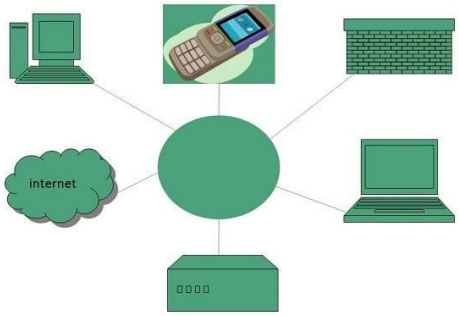
UNIT - 1

Internet

Internet is defined as an Information super Highway, to access information over the web. However, it can be defined in many ways as follows:

- Internet is a world-wide global system of interconnected computer networks.
- Internet uses the standard Internet Protocol (TCP/IP).
- Every computer in internet is identified by a unique IP address.
- IP Address is a unique set of numbers (such as 110.22.33.114) which identifies a computer location.
- A special computer DNS (Domain Name Server) is used to give name to the IP Address so that user can locate a computer by a name.
- For example, a DNS server will resolve a name **http://www.tutorialspoint.com** to a particular IP address to uniquely identify the computer on which this website is hosted.
- Internet is accessible to every user all over the world.



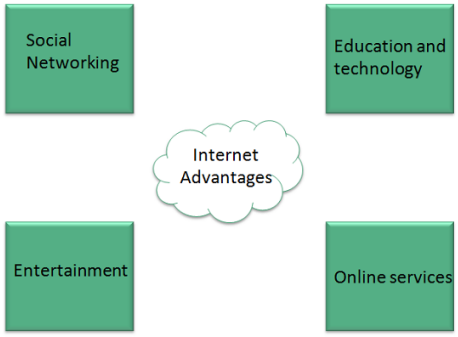
Evolution

The concept of Internet was originated in 1969 and has undergone several technological & Infrastructural changes as discussed below:

- The origin of Internet devised from the concept of **Advanced Research Project Agency Network (ARPANET).**
- **ARPANET** was developed by United States Department of Defense.
- Basic purpose of ARPANET was to provide communication among the various bodies of government.
- Initially, there were only four nodes, formally called **Hosts.**
- In 1972, the **ARPANET** spread over the globe with 23 nodes located at different countries and thus became known as **Internet.**
- By the time, with invention of new technologies such as TCP/IP protocols, DNS, WWW, browsers, scripting languages etc., Internet provided a medium to publish and access information over the web.

Advantages

Internet covers almost every aspect of life, one can think of. Here, we will discuss some of the advantages of Internet:

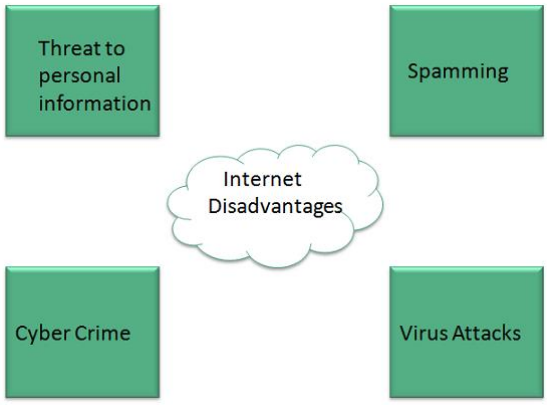


- Internet allows us to communicate with the people sitting at remote locations. There are various apps available on the wed that uses Internet as a medium for communication. One can find various social networking sites such as:
  - Facebook
  - Twitter
  - Yahoo
  - Google+
  - Flickr
  - Orkut
- One can surf for any kind of information over the internet. Information regarding various topics such as Technology, Health & Science, Social Studies, Geographical Information, Information Technology, Products etc can be surfed with help of a search engine.
- Apart from communication and source of information, internet also serves a medium for entertainment. Following are the various modes for entertainment over internet.
  - Online Television
  - Online Games
  - Songs
  - Videos
  - Social Networking Apps
- Internet allows us to use many services like:
  - Internet Banking
  - Matrimonial Services
  - Online Shopping
  - Online Ticket Booking
  - Online Bill Payment
  - Data Sharing
  - E-mail

- Internet provides concept of **electronic commerce**, that allows the business deals to be conducted on electronic systems

Disadvantages

However, Internet has proved to be a powerful source of information in almost every field, yet there exists many disadvantages discussed below:

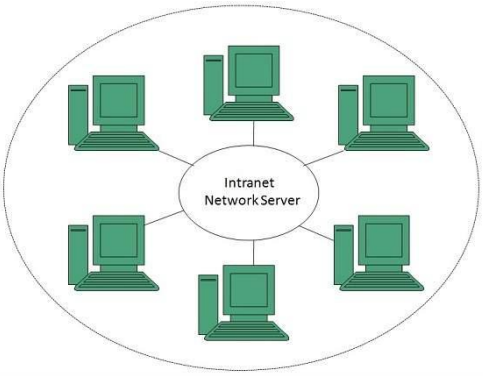


- There are always chances to lose personal information such as name, address, credit card number. Therefore, one should be very careful while sharing such information. One should use credit cards only through authenticated sites.
- Another disadvantage is the **Spamming**. Spamming corresponds to the unwanted e-mails in bulk. These e-mails serve no purpose and lead to obstruction of entire system.
- **Virus** can easily be spread to the computers connected to internet. Such virus attacks may cause your system to crash or your important data may get deleted.
- Also a biggest threat on internet is pornography. There are many pornographic sites that can be found, letting your children to use internet which indirectly affects the children healthy mental life.
- There are various websites that do not provide the authenticated information. This leads to misconception among many people.

