

The background of the slide is a blue gradient with a pattern of binary code (0s and 1s) floating around. On the left side, there is a partial view of a laptop screen and keyboard.

Unità di apprendimento 1

**Architettura di rete e
metodologia di sviluppo**

The background of the slide is a blue gradient with a pattern of binary code (0s and 1s) in a lighter blue color. On the left side, there is a partial view of a laptop screen and keyboard.

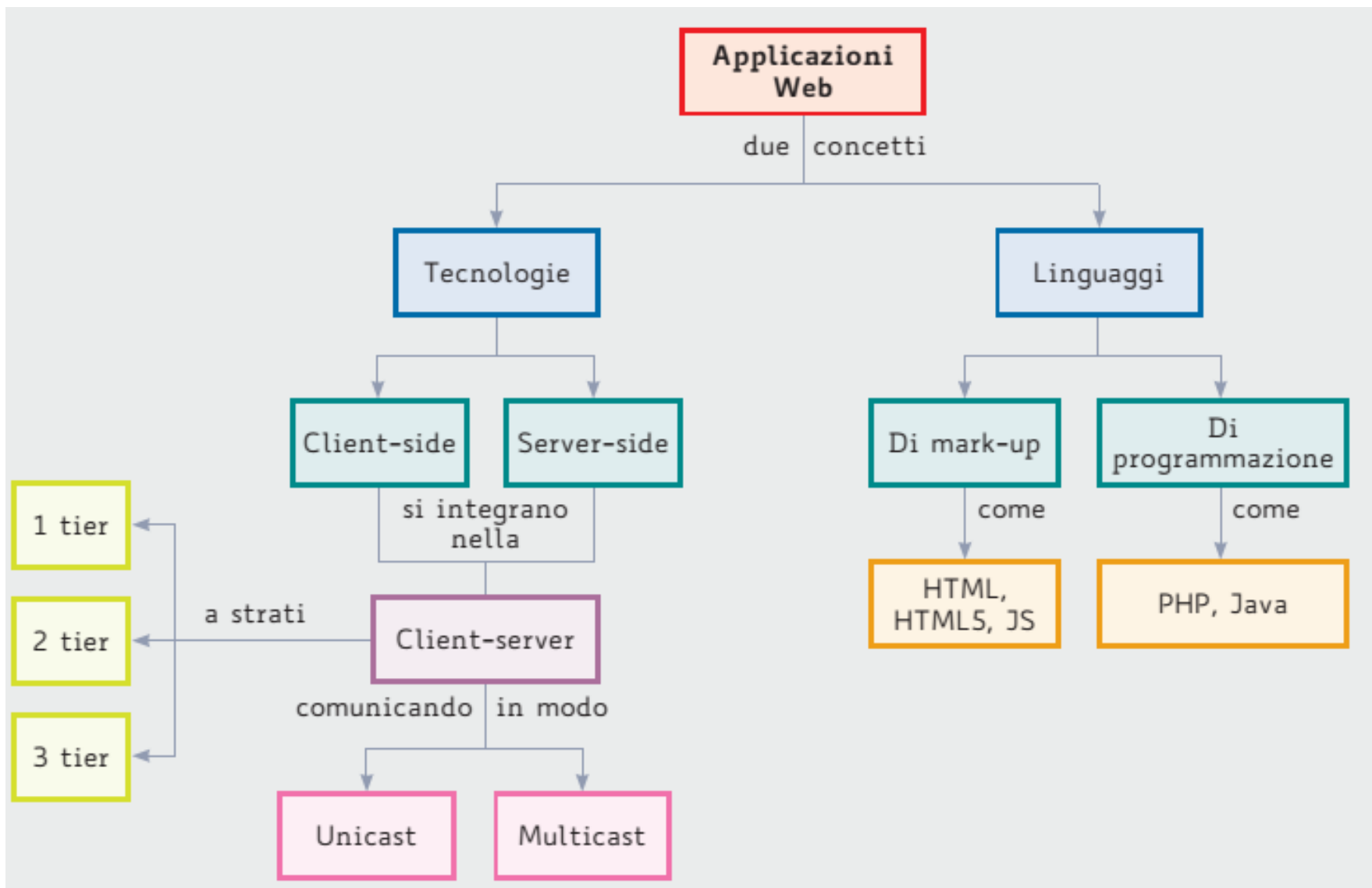
Unità di apprendimento 1

Lezione 3

**Le applicazioni Web
e il modello client-server**

In questa lezione impareremo:

