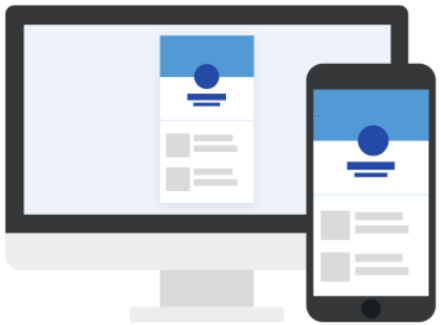
APPLICAZIONI WEB



APPLICAZIONE

- Un'applicazione è un software dedicato alla produttività dell'utente: scrivere, disegnare, programmare, etc
- Ha un'interfaccia utente e può essere composta da vari artefatti software: programmi eseguibili, librerie, script, file di configurazione e risorse necessarie al suo funzionamento
- Le applicazioni si possono suddividere in:
  - ***Applicazioni standalone***
  - ***Applicazioni distribuite***

APPLICAZIONI STANDALONE

- Un'***applicazione standalone*** (autonoma) non necessita di risorse remote per il proprio funzionamento, dunque non richiede una connessione di rete.

  (L'applicazione può comunque usare la rete per vari scopi.)

- Tutti gli artefatti software necessari al suo funzionamento sono presenti nel sistema dove è installata

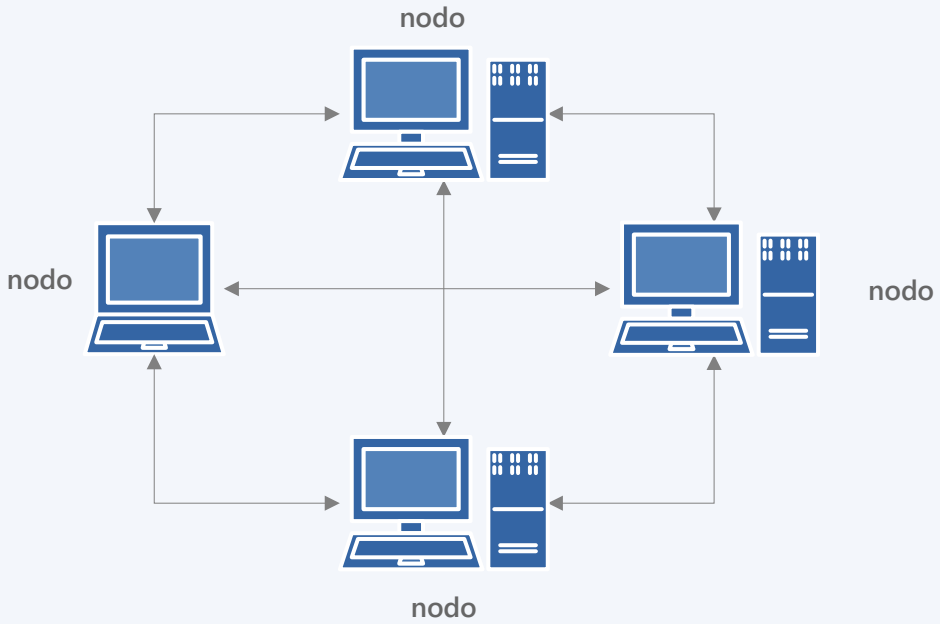
APPLICAZIONI DISTRIBUITE

- Un'***applicazione distribuita***, o *applicazione di rete*, è un software in esecuzione su più computer
- Gli artefatti software sono eseguiti come processi ospitati in sistemi distinti e comunicano tra loro utilizzando un *protocollo di rete*
- Le applicazioni distribuite si distinguono per la loro architettura:
  - ***Applicazioni peer-to-peer***
  - ***Applicazioni client-server***

