

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***

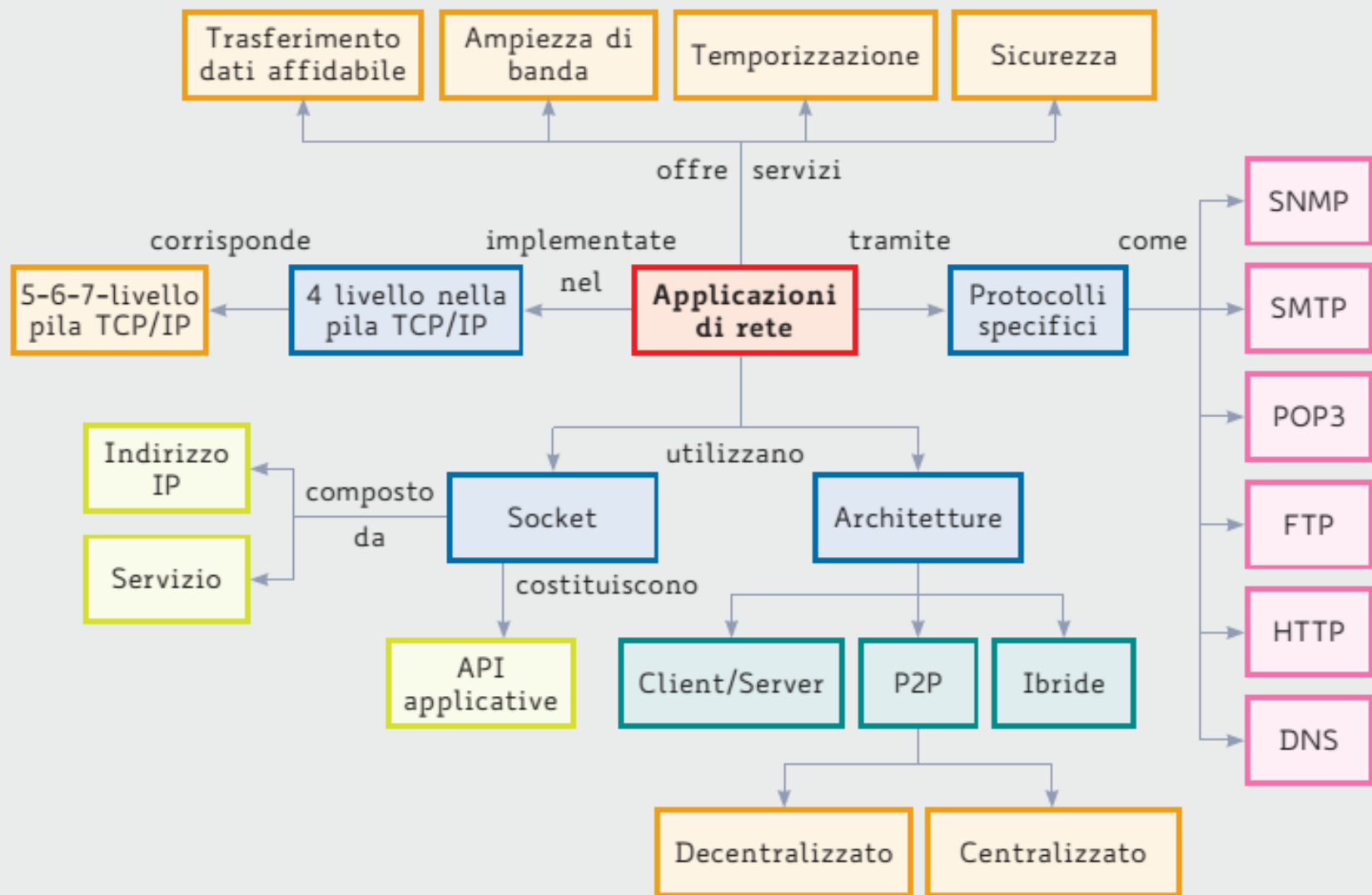
## ***Lezione 4***

**Le applicazioni  
di rete**

# **In questa lezione impareremo:**

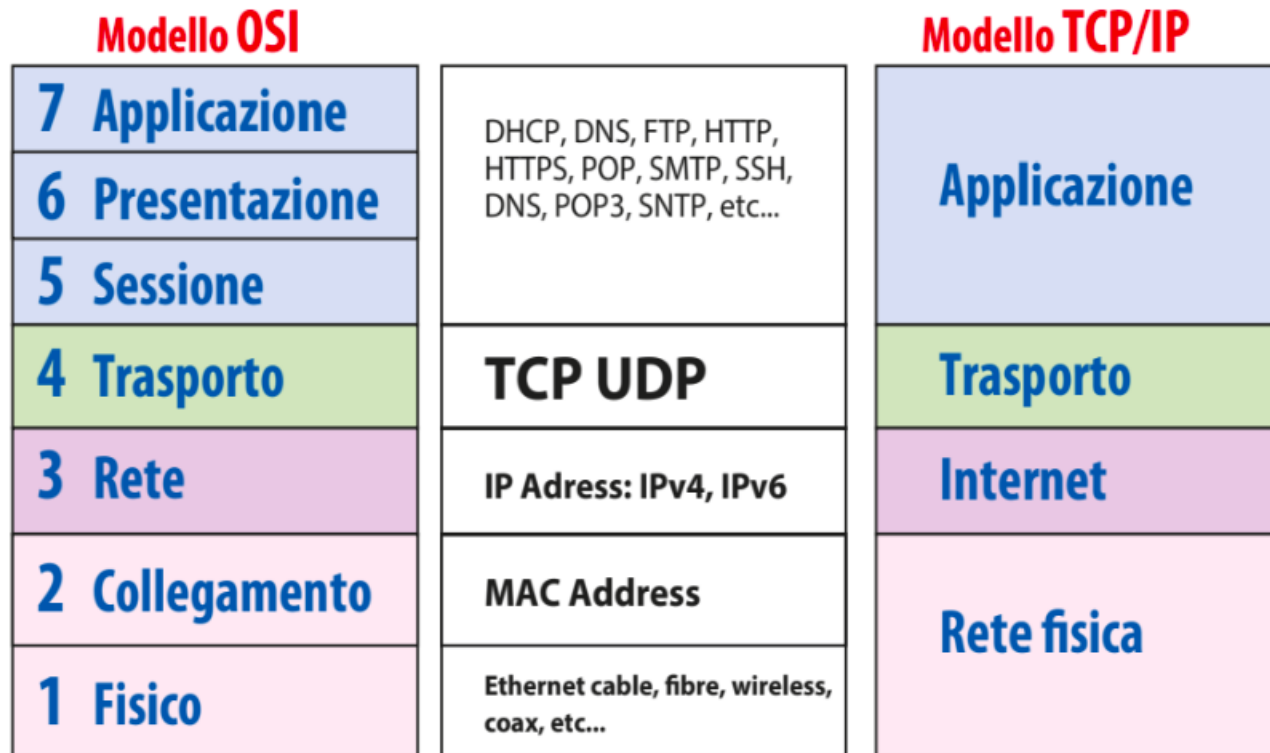
---

- **il concetto di applicazione di rete**
- **le tipologie di applicazione**
- **a scegliere i protocolli per le applicazioni di rete**



# Il modello ISO/OSI e le applicazioni

- Nel modello **ISO/OSI** e **TCP/IP** il livello delle applicazioni comprende i protocolli utilizzati dalle applicazioni di rete che vengono utilizzate dall'utente finale



