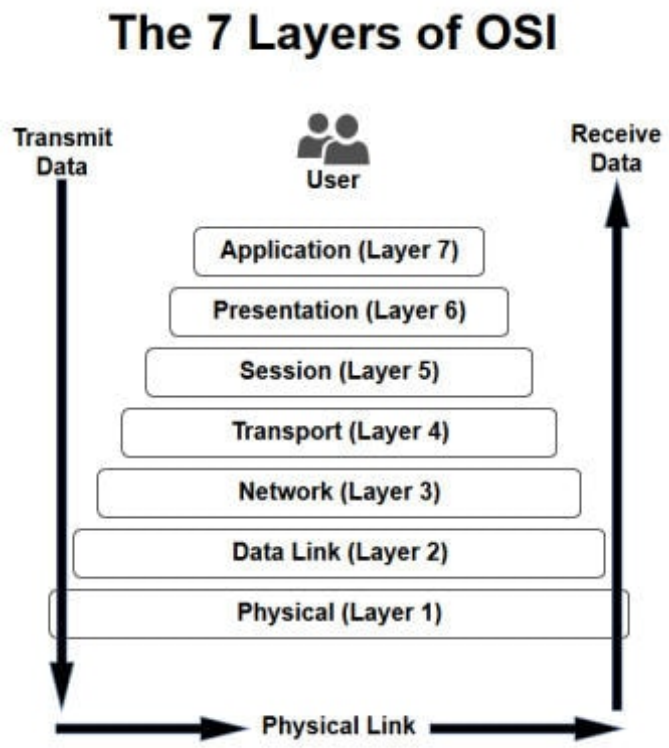
Introduction to OSI Layers

International standard organization (ISO) established a committee in 1977 to develop an architecture for computer communication. Open Systems Interconnection (OSI) reference model is the result of this effort. In 1984, the Open Systems Interconnection (OSI) reference model was approved as an international standard for communications architecture. The term "open" denotes the ability to connect any two systems which conform to the reference model and associated standards.

The OSI model is now considered as the primary Architectural model for inter-computer communications. The OSI model describes how information or data makes its way from application programs (such as gmail) through a network medium (such as wire) to another application program located on another network.

The OSI reference model divides the problem of moving information between computers over a network medium into 7 smaller and more manageable problems. This separation into smaller more manageable functions is known as layering.

Each layer provides a service to the layer above it in the protocol specification. The lower 4 layers (transport, network, data link and physical — Layers 4, 3, 2, and 1) are concerned with the flow of data from end to end through the network. The upper 3 layers of the OSI model (application, presentation and session—Layers 7, 6 and 5) are orientated more toward services to the applications. Data is Encapsulated with the necessary protocol information as it moves down the layers before network transit.



Layer 7 – Application Layer

The Application Layer is the one at the top - it's what most users see. In the OSI model, this is the layer that is the "closest to the end user". The Application layer provides the interface between the user application and the network. A web browser and an email client are examples of user applications.
It is important to understand that the user application itself does not reside at the Application layer - the protocol does. The user interacts with the application, which in turn interacts with the application layer protocol. Examples of Application layer protocols include - HTTP via a web browser, POP3 and SMTP via an email client, FTP.

Layer 6 - The Presentation Layer

Presentation layer is also called the Translation layer. It takes the data from the application layer and changes it as per the required format to transmit over the network. In other words, presentation Layer takes care that the data is sent in such a way that the receiver will understand the information (data) and will be able to use the data. While receiving the data, presentation layer transforms the data to be ready for the application layer.

The functions of the presentation layer are :

Translation : For example, ASCII to EBCDIC.

Encryption/ Decryption : Data encryption translates the data into another form or code. The encrypted data is known as the cipher text and the decrypted data is known as plain text. A key value is used for encrypting as well as decrypting data.

Compression: Reduces the number of bits that need to be transmitted on the network.

This layer could be part of OS.

Layer 5 – The Session Layer

When two devices, need to "speak" with one another, a session needs to be created and this is done by the Session Layer. Session layer provides mechanism for controlling the dialogue between the two end systems. It defines how to start, control and end conversations (called sessions) between applications.

This layer requests for a logical connection to be established on an end-user's request. Any necessary username or password validation is also handled by this layer. Session layer is also responsible for terminating the connection.

This layer provides services like dialogue discipline which can be full duplex (simultaneous two-way communication) or half duplex (two-way communication, but not simultaneous). Session layer can also provide a check-point mechanism such that if a failure of some sort occurs between checkpoints, all data can be retransmitted from the last checkpoint.

Layer 4 - Transport Layer

Transport Layer is called as Heart of OSI model.

It is responsible for the End to End delivery of the complete message. Transport layer also provides the acknowledgement of the successful data transmission and re-transmits the data if error is found.

