



Sessional Assignment

Software Design & Architecture

Submitted By: Mudassir Ahmad Khan
ID# 14086
BSSE (6th Semester)

Submitted To: Mrs. Aasma Khan
(Lecturer)

Question: Choose an software architectural style of your choice and give its explanation.

Ans:

CLIENT SERVER ARCHITECTURAL STYLE

- Client-server architecture (client/server) is a network architecture in which each computer or process on the network is either a client or a server.
-
- 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.

The Internet has brought revolutionary change in the world of technologies, bringing the entire globe interconnected. But it follows some specific architecture and structure for communication. The popular is the client-server architecture which is a computing model, where the server hosts, distributes and controls the majority of the resources as well as services to be used by the client. Such structural designs are made up of one or more client systems connected to central or main servers through a network, which we usually know as an Internet connection. All such systems associated with it share computing resources.

The client-server architecture is also termed as a network-computing structure because every request and their associated services are distributed over a network. So now the question is how the thing works? In the client-server architecture, when the client computer sends a request for data to the server through the internet, the server accepts the requested, process it and deliver the data packets requested back to the client. One special feature is that the server computer has the potential to manage numerous clients at the same time. Also, a single client can connect to numerous servers at a single timestamp, where each server provides a different set of services to that specific client.

Let us take a scenario where we require today's weather data for our city. There will be someone, i.e., the server in which periodic updates regarding the weather will be updated and stored. Before the existence of computers, people get such news through the daily newspaper or may listen to the radio broadcasting weather news. In the modern world with computers and servers, there are two participants — first, the users, who scrounge for the weather report and the second is the weather info up-loaders (in the server) who provide the weather-related information. So these two participants are termed individually.

- First of all, consumers who type specific URLs for weather reports and consume particular information. They are also called customers.
- Second, there are providers, who upload their data on their system/server to provide information through the Internet. They are also called servers.

So, the clients and servers are two different computers in different parts of the world that are connected through the Internet. However, it is not compulsory to have the client, and the server resides miles apart, rather it could remain within the same building as well.

So, now we as a user can type that specific URL (let suppose any weather-forecasting site) and surf for the weather report. But some additional factors come into action with such online data gathering technologies. Your newspaper or radio uses your local language to give you the weather report and other information. But, for the client-server architecture on the web, specific factors need to be considered:

- A specific set of languages along with a communication standard, exclusively a protocol for the interaction of two systems. The most popular are the HTTP and HTTPS (Hyper Text Transfer Protocol Secure).

- Mechanism and protocol for requesting the required aspects from the server. That could be in any structure of formatted data. Mainly implemented and popular formats are done in XML and JSON.
- Next, the server responds by sending a reply in a structure of formatted data, (usually XML or JSON).

Client Working:

A client computer provides the user-interaction facility (interface) and some or all of the application processing.

Server Working:

A server computer might provide high-volume storage capacity, heavy data crunching, and/or high resolution graphics.

Typically, several client computers are connected through a network (or networks) to a server which could be a large PC, minicomputer or a mainframe computer. Every computer connected to a website acts as a client while the website computer acts as a server. Also called CLIENT-SERVER ENVIRONMENT.

Components of a Client Server architectural style

A client/server network has three main components:

- workstations
- servers
- network devices that connect them

1. 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.

2. 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.

3. 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 Architectural Style

- 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.

Types of Client Server Architectural Style:

There are various types of client-server architecture which are described as below:

1. The Two Tiers Architecture:

In this type of architecture, the workload is divided between the server (host of the system) and the client (which hosts the user interface).

In reality these are located on separate computers but there is no absolute requirement of this, providing that the tiers are logically separated can be hosted (e.g. development and testing) on the same computer.

Advantages:

- **Ease in Developing Applications:** Applications can easily be developed due to simplicity.
- **User Satisfaction:** Maximum user satisfaction is gained with accurate and fast prototyping of applications through robust tools.
- **Applicable for Homogeneous Environment:** Since this contains static business rules it's more applicable for homogeneous environment.
- **High Performance:** Database server and business logic is physically close, which offers higher performance.