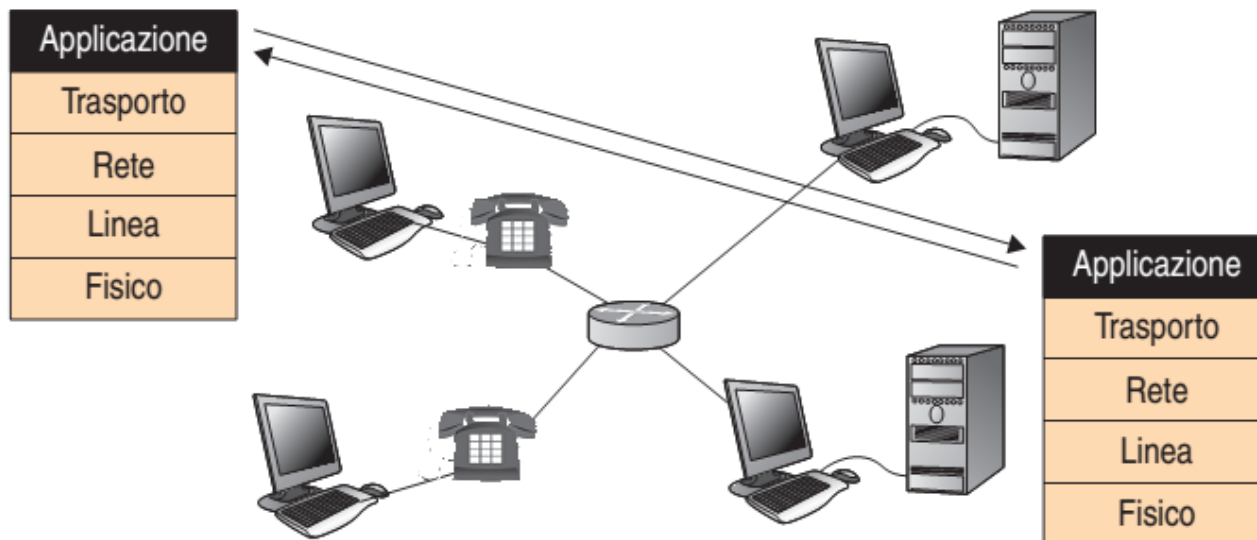
# Il modello ISO/OSI e le applicazioni

---

- Il livello **applicazione** mette a disposizione vari protocolli utilizzati da tutte le applicazioni di rete **generali** (posta elettronica, condivisione di file, web, ecc.), tra cui:
  - **SNMP**, Simple Network Management Protocol
  - **SMTP**, Simple Mail Transfer Protocol
  - **POP3**, Post Office Protocol
  - **FTP**, File Transfer Protocol
  - **HTTP**, HyperText Transfer Protocol
  - **DNS**, Domain Name System
- Esistono anche applicazioni di rete **proprietarie** che possono usare anche protocolli proprietari

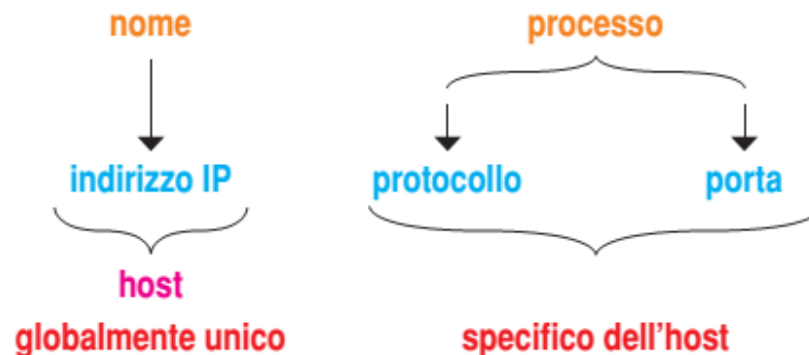
# Applicazioni di rete

- L'applicazione di rete prende anche il nome di **applicazione distribuita**, ovvero i processi hanno la necessità di scambiare informazioni con altri processi della medesima applicazione



# Identificazione mediante socket

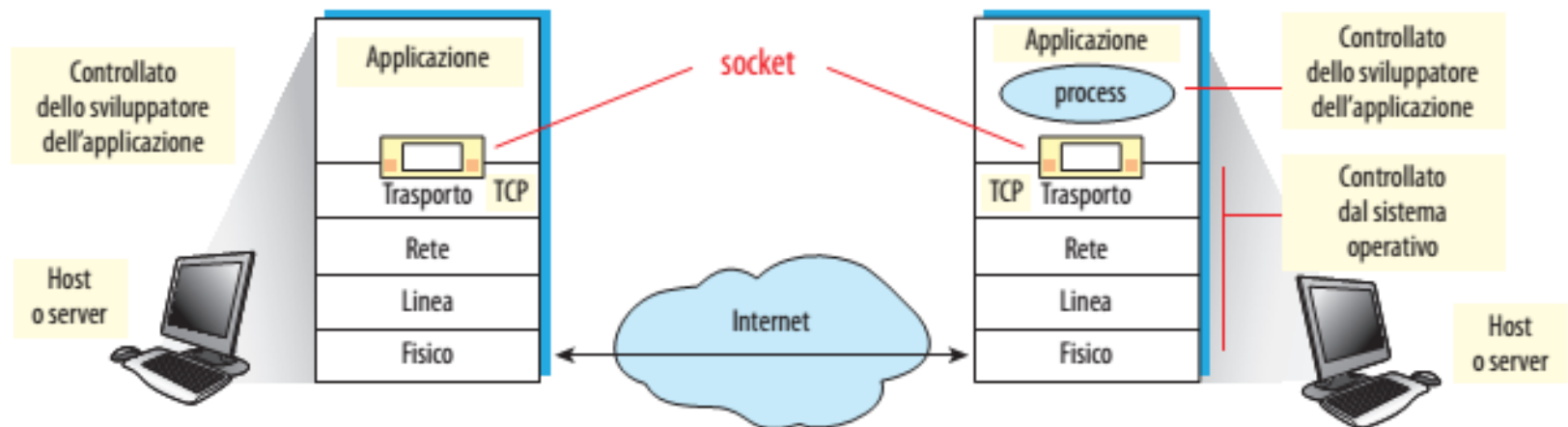
- Il processo mittente deve poter **identificare** il processo destinatario in modo univoco per potergli inviare un messaggio
- L'identificazione deve tenere conto di due informazioni:
  - **dove** si trova l'host, tramite **l'indirizzo IP**
  - con quale **processo** di quel **determinato host** comunicare, tramite il **numero di porta** assegnatogli univocamente