- **At sender's side:**

Transport layer receives the formatted data from the upper layers, performs segmentation and also implements Flow & Error control to ensure proper data transmission. It also adds Source and Destination port number in its header and forwards the segmented data to the Network Layer.

Note: The sender needs to know the port number associated with the receiver's application. Generally this destination port number is configured, either by default or manually. For example, when a web application makes a request to a web server, it typically uses port number 80, because this is the default port assigned to web applications. Many applications have default port assigned.

- **At receiver's side:**

Transport Layer reads the port number from its header and forwards the data which it has received to the respective application. It also performs sequencing and reassembling of the segmented data.

The functions of the transport layer are :

Segmentation and Reassembly:

This layer accepts the message from the (session) layer , breaks the message into smaller units called segments . Each of the segment produced has a header associated with it. The transport layer at the destination station reassembles the message from these segments.

Service Point Addressing:

In order to deliver the message to correct process, transport layer header includes a type of address called service point address or port address. Thus by specifying this address, transport layer makes sure that the message is delivered to the correct process.

The services provided by transport layer :

Connection Oriented Service:

It is a three phase process which includes – Connection Establishment, Data Transfer & Termination / disconnection. In this type of transmission the receiving device sends an acknowledgment, back to the source after a packet or group of packets is received. This type of transmission is reliable and secure.

Connection less service:

It is a one phase process and includes data transfer. In this type of transmission the receiver does not acknowledge receipt of a packet. This approach allows for much faster communication between devices. Connection oriented Service is more reliable than connection less Service.

Data in the Transport Layer is called as Segments. Transport layer is a part of the OS.

Layer 3 - Network Layer

Network layer is implemented by networking devices such as routers.

Network layer works for the transmission of data from one machine to another located in different networks. A Segment is often referred as Packet in Network layer. The main task of Network Layer is "Packet Routing" i.e. selection of shortest path to transmit the packet, from the number of routes available. For example, a computer in Mumbai may want to connect to a server in Banglore. The network layer will now decide that out of thousands of different paths available which one to take. The sender & receiver's IP address are placed in the header by network layer.

The functions of the Network layer are :

Routing: The network layer protocols determine which route is suitable from source to destination. This function of network layer is known as routing.

Logical Addressing: In order to identify each device on internet, network layer defines an addressing scheme. The sender & receiver's IP address are placed in the header by network layer. Such an address distinguishes each device uniquely and universally.

Layer 2 - The Data-Link Layer

The Data-Link Layer is implemented by networking devices such as switches.

While the Network layer is concerned with transporting data ***between networks***, the Data-Link layer is responsible for transporting data ***within a network***. Hence, the data link layer is responsible for the node to node delivery of the message.

The functions of the data Link layer are :

Framing: Framing is that function of the data link layer in which packet received from Network layer is further divided into frames depending on the allowed frame size of NIC(Network Interface Card). Special bit patterns are attached to the beginning and end of each frame.

Physical addressing: After creating frames, Data link layer adds physical addresses (MAC address) of sender and/or receiver in the header of each frame.

Error control: Data link layer provides the mechanism of error control in which it detects and retransmits damaged or lost frames.

Flow Control: The data rate must be constant on both sides else the data may get corrupted thus , flow control coordinates that amount of data that can be sent before receiving acknowledgement.

Access control: When a single communication channel is shared by multiple devices, MAC sub-layer of data link layer helps to determine which device has control over the channel at a given time.

Data Link Layer is divided into two sub layers :
Logical Link Control (LLC)
Media Access Control (MAC)

A switch keeps a record of the MAC addresses of all the devices connected to it. With this information, a switch can identify which system is sitting on which port. So when a frame is received, it knows exactly which port to send it to

Layer 1 - Physical Layer

The lowest layer of the OSI reference model is the physical layer. It is responsible for the actual physical connection between the devices. Hub, Repeater, Modem, Cables are Physical Layer devices.

The physical layer contains information in the form of bits. It defines rules by which bits are passed from one system to another on a physical communication medium.
It covers all mechanical, electrical, functional and procedural aspects for physical communication. Characteristics as voltage levels, timing of voltage changes, physical data rates, maximum transmission distances, physical connectors and other similar attributes are defined by physical layer specifications.

Network Layer, Data Link Layer and Physical Layer are also known as Lower Layers or Hardware Layers.