- **gli aspetti essenziali delle tecnologie per il Web**
- **le caratteristiche del modello client-server**
- **l'evoluzione del modello client-server**

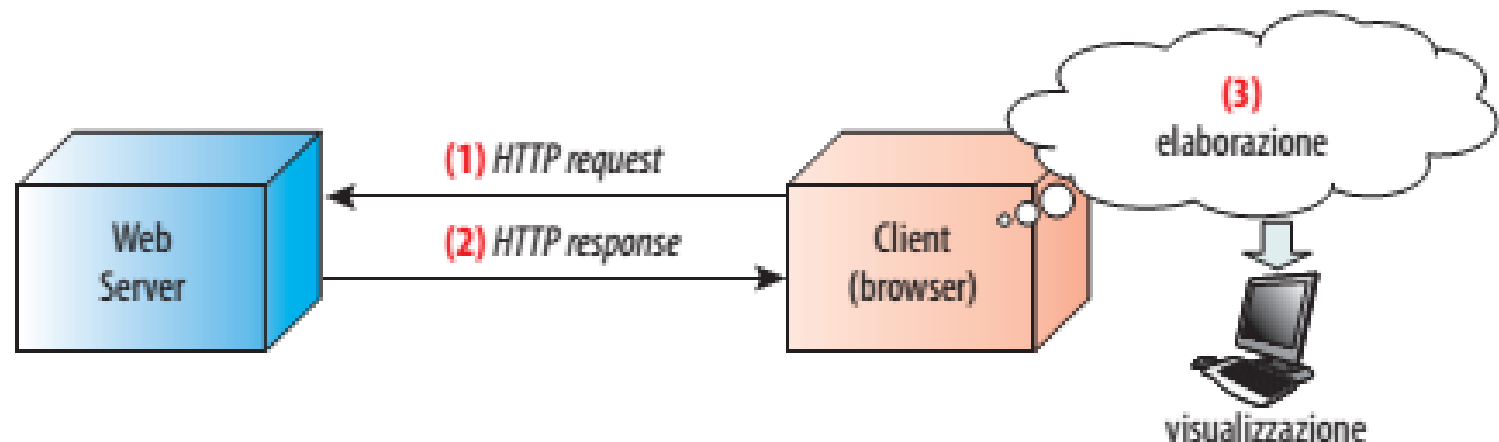


Applicazioni Web: generalità

- Con il termine di **applicazione web** viene definito il software sviluppato e utilizzato attraverso tecnologie Web e linguaggi specifici
- Alla base delle applicazioni web abbiamo due concetti fondamentali:
 - tecnologie **client-side** e **server-side**
 - linguaggi di **mark-up** e linguaggi di **programmazione**

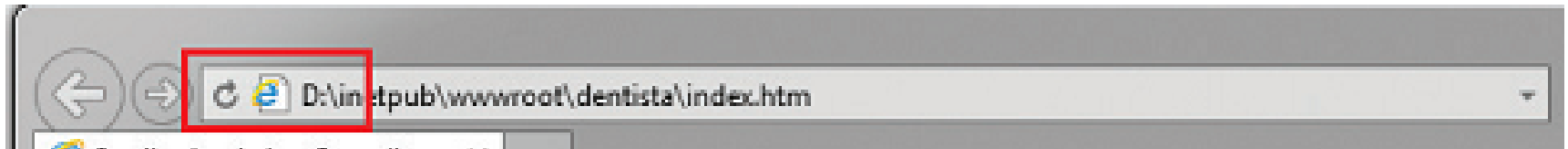
Tecnologie del Web

- Le tecnologie del web si distinguono in base al **posto** in cui avvengono le elaborazioni
 - Tecnologie client-side**: sono le strutture tecnologiche in cui l'elaborazione avviene sul client, tipicamente nel **browser**



