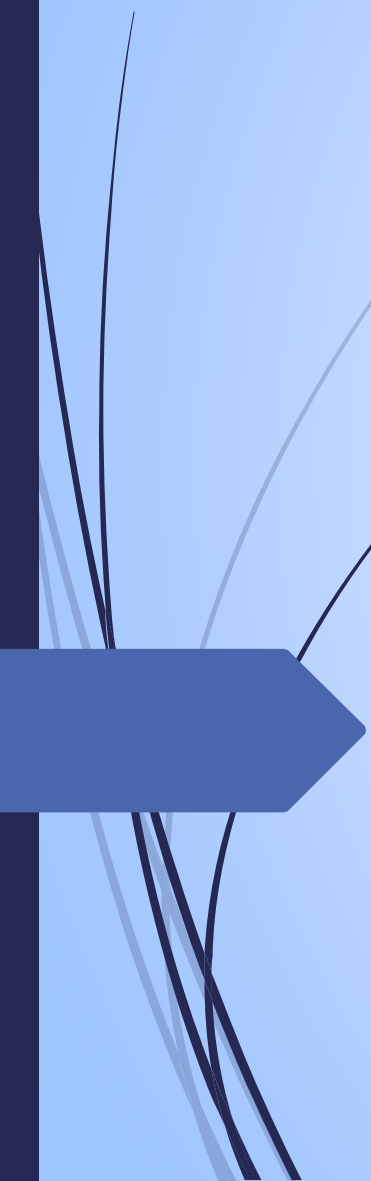


SISTEMI DISTRIBUITI



“Tu sai di averne uno quando il guasto di un computer di cui non hai mai sentito parlare non ti permette di fare il tuo lavoro.”

-Lamport

Presentazione creata da:
Lucero Allison
Guevara Katherin
Rando Marco

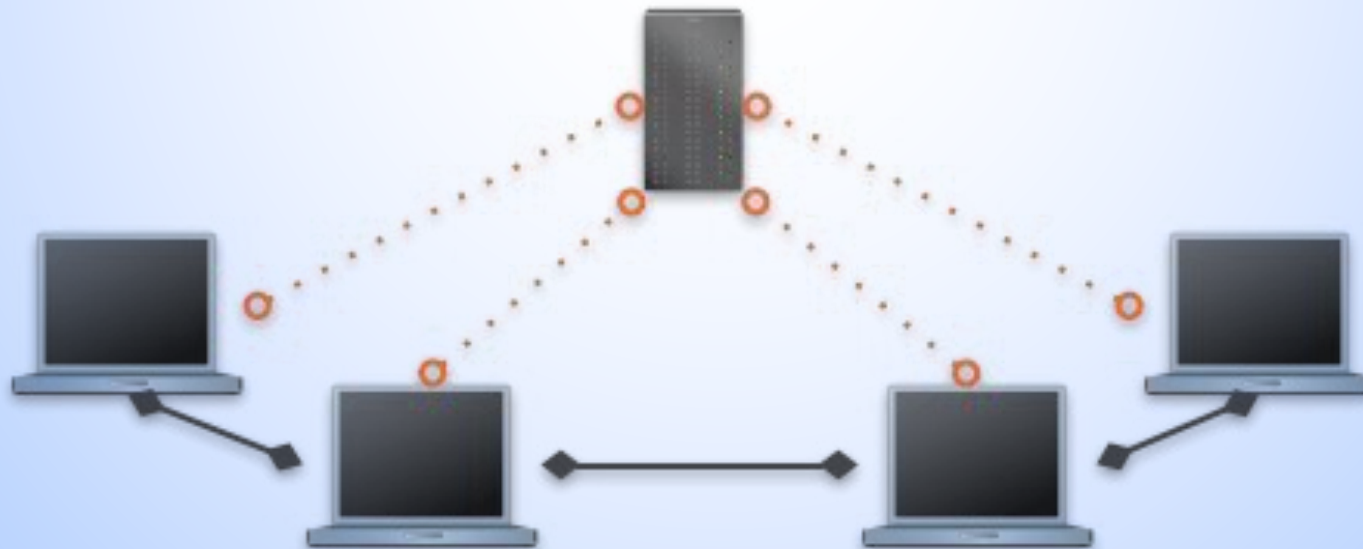
Indice

- Definizione di sistema distribuito
 - Organizzazione di un sistema distribuito
 - Caratteristiche
- Architettura HW
- Accenni modello ISO-OSI
- Layer-Tier



Definizione di sistema distribuito

Sistema informatico costituito da un insieme di processi interconnessi tra loro in cui le comunicazioni avvengono solo esclusivamente tramite lo scambio di opportuni messaggi.



Organizzazione di un sistema distribuito

Offrire una visione unica del sistema che in realtà è composto da computer e reti eterogenei



Obiettivo

- Visione di un sistema unico



Soluzione

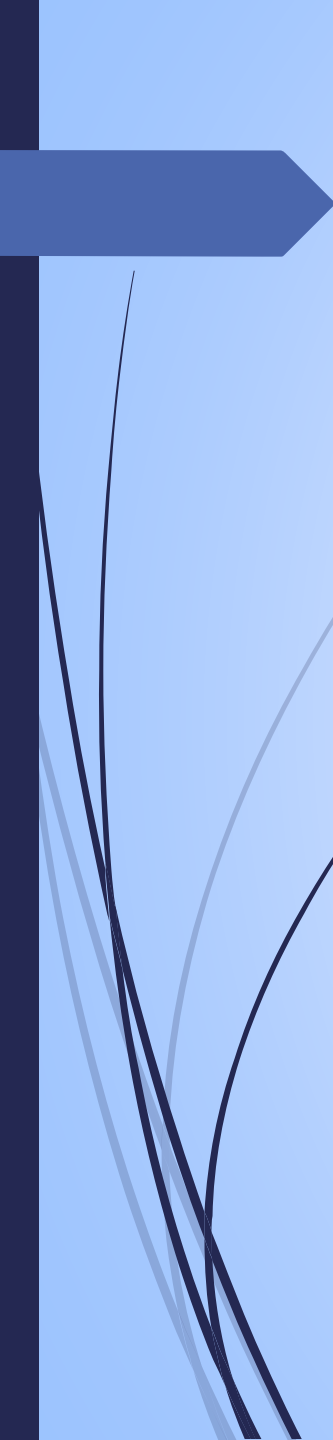
- utenti e applicazioni
- strato software
- sistema operativo
- middleware

Caratteristiche

Quali sono le caratteristiche principali che un sistema distribuito deve avere?