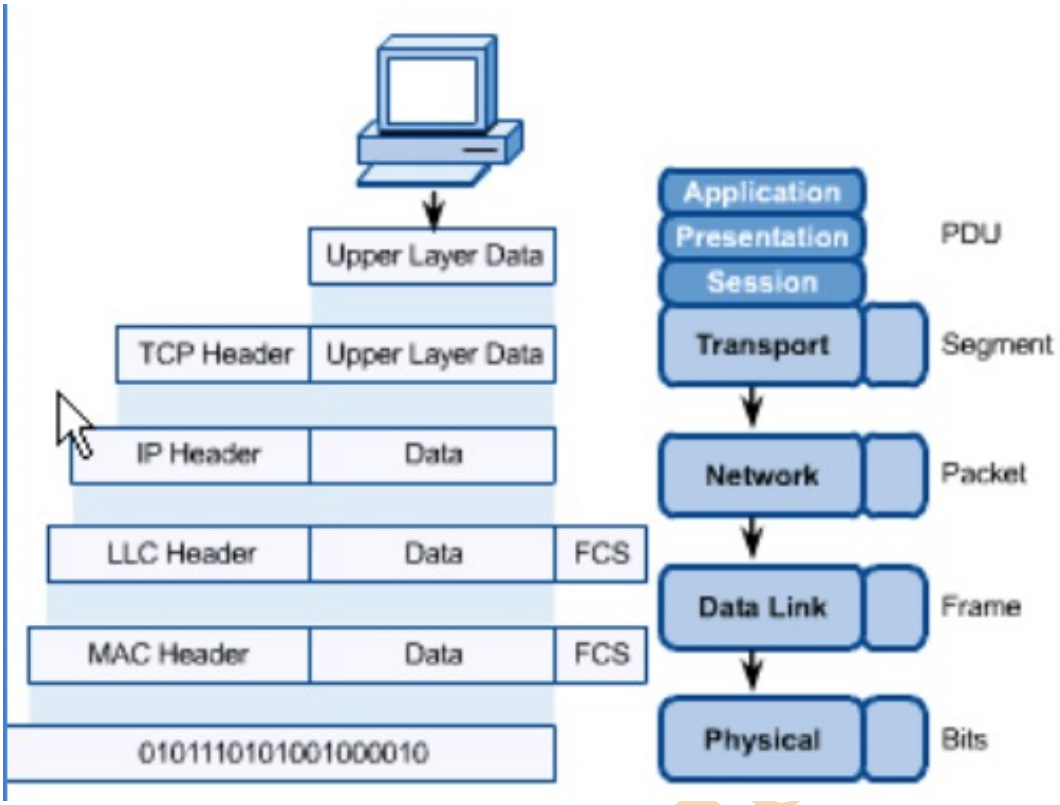
Encapsulation and Layered Communication

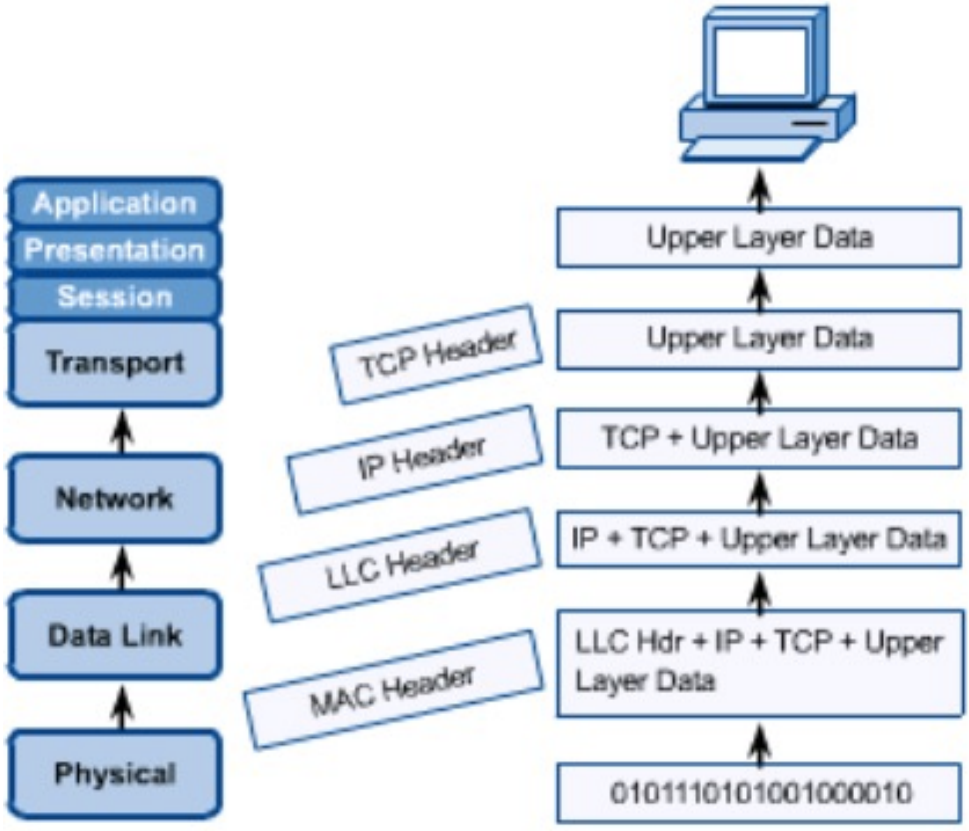
As data is passed from the user application down the layers of the OSI model, each layer adds a **header** containing protocol information specific to that layer. These headers are called **Protocol Data Units (PDUs)**, and the process of adding these headers is called **encapsulation**. Note that in the TCP/IP protocol suite only the lower layers perform encapsulation. The data along with PDU of each layer is identified with a different term:

Layer	Data along with PDU is called:
Application	-
Presentation	-
Session	-
Transport	Segments
Network	Packets
Data-Link	Frames
Physical	Bits

Encapsulation Illustrated

The following diagrams illustrates how basic encapsulation occurs:





During encapsulation on the sending host:

- Data from the user application is handed off to the Transport layer.
- The Transport layer adds a header containing protocol-specific information and then hands the segment to the Network layer.
- The Network layer adds a header containing source and destination logical address and then hands the packet to the Data-Link layer.
- The Data-Link layer adds a header containing source and destination physical address and other hardware-specific information and hands the frame to the physical layer.
- The Physical layer then transmits the data as bits.

During decapsulation on the receiving host, the reverse occurs:

- The frame is received from the physical layer.
- The Data-Link layer receives the frame, processes its header, strips it off, and then hands the packet to the Network layer.
- The Network layer processes packet's header, strips it off, and then hands the segment to the Transport layer.
- The Transport layer processes segment's header, strips it off, and then hands the data to the user application.

Web System architecture - 1,2,3 and n tier

The architecture of a software may consist of One Tier or Two Tiers or Three Tiers or N-Tiers. A "tier" can also be referred to as a "layer".

Three layers involved in the application are:

1) Presentation Tier
2) Business Logic Tier or Application Tier
3) Data Tier or Data Access Tier or Database Tier

1) Presentation Tier

The presentation tier is the front end layer in the n-tier system and consists of the user interface. It is what the user sees. This user interface is often a graphical one accessible through a web browser or web-based application and displays content and information useful to an end user. This tier is often built on web technologies such as HTML5, CSS, JavaScript or through other popular web development frameworks.

Presentation Tier is also known as Client layer because it is this layer which the user sees while using the software. This layer communicates with Application layer. This layer takes information from the user in terms of keyboard actions, mouse clicks etc. and passes on this information to the Application tier.

Example: Login page of Gmail where an end user could see text boxes and buttons to enter user id & password.

2) Business Logic Tier or Application Tier

The Application tier contains the functional business logic which makes the application do some work. It is often written in Java, Python, C++, etc. This layer acts as a mediator between the Presentation and the Data layer. Complete business logic will be written in this layer.

Considering Gmail's login page example the Application tier will take the user-id & password from the user, will access the Data Tier to find a match for the same and will grant or deny access to the user.