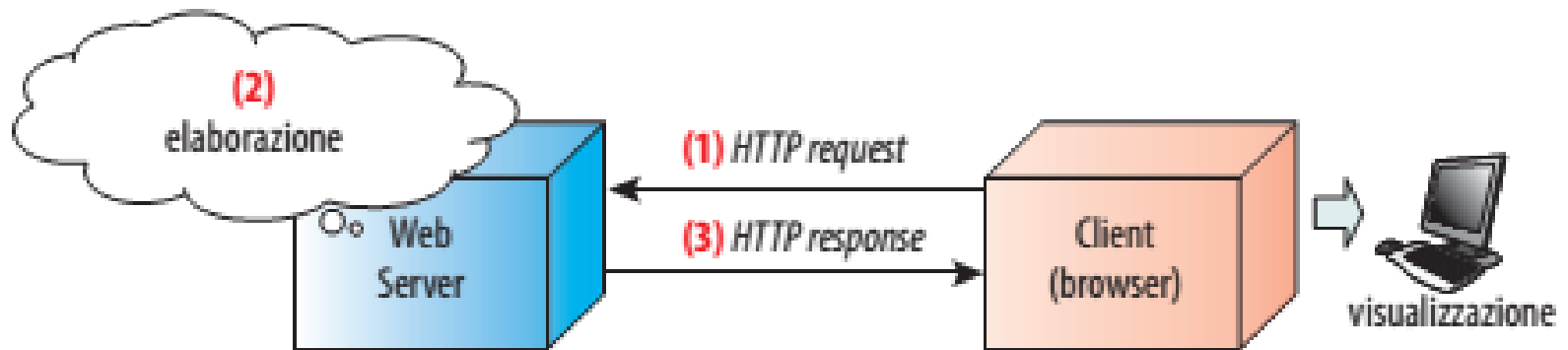
Tecnologie del Web

- Per visualizzare una pagina che utilizza una tecnologia client-side non abbiamo bisogno necessariamente di un **Web server**
- Basta memorizzare una pagina web sul file system locale e indicare al browser il percorso ad essa



Tecnologie del Web

- **Tecnologie server-side**: l'elaborazione avviene sul **server**, tipicamente in quello che chiamiamo Web server



Tecnologie del Web

- Per visualizzare una pagina che utilizza una tecnologia **server-side** abbiamo bisogno di un **Web server** che elabori il codice della pagina ed è, quindi, necessario connettersi e richiedere la pagina tramite un URL

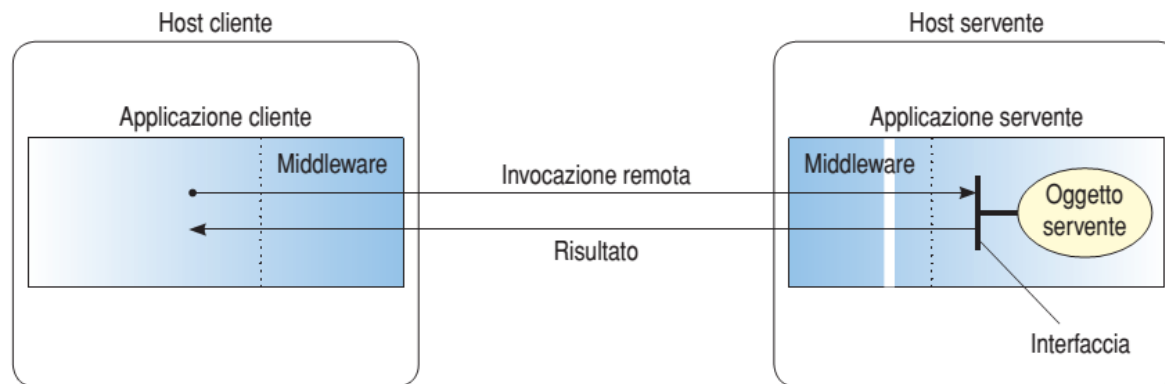


Tecnologie del Web

- Le parti di codice che utilizzano tecnologie client-side **sono** visibili
- Viceversa le parti di codice che utilizzano tecnologie server-side **non sono** visibili
- Perché?
- Tipologie di linguaggi del web:
 - linguaggi di **mark-up** => documenti strutturati
 - linguaggi di **programmazione** => sequenze di istruzioni