- consentire facilmente la connessione tra utenti e risorse
- essere trasparente
- essere aperto
- essere flessibile
- essere scalabile



“un sistema distribuito è formato da un insieme di entità indipendenti che cooperano per raggiungere un fine comune e tale cooperazione si effettua attraverso scambio di messaggi”



Come si può classificare??

La classificazione di un'architettura hw non è semplice infatti non esiste un modo univoco per determinare la tassonomia d'appartenenza di un determinato hw infatti in base ai fattori un'architettura può 'ricadere' in una piuttosto che in un'altra categoria.



Tassonomia di FLYNN

Una delle classificazioni più citate è quella di Flynn (1966) essa si basa sui due flussi di informazioni





Architettura HW

Un sistema distribuito è formato da più CPU, ma queste possono essere organizzate in diversi modi.

Possiamo distinguere tra:

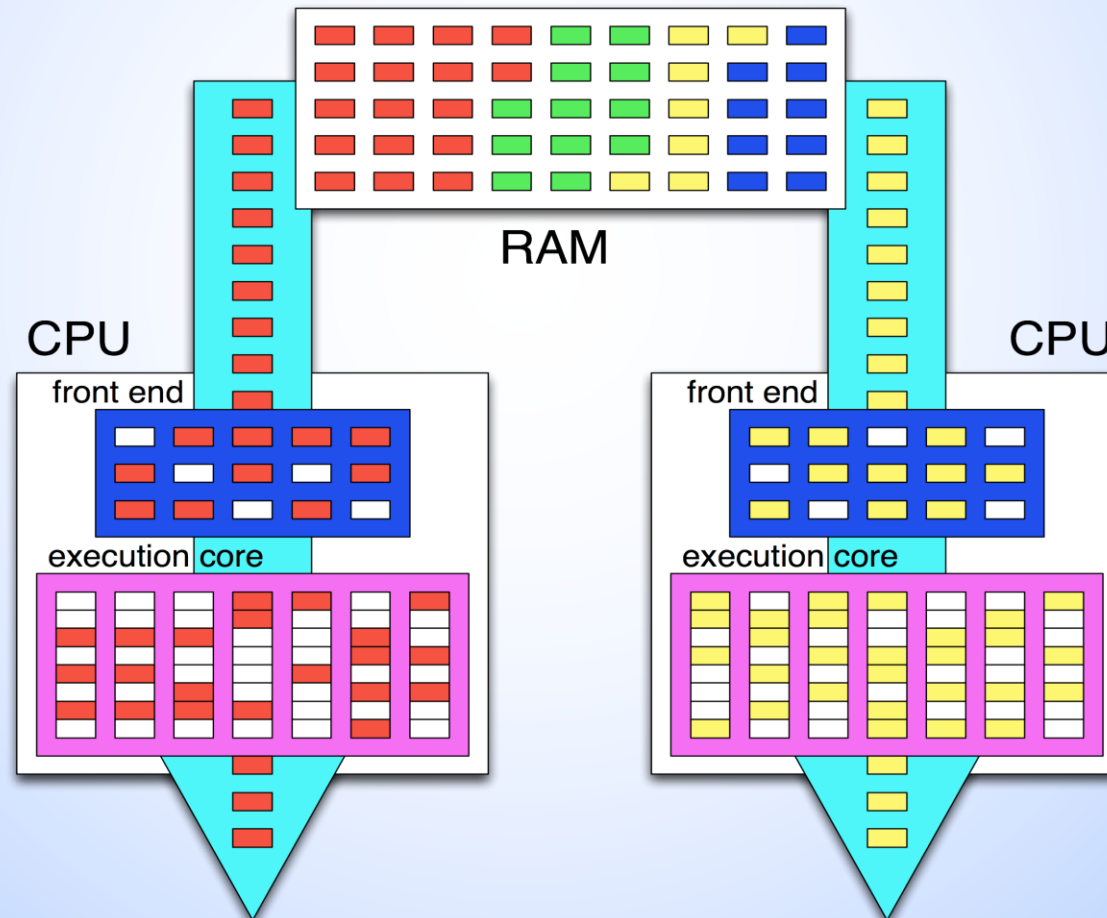
- Multiprocessori
- Multicomputer

All'interno un'ulteriore distinzione è data dalla rete di interconnessione:

- basata su BUS
- basata su SWITCH

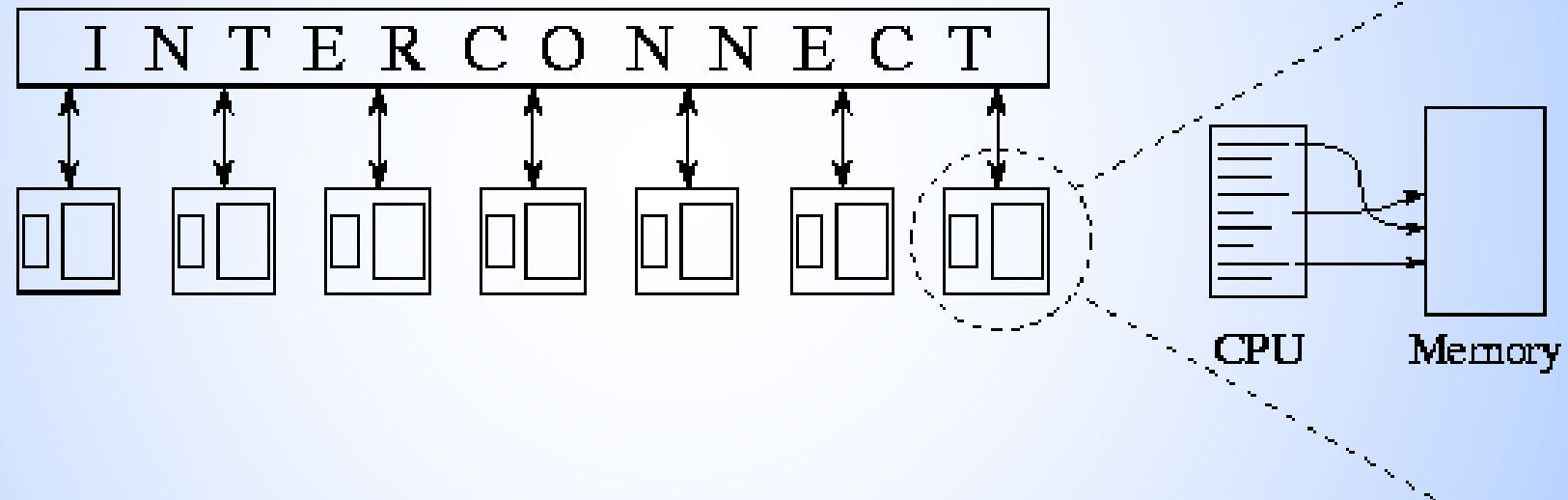
Multiprocessori

sistema di elaborazione composto da due o più processori operanti in parallelo

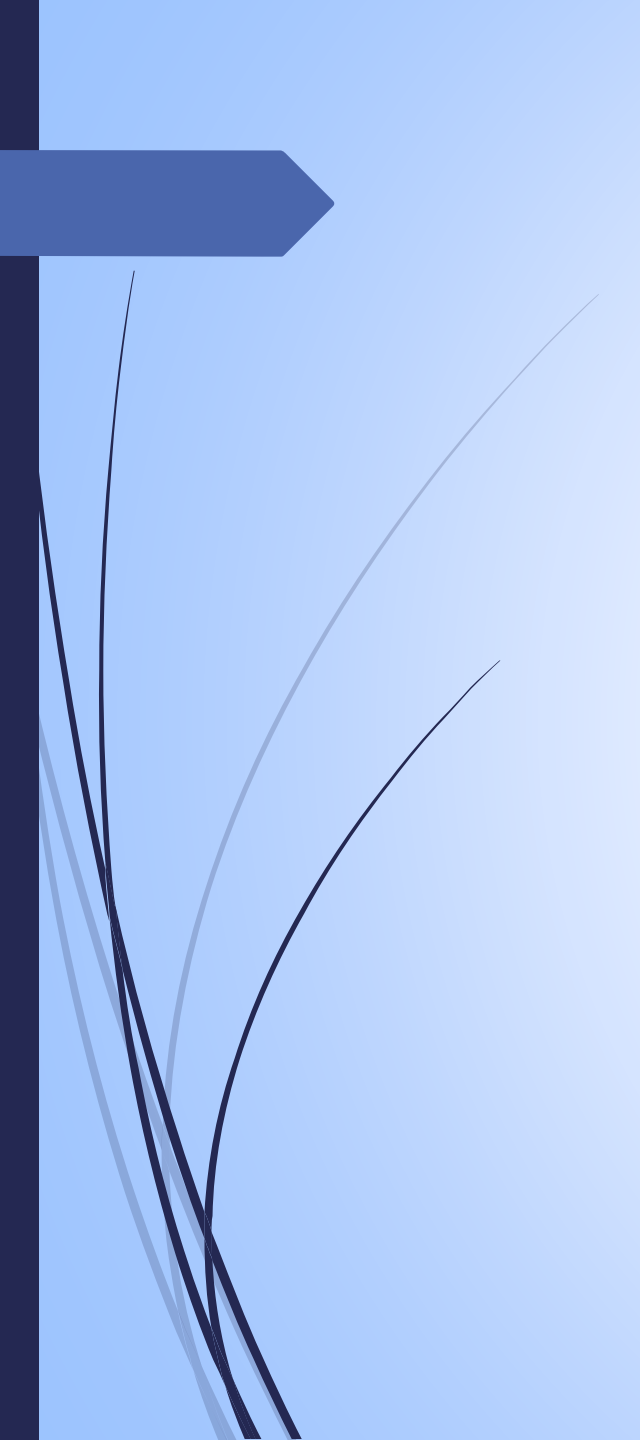


Multicomputer

“ Un computer composto da diversi computer . ”



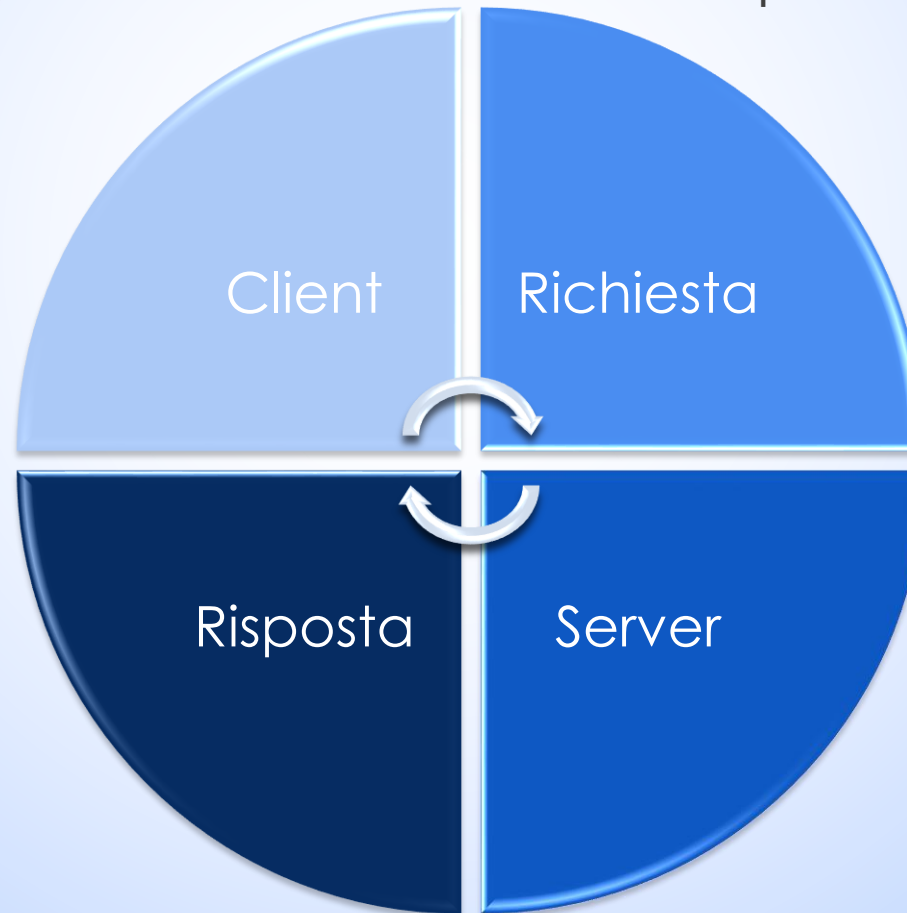
NB: Un computer multicore non è un multicomputer perché i più core condividono una memoria comune .



