



UNIVERSITÀ  
DEGLI STUDI  
DI TERAMO



# Applicazioni Web: Architetture e Sviluppo

Prof.ssa Romina Eramo

Università degli Studi di Teramo

Dipartimento di Scienze della Comunicazione

[reram@unite.it](mailto:reram@unite.it)

# Applicazioni web statiche vs dinamiche

---

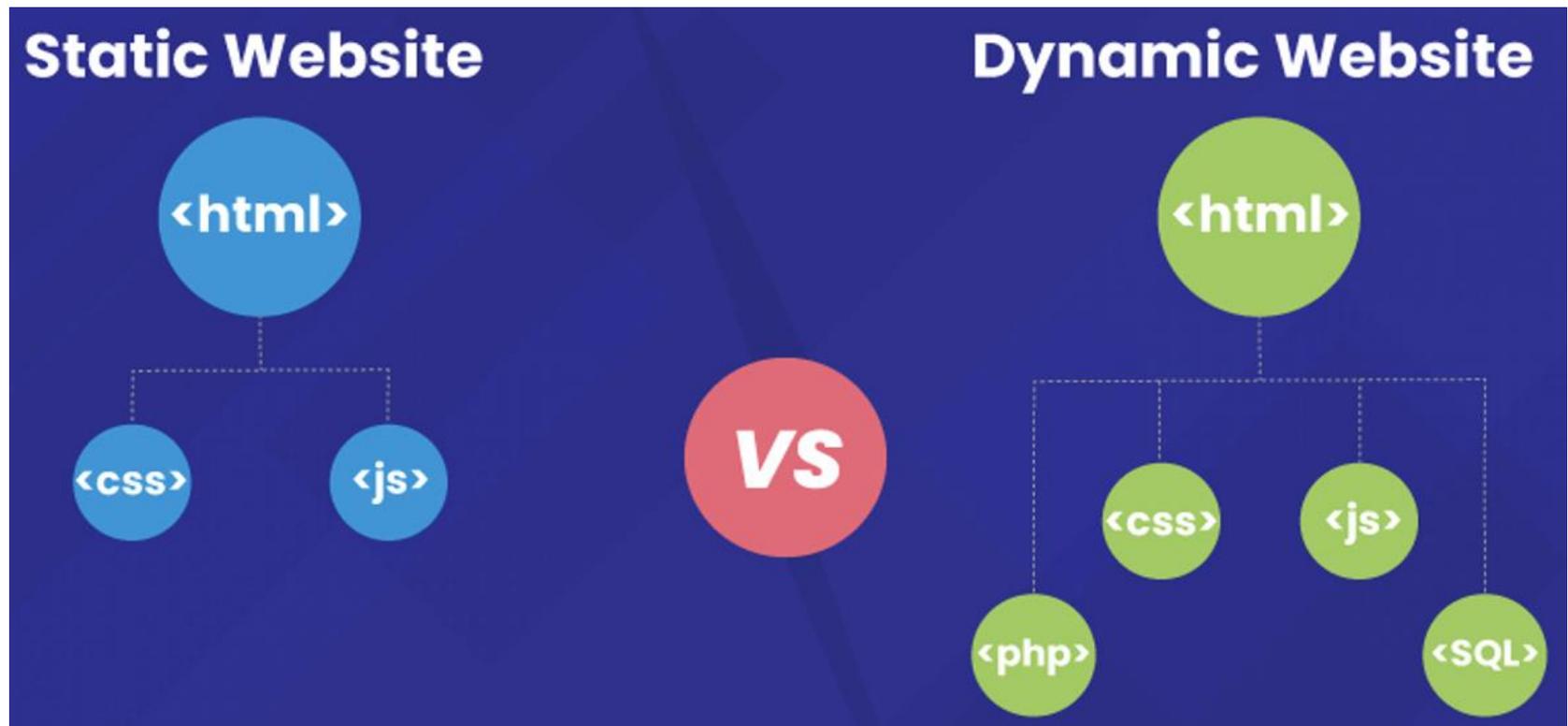
- » Quando la potenza di calcolo era un bene prezioso, le pagine web venivano spesso servite come file di testo immutabili. Questo tipo di contenuto statico è molto efficiente, ma può rapidamente diventare stagnante se non aggiornato.
- » Con l'avvento dell'informatica a basso costo e delle velocità Internet più elevate, gli sviluppatori hanno creato piattaforme e architetture lato server in grado di generare contenuti specifici per l'utente al volo. Questa forma di contenuto è nota come **contenuto dinamico**.
  - Il contenuto dinamico è una pagina web (componente) che cambia in base ai segnali dell'utente che includono il comportamento durante la sessione, i dati dell'utente e le caratteristiche dell'utente

# Applicazioni web statiche vs dinamiche

---

Static	Dynamic
<ul style="list-style-type: none"><li>• Prebuilt content is the same each time the page is loaded</li></ul>	<ul style="list-style-type: none"><li>• Content is generated "on-the-fly" and changes regularly</li></ul>
<ul style="list-style-type: none"><li>• Content only changes when someone updates and publishes the file (sends it to the web server)</li></ul>	<ul style="list-style-type: none"><li>• Page contains "server-side" code, allows the server to generate unique content when the page is loaded</li></ul>
<ul style="list-style-type: none"><li>• HTML code</li></ul>	<ul style="list-style-type: none"><li>• PHP, ASP, and JSP or other</li><li>• Can pull content from a database</li></ul>
<ul style="list-style-type: none"><li>• Example: About Us page with corporate background, mission, vision, etc.</li></ul>	<ul style="list-style-type: none"><li>• Example: upcoming events on a homepage pulling from a calendar and changing each day</li></ul>

