

Table of Contents



Client server communication

Example

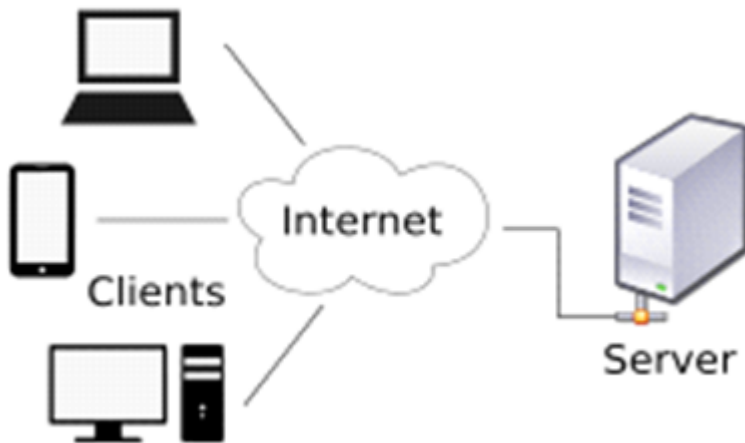
Client-host and server-host

Issues in client server communication

Client server communication

- Clients and servers exchange messages in a request response messaging pattern. The client sends a request, and the server returns a response. This exchange of messages is an example of inter-process communication.
- To communicate, the computers must have a common language, and they must follow rules so that both the client and the server know what to expect. The language and rules of communication are defined in a communication.
- All client-server protocols operate in the application layer. The application-layer protocol defines the basic patterns of the dialogue.
- To formalize the data exchange even further, the server may implement an API (such as a web service). The API is an abstraction layer for such resources as databases and custom software. By restricting communication to a specific content format. By abstracting access, it facilitates cross platform data exchange.
- A server may receive requests from many different clients in a very short period of time. Because the computer can perform a limited number of tasks at any moment, it relies on a scheduling system to prioritize incoming requests from clients in order to accommodate them all in turn.
- To prevent abuse and maximize uptime, the server's software limits how a client can use the server's resources. Even so, a server is not immune from abuse.
- A denial of service attack exploits a server's obligation to process requests by bombarding it with requests incessantly. This inhibits the server's ability to respond to

legitimate requests that can make the communication of web easier.



Client-server model of computing is a distributed application structure that partitions tasks or workloads between service providers, called servers, and service requesters, called clients.

Often clients and servers communicate over a computer network on separate hardware, but both client and server may reside in the same system.

A server machine is a host that is running one or more server programs which share its resources with clients. A client does not share any of its resources, but requests a server's content or service function. Clients therefore initiate communication sessions with servers which await (listen for) incoming requests.

Example

1. When a bank customer accesses online banking services with a web browser (the client), the client initiates a request to the bank's web server.
2. The customer's login credentials may be stored in a database, and the web server accesses the database server as a client.

3. An application server interprets the returned data by applying the bank's business logic, and provides the output to the web server.
4. Finally, the web server returns the result to the client web browser for display.

In each step of this sequence of client-server message exchanges, a computer processes a request and returns data. This is the request-response messaging pattern. When all the requests are met, the sequence is complete and the web browser presents the data to the customer.

This example illustrates a design pattern applicable to the client-server model: separation of concerns.

Client-host and server-host

Client-host and server-host have subtly different meanings than client and server.

A host is any computer connected to a network. Whereas the words server and client may refer either to a computer or to a computer program, server-host and user-host always refer to computers. The host is a versatile, multifunction computer; clients and servers are just programs that run on a host. In the client-server model, a server is more likely to be devoted to the task of serving.

Issues in client server communication

- Addressing
- Blocking versus non blocking
- buffered versus unbuffered
- reliable versus unreliable

- server architecture concurrent versus sequential scalability