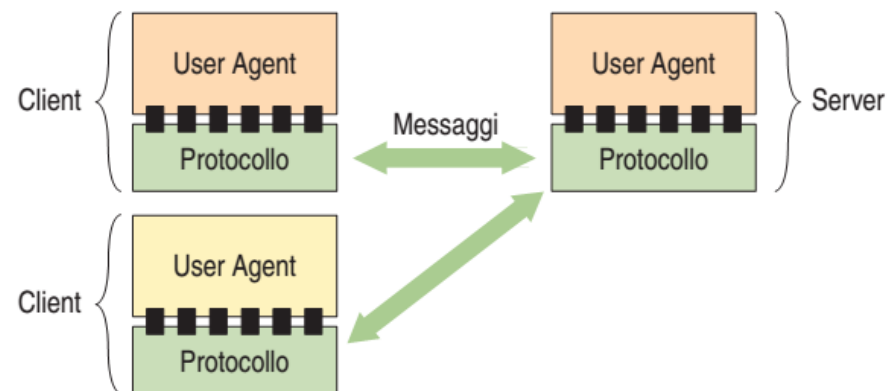
# Identificazione mediante socket

- L'identificazione univoca avviene **conoscendo a priori** sia l'indirizzo IP che il numero di porta associato al processo in esecuzione su un host
- Questo meccanismo prende il nome di **socket**, l'interfaccia di comunicazione di livello trasporto tra mittente e destinatario



# Identificazione mediante socket

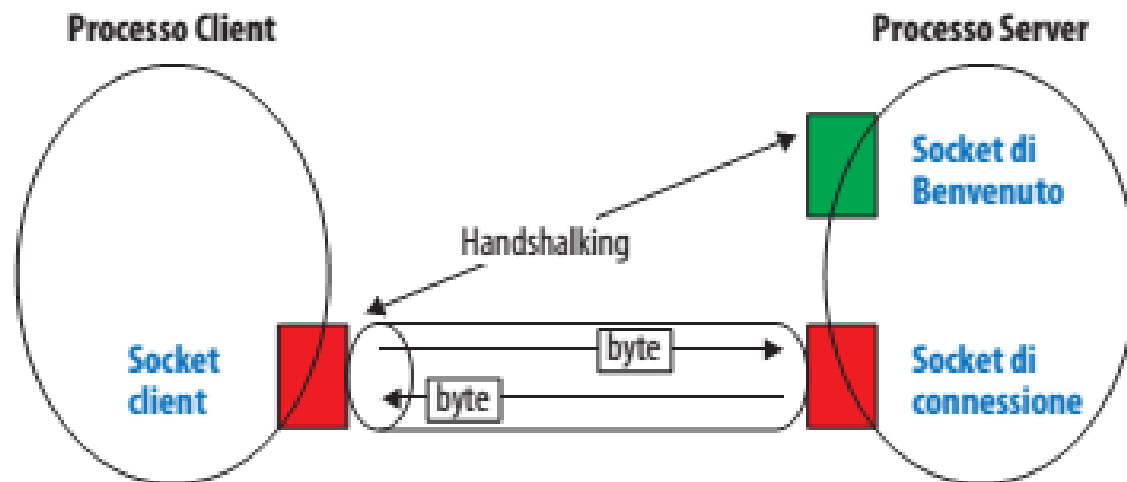
- L'applicazione di rete può essere vista come composta da due parti:
  - una **user agent**, che funge da **interfaccia** tra l'utilizzatore dell'applicazione e gli aspetti comunicativi
  - il **protocollo**, che permette di regolamentare la comunicazione



- In un browser Web che componenti rappresentano queste due parti?

