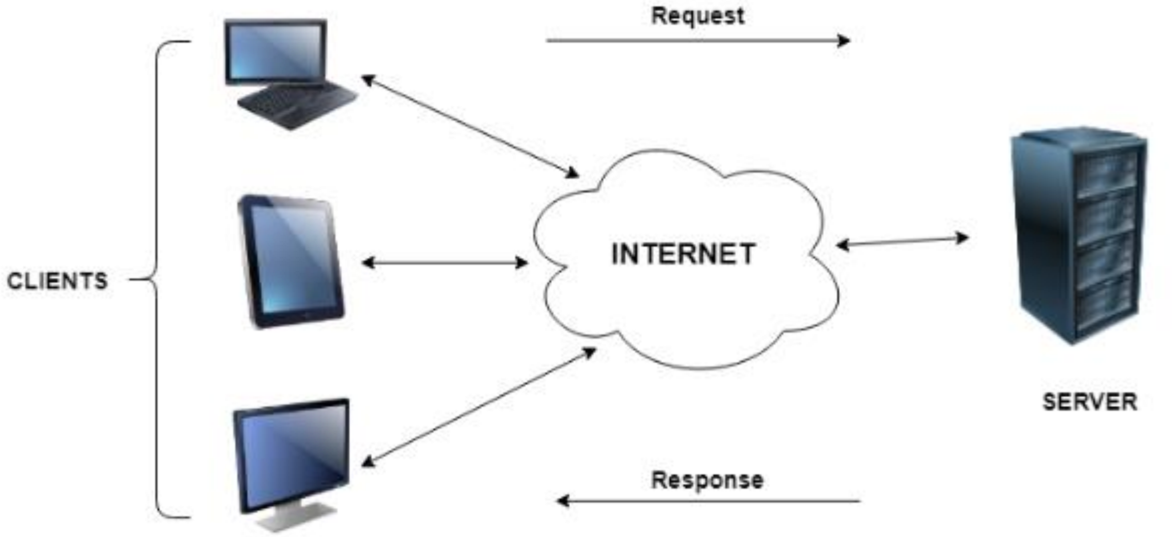
In client server computing, the clients requests a resource and the server provides that resource. A server may serve multiple clients at the same time while a client is in contact with only one server. Both the client and server usually communicate via a computer network but sometimes they may reside in the same system.



Characteristics of Client Server Computing

- The client server computing works with a system of request and response. The client sends a request to the server and the server responds with the desired information.
- The client and server should follow a common communication protocol so they can easily interact with each other. All the communication protocols are available at the application layer.
- A server can only accommodate a limited number of client requests at a time. So it uses a system based to priority to respond to the requests.
- An example of a client server computing system is a web server. It returns the web pages to the clients that requested them.

Difference between Client Server Computing and Peer to Peer Computing

- In client server computing, a server is a central node that services many client nodes. On the other hand, in a peer to peer system, the nodes collectively use their resources and communicate with each other.
- In client server computing the server is the one that communicates with the other nodes. In peer to peer to computing, all the nodes are equal and share data with each other directly.

Advantages of Client Server Computing

The different advantages of client server computing are −

- All the required data is concentrated in a single place i.e. the server. So it is easy to protect the data and provide authorization and authentication.
- The server need not be located physically close to the clients. Yet the data can be accessed efficiently.
- It is easy to replace, upgrade or relocate the nodes in the client server model because all the nodes are independent and request data only from the server.
- All the nodes i.e clients and server may not be build on similar platforms yet they can easily facilitate the transfer of data.

Disadvantages of Client Server Computing

The different disadvantages of client server computing are −

- If all the clients simultaneously request data from the server, it may get overloaded. This may lead to congestion in the network.
- If the server fails for any reason, then none of the requests of the clients can be fulfilled. This leads of failure of the client server network.
- The cost of setting and maintaining a client server model are quite high.
- Client server architecture is totally flexible to Hures multiple client program. Client server architecture in which the presentation, the application processing, and the data management are logically a separate process, which makes it very flexible and expandable approach to building system software. Unlike typical client/server applications, this architecture services multiple clients, supports client disconnection and provides a flexible framework for adding application.

One -Tier Architecture:

- One tier application is a simple program which no needs to access the network when it is running. This one tier application includes most of the simple desktop applications like word processors or compilers. One tier architecture application runs on a single computer.
- Web browser is part of a two-tier application (a Web server being the other part). But what happens if that Web browser downloads a Java applet and runs it? If the applet doesn't access the network while running, is it a one-tier or two-tier application? For present purposes, we will say that the self-contained applet is a one-tier application, since it is contained entirely on the client computer.
- One-tier architecture has a huge advantage, simplicity being one of them. No One-tier applications handle any network protocols, so their code is simple. The user's requests don't need to cross the network, wait for their turn at the server. This has the added effect of not weighing down your network with extra traffic, and not weighing down your server with extra work.

Two-Tier architecture:

- Two-tier architecture actually has got three parts which are client, server, and protocol. The protocol links the gap between the client and server tiers. The two-tier design is very effective for network programming as well as for GUI programs, in which you can allocate functionality to the host. GUI code lives on the client host, and the so-called business logic lives on the server host. Two-tier application is a client-server program with a GUI front-end written in a high-level language like Java, C++, or Visual Basic. In the two-tier program, you can see the clear separation between front and back tier.
- Input is given by the HTTP get request, via an HTML form which the user fills out. Its output is one or more HTML files. All the calculation happens on the server.
- In some cases, you can write a two-tier application without writing a server or designing a protocol. For example, you can write a Web browser that talks to a Web server using the HTTP protocol. However, if you have to write your own server, or design and execute your own protocol, you can spend more time writing your program than you would if you were writing a one-tier application.

Three-Tier architecture:

- A three-tier will need to store data on a server. The information is stored on the file system. However data honesty issues arise when multiple clients ask the server to do tasks. Since file systems have concurrency controls at best common solution is to add a third program or database.

- Databases specialize in storing, retrieving, and indexing data. Just as two-tier architecture separates GUI and business logic, three-tier architecture lets you to separate business logic and data access. You can also offer highly optimized data indices and retrieval methods, and provide for replication, backup, redundancy, and load-balancing procedures specific to your data's needs. Separating code into client and server code increases the scalability of your application so does placing data on a dedicated process, host, or series of hosts.
- SQL RDBMS, like those from Oracle and Sybase, other database types. You may have heard the names of some of these other types OODBs, ORDBs.

The three tier architecture has the following three tiers:

- Presentation Tier: This is by far the top upmost level of the application. This tier displays information related to such services as browsing merchandise and shopping cart.
- Application Tier: Is the logic tier pulled out from the presentation tier. This controls application by performing detailed processing.
- Data Tier: consist of database server. Here the information is stored and retrieved.

N-Tire architecture:

- A good example of N tier application is a stock trading system. In this system, we have multiple data feeds arrive from different sources, multiple databases are accessed and multiple clients run specialized applications.