Intranet

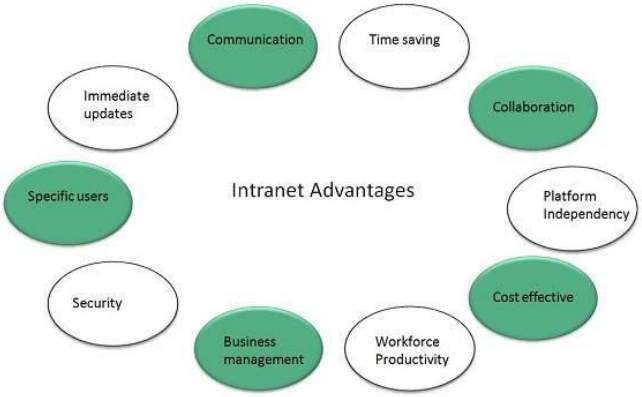
Intranet is defined as private network of computers within an organization with its own server and firewall. Moreover we can define Intranet as:

- Intranet is system in which multiple PCs are networked to be connected to each other. PCs in intranet are not available to the world outside of the intranet.
- Usually each company or organization has their own Intranet network and members/employees of that company can access the computers in their intranet.
- Every computer in internet is identified by a unique IP address.
- Each computer in Intranet is also identified by a IP Address, which is unique among the computers in that Intranet.



Benefits

Intranet is very efficient and reliable network system for any organization. It is beneficial in every aspect such as collaboration, cost-effectiveness, security, productivity and much more.



Communication

Intranet offers easy and cheap communication within an organization. Employees can communicate using chat, e-mail or blogs.

Time Saving

Information on Intranet is shared in real time.

Collaboration

Information is distributed among the employees as according to requirement and it can be accessed by the authorized users, resulting in enhanced teamwork.

Platform Independency

Intranet can connect computers and other devices with different architecture.

Cost Effective

Employees can see the data and other documents using browser rather than printing them and distributing duplicate copies among the employees, which certainly decreases the cost.

Workforce Productivity

Data is available at every time and can be accessed using company workstation. This helps the employees work faster.

Business Management

It is also possible to deploy applications that support business operations.

Security

Since information shared on intranet can only be accessed within an organization, therefore there is almost no chance of being theft.

Specific Users

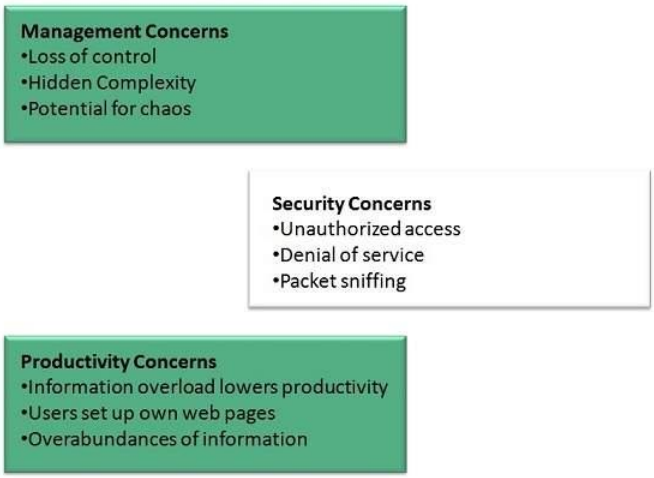
Intranet targets only specific users within an organization therefore, once can exactly know whom he is interacting.

Immediate Updates

Any changes made to information are reflected immediately to all the users.

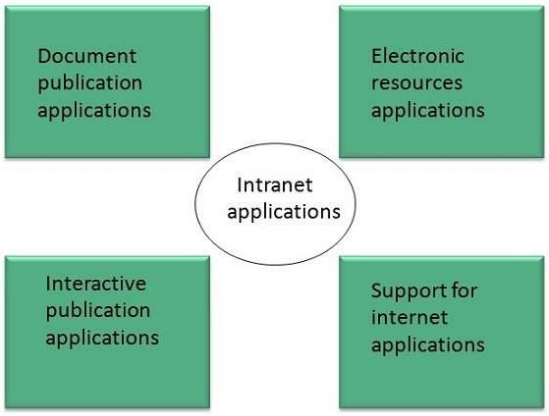
Issues

Apart from several benefits of Intranet, there also exist some issues.. These issues are shown in the following diagram:



Applications

Intranet applications are same as that of Internet applications. Intranet applications are also accessed through a web browser. The only difference is that, Intranet applications reside on local server while Internet applications reside on remote server. Here, we've discussed some of these applications:



Document publication applications

Document publication applications allow publishing documents such as manuals, software guide, employee profits etc without use of paper.

Electronic resources applications

It offers electronic resources such as software applications, templates and tools, to be shared across the network.

Interactive Communication applications

Like on internet, we have e-mail and chat like applications for Intranet, hence offering an interactive communication among employees.

Support for Internet Applications

Intranet offers an environment to deploy and test applications before placing them on Internet.

Internet vs. Intranet

Apart from similarities there are some differences between the two. Following are the differences between Internet and Intranet:

Intranet	Internet
Localized Network.	Worldwide Network
Doesn't have access to Intranet	Have access to Internet.
More Expensive	Less Expensive
More Safe	Less Safe
More Reliability	Less Reliability

World Wide Web

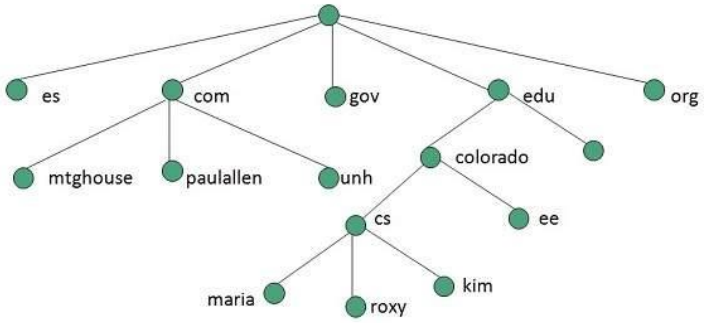
WWW stands for **World Wide Web.** A technical definition of the World Wide Web is : all the resources and users on the Internet that are using the Hypertext Transfer Protocol (HTTP).