Il modello client-server

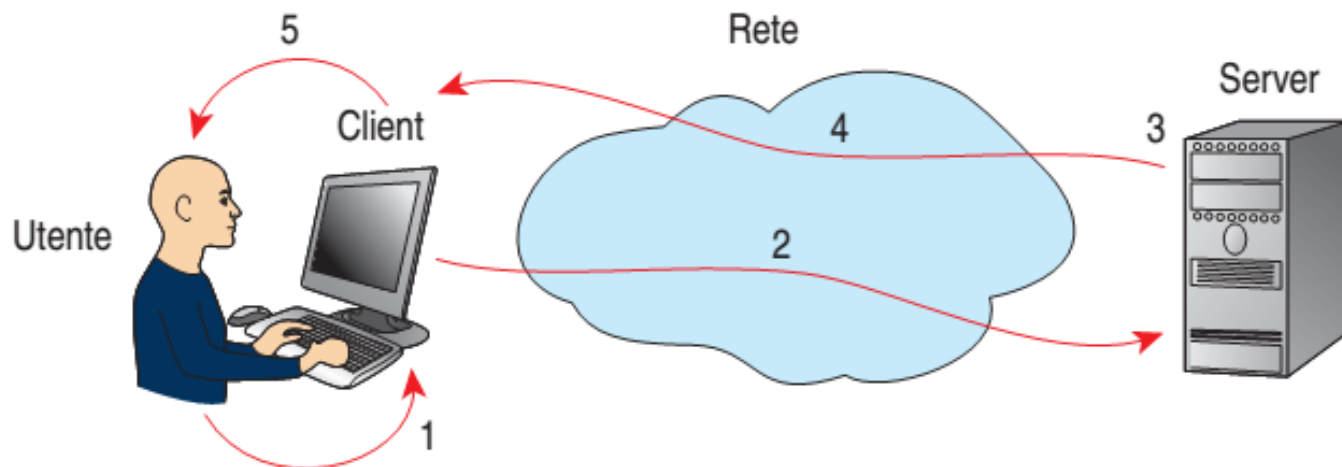
- Il modello **client-server** è costituito da:
 - un insieme di **host** che gestiscono una (o più) risorse, i **server**
 - un insieme di client che ne richiedono l'accesso



- Attenzione! Non definiamo server o client gli host, ma i **processi** che sono in esecuzione su di essi (ripensando all'actor, che ne deduci?)

Il modello client-server

- 1 il **client** *manda* una richiesta al **server**
- 2 il **server** (in attesa) *riceve* la richiesta
- 3 il **server** *esegue* il servizio richiesto (generando un thread concorrente)
- 4 il **server** *manda una risposta* ed eventualmente dei dati
- 5 il **client** *riceve* la risposta ed eventualmente i dati



Il modello client-server

- Servizi tipici delle architetture client-server
 - Telnet
 - HTTP
 - FTP,
 - SMTP, IMAP, ...