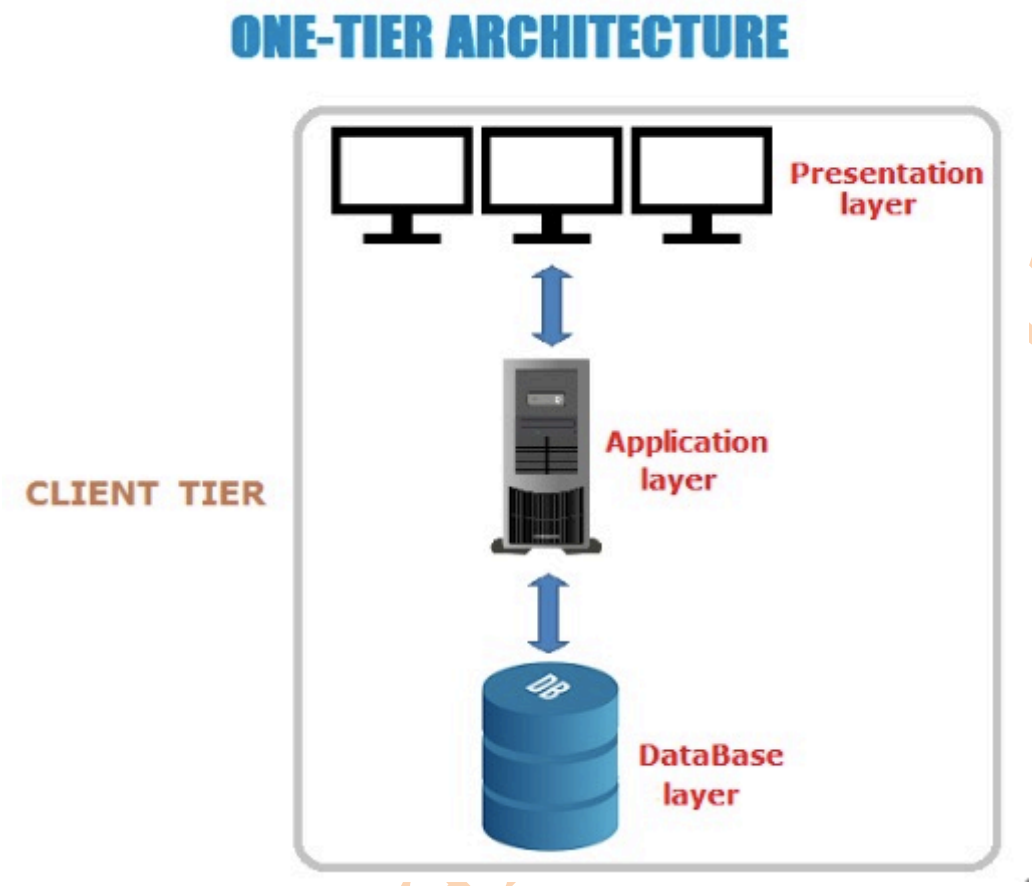
3) Data Tier

The data tier comprises of the database/data storage system. Examples of such systems are MySQL, Oracle, Microsoft SQL Server, MongoDB, etc. Data in this tier is accessed by the Application layer.

Types of Software Architecture:

A) One Tier application

One tier application is also called standalone application. It has all the layers such as Presentation, Business, Data Tiers in a single software package. Applications which handle all the three tiers such as Tally, MS Office etc. come under one tier application. The data is stored in the local system or a shared drive.

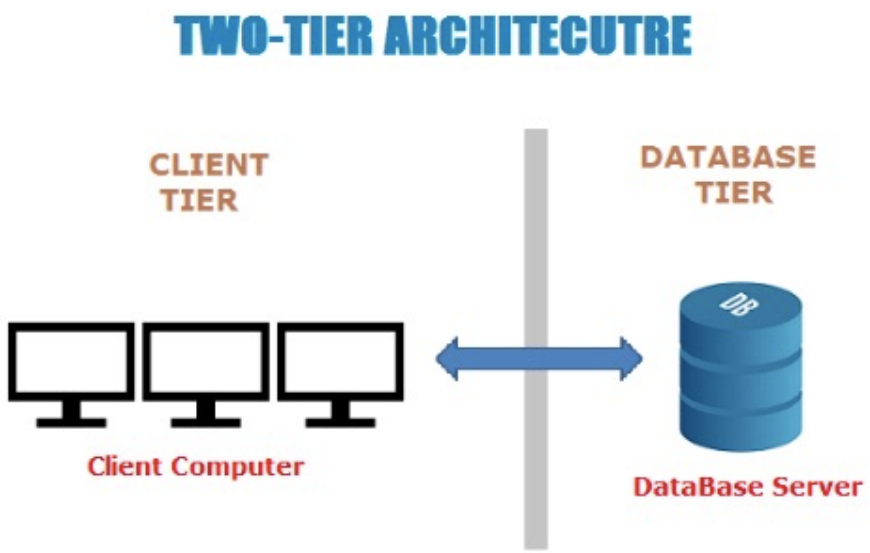


B) Two Tier application

A two-tier application is also called Client-Server application. The Two-tier architecture is divided into two parts:

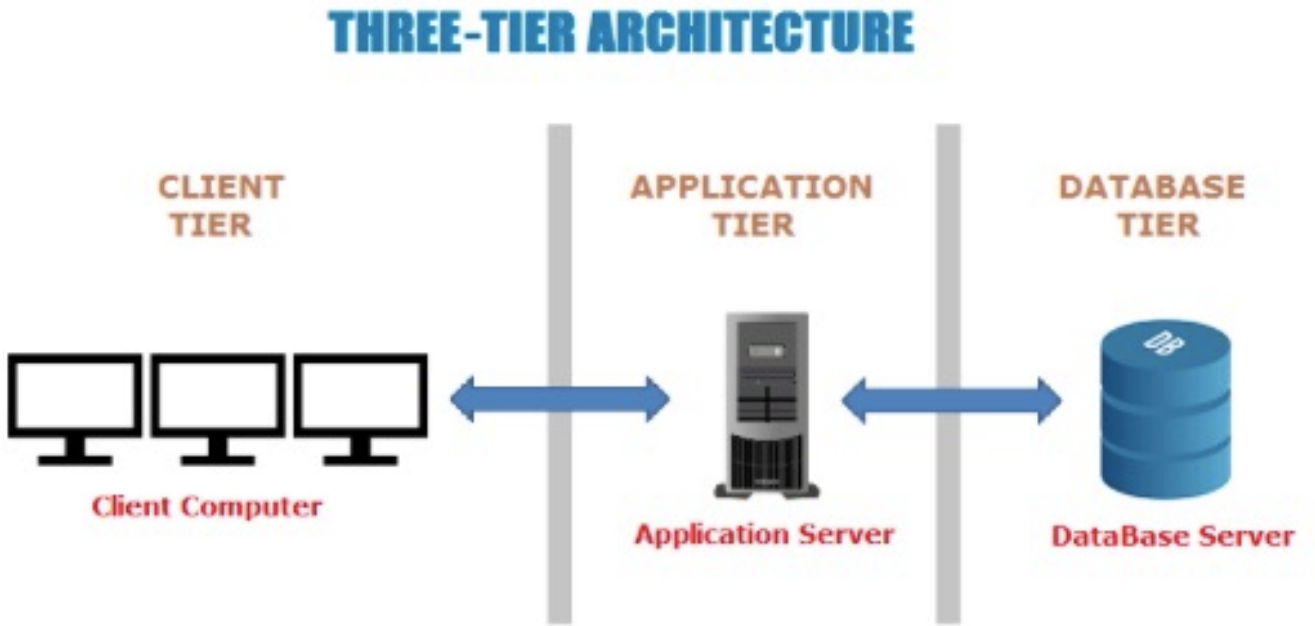
1. Client Tier
2. Database Tier

Client Tier handles both Presentation and Application layers and the Server system handles Database layer. It is also known as client server application. The communication takes place between the Client and the Server. Client system sends the request to the Server system and the Server system processes the request and sends back the data to the Client System.



C) Three Tier application

The typical structure for a 3-tier architecture would have the presentation tier deployed to a desktop, laptop, tablet or mobile device either via a web browser or a web-based application.
The underlying application tier is usually hosted on an application server but can also be hosted in the cloud.
And the data tier would normally comprise of one or more relational databases or other types of database systems hosted either on a database server.



D) n-Tier application

n-Tier application is also called Distributed application. It is similar to three tier architecture but the number of application servers are increased in order to distribute the business logic.

U R L – Uniform Resource Locator

A Uniform Resource Locator (URL) is the address of a resource on the Internet. A URL indicates the location of a resource as well as the protocol used to access it. URLs occur most commonly to reference web pages (http), but are also used for file transfer (ftp), email (mailto), database access (JDBC), and many other applications.

A URL contains the following information:

1) The protocol used to a access the resource
2) The location of the server (whether by IP address or domain name)
3) The port number on the server (optional)
4) The location of the resource in the directory structure of the server
5) A fragment identifier (optional)

Uniform Resource Locator is also known as a Universal Resource Locator (URL) or Web address. Uniform Resource Locator was defined in RFC 1738 in 1994 by Sir Tim Berners-Lee, the inventor of the World Wide Web.

All URLs are presented in the following order:

- Scheme name (also called protocol)
- Colon and two slashes
- Location of the server or Domain Name
- The port No.(optional) and
- Location(directory) of the resource on the server.
- Fragment identifier (optional)

So, the format will look like this:

scheme://location:port/file-on-server.htm?querystring=1

While most website URLs begin with "http" several other schemes also exist. Below is a list of various URL schemes:

- http – a webpage, website directory, or other file available over HTTP
- ftp – a file or directory of files available to download from an FTP server
- telnet – a Unix-based computer system that supports remote client connections
- mailto - an email address (often used to redirect browsers to an email client)

Consider the following example to understand URL better

http://www.study-circle.org/Batches/2.pdf

This URL indicates that there is a file named "2.pdf" inside the directory(folder) "Batches" on a website with domain name "www.study-circle.org".

This site uses regular http protocol, while sites which deal sensitive information use a secure version of this scheme called "https". "https" is mostly used by those sites which indulge in some kind of financial transaction like e-commerce & banking websites.

The "org" in the above site is called Top-level domain (TLD). It is the last segment of a domain name that follows immediately after the "dot" symbol. TLDs are mainly classified into two categories: generic TLDs and country-specific TLDs.

Examples of some of the popular gTLDs include .com, .org, .net, .gov, .biz and .edu.

Examples of some of the popular ccTLDs include .in, .us, .uk.
cc stands for "country code"

Two common elements of confusion about URLs:

- The "www" is not actually part of the technical protocol. Websites just started using this to indicate the user that she is using the World Wide Web. This is why if you go to http://study-circle.org, it redirects you to http://www. study-circle.org.

- Most users access the Internet via a Web browser, which inserts port 80 on HTTP connections behind the scenes. This is why if you type http://www.study-circle.org:80, you will see the same website as if there were no port number.

Finally, the following URL demonstrates a fragment identifier, more commonly known as a querystring.

http://www.google.com/some-page?search=hello

This is saying that to use the HTTP protocol to send a request to the website at google.com (over port 80) and to ask for "some-page" and send in the search variable "hello". This is why you'll sometimes see an extremely long URL as many variables are being sent to the Web server in more interactive Web applications.