[...]

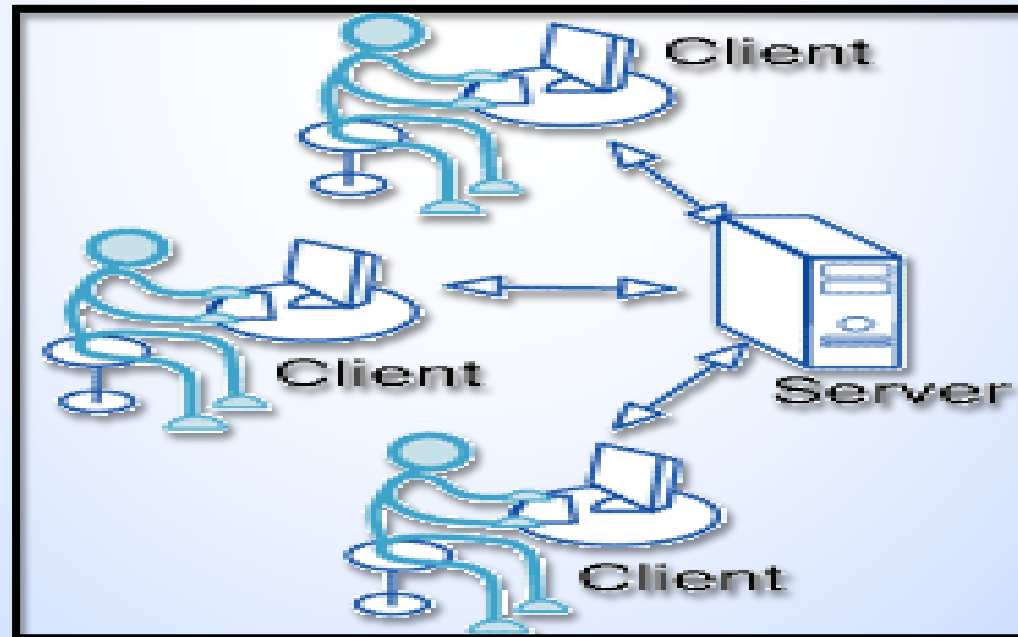
Definizione

Genericamente un computer *client* si connette ad un *server* per la fruizione di un certo servizio, quale ad esempio la condivisione di una certa risorsa hardware/software con altri client, appoggiandosi alla sottostante architettura protocollare.