Limitations:

The two tier architecture proved to be a good solution when user population work is usually small (up to about 100 concurrent users) but it rapidly proved to have a number of limitations:

- **Performance:** Performance begins to deteriorate as the population grows. This is due to the reason that each user has its own connection and the server has to keep all these connections live even when no work is being done.
- **Security:** Each user must have their own individual access to the database, and whatever rights may be required in order to run the application.
- **Capability:** Independent of the type of client, much of the data processing has to be located in the database making it totally dependent upon the capabilities and implementation provided by the database manufacturer. This limits the application functionality.
- **Portability:** As the two-tier architecture is dependent upon the specific database implementation, porting an existing application to a different dbms becomes a major issue.

Having said that, this architecture has found a new lease of life in the internet age, it can work well in a disconnected environment. However, in many ways this implementation harks back to the mainframe architecture and indeed, a browser based, two-tier application, can suffer from many ways of the same issue.

2. The Three-Tier Architecture:

To overcome the limitations of the two-tier architecture, an additional tier was introduced - creating what is now the standard three-tier client-server model. The purpose of the additional tier (called as “middle” or “rule” tier) is to handle application execution and database management, as with the two-tier model, the tiers can either be implemented on different physical machines, or multiple may be co-hosted on a single machine.

In the three-tier architecture the functional process logic, data access, computer data storage and user interface are developed and maintained as independent modules on separate platforms.

The three-tiers in a three-tier architecture are:

- **Presentation Tier:** Occupies the top level and displays information related to services available on a website. This tier communicates with other tiers by sending results to the browser and other tiers in the network.
- **Application Tier:** Also called the middle tier, logic tier or business logic, this tier is pulled from the presentation tier. It controls application functionality by performing detailed processing.

Data Tier: Houses the database server where information is stored and retrieved. Data in this tier is kept independent of application servers or business logic.

Advantages:

- **Improved Data Integrity:** Data corruption through client application can be eliminated as the data passes through the middle tier for updating the database ensures its validity.
- **Enhanced Security:** The placement of business logic on a centralized server makes the data more secure. Data security is enhanced on a service-by-service basis as the client does not interact with the database directly.
- **Hidden Database Structure:** The actual structure of the database often remains hidden from clients, enabling any change in the database to be hidden.

Limitations:

- **Complexity of Communication:** Usually more efforts should be ensured when creating 3-tier applications as the communication points are increased (client to middle tier to server) and the performance is increased by tools like Visual Basics, Power Builder etc.

3. n-Tiers Architecture:

Often referred as Multitier Architecture. It is a client-server architecture in which presentation, application processing, and data management functions are physically separated. It is an expanded form of three-tier architecture.

Advantages:

It provides a model by which developers can create flexible and reusable applications. By segregating an application into tiers, developer acquires the option of modifying or adding a specific layer, instead of reworking the application.

Limitation:

Difficult to Implement: Due to componentization of tiers, the complex structure is difficult to implement or maintain.

Advantages of Client-Server Architecture Style:

Organizations often seek opportunities to maintain services and quality competition to sustain its market position with the help of technologies. Deployment of client-server computing in an organization will effectively increase its productivity through the usage of cost-effective user interface, enhanced data storage, vast connectivity and reliable application services

Improved Data Sharing:

Data is retained by usual business processes and manipulated on a server is available for designated users (clients) over an authorized access.

Integration of Services: Every client is given the opportunity to access corporate information via desktop interface eliminating the necessity to log into a terminal mode or processor.

Shared Resources Amongst Different Platforms: Application used for client-server model is built regardless of the hardware platform or technical background of the entitled software (operating system software) providing an open computing environment, enforcing users to obtain the services of clients and servers (database, application and communication services)

Data Processing Capability Despite the Location: Client-server users can directly log into a system despite of the location or technology of the processors.

Easy Maintenance: Client-server architecture is distributed model representing dispersed responsibilities among independent computers integrated across a network. Therefore, it's easy to replace, repair, upgrade and relocate a server while client remains unaffected. This unaware change is called as Encapsulation.

Security: Servers have better control access and resources to ensure that only authorized clients can access or manipulate data and server updates are administered effectively.

Disadvantages of Client-Server Architecture Style:

Overloaded Servers: When there are frequent simultaneous client requests, server severely get overloaded, forming traffic congestion.

Impact of Centralized Architecture: Since it is centralized, if a critical server failed, client requests are not accomplished. Therefore, client-server lacks the robustness of a good network.

Thank You!