A broader definition comes from the organization that Web inventor **Tim Berners-Lee** helped found, the **World Wide Web Consortium (W3C).**

The World Wide Web is the universe of network-accessible information, an embodiment of human knowledge.

In simple terms, The World Wide Web is a way of exchanging information between computers on the Internet, tying them together into a vast collection of interactive multimedia resources.

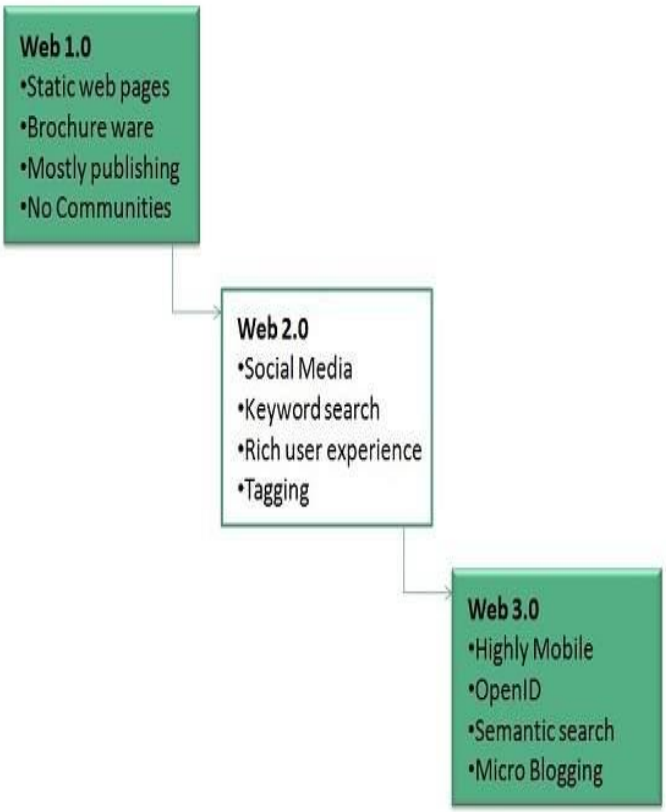
Internet and **Web** is not the same thing: Web uses internet to pass over the information.



Evolution

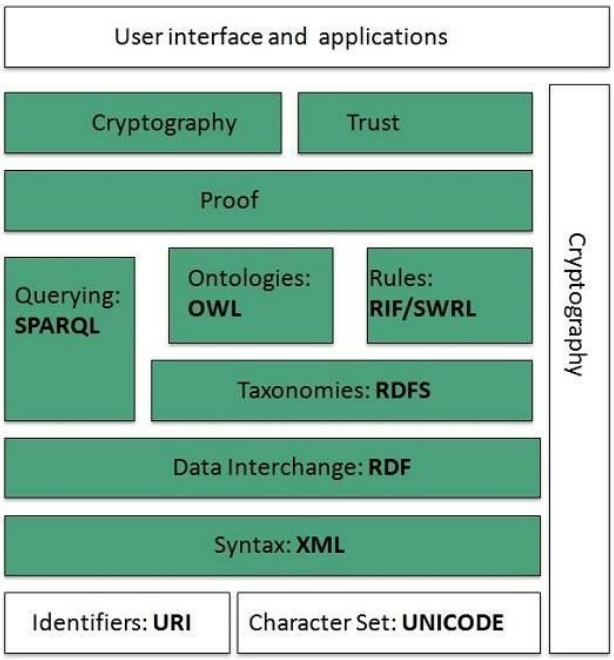
World Wide Web was created by **Timothy Berners Lee** in 1989 at **CERN** in **Geneva.** World Wide Web came into existence as a proposal by him, to allow researchers to work together effectively and efficiently at **CERN.** Eventually it became **World Wide Web.**

The following diagram briefly defines evolution of World Wide Web:



WWW Architecture

WWW architecture is divided into several layers as shown in the following diagram:



Identifiers and Character Set

Uniform Resource Identifier (URI) is used to uniquely identify resources on the web and **UNICODE** makes it possible to built web pages that can be read and write in human languages.

Syntax

XML (Extensible Markup Language) helps to define common syntax in semantic web.

Data Interchange

Resource Description Framework (RDF) framework helps in defining core representation of data for web. RDF represents data about resource in graph form.

Taxonomies

RDF Schema (RDFS) allows more standardized description of **taxonomies** and other **ontological** constructs.

Ontologies

Web Ontology Language (OWL) offers more constructs over RDFS. It comes in following three versions:

- OWL Lite for taxonomies and simple constraints.
- OWL DL for full description logic support.
- OWL for more syntactic freedom of RDF

Rules

RIF and **SWRL** offers rules beyond the constructs that are available from **RDFs** and **OWL.** Simple Protocol and **RDF Query Language (SPARQL)** is SQL like language used for querying RDF data and OWL Ontologies.

Proof

All semantic and rules that are executed at layers below Proof and their result will be used to prove deductions.

Cryptography

Cryptography means such as digital signature for verification of the origin of sources is used.

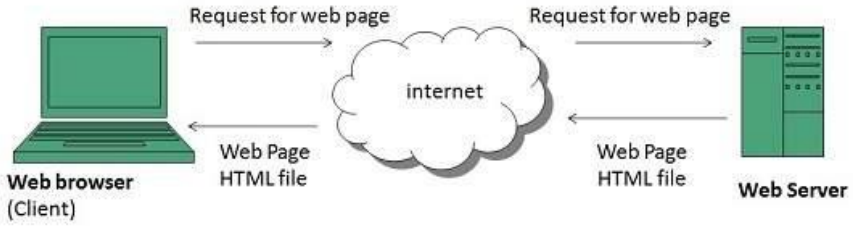
User Interface and Applications

On the top of layer **User interface and Applications** layer is built for user interaction.

