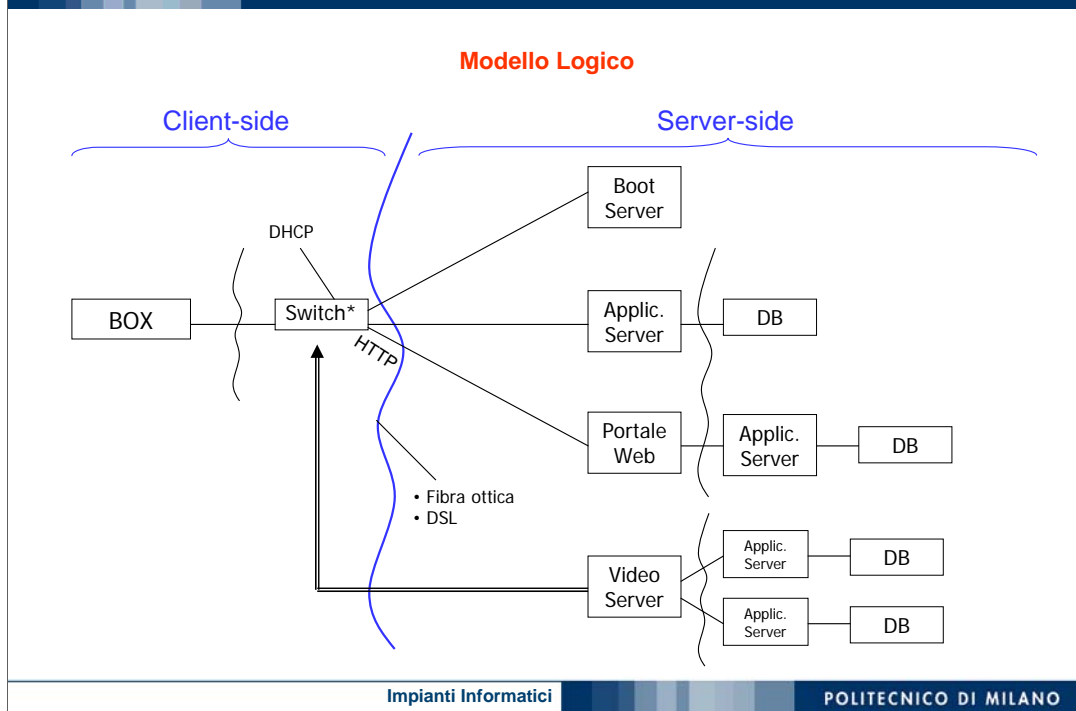
1. si può generalmente notare una suddivisione in un numero di livelli differente dai 3 appena analizzati;
  2. i livelli applicativi sono infatti solitamente superiori in numero rispetto ai dai livelli logici. Ad esempio, nello schema mostrato, ci sono 5 livelli applicativi:
  3. Il browser con cui interagisce l'utente
  4. Il server web che intercetta le richieste http del client
  5. Un servlet container, un particolare tipo di application server
  6. Un altro application server, EJB-enabled,
  7. E infine un database server,
  8. collegato ai livelli precedenti mediante JDBC.
- Ovviamente la web application può basarsi su client differenti dal browser,
9. sia di tipo thin, quindi con sole capacità di visualizzazione,
  10. che fat, con inclusione di maggiori funzionalità.
- Nonostante una web application sia basata sul
11. protocollo http, il collegamento tra i vari livelli potrebbe avvenire anche mediante
  12. altri protocolli o tecnologie, ad esempio mediante l'uso di socket



### Web Application per il Video On Demand (VOD)

Il Box è un PC ed opera come un browser (pur non avendo la classica interfaccia dei browser), scambiando messaggi HTTP con una WebApp. Lo streaming del video richiesto avviene poi con un altro protocollo





STB = Set Top Box