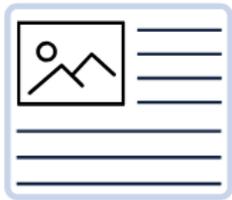
# Applicazioni web statiche vs dinamiche



# Applicazioni web statiche vs dinamiche

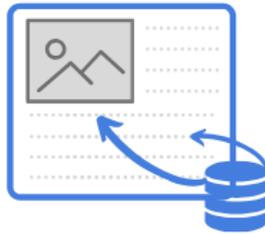
## CODE

### Static



Content is hard-coded on the page

### Dynamic



Dynamic references to content that are controlled externally with a CMS or database

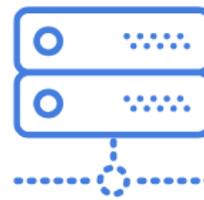
## DELIVERY

### Static



Deliver static code that is pre-rendered (usually via a Content Delivery Network)

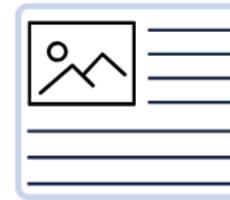
### Dynamic



Code is rendered in real time by the server

## CLIENT BROWSER

### Static



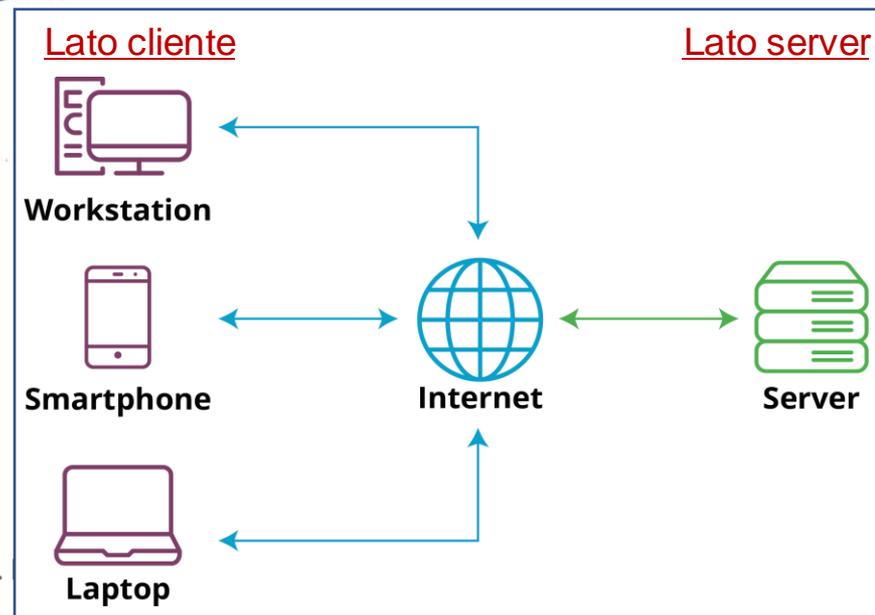
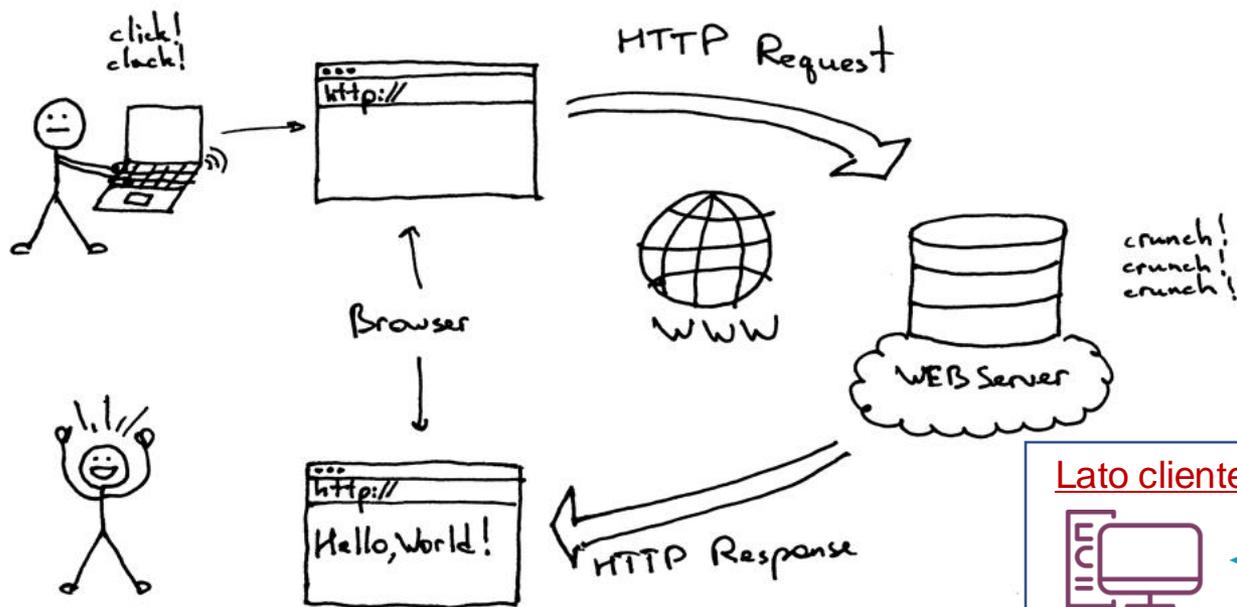
Page doesn't change; remains static for all who access it