WWW Operation

WWW works on client- server approach. Following steps explains how the web works:

1. User enters the URL (say, **http://www.tutorialspoint.com**) of the web page in the address bar of web browser.
2. Then browser requests the Domain Name Server for the IP address corresponding to www.tutorialspoint.com.
3. After receiving IP address, browser sends the request for web page to the web server using HTTP protocol which specifies the way the browser and web server communicates.
4. Then web server receives request using HTTP protocol and checks its search for the requested web page. If found it returns it back to the web browser and close the HTTP connection.
5. Now the web browser receives the web page, It interprets it and display the contents of web page in web browser's window.



Future

There had been a rapid development in field of web. It has its impact in almost every area such as education, research, technology, commerce, marketing etc. So the future of web is almost unpredictable.

Apart from huge development in field of WWW, there are also some technical issues that W3 consortium has to cope up with.

User Interface

Work on higher quality presentation of 3-D information is under deveopment. The W3 Consortium is also looking forward to enhance the web to full fill requirements of global communities which would include all regional languages and writing systems.

Technology

Work on privacy and security is under way. This would include hiding information, accounting, access control, integrity and risk management.

Architecture

There has been huge growth in field of web which may lead to overload the internet and degrade its performance. Hence more better protocol are required to be developed.

Web Page

Web page is a document available on world wide web. Web Pages are stored on web server and can be viewed using a web browser.

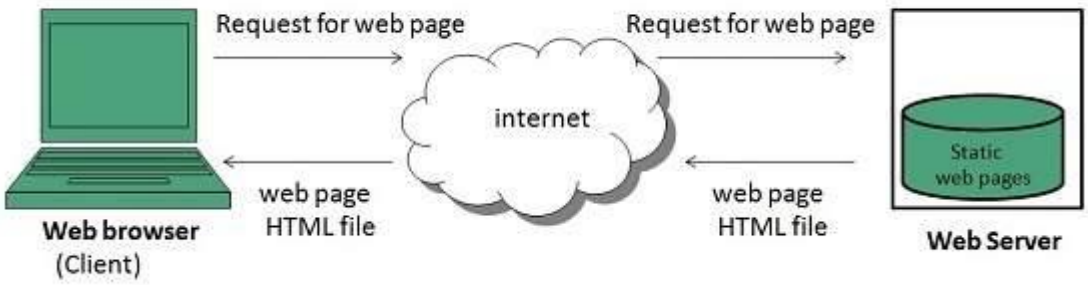
A web page can contain huge information including text, graphics, audio, video and hyper links. These hyper links are the link to other web pages.

Collection of linked web pages on a web server is known as **website.** There is unique **Uniform Resource Locator (URL)** is associated with each web page.

Static Web page

Static web pages are also known as flat or stationary web page. They are loaded on the client's browser as exactly they are stored on the web server. Such web pages contain only static information. User can only read the information but can't do any modification or interact with the information.

Static web pages are created using only HTML. Static web pages are only used when the information is no more required to be modified.



Dynamic Web page

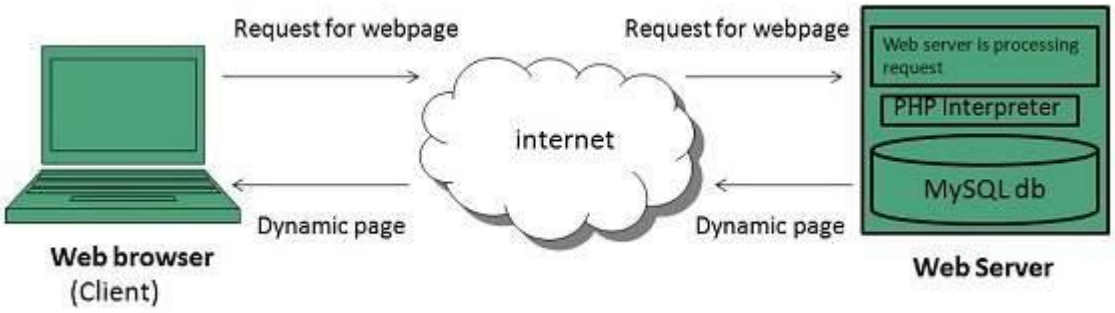
Dynamic web page shows different information at different point of time. It is possible to change a portaion of a web page without loading the entire web page. It has been made possible using **Ajax** technology.

Server-side dynamic web page

It is created by using server-side scripting. There are server-side scripting parameters that determine how to assemble a new web page which also include setting up of more client-side processing.

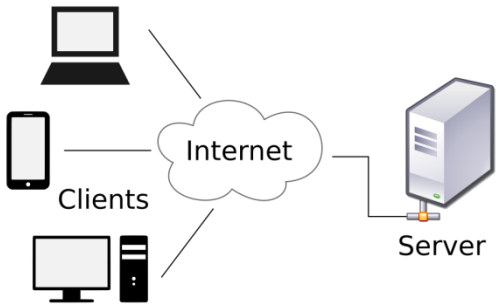
Client-side dynamic web page

It is processed using client side scripting such as JavaScript. And then passed in to **Document Object Model (DOM).**



Web Client

A web client is an application program or web browser that communicates with web server using Hypertext transfer protocol. Web client acts as an interface between server and clients, through which messages are sent to web server. It collects the processed data from the web server. E.g. Google chrome, Internet explorer, opera, firefox, safari.



Roles of web client:

- It acts as an interface between server and client and displays web document to the client.
- Web client requests the server for the webpages and resources.
- It sends HTTP requests and get HTTP response.
- It stores the cookies for different websites.
- It helps to bookmark the websites.