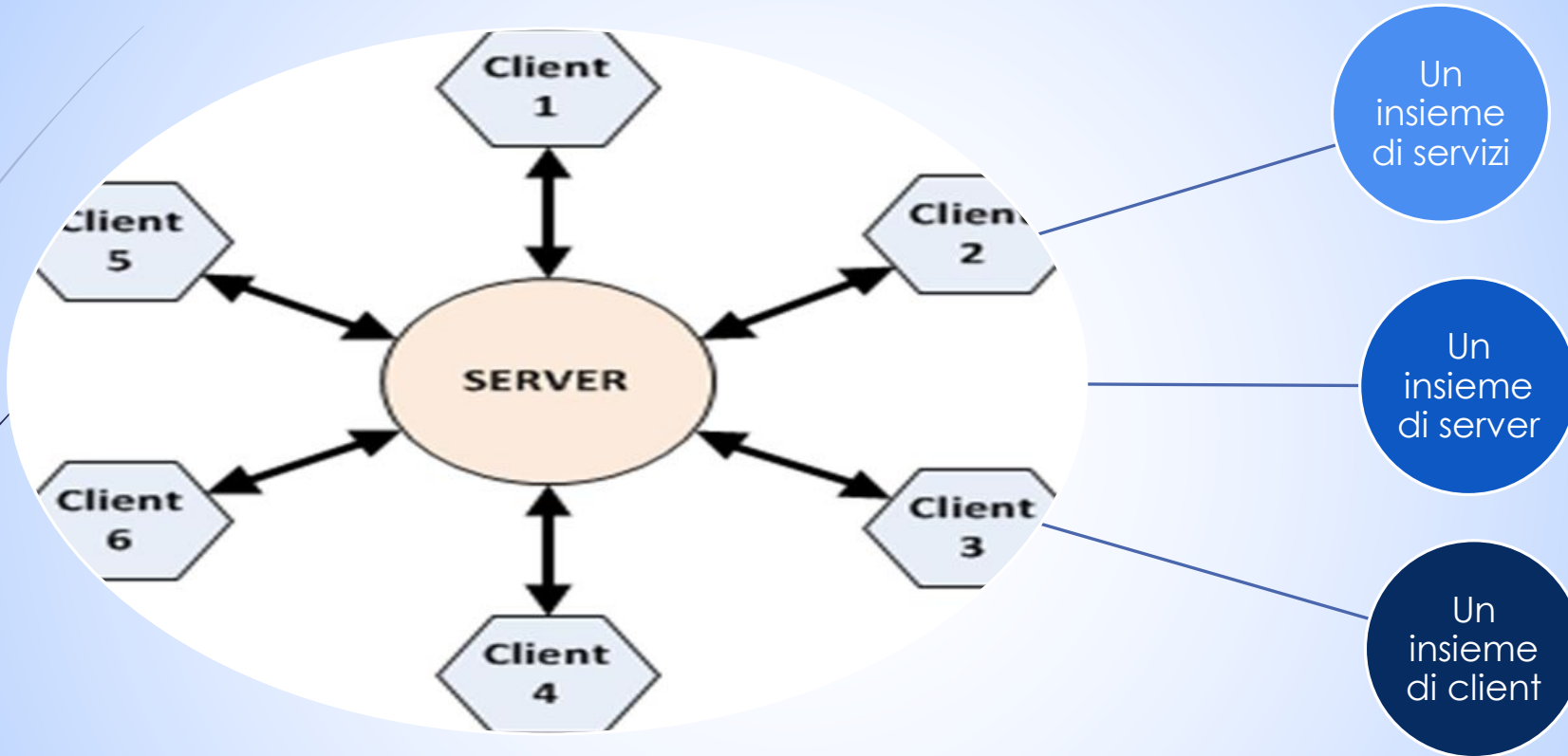


Descrizione

Più semplicemente, i sistemi *client/server* sono un'evoluzione dei sistemi basati sulla condivisione di risorse: la presenza di un *server* permette ad un certo numero di *client* di condividere le risorse del server stesso, lasciando la gestione degli accessi alle risorse a quest'ultimo, per evitare conflitti di utilizzazione tipici dei primi sistemi informatici.



Organizzazione di un sistema come...



CONSEGUENZE



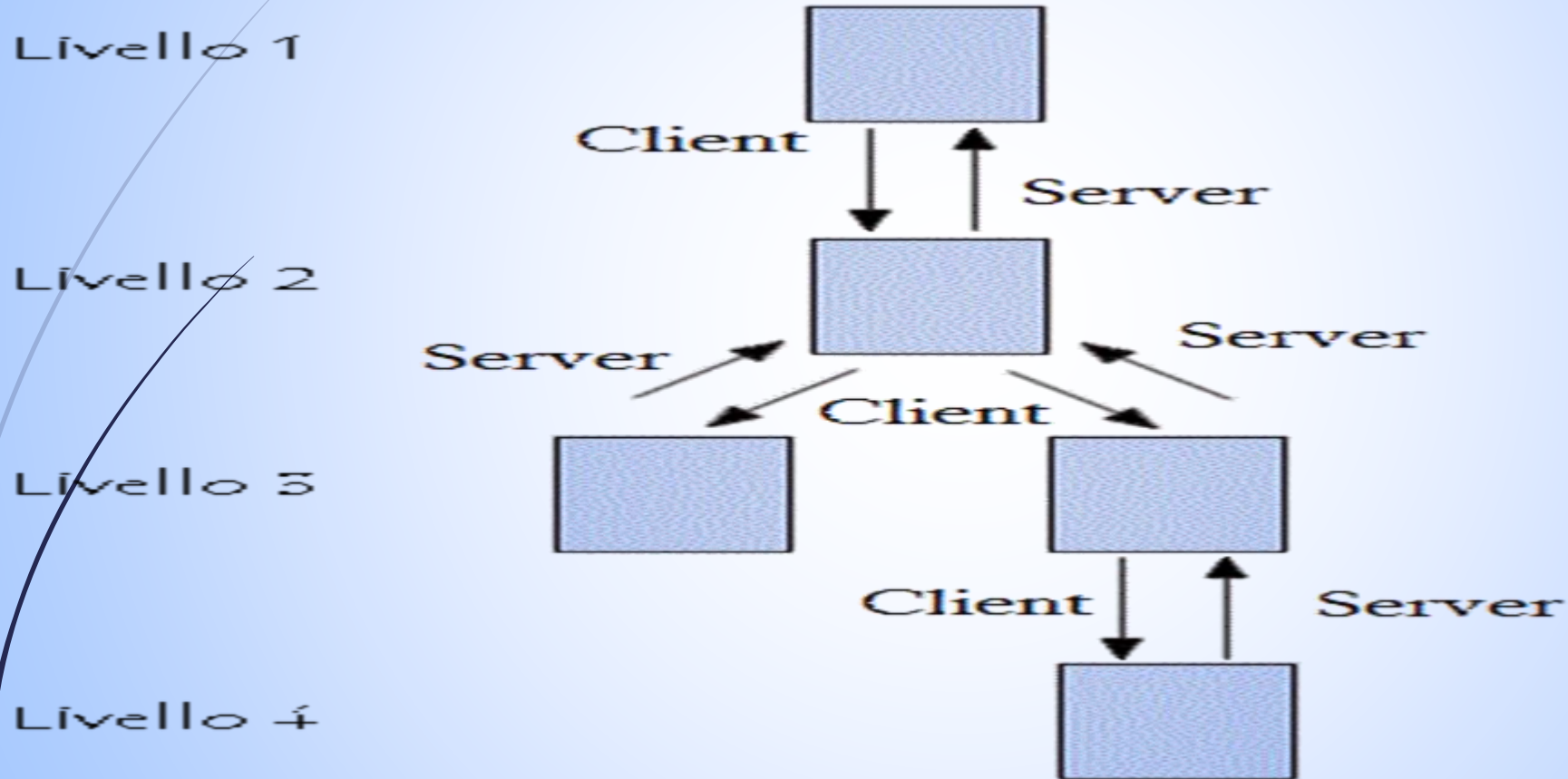
- 1) condivisione di risorse
- 2) centralizzazione di elaborazione



- 1) overhead della comunicazione

Architetture Client/Server

- Generalmente le architetture C/S sono normalmente organizzate a livelli



- Ogni livello funge da client per il livello successivo e da server per il precedente