### Dynamic

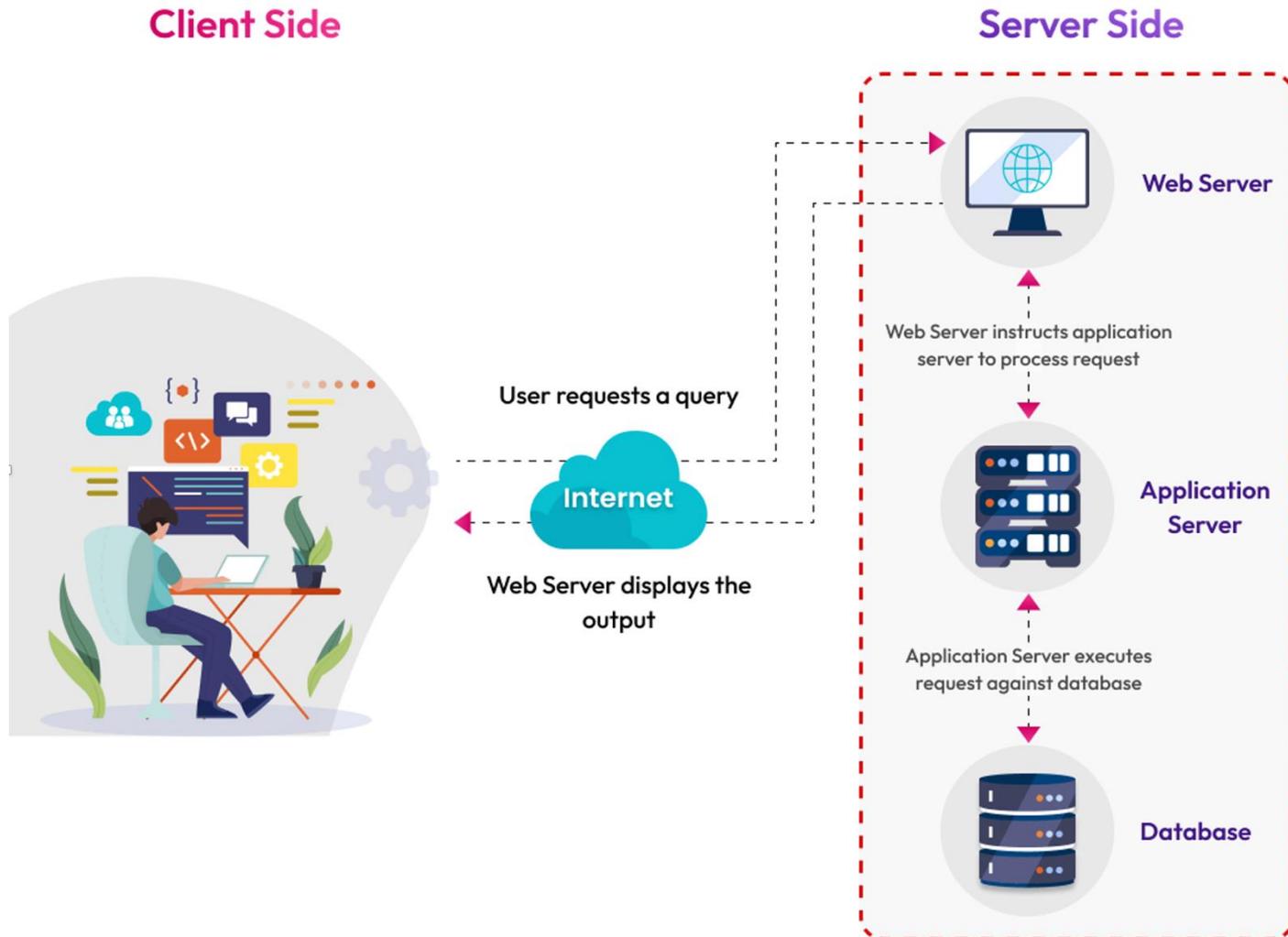


Use JavaScript to change page content, animation, etc. in realtime

# Architettura client-server (riepilogo)



# Architettura client-server (*riepilogo*)



# Significato di “livelli”

---

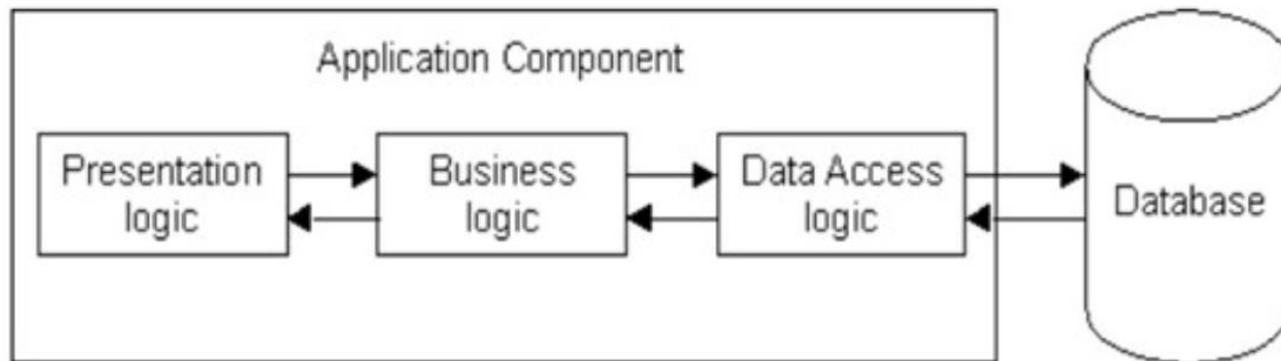
- » Le architetture N-tier hanno gli stessi componenti
  - Presentazione
  - Affari/Logica
  - Dati
- » Le architetture N-tier cercano di separare i componenti in diversi livelli/layer
  - Livello: separazione fisica
  - Livello: separazione logica

