
# Assignment # 01

- Subject : Software Design and Architecture
- Submitted To : Madam Aasma khan
- Submitted by : Akmal afridi
- Degree : BS(SE)
- ID # 14226
- Semester : 6th
- Date : 09/06/2020

***Q.1:- Choose an software architectural style of your choice and give its explanation that must cover the below given components of an architecture style:***

- ***Elements/components***

***– that perform functions required by a system***

- ***Connectors***

***– that enable communication, coordination, and cooperation among elements***

- ***Constraints***

***– that define how elements can be integrated to form the system***

- ***Attributes***

***– that describe the advantages and disadvantages of the chosen structure***

***Ans:-***

## *CLIENT SERVER ARCHITECTURAL STYLE*

Client-server architecture, architecture of a computer network in which many clients (remote processors) request and receive service from a centralized server (host computer). Client computers provide an interface to allow a computer user to request services of the server and to display the results the server returns.

## *Components of a Client Server*

A client/server network has three main components: workstations, servers and the network devices that connect them

## *Workstations*

Workstations, or client computers, initially differentiate themselves by the operating systems running them. In a client/server network, Windows 2000, Windows XP, Windows Vista and Windows 7 are examples of workstation operating systems. Aside from being relatively cheaper than server operating systems, their functions and processes are essentially intended for client computers. Centralized databases, shared programs, management and security policies are not part of their operating systems. What they have are localized versions of databases, programs and policies that can be applied individually to them. Workstations also have lower technical specifications than servers in the areas of memory, hard drive space and processor speed, because they are not required to process requests or record data from multiple computers.

## *Servers*

Servers are distinguished by different sets of operating systems like Windows 2000 Server, Windows 2003 or Windows 2008. They also have higher memory and hard drive space and faster processors because they store and service multiple (and often simultaneous) requests from workstations. A server can assume many roles in a client/server network. It can be a file server, a mail server, a database server and domain controller all at the same time. A well-set-up network, however, delineates these roles to different servers to

optimize performance. A server, regardless of what role it has, essentially acts as a centralized repository of network files, programs, databases and policies. It makes for easier management and backup because it is not dependent to individual user configurations, but can be universally and uniformly implemented across the network.

## *Network Devices*

Network devices connect workstations and servers. They ensure that requests to and from workstations are routed properly to the correct server. Several network devices each provide different types of network connectivity. In a simple client/server network, a hub can connect a server to multiple workstations. It acts as a repeater, passing on data from one device to another. Bridges separate network segments. This is useful for offices with several departments to distinguish which department a particular workstation belongs to. Another network device, a switch, is similar to a bridge, but can detect conflicts between network segments like same IP addresses or computer names across departments. Wide-area networks use routers to connect network segments in different locations. Routers are also used to connect networks, or route information to the Internet.

## *Other Components*

Client/server networks usually have network printers or scanners, which are shared and can be used by all computers in the network. Instead of installing them individually to each computer, they can be placed in one location that everyone can access. This saves both space and money.

## *Connectors Of Client Server Protocols,*

## *Remote procedure calls (RPC)*

## *Protocols*

A communications protocol that provides a structure for requests between client and server in a network. For example, the Web browser in the user's computer (the client) employs the HTTP protocol to request information from a website on a server.

## *Remote procedure calls (RPC)*

A remote procedure call is an interprocess communication technique that is used for client-server based applications. It is also known as a subroutine call or a function call.A client has a request message that the RPC translates and sends to the server. This request may be a procedure or a function call to a remote server. When the server receives the request, it sends the required response back to the client. The client is blocked while the server is processing the call and only resumed execution after the server is finished.

## *Constraints Of Client Server*

## *Two levels, typically many clients with one server.*

- ***From client to server and server to client.***
- ***Constraints:***

clients cannot communicate directly with each other. If needed, the server

acts as a message relay for the clients to communicate.

○ Only clients can initiates communication

● All workloads are done at the server side

. Most applicable to specific kinds of problems?

● Where data can be centralized and easy to do for collaboration

● Where all clients are requesting the same type of data.

● Where clients can give specific information to request different data dynamically

● Less computational burden on the client side, which make client more lightweight.

● When clients are unable to do the heavy computation and the computation are done on the server side.

● Provide better data integrity and backup system, thus higher reliability.

● In general, people can access the data at anytime as long as they have network and authorization.

● When mobility is needed, applicated and data can be easily moved and replicated. Engender specific kinds of change resilience (Advantage)?

● Centralization Attributes Of Client Server Advantages

● Centralization of control: ○ A dedicated server controls the access of resources and integrity of the data so that a program or unauthorized client cannot damage the system easily. ○ Changes only need to be done on the server and the clients will be able to receive ○ Network

processing is done centrally, not at individual computers, which reduce the burden of the OS.

● ***Scalability:*** ○ You can increase the capacity of clients and servers separately. Any element can be increased or enhanced at any time, or you can add new nodes to the network. ○ You can add resources in the form of network segments, computers and servers to a client server network without major interruptions to the network. ○ Update task for data or other resources more efficient and easier to managed.

● ***Easy maintenance:***

○ Since backup, security and antivirus are centralized, it is easier to setup and troubleshoot, where everything takes place at one physical server.

○ Fewer support staff are needed to manage centralized security accounts than would be needed if security and resource access had to be configured on each individual computer on the network.

## *Disadvantages*

● Single point of failure : Since there's a reliance on the central server, if it fails, client requests cannot be done.

● Traffic congestion: Happens when a large number of simultaneous clients send requests to the same server. This might cause the server to slow down or even shut down.

● Cost : The cost of server hardware and software is much greater than the cost of buying desktop hardware and software licences. Thus it is expensive to scale or even hard to scale. Support/inhibit specific NFPs?

● ***Complexity***

○ Isolates the functionality to the server and the interaction to the client ○ Fairly cohesive

○ The topological hierarchy is very simple, only 2 nodes depth with server and client

● Scalability/Heterogeneity

○ Pros

■ Components are focused 3

■ Connectors are direct and simple

○ Cons

■ It is bottlenecked by the server

■ Does not replicate data(unless the model is extended by having multiple servers, load balancing...)

■ It is expensive to scale for more client connecting to the server and requires either more sophisticated hardware or more servers.

● Portability

○ The client can be ported and still use the same server

● Evolvability

○ Changes to the server processing does not affect the client

○ Changes to the client interface does not have to affect the server

○ No implicit connectors

- Dependability
  - Does support exception handling
  - Clients can connect and disconnect without affecting others
  - Only have to backup the server
- Reliability
  - A single point of failure. When the server is down, there will be no services