Web Server

Overview

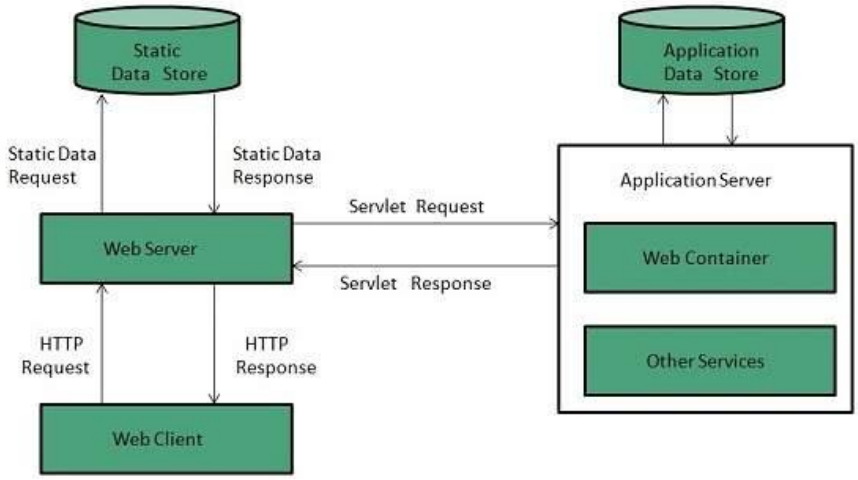
Web server is a computer where the web content is stored. Basically web server is used to host the web sites but there exists other web servers also such as gaming, storage, FTP, email etc.

Web site is collection of web pages whileweb server is a software that respond to the request for web resources.

Web Server Working

Web server respond to the client request in either of the following two ways:

- Sending the file to the client associated with the requested URL.
- Generating response by invoking a script and communicating with database



Key Points

- When client sends request for a web page, the web server search for the requested page if requested page is found then it will send it to client with an HTTP response.
- If the requested web page is not found, web server will the send an **HTTP response:Error 404 Not found.**
- If client has requested for some other resources then the web server will contact to the application server and data store to construct the HTTP response.

Architecture

Web Server Architecture follows the following two approaches:

1. Concurrent Approach
2. Single-Process-Event-Driven Approach.

Concurrent Approach

Concurrent approach allows the web server to handle multiple client requests at the same time. It can be achieved by following methods:

- Multi-process
- Multi-threaded
- Hybrid method.

Multi-processing

In this a single process (parent process) initiates several single-threaded child processes and distribute incoming requests to these child processes. Each of the child processes are responsible for handling single request.

It is the responsibility of parent process to monitor the load and decide if processes should be killed or forked.

Multi-threaded

Unlike Multi-process, it creates multiple single-threaded process.

Hybrid

It is combination of above two approaches. In this approach multiple process are created and each process initiates multiple threads. Each of the threads handles one connection. Using multiple threads in single process results in less load on system resources.

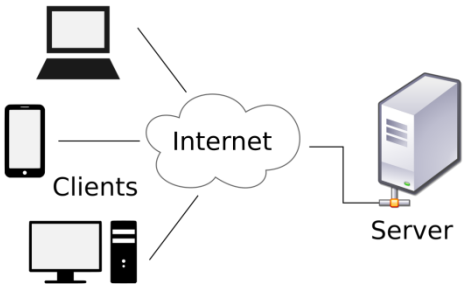
Client Server Architecture

Client/Server Model

The client–server model is a computing model that acts as distributed application which partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients. Often clients and servers communicate over a computer network on separate hardware, but both client and server may reside in the same system. A server machine is a host that is running one or more server programs which share their resources with lients. A client does not share any of its resources, but requests a server's content or service function. Clients therefore initiate communication sessions with servers which await incoming requests.

Client/Server Architecture

Client server network architecture consists of two kinds of computers: clients and servers. Clients are the computers that that do not share any of its resources but requests data and other services from the server computers and server computers provide services to the client computers by responding to client computers requests. Normally servers are powerful computers and clients are less powerful personal computers. Web servers are included as part of a larger package of internet and intranet related programs for serving e- mail, downloading requests for FTP files and building and publishing web pages.



Advantages

- The client/ server architecture reduces network traffic by providing a query response to the user rather than transferring total files.
- The client/ server model improves multi-user updating through a graphical user interface (GUI) front end to the shared database.
- Easy to implement security policies, since the data are stored in central location
- Simplified network administration

Disadvantages

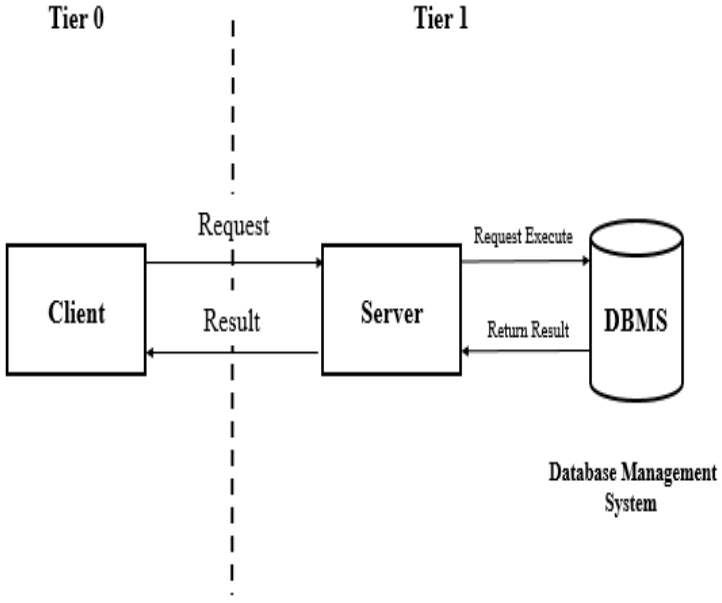
- Failure of the server causes whole network to be collapsed
- Expensive than P2P, Dedicated powerful servers are needed
- Extra effort are needed for administering and managing the server.