[https://en.wikipedia.org/wiki/Multitier\\_architecture](https://en.wikipedia.org/wiki/Multitier_architecture)

# Significato di “livelli”

---

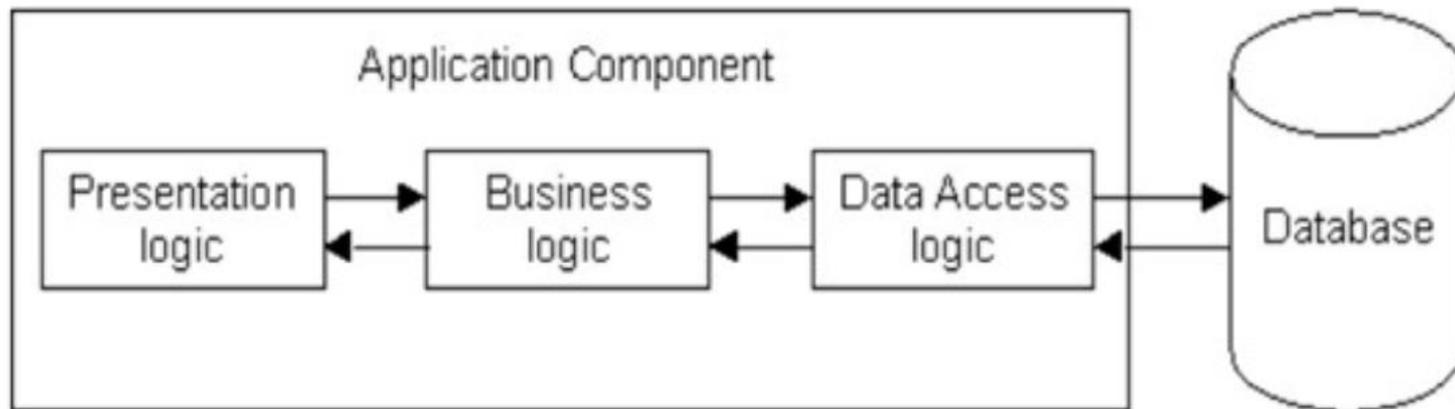
- » Il database viene eseguito sul server
  - Separato dal cliente
  - Facile passare a un database diverso
- » Livelli di presentazione e logica ancora strettamente connessi
  - Carico elevato sul server
  - Potenziale congestione sulla rete
  - Presentazione ancora legata alla logica aziendale



# Architettura a 1 livello

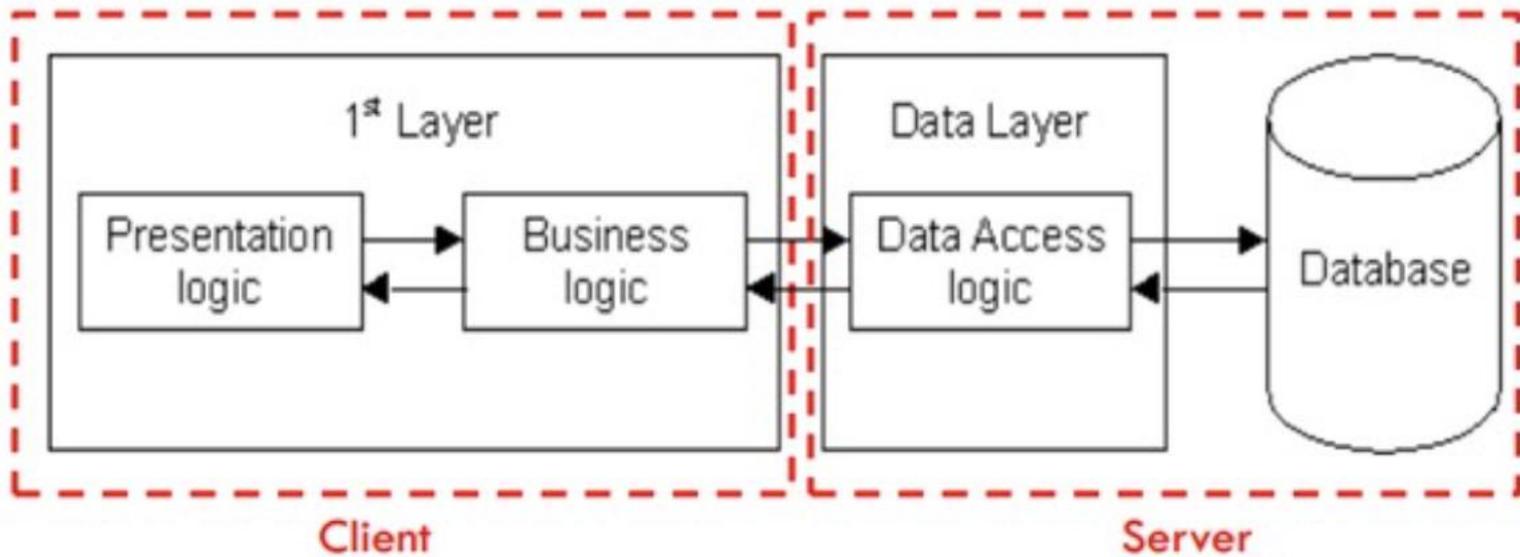
---

- » Tutti e 3 gli strati sono sulla stessa macchina
  - Tutto il codice e l'elaborazione conservati su un'unica macchina
- » Presentazione, logica, livelli dati sono strettamente connessi
  - S scalabilità, portabilità, manutenzione