# Identificazione mediante socket

- Il server offre uno o più servizi a più client e, quindi, deve gestire la concorrenza:
  - sul server è presente un **socket di connessione o di benvenuto** che rimane in ascolto e accetta le richieste dei client
  - crea dinamicamente un nuovo **thread** a cui gli assegna un **socket di connessione**



# Scelta dell'architettura per l'applicazione di rete

---

- Il primo passo che il programmatore deve effettuare per progettare una applicazione di rete è la **scelta** della architettura dell'applicazione
- Architetture attualmente utilizzate:
  - client-server
  - peer-to-peer (P2P)
  - architetture ibride

# Architettura client-server

---

- Nella architettura **client-server** dovendoci essere sempre un server attivo che offre un servizio, che tipo di indirizzo IP avrà?
- Invece i **client** generalmente che tipo di indirizzo IP hanno?
- Un client è in grado di comunicare direttamente con gli altri client?

# Scelta dell'architettura per l'applicazione di rete

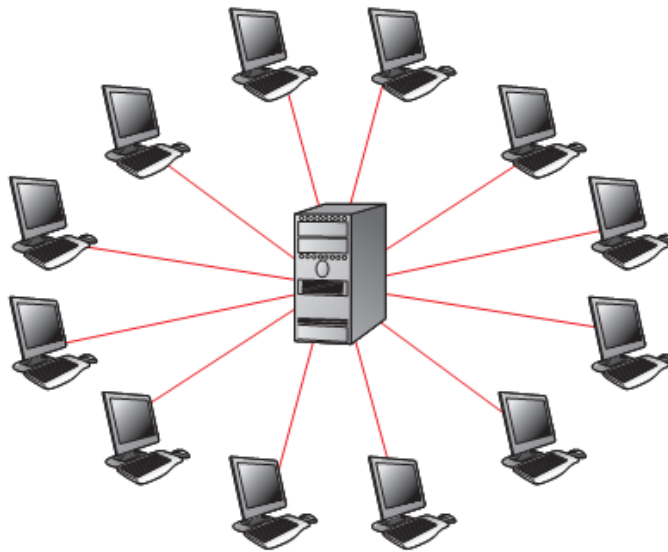
---

- Il server deve possedere un indirizzo IP fisso, statico per poter essere raggiunto dai client (ricordate? il mittente deve sapere a priori...)
- Quello dei client generalmente è dinamico, per motivi di costi il provider ce ne assegna uno ad ogni sessione
- Un client comunica direttamente solo coi server

# Scelta dell'architettura per l'applicazione di rete

---

- Un tipico esempio di questa architettura è il **WWW**



- Come si potrebbero risolvere eventuali problemi di **congestione** sul server a causa di troppe richieste ricevute?

# Architettura peer-to-peer (P2P)

---

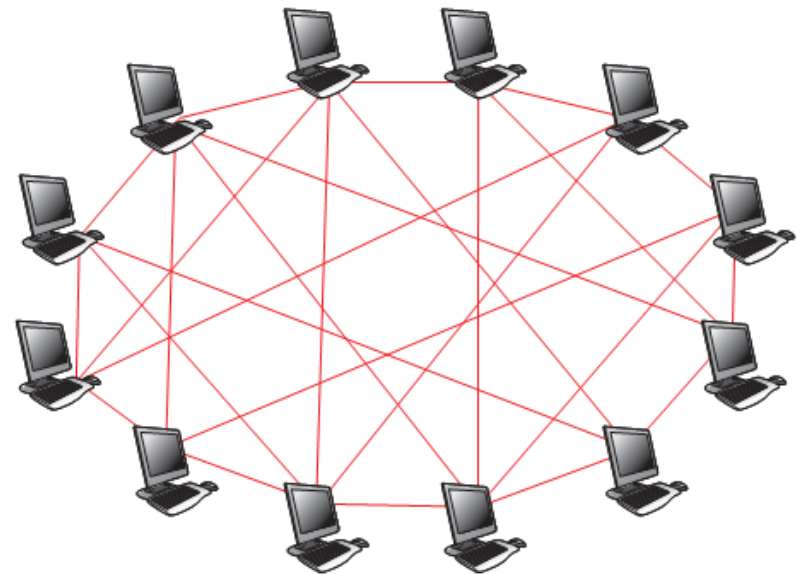
- Un **sistema P2P** è formato da un insieme di entità autonome (**peers**), capaci di auto organizzarsi, che dialogano direttamente e condividono un insieme di risorse distribuite presenti all'interno di una rete
- Il sistema utilizza tali risorse per fornire una determinata funzionalità in modo completamente o parzialmente decentralizzato o centralizzato



