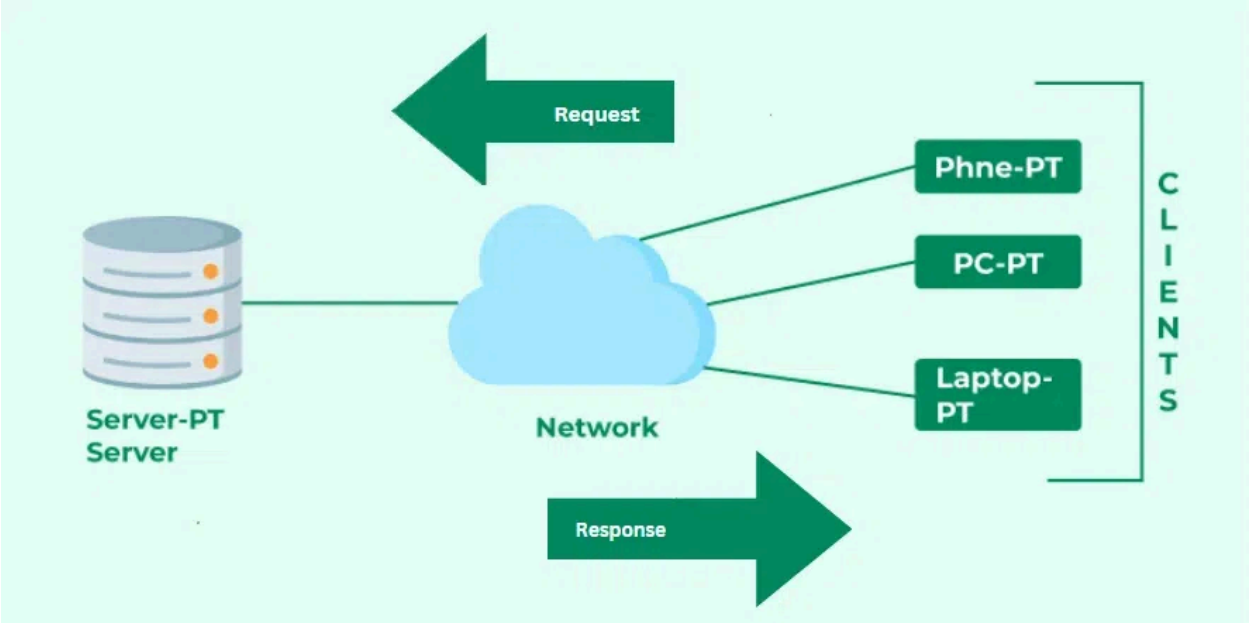
Client - Server Architecture



Client: A client is any device or software that initiates communication by requesting data or services from a server. Common client applications include:

Web browsers (e.g., Chrome, Firefox)

Email apps (e.g., Gmail, Outlook)

Server: A server is a powerful system that listens for and responds to client requests by delivering data or performing tasks. Servers often handle multiple simultaneous client requests. Common server applications include:

- Web Servers (e.g., Apache, Nginx)
- Email Servers
- Database servers

How the Browser Interacts With the Servers?

https://github.com/andrewsmathew135/Client_Server_Architecture

The process of interacting with servers through a browser involves several steps:

1. User Enters the URL (Uniform Resource Locator):
The user types a website address (e.g., www.example.com) into the browser's address bar.

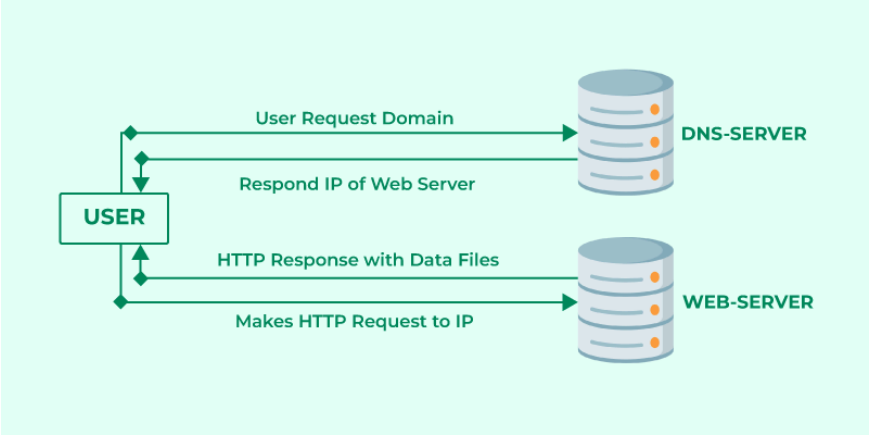
2. DNS (Domain Name System) Lookup:
The browser contacts a DNS server to convert the domain into an IP address.

3. Establishing a Connection:
The browser sends an HTTP/HTTPS request to the server using the resolved IP address.

4. Server Responds:
The server sends back website files (HTML, CSS, JavaScript, images).

5. Browser Renders the Webpage

- **DOM interpreter:** Processes HTML to structure the page.
- **CSS interpreter:** Applies styles
- **JavaScript Engine:** Adds interactivity (using JIT compilation for performance).



https://github.com/andrewsmathew135/Client_Server_Architecture

HTTP Requests

HTTP Requests are messages sent by the client to request data from the server or to perform some actions. Different HTTP requests are:

- **GET:** GET request is used to read/retrieve data from a web server. GET returns an HTTP status code of **200 (OK)** if the data is successfully retrieved from the server.
- **POST:** POST request is used to send data (file, form data, etc.) to the server. On successful creation, it returns an HTTP status code of **201**.
- **PUT:** A PUT request is used to modify the data on the server. It replaces the entire content at a particular location with data that is passed in the body payload. If there are no resources that match the request, it will generate one.
- **PATCH:** PATCH is similar to PUT request, but the only difference is, it modifies a part of the data. It will only replace the content that you want to update.
- **DELETE:** A DELETE request is used to delete the data on the server at a specified location.