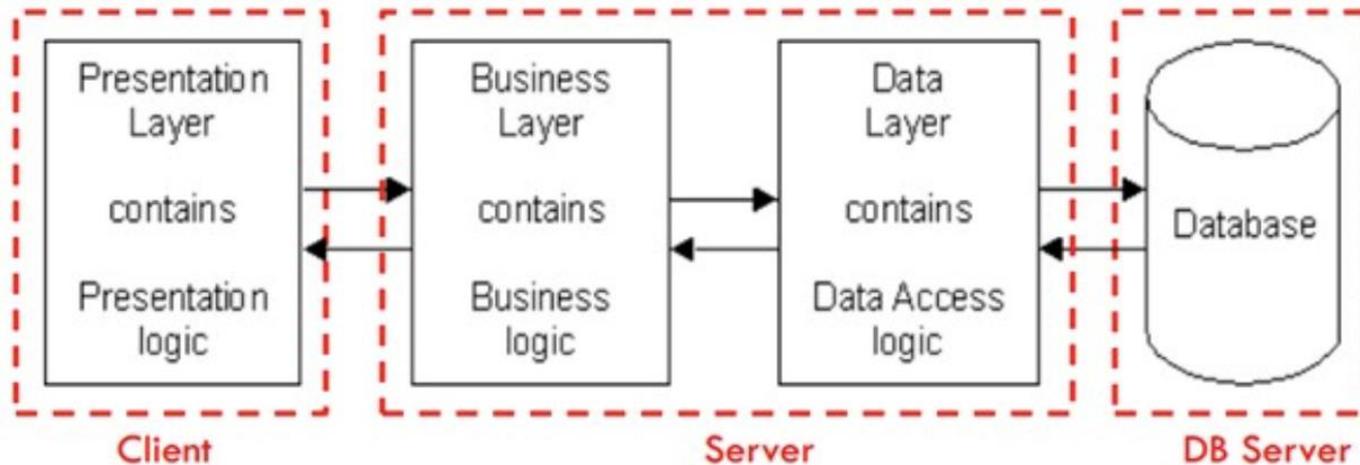
# Architettura a 2 livelli

- » Il database viene eseguito sul server
  - Separato dal cliente
  - Facile da passare a un database diverso
- » Livelli di presentazione e logica ancora strettamente connessi (accoppiati)



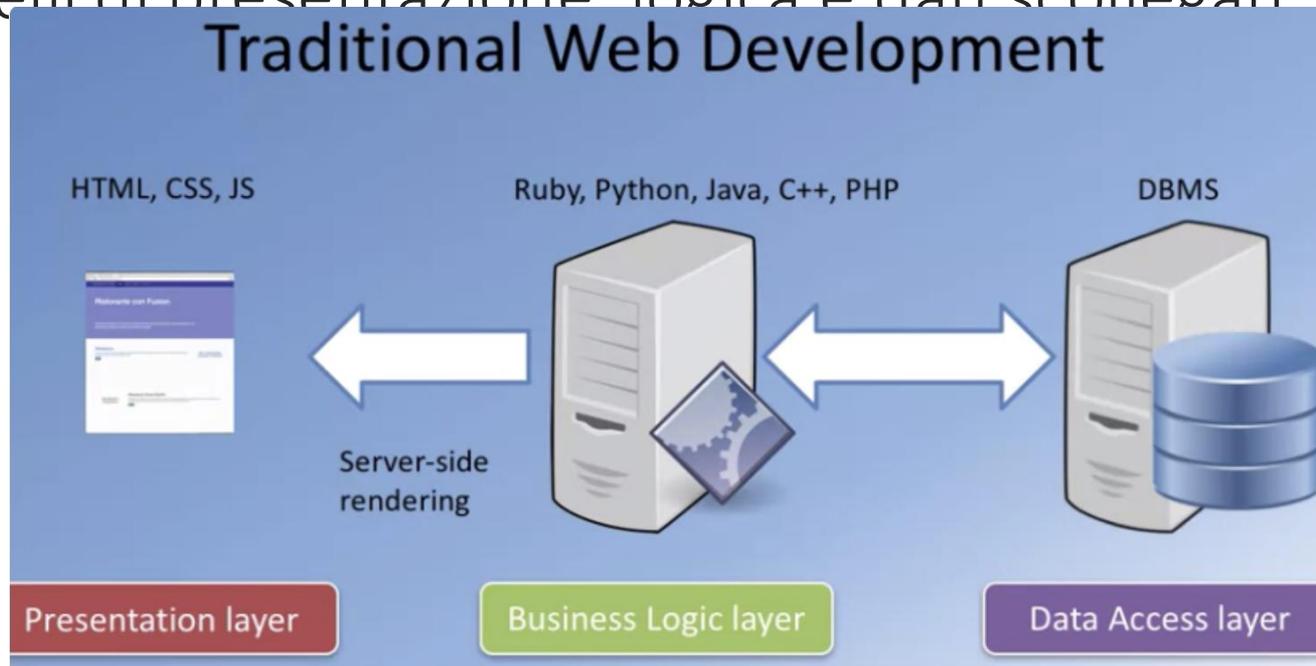
# Architettura a 3 livelli

- » Ogni livello può potenzialmente essere eseguito su una macchina diversa
- » Livelli di presentazione, logica e dati scollegati



# Architettura a 3 livelli

- » Ogni livello può potenzialmente essere eseguito su una macchina diversa
- » Livelli di presentazione, logica e dati scollegati



## Presentation tier

The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.