# P2P decentralizzato

---

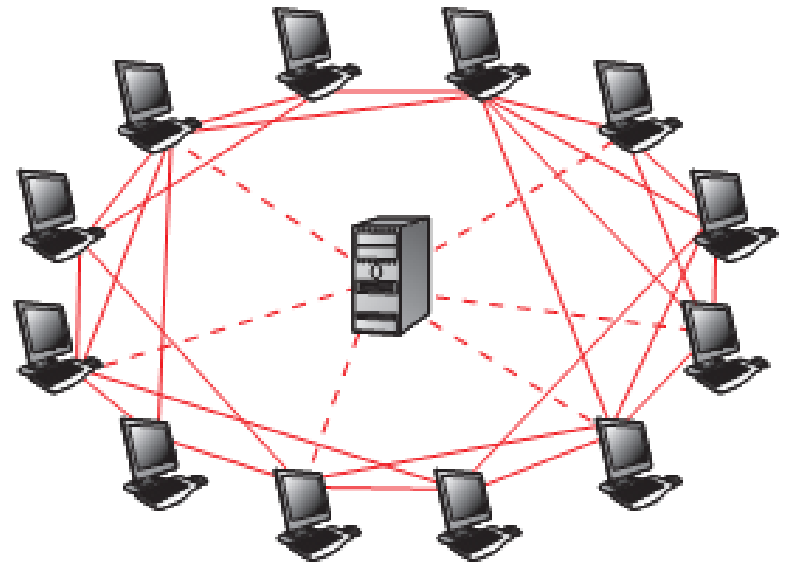
- Nella architettura completamente decentralizzata un peer ha sia la funzione di client che di server (**servent**)
- È impossibile localizzare una risorsa mediante un indirizzo IP statico, i partecipanti cambiano continuamente
- Il sistema si adatta senza che vi sia un gestore
- Si utilizza un meccanismo di passaparola per le richieste



# P2P centralizzato

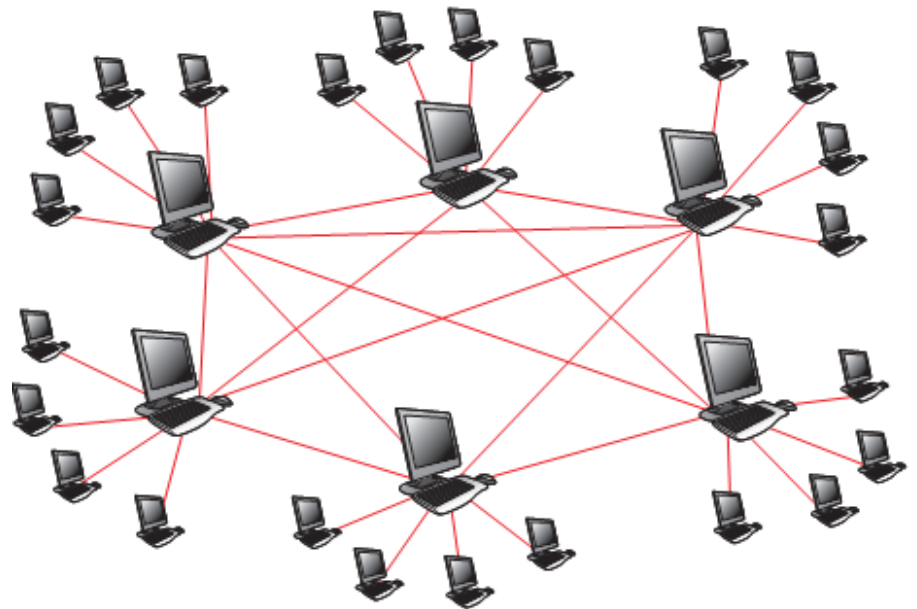
---

- Un server centrale conserva le informazioni sui peer per offrire una ricerca centralizzata
- I peer sono responsabili:
  - di conservare i dati e le informazioni (il server centrale non memorizza i file)
  - di informare il server sui file che intendono condividere
  - di permettere ai peer che lo richiedono di scaricare le risorse condivise