Client/Server architecture can be of different model based on the number of layers it holds. Some of them are;

2-Tier Architecture

2-tier architecture is used to describe client/server systems where the client requests resources and the server responds directly to the request, using its own resources. This means that the server does not call on another application in order to provide part of the service. It runs the client processes separately from the server processes, usually on a different computer:

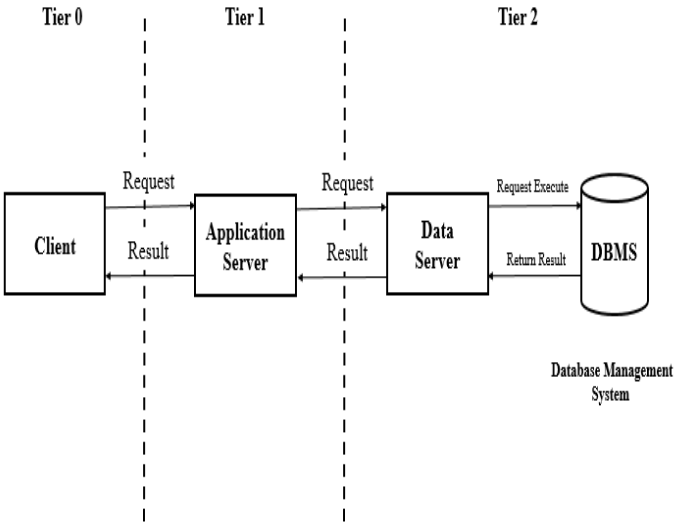
- The client processes provide an interface for the customer, and gather and present data usually on the customer's computer. This part of the application is the presentation layer
- – The server processes provide an interface with the data store of the business. This part of the application is the data layer
- The business logic that validates data, monitors security and permissions, and performs other business rules can be housed on either the client or the server, or split between the two.
    - Fundamental units of work required to complete the business process
    - Business rules can be automated by an application program.



3-Tier Architecture

In 3-tier architecture, there is an intermediary level, meaning the architecture is generally split up between:

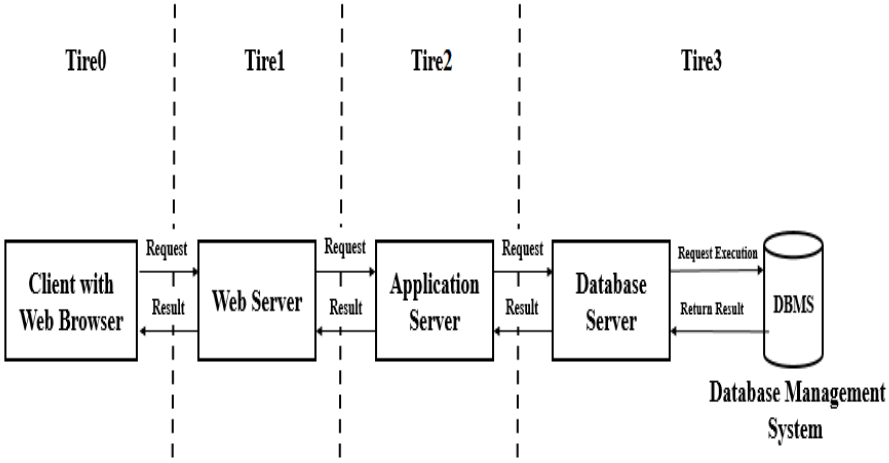
- A client, i.e. the computer, which requests the resources, equipped with a user interface (usually a web browser) for presentation purposes
- The application server (also called middleware), whose task it is to provide the requested resources, but by calling on another server
- The data server, which provides the application server with the data it requires



N-Tier Architecture (multi-tier)

N-tier architecture (with N more than 3) is really 3 tier architectures in which the middle tier is split up into new tiers. The application tier is broken down into separate parts. What these parts are differs from system to system. The following picture shows it:

The primary advantage of N-tier architectures is that they make load balancing possible. Since the application logic is distributed between several servers, processing can then be more evenly distributed among those servers. N-tiered architectures are also more easily scalable, since only servers experiencing high demand, such as the application server, need be upgraded. The primary disadvantage of N-tier architectures is that it is also more difficult to program and test an N-tier architecture due to its increased complexity.



Advantages of Multi-Tier Client/Server architectures include:

- Changes to the user interface or to the application logic are largely independent from one another, allowing the application to evolve easily to meet new requirements.
- Network bottlenecks are minimized because the application layer does not transmit extra data to the client, only what is needed to handle a task.
- The client is insulated from database and network operations. The client can access data easily and quickly without having to know where data is or how many servers are on the system.
- Database connections can be 'pooled' and thus shared by several users, which greatly reduces the cost associated with per-user licensing.
- The organization has database independence because the data layer is written using standard SQL which is platform independent. The enterprise is not tied to vendor-specific stored procedures.
- The application layer can be written in standard third or fourth generation languages, such as ASP, PHP with which the organization's in-house programmers are experienced.

What kind of systems can benefit?

Generally, any Client/Server system can be implemented in an 'N-Tier' architecture, where application logic is partitioned among various servers. This application partitioning creates an integrated information infrastructure which enables consistent, secure, and global access to critical data. A significant reduction in network traffic, which leads to faster network communications, greater reliability, and greater overall performance is also made possible in a 'N-Tier' Client/Server architecture.

The HTTP Request and Response

The HTTP protocol can be likened to a conversation based on a series of questions and answers, which we refer to respectively as *HTTP requests* and *HTTP responses*.

The contents of HTTP requests and responses are easy to read and understand, being near to plain English in their syntax.

This section examines the structure of these requests and responses, along with a few examples of the sorts of data they may contain.