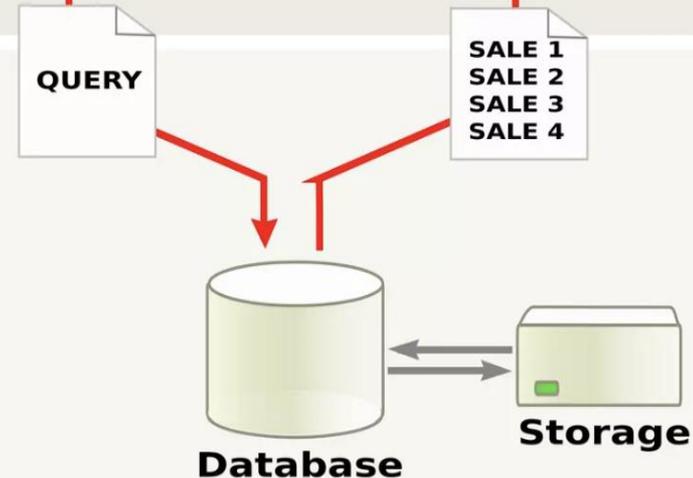
## Logic tier

This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.

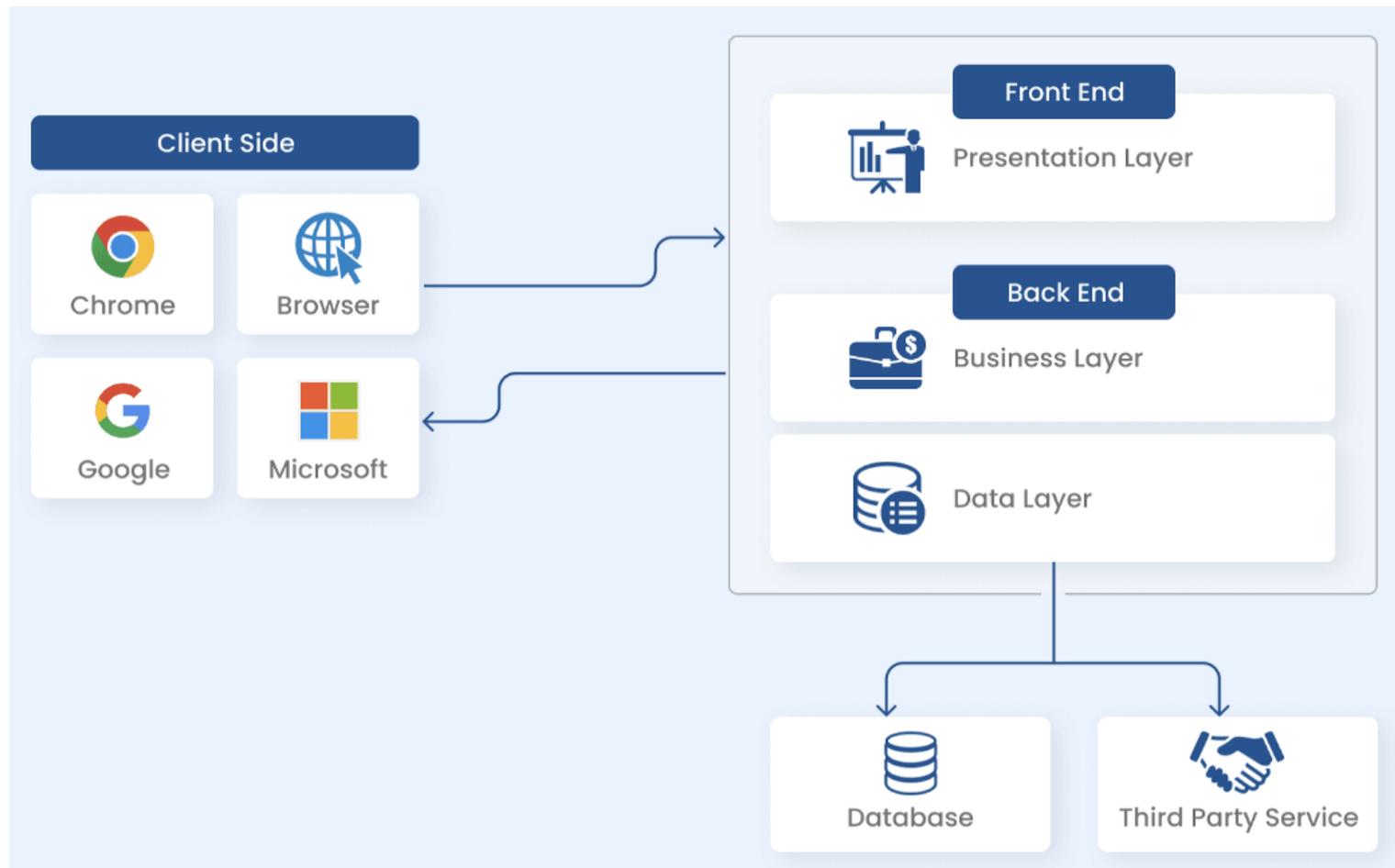