APPLICAZIONI PEER-TO-PEER

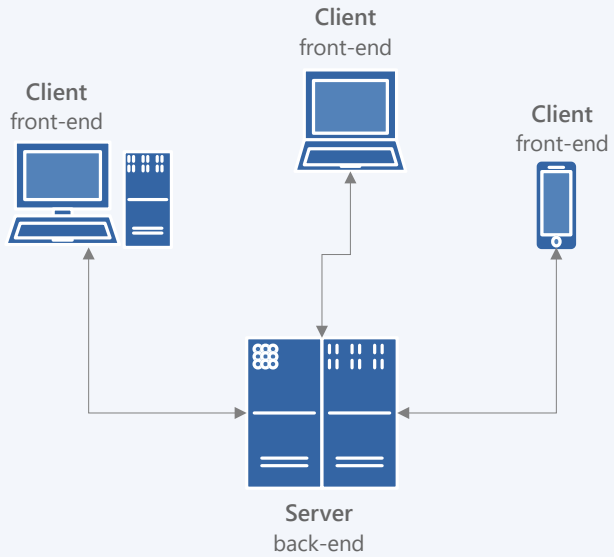
- Le funzioni dell'applicazione sono replicate in più computer. Ogni computer è definito ***nodo dell'applicazione***, esegue lo stesso software e può comunicare con tutti gli altri nodi

- Non esiste un ***nodo dell'applicazione*** speciale rispetto agli altri. Ogni nodo fornisce dei servizi agli altri nodi e "consuma" i servizi degli altri nodi



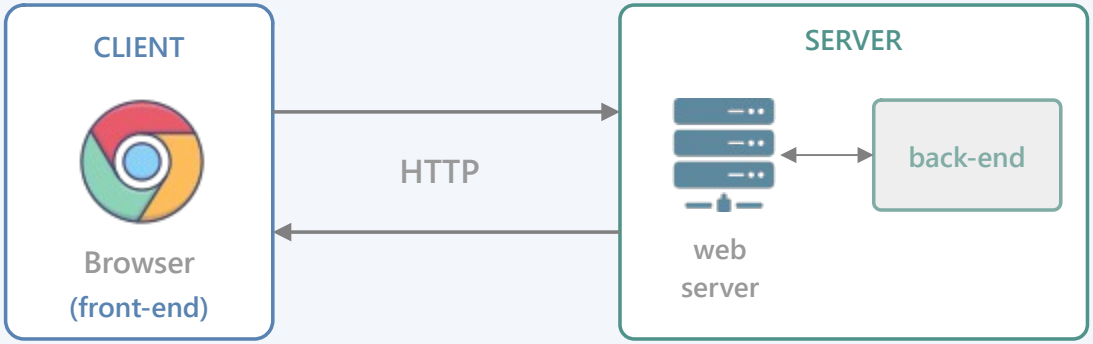
APPLICAZIONI CLIENT-SERVER

- Nelle architetture ***client-server***, funzioni distinte dell'applicazione vengono eseguite su computer distinti.
- Il termine *client-server* si riferisce all'esistenza di due funzioni principali:
  - ***Client***: implementa l'interfaccia utente e dunque viene usata dall'utente finale. Gestisce il cosiddetto ***front-end*** dell'applicazione.
  - ***Server***: fornisce i servizi ai client. Rappresenta il cosiddetto ***back-end*** dell'applicazione.



APPLICAZIONI WEB

- Le *applicazioni web* sono:
  - *Applicazioni client-server,*
  - che usano il protocollo HTTP per la comunicazione di rete,
  - un *web browser* come *client*,
  - il linguaggio HTML per l'interfaccia utente
- Il *back-end* dell'applicazione è collegato a un software chiamato *web-server*



ARCHITETTURA DELLE APPLICAZIONI WEB

- Le *applicazioni web* si distinguono per l'architettura e, dunque, per l'implementazione del *front-end* e del *back-end*:
  - Siti web statici
  - ***Server-side rendering*** (*front-end* implementato nel *server*)
  - ***Client-side rendering*** (*front-end* implementato nel *client*)
  - Architetture miste