# P2P ibrido (o parzialmente centralizzato)

- Il **P2P ibrido** è un **P2P parzialmente centralizzato** dove sono presenti alcuni peer (detti supernodi o super-peer o ultra-peer) che hanno anche la funzione di indicizzazione
- I super-peer vengono determinati dinamicamente tramite elezione



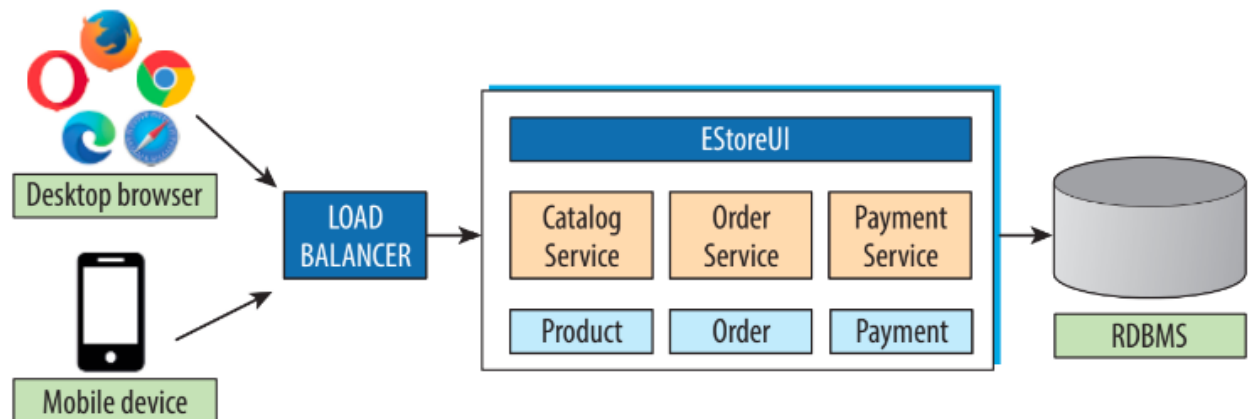
# Applicazioni per il mobile computing

---

- Le applicazioni mobile sono specificatamente progettate per dispositivi mobili e per essere utilizzate in movimento
- Hanno una interfaccia utente specifica
- Utilizzano il touch screen, lo swipe, la localizzazione, ecc.
- Possono essere sviluppate per una piattaforma specifica (native)
- Esistono anche applicazioni ibride che sfruttano i dati raccolti del dispositivo e li condividono con un server per elaborarli

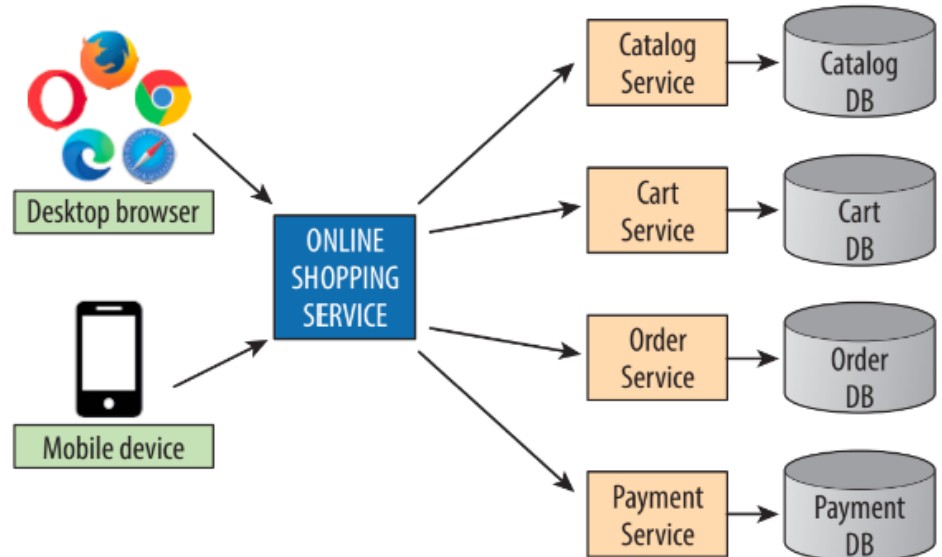
# Applicazioni web monolitiche e a microservizi