## Data tier

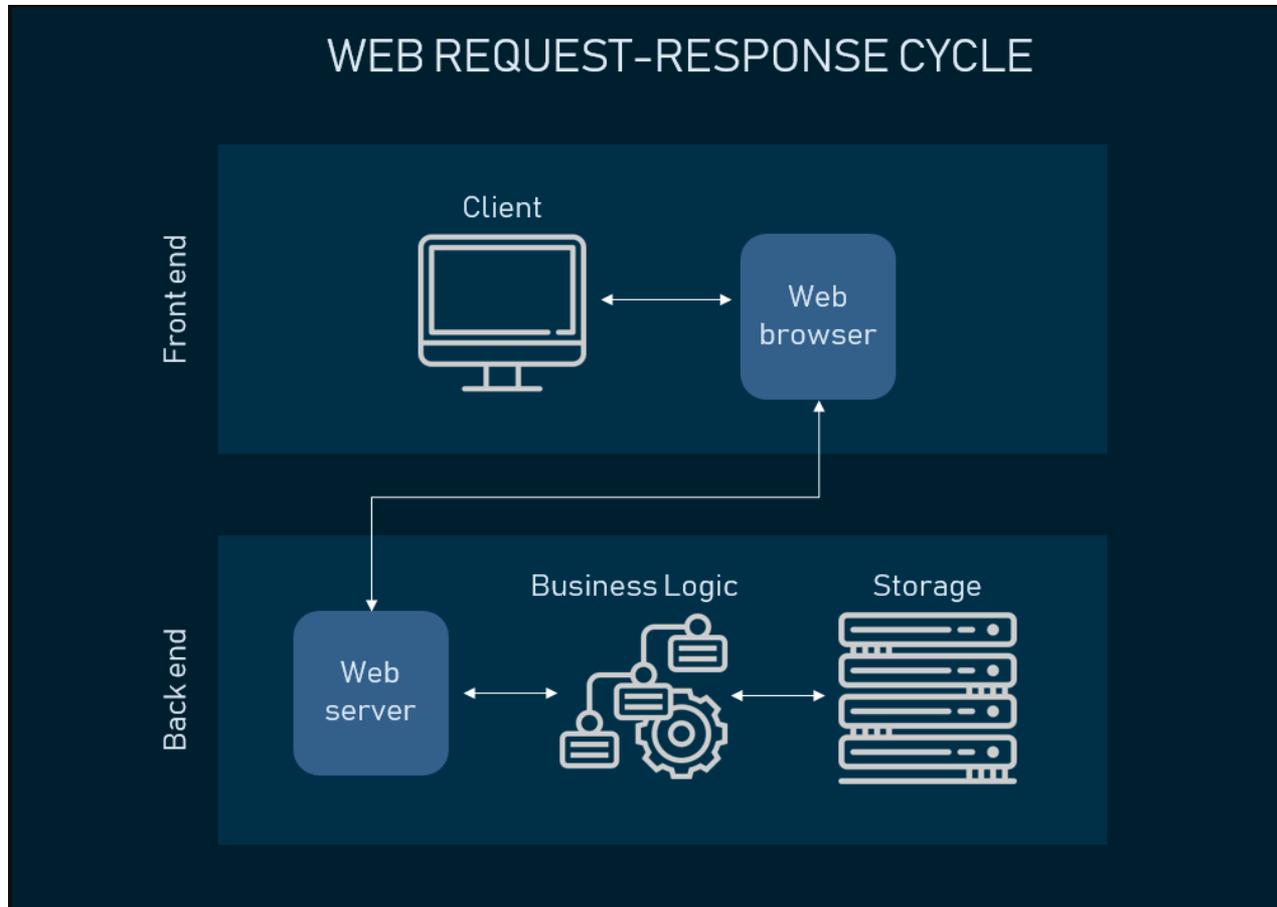
Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.



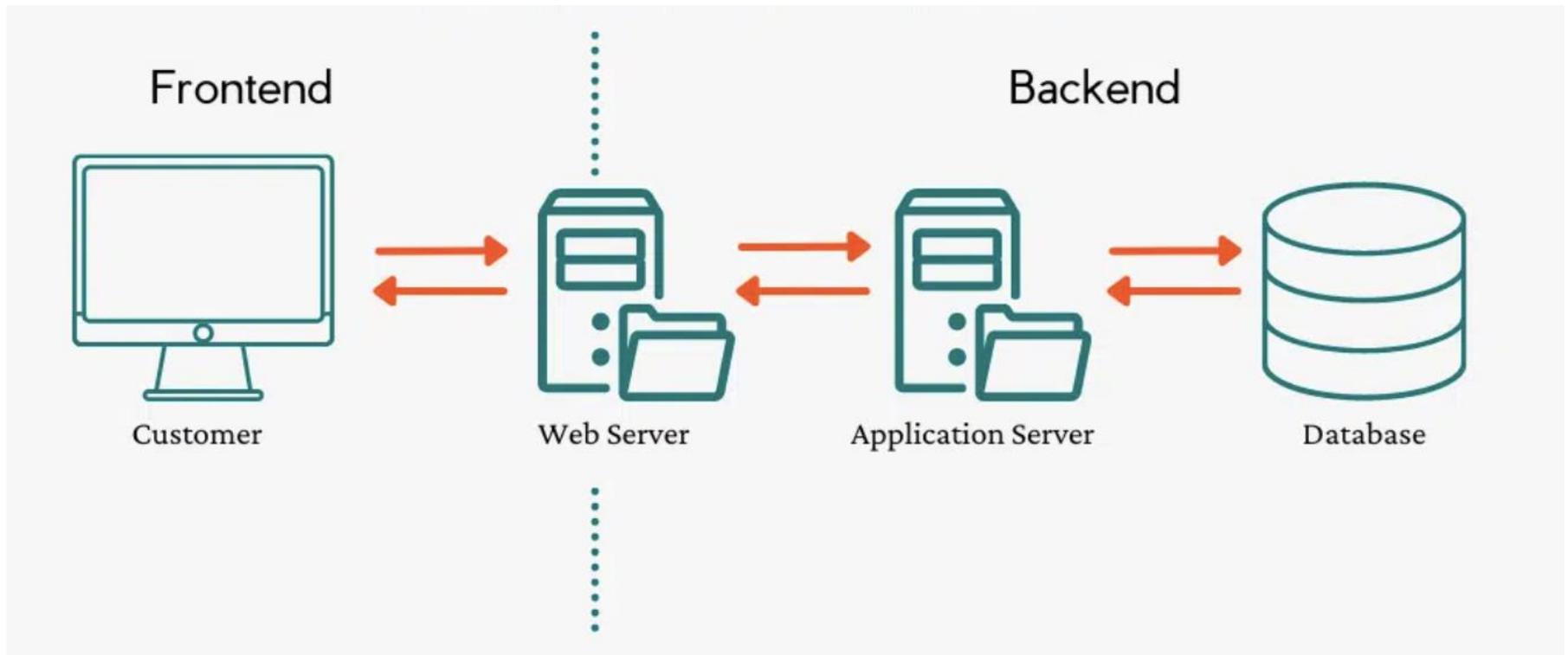
# Architettura standard delle applicazioni Web