The HTTP Request

After opening a connection to the intended server, the HTTP client transmits a request in the following format:

- An opening line
- Optionally, a number of *header lines*
- A blank line
- Optionally, a message body

The opening line is generally split into three parts; the name of the *method*, the path to the required *server resource*, and the *HTTP version* being used. A typical opening line might read:

```
GET /sams/testpage.html HTTP/1.0
```

In this line we are telling the server that we are sending an HTTP request of type `GET` (explained more fully in the next section), we are sending this using HTTP version 1.0, and the server resource we require (including its local path) is

```
/sams/testpage.html.
```

Note

In this example the server resource we seek is on our own server, so we have quoted a relative path. It could of course be on another server elsewhere, in which case the server resource would include the full URL.

Header lines are used to send information about the request, or about the data being sent in the message body. One parameter and value pair is sent per line, the parameter and value being separated by a colon. Here's an example:

```
User-Agent: [name of program sending request]
```

For instance, Internet Explorer v5.5 offers something like the following:

```
User-agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0)
```

A further example of a common request header is the `Accept:` header, which states what sort(s) of information will be found acceptable as a response from the server:

```
Accept: text/plain, text/html
```

By issuing the header in the preceding example, the request is informing the server that the sending application can accept either plain text or HTML responses (that is, it is not equipped to deal with, say, an audio or video file).

Note

HTTP request methods include `POST`, `GET`, `PUT`, `DELETE`, and `HEAD`. By far the most interesting in our pursuit of Ajax are the `GET` and `POST` requests. The `PUT`, `DELETE`, and `HEAD` requests are not covered here.

The HTTP Response

In answer to such a request, the server typically issues an HTTP response, the first line of which is often referred to as the *status line*. In that line the server echoes the HTTP version and gives a response status code (which is a three-digit integer) and a short message known as a *reason phrase*. Here's an example HTTP response:

```
HTTP/1.0 200 OK
```

The response status code and reason phrase are essentially intended as machine-and human-readable versions of the same message, though the reason phrase may actually vary a little from server to server. Table 3.1 lists some examples of common status codes and reason

phrases. The first digit of the status code usually gives some clue about the nature of the message:

- 1**—Information
- 2**—Success
- 3**—Redirected
- 4**—Client error
- 5**—Server error

Some Commonly Encountered HTTP Response Status Codes

Status Code	Explanation
`200 - OK`	The request succeeded.
`204 - No Content`	The document contains no data.
`301 - Moved Permanently`	The resource has permanently moved to a different URI.
`401 - Not Authorized`	The request needs user authentication.
`403 - Forbidden`	The server has refused to fulfill the request.
`404 - Not Found`	The requested resource does not exist on the server.
`408 - Request Timeout`	The client failed to send a request in the time allowed by the server.
`500 - Server Error`	Due to a malfunctioning script, server configuration error or similar.

URL: Uniform Resource Locator

URL stands for Uniform Resource Locator. It is the address of a resource, which can be a specific webpage or a file, on the internet. It is also *known as web address* when it is used with http. It was created in 1994 by Tim Berners-Lee. URL is a specific character string that is used to access data from the World Wide Web. It is a type of URI (Uniform Resource Identifier).

Every URL contains the following information:

- The scheme name or protocol.
- A colon, two slashes.
- A host, normally called a domain name but sometimes as a literal IP address.
- A colon followed by a port number.
- Full path of the resource.

The URL of a web page is displayed above on the page in the address bar. A typical URL looks like this:

http://www.javatpoint.com/full-form

The above URL contains:

- **protocol**: http
- **host or domain**: www.javatpoint.com
- **Path of the resource**: /full-form

A URL can be entered manually by typing it in the address bar of your web browser. If the URL does not contain a valid server, a browser may display a "Server not found" error and if the path in the URL is incorrect, the browser may display a "404 error". A URL does not contain spaces and uses forward slashes to represent different directories. So, dashes and underscores are used separate the words of a web address.

What is URI

URI stands for Uniform Resource Identifier. It is a generic term for all the name and addresses which show objects on the World Wide Web. It is generally a sequence of characters which identifies a logical resource or the name and location of a file or resource in a uniform format.

A URI can be of two types: Uniform Resource Locator (URL) and Uniform Resource Names (URNs). It enables resources to be accessed by other computers across a network or over the World Wide Web.

Server-side Scripting

Server-side scripting is a programming technique for creating code that may run software on the server side. In other words, server-side scripting is any scripting method that may operate on a web server. At the server end, actions such as website customization, dynamic changes in website content, response creation to user requests, database access, and many more are carried out.

Server-side scripting creates a communication channel between a server and a client. Previously, ***CGI (Common Gateway Interface)*** scripts were used to implement server-side scripting, and CGI was created to execute scripts written in computer languages such as C++ or Perl on websites.

The server-side is made up of ***three*** parts: ***the database, the server, the APIs,*** and the backend web software written in the server-side scripting language. When a browser requests a page with server-side scripting, the web server evaluates the script before delivering the page to the browser. In this case, script processing may entail collecting information from a database, performing simple computations, or selecting the relevant material to be shown on the client end. The output is provided to the web browser when the script is processed. The web server hides the scripts from the end user until the content is delivered, making the data and source code safer.

Server-side scripting languages

There are various server-side scripting languages. Some main server-side scripting languages are as follows:

Python

It is an open-source language that is very powerful and easy to learn. It is suitable for beginners because it is simple to learn and read. It is believed to be used by Google and YouTube. It is a OOPs language with dynamic typing and data structures. It has grown to be one of the most popular languages for both quick application development and web development.

PHP

It is an open-source server-side scripting programming language mainly designed for web apps and is the most utilized scripting language. It allows you to retrieve and manipulate data from a database and is utilized along with SQL to query the database. It is a fast and simple language to learn and develop, and Facebook, Wikipedia, and WordPress utilize it.

Ruby

It is a free and open-source programming language that was developed and firstly introduced in the early 1990s. It is a dynamic language that is simple to read and write and an OOPs language that is interpreted as it runs. It has evolved continuously since its development and is one of the most utilized web development languages.