Architettura 1 tier

- Architettura priva di un server
- Non è distribuita

CONSEGUENZE



Combinazione
strati

No problemi
compatibilità

Costo basso

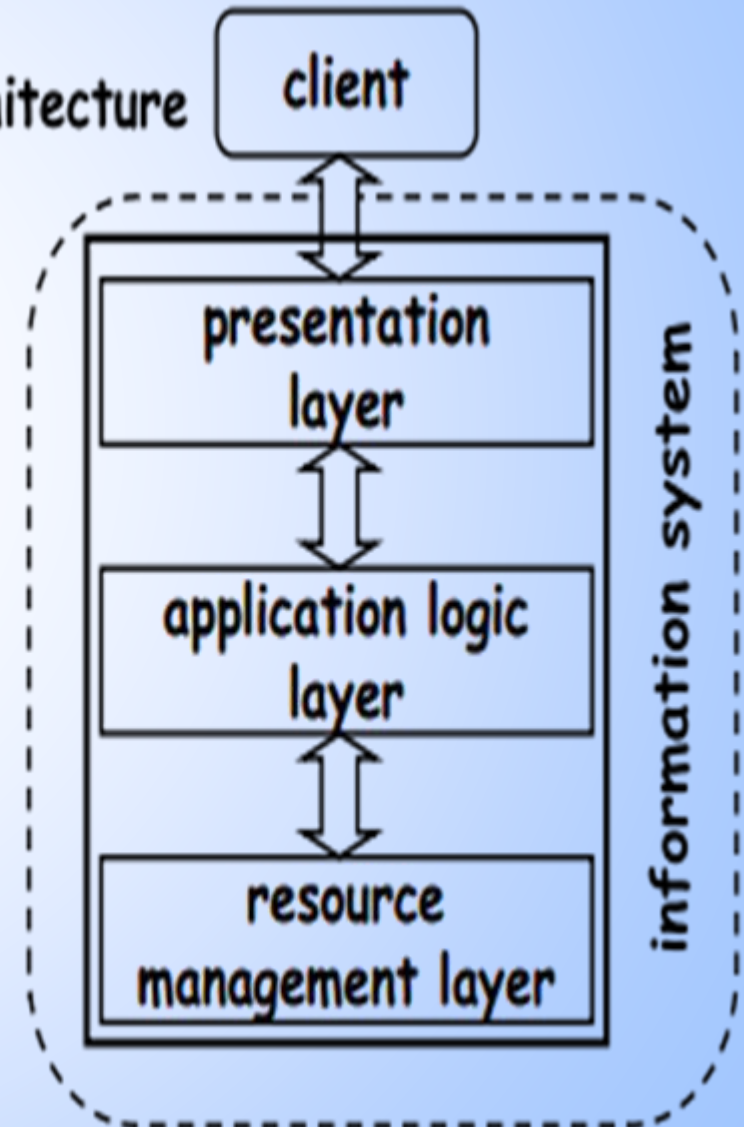
No overhead



Non aperto

Difficoltà
Manutenzione

1-tier architecture

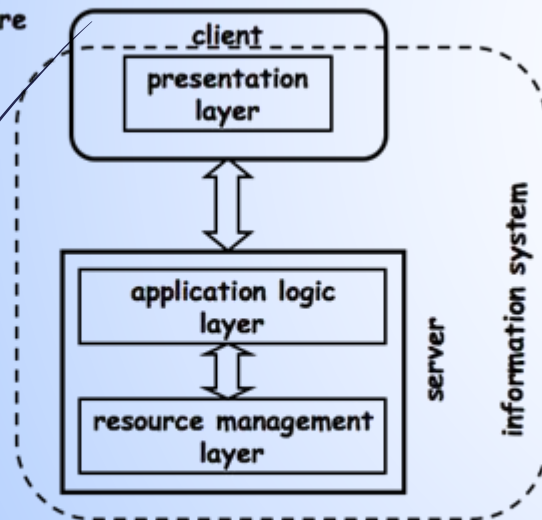


Architettura 2 tier

- Comprende 2 livelli(client e server)
- 2 tipi di architettura 2 tier (thin e fat client)