# Ciclo di richiesta-risposta Web



# Frontend contro Backend



<https://elevatex.de/blog/it-insights/frontend-vs-backend-vs-fullstack-differences/>

# Frontend contro Backend

---

- » **Frontend:** è il livello di presentazione, si occupa dell'interattività e dell'esperienza utente; include design, menu, post, pagine, media, commenti e navigazione; riguarda il Web design.
- Il frontend semplifica e astrae i dati e i processi del backend rappresentandoli e visualizzandoli come accessi intuitivi.
  - Può trattarsi, ad esempio, di un'interfaccia utente grafica (GUI).
  - Ciò vale anche per le applicazioni software, i siti web su Internet e i giochi per computer.

<https://elevatex.de/blog/it-insights/frontend-vs-backend-vs-fullstack-differences/>

# Frontend contro Backend

---

- » Backend: è responsabile di tutti i processi che vengono eseguiti in background su un sito web o un software. Questo vale, ad esempio, per server, database o dati salvati
  - Nel web, Internet svolge un ruolo cruciale nel collegamento tra il backend e il frontend
  - *Esempio: album fotografici privati su un cloud*
    - » Il frontend è la parte con cui l'utente interagisce
    - » Il backend assicura che gli album fotografici siano salvati su un server. L'utente può accedere agli album fotografici tramite Internet in qualsiasi momento

<https://elevatex.de/blog/it-insights/frontend-vs-backend-vs-fullstack-differences/>

# Tecnologie client-server in un'applicazione basata sul Web

---



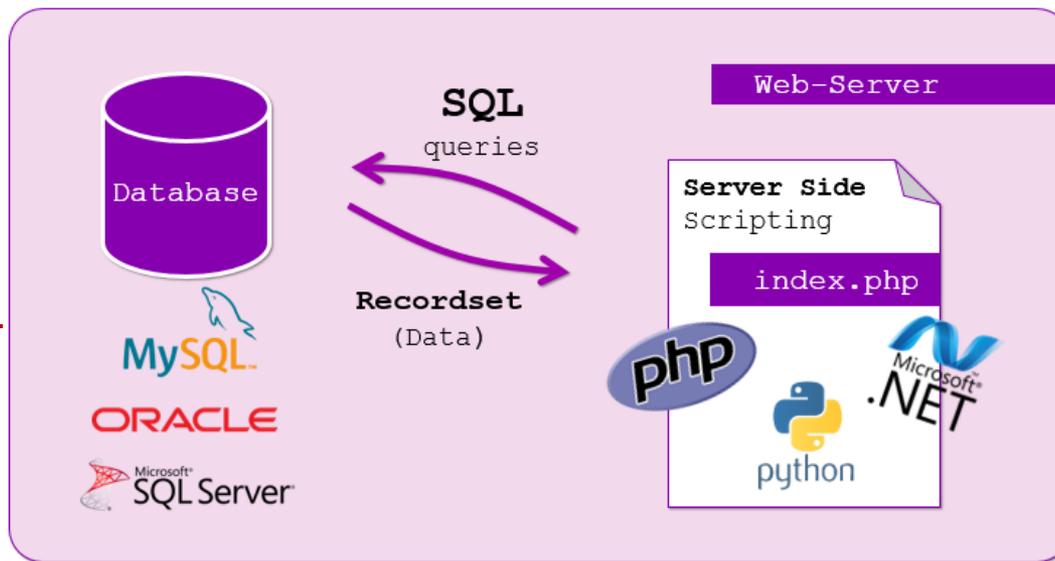
» Tecnologie lato client:  
– HTML, CSS, JavaScript

» Tecnologie lato server:  
– PHP, ASP, .Net, C#, Python, ecc.

» Database per archiviare e recuperare informazioni:  
– MySQL, Oracle, SQL Server

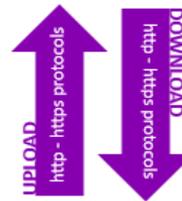
» Gli script lato server interagiscono con il database tramite query SQL per selezionare, aggiornare, inserire o eliminare dati. Quindi combinano questi dati con tag HTML per creare codice HTML, CSS e JavaScript che viene poi inviato al computer client per essere visualizzato nel browser Web come verrebbe visualizzata una pagina Web statica.





Request a page using **URL**  
Can send data using:

- **QueryString**  
e.g. `news.php?id=2`
- **POST** content of an HTML **form**



Send all the **files** (HTML, CSS, JavaScript, PNG, JPG, GIF, MP3/4...) to the **web browser**.