Distinzione tra server e client

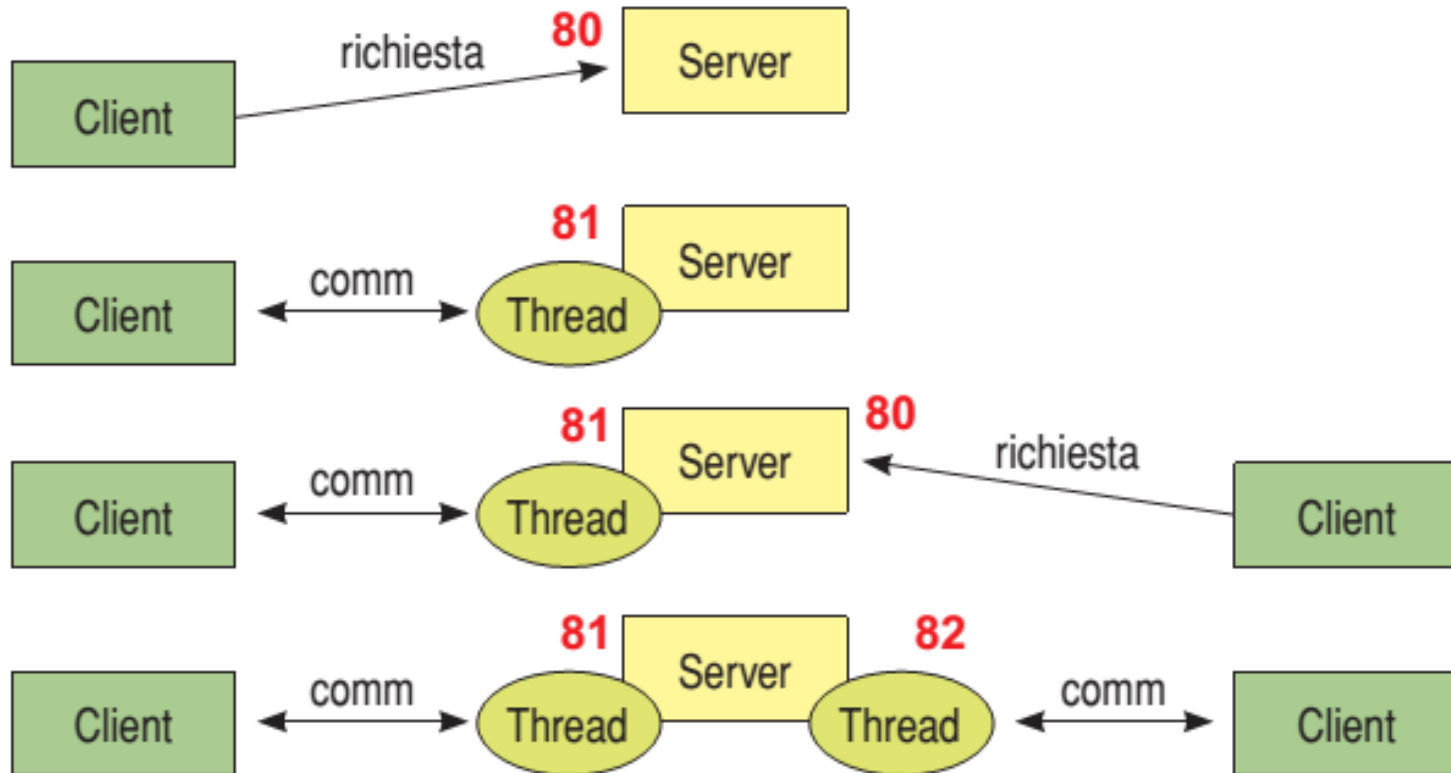
- Un client per comunicare con un server usando il protocollo **TCP/IP** deve “connettersi” alla **socket** dell’host dove il server è in esecuzione specificando l’indirizzo
- Una socket è formata dalla coppia
 - **<indirizzo IP:numero della porta>**
- e permette di individuare univocamente il processo che sta richiedendo/offrendo un servizio

Comunicazione unicast e multicast

- Nel modello client-server possiamo avere due tipi di comunicazione:
- **unicast**: il server risponde alle richieste dei **singoli client** (ad es. una pagina web)
- **multicast**: il server risponde alle richieste di **più client** (ad es. uno streaming a un gruppo multicast)

Distinzione tra server e client

- Nel caso di trasmissione **multicast**



Livelli e strati

- Le architetture client-server sono normalmente organizzate a livelli (tier)
- Ad ogni livello corrisponde un **nodo** (o un gruppo di nodi) che offre una funzionalità
- Livelli nelle applicazioni **web**:
 - front-end o **presentation tier**
 - logica applicativa o **middle tier**
 - back-end o **data tier**

Livelli e strati

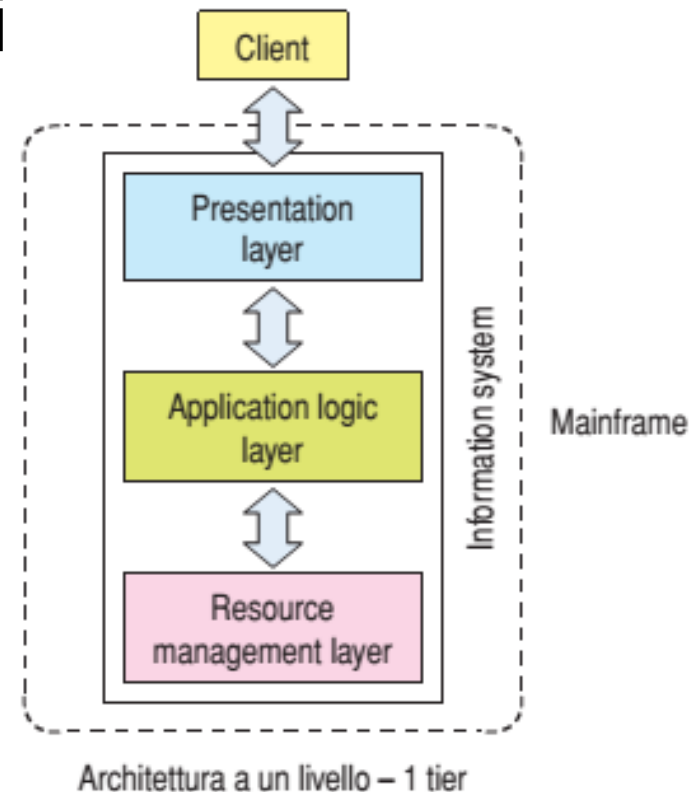
- Spesso il modello client-server a livelli è combinato con quello a strati che invece si utilizza per le applicazioni informatiche **in generale**, ovvero:
 - **Presentation Layer (PL)**: è composta dall'insieme delle procedure o moduli dedicate all'acquisizione e alla presentazione dei dati all'utente (es. moduli per creare le pagine web)