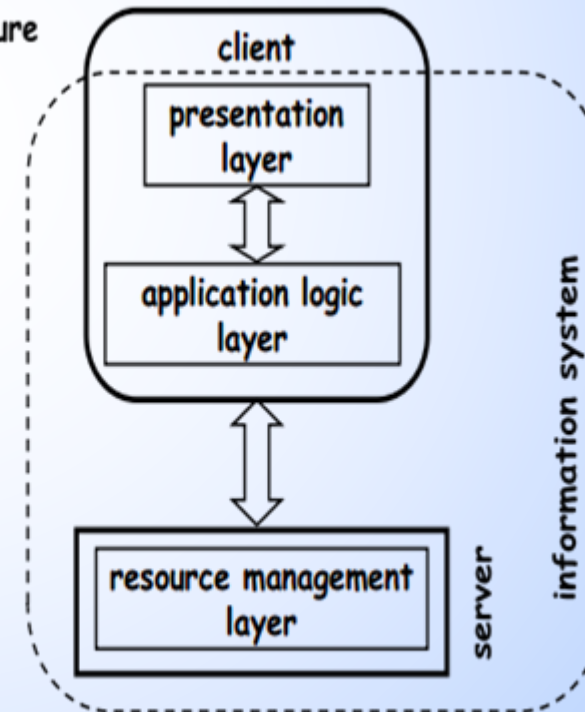
Thin client

2-tier architecture



Fat client

2-tier architecture



Vantaggi e svantaggi architettura 2 tier

THIN CLIENT

VANTAGGI

- Semplicità
- Potenza client limitata
- Client 'semplice'
- Buone prestazioni

SVANTAGGI

- Carico Concentrato
- Potenza del client non utilizzata

FAT CLIENT

VANTAGGI

- Distribuzione del carico
- Sviluppo nuovi c/s

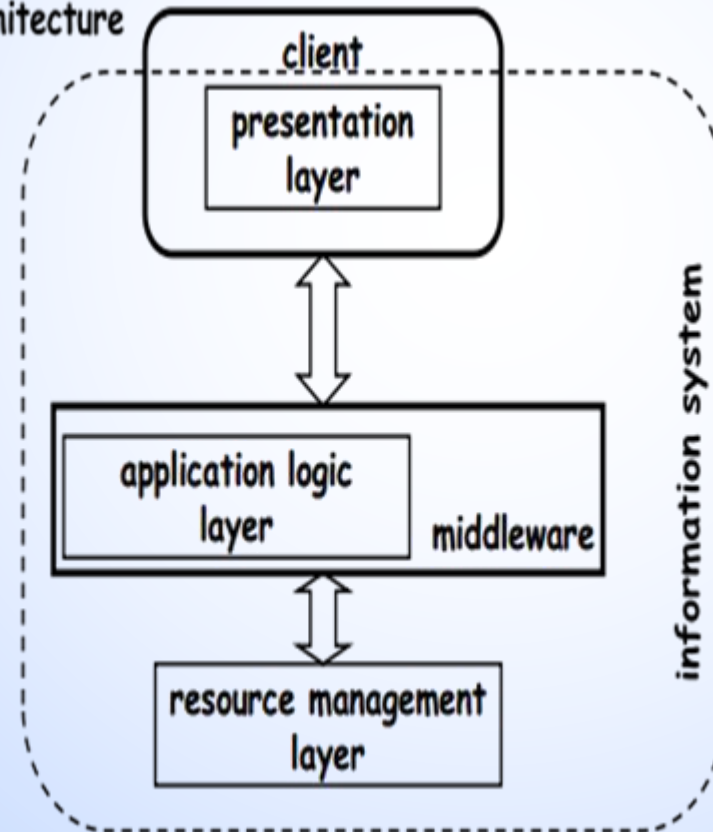
SVANTAGGI

- Client potenti
- Gestione complessa

Architettura 3 tier

- tre strati funzionali sono separati su tre diversi livelli di deployment

3-tier architecture



😊 **Vantaggi**

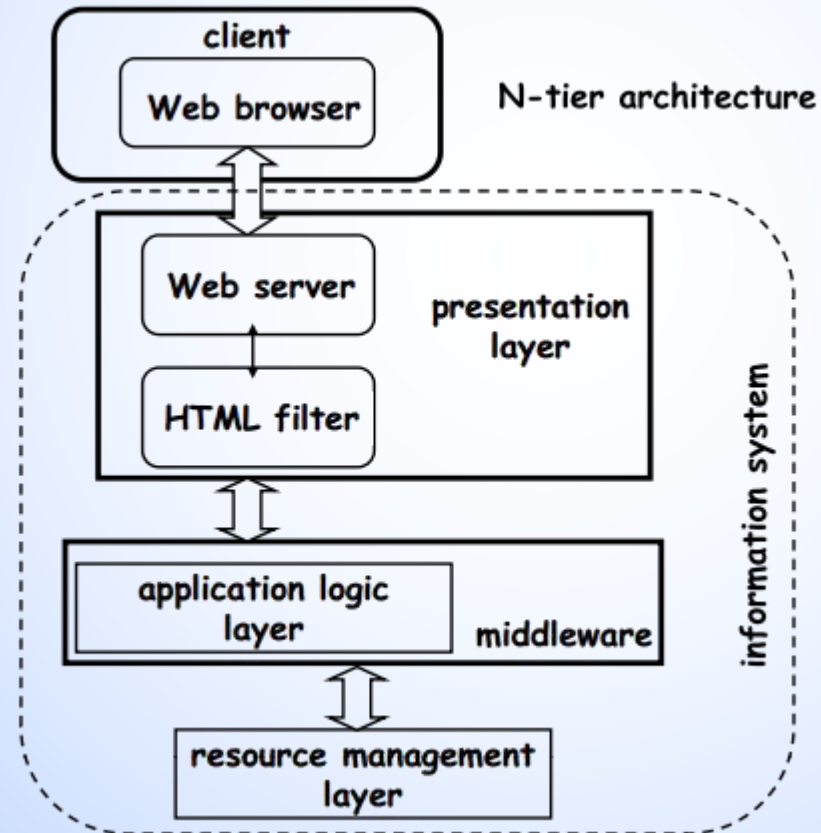
Migliori prestazioni
Scalabile
Affidabile
Gestione semplice

😞 **Svantaggi**

Maggiore overhead
Maggior complessità
Latenza del Servizio

Architettura n-tier

- generalizzazione del modello client/server a tre livelli ad un numero qualunque di livelli e server intermedi



Vantaggi e Svantaggi comuni



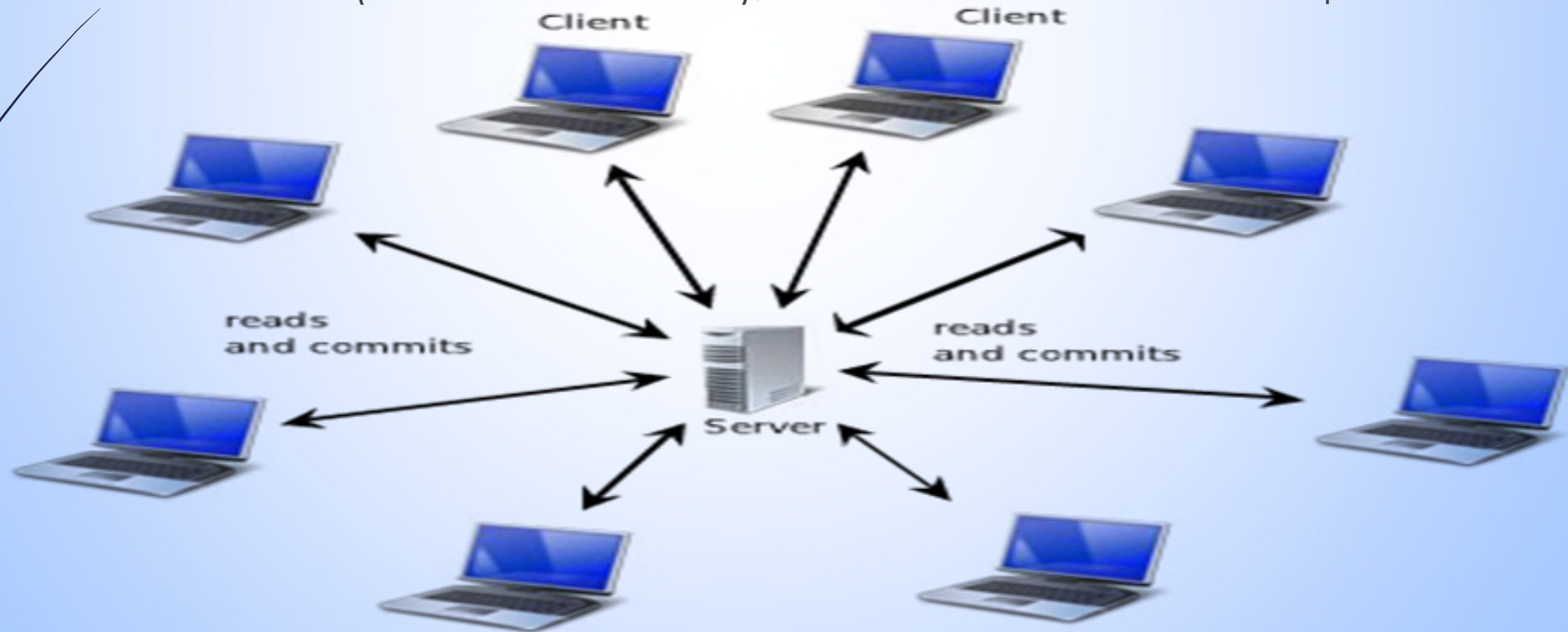
- Flessibilità
- Funzionalità
- Possibilità distribuzione