- Monolitica se viene sviluppata e distribuita come una **singola entità**
- I suoi componenti fanno parte di un **singolo programma** su una **singola piattaforma**
- L'applicativo implementa tutte le attività
- Ha tempi di sviluppo minori, ma è poco scalabile e dunque potrebbe richiedere più sforzi per l'adeguamento
- È più difficile la manutenzione del codice



# Applicazioni web monolitiche e a microservizi

- A microservizi se viene separata in componenti piccoli e indipendenti
- Ha un maggior grado di flessibilità e di scalabilità
- È più facile la manutenzione
- Ogni microservizio è sviluppato autonomamente
- Collaborano tra loro per offrire la funzionalità dell'applicazione nel suo complesso tramite API e protocolli
- Ogni microservizio può essere sviluppato con un linguaggio di programmazione diverso



# Applicazioni web monolitiche e a microservizi

---

- Caratteristiche principali dell'architettura a microservizi:
  - **Decomposizione dell'applicazione**, viene suddivisa in componenti autonome
  - **Scalabilità**, scalare e distribuire in modo indipendente
  - **Agilità nello sviluppo**, sviluppo, test e rilascio autonomo
  - **Flessibilità tecnologica**, uso di tecnologie specifiche e linguaggi differenti

# Applicazioni web monolitiche e a microservizi

---

- **Adattabilità al cambiamento**, seguire la naturale evoluzione tecnologica mantenendo meccanismi di interfacciamento
- **Individuazione e isolamento dei difetti**, la presenza di un difetto può essere circoscritta
- **Collaborazione tra team**, differenti team di sviluppo per ognuno di essi
- **Dati decentralizzati**, possibilità di avere un archivio dedicato



# Servizi offerti dallo strato di trasporto alle applicazioni

---

- Ogni applicazione deve scegliere tra i protocolli di trasporto quale adottare per realizzare un protocollo applicativo
- La scelta viene effettuata in base alle specifiche esigenze della applicazione in termini di servizi:
  - trasferimento dati affidabile, consegna completa e corretta dei dati (es. posta elettronica)
  - ampiezza di banda, larghezza di banda minima disponibile (es. Web-TV)
  - temporizzazione, piccoli ritardi (es. telefonia VoIP)
  - sicurezza, cifratura dei dati (es. applicazioni bancarie)

## SCELTA MULTIPLA



## AREA DIGITALE



Esercizi per il recupero  
e l'approfondimento

**1 Quale tra i seguenti non è un protocollo applicativo?**

- a HTTP
- b POP3
- c DNS
- d SMNP
- e SMTP
- f FTP

**2 Quale tra le seguenti non è una applicazione di rete?**

- a Posta elettronica
- b Condivisione di file P2P
- c Scheduler dei processi
- d Telefonia via Internet
- e Videoconferenza in tempo reale
- f TV in streaming
- g Telnet

**3 Che cosa significa SMTP?**

- a Simple Message Transfer Protocol
- b System Mail Transfer Protocol
- c Simple Mail Transfer Protocol
- d System Message Transfer Protocol

**4 Qual è il significato di API?**

- a Application Protocol Internet
- b Application Protocol Interface
- c Application Programming Interface
- d Application Programming Internet

**5 Quali tra le seguenti non sono architetture di una applicazione di rete?**

- a client-server
- b B2B
- c B2C
- d P2P

**6 Quale tra i seguenti servizi offerti non è garantito dallo strato di trasporto alle applicazioni?**

- a trasferimento dati affidabile
- b ampiezza di banda
- c velocità di comunicazione
- d temporizzazione sicurezza
- e sicurezza

## VERO/FALSO



- 1 Il protocollo HTTP è a livello di trasporto.
- 2 Il protocollo TCP è a livello di applicazione.
- 3 L'applicazione di rete prende anche il nome di applicazione distribuita.
- 4 L'applicazione di rete prende anche il nome di applicazione multiutente.
- 5 Sullo stesso host possono essere in esecuzione molti processi.
- 6 Sullo stesso host può essere in esecuzione una sola applicazione.
- 7 Un protocollo è una parte integrante di una applicazione ed è sviluppato all'interno di essa.
- 8 In un'architettura client-server gli indirizzi IP devono essere statici.
- 9 Il termine server è ottenuto dalla contrazione server-client.
- 10 Nel P2P decentralizzato i server hanno indirizzo IP statico.