Livelli e strati

- **Resource Management Layer:** è composto dall'insieme delle procedure che gestiscono i dati, cioè memorizzano e recuperano le informazioni persistenti dagli archivi di massa delle basi di dati
- **Business Logic Layer (BLL) o Resource Management Layer (RML):** è il “corpo centrale” dell'applicazione che comprende la logica dell'elaborazione e le definizioni delle relazioni esistenti tra le diverse entità

Architettura a un livello – 1 tier

- In questa architettura vi è un mainframe al quale erano collegati i terminali “stupidi”
- Non è una architettura client-server
- è l'architettura che si utilizzava prima dell'avvento dei sistemi distribuiti

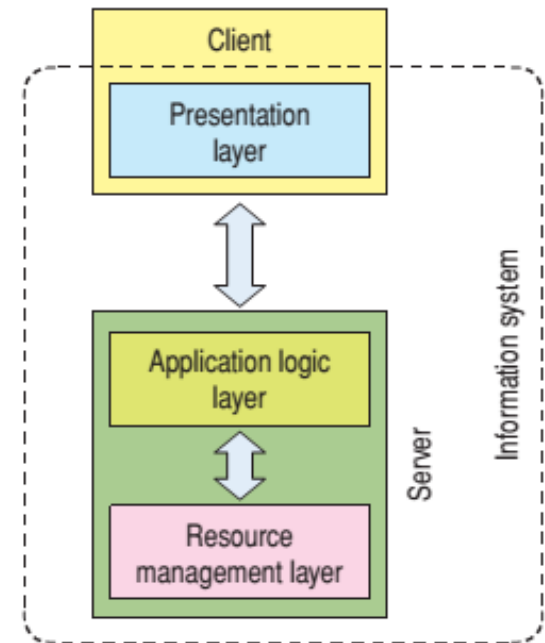


Architettura a due livelli – 2 tier

- Con l'avvento delle reti locali sono nate le architetture client-server dove le funzionalità e le responsabilità erano suddivise su due livelli:
 - un livello server
 - un livello client
- Possiamo individuare due sottocategorie per architetture 2 tier
 - thin-client
 - thick-client

Architettura a due livelli – 2 tier

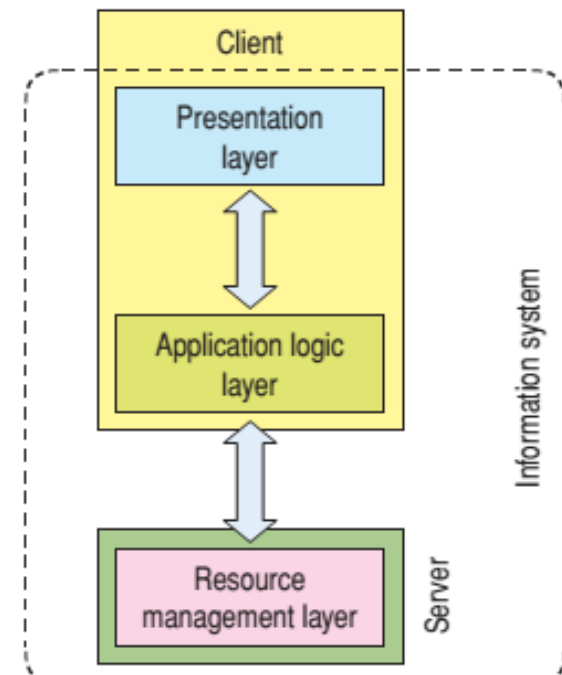
- Nel modello **thin-client**:
 - il server è responsabile della logica applicativa e della gestione dei dati
 - il client è responsabile della esecuzione del software di presentazione