Features of Server-side Scripting

There are various features of server-side scripting. Some main features of the server-side scripting are as follows:

1. It is connected with data access, error handling, and data processing speed.
2. It is processed and interacts with the server.
3. Using a highly integrated programming language makes it more secure than client-side scripting.

What is Client-side Scripting?

Client-side scripting generates code that may be executed on the client end without needing server-side processing. These scripts are typically embedded into HTML text. Client-side scripting may be utilized to check the user's form for problems before submitting it and to change the content based on the user input. The web needs three components to function: client, database, and server.

The client-side scripting may significantly reduce server demand. It is intended to be utilized as a scripting language with a web browser as the host program. The ***HTML*** and ***CSS*** are delivered as plain text when a user uses a browser to request a webpage from the server, and the browser understands and renders the web content at the client end.

Client-side Scripting Languages

There are various client-side scripting languages. Some main client-side scripting languages are as follows:

HTML

It is not a scripting language; it is a markup language. However, it serves as the basic language for client-side web development, also referred to as front-end. The presence of hypertext on a page denotes its hyperlinks. The markup language uses tags to define the structure and layout. It is a programming language that is mainly used to design a web page's structure and layout.

CSS

CSS is an abbreviation for Cascading Style Sheets. It provides a technique for creating graphic elements that help a web application's appearance look more appealing. A style tag in a web page defines all the specifics regarding the web page's presentation, including its border styles, image styles, colour, font styles, borders, format, font size, margins, padding, etc.

JavaScript

It is a client-side scripting language designed for a specific purpose, but several JavaScript frameworks are already utilized as server-side scripting technologies.

VBScript

VBScript is based on Visual Basic, which was created by Microsoft in 1996. It is a scripting programming language that is lightweight, fast, and easy to learn. It is not a OOPs language but is similar to JavaScript.

Features of Client-side Scripting

There are various features of client-side scripting. Some main features of the client-side scripting are as follows:

1. It is intended to execute code on which a web browser runs, and the results of the inputs are delivered to an accessible user.
2. Client-side scripting enables greater involvement with clients via the browser and is used to validate programs and functionality based on the request.
3. The client does not include any contact with the server in client-side scripting; the only interaction is receiving the requested data.

Difference between Server-side Scripting and Client-side Scripting

Features	Server-side Scripting	Client-side Scripting
Primary Function	The main function of this scripting is to manipulate and grant access to the requested database.	The main purpose of this scripting is to give the requested output to the end-user.
Uses	It is employed at the backend, where the source code is invisible or concealed on the client side.	It is utilized at the front end, which users may view through the browser.
Processing	It needs server interaction.	It doesn't need any server interaction.
Security	It is more secure while working on a web app.	It is less secure than server-side scripting due to the code accessibility offered to the client.
Running	It executes on the web server.	It executes on the remote

		computer system.
Dependability	It doesn't depend on the client.	It depends on the user's browser version.
File Access	It offers complete access to the file that is stored in the web database server.	It doesn't offer any access to the files on the web servers.
Code Allowance	It enables the backend developer to hide the source code from the user.	The user is given access to the written code after confirming their requirements.
Occurrence	It only responds after the user begins the browsing request.	It happens when the browser processes all of the codes and then acts according to the client's needs.
Affect	It may reduce the server load.	It may effectively customize web pages and offer dynamic websites.
Languages Involved	The server-side scripting programming languages, such as PHP, ColdFusion, Python, ASP.net, Java, C++, Ruby, C#, etc.	Its programming languages are HTML, CSS, and JavaScript.

What is Web 1.0?

Web 1.0 is all about reading and providing your own information. Web 1.0 is primarily used in the first stage of worldwide evolution, in which the majority of users are consumers. Static web pages are the most common way to connect millions of people around the globe. Web 1.0 provides us with the feature for the content delivery network (CDN), which enables the piece of information on the websites. It can be used to provide personal information. The cost for viewing is per page viewed. It has database directories to enable users to retrieve a particular piece of information. The era of Web 1.0 started roughly in 1989 and ended in 2005.

What is Web 2.0?

Web 2.0 is all about interacting with the user, along with reading, writing, and creating to a certain extent on the programmable web. It is a worldwide website that highlights user

usability and interoperability for the end user. AJAX and JavaScript have been very popular for creating Web 2.0. The tools that are used in Web 2.0 are blogging, tagging, social networking, podcasting, etc. It started roughly in 1999 and ended in 2012.

Difference between Web 1.0, Web 2.0

	Web 1.0	Web 2.0
1.	It connects real people to the world wide web (www).	It connects real people who use the World Wide Web (www).
2.	It is about providing your own information.	It is about the overload or sharing of information.
3.	Web 1.0 is generally referred to as the "read-only web" or "visual/interactive web," making content available online for viewing.	Users can read, write, and collaborate on the Programmable Web to a certain extent.
4.	Internet users are restricted to reading content from the web.Examples: email, P2P file sharing, content, file or web servers, search engines, and business portals.	It enhances the interactivity of the user's Web service endpoint with an add-on published feature. **Examples** are Instant messaging, Ajax and JavaScript frameworks, and Adobe Flex.
5.	Examples of the web era include the millions of homepages with static websites and DOT COM bubbles.	Examples are Twitter, Flickr, Facebook, and YouTube for creating wikis and blogs.
6.	Web 1.0 is commonly used in HTML, HTTP, URLs, and portals.	Web 2.0 is commonly used in XML and RSS.
7.	It is mainly focused on the company.	It is mainly focused on the community.
8.	It started in 1989-2005.	It started in 1999-2012.
9.	It is like a banner advertisement.	It is interactive advertising.
10.	Web 1.0 is used in Encyclopedia Britannica online.	Web 2.0 is used in Wikipedia.