Domain Name System (DNS)

Every website/computer on the internet has an associated IP address, for example 172.16.254.1. This IP address uniquely identifies that website on the internet. So the Internet basically runs on IP addresses which computers and other routing devices can understand. So If we want to visit any website we should be knowing its IP address. However, it is very difficult for we human beings to remember this numeric address for every website. That's why every website has an associated "Domain Name", for example, www.sandeepgupta.org.

Domain Name System (DNS) is a mechanism that converts the domain name we type in our web browser address bar to the IP address of Web server hosting the site. In other words, DNS is like a phone book for the Internet. If you know a person's name but don't know her telephone number, you can simply look it up in a phone book. DNS provides this same service to the Internet.

How does DNS work?

When you visit a website such as *sandeepgupta.org,* your computer follows a series of steps to turn the human-readable web address into a machine-readable IP address. This happens every time you use a domain name, whether you are viewing websites, sending email or listening to some Internet radio station.

Step 1: Search Local DNS Cache

Lets say you type *sandeepgupta.org* in your web browser. Your web browser or Operating System will first search the computer's Local DNS cache for the equivalent IP address. The Local DNS Cache stores information about all those websites which your computer has recently visited. If your computer doesn't already know the answer, it needs to perform a DNS query to find out.

Step 2: Ask the Resolver (Recursive DNS Server)

If the information is not stored locally, your computer queries (contacts) your Internet Service Provider's (ISP's) Recursive DNS Server also called Resolver Server or Local DNS Server. This server first checks its own cache memory for the required IP address. Most of the times the required IP address is found here and is then returned to the user.

Step 3: Ask the Root Server

If the recursive server doesn't have the answer, it queries the Root Server also called Root Name Server. As the name suggests, the root server sits at the top (root) of the DNS hierarchy. There are 13 sets of root servers strategically placed all around the world. These 13 sets are operated by 12 different organizations. Each set has its own unique IP address. When the root server receives query for sandeepgupta.org the root server will not know its IP address. But the root server does know where to send the Resolver to help it find the IP address. The root server will read your domain name from right to left and will direct the Resolver to TLD (Top Level Domain) Server for the ".org" domain.

Step 4: Ask the TLD Server

Each TLD such as .com, .org, and .net have their own set of servers called TLD Servers, which act like a receptionist for each TLD. These servers don't have the information we need, but they can refer us directly to the servers that do have the information. The TLD servers review the next part of your domain name and direct your query to the Authoritative Name Server responsible for your specific domain.

Step 4: Ask the Authoritative Name Server

The Authoritative Name Server is responsible for knowing all the information about a specific domain which is stored in its DNS records. The Resolver will now ask the Authoritative Name Server for the IP address of sandeepgupta.org.