Architettura a due livelli – 2 tier – modello **thin**

Architettura a due livelli – 2 tier

- Nel modello **thick-client** (o fat-client):
 - il **server** è responsabile della gestione dei dati
 - il **client** è responsabile della presentazione e della logica applicativa
- Successivo al thin-client
- Ha favorito la connessione di host di tipo diverso



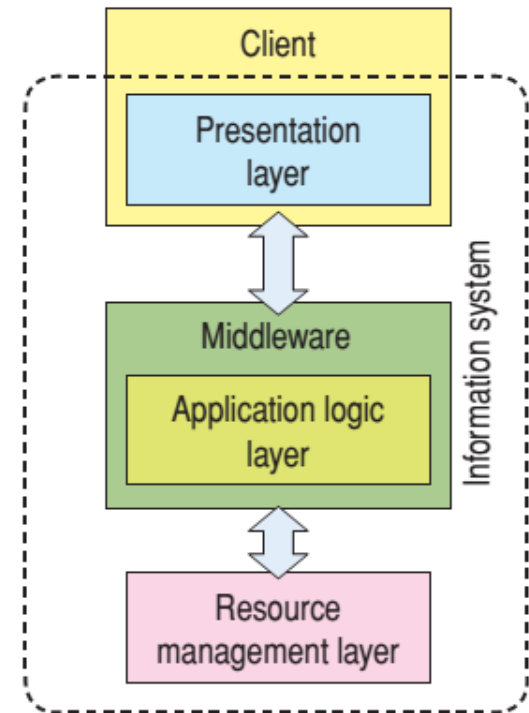
Architettura a due livelli – 2 tier – modello **thick**

Architettura a due livelli – 2 tier

- Il limite delle architetture client/server a due livelli è che sono **poco scalabili**
 - il server deve gestire la connessione e lo stato della sessione di ciascun client
 - questo carico di elaborazione porta alla limitazione del numero di client che possono essere gestiti contemporaneamente

Architettura a tre livelli – 3 tier

- I vantaggi dell'introduzione del middleware (software che agisce da intermediario)
 - elaborazione distribuita
 - scalabilità
 - sicurezza
- Svantaggi
 - tempi di comunicazione
 - progettazione, sviluppo e amministrazione



Architettura a tre livelli – 3 tier

Architettura a n tier

- Le architetture client-server a N livelli sono una generalizzazione del modello client-server a tre livelli
 - vengono scomposti e introdotti un numero qualunque di livelli e server intermedi
- Questa scomposizione viene effettuata per suddividere ulteriormente i compiti dei vari strati
 - prende anche il nome di applicazione **multi-tier**

SCELTA MULTIPLA



1 Nelle applicazioni Web con le tecnologie client-side:

- a l'elaborazione avviene sul client
- b l'elaborazione avviene sul server
- c l'elaborazione avviene sia sul client sia server
- d l'elaborazione avviene prima sul client e successivamente sul server

2 Indicare quali di queste affermazioni sono false per un modello client-server:

- a una macchina server non può essere anche client
- b non sono gli host a essere server o client ma i processi
- c un host può essere contemporaneamente sia client che server
- d ogni processo server può a sua volta diventare client
- e un processo client non può diventare server
- f un servizio è un'entità astratta che viene fornito da uno o più server

3 Quali caratteristiche non sono caratteristiche delle applicazioni di tipo enterprise:

- a molti utenti concorrenti che richiedono i servizi
- b una logica applicativa poco complessa
- c archivi di grandi dimensioni con organizzazione di dati complessa e distribuita

d richiedono modesti requisiti di sicurezza
e gli utenti possono riferirsi a tutte le risorse in un modo esclusivo

4 Le applicazioni client-server di tipo enterprise hanno (indicare la risposta errata):

- a molti utenti concorrenti che richiedono i servizi
- b una logica applicativa complessa
- c archivi di grandi dimensioni con organizzazione di dati centralizzata
- d notevoli requisiti di sicurezza
- e sistemi transazionali

5 Quale tra le seguenti non è una tipica applicazione delle architetture client-server?

- | | |
|----------|--------|
| a Telnet | d DNS |
| b HTTP | e SMTP |
| c FTP | |

6 Nelle applicazioni possiamo individuare tre tipi principali di funzionalità che corrispondono a una struttura in tre strati o livelli (modello three-tier):

- a front-end, presentation tier, back-end
- b middle tier, front-end, presentation tier
- c back-end, data tier, front-end
- d presentation tier, middle tier, data tier
- e front-end, logica applicativa, middle tier