SITI WEB STATICI

- I *siti web statici* implementano un servizio documentale che consente la navigazione tra i contenuti, memorizzati in pagine HTML
- La parola “statico” indica che i contenuti inviati al client non vengono elaborati; sono memorizzati in file e sono sempre gli stessi
- I *siti web statici* non rappresentano delle vere e proprie applicazioni; non devono, cioè, essere programmati. *Web browser* e *web server* forniscono già tutte le funzioni richiese

SERVER-SIDE RENDERING (SSR)

- Con questa architettura, nel *server* vengono stabiliti sia contenuti, sia la loro modalità di visualizzazione:

  il codice HTML, comprensivo dei contenuti, viene generato sul *server* e successivamente spedito al *client*. Il browser ha un ruolo passivo: riceve il contenuto HTML e lo visualizza

- L'interattività è molto limitata, poiché le azioni dell'utente vengono elaborate sul server:

  si riduce alla selezione (o digitazione) di un *hyperlink*, oppure all'invio di un form HTML

CLIENT-SIDE RENDERING (CSR)

- In questa architettura, l'interfaccia utente viene programmata per essere eseguita sul *client*
- È possibile avere un alto livello di interattività, poiché le azioni dell'utente vengono gestite dal codice in esecuzione nel browser
- Queste applicazioni vengono definite anche RIA (*Rich Internet Application*), per via della loro interfaccia utente ricca di funzioni

PROTOCOLLO HTTP

- Il protocollo HTTP si appoggia al *protocollo di trasporto* TCP e si basa su una comunicazione che è di tipo ***richiesta→risposta***:

  il *client* apre una connessione TCP, invia al *server* una *richiesta*, legge la *risposta* del *server* e chiude la connessione

- Il protocollo usa prevalentemente il formato testo ed è *case insensitive*

RICHIESTE HTTP

- Il protocollo HTTP prevede vari tipi di richiesta, ognuna delle quali è identificata da una parola chiave definita *metodo*: GET, POST, PUT, etc.
- Ogni richiesta è strutturata in tre parti, l'ultima delle quali è preceduta da una riga vuota:
  - ***Start line***: definisce il *metodo* e l'URL della risorsa richiesta
  - ***Headers***: elenco di coppie chiave-valore che memorizzano delle informazioni sulla richiesta
  - ***Corpo***: è opzionale; se presente memorizza dei dati che il *client* invia al *server*

ESEMPIO DI UNA RICHIESTA HTTP

In figura è riportato il contenuto di una ipotetica richiesta della pagina **home.html** del sito **www.superstore.it**.

Start line	`GET /home.html HTTP/1.1`

Headers	`Host: www.superstore.it`
	`User-Agent: Mozilla/5.0`
	`Accept: text/html`

RISPOSTE HTTP

- Il protocollo HTTP prevede vari tipi di risposta, ognuna delle quali è identificata da uno ***status code***: 200, 300, etc.
- Ogni risposta è strutturata in tre parti, l'ultima delle quali è preceduta da una riga vuota:
  - ***Status line***: definisce lo *status code* e un breve testo informativo sulla risposta
  - ***Headers***: elenco di coppie chiave-valore che memorizzano delle informazioni sulla risposta
  - ***Corpo***: è opzionale; se presente memorizza dei dati che il *server* invia al *client*

ESEMPIO DI UNA RISPOSTA HTTP

In figura è riportato il contenuto di una ipotetica risposta alla richiesta mostrata in precedenza:

Status line

Headers

Corpo

```
HTTP/1.1 200 OK
Date: Thu, 04 Jan 2024 10:31:44 GMT
Server: Apache
Last-Modified: Wed, 24 Mar 2021 18:03:39 GMT
ETag: "b7-5be4c1e598f4e"
Accept-Ranges: bytes
Content-Length: 137
Vary: Accept-Encoding
Connection: close
Content-Type: text/html

<html>
    <head>
        <link rel="stylesheet" href="stili.css">
    </head>
    <body>
        <h1>SUPERSTORE</h1>
    </body>
</html>
```