**TITLE** : Study of Client and server architecture

**NAME** : Rajkrithika Balakrishnan

**ABSTRACT** :

The primary distinction between a client and a server is that a client is a device or program that makes service requests via the internet, whereas a server is a device or program that fulfills such requests for clients. client computers offer an interface via which a computer user can ask a server for services and view the results the server provides. Client queries are received by servers, which subsequently respond to them.In order to reply to requests from clients, servers must first await their arrival. Clients should not need to be aware of the characteristics of the system (i.e., the hardware and software) that is providing the service if a server has a standardized transparent interface, as this will make it easier for clients to use the server. Clients are frequently positioned at workstations or on personal computers, whereas servers are placed elsewhere on the network, typically on more capable hardware. This computing architecture works best when both the client and the server have certain duties that they regularly do.

When using these systems, factors including data security, user training, technical support, and pricing must be taken into account. Since strict constraints severely reduce the usability of client-server systems when they are applied, data security is a serious problem. Concerns about training and technical support must also be taken into account because they may add unforeseen expenses to the generally low hardware costs, frequently cancelling out any savings attained. As I studied In a centralized system, Data is saved on the central computer, which may be a mainframe or a minicomputer, since end users only have dumb terminals that are incapable of handling local data storage. By creating passwords for various data authorization levels, data security is traditionally controlled. Data control is centered on access to the main computer and the computer itself. Numerous control points and multiple control layers are required with the development of DDP, including the client-server system. I am interested in the issues with **end user’s** One cannot overstate the importance of security awareness. Example : Instead of leaving the workstation on if an employee needs to take a quick coffee break while accessing data from data servers, the employee should log off the system. When an employee is prepared to wrap up the day's work, workstations and floppy disks should be closed up. Nobody should ever tell someone else their password. These are just a few examples of straightforward controls that people may use without difficulty but frequently forget to use regularly. Sametime we aren't able to overcome it as it happens with human error.

**REFERENCE** :

1. “The Client Server Model”, Clients and Servers, WebDev Cave, 20 May 2016

2. “Client–Server Model.” Wikipedia, Wikimedia Foundation, 24 Aug. 2018