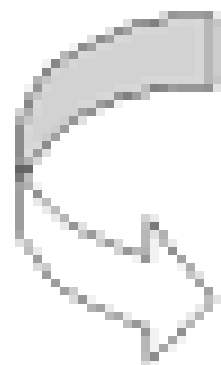


- Prestazioni
- Complessità

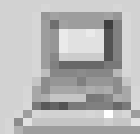
Funzionamento

- ▶ un processore (client) richiede l'accesso ad informazioni dislocati su un altro calcolatore, e dunque assume il ruolo di "cliente";
- ▶ L'altro processore (server) risponde alla richiesta ricevuta inoltrando le informazioni in questione, e dunque assume il ruolo di "servente";
- ▶ in pratica, una volta che il client ha ottenuto l'accesso attraverso la fase di handshake (lett. stretta di mano), il server si mette a "totale" disposizione.



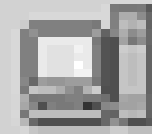


UTENTE



CLIENT

invio richiesta



SERVER



ELABORAZIONE
RICHIESTA



SERVER

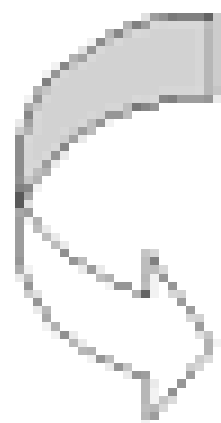
ricezione risposta



CLIENT



UTENTE

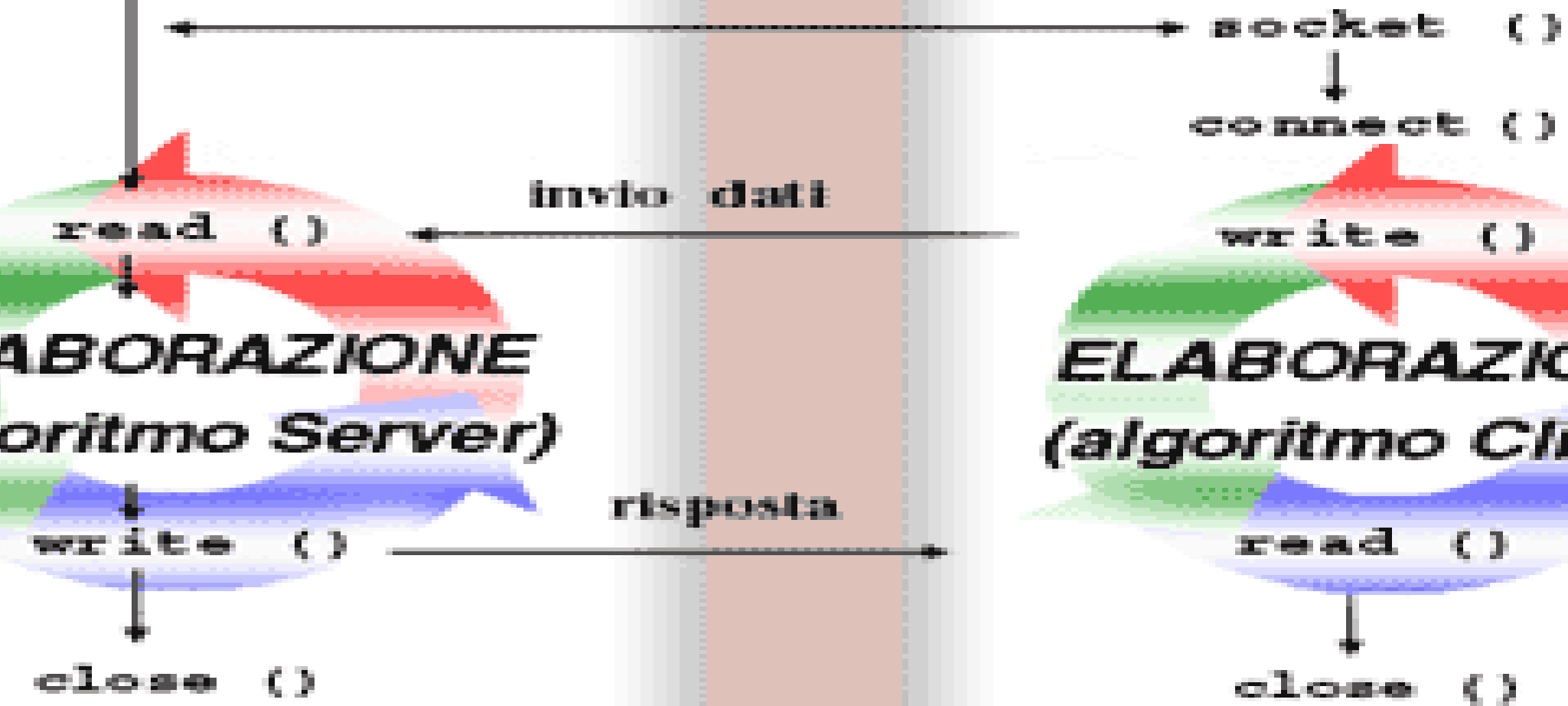


```
socket ()
↓
bind ()
↓
listen ()
↓
accept ()
```

Server

rete

Client



SERVER

```
ServerSocket ss=new ServerSocket(port);  
Socket s=ss.accept();  
s.getInputStream();  
s.getOutputStream();
```

I

O

Elaborazione

CLIENT

```
Socket s=new Socket(ind,port);  
s.getInputStream();  
s.getOutputStream();
```

I

O