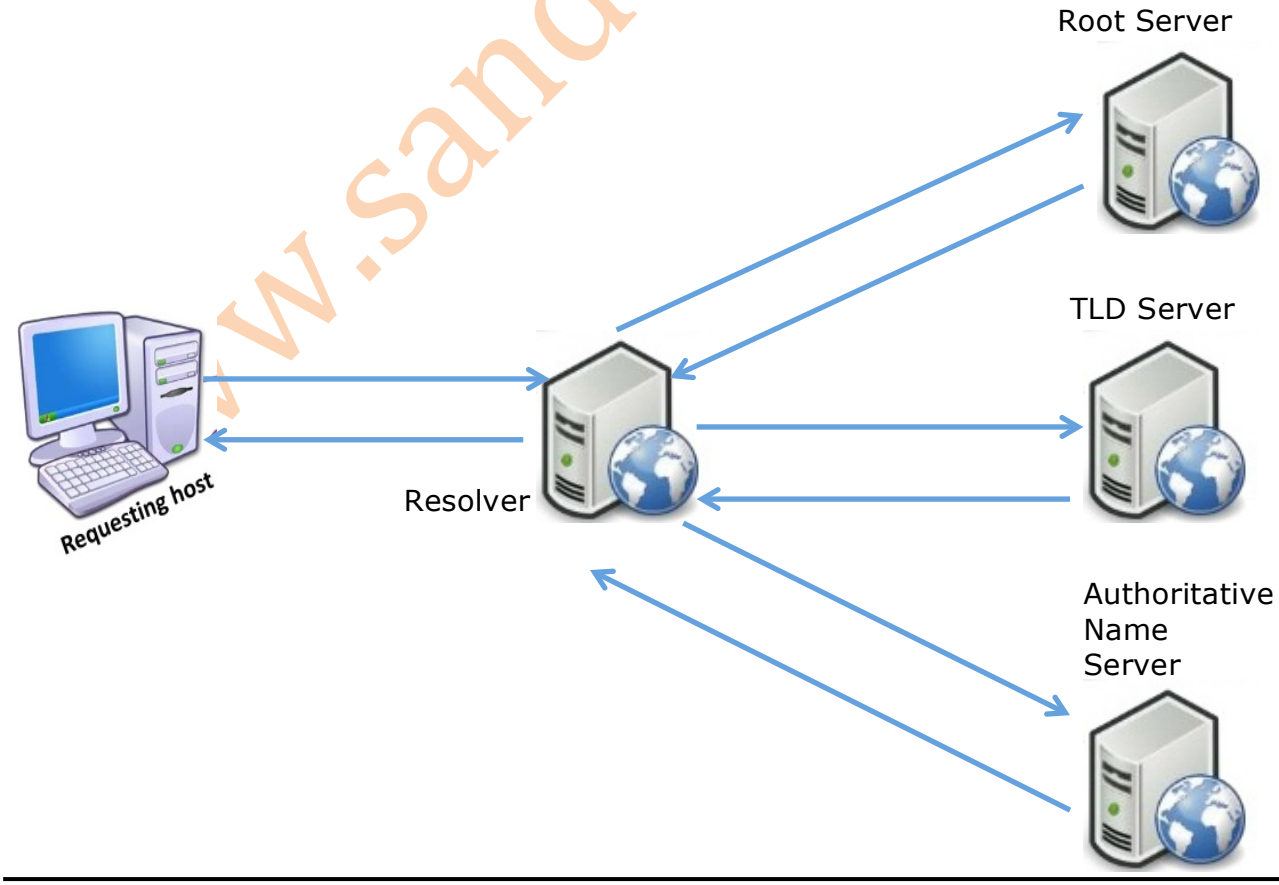
Step 6: Retrieve the IP Address

The Resolver receives the IP address from the Authoritative Name Server and stores the it in its local cache for future use. If anyone else requests IP address for sandeepgupta.org, the Resolver will already have the answer and will not need to go through the lookup process again. All IP addresses locally stored have a time-to-live value, which is like an expiration date. After a while, the Resolver will need to ask for a new copy of the IP address to make sure the information doesn't become out-of-date.

Step 7: Receive the answer

Armed with the answer, the Resolver returns the IP address to your computer. Your computer stores the IP address in its Local DNS cache and then passes this IP address to your browser. The browser then opens a connection to the webserver and receives the website.

This entire process, from start to finish, takes only milliseconds to complete.



Overview of HTTP & FTP

A protocol is a set of rules used by computers that are connected or networked together. These rules specify how the computers should communicate with each other.

1) HTTP

The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed systems. This is the foundation for data communication for the World Wide Web (i.e. internet) since 1990.

HTTP is a TCP/IP based communication protocol, that is used to deliver data (HTML files, image files, query results, etc.) on the World Wide Web. The default port is TCP 80, but other ports can be used as well. It provides a standardized way for computers to communicate with each other. HTTP specification specifies how clients' request data will be constructed and sent to the server, and how the servers respond to these requests.

Basic Features

There are three basic features that make HTTP a simple but powerful protocol.

HTTP is connectionless: The HTTP client, i.e., a browser initiates an HTTP request and after a request is made, the client disconnects from the server and waits for a response. The server processes the request and re-establishes the connection with the client to send a response back.

HTTP is media independent: It means, any type of data can be sent by HTTP as long as both the client and the server know how to handle the data content. It is required for the client as well as the server to specify the content type using appropriate MIME (Multipurpose Internet Mail Extensions) type.

HTTP is stateless: As mentioned above, HTTP is connectionless and it is a direct result of HTTP being a stateless protocol. The server and client are aware of each other only during a current request. Afterwards, both of them forget about each other. Due to this nature of the protocol, neither the client nor the browser can retain information between different requests across the web pages.

HTTPS: HTTPS means Hyper Text Transfer Protocol Secure. Basically, it is the secure version of HTTP. In https, communications between the browser and website are encrypted by Transport Layer Security (TLS) or its predecessor Secure Sockets Layer (SSL).

2) FTP

File Transfer Protocol (FTP) is the commonly used protocol for exchanging files over the Internet. FTP uses the Internet's TCP/IP protocols to enable data transfer.

FTP uses a client-server architecture. Users provide authentication using a sign-in protocol, usually a username and password, however some FTP servers may be configured to accept anonymous FTP logins where you don't need to identify yourself before accessing files. Most often, FTP is secured with SSL/TLS.

How to FTP ??
Files can be transferred between two computers using FTP software. The user's computer is called the local host machine and is connected to the Internet. The second machine, called the remote host, is also running FTP software and connected to the Internet.
The local host machine connects to the remote host's IP address.
The user would enter a username/password (or use anonymous).
FTP software may have a GUI, allowing users to drag and drop files between the remote and local host. If not, a series of FTP commands are used to log in to the remote host and transfer files between the machines.

Common Uses of FTP
FTP is most commonly used to download a file from a server using the Internet or to upload a file to a server (e.g. uploading a web page file to a Web server).